A PINN-Based Friction-Inclusive Dynamics Modeling Method for Industrial Robots

Hongbo Hu, Zhikai Shen, and Chungang Zhuang, *Member, IEEE*

***Abstract*—High-precision dynamics and friction models are crucial for high-performance control and operation of industrial robots. However, due to the requirement for model linearization, mainstream identification-based modeling methods struggle to capture nonlinear features of the model. In recent years, physics-informed neural network (PINN)-based methods have achieved interpretable nonlinear robotic dynamics and friction modeling, but suffer from suboptimal accuracy due to the lack of comprehensive modeling and learning strategies. This article presents a PINN-based friction-inclusive dynamics modeling method for industrial robots. A hybrid learning strategy for robot dynamics and friction is designed, ensuring modeling accuracy while avoiding reliance on joint torque component labels. Furthermore, residual error compensation is integrated into the proposed PINN to enhance its capability to learn nonlinear features. Experimental validation on two different robots demonstrates the effectiveness of the proposed method. Compared with other advanced methods, the average joint torque error is reduced by an average of 390.69%.**

***Index Terms*—Dynamics modeling, friction modeling, hybrid learning strategies, industrial robot, physics informed neural networks (PINNs).**

I. Introduction

With the advancement of robotics technology, robots have found widespread applications across various fields, raising higher performance requirements. High-precision dynamics and friction models are crucial for the high-performance control and operation of industrial robots. Parameter identification methods have become popular in robot dynamics modeling in recent years, using algorithms like least squares to regress dynamics parameters of robot joints, thereby obtaining inverse dynamics models [1], [2]. On the other hand, the rapid development of deep learning (DL) and its application to many robotic tasks have stimulated the advancement of learning-based dynamics and friction modeling techniques [3], [4]. However, due to the inherently complex nonlinearities of robotic models, existing learning-based methods still struggle to improve modeling accuracy.

Identification-based modeling methods typically employ the recursive Newton-Euler formulation to construct inverse dynamics models based on rigid body assumptions, achieving high modeling accuracy. This process usually includes dynamics model linearization, minimum inertial parameter set extraction, excitation trajectory design, and dynamics parameter identification. In recent years, numerous researchers have proposed various improved identification methods. Jubien et al. [5] achieved high modeling accuracy using a weighted least squares dynamics identification method. Sousa and Cortesão [6] enhanced modeling accuracy by considering the physical constraints of robot joint models. Han et al. [7] further designed a three-loop iterative learning strategy for robot dynamics, achieving high-precision robot identification and modeling. However, due to the linearization requirement of identification methods, robot dynamics and friction models are often simplified to linear systems [8], [9], resulting in some loss of modeling accuracy and a lack of capacity to learn nonlinear characteristics. On the other hand, existing research on robot joint friction rarely synchronously models friction during overall dynamics modeling [10], [11].

Learning-based approaches, particularly those based on physics-informed neural networks (PINNs), have proven effective in many industrial applications such as equipment modeling [12], parameter estimation [13], and impedance identification [14]. However, existing learning-based methods currently lag behind identification-based methods in robot modeling accuracy. Despite this, their superior nonlinear learning capabilities and generalizability offer broader application prospects, making them a significant research direction in robot dynamics modeling [15], [16]. Lutter et al. [17], [18] proposed the deep Lagrangian networks (DeLaN) framework, which uses structured Lagrangian network layers for robot dynamics modeling. However, it does not model nonconservative forces such as joint friction, limiting its application scope. Gupta et al. [19] extended DeLaN with dissipative force networks to handle nonconservative forces, enhancing the network's ability to learn robot friction and dissipative forces. Lahoud et al. [20] incorporated actual physical friction models, integrating a Coulomb-viscous friction model, achieving better overall modeling performance. Nonetheless, these PINN-based methods do not specifically design learning strategies for the dynamics and friction components of joint torques.

Received 17 July 2024; revised 4 September 2024; accepted 30 September 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4700300 and the NSFC-Shenzhen Robot Basic Research Center project under Grant U2013204. *(Corresponding author: Chungang Zhuang.)*

The authors are with the Robotics Institute, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: cgzhuang@sjtu.edu.cn).

Digital Object Identifier 10.1109/TIE.2024.3476977

Moreover, industrial robots typically lack joint torque sensors due to cost constraints. Therefore, learning joint dynamics and friction models solely from joint current signals, without sensor-acquired torque labels, is crucial for enhancing the accuracy of industrial robot models. Additionally, the structural design of industrial robots results in significant differences in the torques experienced by the lower and upper joints, leading to varying dynamics and friction characteristics across joints. This presents a challenge for learning-based methods. Thus, it is essential to design appropriate learning strategies to ensure that each robot joint's dynamics and friction characteristics can be adequately learned, even without sensor torque labels.

To address these challenges, a PINN-based friction-inclusive dynamics modeling method is proposed. Unlike previous studies, this method, aimed at multijoint industrial robots without joint torque sensors, constructs a structured PINN combining Lagrangian dynamics and the Stribeck friction model. Innovatively, a hybrid learning strategy with dual loops was designed to account for the different complexities of the robot's joint dynamics model and friction model. Finally, historical information of robot motion is extracted to predict and compensate for residual errors. This residual compensation (RC) module is then integrated into the proposed PINN modeling framework. The effectiveness of the proposed method is validated on two different robots. Cross-validation experiments evaluate the modeling accuracy, comparing it with other advanced methods, and the load generalization experiments verify the method's applicability. The primary contributions are:

1) A new PINN-based method for industrial robot dynamics modeling has been proposed to achieve interpretable modeling of robotic dynamics and friction, and to effectively learn the nonlinear features of the robot model.
2) A hybrid learning strategy for the dynamics and friction components of robot joint torques has been designed. This strategy ensures modeling accuracy while avoiding dependence on the labels of joint torque components.
3) A residual error compensation module is integrated into the proposed PINN to handle the difficult-to-model torque residuals. Separately addressing residual components at different frequencies achieves higher modeling accuracy.

The remainder of this article is structured as follows. Section II outlines the theoretical foundations and preparatory work. Section III proposes a PINN-based dynamics modeling method for industrial robots that accounts for nonlinear friction. In Section IV, the experimental validation is presented, and the modeling results is provided. Finally, Section V provides the conclusion.

