Learning Soft Robot Dynamics using Differentiable Kalman Filters and Spatio-Temporal Embeddings

Xiao Liu[1], Shuhei Ikemoto[2], Yuhei Yoshimitsu[2], and Heni Ben Amor[1]

Abstract— **This paper introduces a novel approach for modeling the dynamics of soft robots, utilizing a differentiable filter architecture. The proposed approach enables end-to-end training to learn system dynamics, noise characteristics, and temporal behavior of the robot. A novel spatio-temporal embedding process is discussed to handle observations with varying sensor placements and sampling frequencies. The efficacy of this approach is demonstrated on a tensegrity robot arm by learning end-effector dynamics from demonstrations with complex bending motions. The model is proven to be robust against missing modalities, diverse sensor placement, and varying sampling rates. Additionally, the proposed framework is shown to identify physical interactions with humans during motion. The utilization of a differentiable filter presents a novel solution to the difficulties of modeling soft robot dynamics. Our approach shows substantial improvement in accuracy compared to state-of-the-art filtering methods, with at least a 24% reduction in mean absolute error (MAE) observed. Furthermore, the predicted end-effector positions show an average MAE of 25.77mm from the ground truth, highlighting the advantage of our approach. The code is available at `https://github.com/ir-lab/soft_robot_DEnKF`.**

I. Introduction

Soft robots are deformable structures that can be actuated and are composed of materials that form smooth curved shapes [1]. Robots of this type have the ability to perform a large range of movements, including extension, contraction, bending, shearing, and twisting, which makes them highly adaptable to confined spaces [2], [3] such as in medical [4] and industrial settings. Tensegrity structures, composed of compressive members that are supported by tensile cables [5], have been utilized in the design of soft robots. Such tensegrity robots have become popular in recent years since they bridge the gap between an inherently flexible system and the ability to use rigid components [6]–[8]. Their design allows for effective resistance against compressive forces in specific directions while also maintaining overall flexibility.

However, modeling the dynamics of soft robots with high-redundancy, flexibility, and a large number of degrees of freedom (DoF) is a daunting task due to their intricate, nonlinear design properties. Although numerical simulation tools [9] based on discrete differential geometry have provided insightful ways to analyze robot dynamics, real robots are still widely modeled using analytical approaches, such as the finite element method [10] and Cosserat rod theory [11]. However, these analytical methods also have limitations, such as the constraints imposed by limited computing resources and model generalizability. Different types of robots may require different mathematical modeling approaches. Moreover, a change to the properties of the robot, e.g., sensing or actuation, may require substantial human expertise and time investment to adjust the underlying model and its parameters. Soft robots also present modeling challenges due to uncertainties with regard to optimal number of sensors, sensor placement and temporal sampling frequency required to capture their dynamics accurately [12]. More generally, soft robots are a manifestation of complex and highly non-linear systems which are known to be hard to model.

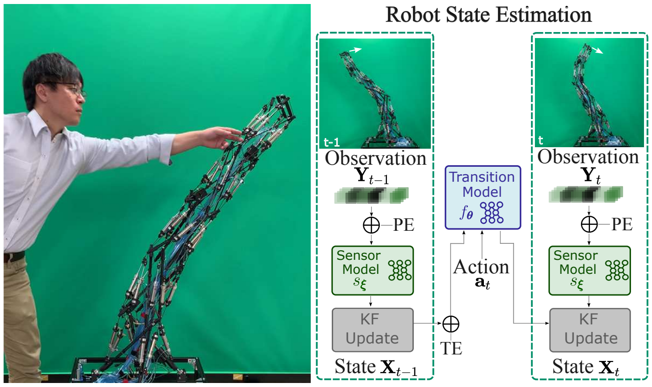

Fig. 1: **Left**: A human is shown interacting with a soft robot; **Right**: The differentiable Kalman filtering framework includes temporal and positional embedding modules (TE and PE) and a sensor model for projecting encoded raw observations to observation space. The filtering process involves utilizing the Kalman filter update step, where a stochastic transition model with an ensemble state is used to approximate the posterior distribution.

In this paper, we argue that recent innovations in modeling complex systems are particularly well-suited for learning and predicting the non-linear dynamics of soft robots. In particular, we focus on deep state-space models (DSSM) [13]. These models learn to estimate states and measurements from observed sequences in a data-driven fashion [13]–[15]. Accordingly, the use of DSSMs can help to overcome the aforementioned modeling challenges associated with soft robots. One set of algorithms based on DSSMs, namely Differentiable Filters (DFs), focuses on learning state transition and measurement models from data while preserving the mechanisms inherent to Bayesian recursive filtering. These properties render DFs ideal for systems with complex dynamics and sensor observations. In addition, DFs also

[1] X. Liu and H. Ben Amor are with the School of Computing and Augmented Intelligence, Arizona State University `{xliu330, hbenamor}@asu.edu`

[2] S. Ikemoto and Y. Yoshimitsu are with Department of Human Intelligence Systems, Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology `ikemoto@brain.kyutech.ac.jp` and `yoshimitsu.yuhei608@mail.kyutech.jp`

provide interpretable state representations and uncertainty estimation techniques, which is crucial for safety-critical systems [16].

Extending prior works on DSSMs, this paper introduces a novel differentiable filter called differentiable Ensemble Kalman Filters (DEnKF), for modeling soft robots. It offers an end-to-end learning approach to estimate the state of the system, which is highly nonlinear and difficult to model analytically. The main contributions are:

- The introduction of a positional embedding process that enables the spatial generalization of DEnKF by encoding observations with respect to their positions. As a result, learned models can account for changes to the location of sensors on the robot body.
- The use of a temporal embedding process that allows DEnKF to perform inference at variable rates and account for a multitude of time-delays due to sensing, hardware or communication.
- The modular structure of the framework separates the state dynamics from the end-to-end learning process, ensuring that the state remains dynamic even in the absence of observations.
- The paper also demonstrates a downstream application task of the framework for estimating human contact and physical interactions with the robot.

II. Related work

Soft robots are known to be difficult to model due to their highly nonlinear and often complex behavior [1]. To address this challenge, previous works have proposed various modeling approaches with different assumptions.

