Dynamics Modeling of Soft Robots Based on Attention-enhanced Lagrangian Deep Neural Networks

Wei, Yeqi; Shao, Xiangyu; Liu, Jingyue; Zhang, Shaojie; Xu, Linke

DOI
10.1109/FASTA61401.2024.10595110

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the 3rd Conference on Fully Actuated System Theory and Applications, FASTA 2024

Citation (APA)
Wei, Y., Shao, X., Liu, J., Zhang, S., & Xu, L. (2024). Dynamics Modeling of Soft Robots Based on Attention-enhanced Lagrangian Deep Neural Networks. In *Proceedings of the 3rd Conference on Fully Actuated System Theory and Applications, FASTA 2024* (pp. 200-207). IEEE. https://doi.org/10.1109/FASTA61401.2024.10595110

Proceedings of the 3rd Conference on Fully Actuated System Theory and Applications
May 10-12, 2024, Shenzhen, China

2024 3rd Conference on Fully Actuated System Theory and Applications (FASTA) | 979-8-3503-7369-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/FASTA61401.2024.10595110

Dynamics Modeling of Soft Robots Based on Attention-enhanced Lagrangian Deep Neural Networks

Yeqi Wei[1], Xiangyu Shao[1,*], Jingyue Liu[2], Shaojie Zhang[1], Linke Xu[1]
1. School of Astronautics, Harbin Institute of Technology, Harbin 150001, P. R. China
E-mail: xiangyushao@hit.edu.cn
2. Department of Cognitive Robotics, Delft University of Technology, Delft 2628 CD, The Netherlands

Abstract: This study explores a method for the dynamic modeling of soft robots, focusing on enhancing the deep learning-based Lagrangian modeling approach through the attention mechanism, which enriches the training process by allocating focused attention and analytical weighting to critical state features, thereby increasing the model's sensitivity to changes in the robot's state. We compared our method through simulation, demonstrating that the model is effective in long-term prediction and noise rejection.

Key Words: Attention-enhanced deep learning, Lagrangian neural networks, Soft robots, Predictive modeling, Machine learning in robotics.

1 Introduction

Soft robots are an interdisciplinary field that intersects engineering and biomechanics, demonstrating the enormous potential in sensitive owing to the unique characteristics of soft materials [1]. However, the inherent flexibility of soft materials introduces complexity in their kinematics and dynamics analysis concurrently, making traditional methods for evaluating its motion state insufficient [2, 3]. The inherent nonlinear characteristics and extensive degrees of freedom of flexible robots pose challenges and requires new modeling method to facilitate its application.

The latest advances in deep learning have propelled the capabilities of soft robots. Dong *et al.* demonstrated how to customize convolutional neural networks to process sensory data from soft robot skin, thereby enhancing environmental interaction [4]. Keene *et al.* have made progress in applying recursive neural networks to predictive modeling of soft robot dynamics, allowing for expected control actions [5]. However, due to the complex variability of soft material behavior, traditional deep learning methods often require a large amount of data sets or a long training time, and these objective conditions are sometimes difficult to gather together.

As shown in [6], attention mechanism fundamentally changes the field of machine learning by enabling the model to selectively focus on the data most relevant to the task at hand, providing a promising direction for overcoming these data challenges. These mechanisms were initially popularized in the field of natural language processing [7], they were also used as sequence to sequence models, but have not been fully explored in the context of robot control.

This paper proposes an innovative integration of attention mechanism into Lagrangian neural network (LNN) for modeling soft robots. As demonstrated in [8], the Lagrangian framework essentially explains the physical properties of motion and provides a natural fit for modeling soft robotic systems. Our method is based on the works of Liu [9], who introduced an LNN architecture for soft robots to achieve high-precision prediction of their states but yet introduce attention mechanisms.

By embedding attention in LNN, we aim to address the challenge of effectively modeling the nonlinear and high-dimensional dynamic characteristics of soft robots. As Wang proposed in [10], the attention mechanism enhances the prediction accuracy of models in the field of natural language. Inspired by its impressive improvement, this paper introduces the attention machine to the deep learning based modeling of soft robots to improve the modeling accuracy and prediction ability of the model. The potential benefits of this approach include reducing the need for large datasets and improving real-time response capabilities, which are crucial for deploying soft robots in real-world scenarios. The contributions of this paper is summarised as : (i) Introduce a new attention enhanced LNN architecture, (ii) demonstrate its advantages over traditional LNN through simulation, (iii) and its generalization was tested in a noisy environment.

2 Theoretical Basis and Related Work

The Lagrangian dynamics elucidates the motion of dynamic systems by balancing kinetic energy T and potential energy V [11]. The Lagrangian function $L(q, \dot{q})$ is defined as $L(q, \dot{q}) = T(\dot{q}) - V(q)$, providing a robust theoretical framework for modeling the complex motion of mechanical systems. This formula has become indispensable in the context of soft robots, where the complex interactions between elastic materials and various components challenge traditional rigid body dynamics.

Deep learning, as a subset of machine learning, has completely changed the way we model soft robots. Its advantage lies in its ability to learn from sequence data, capture temporal dependencies and subtle patterns. In view of this, deep learning has the potential to improve the traditional modeling methods of soft robot systems. It provides a method for extracting complex high-dimensional data into actionable insights and may also address issues such as the nonlinear and dynamic characteristics of soft robots. Therefore, integrating deep learning techniques into the modeling process can improve the prediction accuracy, adaptability, and efficiency

* Corresponding Author

This work is supported in part by the National Natural Science Foundation of China under Grant 62303137, in part by the Special funding of the China Postdoctoral Science Foundation under Grant 2023TQ0092, and in part by the Heilongjiang Postdoctoral Science Foundation under Grant LBH-Z20057.

of soft robot control systems.