II. Preliminaries

A. Robot Dynamics Modeling and Identification

For an n degree of freedom (DOF) serial robot, the dynamics model can typically be expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \underbrace{\dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right)^T}_{:=\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \quad (1)$$

where $\boldsymbol{q} \in \Re^{n\times 1}$ represents the robot joint angle and $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^{n\times 1}$ represent the robot kinematic states, $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{n\times n}$ is the mass matrix, which is symmetric and positive definite, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \Re^{n\times n}$ is the Coriolis and centrifugal matrix, $\boldsymbol{G}(\boldsymbol{q}) \in \Re^{n\times 1}$ represents the gravity vector, and $\boldsymbol{\tau} \in \Re^{n\times 1}$ is the total torque of each joint of the robot, which is typically composed of dynamics torque $\boldsymbol{\tau}_{\text{dyn}} \in \Re^{n\times 1}$, friction torque $\boldsymbol{\tau}_{\text{fri}} \in \Re^{n\times 1}$, and torque error $\boldsymbol{\tau}_{\text{err}} \in \Re^{n\times 1}$ caused by other difficult-to-model strong nonlinear factors

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\text{dyn}} + \boldsymbol{\tau}_{\text{fri}} + \boldsymbol{\tau}_{\text{err}}. \quad (2)$$

For most physical dynamics modeling and parameter identification methods, due to the linearization requirements, the classical Coulomb-viscous friction model is employed to characterize the main features of friction in robot joints. The friction torque for the kth joint can be formed as follows:

$$\tau_{\text{fri}}^k = f_c^k \text{sign}(\dot{q}_k) + f_v^k \dot{q}_k + f_b^k \quad (3)$$

where f_c^k, f_v^k, and f_b^k denote the Coulomb coefficient, the viscous coefficient and the friction offset of the kth joint, respectively, and sign $(\cdot)$ is a sign function.

However, when robot joints operate at low speeds, the Stribeck effect manifests, characterized by a reduction in friction force with increasing velocity [21]. When considering the Stribeck effect, the friction torque and the coefficient of friction exhibit a nonlinear relationship.

B. Framework for Dynamics Modeling Based on PINNs

DeLaN [17] is a novel dynamics modeling framework based on PINN. This framework deconstructs the internal structure of robotic Lagrangian dynamics and builds upon it an interpretable network architecture that strictly ensures the learned dynamics models adhere to actual physical constraints.

For the mass matrix $\boldsymbol{M}(\boldsymbol{q})$ in the dynamics model of the robot as denoted in (1), due to its positive definite nature, a general decomposition can be performed as follows:

$$\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{L}(\boldsymbol{q})\boldsymbol{L}^T(\boldsymbol{q}) \quad (4)$$

where $\boldsymbol{L}(\boldsymbol{q}) \in \Re^{n\times n}$ is a lower triangular matrix. Therefore, for the dynamics model described in (1), it is sufficient to use a deep neural network to learn $\boldsymbol{L}(\boldsymbol{q})$ and $\boldsymbol{G}(\boldsymbol{q})$, thereby reducing the complexity of the network while strictly ensuring the physical constraint of the positive definiteness of the mass matrix $\boldsymbol{M}(\boldsymbol{q})$. The process of predicting $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{G}(\boldsymbol{q})$ through the deep neural network can be described as follows:

$$\begin{aligned} \hat{\boldsymbol{M}}(\boldsymbol{q}) &= \hat{\boldsymbol{L}}(\boldsymbol{q};\theta)\hat{\boldsymbol{L}}(\boldsymbol{q};\theta)^T \\ \hat{\boldsymbol{G}}(\boldsymbol{q}) &= \hat{\boldsymbol{G}}(\boldsymbol{q};\psi) \end{aligned} \quad (5)$$

where $\hat{\boldsymbol{L}}$, $\hat{\boldsymbol{G}}$, and $\hat{\boldsymbol{M}}$ represent the network predictions, while θ and ψ are the associated network parameters. Therefore, the optimization problem for training the network in the dynamics part can be described as follows:

$$(\theta^*, \psi^*) = \arg\min_{\theta,\psi} \ell\left(\Theta(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}};\theta,\psi), \boldsymbol{\tau}_{\text{dyn}}\right) \quad (6)$$

where Θ represents the robot inverse dynamics model. ℓ denotes the loss function of the neural network.

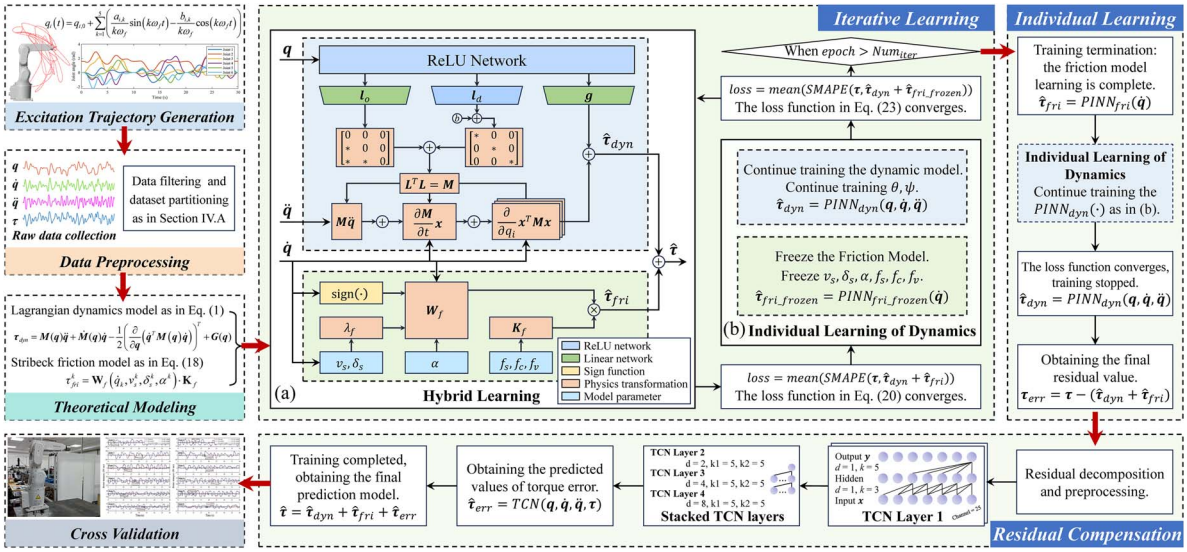

Fig. 1. Proposed PINN-based method of friction-inclusive dynamics modeling. (a) Hybrid learning of dynamics and friction. (b) Individual learning of dynamics.

The lower triangular matrix $\boldsymbol{L}$ can be decomposed into a diagonal matrix $\boldsymbol{L}_d$ and a strictly lower triangular matrix $\boldsymbol{L}_o$. To ensure that $\boldsymbol{L}$ has positive diagonal elements and that $\boldsymbol{M}$ has positive eigenvalues, the output layer for $\boldsymbol{L}_d$ should utilize a non-negative activation function, such as ReLU.

III. Methodology

In this section, the proposed hybrid modeling method for robot dynamics that accounts for joint friction based on PINN is elaborated in detail. As depicted in Fig. 1, the method consists of three stages. After completing the theoretical modeling, the first step involves iterative learning that integrates the rigid body dynamics model with the Stribeck friction model. The second step focuses on individual learning for the more complex rigid body dynamics model. Finally, a temporal convolutional network (TCN) is designed within the proposed PINN to compensate for strongly nonlinear residual errors. The remainder of this section will provide a detailed exposition of the construction and hybrid learning strategies for the robot dynamics and friction models, as well as the TCN network structure for residual compensation.

