TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Zhang, C., Liu, J., Shahabi, E., Pan, W., & Santina, C. D. (2026). End-to-End Learning of Soft-Robot Dynamics from Event-Based Camera Streams. In *Proceedings of the 9th IEEE RAS International Conference on Soft Robotics, ROBOSOFT 2026* (pp. 438-445). IEEE. https://doi.org/10.1109/RoboSoft67810.2026.11522882

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Proceedings of 2026 IEEE
9th International Conference on Soft Robotics (RoboSoft)
Aprilr 7 - 11, 2026, Kanazawa, Japan

2026 IEEE 9th International Conference on Soft Robotics (RoboSoft) | 979-8-3315-8215-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/RoboSoft67810.2026.11522882

End-to-End Learning of Soft-Robot Dynamics from Event-Based Camera Streams

Chuhan Zhang[a,†], Jingyue Liu[a,†], Ebrahim Shahabi[a], Wei Pan[b], Cosimo Della Santina[a,c]

***Abstract*— Modeling and predicting the motion of soft robots remains challenging due to their infinite-dimensional and highly nonlinear dynamics. A promising direction is to learn dynamics directly from high-dimensional sensory streams. Yet, standard RGB cameras suffer from motion blur, making it challenging to capture fast transients and biasing learning toward steady-state behaviors. Here, we turn to event-based cameras, which provide asynchronous, high-frequency visual information better suited to capturing dynamic deformations. We propose a learning architecture that encodes two-channel event frames from a DVS sensor through a convolutional autoencoder while jointly learning a compact latent representation of the robot's dynamics. Within this space, we test several latent models, including a novel spiking–harmonic latent oscillator network (snLON), in which spiking neurons capture the event structure of the data stream and drive a latent Oscillator Network that represents the underlying mechanical dynamics. Validated in both simulation and real-world experiments, the proposed framework predicts long-horizon soft-robot motion with high accuracy and consistency from a single initial event frame and control sequence.**

I. Introduction

Soft robots offer a demanding benchmark for data-driven modeling. Their rich, highly nonlinear, and spatially distributed behavior stubbornly resists simple approximation [1]–[3]. In recent years, vision-based learning has emerged as a powerful strategy to confront this complexity, learning compact dynamical representations directly from high-dimensional sensory streams to capture the full distribution of the robot's shapes and motions [4]–[6]. Vision provides dense, spatially resolved state information, revealing the complete deformation field without intrusive sensing or changes to the robot's mechanics. This makes it a natural tool for modeling systems whose behavior is intrinsically distributed. Recent studies have leveraged this idea in various ways. For example, some predict deformation or shape directly from input frames [7], [8], others employ camera feedback in eye-in-hand configurations for visual servoing [9]–[11]. In contrast, [12] learns soft-robot dynamics directly from image sequences, recovering low-dimensional yet physically interpretable models suitable for control.

To exploit visual observations effectively, recent work has drawn on the broader world-model paradigm [13], where autoencoders are combined with low-dimensional latent dynamics to learn end-to-end representations of the physical world from pixels [14]–[16]. Applications of these concepts to soft robotics include [17], which uses multilayer perceptrons to model latent dynamics, and [18], [19], which adopts recurrent networks that integrate visual and proprioceptive cues. Beyond purely neural approaches, physics-inspired formulations have been proposed to govern the latent evolution [20]. Models based on harmonic oscillators [21]–[23] or Koopman operators [24] can also be used to introduce well-structured temporal dynamics, but have not yet been tested in this context.

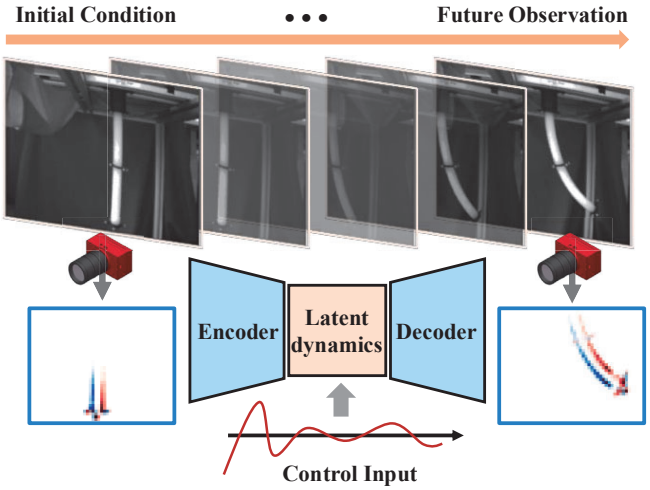


Fig. 1. Pictorial representation of the task addressed in this work and the overall methodology. The goal is to learn a dynamic representation of a soft robot directly from pixel space—that is, to predict the full future evolution from a single image and the control input. Instead of relying on RGB video, we employ event-based camera data, which is better suited to capturing rapid motions and transients. The proposed framework combines an autoencoder with a trainable latent dynamical model, explored in four variants.

However, all these approaches rely exclusively on RGB cameras. This is an issue because these cameras are prone to motion blur and obscure fast transients [25], making them ill-suited to capture the rapid deformations that occur when the robot moves between steady states. As a result, learning from RGB streams inherently biases models toward static behaviors undermining their dynamic nature. In this work, we propose turning to event-based cameras, which provide sparse, low-latency, and high-dynamic-range data by reporting only local brightness changes with microsecond

† These authors contributed equally to this work.

[a] The authors are with the Cognitive Robotics Department, Delft University of Technology, Delft, The Netherlands. [b] The authors are with the Department of Computer Science, The University of Manchester, UK. [c] The authors are also with the Institute of Robotics and Mechatronics German Aerospace Center (DLR) Oberpfaffenhofen 82234, Germany.