The contribution of Liu in this regard is impressive, as they used Physically Informed Neural Networks (PINNs) to infer physically reasonable matrices from data [9]. By establishing deep learning models on the physical laws governing robot motion, their approach embodies the synergistic effect between data-driven methods and classical mechanics. However, although this method has many advantages, including improved sample efficiency and model interpretability, there is still room for improvement due to the need for a large number of datasets and training cycles, resulting in a decline in long-term prediction accuracy.

Introducing the attention mechanism from the transformer architecture into LNN is a method to improve the model. The success of attention mechanisms in machine learning, especially in processing sequential data, has already proven their potential in enhancing model focus and handling complex patterns. In our framework, it is utilized to enhance the sensitivity of LNNs to key system states, thereby enabling more accurate predictions of complex dynamic behaviors. Our model applies Lagrangian neural networks (LNN) within the PINNs framework to determine the mass matrix $\mathcal{M}$, dissipation matrix $\mathcal{D}$, and state transition matrix $\mathcal{I}$ from a large dataset of soft robot states q and inputs u. Subsequently, we use attention layers to enhance this framework, combining attention mechanisms acting on the physical derivation matrix and the state and input data of soft robots. By refining the attention related matrices Q, K, and V, the accuracy of the model has been improved.

The mathematical expression can be simplified as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V represent the query, key, and value, respectively. In our application, the key is a combination of the current state and input, while the query is the learned physical matrix . This structure allows the model to focus more on the parts of the system dynamics most relevant to the current input and state.

Through this approach, the attention-enhanced LNN model can predict the dynamic behavior of soft robots in different states more accurately than traditional LNNs, providing higher predictive accuracy and adaptability of the model.

3 Attention-Enhanced LNNs Design

3.1 Modeling Non-Conservative Forces in LNNs

Although traditional Lagrangian neural networks (LNNs) perform well in system dynamics, real-world applications often involve non conservative forces and require a new approach to cover this impact. We propose an extended LNN framework:

The original equations of motion for LNNs stem from the standard Lagrangian mechanics, where the dynamics are fully described by conservative forces. Traditionally, the Lagrangian $\mathcal{L}(q, \dot{q})$ encapsulates the system's total energy, combining kinetic $T(\dot{q})$and potential $V(q)$ energies. From the Euler-Lagrange equation, we obtain the standard form of the motion equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = 0 \tag{2}$$

Considering the limitations of traditional Lagrangian neural networks (LNNs) in handling non conservative forces, a new approach is needed to consider energy dissipation and the impact of inputs on the system. Therefore, we propose an enhanced framework that incorporates the effects of non conservative forces and external inputs into the LNN This is achieved by extending the classical Lagrangian mechanics to include a dissipation matrix $\mathcal{D}(q, \dot{q})$ and a state transition matrix $\mathcal{I}(q)$. The dissipation matrix accounts for energy loss due to non-conservative forces, while the state transition matrix models the influence of external control inputs on the system.

Accordingly, the motion equation is modified to integrate these matrices, providing a comprehensive representation of the system's dynamics under the influence of both conservative and non-conservative forces, as well as external inputs:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} + \mathcal{D}(q, \dot{q})\dot{q} = \mathcal{I}(q)\tau \tag{3}$$

where τ denotes the external control inputs applied to the system. To analyze the system's dynamics, we derive the mass matrix $\mathcal{M}(q)$ by calculating the Hessian matrix $H(\mathcal{L})$ of the Lagrangian $\mathcal{L}$ with respect to the generalized velocities $\dot{q}$, which represents second-order partial derivatives and encapsulates the system's inertia.

$$H(\mathcal{L}) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_1^2} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_1 \partial \dot{q}_2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_1 \partial \dot{q}_n} \\ \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_2 \partial \dot{q}_1} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_2^2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_2 \partial \dot{q}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_n \partial \dot{q}_1} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_n \partial \dot{q}_2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}_n^2} \end{bmatrix} \tag{4}$$

where n denotes for the number of system segments. The mass matrix $\mathcal{M}(q)$ is derived from this Hessian matrix :

$$\mathcal{M}(q) = \frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2} \tag{5}$$

We used the method of Moore-Penrose pseudo inverse to invert the mass matrix and obtained the pseudo inverse of the mass matrix for solving the acceleration of the system.

$$\ddot{q} = \mathcal{M}^{-1}(q) \cdot \left(\mathcal{I}(q)\tau - \frac{\partial \mathcal{L}}{\partial q} + \mathcal{D}(q, \dot{q})\dot{q}\right) \tag{6}$$

This equation reflects the cumulative effect of both internal and external forces , including conservative and non-conservative forces, as well as the influence of external control inputs τ.

We modify the Euler-Lagrange equation by using the attention mechanism:

$$\ddot{q} = \left(\frac{\partial^2 \mathcal{L}(q, \dot{q})}{\partial \dot{q}^2}\right)^{-1} \left(\text{Attention}(\tau) - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} - \mathcal{D}(q, \dot{q})\dot{q}\right) \tag{7}$$

Here, Attention(τ) denotes the application of the attention mechanism to these forces, allowing the model to prioritize the most influential factors dynamically. The Attention function is defined during the training of the neural network and is tailored to identify and weigh the various forces acting on the system.