A. Friction-Inclusive Dynamics Modeling

Excitation experiments are conducted to acquire the robot joint currents $\boldsymbol{i} \in \Re^{n\times 1}$, joint angles $\boldsymbol{q}$, and joint angular velocities $\dot{\boldsymbol{q}}$. The joint angular accelerations $\ddot{\boldsymbol{q}}$ are derived by differentiating the angular velocities. The total joint torques are computed based on the joint currents.

$$\boldsymbol{\tau} = \mathbf{S}\boldsymbol{i} \tag{7}$$

where $\mathbf{S} \in \Re^{n\times n}$ is the joint drive gain matrix that converts motor currents into measured torques. The total torque encompasses both the dynamics and friction torques.

The PINN-based method is not constrained by the linearization of models, allowing the Stribeck effect to be conveniently incorporated into the training of the overall dynamics model. The proposed modeling method considers the Stribeck effect and integrates the learning of empirical parameters v_s, δ_s, α within the overall dynamics model. This approach makes the model's learning process independent of the selection of prior values. The proposed PINN structure for the dynamics and friction components is shown in Fig. 1(a). The dynamics component is designed based on the DeLaN network architecture [17]. As mentioned in Section II.B, the robot rigid body dynamics model can be represented as

$$\begin{aligned}\Theta(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}};\theta,\psi) = \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\ddot{\boldsymbol{q}} + \frac{d}{dt}\left(\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\right)\dot{\boldsymbol{q}} \\ -\frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\dot{\boldsymbol{q}}\right)\right)^T + \hat{\boldsymbol{G}}\end{aligned} \tag{8}$$

where $\hat{\boldsymbol{L}}$ and $\hat{\boldsymbol{G}}$ can be learned through a Lagrangian computation layer based on a ReLU network, while θ and ψ are iteratively updated through network training, eventually converging to steady values. For the partial derivative terms $d\left(\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\right)/dt$ and $\partial\left(\dot{\boldsymbol{q}}^T\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\dot{\boldsymbol{q}}\right)/\partial\boldsymbol{q}$ in the dynamics model, since time t is not an input to the dynamics model, it is challenging to use automatic differentiation for these terms. Therefore, the following chain decomposition is performed:

$$\frac{d}{dt}\left(\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\right) = \hat{\boldsymbol{L}}\frac{d\hat{\boldsymbol{L}}(\boldsymbol{q};\theta)^T}{dt} + \frac{d\hat{\boldsymbol{L}}(\boldsymbol{q};\theta)}{dt}\hat{\boldsymbol{L}}^T. \tag{9}$$

Since the network parameters θ are independent of time t and can be considered time-invariant, the following holds:

$$\frac{d\hat{\boldsymbol{L}}(\boldsymbol{q};\theta)}{dt} = \frac{\partial\hat{\boldsymbol{L}}}{\partial\boldsymbol{q}}\frac{\partial\boldsymbol{q}}{\partial t} = \frac{\partial\hat{\boldsymbol{L}}}{\partial\boldsymbol{q}}\dot{\boldsymbol{q}}. \tag{10}$$

Due to the compositional structure and differentiability of the network, the derivative of $\hat{\boldsymbol{L}}$ with respect to the network input can be computed by recursively applying the chain rule

$$\frac{\partial\boldsymbol{L}}{\partial\boldsymbol{q}} = \frac{\partial\boldsymbol{L}}{\partial\boldsymbol{h}_{N-1}}\frac{\partial\boldsymbol{h}_{N-1}}{\partial\boldsymbol{h}_{N-2}}\cdots\frac{\partial\boldsymbol{h}_1}{\partial\boldsymbol{q}} \tag{11}$$

where $\boldsymbol{h}_i$ denotes the output of the ith network layer, obtained through an affine transformation followed by a nonlinear activation function $g_i(\bullet)$:

$$\boldsymbol{h}_i = g_i\left(\boldsymbol{W}_i^T \boldsymbol{h}_{i-1} + \boldsymbol{b}_i\right) \tag{12}$$

where $\boldsymbol{h}_{i-1}$ is the output of the (i-1)th network layer and also serves as the input to the ith network layer, thus yielding

$$\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{h}_{i-1}} = \operatorname{diag}\left(g'\left(\boldsymbol{W}_i^T \boldsymbol{h}_{i-1} + \boldsymbol{b}_i\right)\right) \boldsymbol{W}_i \tag{13}$$

where $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ represent the network parameters of the ith layer. By substituting (13) into (11), a chained computation can be performed to obtain $\partial \boldsymbol{L}/\partial \boldsymbol{q}$, and then $d\left(\hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T\right)/dt$ can be computed. The computation of $\partial\left(\dot{\boldsymbol{q}}^T \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T \dot{\boldsymbol{q}}\right)/\partial \boldsymbol{q}$ is similar

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{q}_i}\left(\dot{\boldsymbol{q}}^T \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T \dot{\boldsymbol{q}}\right) &= \operatorname{tr}\left[\left(\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^T\right)^T \frac{\partial \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T}{\partial \boldsymbol{q}_i}\right] \\ &= \dot{\boldsymbol{q}}^T\left(\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{q}_i}\boldsymbol{L}^T + \boldsymbol{L}\frac{\partial \boldsymbol{L}^T}{\partial \boldsymbol{q}_i}\right)\dot{\boldsymbol{q}}.\end{aligned} \tag{14}$$

Therefore, the dynamics torque can be represented by (6), with each term being inferred through the proposed PINN.

For the joint friction component, the Stribeck effect is considered to make the modeling of the robot joint torque more consistent with the actual friction characteristics

$$\begin{aligned}\tau_{fri}^k &= Fc_k + (Fs_k - Fc_k)\cdot \lambda_k + Fv_k \\ &= \begin{bmatrix}\lambda_k \cdot \operatorname{sign}(\dot{q}_k) & (1-\lambda_k)\cdot \operatorname{sign}(\dot{q}_k) & \dot{q}_k\end{bmatrix} \cdot \begin{bmatrix} f_s^k \\ f_c^k \\ f_v^k \end{bmatrix}\end{aligned} \tag{15}$$

$$\begin{aligned} Fc_k &= \operatorname{sign}(\dot{q}_k)\cdot f_c^k \\ Fs_k &= \operatorname{sign}(\dot{q}_k)\cdot f_s^k \\ Fv_k &= \dot{q}_k \cdot f_v^k \\ \lambda_k &= \exp\left(-\left|\dot{q}_k / v_s^k\right|^{\delta_s^k}\right)\end{aligned} \tag{16}$$

where v_s^k represents the critical velocity for the Stribeck effect of the friction in the kth joint, corresponding to the joint velocity at which the friction torque is minimized. δ_s^k is an empirical coefficient related to factors such as the materials of the friction surfaces. Define $[f_s^k \quad f_c^k \quad f_v^k]^{\mathrm{T}} = \mathbf{K}_f$ as the friction coefficient. In fact, viscous friction Fv_k better fits the actual friction conditions when it is proportional to an exponential function of joint velocity [22]

$$Fv_k = |\dot{q}_k|^{\alpha^k}\cdot \operatorname{sign}(\dot{q}_k)\cdot f_v^k \tag{17}$$

where $\alpha^k \in \Re$ represents the exponent related to the velocity in the viscous friction of the kth joint. Therefore, the friction model in (15) can be rewritten as

$$\begin{aligned}\tau_{\mathrm{fri}}^k &= \operatorname{sign}(\dot{q}_k)\cdot\begin{bmatrix}\lambda_k & (1-\lambda_k) & |\dot{q}_k|^{\alpha^k}\end{bmatrix}\cdot \mathbf{K}_f \\ &= \mathbf{W}_f\left(\dot{q}_k, v_s^k, \delta_s^k, \alpha^k\right)\cdot \mathbf{K}_f\end{aligned} \tag{18}$$

where $\mathbf{W}_f$ represents the friction model matrix. The empirical parameters $v_s^k, \delta_s^k, \alpha^k$, and the friction coefficient $\mathbf{K}_f$ are treated as intrinsic parameters of the PINN.