The work was supported under the European Union's Horizon Europe Program from Project EMERGE - Grant Agreement No. 101070918.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

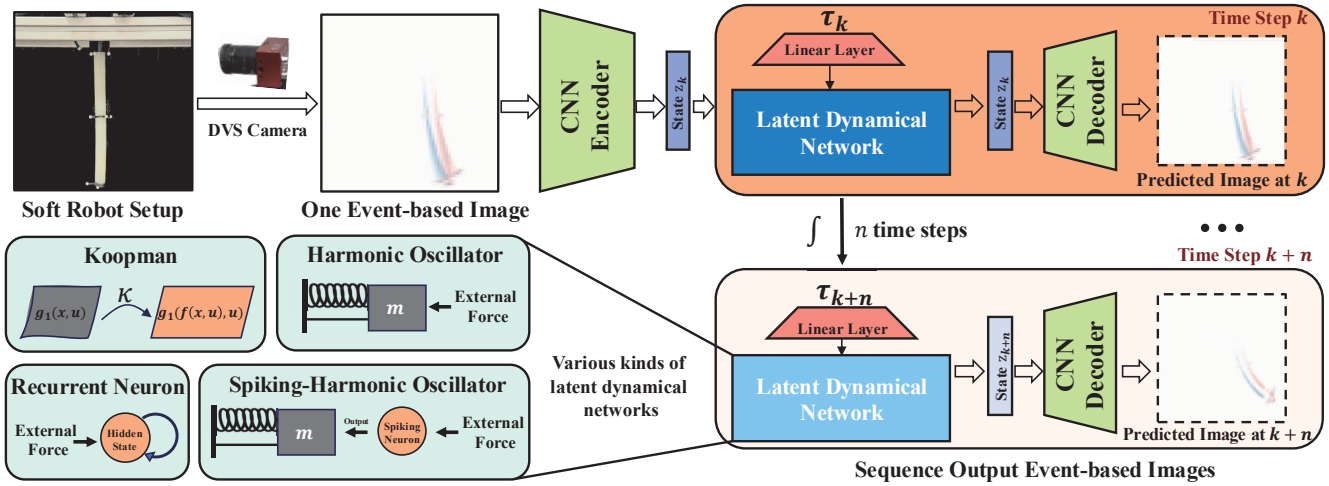


Fig. 2. Overview of the event-driven framework for soft robot motion prediction. A DVS camera captures events from the moving soft robot. These events are combined into an event-based image $\mathbf{I}_k$ and fed into a CNN-based encoder to extract a low-dimensional state $\mathbf{z}_k$. The corresponding states $\{\mathbf{z}_k, \mathbf{z}_{k+1}, ..., \mathbf{z}_{k+n}\}$ are integrated by the physical dynamics of the latent dynamical network using sequential inputs $\{\boldsymbol{\tau}_k, \boldsymbol{\tau}_{k+1}, ..., \boldsymbol{\tau}_{k+n}\}$. These states are then passed through a CNN decoder to generate a sequence of event-based images, providing the predicted motion of the soft robot. We provide different kinds of latent dynamical networks, including Koopman, RNNs like LSTM, harmonic oscillator networks, and the proposed snLON.

precision [26], [27]. We propose, for the first time, to exploit these data streams for pixel-to-pixel dynamic model learning, as shown in Fig. 1.

We introduce an autoencoder architecture and compare four variants, each relying on a different class of latent dynamical networks: LSTM, Koopman, Harmonic Oscillator Networks [28], and the proposed Spiking–Harmonic Oscillator Network (snLON). While all models are novel in the context of learning from event-based data, snLON is introduced here for the first time. It is specifically designed for settings that require architectures capable of processing asynchronous inputs while preserving the physical structure of the underlying motion. In this formulation, spiking neurons encode the event-driven control input, while the soft robot's latent state evolves according to a harmonic oscillator, yielding a biologically inspired yet physically grounded dynamic core. We extensively evaluate performance across four new datasets introduced in this work — two synthetic and two experimental — describing dynamic motions of two soft robots: one actuated via a revolut joint at its base and the other via tendons. Among all compared models, the proposed snLON consistently achieves the most accurate and robust long-horizon predictions of soft-robot motion.

II. Methodology

In this section, the overview of the framework is introduced in Sec. II-A. Sec. II-B illustrates the LSTM-based dynamical network. Subsequently, Sec. II-C shows using Koopman in the latent dynamical network. Sec. II-D illustrates the harmonic oscillator network, and Sec. II-E shows the proposed snLON.

Algorithm 1 Pseudocode for the training process of the event-driven framework for soft robot motion prediction.

Result: Generate the predicted event-based images $\texttt{I_pred}_k$, $\texttt{I_pred}_{k+1}$, $\texttt{I_pred}_{k+2}$, and the integrated states $\texttt{z_int}_{k+1}$ and $\texttt{z_int}_{k+2}$ from the latent dynamical network.

Input: Event-based images $\texttt{I}_k$, $\texttt{I}_{k+1}$, $\texttt{I}_{k+2}$, external input τ_k, τ_{k+1}, latent dynamical network `dyn_net`, position dimension in latent space `pos_dim`.

$\texttt{z}_k \leftarrow \text{encoder}(\texttt{I}_k)$, $\texttt{z}_{k+1} \leftarrow \text{encoder}(\texttt{I}_{k+1})$,
$\texttt{z}_{k+2} \leftarrow \text{encoder}(\texttt{I}_{k+2})$
$\texttt{z_int}_{k+1} \leftarrow \text{SolveODE}(\texttt{dyn_net}, \texttt{z}_k, \tau_k)$
$\texttt{z_int}_{k+2} \leftarrow \text{SolveODE}(\texttt{dyn_net}, \texttt{z_int}_{k+1}, \tau_{k+1})$
$\texttt{I_pred}_k \leftarrow \text{decoder}(\texttt{z}_k[:\texttt{pos_dim}])$
$\texttt{I_pred}_{k+1} \leftarrow \text{decoder}(\texttt{z_int}_{k+1}[:\texttt{pos_dim}])$
$\texttt{I_pred}_{k+2} \leftarrow \text{decoder}(\texttt{z_int}_{k+2}[:\texttt{pos_dim}])$

Return $\texttt{I_pred}_k$, $\texttt{I_pred}_{k+1}$, $\texttt{I_pred}_{k+2}$, $\texttt{z_int}_{k+1}$, $\texttt{z_int}_{k+2}$

A. Framework Overview

Fig. 2 presents an overview of the event-driven framework for soft robot motion prediction. After data preprocessing, the event streams from the DVS camera are converted into a single event-based image that provides both position and velocity information of the soft robot. A CNN-based encoder then transforms this image into a low-dimensional latent space $\mathbf{z}_k$, where k represents the discrete time step. It is worth noting that the dimensionality of the latent space is much lower than that of the original image space. The state $\mathbf{z}_k$, together with the external control input $\boldsymbol{\tau}_k$ after the linear transform, is passed into the latent dynamical