Soft robot modeling: A common approach for modeling soft robots is through steady-state models, which are equivalent to the kinematic model and assume that the system remains at rest in the absence of external forces. Steady-state models have been used in previous works, such as in [17], but they have limitations in terms of reachability, efficiency, and speed of the controller. Another approach for modeling soft robots involves using the finite element method (FEM) to develop geometrically exact models, as proposed in [10]. FEM-based models have been shown to accurately capture the behavior of soft robots. However, despite recent advancements in FEM algorithm optimizations, real-time applications necessitate further advancements in hardware technology. Data-driven methods, such as using multiple layer perceptrons to develop mappings from action space to state space, have also been proposed for soft robot modeling [17], [18], but these methods have limitations in terms of generalizability. Dynamic models, which account for the time-varying behavior of the system, have been shown to provide advantages for effective motion planning. For example, in [19], a model-based dynamic controller is proposed under the piecewise constant curvature assumption, and in [20], modeling based on a first-order dynamical system is used. However, due to the specificity of each dynamic model developed for different soft robot physics, it is challenging to achieve generalization. Recently, a state estimation approach based on a sparse Gaussian process regression and Cosserat rod theory is proposed in [21], providing a general way of continuum robot modeling. However, this approach also has its own limitations, such as the need for high computational resources and the difficulty of incorporating prior knowledge of physical constraints.

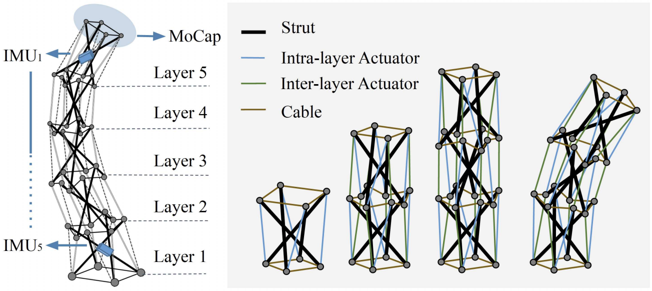

Fig. 2: The tensegrity robot: the robot contains 5 layers where each layer is a flexible tensegrity module with struts, stiff cables, and actuators. The sensory data associated with the robot are the IMUs, MoCap, and the pressure vector readings from the pneumatic cylinders actuators.

In general, each modeling approach is associated with certain advantages and limitations, and the selection of a particular approach hinges on diverse factors, including the particular application's demands, available resources, and the feasibility of sensor deployment.

Differentiable filters: Differentiable Filters (DFs) are composed of neural networks with algorithmic priors of Bayesian filter techniques to provide learning-based approaches for the forward and measurement models in recursive filtering. BackpropKF [22] trains Kalman Filters using backpropagation with the integration of feed-forward networks and convolutional neural networks. Similarly, differentiable algorithm networks [23] introduce neural network components that encode differentiable robotic algorithms, akin to Differentiable Particle Filters (DPFs) [24], [25]. DPFs employ algorithmic priors to increase learning efficiency and variations have been explored using adversarial methods for posterior estimation [26]. DFs were analyzed in [15] for training and modeling uncertainty with noise profiles. The authors implemented the DFs as multi-layer perceptrons enveloped in an RNN layer, and tested them on real-world tasks in [15], [16]. The results showed that end-to-end learning is crucial for accurately learning noise models.

III. Soft robot Modeling with DEnKF

This section provides a detailed examination of the tensegrity robot structure, the bending motion mechanism, and the relevant sensory information. It then introduces the differentiable ensemble Kalman Filters (DEnKF) and its learning process as a method for propagating the robot state forward in time and correcting it using sensory data. To improve the framework's robustness and generalization across diverse sensor placements and inference rates, two enhancements – Temporal and Positional embedding (TE and PE) – are detailed to show how encoded state and observation features can be learned.

A. Preliminaries

The soft robot system employed in our work is a tensegrity robot arm developed in [8]. This robot is built using a strict tensegrity structure, which consists of elements like struts, cables, spring-loaded cables, and actuated cables. The arm has five layers, as depicted in Fig. 2. Each of these modules can be viewed as a layer of an arm-like tensegrity structure. By applying an external force, various continuous bending postures can be produced. As shown in Fig. 2, the stiff cables maintain the longitudinal length while external forces contribute solely to changes in the bending direction. The sensory data associated with the robot, which is used to describe its kinematics, includes Inertial Measurement Unit (IMU) data, optical motion capture (MoCap) data, and proportional pressure control valves data. There are five IMUs mounted at each strut of the robot's layers, with each layer having four different struts for IMU placement. Thus, there are 20 different locations for IMU placement.

The state of a soft robot at time t is represented by a 7-dimensional vector $\mathbf{x}_t = [x, y, z, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_w]^T$, which denotes the position and orientation of the robot's hand tip (end-effector). The quaternion vector $\mathbf{q}$ represents the posture of the robot relative to the base frame (layer 1's bottom). The action $\mathbf{a}_t$ of the system is the pressure vector of the 40 pneumatic cylinder actuators, where $\mathbf{a}_t \in \mathbb{R}^{40}$. The raw observation $\mathbf{y}_t$ consists of 5 IMU readings, where $\mathbf{y}_t \in \mathbb{R}^{30}$, with each IMU providing a 6-dimensional vector of accelerations and angular velocities relative to its location. An example of the IMU placements is shown in Fig. 3, where the blue cubes represent the location of the IMUs on each strut at each layer. Each integer number $\mathcal{Z}_i$ denotes a location label, and a vector of integers $\mathcal{Z}$ is recorded for five IMU locations, i.e., $\mathcal{Z} = [1, 5, 9, 14, 18]$, with $1 \leq \mathcal{Z}_i \leq 20$.

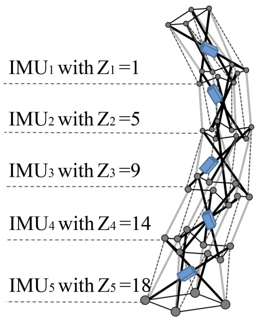

Fig. 3: The placement of the IMUs denoted by $\mathcal{Z}$.