The dynamics and friction models are simultaneously involved in iterative training and parameter updating. Due to the coupling of the dynamics model of all joints, the learning process for each joint is synchronized. The proposed PINN modeling network does not rely on labels for the dynamics and friction components of the joint torques. By employing the hybrid learning strategy designed in Section III.B, the dynamics and friction components of the torques can be progressively separated. The proposed PINN modeling method ultimately outputs the predicted total torque for each joint

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_{\mathrm{dyn}} + \hat{\boldsymbol{\tau}}_{\mathrm{fri}}. \tag{19}$$

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 loss function of the network is designed based on symmetric mean absolute percentage error (SMAPE) [23] as follows:

$$\operatorname{loss}(\boldsymbol{\tau}, \hat{\boldsymbol{\tau}}) = \frac{1}{6n}\sum_{j=1}^{6}\sum_{i=1}^{n}\frac{2|\boldsymbol{\tau}(i) - \hat{\boldsymbol{\tau}}(i)|}{(|\boldsymbol{\tau}(i)| + |\hat{\boldsymbol{\tau}}(i)|)}. \tag{20}$$

B. Dynamics-Friction Hybrid Learning Strategy

When the loss function in (20) converges, it indicates the completion of the hybrid learning of the dynamics and friction torque components. However, in reality, the dynamics and friction components of the robot joint model exhibit significant differences in complexity. The former involves multi-joint coupling and complex chain structures requiring the solution of partial differential equations, while the latter focuses on friction modeling for individual joints. Consequently, when the hybrid learning process of both components is completed, the learning of the friction component has converged. In contrast, the modeling of the dynamics component may still require further updates.

Therefore, after completing the hybrid learning step, it is recommended to conduct individual training for the complex dynamics component, as shown in lines 11–17 of Algorithm 1, corresponding to Fig. 1(b). The main purpose of this step is to enable the proposed method to correctly and quickly distinguish between the dynamics component and friction component of robot joints, addressing the challenge industrial robots face in accurately differentiating these torque components due to the lack of sensor labels. The network parameters v_s^k, δ_s^k, α^k and the friction coefficient f_s^k, f_c^k, f_v^k are frozen and no longer updated. Thus, the predicted torque for the friction component remains fixed as

$$\hat{\boldsymbol{\tau}}_{\mathrm{fri}} = \mathrm{PINN}_{\mathrm{fri\_frozen}}(\dot{\boldsymbol{q}}) \tag{21}$$

where $\mathrm{PINN}_{\mathrm{fri\_frozen}}(\cdot)$ represents the frozen joint friction network model. On the other hand, the network for the dynamics continues to be trained. The parameters θ and ψ are retained and will continue to be updated. The dynamics torque can be calculated as follows:

$$\hat{\boldsymbol{\tau}}_{\mathrm{dyn}} = \mathrm{PINN}_{\mathrm{dyn}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{22}$$

Algorithm 1: PINN-Based Dynamics and Friction Modeling Algorithm

Input: Filtered data: $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{i}$
Joint drive gain matrix: $\mathbf{S}$
Maximum number of iterations: Num_{iter}
Output: Predicted torque: $\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\tau}}_{dyn}, \hat{\boldsymbol{\tau}}_{fri}$

1 Calculate measured torque $\boldsymbol{\tau} = \mathbf{S}\boldsymbol{i}$
2 **START**
3 **While** *epoch* $\leq Num_{iter}$ **do**
4 **Repeat**
5 Simultaneously train the dynamic model and the friction model
6 Calculate dynamic torque ←Eq. (22)
7 Calculate friction torque: $\hat{\boldsymbol{\tau}}_{fri} = PINN_{fri}(\dot{\boldsymbol{q}})$
8 Calculate loss ←Eq. (20)
9 Update the parameters of both $PINN_{dyn}$ and $PINN_{fri}$
10 **Until** *the loss function converges*
11 Freeze the parameters of the friction model ←Eq. (21)
12 **Repeat**
13 Train only the dynamic model (**Inner loop**)
14 Calculate dynamic torque ←Eq. (22)
15 Calculate loss ←Eq. (23)
16 Update the parameters of $PINN_{dyn}$
17 **Until** *the loss function converges*
18 **End**
19 Freeze the parameters of the friction model ←Eq. (21)
20 **Repeat**
21 Train only the dynamic model (**Outer loop**)
22 Calculate dynamic torque ←Eq. (22)
23 Calculate loss ←Eq. (23)
24 Update the parameters of $PINN_{dyn}$
25 **Until** *the loss function converges*
26 **END**
27 Obtain predicted total torque ←Eq. (19)
28 **Return** $\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\tau}}_{dyn}, \hat{\boldsymbol{\tau}}_{fri}$

where $\mathrm{PINN}_{\mathrm{dyn}}(\cdot)$ represents the joint dynamics network model. The overall loss function of the network is modified as

$$\mathrm{loss}(\boldsymbol{\tau}, \hat{\boldsymbol{\tau}}) = \mathrm{loss}\left(\boldsymbol{\tau}, \hat{\boldsymbol{\tau}}_{\mathrm{dyn}} + \hat{\boldsymbol{\tau}}_{\mathrm{fri\_frozen}}\right). \tag{23}$$

To enable the proposed robot modeling method to be applicable in situations where it is difficult to obtain labels for joint dynamics torque and friction torque, a vital issue is designing a learning strategy that ensures both components of the torques can be modeled with high accuracy. The proposed learning strategy is based on a cyclic iterative approach, allowing the PINN network model to switch and iterate multiple times between hybrid learning and the separate learning of the dynamics component. This iterative switching gradually aligns the modeling of both torque components with the actual conditions, thereby effectively enhancing the overall modeling accuracy of robot joints using the PINN method.