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

network. By integrating over n time steps, the future latent state $\mathbf{z}_{k+n}$ is obtained. A CNN-based decoder then reconstructs a sequence of predicted robot motions from the latent states. Therefore, with this pipeline, the hidden state of the latent dynamical network can be initialized using only a single event-based image. The control input operates within the latent dynamical network, allowing the low-dimensional latent space to evolve over time through the network's dynamics and map back to the original high-dimensional image space. We adopt a three-time-step rollout for the network to train. For training, we use two loss terms. The first is an image reconstruction loss, where the predicted event-based image is compared with the ground-truth image using a binary cross-entropy (BCE) loss. The second loss is defined in the latent space: we compute the difference between the integrated latent state $\mathbf{z}_{\text{pred}}$, obtained by evolving the oscillator dynamics, and the encoded latent state z extracted directly from the CNN encoder using input images. This latent consistency loss helps align the learned dynamics with the encoder's representation, ensuring that the latent state evolves smoothly and remains physically meaningful over time. The overview pseudo-code of the algorithm is provided in Algo. 1. In the following sections, different types of latent dynamical networks are explained and compared.

B. LSTM

Given the temporal mechanism of RNNs, their hidden states can also be iterated over time steps. Therefore, in this work, we also consider RNNs as the baseline model. For better performance, the more complex RNN network, LSTM, is employed. The iteration of the $2l$-dimension hidden state $\mathbf{z} \in \mathbb{R}^{2l}$ is learned purely through the following equation:

$$
\begin{aligned}
\mathbf{i}_{k+1} &= \sigma(\mathbf{A}_{\mathrm{i}}\mathbf{h}_k + \mathbf{B}_{\mathrm{i}}\boldsymbol{\tau}_{k+1} + \mathbf{b}_{\mathrm{i}}),\\
\mathbf{f}_{k+1} &= \sigma(\mathbf{A}_{\mathrm{f}}\mathbf{h}_k + \mathbf{B}_{\mathrm{f}}\boldsymbol{\tau}_{k+1} + \mathbf{b}_{\mathrm{f}}),\\
\mathbf{o}_{k+1} &= \sigma(\mathbf{A}_{\mathrm{o}}\mathbf{h}_k + \mathbf{B}_{\mathrm{o}}\boldsymbol{\tau}_{k+1} + \mathbf{b}_{\mathrm{o}}),\\
\hat{\mathbf{c}}_{k+1} &= \tanh(\mathbf{A}_{\mathrm{c}}\mathbf{h}_{\mathrm{k}} + \mathbf{B}_{\mathrm{c}}\boldsymbol{\tau}_{\mathrm{k+1}} + \mathbf{b}_{\mathrm{c}}),\\
\mathbf{c}_{k+1} &= \mathbf{f}_{k+1} \odot \mathbf{c}_k + \mathbf{i}_{k+1} \odot \hat{\mathbf{c}}_{k+1},\\
\mathbf{h}_{k+1} &= \mathbf{o}_{k+1} \odot \tanh(\mathbf{c}_{\mathrm{k+1}}),\\
\mathbf{z}_{k+1} &= (\mathbf{h}_{k+1}, \mathbf{c}_{k+1}),
\end{aligned}
\tag{1}
$$

where $\mathbf{A}_* \in \mathbb{R}^{l\times l}$, $\mathbf{B}_* \in \mathbb{R}^{l\times m}$ for control input $\boldsymbol{\tau}_k \in \mathbb{R}^m$ at time step k. $\mathbf{b}_* \in \mathbb{R}^l$ are the bias vectors. $\sigma(\cdot)$ is the Sigmoid function, which is applied to calculate the input gate $\mathbf{i}_k \in \mathbb{R}^l$, forget gate $\mathbf{f}_k \in \mathbb{R}^l$, and output gate $\mathbf{o}_k \in \mathbb{R}^l$. Tanh activation function is used for generating candidate cell state $\hat{\mathbf{c}}_k \in \mathbb{R}^l$. The latent state $\mathbf{z}_k$ is formed by concatenating the hidden state $\mathbf{h}_k \in \mathbb{R}^l$ and the cell state $\mathbf{c}_k \in \mathbb{R}^l$. $\mathbf{A}_*$, $\mathbf{B}_*$, and $\mathbf{b}_*$ are the learnable parameters. With the activation functions, the LSTM-based latent dynamical network can provide a rich nonlinear representation.

C. Koopman (control-affine latent dynamics)

This approach leverages the conceptual foundations of modern Koopman theory [29], which emphasizes linear representations of nonlinear dynamics via learned coordinate transformations. This allows the latent dynamics to evolve approximately linearly, facilitating multi-step prediction, in a manner analogous to Koopman operator embeddings. The evolution of $2l$-dimensional latent state $\mathbf{z} \in \mathbb{R}^{2l}$ according to the linear, control-affine transition with a bias term:

$$
\mathbf{z}_{k+1} = \mathbf{A}\mathbf{z}_k + \mathbf{B}\boldsymbol{\tau}_k + \mathbf{b}, \tag{2}
$$

where $\mathbf{A} \in \mathbb{R}^{2l\times 2l}$, $\mathbf{B} \in \mathbb{R}^{2l\times m}$ for control input $\boldsymbol{\tau}_k \in \mathbb{R}^m$, and $\mathbf{b} \in \mathbb{R}^{2l}$ is a bias vector.

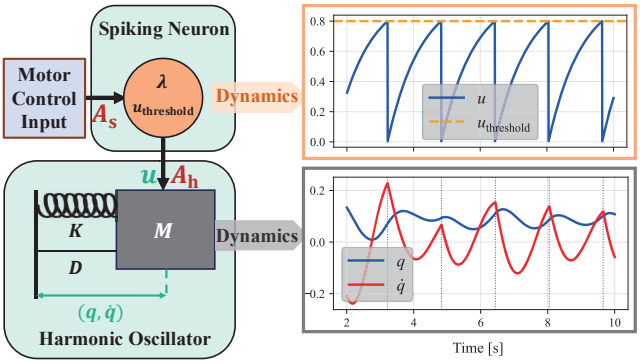


Fig. 3. Schematic representation of snLON. On the left, we show the structure of snLON. The proposed Spiking Harmonic Oscillator Networks latent dynamic model mixes spiking neurons and harmonic oscillators to simultaneously represent the event-based nature of the data stream and the physical characteristics of the underlying dynamics. On the right, we show the membrane potential of the spiking neuron and the corresponding state of the harmonic oscillator under a constant external input of 1. The figure illustrates that the interaction between the neuron and the oscillator produces diverse temporal dynamics. This indicates that the combined system can represent a wide range of behaviors in the latent space.