B. Bayesian Filtering

Recursive Bayesian filtering is a technique used to estimate the state $\mathbf{x}_t$ of a discrete-time dynamical system, given a sequence of actions $\mathbf{a}_{1:t}$ and noisy observations $\mathbf{y}_{1:t}$. The posterior distribution of the state can be represented by the following equation:

$$\begin{aligned} & p(\mathbf{x}_t|\mathbf{a}_{1:t}, \mathbf{y}_{1:t}, \mathbf{x}_{1:t-1}) \\ & \propto p(\mathbf{y}_t|\mathbf{a}_t, \mathbf{x}_t)\ p(\mathbf{x}_t|\mathbf{a}_{1:t-1}, \mathbf{y}_{1:t-1}, \mathbf{x}_{1:t-1}). \end{aligned} \tag{1}$$

We can denote the belief of the state as $\text{bel}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{a}_{1:t}, \mathbf{y}_{1:t}, \mathbf{x}_{1:t-1})$. Assuming the Markov property, where the next state is dependent only on the current state, we get the following expression:

$$\text{bel}(\mathbf{x}_t) = \underbrace{p(\mathbf{y}_t|\mathbf{x}_t)}_{\text{observation model}} \prod_{t=1}^{t} \overbrace{p(\mathbf{x}_t|\mathbf{a}_t, \mathbf{x}_{t-1})}^{\text{state transition model}} \text{bel}(\mathbf{x}_{t-1}), \tag{2}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the observation model and $p(\mathbf{x}_t|\mathbf{a}_t, \mathbf{x}_{t-1})$ is the transition model. The transition model describes the laws that govern the evolution of the system state, while the observation model identifies the relationship between the hidden, internal state of the system and observed, noisy measurements.

In our work, we introduce a method for state estimation called differentiable ensemble Kalman Filters (DEnKF). Our approach combines the traditional EnKF algorithm with recent advancements in stochastic neural networks (SNNs) [27] who established a connection between Dropout training and Bayesian inference in deep Gaussian processes. The EnKF algorithm involves updating the approximate posterior distribution by propagating each ensemble member forward in time. In DEnKF approach, we maintain the core algorithmic steps of EnKF while leveraging the capabilities of SNNs. To represent the initial state distribution, we use an ensemble of E members, denoted by $\mathbf{X}_0 = [\mathbf{x}_0^1, \ldots, \mathbf{x}_0^E]$, where $E \in \mathbb{Z}^+$. The key difference between our approach and traditional EnKF is the implicit modeling of process noise through sampling from a stochastic neural network.

Prediction Step: We leverage the stochastic forward passes from a trained state transition model to update each ensemble member:

$$\mathbf{x}_{t|t-1}^i \sim f_{\boldsymbol{\theta}}(\mathbf{x}_{t|t-1}^i|\mathbf{a}_t, \mathbf{x}_{t-1|t-1}^i),\ \forall i \in E. \tag{3}$$

Matrix $\mathbf{X}_{t|t-1} = [\mathbf{x}_{t|t-1}^1, \cdots, \mathbf{x}_{t|t-1}^E]$ holds the updated ensemble members which are propagated one step forward through the state space. Note that sampling from the transition model $f_{\boldsymbol{\theta}}(\cdot)$ (using the SNN methodology described above) implicitly introduces a process noise.

Update Step: Given the updated ensemble members $\mathbf{X}_{t|t-1}$, a nonlinear observation model $h_{\boldsymbol{\psi}}(\cdot)$ is applied to transform the ensemble members from the state space to observation space. Following our main rationale, the observation model is realized via a neural network with weights $\boldsymbol{\psi}$. Accordingly, the update equations for the EnKF become:

$$\mathbf{H}_t\mathbf{X}_{t|t-1} = \left[h_{\boldsymbol{\psi}}(\mathbf{x}_{t|t-1}^1), \cdots, h_{\boldsymbol{\psi}}(\mathbf{x}_{t|t-1}^E)\right], \tag{4}$$

$$\begin{aligned} \mathbf{H}_t\mathbf{A}_t &= \mathbf{H}_t\mathbf{X}_{t|t-1} \\ &- \left[\frac{1}{E}\sum_{i=1}^{E} h_{\boldsymbol{\psi}}(\mathbf{x}_{t|t-1}^i), \cdots, \frac{1}{E}\sum_{i=1}^{E} h_{\boldsymbol{\psi}}(\mathbf{x}_{t|t-1}^i)\right]. \end{aligned} \tag{5}$$

$\mathbf{H}_t\mathbf{X}_{t|t-1}$ is the predicted observation, and $\mathbf{H}_t\mathbf{A}_t$ is the sample mean of the predicted observation at t. EnKF treats observations as random variables. Hence, the ensemble can incorporate a measurement perturbed by a small stochastic noise thereby accurately reflecting the error covariance of the best state estimate [28]. In our differentiable version of the EnKF, we also incorporate a sensor model which can learn

TABLE I: Differentiable filters' learnable sub-modules.

$f_{\boldsymbol{\theta}}$:	2×SNN(64, ReLu), 2×SNN(128, ReLu), 1×SNN(S, -)
$h_{\boldsymbol{\psi}}$:	2×fc(32, Relu), 2×fc(64, ReLu), 1× fc(O, -)
$r_{\boldsymbol{\zeta}}$:	2×fc(16, ReLu), 1×fc(O, -)
$s_{\boldsymbol{\xi}}$:	fc(128, ReLu), flatten(), 2×SNN(512, ReLu), 1×SNN(256, ReLu), 1×SNN(128, ReLu), 1×SNN(O, -)

fc: fully connected, conv: convolution, S, O: state and observation dimension.