Furthermore, to achieve high modeling accuracy for all joints, including those with smaller torques, individual learning stage for the dynamics component is conducted after the iterative learning stage, as shown in lines 19–25 of Algorithm 1. The pseudocode

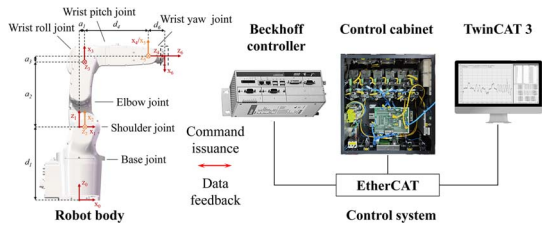

Fig. 2. Robotic experimental platform.

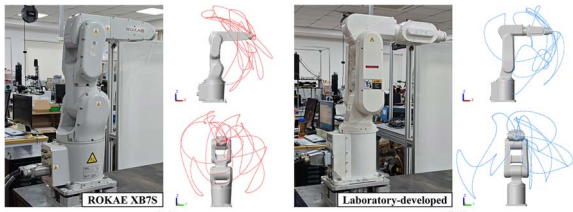

Fig. 3. Robotic excitation trajectories. (a) ROKAE XB7S robot trajectory. (b) Laboratory-developed robot trajectory.

for the learning strategy of the proposed friction-inclusive joint dynamics modeling method is summarized in Algorithm 1.

C. Residual Error Compensation

Due to the influence of nonlinear factors that are difficult to model physically, there will inevitably be residuals between the predicted total joint torques and the actual values. To further improve modeling accuracy, the proposed residual compensation module follows the process shown in Fig. 1.

The torque residuals of robot joints can generally be divided into two parts: one part is related to the motion of the robot body and has a frequency distribution similar to the robot's movement; the other part is related to sudden changes in motor motion and typically has a higher frequency. By setting a critical point based on the robot's motion frequency, the torque residuals can be decomposed into two parts with different frequencies. This decomposition makes it easier to learn the characteristics of each part of the error separately.

After the dynamics torque and friction torque have both been learned, the target labels for the torque residuals can be obtained and further decomposed:

$$\begin{aligned} \boldsymbol{\tau}_{\mathrm{err}} &= \boldsymbol{\tau} - \hat{\boldsymbol{\tau}}_{\mathrm{dyn}} - \hat{\boldsymbol{\tau}}_{\mathrm{fri}} \\ \boldsymbol{\tau}_{\mathrm{err1}} &= \kappa(\boldsymbol{\tau}_{\mathrm{err}}, f_{\mathrm{cut}}) \\ \boldsymbol{\tau}_{\mathrm{err2}} &= \boldsymbol{\tau}_{\mathrm{err}} - \boldsymbol{\tau}_{\mathrm{err1}} \end{aligned} \tag{24}$$

where $\kappa(\cdot)$ represents the low-pass filter, and f_{cut} denotes the cut-off frequency determined by the robot motion frequency.

A sliding window of length l is utilized to process the kinematic quantities $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ and the measured joint torques $\boldsymbol{\tau}$, thereby obtaining the input for the compensation module.

$$\begin{aligned} \boldsymbol{X}_t &= \left[\boldsymbol{\chi}_{t-l+1}, \ldots, \boldsymbol{\chi}_t\right] \\ \boldsymbol{\chi}_t &= \left[q_t, \dot{q}_t, \ddot{q}_t, \tau_t\right]^{\mathrm{T}} \end{aligned} \tag{25}$$

where $\boldsymbol{\chi}_t$ denotes the robot state vector at time t and the input $\boldsymbol{X}_t \in \Re^{4n \times l}$ is the combination of continuous state vectors over a past duration of length l. The historical information in the input is

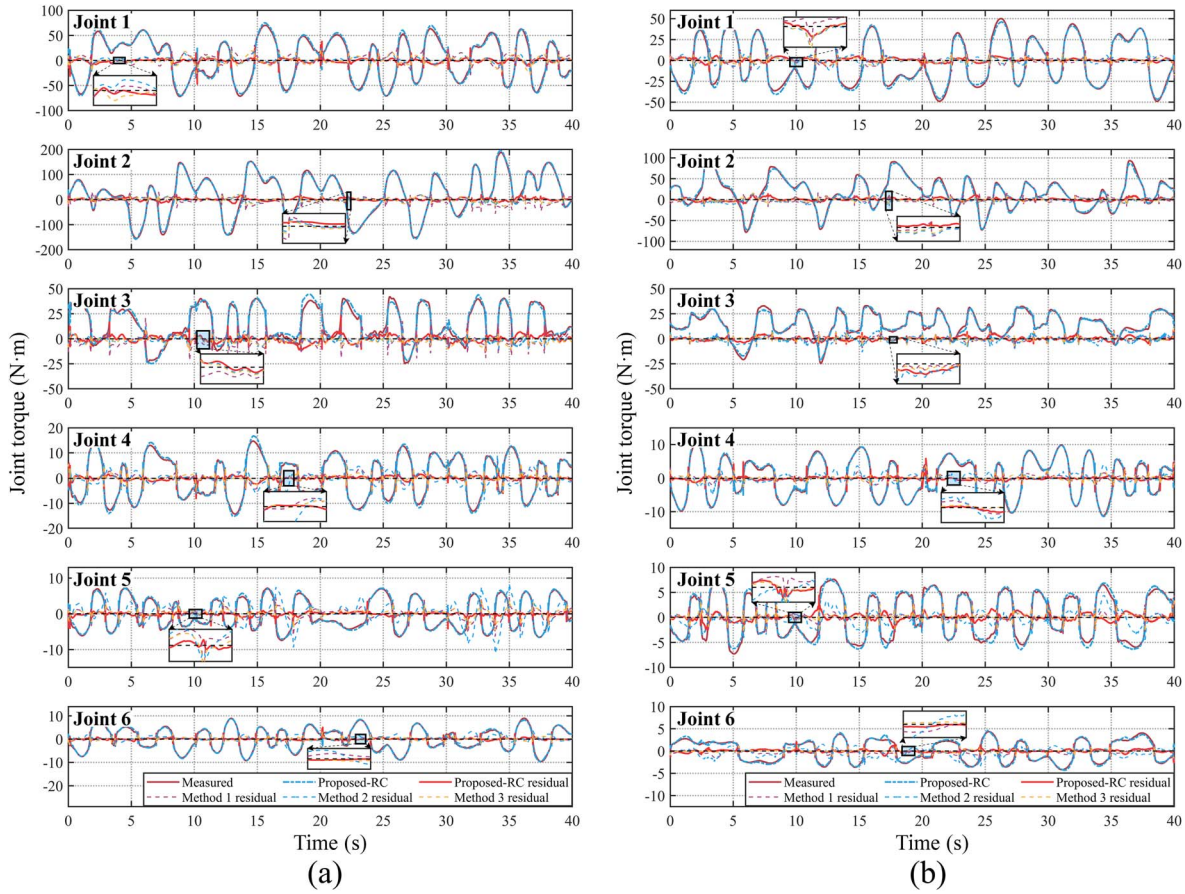

Fig. 4. Cross-validation experimental results. (a) Torque prediction results for the ROKAE XB7S robot. (b) Torque prediction results for the laboratory-developed robot.

extracted through the stacked 1-D convolutional architecture and dilated convolutions. The dilated convolution can be described as

$$\boldsymbol{y}(t) = (\boldsymbol{x} *_d \boldsymbol{f})(t) = \sum_{i=0}^{k-1} x(t - di) \cdot f(i) \tag{26}$$

where $*_d$ denotes the dilated convolution operation, $\boldsymbol{x} \in \Re^{n\times 1}$ is the input data, $\boldsymbol{f} \in \Re^{k\times 1}$ signifies the dilated convolutional filter, k is the filter size, and d is the dilation factor.