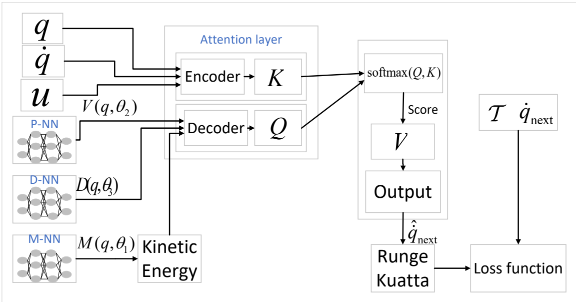

Fig. 1: The structure of the proposed neural networks

3.2 Encoder and Position Encoding

In our proposed architecture, the encoder is a sequence of blocks, each refining the robot's state representation through operations that transcend traditional layer functions. The initial transformation, dubbed 'Input Embedding', projects input variables such as positions, velocities, and control inputs onto a high-dimensional space. We apply Positional Encoding (PE) to the input sequence consisting of the current state q, velocity $\dot{q}$, and control input u:

$$E = \text{Embedding}(q, \dot{q}, u) \tag{8}$$

PE is defined for each element in the sequence:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{9}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{10}$$

where pos represents the position in the sequence and i corresponds to the dimension. The embedded input E_i is then combined with its positional encoding PE_i to form the initial high-dimensional representation E_i' for the robot's state. This representation, acknowledging both the state and its sequence position.

In the design of our neural architecture, the Multilayer Perceptron (MLP) layers play a pivotal role in capturing the nonlinear interactions within the robot's state variables. Each MLP layer, with its weights and biases, transforms the input data into a more abstract representation. Following each linear transformation, a Rectified Linear Unit (ReLU) activation function is applied, introducing the necessary non linearity:

$$\text{ReLU}(x) = \max(0, x) \tag{11}$$

Negative inputs to the function are reset to zero, which can be interpreted as deactivating the corresponding features or neurons. This ensures that only the features with positive activation contribute to the next layer, simplifying the learning process and making it more efficient. By discarding negative activation, it encourages the model to focus on the most salient features that are positively correlated with the robot's dynamic behavior, leading to a sparse and explainable feature space.

The MLP refinement process consists of alternating layers of linear transformations and ReLU activations, expanding and compressing the embedded input data in a manner that allows the network to construct a rich and intricate mapping from the robot's current state and inputs to its subsequent state. This series of operations is formalized as follows:

$$H_1 = \text{ReLU}(\mathbf{W}_1 E_i' + \mathbf{b}_1) \tag{12}$$

$$H_2 = \text{ReLU}(\mathbf{W}_2 H_1 + \mathbf{b}_2) \tag{13}$$

$$\vdots$$

$$K = \text{ReLU}(\mathbf{W}_n H_{n-1} + \mathbf{b}_n) \tag{14}$$

where $\mathbf{W}_j$ and $\mathbf{b}_j$ represent the weights and biases of the j-th layer, H_j represents the output of the j-th layer, and K denotes the final output of the MLP, serving as the output of the encoder layer.

3.3 Decoder and State Prediction

By providing the key (K) for the attention mechanism, the input of the decoder involves preprocessing the state data to align with the decoder's requirements. Three MLP layers are utilized to approximate the kinetic energy matrix $\mathcal{M}$, dissipation matrix $\mathcal{D}$, and state transition matrix $\mathcal{I}$:

$$\hat{\mathcal{M}} = MLP_M(q; \Theta_M) \tag{15}$$

$$\hat{\mathcal{D}} = MLP_D(q, \dot{q}; \Theta_D) \tag{16}$$

$$\hat{\mathcal{I}} = MLP_I(q, \tau; \Theta_I) \tag{17}$$

These layers project the state variables onto a structured form that encapsulates the robot's dynamic properties. The decoder then uses these matrices to perform attention-driven predictions of the robot's future state.

To calculate the attention scores, the decoder's input query (Q) interacts with the keys (K) obtained from the encoder:

$$\text{Attention}_{\text{scores}} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{18}$$

where d_k is the scaling factor to avoid overly large values within the softmax function. The obtained attention scores reflect the importance of each state variable in predicting future dynamics. The predicted next state q_{next} is then computed as a weighted sum of the values V, guided by these scores:

$$q_{\text{next}} = \text{Attention}_{\text{scores}} \cdot V \tag{19}$$

This predicted state passes through additional layers in the decoder, each enhancing the prediction with a non-linear transformation:

$$q' = \text{LayerNorm}(\text{FFN}(q_{\text{next}}) + q_{\text{next}}) \tag{20}$$

$$q_{\text{predicted}} = \text{LayerNorm}(\text{FFN}(q') + q') \tag{21}$$

Finally, the decoder outputs a prediction of future states as a probability distribution, selecting the state with the highest likelihood as the next step:

$$q_{\text{future}} = \text{Softmax}(\text{Linear}(q_{\text{predicted}})) \tag{22}$$

Through this streamlined process, the decoder effectively utilizes the dynamic models and attention mechanism to forecast the system's progression, enabling real-time adaptation to complex dynamics.

3.4 Incorporating Loss Functions for Network Training and Regularization

To maintain compliance with physical principles and ensure the reliability of dynamic predictions, our framework integrates specific loss functions at various stages of the LNN. These loss functions guide the network towards physically plausible solutions and reinforce the accuracy of the learned dynamics.