D. Harmonic Oscillator Network (nLON)

Harmonic oscillator mechanics are famous for their stability and meaningful physical representation. We leverage the coupled *linear* mechanical oscillator latent model with stiffness interconnection to capture the oscillatory structure of soft-robot motion. The dynamics of $2l$-dimensional latent state $\mathbf{z} \in \mathbb{R}^{2l}$ are constrained to follow a structured harmonic oscillator form with positive-definite mass and stiffness matrices and bounded damping as:

$$
\begin{aligned}
&\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{D}\,\dot{\mathbf{q}} + \mathbf{K}\,\mathbf{q} = \mathbf{A}\,\boldsymbol{\tau},\\
&\mathbf{z} := [\mathbf{q}^\top\ \dot{\mathbf{q}}^\top]^\top,
\end{aligned}
\tag{3}
$$

where $\mathbf{M} \succ 0$, $\mathbf{D} \succeq 0$, $\mathbf{K} \succ 0$, and $\mathbf{A} \in \mathbb{R}^{l\times m}$ maps control inputs $\boldsymbol{\tau} \in \mathbb{R}^m$ to latent generalized forces. $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$, and $\mathbf{A}$ are learnable parameters. This structure biases the model toward passive, well-conditioned dynamics.

E. Spiking Harmonic Oscillator Networks (snLON)

The latent dynamical networks described above encode the position and velocity information from the input event-based image into the latent space, while ignoring the corresponding expression of the external control command $\boldsymbol{\tau}$. In this

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

section, we use bio-inspired spiking neurons to encode the control input of the motor into the latent space $\mathbf{z} \in \mathbb{R}^{3l}$. Just as the output of the motors drives the motion of a soft robot, the output of the spiking neuron drives the motion of the harmonic oscillator in the latent space. The dynamics of the proposed snLON (see Fig. 3) are as below:

$$\begin{aligned} &\dot{\mathbf{u}} = \frac{1}{\boldsymbol{\lambda}} \odot (-\mathbf{u} + \mathbf{A}_{\mathrm{s}}\boldsymbol{\tau}), \\ &\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{D}\,\dot{\mathbf{q}} + \mathbf{K}\,\mathbf{q} = \mathbf{A}_{\mathrm{h}}\mathbf{u}, \\ &\text{if } \mathbf{u} > \mathbf{u}_{\mathrm{threshold}},\ \mathbf{u} = \mathbf{u} - \mathbf{u}_{\mathrm{threshold}}, \\ &\mathbf{z} := [\mathbf{q}^\top\ \mathbf{u}^\top\ \dot{\mathbf{q}}^\top]^\top, \end{aligned} \tag{4}$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$, and $\boldsymbol{\tau}$ are same to the definition in Sec. II-D. $\mathbf{A}_{\mathrm{s}} \in \mathbb{R}^{l\times m}$ maps the external control input $\boldsymbol{\tau}$ to the stimulus of the spiking neuron, and $\mathbf{A}_{\mathrm{h}} \in \mathbb{R}^{l\times l}$ maps the spiking neuron output to the harmonic oscillator forces. $\mathbf{u} \in \mathbb{R}^l$ represents the membrane potential of the spiking neuron, and $\boldsymbol{\lambda} \in \mathbb{R}^l$ is the decay rate. The soft reset mechanism [30] is utilized in snLON: when the membrane potential exceeds the threshold potential $\mathbf{u}_{\mathrm{threshold}} \in \mathbb{R}^l$, the membrane potential is reset by reducing the threshold potential. In addition to the learnable parameters of the harmonic oscillator described in Sec. II-D, the spiking neurons also include $\boldsymbol{\lambda}$ and $\mathbf{u}_{\mathrm{threshold}}$ as learnable parameters.

III. Experiments and Results

In this section, the experiment details are introduced. Sec. III-A shows the simulation and real robot setups for two scenarios. Data processing is introduced in Sec. III-B, and the network structure is shown in Sec. III-C.

A. Experiment Scenarios

Two experiment scenarios are designed for validating: Direct Actuation (DA) and Cable Actuation (CA). Fig. 4 describes the hardware details for the experiments. Both soft robotic arms in the DA and CA scenarios are fabricated with Dragon Skin 30, a silicone chosen for its high flexibility and long-lasting durability. This material forms the primary structure of the arm, allowing it to move smoothly and adapt to both air and underwater environments. The Dynamixel servo motors provide actuation. For the DA scenario, one Dynamixel servo motor is fixed at the bottom of the robotic arm to provide rotational motion. For the CA scenario, two Dynamixel servo motors control two cables respectively, allowing the soft robotic arm to swing within a plane. Additional structural components and connectors are produced using a Prusa 3D printer. These pieces are designed to fit well with both the arm and the motor while maintaining a modular assembly. The Inivation DAVIS346 camera is used to collect the event streams during the soft robot's movement.

The simulation environment is also used to evaluate the proposed models. In this work, the soft robotic arms are modeled using Cosserat rod theory and the Elastica framework [31]. Since event streams cannot be directly generated from the simulation, we apply the v2e toolbox [32] to convert simulated RGB frames into event data. To ensure smoother event conversion, the original frames are first interpolated using the Super SloMo model [33].

B. Data Processing

The input event stream is represented in the form (x, y, p, t), where (x, y) denotes the pixel location of the event, p indicates the polarity corresponding to an increase or decrease in brightness, and t is the timestamp at which the event occurs. The event data are first segmented into fixed time windows of length T. Each window is further divided into K equal sub-intervals to preserve temporal structure. For every sub-interval, positive and negative events are accumulated separately into two count maps I_{p} and $I_{\mathrm{n}} \in \mathbb{R}^{H\times W}$. This results in two sequences of K count tensors, one per polarity. To obtain a compact two-channel event representation, the count maps across the K sub-intervals are combined using a logarithmic time-decay weighting, which places slightly higher emphasis on more recent activity. The resulting encoded representation forms an event frame $I \in \mathbb{R}^{2\times H\times W}$, where the first channel corresponds to positive events and the second to negative events. A spatial cropping and resizing step is then applied to produce an input image of size $I \in \mathbb{R}^{2\times 64\times 64}$. This procedure converts the asynchronous event stream into a structured spatiotemporal representation suitable for neural network processing, while retaining both polarity information and temporal dynamics. In this work, the time window T for event data processing is set as 0.04s, and the interval number K is set as 5.