The TCN module is composed of stacked 1-D convolutional layers with varying dilation factors and filter sizes. The consistency in the number of channels is maintained by padding layers and chopping layers, thereby enhancing the feature learning capability [24]. The torque residuals can be modeled and predicted as

$$\hat{\boldsymbol{\tau}}_{\text{err}} = \hat{\boldsymbol{\tau}}_{\text{err1}} + \hat{\boldsymbol{\tau}}_{\text{err2}} = \text{TCN}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\tau}) \tag{27}$$

where $\hat{\boldsymbol{\tau}}_{\text{err1}}$ and $\hat{\boldsymbol{\tau}}_{\text{err2}}$ represent the predicted residual components at different frequencies, both learned by the same TCN network branch structure. Therefore, the final predicted total joint torques can be obtained

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_{\text{dyn}} + \hat{\boldsymbol{\tau}}_{\text{fri}} + \hat{\boldsymbol{\tau}}_{\text{err}}. \tag{28}$$

IV. Experiments

In this section, the robot dynamics model identification experiments are conducted on the ROKAE XB7S robot and another laboratory-developed robot. Cross-validation and generalization experiments validated the effectiveness of the proposed modeling method. Both robots utilize the Beckhoff C6930 industrial computer as the primary controller, connected via EtherCAT for real-time data acquisition as in Fig. 2. Data sampling and control commands operate at 1 kHz. The openness and real-time capabilities of the Beckhoff platform enable deployment of the proposed model through the Beckhoff TE1400 module in actual robot control scenarios. The simulations in this article are conducted in MATLAB Simscape, with the network construction and training implemented through Python PyTorch, and finally deployed to the Beckhoff platform via TwinCAT.

A. Modeling Dataset Generation

The trajectories designed based on a fifth-order Fourier series are used in experiments to maximize the activation of model characteristics during the robot's motion [25]. Fig. 3 shows the trajectories of two different robots. The excitation experiment aims to collect the robot's kinematic states and joint current data for subsequent modeling and learning. The motion of the kth joint of the robot can be expressed as

$$q_k(t) = q_{k,0} + \sum_{i=1}^{5} \left(\frac{a_{k,i}}{i\omega_f} \sin(i\omega_f t) - \frac{b_{k,i}}{i\omega_f} \cos(i\omega_f t) \right) \tag{29}$$

TABLE I
COMPARISON OF MODELING METHODS

Method	Publication	Technique
Method 1	Jubien et al. [5]	Weighted least squares
Method 2	Sousa and Cortesão [6]	Ordinary least squares with physical constraints
Method 3	Han et al. [7]	Iterative reweighted least squares
Method 4	Lutter et al. [17]	Lagrangian networks (DeLaN)
Method 5	Gupta et al. [19]	DeLaN with dissipative networks
Method 6	Lahoud et al. [20]	DeLaN with Coulomb-viscous friction model
Proposed	/	Hybrid-enhanced DeLaN with Stribeck friction model
Proposed with RC	/	

where t denotes time, $q_{k,0}$ represents the initial position, ω_f signifies the fundamental frequency, $a_{k,i}$ and $b_{k,i}$ denote the ith order trajectory coefficients. Typically, the fundamental frequency is set to 00.1 Hz. The trajectory coefficients are determined through an iterative optimization technique [26].

For each robot, data from 12 different excitation trajectories are collected, with each trajectory lasting 10 s. A fifth-order Butterworth low-pass filter with a cutoff frequency of 50 Hz is applied in both forward and reverse directions to filter the raw data, eliminating high-frequency noise. Then, the filtered data is downsampled to shorten the model training and learning time. However, for residual modeling, downsampling is avoided to preserve high-frequency residual characteristics in the data. The total joint torque measurements are then calculated according to (7). The dataset is randomly divided into training, validation, and test subsets in proportions of 60%, 20%, and 20%, respectively.

B. Hybrid Modeling of Dynamics and Friction

After generating the dataset, the proposed PINN is trained according to the learning strategies described in Section III. The network has three hidden layers, with the number of nodes set to [128, 128, 128]. Adaptive moment estimation (Adam) is chosen as the optimizer for the network. Without loss of generality, Fig. 4 provides a detailed description of the torque modeling results for the robot joints. Table I shows a detailed comparison of different methods. Both the proposed method and other comparative methods use the same optimized excitation trajectory as the dataset, avoiding the impact of different trajectories on the modeling results. Fig. 4(a) and 4(b), respectively, illustrates the results of the ROKAE XB7S robot and the laboratory-developed robot. Joints 1–3 represent the robot's base joint, shoulder joint, and elbow joint, respectively. Joints 4–6 represent the robot's wrist roll, wrist pitch, and wrist yaw joints, respectively. The three lower joints primarily bear the torque, while the upper joints experience smaller torque with a higher proportion of friction components. Owing to the designed hybrid learning strategy, all joints achieve high modeling accuracy.

Fig. 5 shows the good fitting performance of the proposed method for the friction component. The distribution of the friction component with respect to velocity aligns with typical friction characteristics.

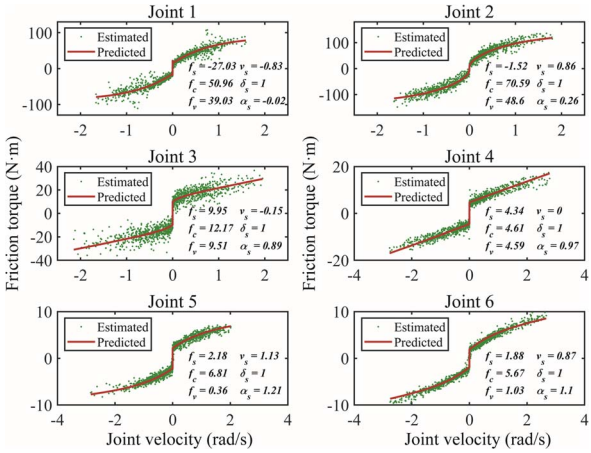

Fig. 5. Friction torque prediction results of robot joints.