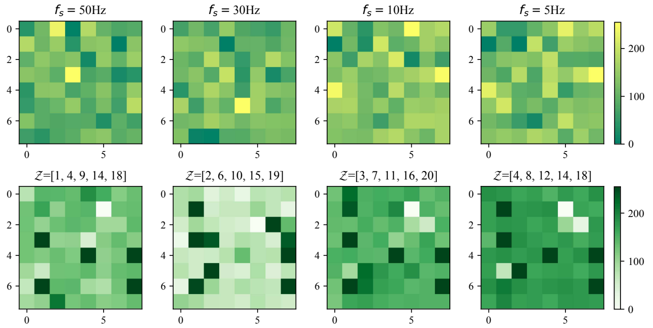

Fig. 4: The top and bottom figures show the output latent feature maps from the temporal embedding process and the positional embedding process, respectively, for various sampling frequencies f_s and IMU positions $\mathcal{Z}$.

projections between a latent space and observation space. We train a stochastic sensor model $s_{\boldsymbol{\xi}}(\cdot)$:

$$\tilde{\mathbf{y}}_t^i \sim s_{\boldsymbol{\xi}}(\tilde{\mathbf{y}}_t^i|\mathbf{y}_t),\ \forall i \in E, \tag{6}$$

where $\mathbf{y}_t$ represents the raw observation. Sampling yields learned observations $\tilde{\mathbf{Y}}_t = [\tilde{\mathbf{y}}_t^1, \cdots, \tilde{\mathbf{y}}_t^E]$ and the sample mean $\tilde{\mathbf{y}}_t = \frac{1}{E}\sum_{i=1}^{i} \tilde{\mathbf{y}}_t^i$. The innovation covariance $\mathbf{S}_t$ can then be calculated as:

$$\mathbf{S}_t = \frac{1}{E-1}(\mathbf{H}_t\mathbf{A}_t)(\mathbf{H}_t\mathbf{A}_t)^T + r_{\boldsymbol{\zeta}}(\tilde{\mathbf{y}}_t), \tag{7}$$

where $r_{\boldsymbol{\zeta}}(\cdot)$ is the measurement noise model implemented using multi-layer perceptron (MLP), it takes a learned observation $\tilde{\mathbf{y}}_t$ at time t and provides a stochastic noise in the observation space by constructing the diagonal of the noise covariance matrix. The final estimate of the ensemble $\mathbf{X}_{t|t}$ can be obtained by performing the KF update step:

$$\mathbf{A}_t = \mathbf{X}_{t|t-1} - \frac{1}{E}\sum_{i=1}^{E} \mathbf{x}_{t|t-1}^i, \tag{8}$$

$$\mathbf{K}_t = \frac{1}{E-1}\mathbf{A}_t(\mathbf{H}_t\mathbf{A}_t)^T\mathbf{S}_t^{-1}, \tag{9}$$

$$\mathbf{X}_{t|t} = \mathbf{X}_{t|t-1} + \mathbf{K}_t(\tilde{\mathbf{Y}}_t - \mathbf{H}_t\mathbf{X}_{t|t-1}), \tag{10}$$

where $\mathbf{K}_t$ is the Kalman gain. In inference, the ensemble mean $\bar{\mathbf{x}}_{t|t} = \frac{1}{E}\sum_{i=1}^{E} \mathbf{x}_{t|t}^i$ is used as the updated state. The neural network structures for sub-modules are described in Table I.

C. Spatio-Temporal Embedding

The aim of the temporal and positional embedding is to increase the generalizability of the robot state estimator under varied inference frequencies and sensor positions. Taking inspiration from language models, which encode different positional information of words in a sentence [29], we incorporate temporal and positional labels in a similar fashion. We opt to use sinusoidal functions implemented in [29] with varying frequencies using the temporal and positional embedding functions:

$$\mathrm{PE}_{(pos,i)} = \begin{cases} \sin(\frac{pos}{10000^{2i/d_m}}) & \text{if } i \bmod 2 = 0 \\ \cos(\frac{pos}{10000^{2i/d_m}}) & \text{otherwise} \end{cases} \tag{11}$$

where pos is the position and i is the dimension, d_m is the dimension of the feature space. The embedding function is first passed the different sensor placements denoted as $\mathcal{Z}$, and then the sensor model takes the embedded input features from the observation:

$$\tilde{\mathbf{y}}_t^i \sim s_{\boldsymbol{\xi}}(\tilde{\mathbf{y}}_t^i|\mathbf{y}_t, \boxed{\mathcal{Z}}),\ \forall i \in E. \tag{12}$$

The latent feature maps displayed in Fig. 4 exhibit varied encodings of $\mathcal{Z}$, which enables $s_{\boldsymbol{\xi}}$ to learn from the distinctive features presented in each encoding. The temporal embedding process is used when the system progresses at different speeds. In our system, we offer four options for the sampling frequency f_s – 5Hz, 10Hz, 30Hz, and 50Hz. The temporal embedding process is integrated within the latent space of the state transition model as follows:

$$\mathbf{x}_{t|t-1}^i \sim f_{\boldsymbol{\theta}}(\mathbf{x}_{t|t-1}^i|\mathbf{a}_t, \mathbf{x}_{t-1|t-1}^i, \boxed{f_s}),\ \forall i \in E. \tag{13}$$

Here, the input state $\mathbf{X}_{t-1}$ along with its corresponding f_s are encoded by $f_{\boldsymbol{\theta}}(\cdot)$ into a 64-dimensional latent vector, which is then used to apply the state transition. The encoded latent vector from the same initial state $\mathbf{X}_0$ is shown in Fig. 4 for different values of f_s. By doing so, $f_{\boldsymbol{\theta}}(\cdot)$ is able to learn the transitions from the unique state features.

IV. Experiment

This section presents a series of experiments that are conducted to evaluate the performance of the proposed differentiable ensemble Kalman filters (DEnKF) for state tracking. A comparison with baseline differentiable filters [15], [24] is also discussed. Furthermore, two downstream tasks of state estimation are performed, namely estimation with missing observations and virtual force estimation.

A. Experimental Setup

Data: The dataset is obtained by performing optical motion capture on the real tensegrity robot hand tip while supplying randomly generated desired pressure vectors to the pneumatic cylinder actuators. As mentioned in Sec. III-A, we record 40-dimensional pressure vectors as the action $\mathbf{a}_t \in \mathbb{R}^{40}$, 5 IMU readings $\mathbf{y}_t \in \mathbb{R}^{30}$ with the corresponding position $\mathcal{Z}$, and a 7-dimensional state $\mathbf{x}_t$. We collect 10 time-series data $D_1 - D_{10}$ (shown as Table III), with each continuous sampling lasting for one hour. There are 12,000 trials collected in total. During each trial, the robot is moved from the current equilibrium posture to the next equilibrium posture by applying the new desired pressure. We down-sample each time-series data to obtain the dataset with different sampling frequencies f_s. All data is gathered through the a ROS2 network, and their synchronization is achieved using the "message_filters" package.