C. Network Structures

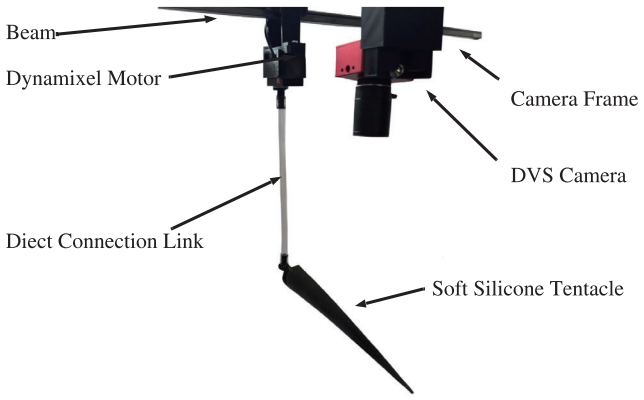
The encoder consists of three convolutional layers with a kernel size of 4×4 and a stride of 2, each followed by batch normalization and a Swish activation. These layers progressively reduce the spatial resolution and expand the feature dimension from 32 to 64 to 128 channels. The resulting feature map is flattened and passed through a linear layer to produce the latent state $\mathbf{z}$. The decoder mirrors the encoder structure in reverse. The latent state from the latent dynamical network is first mapped back to the feature map size using a fully connected layer and reshaped accordingly. Then, three transposed convolutional layers with batch normalization and Swish activation are used to upsample the feature map back to the original resolution. The final transposed convolution outputs ten channels, corresponding to five positive and five negative temporal slices. These slices are combined using a fixed logarithmic time–decay weighting to produce the final two–channel reconstructed event image. In this work, we set the dimensionality of the latent state $\mathbf{z}$ to 3.

D. Results

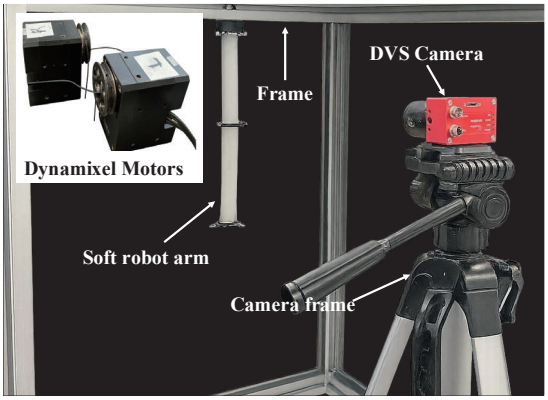
During testing, only the initial event-based image at the first time step is provided to the model. With the given control input sequence $\boldsymbol{\tau}$, the model then evolves autonomously and generates the subsequent sequence of output images.

The quantitative results of four different kinds of latent dynamical networks on four datasets are shown in Tab. I.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.



(a) Experimental Setup #1 (DA) [28]



(b) Experimental Setup #2 (CA)

Fig. 4. Illustration of the experimental setups used to collect the two real-world datasets. Panel (a) Direct Actuation (DA) scenario: a silicone tentacle is connected to a single Dynamixel motor through a rigid link, which provides the sole control input. The DVS camera is mounted along the beam to record the event stream. Panel (b) Cable Actuation (CA) scenario: a silicone arm is attached to a fixed frame and actuated by two Dynamixel motors controlling independent tendons, enabling two-dimensional motion.

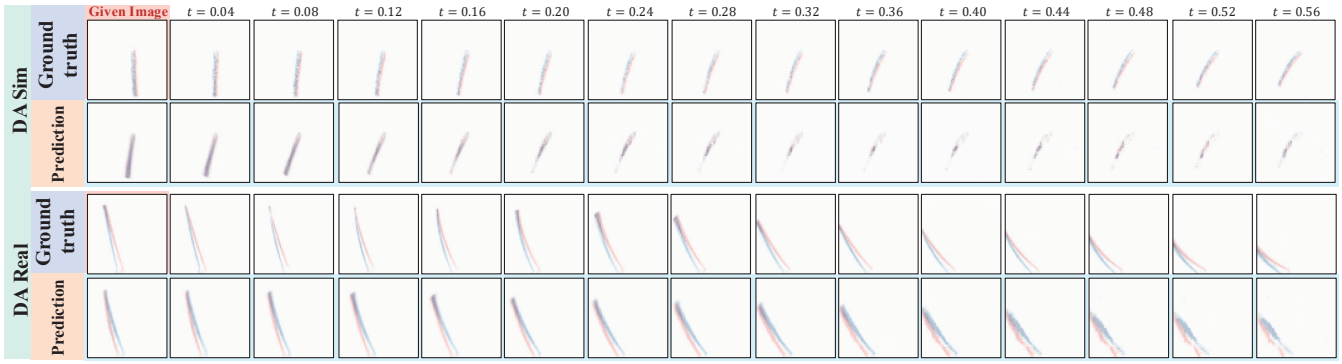


Fig. 5. Visualization of the proposed snLON's prediction performance over 14 time steps on the DA scenario. The first two rows display the simulation environment results, and the last two rows present the real experiment results. In the test, only the image from the first time step is given to the model. The snLON encoder then mapped this image into the latent space. At each following step, a one-dimensional control input from the DA scenario, which controls the rotation, is applied. The spiking harmonic oscillator dynamics then evolve in the latent space, and the model decodes these latent states to reconstruct event-based images over time. Even though the model received only a single initial image, it was able to produce accurate predictions over a 14-step sequence.