In the preprocessing process, a loss function is used to ensure accurate approximation of quality, dissipation, and input transformation matrix. The loss function is not directly compared with the objective matrix, but rather obtains the accuracy of predicted states by integrating these matrices into the system's motion equations. This method is consistent with the physical informed properties of the LNN framework and provides a robust approach for learning the dynamic characteristics of the system:

$$L_{\text{preprocess}} = \text{mean}(\|\text{forward}_{\text{model}}(q, qd, \tau) - [q_{\text{next}}, qd_{\text{next}}]\|^2) \tag{23}$$

In this equation, $\text{forward}_{\text{model}}$ denotes the forward model that computes the predicted future states given the current states q and qd, and the control inputs τ, utilizing the approximated matrices $\hat{\mathcal{M}}$, $\hat{\mathcal{D}}$, and $\hat{\mathcal{I}}$. The loss function minimizes the discrepancy between the predicted states and the true next states, q_{next} and qd_{next}, which are provided as part of the training data. This discrepancy serves as a proxy to the accuracy of the approximated matrices and the model's ability to capture the true dynamics of the system.

In the attention mechanism, we focus on the difference between the predicted state of the system obtained by the decoder and the actual subsequent state:

$$L_{\text{state\_prediction}} = \|q_{\text{predicted}} - q_{\text{actual\_next}}\|^2 \tag{24}$$

This loss function ensures that the model's attention-driven predictions are closed to the true system dynamics observed in the data.

The composite loss function for the entire LNN, including preprocessing, attention mechanism, and state prediction phases :

$$L_{\text{composite}} = L_{\text{state\_prediction}} + L_{\text{physics}}(\theta) + L_{\text{matrix}} \tag{25}$$

Here, $L_{\text{physics}}(\theta)$ ensures the predictions adhere to physical principles, and L_{matrix} encourages the structural integrity of the dynamics matrices. The training process guided by this composite loss function satisfies the laws of physics and attention mechanisms. By optimizing this composite loss, LNN has learned to further predict the future state of dynamic systems, ensuring accuracy and physical rationality.

4 Integrated Analysis and Implications

4.1 Simulation Results

In this section, we present a comprehensive analysis of our simulation results, aiming to provide a unified perspective on the performance of the Attention-Enhanced Lagrangian Neural Network (LNN) compared to its Normal counterpart. Our assessment spans various training epochs, noise robustness checks, and long-term predictive capabilities. In our Attention-Enhanced Lagrangian Neural Network (LNN), we denote the number of encoder and decoder layers as E and D respectively, the size of the hidden layers as H, and the number of self-attention heads as A. For the attention mechanism, we specify the dimensional of the 'query', 'key', and 'value' projections as P. We report results on a model with the following configuration: Attention-Enhanced LNN (E=3, D=2, H=128, A=6, P=36).

Training was performed using the Adam optimizer with a learning rate of 0.001, decaying by a factor of 0.96 every 100 epochs. To assess noise robustness, Gaussian noise of varying magnitudes was added to the input data, and the model's performance was monitored accordingly. Training was conducted on a system equipped with a NVIDIA GTX 4090 Laptop GPU, with 16GB of VRAM, and an Intel i9-13980HX CPU with 64 GB RAM. For the Attention-Enhanced LNN model, convergence was achieved within approximately 150 seconds for one epoch of training on a dataset comprising 60000 samples. During the training phase, the peak GPU memory utilization was recorded at 14.4 GB, with the GPU's average usage maintained at 96% throughout the duration of training. In comparison to a conventional model without an attention mechanism, our Attention-Enhanced LNN demonstrated a 15% increase in training time per epoch, which we attribute to the additional computations required for the attention mechanism. However, each epoch of our model is more efficient in learning. Where a standard model may require in excess of 5000 epochs to converge to an error precision nearing 0.1, our method achieves similar accuracy within just 500 epochs. This reduction in the number of required epochs showcases the efficacy of our attention mechanism, despite the increase in per-epoch training time.

The empirical data, as encapsulated in Tab. 1 demonstrate a improvement in trajectory adherence for the Attention-Enhanced LNN. We observed that, as training epochs increased from 200 to 500, the prediction error decreased , indicating an enhanced ability of the model to capture the complex dynamics of the soft robotic arm. This reduction error decrease from 0.5784 to 0.0998, underscores the model's refined predictive accuracy over time.

Further understanding of the training process and the learning efficiency of the model can be obtained from Tab. 1 and Fig. 3. These numbers show the learning curve trajectories of attention enhancement LNN models at different training stages. It is worth noting that Fig.2 shows the initial stage of training, showing an imprecise prediction state as the model begins to adapt to the complexity of the dataset. As the training progresses, Fig. 3 shows a more gradual improvement, marking a shift in the model from learning basic dynamics to improving its understanding of complex actions. The characteristic of this stage is the smoothness of the learning curve, indicating that as the model begins to converge to a robust representation of system dynamics, the learning process tends to stabilize.