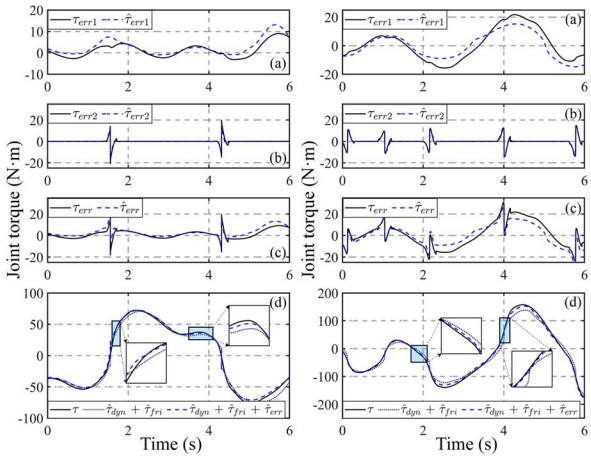

Fig. 6. Residual error compensation results. (a) Prediction results of the low-frequency residual component. (b) Prediction results of the high-frequency residual component. (c) Prediction results of the residual error. (d) Joint torques after residual compensation.

C. Residual Error Compensation

The compensation results for the residual torque errors are illustrated in Fig. 6. The compensation of the low-frequency components of the residuals further enhances the modeling accuracy of both the dynamics and friction models. The high-frequency residuals, which exhibit abrupt changes, reflect the directional changes in the robot joints' movements. The proposed residual error compensation method effectively captures the characteristics of the residual errors, including distinct properties of different frequency components. Owing to the excellent compatibility of the learning-based method, this residual compensation module can be seamlessly integrated into the proposed PINN, achieving higher precision in the modeling of the robot joint torques.

D. Prediction Performance Evaluation and Comparison

Existing methods for robot dynamics and friction modeling can be broadly categorized into identification-based and learning-based approaches. Among these, the learning-based methods utilizing PINNs are significantly superior to other black-box learning methods [27]. Consequently, the proposed method is

TABLE II
TORQUE PREDICTION PERFORMANCE COMPARISON WITH OTHER METHODS EVALUATED BY RMSE (N·M)

Method	Type	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Average
Method 1	Identification-based	80.47	140.07	50.35	10.51	10.11	00.61	50.19
Method 2		60.23	80.83	30.15	20.15	20.23	00.98	30.93
Method 3		70.30	80.67	30.54	10.23	00.84	00.52	30.68
Method 4	PINN-based	860.61	1290.84	730.14	470.27	150.76	320.91	640.25
Method 5		450.77	290.26	190.58	470.41	90.52	110.04	270.10
Method 6		80.35	110.72	110.75	80.48	60.92	40.10	80.55
Proposed		60.29	100.73	40.11	10.38	00.65	00.63	30.97
Proposed with RC		**30.95**	**50.80**	**20.91**	**00.72**	**00.42**	**00.40**	**20.37**

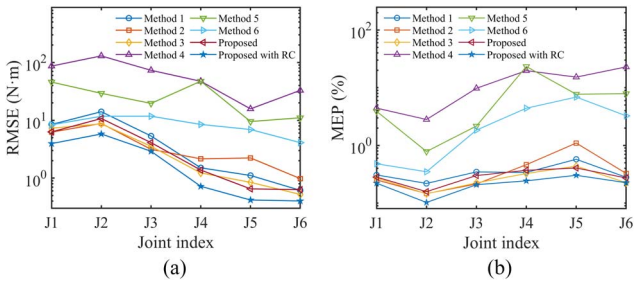

Fig. 7. Performance index comparison of joint torque prediction. (a) RMSE. (b) MEP.

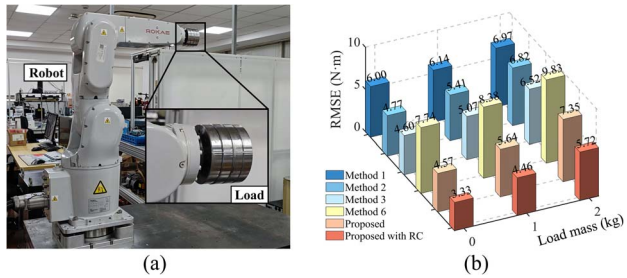

Fig. 8. Load generalization experiments. (a) Experimental platform. (b) Load generalization results.

compared with other advanced identification-based methods and PINN-based methods, as shown in Table II. All methods are trained multiple times to obtain average accuracy results. For the proposed method and other learning-based comparison methods, the primary hyperparameters are kept consistent: the training batch size is set to 50, the learning rate is set to $5e-4$, and the input data dimensions for the network models are identical.

Identification-based methods, particularly Method 3 [7], exhibit high modeling accuracy due to their robust theoretical foundation and optimized identification algorithms. Method 4 [17] lacks modeling of nonconservative forces such as joint friction, resulting in suboptimal modeling performance on the industrial robot used in this experiment. Method 5 [19] builds on Method 4 by incorporating dissipative force networks to address joint friction. Method 6 [20] employs a Coulomb-viscous friction model, achieving better friction modeling results. However, the methods above have yet to design learning strategies specifically for the dynamics and friction components of joint torques.

Root mean squared error (RMSE) [28] and maximum error percentage (MEP) [29] are used to evaluate the modeling accuracy of robot joint torques. The visualization of the torque modeling evaluation for each joint is shown in Fig. 7. Fig. 7(a) illustrates the RMSE values of the modeling errors for different methods. The average modeling error of the proposed method for each joint is 30.9653 N·m, which is comparable to advanced identification-based methods and superior to existing PINN-based methods. After incorporating the residual compensation module into the proposed PINN, the average modeling error is reduced to 20.3667 N·m, outperforming the identification-based methods. Fig. 7(b) depicts the MEP values of the modeling errors for different methods. The proposed method exhibits the highest accuracy in terms of maximum modeling error across all joints, particularly for the latter joints of the robot. The comparison results indicate that the proposed method achieves the best performance. It can be attributed to the proposed hybrid learning strategy, which combines Lagrangian dynamics and the Stribeck friction model, along with the integration of a residual compensation module capable of learning residual components across different frequency characteristics.

On the same computer equipped with a single NVIDIA RTX 4060 GPU (16 GB), the inference time required by the proposed method for a single sample is $1.0790e-05$ s, similar to other PINN-based methods [17], [19], [20] and superior to identification-based methods [5], [6], [7]. After considering residual compensation, the inference time of the proposed method increases to $4.6164e-05$ s. Nevertheless, it still meets the real-time requirements for model deployment in this work (i.e., a control frequency of 1 kHz) as well as most industrial robotic tasks.

E. Load Generalization Experiments

Load generalization experiments are conducted to further validate the effectiveness of the proposed method. The average joint torque RMSE results for the ROKAE XB7S robot under three different load masses are shown in Fig. 8. The integration of the hybrid learning strategy and the residual compensation module enhances the load generalization capability of the proposed modeling method, demonstrating superior torque prediction accuracy across varying load mass conditions compared to other methods.

V. CONCLUSION

This article proposes a new PINN-based dynamics modeling method for industrial robots that accounts for friction. To the best of the authors' knowledge, this work is the first to combine

a nonlinear friction model with PINN dynamics modeling and design corresponding learning strategies. Validation and generalization experiments on different robots demonstrate that the proposed method outperforms other advanced methods in modeling accuracy and exhibits good generalization capabilities. After selecting the appropriate friction model based on the robot's structure, this method is also suitable for other industrial robots. Future research could focus on further optimizing the learning strategies for the hybrid model and exploring the application of the hybrid PINN model in real-time robot control, thereby contributing to high-performance control of robots.