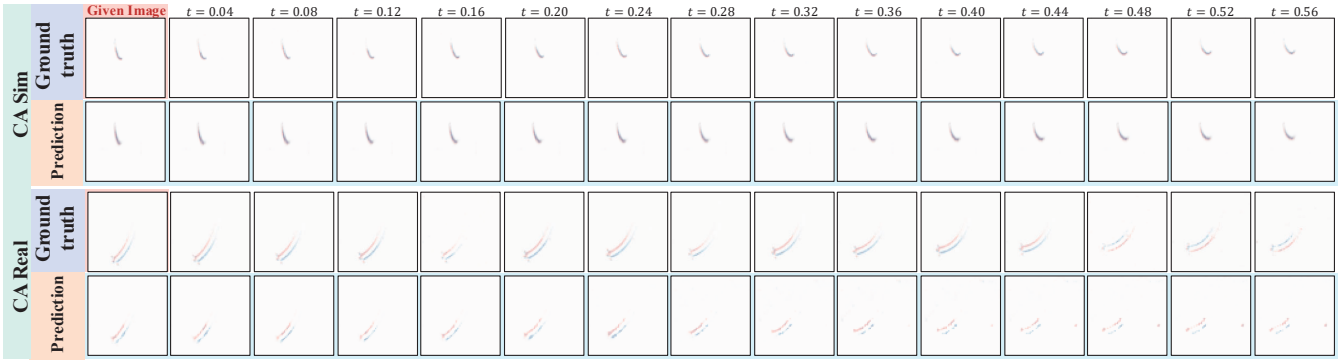


Fig. 6. Visualization of the proposed snLON's prediction performance over 14 time steps on the CA scenario. The first two rows display the simulation environment results, and the last two rows present the real experiment results. In the CA scenario, only the first image in the sequence is provided to the model. The snLON encodes this image into the latent space. Then, at each time step, a two-dimensional motor command that controls the 2D motion is applied. The latent dynamics evolve based on the external inputs, and the decoder generates the corresponding event-based images.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 27,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

TABLE I

QUANTITATIVE RESULTS OF THE PREDICTION PERFORMANCE FOR SIMULATION AND REAL EXPERIMENT ON DA AND CA SCENARIOS. WE TEST FOUR DIFFERENT LATENT DYNAMICAL NETWORKS AND USE PSNR, SSIM, AND MSE AS THE METRIC. NOTE THAT ALL FOUR LATENT DYNAMICS HAVE NEVER BEEN USED IN THIS CONTEXT BEFORE. FURTHERMORE, SNLON IS COMPLETELY NEW TO THIS WORK. WE COMPARED THE AVERAGE METRIC VALUE OVER THE ROLLOUTS WITH THE 14-STEP PREDICTION. ALSO, TO SHOW THE LONG-HORIZON PREDICTION PERFORMANCE, WE COMPARED THE METRIC AT THE FINAL TIME STEP. WE HIGHLIGHT THE BEST-PERFORMING MODEL IN EACH METRIC IN BOLD, AND THE SECOND-BEST MODEL IS INDICATED WITH UNDERLINING.

Scenario	Model	Average over rollout			Final step		
		PSNR↑	SSIM↑	MSE ↓	PSNR↑	SSIM↑	MSE ↓
DA Sim	LSTM	24.0321 ± 2.5983	0.5400 ± 0.0366	4.6150 ± 2.3931	24.9262 ± 2.4192	0.5278 ± 0.0325	3.7730 ± 2.1420
	Koopman	23.9611 ± 3.3290	0.8279 ± 0.0610	5.0668 ± 3.0513	23.3332 ± 1.8922	0.7898 ± 0.0274	5.0750 ± 2.0260
	nLON	24.6453 ± 3.8634	0.8316 ± 0.0523	4.7347 ± 3.4144	25.2983 ± 3.7535	0.8233 ± 0.0434	4.0011 ± 2.7243
	snLON (ours)	**25.3187 ± 3.9221**	**0.8569 ± 0.0337**	**3.9893 ± 2.6543**	**25.6775 ± 3.5864**	**0.8467 ± 0.0430**	**3.6665 ± 2.6781**
DA Real	LSTM	**26.7444 ± 3.6489**	0.7251 ± 0.0643	**2.8861 ± 2.1797**	**26.5006 ± 3.8631**	0.7017 ± 0.0647	**3.2747 ± 3.0302**
	Koopman	25.8674 ± 3.0177	0.8535 ± 0.0663	3.2513 ± 2.2205	26.0647 ± 3.3806	**0.8292 ± 0.0348**	3.3543 ± 2.9130
	nLON	25.9873 ± 3.4673	0.8561 ± 0.0651	3.3460 ± 2.4303	25.1548 ± 3.2511	0.8164 ± 0.0908	4.0616 ± 3.4384
	snLON (ours)	25.2652 ± 3.7411	**0.8643 ± 0.0764**	4.1782 ± 3.4634	24.7883 ± 3.3838	0.8235 ± 0.0782	4.3648 ± 3.1339
CA Sim	LSTM	41.2836 ± 2.8800	0.8979 ± 0.0189	0.0884 ± 0.0424	41.7276 ± 2.4396	0.8997 ± 0.0106	0.0779 ± 0.0396
	Koopman	**43.2016 ± 3.5252**	**0.9675 ± 0.0156**	0.0609 ± 0.0338	42.5238 ± 3.4843	0.9622 ± 0.0171	0.0701 ± 0.0359
	nLON	42.4293 ± 2.4610	0.9649 ± 0.0147	0.0661 ± 0.0337	41.5657 ± 3.5708	0.9572 ± 0.0192	0.0891 ± 0.0475
	snLON (ours)	42.9631 ± 2.6737	0.9655 ± 0.0111	**0.0596 ± 0.0319**	**43.8142 ± 3.3845**	**0.9682 ± 0.0114**	**0.0530 ± 0.0331**
CA Real	LSTM	33.7161 ± 6.4992	0.8329 ± 0.0601	0.9724 ± 1.3058	33.2780 ± 6.3305	0.8380 ± 0.0329	1.0534 ± 1.2588
	Koopman	29.7731 ± 2.8158	0.7680 ± 0.1197	1.3131 ± 1.0447	28.3403 ± 1.5692	0.6506 ± 0.1529	1.5606 ± 0.5360
	nLON	**34.1806 ± 6.0453**	**0.8815 ± 0.0602**	**0.8642 ± 1.3553**	31.2961 ± 3.5701	0.7842 ± 0.2010	0.9778 ± 0.6184
	snLON (ours)	33.5149 ± 5.3975	0.8514 ± 0.0461	0.9382 ± 1.4062	**33.8073 ± 6.8167**	**0.8454 ± 0.0514**	**0.8117 ± 0.6267**