TABLE II: Ablation study of proposed DEnKF with and without positional and temporal embedding processes for state estimation task in different conditions. The MAE error metric of the 10-fold cross-validation is reported.

Method	Fixed $\mathcal{Z}$		Multiple $\mathcal{Z}$		Multiple $\mathcal{Z}$ and f_s	
	EE (mm)	$\mathbf{q}$	EE (mm)	$\mathbf{q}$	EE (mm)	$\mathbf{q}$
DEnKF-Fix	**25.7765±7.827**	0.0648±0.035	51.7728±7.489	0.1996±0.034	106.2310±17.760	0.1722±0.065
DEnKF-PE	29.6578±8.873	**0.0626±0.028**	22.5427±9.146	0.0768±0.033	71.5244±9.411	0.1844±0.022
DEnKF-PE+TE	31.5985±9.582	0.0788±0.021	**21.7658±6.337**	**0.0641±0.026**	**25.7566±4.835**	**0.0466±0.055**

Means±standard errors.

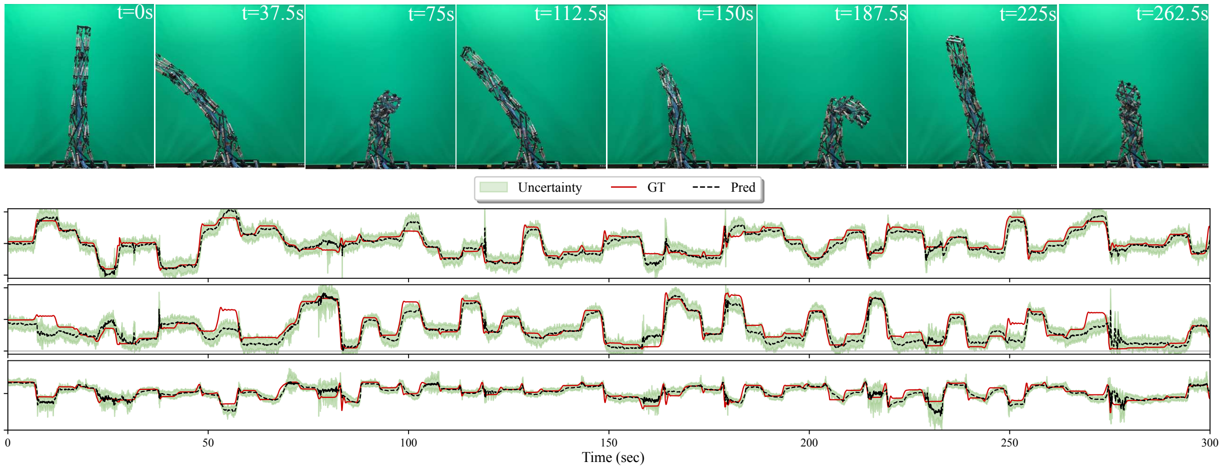

Fig. 5: The real-time estimation of the state on the tensegrity robot arm is demonstrated in this study. Specifically, the **top** figure presents a motion sequence of the robot without applying external forces, while the **bottom** figure showcases the real-time tracking outcomes (along with the corresponding uncertainty) of the positions of the hand tip.

TABLE III: Time-series data with different sensor placements.

D_1 : [1,4,9,14,18],	D_2 : [1,5,9,15,19],	D_3 : [2,6,10,15,19]
D_4 : [2,6,10,16,20],	D_5 : [2,6,10,13,17],	D_5 : [3,7,11,13,17]
D_6 : [3,7,11,14,18],	D_7 : [3,7,11,16,20],	D_8 : [4,8,12,16,20]
D_9 : [4,8,12,14,18],	D_{10} : [4,8,12,15,19]	

Training: Proposed framework contains four sub-modules as listed in Table I. The entire framework is trained in an end-to-end manner via a mean squared error (MSE) loss between the ground truth state $\hat{\mathbf{x}}_{t|t}$ and the estimated state $\bar{\mathbf{x}}_{t|t}$ at every timestep. We also supervise the intermediate modules via loss gradients $\mathcal{L}_{f_{\boldsymbol{\theta}}}$ and $\mathcal{L}_{s_{\boldsymbol{\xi}}}$. Given ground truth at time t, we apply the MSE loss gradient calculated between $\hat{\mathbf{x}}_{t|t}$ and the output of the state transition model to $f_{\boldsymbol{\theta}}$ as in Eq. 14. We apply the intermediate loss gradients computed based on the ground truth observation $\hat{\mathbf{y}}_t$ and the output of the stochastic sensor model $\tilde{\mathbf{y}}_t$:

$$\mathcal{L}_{f_{\boldsymbol{\theta}}} = \|\bar{\mathbf{x}}_{t|t-1} - \hat{\mathbf{x}}_{t|t}\|_2^2, \quad \mathcal{L}_{s_{\boldsymbol{\xi}}} = \|\tilde{\mathbf{y}}_t - \hat{\mathbf{y}}_t\|_2^2. \tag{14}$$

All models in the experiments were trained for 50 epochs with batch size 64, and a learning rate of $\eta = 10^{-5}$. The ensemble size of the Kalman filter was set to 32 ensemble members.

B. State Estimation

In this experiment, we investigate the effectiveness of proposed differentiable filters in estimating the state of the tensegrity robot, and perform comparisons against other differentiable filters baselines. To reiterate, the robot state is defined as $\mathbf{x}_t = [x, y, z, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_w]^T$, the learned observation $\tilde{\mathbf{y}}_t$ is defined to have the same dimension as the robot state, where $\tilde{\mathbf{y}}_t = [x, y, z, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_w]^T$. The state estimator tracks the robot end-effector (EE) in position and orientation while random generated pressure vectors are supplied for the pneumatic cylinder actuators.