Tab. 2 enhances the comparative analysis by quantitatively depicting the epochs needed to converge to a defined accuracy threshold. The Attention-Enhanced LNN, despite not exhibiting a marked advantage in training time, requires significantly fewer epochs to reach comparable precision levels compared to the Normal LNN. However, the expedited learning rate may also introduce a susceptibility to over fit-

Table 1: Prediction Error Across Training Epochs

Training Way	Attention-Enhanced	Attention-Enhanced	Attention-Enhanced	Normal LNN	Attention-Enhanced
Model (Width×Depth)	42×3,5×3,42×2	42×3,5×3,42×2	42×3,5×3,42×2	42×3,5×3,42×2	42×3,5×3,42×2
Sample Number	42000	42000	42000	42000	42000
Training Epoch	200	400	500	500	800
Prediction Error	$0.57(\pm 1.1\times 10^{-3})$	$0.32(\pm 9.4\times 10^{-3})$	$0.09(\pm 8.5\times 10^{-3})$	$0.32(\pm 9.6\times 10^{-3})$	$0.36(\pm 6.5\times 10^{-3})$

Table 2: Training Epoch Required to Achieve Different Accuracies

	Normal LNN				Attention-Enhanced LNN			
Training Epoch	100	500	1000	5500	100	500	1000	5500
Training Time (s)	11873	58934	123343	659278	14366	70131	141647	811495
Simulation Time (s)	30	30	30	30	30	30	30	30
Prediction Error	3.7412	0.7862	0.5316	0.0992	0.8556	0.0998	0.3477	2.4642

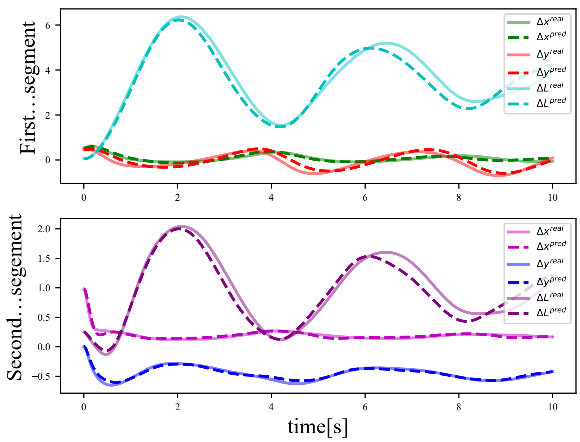

Fig. 2: Differences in positions under 200 training epochs.

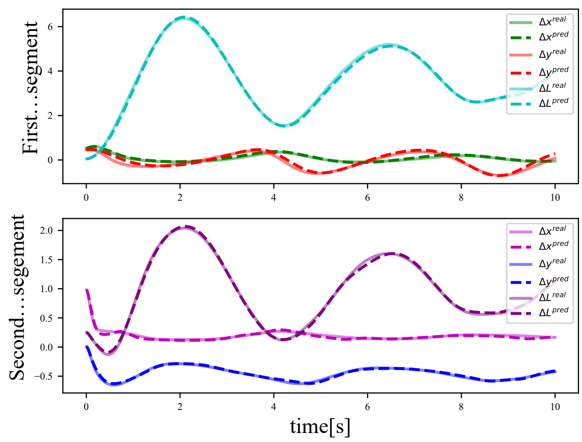

Fig. 3: Differences in positions under 500 training epochs.

ting, particularly at higher epoch counts. In our model without sufficient regularization, it has been observed that for the Attention-Enhanced LNN, the prediction error increases with additional training beyond 500 epochs. This suggests that the model begins to memorize the training data instead of learning to generalize. The convergence patterns shown in Figs. 4 and 5, as well as the training epoch data shown in Tab. 2, collectively emphasize the enhanced learning efficiency provided by attention enhanced LNN.

Figs. 4 and 5 provide a visual representation of this improvement, showing the Attention-Enhanced LNN's trajectory more closely aligning with the reference trajectory compared to the Normal LNN. This alignment not only confirms the quantitative results from Tab. 1 but also shows the qualitative enhancements in the model's behavior. Although both models show signs of divergence in their predicted trajectories as simulation time increases, it is evident that the model with added attention mechanism has smaller errors.

4.2 Model Robustness and Generalization

The robustness against noise was tested by introducing Gaussian white noise with a standard deviation of 0.05 into our simulation environment. Our Attention-Enhanced LNN showcased a advantage in noise resilience. Figs. 6 and 7 depict a subtle yet noticeable superiority in maintaining prediction accuracy despite the added noise. The simulation results collectively point out the effectiveness of integrating attention mechanisms into LNN. These findings indicate that attention enhanced LNNs are not only better at handling complex dynamic behaviors, but also exhibit stronger resilience to environmental noise, which may be a key feature for future applications of fully actuated robot systems in the real world. Our evaluation focuses particularly on the performance of attention enhanced LNN in the presence of Gaussian noise and its ability to maintain prediction accuracy at different training stages. Tab. 2 , as well as Figs. 6 and 7, show a data centric view. During the Gaussian noise interference testing process, white noise with a standard deviation of 0.05 was introduced to test the model. Despite this disturbance, attention enhanced LNN still exhibits relatively good noise resistance. Fig. 6 illustrates this robustness, showing that under the same noise conditions, the prediction error deviation is smaller compared to normal LNN. In terms of generalization, the impact of training period on model performance was quantified. As shown in Tab. 2, attention enhanced LNN achieved lower prediction errors at an earlier stage, indicating a more effective learning process. Fig. 7 further confirms this point, despite the increase in noise in the training data, attention enhanced LNN shows consistent prediction accuracy.