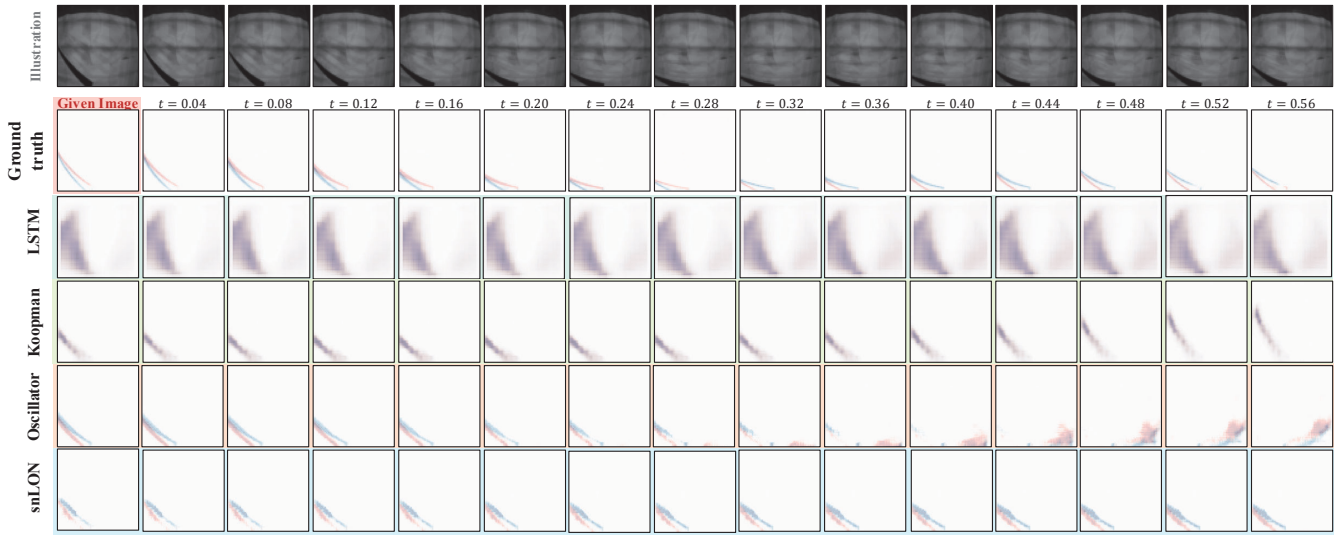


Fig. 7. Visualization comparison among four different latent dynamical networks on the DA scenario. The framework predicts over 14 time steps using only the initial image and the τ sequence as input. To better show the motion of the soft robot, we display the corresponding grayscale images in the first row. Only the first ground-truth event-based image is provided to the model. The following rows show the results from LSTM, Koopman, the Oscillator Network, and the proposed snLON. The results show that snLON maintains clearer predictions over longer time horizons. In contrast, the outputs from LSTM become blurry, and both Koopman and the Oscillator Network show noticeable blurring or incorrect motion directions.

Peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [34], and mean squared error (MSE) are utilized to compare the similarity between predicted rollout and the ground truth. As shown in Table I, the proposed snLON model provides consistently strong performance across both simulation and real-world experiments. In the simulation scenarios, it achieves the highest PSNR and SSIM values and the lowest MSE, demonstrating improved stability and reduced error accumulation over long rollouts when compared to the baselines. In the real experiments,

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

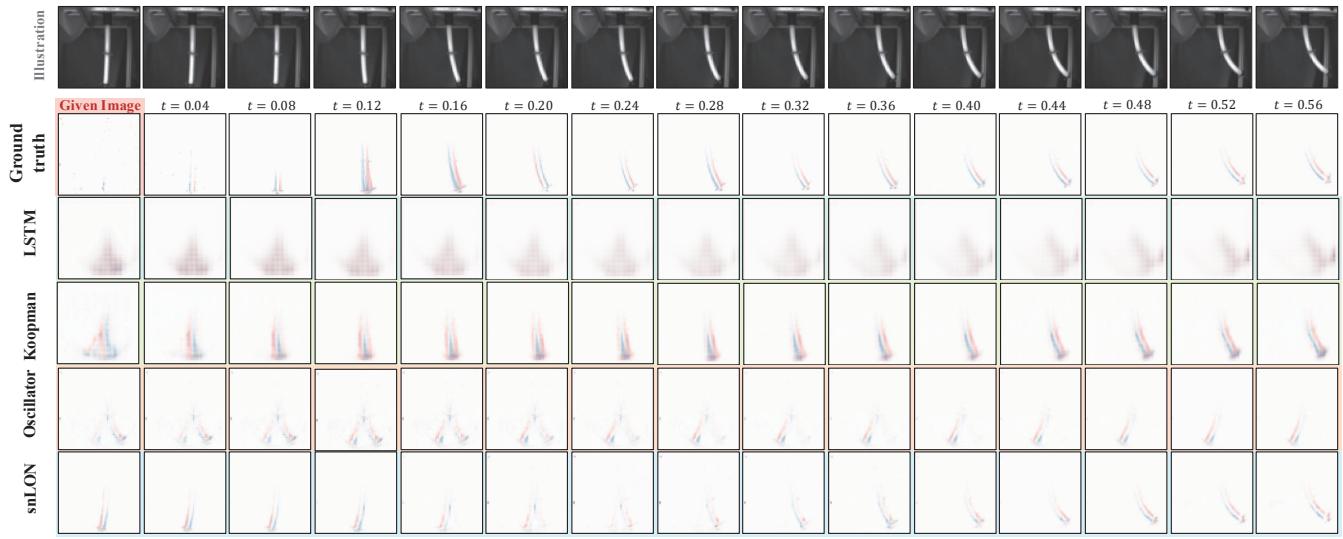


Fig. 8. Visualization comparison among four different latent dynamical networks on the CA scenario. The framework predicts over 14 time steps using only the initial image and the τ sequence as input. Similar to the DA scenario, in the CA test, we show the corresponding grayscale images in the first row to illustrate the movement of the soft robot. During the sequence, the soft robot moves from left to right. Below, we compare the results produced by the four models. The comparison shows that only snLON is able to maintain the correct motion trend of the soft robot while avoiding noticeable blurring in the reconstructed images.

although the performance gap is smaller due to sensor noise and unmodeled physical effects, the snLON model remains competitive and shows clear advantages in preserving spatial structure, achieving the highest SSIM and strong final-step performance in the CA scenario. These results indicate that introducing spiking dynamics into the Oscillator Network improves the robustness and temporal consistency of the latent state evolution, benefiting both simulation environments and real-world conditions.

To more intuitively demonstrate the result of snLON on sequence prediction, we provide Fig. 5 and Fig. 6 to show the visualization in all experiments of DA and CA scenarios. The visualization results show that the snLON model produces accurate multi-step predictions across all scenarios. The predicted event-based images closely match the ground truth, capturing both the overall shape and the temporal evolution of the soft robot's motion, even in more ambiguous real-world experimental scenarios of CA. The model maintains consistency over long time horizons, with minimal error accumulation, and performs similarly in both simulation and real-world experiments. These results visually confirm that the learned latent dynamics generalize well and reliably represent the robot's deformation over time.