Results: The proposed DEnKF leverages both the positional embedding (PE) and temporal embedding (TE) processes. We conducted a comprehensive evaluation of the performance of DEnKF under different $\mathcal{Z}$ conditions and with varying f_s. The experiment involved training and validating three models: a) DEnKF-Fix trained on a single time-series data, b) DEnKF-PE trained with multiple time-series data collected from diverse $\mathcal{Z}$ conditions, and c) DEnKF-PE+TE trained with multiple time-series data from various $\mathcal{Z}$ conditions and f_s values. Table II presents the ablation study results for the three models evaluated with 10-fold cross-validation and the mean absolute error (MAE) metric. In the ablation study, three different conditions are considered – the fixed IMU positions, multiple IMU positions, and multiple IMU positions with various sampling frequencies. The DEnKF-Fix model, trained with a single time-series data and a fixed IMU placement, shows the best performance with an average 25.77mm offset from the ground truth EE positions. However, its performance is limited to fixed IMU placement only. The DEnKF-PE+TE model demonstrates robust performance across different $\mathcal{Z}$ conditions and improves the accuracy of the EE positions and orientations by 58% and 66% respectively, compared to the DEnKF-Fix model.

While DEnKF-PE achieves comparable MAE to DEnKF-PE+TE under multiple $\mathcal{Z}$ conditions, the performance of DEnKF-PE+TE outperforms DEnKF-PE by 64% and 75% on position and orientation respectively, when multiple f_s values are provided.

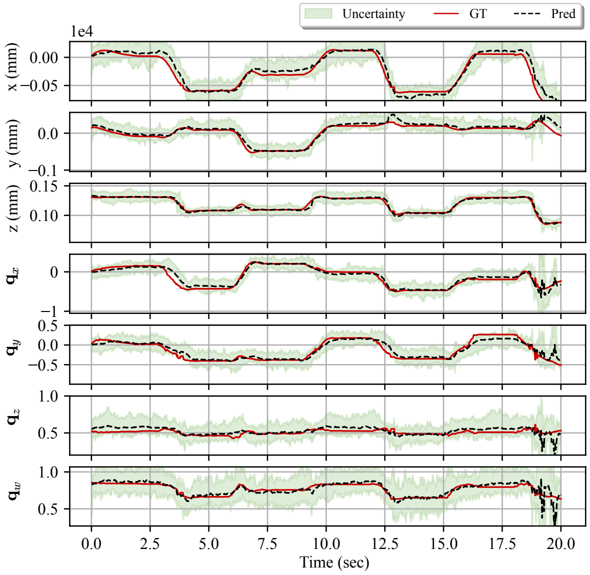

Fig. 6: State estimation results on the EE positions (x, y, z) and orientations presented as $(\mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_w)$.

To perform a comprehensive evaluation of the DEnKF framework, a long-term state estimation experiment lasting for five minutes was carried out on the physical tensegrity robot, as depicted in Fig. 5. The video sequence captures the robot in different configurations at various points in time, while the estimated positions of the EE demonstrate robust and stable state tracking throughout the entire duration of the experiment. Moreover, the complete 7-dimensional state vector, which includes both position and orientation, is shown in Fig. 6. While the orientation vector exhibits a relatively larger uncertainty, the framework is able to accurately track the overall state over time. Further analysis of the state tracking is conducted by visualizing the EE trajectories in 3D for the DEnKF-PE+TE model, as shown in Fig. 7. Each trajectory represents one test trial, and the prediction results of the DEnKF model are shown with the ensemble state outputs representing the uncertainty. The trajectories show the robustness and adaptability of DEnKF across a range of conditions, including multiple distinct $\mathcal{Z}$ and f_s.

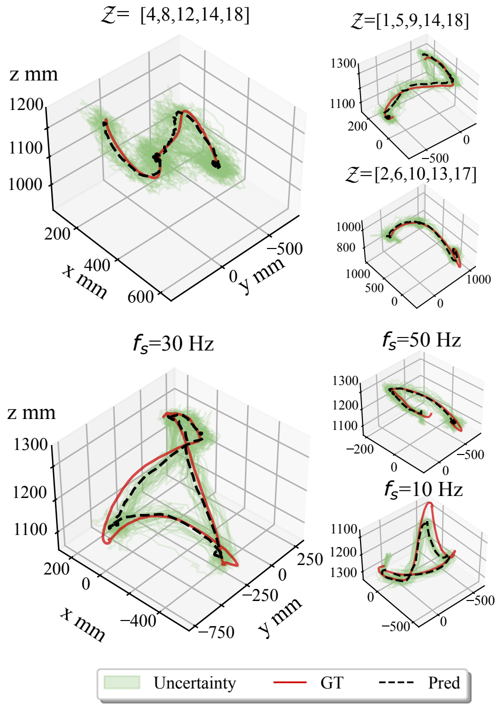

Fig. 7: Testing results on state estimation with various IMU positions $\mathcal{Z}$ and sampling frequencies f_s.

TABLE IV: Comparison with other baselines on state estimation task measured in RMSE and MAE of the EE position with fixed IMU locations $\mathcal{Z}$. Results for dEKF, dPF, and dPF-M-lrn are reproduced for detailed comparisons.

Method	Fixed IMU $\mathcal{Z}$ RMSE	MAE	Wall clock time (s)
dEKF [15]	61.753±1.630	41.960±1.147	**0.047**
DPF [24]	51.184±7.204	34.317±4.205	0.060
dPF-M-lrn [15]	49.799±8.264	33.903±6.964	0.059
DEnKF-Fix (ours)	**31.519**±**9.974**	**25.777**±**7.827**	0.062

Means±standard errors.

Comparison: Table IV presents the performance comparison of our proposed differentiable Kalman filter with fixed sensor model (DEnKF-Fix) against state-of-the-art differentiable filters for state estimation of soft robots, namely differentiable Extended Kalman Filters (dEKF) [15], differentiable Particle Filters (DPF) [24], and the modified differentiable Particle Filter with learned process noise model (dPF-M-lrn) [15]. To ensure a fair comparison, we removed the temporal and positional embedding processes (TE and PE) for the DEnKF and supplied the same sensor model $s_{\boldsymbol{\xi}}$ for all methods. For the DPF and dPF-M-lrn methods, we trained and tested with 100 particles. Our results indicate that the DEnKF-Fix approach outperforms the state-of-the-art methods with a Mean Absolute Error (MAE) of 25.78mm. Specifically, our approach achieved an MAE that is 24%, 25%, and 39% lower than that of dEKF, DPF, and dPF-M-lrn, respectively. Among the baselines, dPF-M-lrn shows slightly better results than others, it does not exhibit any advantages over DEnKF-Fix. Our findings highlight the effectiveness of the DEnKF for state estimation of soft robots, particularly in comparison to existing differentiable filter approaches.