Building on our initial findings regarding the noise re-

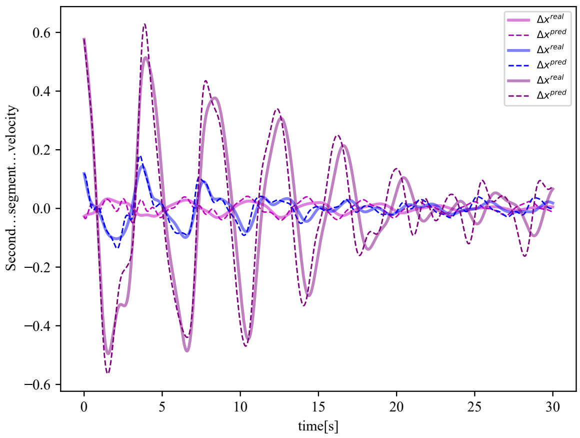

Fig. 4: The velocity of the second segment under Attention-Enhanced 500 training epochs (error 0.18)

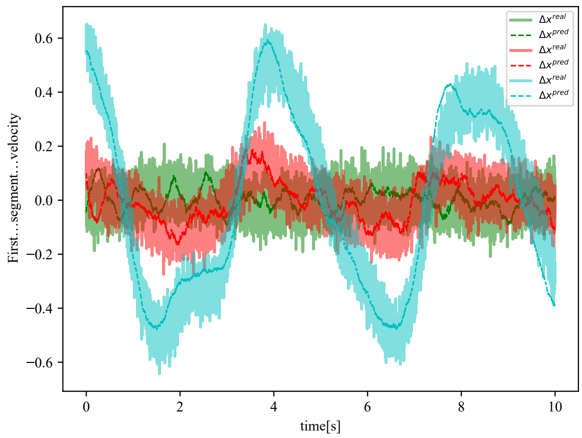

Fig. 6: The velocity of the first segment under Attention-Enhanced LNN (error 0.13)

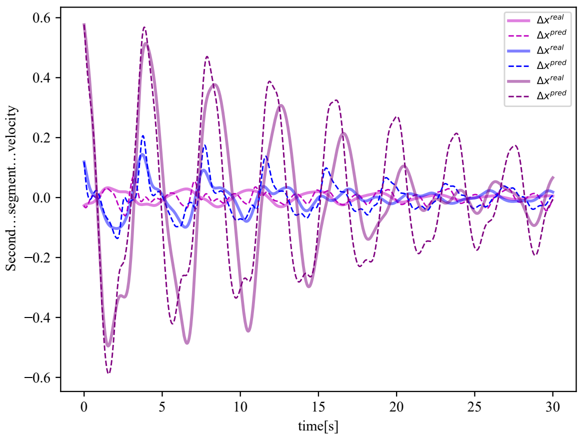

Fig. 5: The velocity of the second segment under normal LNN 500 training epochs (error 0.52)

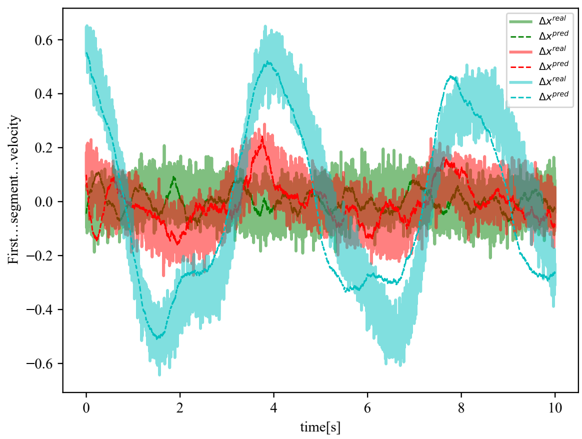

Fig. 7: The velocity of the first segment under Normal LNN (error 0.27)

silience of the Attention-Enhanced Lagrangian Neural Network (LNN). Notably, when introducing a higher level of uniform noise with a standard deviation of 0.1 into the simulation environment, a marked increase in prediction error was observed. This deterioration in performance can be attributed to the more significant impact of noise contamination on the training data, as illustrated in Fig. 8 where the error increased to 2.13. Our study revealed that by augmenting the training dataset size to ten times its original volume, the model's performance improved under identical noise conditions. Fig. 9 showcases this enhancement, with the prediction error notably reduced to 0.9. This improvement underscores the critical role of extensive training data in enhancing model resilience against noise contamination. It suggests that with sufficient data, the Attention-Enhanced LNN can learn to filter out noise-induced anomalies.

4.3 Attention Heat-maps and Dimensional Expansion

In the domain of soft robotics, the articulation and flexibility afforded by each segment of a robotic arm necessitate a multidimensional approach to dynamic modeling. Particularly, for a robotic arm comprised of n segments, where each segment is capable of movement in three-dimensional space—expressed through translations and rotations along the X, Y, and Z axes—the complexity of the system's dynamics significantly increases. This threefold increase in degrees of freedom (DOF) for each segment results in a total of $3n$ DOFs for the entire arm.

The delineation of soft robotic arm dynamics through the mass $\mathcal{M}$, dissipation $\mathcal{D}$, and state transition $\mathcal{I}$ matrices provides a universally applicable framework, adaptable across a wide spectrum of robotic arms configurations. Regardless of the driving mechanisms employed, the materials constituting the arm, or the number of segments it comprises, this matrix representation simplifies the complexity inherent in describing the motion of soft robotic systems. This unified approach enables a consistent and coherent modeling of soft robotics dynamics, facilitating the analysis and control of these highly adaptable and complex systems.