We also provide the visualization comparison in the real experiments of both DA and CA scenarios between all four latent dynamical networks. Fig. 7 and Fig. 8 show that the snLON model produces predictions that remain visually close to the ground truth throughout the full 14-step rollout, accurately preserving the robot's shape and motion trajectory. In contrast, the Oscillator Network begins to drift over time, the Koopman loses fine structural details, and the LSTM quickly collapses to blurred or overly smooth shapes. These differences become more pronounced as the prediction horizon increases.

IV. Conclusion and Future Work

We proposed a latent dynamical network that combines a spiking neuron model with an Oscillator Network for event-based soft robot motion prediction. In this framework, the spiking neuron encodes the external control input, while the Oscillator Network represents the soft robot state. The interaction between these two components provides richer dynamics, allowing the model to maintain accurate predictions across different testing scenarios. We further compared the proposed model with several baselines, including an RNN-based network, a Koopman-based model, and a standard Linear Oscillator Network. The results demonstrate that the proposed snLON achieves superior performance, particularly in long-horizon prediction tasks. Future work will investigate closed-loop latent space control based on this framework.

References

[1] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 30–65, 2023.

[2] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1728–1748, 2023.

[3] E. Milana, C. D. Santina, B. Gorissen, and P. Rothemund, "Physical control: A new avenue to achieve intelligence in soft robotics," *Science Robotics*, vol. 10, no. 102, p. eadw7660, 2025.

[4] Z. Chen, F. Renda, A. Le Gall, L. Mocellin, M. Bernabei, T. Dangel, G. Ciuti, M. Cianchetti, and C. Stefanini, "Data-driven methods applied to soft robot modeling and control: A review," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 2241–2256, 2024.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.

[5] H. Wang, H. Ni, J. Wang, and W. Chen, "Hybrid vision/force control of soft robot based on a deformation model," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 661–671, 2019.

[6] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018.

[7] U. Yoo, H. Zhao, A. Altamirano, W. Yuan, and C. Feng, "Toward zero-shot sim-to-real transfer learning for pneumatic soft robot 3d proprioceptive sensing," *arXiv preprint arXiv:2303.04307*, 2023.

[8] S. L. Li, A. Zhang, B. Chen, H. Matusik, C. Liu, D. Rus, and V. Sitzmann, "Controlling diverse robots by inferring jacobian fields with deep networks," *Nature*, pp. 1–7, 2025.

[9] G. Fang, X. Wang, K. Wang, K.-H. Lee, J. D. Ho, H.-C. Fu, D. K. C. Fu, and K.-W. Kwok, "Vision-based online learning kinematic control for soft robots using local gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1194–1201, 2019.

[10] H. Wang, W. Chen, C. Wang, X. Wang, and R. Pfeifer, "Dynamic modeling and image-based adaptive visual servoing of cable-driven soft robotic manipulator," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 884–11 889, 2014.

[11] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 57–62.

[12] R. Valadas, M. Stölzle, J. Liu, and C. Della Santina, "Learning low-dimensional strain models of soft robots by looking at the evolution of their shape with application to model-based control," in *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2025, pp. 1–8.

[13] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on robot learning*. PMLR, 2023, pp. 2226–2240.

[14] N. Wahlström, T. B. Schön, and M. P. Deisenroth, "Learning deep dynamical models from image pixels," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1059–1064, 2015.

[15] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[16] S. L. Brunton and J. N. Kutz, "Machine learning for partial differential equations," *arXiv preprint arXiv:2303.17078*, 2023.

[17] T. Coleman, R. Babuška, J. Kober, and C. Della Santina, "An empirical investigation on variational autoencoder-based dynamic modeling of deformable objects from rgb data," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*. IEEE, 2024, pp. 921–928.

[18] G. Soter, A. Conn, H. Hauser, and J. Rossiter, "Bodily aware soft robots: integration of proprioceptive and exteroceptive sensors," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2448–2453.

[19] E. Almanzor, F. Ye, J. Shi, T. G. Thuruthel, H. A. Wurdemann, and F. Iida, "Static shape control of soft continuum robots using deep visual inverse kinematic models," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2973–2988, 2023.

[20] M. Stölzle and C. Della Santina, "Input-to-state stable coupled oscillator networks for closed-form model-based control in latent space," *Advances in Neural Information Processing Systems*, vol. 37, pp. 82 010–82 059, 2024.

[21] T. K. Rusch, B. Chamberlain, J. Rowbottom, S. Mishra, and M. Bronstein, "Graph-coupled oscillator networks," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 888–18 909.

[22] A. Ceni, A. Cossu, M. W. Stölzle, J. Liu, C. Della Santina, D. Bacciu, and C. Gallicchio, "Random oscillators network for time series processing," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 4807–4815.

[23] S. Lanthaler, T. K. Rusch, and S. Mishra, "Neural oscillators are universal," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[24] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using koopman operator theory," *IEEE transactions on robotics*, vol. 37, no. 3, pp. 948–961, 2020.

[25] D. Gehrig and D. Scaramuzza, "Low-latency automotive vision with event cameras," *Nature*, vol. 629, no. 8014, pp. 1034–1040, 2024.

[26] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[27] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, p. eaaz9712, 2020.

[28] J. Liu, E. Shahabishalghouni, and C. D. Santina, "Neural linear oscillator networks and their application to soft robots model learning and control," *IEEE Robotics and Automation Letters*, pp. 1–8, 2026.

[29] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.

[30] X. Chen, J. Wu, C. Ma, Y. Yan, Y. Wu, and K. C. Tan, "Pmsn: A parallel multi-compartment spiking neuron for multi-scale temporal processing," *arXiv preprint arXiv:2408.14917*, 2024.

[31] X. Zhang, F. K. Chan, T. Parthasarathy, and M. Gazzola, "Modeling and simulation of complex dynamic musculoskeletal architectures," *Nature communications*, vol. 10, no. 1, p. 4825, 2019.

[32] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e: From video frames to realistic dvs events," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1312–1321.

[33] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9000–9008.

[34] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 07,2026 at 09:42:17 UTC from IEEE Xplore. Restrictions apply.