C. State Estimation with Missing Observation

In the field of soft robotics, sensor failures are common and can be mitigated by using learning-based sensing techniques [30]. The modular structure of the proposed framework offers an additional advantage by enabling compensation for such issues. In this experiment, we investigate the robustness of DEnKF in the event of sensor failures. Specifically, we use the forward model $f_{\boldsymbol{\theta}}$ to update the

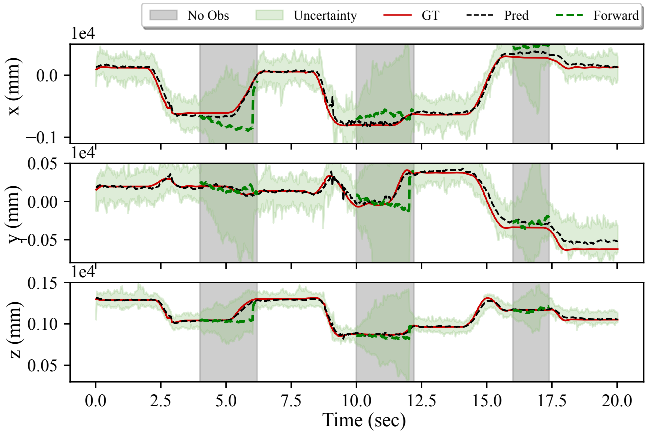

Fig. 8: State estimation results with missing observations (grey) using DEnKF; Only the forward model $f_{\boldsymbol{\theta}}$ is used when no observation is obtained.

robot state in the absence of observations. The experiment is conducted on a trained DEnKF-PE+TE model, which has not been exposed to such scenarios during training.

Results: In the experiment, we enable random 12.5% and 6.26% time windows with no observations for each testing trial (with 20 seconds). Figure 8 demonstrates one of the results when the state estimator is in the scenario where no observations are available. The prediction results show the state tracking outcomes with observation, while the green line represents the case where the forward model $f_{\boldsymbol{\theta}}$ alone is used. The uncertainty, which is described by the distribution of the ensemble members, increases when no observations are obtained and decreases when observations are obtained again.

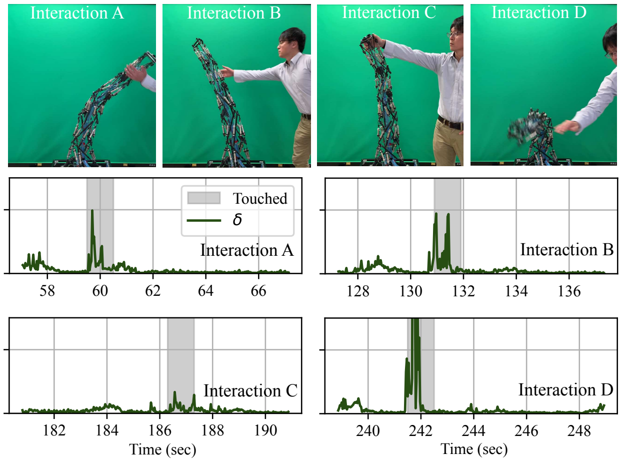

Fig. 9: Experiment on estimating virtual forces: in a testing sequence of 5 minutes, the robot is subjected to different forces labeled as A, B, C, and D, the virtual forces δ is captured during each interaction.

D. Virtual Force Estimation

An important downstream application of DEnKF is virtual force estimation. During training, the forward model $f_{\boldsymbol{\theta}}$ learns the system dynamics with given actions and previous states in the prediction step of DEnKF. However, in the correction step, which involves the sensor model $s_{\boldsymbol{\xi}}$ and Kalman update, only a time-invariant mapping between observation space and state space is conducted, without considering the state dynamics. In other words, the correction step estimates the state based on the posture configurations of the robot. Therefore, the correction step always generates a “corrected" state $\bar{\mathbf{x}}_{t|t}$ at time t regardless of whether a force is applied to the robot or not, but the prediction step outputs an estimated state $\hat{\mathbf{x}}_{t|t-1}$ assuming no force is applied. The virtual force defined as $\delta = \|\bar{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}\|_p$, it is estimated between these two different outputs using the Minkowski distance:

$$\|\bar{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}\|_p = \left(\sum |\bar{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}|^p\right)^{\frac{1}{p}} \tag{15}$$

with $p = 10$.

Results: In this experiment, we apply random pressure vectors to the pneumatic cylinder actuators and conduct several interactions with the robot by applying external forces of different directions and magnitudes. The forces are applied four times, labeled A, B, C, and D, during a test sequence of 5 minutes as shown in Fig. 9. We capture the virtual force δ value during testing, also shown in Fig. 9. The gray areas indicate the time window of interaction. It is apparent that δ reflects the forces during each interaction. Interestingly, the magnitude of the Minkowski distance δ is proportional to the actual forces applied to the robot. For instance, in interaction D, where the actual forces are significant enough to entirely change the posture configurations of the robot (snapshot D in Fig. 9), the δ value reaches a higher value (>400) compared to the other interactions.

V. Conclusions

This paper presents a comprehensive study on the modeling of soft robot dynamics using differentiable ensemble Kalman filters (DEnKF). To enhance the spatio-temporal generalizability of state estimation, the proposed approach integrates temporal and positional embedding (TE and PE) processes. Through experiments on a highly nonlinear system, specifically the tensegrity robot arm, the proposed DEnKF demonstrates stable and accurate state tracking. In comparison to other differentiable filter frameworks, the proposed DEnKF outperforms the baselines and is capable of handling other downstream tasks, such as missing observation and virtual force estimation. Notably, the proposed DEnKF approach allows for fine-grained analysis of the state forward model and sensor model, which is not supported by other baselines. Because those baselines are RNN-based filters and the forward model has to remain in the same hidden state until an observation is processed.