To elucidate the operational dynamics of this mechanism, we employed attention heat-maps. In the context of modeling the dynamics of a two-segment soft robotic arm, atten-

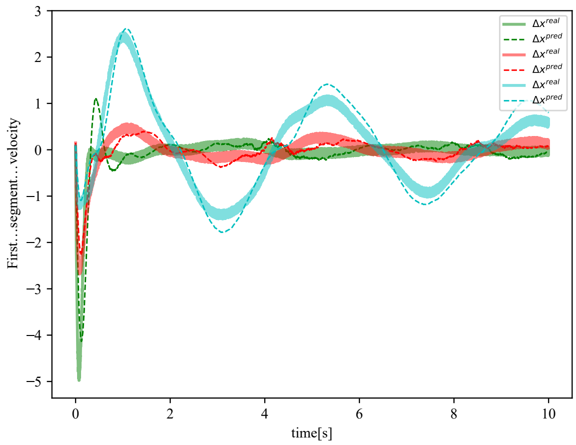

Fig. 8: Uniform noise impact with standard training dataset (error 2.13)

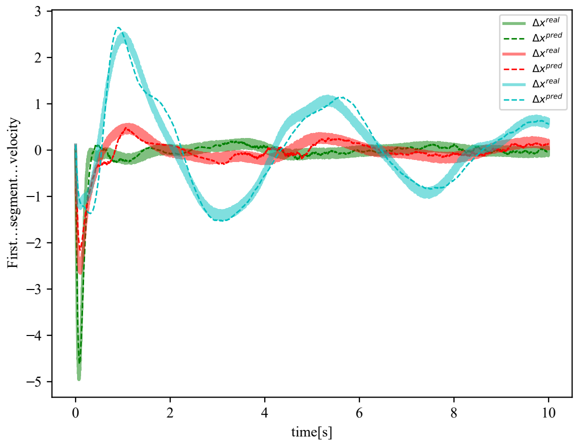

Fig. 9: Uniform noise impact with standard training dataset (error 0.94)

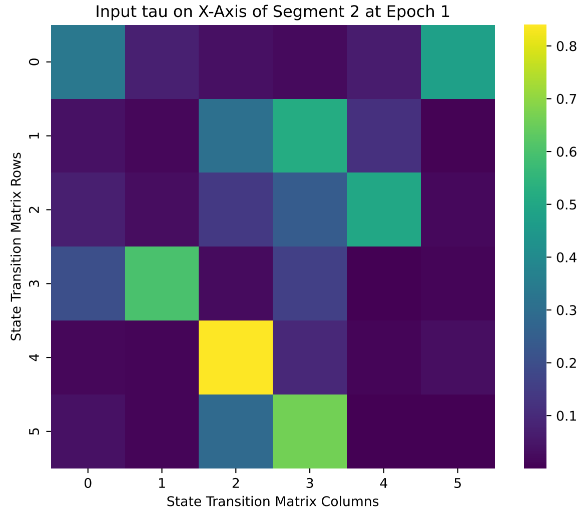

Fig. 10: Attention Distribution for X-Axis Input on Segment 2 at Epoch 1

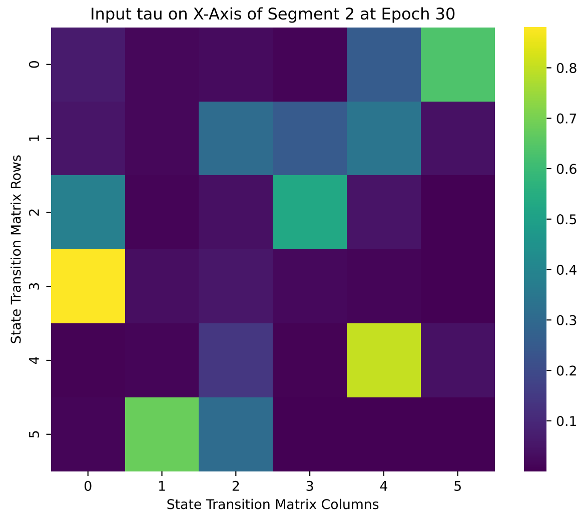

Fig. 11: Attention Distribution for X-Axis Input on Segment 2 at Epoch 30

tion heat-maps provide visualizations of how specific input features influence the system's state transition matrix. Contrary to a broader interpretation of input contributions, each heat-map focuses on the contribution of a single input feature—such as the external force applied in the X direction of the second segment—towards the entire 6×6 state transition matrix $\mathcal{I}$.

The axes of the heat-map correspond to the dimensions of the state transition matrix, $\mathcal{I}$, representing the robotic arm's dynamics. The horizontal and vertical axes both span the matrix's dimensions, reflecting the interrelation between the applied input and the arm's resultant states. Each cell within the heat-map illustrates the magnitude of influence that the selected input feature has on the corresponding element of the state transition matrix. A brighter cell indicates a higher level of contribution, signifying a stronger impact of the specific input feature on the arm's dynamic behavior. By examining heat-maps from different stages of the training process, we observe a transition from an initially uniform or random distribution of attention to a more focused allocation. In the early training phase, as depicted by the heat-map at epoch 1, the attention allocation appears almost stochastic, lacking any discernible pattern. As training progresses to epoch 30, the heat-map begins to reveal emergent patterns of focused attention. These nascent structures signify the model's initial steps towards recognizing the more influential input features. By epoch 100, the heat-map displays a highly refined attention distribution, where the model's focus is concentrated on specific inputs that it has learned to associate with significant impacts on the system's behavior.

This focused exploration of attention heat-maps not only demystifies the model's adaptive learning mechanism but also exemplifies the precision with which it discerns the influence of individual input features on the robotic arm's dynamics.