REFERENCES

[1] J. Zhang et al., "Parameter identification of hydraulic manipulators considering physical feasibility and control stability," *IEEE Trans. Ind. Electron.*, vol. 71, no. 1, pp. 718–728, Jan. 2024.

[2] Q. Leboutet, J. Roux, A. Janot, J. R. Guadarrama-Olvera, and G. Cheng, "Inertial parameter identification in robotics: A survey," *Appl. Sci.*, vol. 11, no. 9, p. 4303, 2021.

[3] Y. M. Zhao, Y. Lin, F. Xi, and S. Guo, "Calibration-based iterative learning control for path tracking of industrial robots," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2921–2929, May 2015.

[4] F. Guo et al., "Model-based deep learning for low-cost IMU dead reckoning of wheeled mobile robot," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7531–7541, Jul. 2024.

[5] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the Kuka LightWeight robot: Comparison between actual and confidential Kuka's parameters," in *Proc. IEEE ASME Int. Conf. Adv. Intell. Mechatron. (AIM)*, Jul. 2014, pp. 483–488.

[6] C. D. Sousa and R. Cortesão, "Inertia tensor properties in robot dynamics identification: A linear matrix inequality approach," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 1, pp. 406–411, Feb. 2019.

[7] Y. Han, J. Wu, C. Liu, and Z. Xiong, "An iterative approach for accurate dynamic model identification of industrial robots," *IEEE Trans. Robot*, vol. 36, no. 5, pp. 1577–1594, Oct. 2020.

[8] M. Gautier, "Dynamic identification of robots with power model," in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, vol. 3, Apr. 1997, pp. 1922–1927.

[9] J. Swevers, W. Verdonck, and J. D. Schutter, "Dynamic model identification for industrial robots," *IEEE Contr. Syst. Mag*, vol. 27, no. 5, pp. 58–71, Oct. 2007, doi: 10.1109/MCS.2007.904659.

[10] K. Guo, Y. Pan and H. Yu, "Composite learning robot control with friction compensation: A neural network-based approach," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7841–7851, Oct. 2019.

[11] M. Ruderman and M. Iwasaki, "Sensorless torsion control of elastic-joint robots with hysteresis and friction," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1889–1899, Mar. 2016.

[12] B. Kim and M. Shin, "A novel neural-network device modeling based on physics-informed machine learning," *IEEE Trans. Electron. Devices*, vol. 70, no. 11, pp. 6021–6025, Nov. 2023.

[13] S. Zhao, Y. Peng, Y. Zhang, and H. Wang, "Parameter estimation of power electronic converters with physics-informed machine learning," *IEEE Trans. Power Electron.*, vol. 37, no. 10, pp. 11567–11578, Oct. 2022.

[14] M. Zhang, Q. Xu, and X. Wang, "Physics-informed neural network based online impedance identification of voltage source converters," *IEEE Trans. Ind. Electron.*, vol. 70, no. 4, pp. 3717–3728, Apr. 2023.

[15] H. Lee, "Physics-based cooperative robotic digital twin framework for contactless delivery motion planning," *Int. J. Adv. Manuf. Technol.*, vol. 128, no. 3–4, pp. 1255–1270, Jul. 2023.

[16] X. Yang, Z. Zhou, L. Li, and X. Zhang, "Collaborative robot dynamics with physical human–robot interaction and parameter identification with PINN," *Mech. Mach. Theory*, vol. 189, Nov. 2023, Art. no. 105439.

[17] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *Proc. Int. Conf. Learn. Represent (ICLR)*, May 2019.

[18] M. Lutter and J. Peters, "Combining physics and deep learning to learn continuous-time dynamics models," *Int. J. Rob. Res.*, vol. 42, no. 3, pp. 83–107, Apr. 2023.

[19] J. K. Gupta, K. Menda, Z. Manchester, and M. Kochenderfer, "Structured mechanical models for robot learning and control," in *Proc. Mach. Learn. Res. (PMLR)*, vol. 120, Jun. 2020, pp. 328–337.

[20] M. G. Lahoud, G. Marchello, M. D'Imperio, A. Mueller, and F. Cannella, "A deep learning framework for non-symmetrical coulomb friction identification of robotic manipulators," in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, May 2024, to be published.

[21] N. Kammerer and P. Garrec, "Dry friction modeling in dynamic identification for robot manipulators: Theory and experiments," in *Proc. IEEE Int. Conf. Mechatron. (ICM)*, Feb. 2013, pp. 422–429.

[22] M. Iskandar and S. Wolf, "Dynamic friction model with thermal and load dependency: modeling, compensation, and external force estimation," in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, May 2019, pp. 7367–7373.

[23] T. Tsutsui and T. Matsuzawa, "Virtual metrology model robustness against chamber condition variation using deep learning," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 428–433, Nov. 2019.

[24] S. Z. Tan, J. X. Yang, and H. Ding, "A prediction and compensation method of robot tracking error considering pose-dependent load decomposition," *Robot. Comput.-Integr. Manuf.*, vol. 80, Apr. 2023, Art. no. 102476.

[25] K.-J. Park, "Fourier-based optimal excitation trajectories for the dynamic identification of robots," *Robotica*, vol. 24, no. 5, pp. 625–633, Sep. 2006.

[26] V. Bonnet et al., "Optimal exciting dance for identifying inertial parameters of an anthropomorphic structure," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 823–836, Aug. 2016.

[27] S. Ayankoso and P. Olejnik, "Time-series machine learning techniques for modeling and identification of mechatronic systems with friction: A review and real application," *Electronics*, vol. 12, no. 17, pp. 3669, Aug. 2023.

[28] Y. Wei, et al., "Contact force estimation of robot manipulators with imperfect dynamic model: on Gaussian process adaptive disturbance Kalman filter," *IEEE Trans. Autom. Sci. Eng.*, early access, doi: 10.1109/TASE.2023.3280750.

[29] S. Wang, X. Shao, L. Yang, and N. Liu, "Deep learning aided dynamic parameter identification of 6-DOF robot manipulators," *IEEE Access*, vol. 8, pp. 138102–138116, 2020.

Hongbo Hu received the B.E. degree in mechanical engineering from Chongqing University, Chongqing, China, in 2022. He is currently working toward the Ph.D. degree in mechanical engineering with Shanghai Jiao Tong University, Shanghai, China.

His research interests include robot force control, robot dynamic modeling, and calibration.

Zhikai Shen received the B.E. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2022. He is currently working toward the M.E. degree in mechanical engineering with Shanghai Jiao Tong University, Shanghai, China.

His research interests include robot force control, robot dynamic modeling, and identification.

Chungang Zhuang (Member, IEEE) received the Ph.D. degree in mechanical engineering from the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007.

Currently, he is an Associate Professor with the School of Mechanical Engineering, Shanghai Jiao Tong University. His research interests include machine vision, robot force control, and structural topology optimization.