With respect to limitations, this study focuses exclusively on the application of IMU sensors, while other types of soft robot sensors, such as fiber-based deformation sensors [31] and tendon/backbone strain sensors [32], have not been tested. It is important to acknowledge that each sensor has unique characteristics, and a fusion module may be able to leverage the advantages of multiple sensors based on the specific application requirements. Regarding force estimation, while the Minkowski distance has demonstrated the ability to reflect actual external forces, it is important to note that actual forces can also be learned and calibrated using supervised learning techniques. Additionally, other distance metrics can be explored to find the best fit for a particular application. Therefore, further research is needed to investigate the potential benefits and limitations of different types of soft robot sensors and distance metrics for force estimation.

In future research, the authors aim to extend the functionality of the DEnKF framework by exploring the calibration of virtual forces to actual values for improved force estimation. Additionally, the authors plan to investigate alternative state representations, such as segment-wise estimation of the soft robot, which may facilitate a more detailed understanding of the deformation characteristics of the robot and enhance estimation accuracy. These proposed efforts are expected to enhance the versatility and efficacy of the DEnKF framework and further advance its applicability in the domain of soft robotics.

Acknowledgement

This research was funded by the JSPS Grants-in-Aid for Scientific Research (KAKENHI) program under grant numbers 19K0285, 19H01122, and 21H03524. This research was also funded partially by "The Global KAITEKI Center" (TGKC) of the Global Futures Laboratory at Arizona State University.

References

[1] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, and S. Kim, "Soft robot review," *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 3–15, 2017.

[2] K. Lee, Y. Wang, and C. Zheng, "Twister hand: Underactuated robotic gripper inspired by origami twisted tower," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 488–500, 2020.

[3] E. Q. Yumbla, Z. Qiao, W. Tao, and W. Zhang, "Human assistance and augmentation with wearable soft robotics: a literature review and perspectives," *Current Robotics Reports*, pp. 1–15, 2021.

[4] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[5] R. E. Skelton and M. C. Oliveira, *Tensegrity Systems*. Springer Nature, 2009.

[6] E. Jung, V. Ly, N. Cessna, M. L. Ngo, D. Castro, V. SunSpiral, and M. Teodorescu, "Bio-inspired tensegrity flexural joints," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5561–5566.

[7] K. Kim, A. K. Agogino, and A. M. Agogino, "Rolling locomotion of cable-driven soft spherical tensegrity robots," *Soft robotics*, vol. 7, no. 3, pp. 346–361, 2020.

[8] S. Ikemoto, K. Tsukamoto, and Y. Yoshimitsu, "Development of a modular tensegrity robot arm capable of continuous bending," *Frontiers in Robotics and AI*, vol. 8, 2021.

[9] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, "Dynamic simulation of articulated soft robots," *Nature communications*, vol. 11, no. 1, p. 2233, 2020.

[10] Z. Gong, J. Cheng, K. Hu, T. Wang, and L. Wen, "An inverse kinematics method of a soft robotic arm with three-dimensional locomotion for underwater manipulation," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 516–521.

[11] F. Renda, F. Giorgio-Serchi, F. Boyer, C. Laschi, J. Dias, and L. Seneviratne, "A unified multi-soft-body dynamic model for underwater soft robots," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 648–666, 2018.

[12] A. W. Mahoney, T. L. Bruns, P. J. Swaney, and R. J. Webster, "On the inseparable nature of sensor selection, sensor placement, and state estimation for continuum robots or "where to put your sensors and how to use them"," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4472–4478.

[13] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[14] A. Klushyn, R. Kurle, M. Soelch, B. Cseke, and P. van der Smagt, "Latent matters: Learning deep state-space models," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[15] A. Kloss, G. Martius, and J. Bohg, "How to train your differentiable filter," *Autonomous Robots*, pp. 1–18, 2021.

[16] M. A. Lee, B. Yi, R. Martín-Martín, S. Savarese, and J. Bohg, "Multimodal sensor fusion with differentiable filters," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 444–10 451.

[17] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017.

[18] H. Jiang, Z. Wang, X. Liu, X. Chen, Y. Jin, X. You, and X. Chen, "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6127–6133.

[19] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.

[20] T. George Thuruthel, F. Renda, and F. Iida, "First-order dynamic modeling and control of soft robots," *Frontiers in Robotics and AI*, vol. 7, p. 95, 2020.

[21] S. Lilge, T. D. Barfoot, and J. Burgner-Kahrs, "Continuum robot state estimation using gaussian process regression on se (3)," *The International Journal of Robotics Research*, vol. 41, no. 13-14, pp. 1099–1120, 2022.

[22] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop kf: Learning discriminative deterministic state estimators," in *Advances in neural information processing systems*, 2016, pp. 4376–4384.

[23] P. Karkus, X. Ma, D. Hsu, L. P. Kaelbling, W. S. Lee, and T. Lozano-Pérez, "Differentiable algorithm networks for composable robot learning," *arXiv preprint arXiv:1905.11602*, 2019.

[24] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," *arXiv preprint arXiv:1805.11122*, 2018.

[25] X. Chen, H. Wen, and Y. Li, "Differentiable particle filters through conditional normalizing flow," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–6.

[26] Y. Wang, B. Liu, J. Wu, Y. Zhu, S. S. Du, L. Fei-Fei, and J. B. Tenenbaum, "Dualsmc: Tunneling differentiable filtering and planning under continuous pomdps," *arXiv preprint arXiv:1909.13003*, 2019.

[27] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[28] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] B. W. K. Ang and C.-H. Yeow, "A learning-based approach to sensorize soft robots," *Soft Robotics*, vol. 9, no. 6, pp. 1144–1153, 2022.

[31] H. Liu, A. Farvardin, S. A. Pedram, I. Iordachita, R. H. Taylor, and M. Armand, "Large deflection shape sensing of a continuum manipulator for minimally-invasive surgery," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 201–206.

[32] K. Xu and N. Simaan, "An investigation of the intrinsic force sensing capabilities of continuum robots," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 576–587, 2008.