5 Conclusion

In summary, this study demonstrates that the dynamic system modeling of soft robots has been substantially improved by integrating attention mechanisms into Lagrangian neural networks (LNNs). Our attention enhancement model shows a significant improvement in prediction accuracy and robustness against environmental noise, demonstrating its ability to capture the complex behavior of soft robot systems. The empirical results indicate that attention mechanism reduces

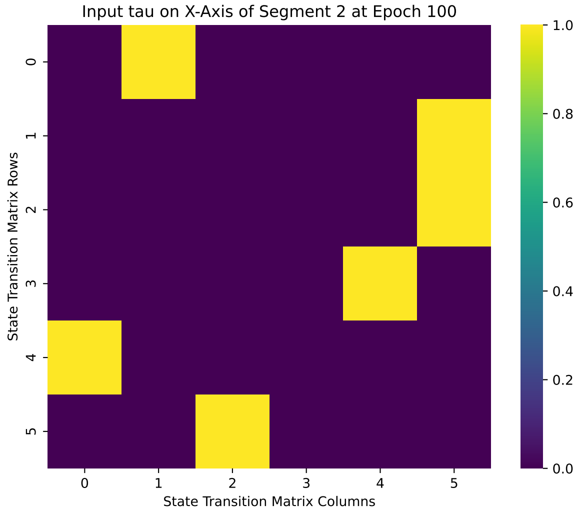

Fig. 12: Attention Distribution for X-Axis Input on Segment 2 at Epoch 100

computational requirements and improves long-term predictive performance. This is consistent with our goal of developing models that are not only efficient in computing resources, but also sufficiently versatile to facilitate real-world applications. Looking ahead to the future, there are several ways for further research. One direct direction is to explore the integration of more complex attention mechanisms, such as sparse attention and local attention. The other direction is to explore the location of adding attention mechanisms, such as introducing attention mechanisms when training physical matrices. In addition, the intersection of soft robots and deep learning provides a new platform for innovative control strategies. Future work may involve deploying these models in real-time control systems and testing their performance in physical environments.

References

[1] A. Banerjee. “The Impact of Soft Robotics In Today’s World: Applications, Challenges Faced, And Future Outlook,” *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)* Dec. 2022

[2] X. Shao, P. Pustina, M. Stölzle, et al. Model-based control for soft robots with system uncertainties and input saturation[J]. *IEEE Transactions on Industrial Electronics*, 2023.

[3] Salvador Cobos et al. “Efficient human hand kinematics for manipulation tasks” 2008 *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[4] DongWook Kim and Yong-Lae Park, ”Contact localization and force estimation of soft tactile sensors using artificial intelligence”, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] Keene Chin, Tess Hellebrekers and Carmel Majidi, ”Machine learning for soft robotic sensing and control”, *Advanced Intelligent Systems*, vol. 2, no. 6, pp. 1900171, 2020.

[6] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[7] J. F. DeRose, J. Wang and M. Berger, ”Attention Flows: Analyzing and Comparing Attention Mechanisms in Language Models,” in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pp. 1160-1170, Feb. 2021.

[8] W. N. White, D. D. Niemann and P. M. Lynch, ”The presentation of Lagrange’s equations in introductory robotics courses,” in IEEE Transactions on Education, Feb. 1989

[9] Jingyue Liu, Pablo Borja, Cosimo Della Santina,”Physics-informed Neural Networks to Model and Control Robots: a Theoretical and Experimental Investigation”,arXiv:2305.05375,Jul 2023

[10] H. Wang and M. Tu, ”Enhancing Attention Models via Multi-head Collaboration,” 2020 International Conference on Asian Language Processing (IALP), Kuala Lumpur, Malaysia, 2020

[11] Miles Cranmer, Sam Greydanus, Stephan Hoyer,”Lagrangian Neural Networks”,arXiv:2003.04630,Jul 2020

[12] J. -Y. Lee, W. -B. Kim, W. -Y. Choi and K. -J. Cho, ”Soft Robotic Blocks: Introducing SoBL, a Fast-Build Modularized Design Block,” in IEEE Robotics and Automation Magazine, vol. 23, no. 3, pp. 30-41, Sept. 2016,

[13] Bryan A. Jones and Ian D. Walker, ”Kinematics for multisection continuum robots”, *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43-55, 2006.

[14] Jeesoo Ha, Dooyoung Kim and Sungho Jo, ”Use of deep learning for position estimation and control of soft glove”, *2018 18th International Conference on Control Automation and Systems (ICCAS)*, 2018.

[15] Liu Wuji et al., ”Distance-directed target searching for a deep visual servo sma driven soft robot using reinforcement learning”, *Journal of Bionic Engineering*, vol. 17, no. 6, pp. 1126-1138, 2020.

[16] Pouya Abdollahzadeh and Saber Azizi, ”On thedynamics of a magnetostriction-based soft robotic manipulator: Closed form and Machine Learning approaches”, *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 2019.

[17] Guanda Li, Jun Shintake and Mitsuhiro Hayashibe, ”Deep Reinforcement Learning Framework for Underwater Locomotion of Soft Robot”, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[18] Sreeshankar Satheeshbabu et al., ”Continuous control of a soft continuum arm using deep reinforcement learning”, *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 2020.

[19] Sudipta Chattopadhyay et al., ”Control of single-segment continuum robots: reinforcement learning vs. neural network based PID”, *2018 International Conference on Control Power Communication and Computing Technologies (ICCPCCT)*, 2018.