α-MDF: An Attention-based Multimodal Differentiable Filter for Robot State Estimation

Anonymous Author(s)
Affiliation
Address
email

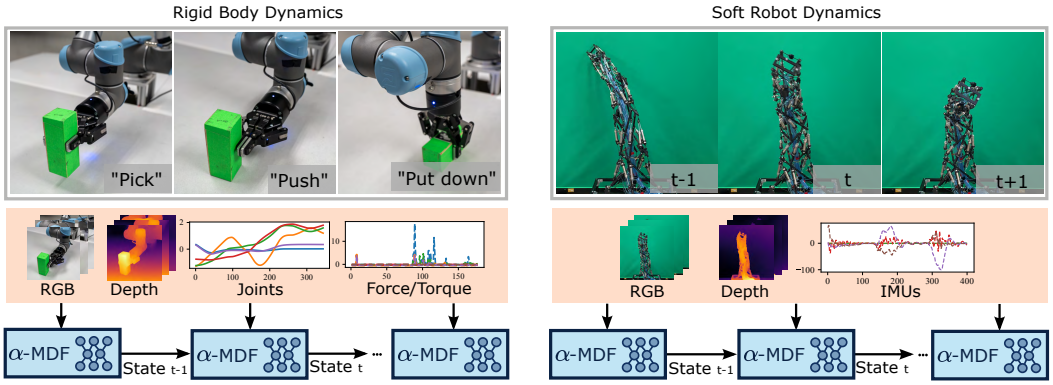

Figure 1: The attention-based Multimodal Differentiable Filters (α-MDF) framework enables robot state estimation in multimodal settings, applicable to both rigid body robots and soft robots.

Abstract: Differentiable Filters are recursive Bayesian estimators that derive the state transition and measurement models from data alone. Their data-driven nature eschews the need for explicit analytical models, while remaining algorithmic components of the filtering process intact. As a result, the gain mechanism – a critical component of the filtering process – remains non-differentiable and cannot be adjusted to the specific nature of the task or context. In this paper, we propose an attention-based Multimodal Differentiable Filter (α-MDF) which utilizes modern attention mechanisms to learn multimodal latent representations. Unlike previous differentiable filter frameworks, α-MDF substitutes the traditional gain, e.g., the Kalman gain, with a neural attention mechanism. The approach generates specialized, context-dependent gains that can effectively combine multiple input modalities and observed variables. We validate α-MDF on a diverse set of robot state estimation tasks in real world and simulation. Our results show α-MDF achieves significant reductions in state estimation errors, demonstrating nearly 4-fold improvements compared to state-of-the-art sensor fusion strategies for rigid body robots. Additionally, the α-MDF consistently outperforms differentiable filter baselines by up to 45% in soft robotics tasks.

Keywords: Differentiable Filters, Sensor Fusion, Multimodal Learning.

1 Introduction

Recursive Bayesian filters, in particular Kalman filters, are a core component of many robotic and autonomous systems [1]. These filters offer a probabilistic framework that enables effective state estimation and allowing robots to perceive and respond to dynamic environmental conditions [2, 3, 4]. Constructing the analytical models and characterizing their noise profiles can be an overwhelming undertaking and requires supplementary measures, e.g., system identification [5]. In addition,

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

scalability still poses a significant obstacle, particularly when dealing with non-linear and high-dimensional systems. Advanced techniques such as the ensemble Kalman filter [6] have been developed to tackle this challenge. However, they may still require careful manual design of the data pipeline and filtering process, especially in the presence of multimodal data sources. A potential alternative methodology is to derive the underlying models for filtering from data alone. Recent advancements in Deep state-space models (DSSMs) [7] provide effective solutions for understanding the state and measurement estimation from observed sequences as data-driven approaches [7, 8, 9]. Such approaches do not need to derive explicit system dynamics, which is essential and challenging in traditional filtering techniques. A subclass of algorithms derived from DSSMs, called Differentiable Filters (DFs), focus on learning state transition and measurement models from data while retaining the fundamental principles of Bayesian recursive filtering. This combination of properties renders DFs particularly well-suited for systems with complex dynamics and diverse sensor observations.

In this paper, we introduce a novel class of differentiable filters built upon neural attention mechanisms. The key innovation lies in the substitution of the traditional Kalman gain with an attention mechanism for filtering with multimodal observations. This approach allows for the learning of a highly specialized and task-specific gain mechanism. The utilization of multimodal observations, also known as multimodal learning [10], has shown substantial advantages in various robotics applications [11, 12, 13]. By harnessing information from diverse modalities such as vision, language [14, 15, 16], and tactile sensing [17, 18], robots can learn to better interpret their surroundings, produce more accurate estimates of their own internal state, and consequently improve the overall decision-making process. We propose the attention-based Multimodal Differentiable Filters (α-MDF) framework, as shown in Fig. 2, each module is learnable and operates in latent space. The primary contributions are: (1) **Attention Gain**: Our approach is an attention-based strategy that replaces the conventional Kalman gain in the measurement update step, as depicted by the colored blocks in Fig. 2. The gain mechanism is learned to update the current state based on multimodal observations. (2) **Latent Space Filtering**: our proposed differentiable framework operates in a latent space, learning high-dimensional representations of system dynamics and capturing intricate nonlinear relationships. This approach proves particularly advantageous for highly nonlinear systems. (3) **Empirical evaluations**: α-MDF achieves significant reductions in state estimation errors, demonstrating nearly 4-fold improvements compared to state-of-the-art sensor fusion strategies in multimodal manipulation tasks. Furthermore, α-MDF accurately models the non-linear dynamics of soft robots, consistently surpassing differentiable filter baselines by up to 45%.

2 Related Work

Differentiable Filters (DFs) have garnered attention as learnable non-linear state-space models [19, 20, 21]. Previous works [22, 23] have integrated neural network components into robotic algorithms, such as BackpropKF, which combines backpropagation with neural networks to train Kalman Filters. Similarly, research in Differentiable Particle Filters (DPFs) [24, 25] has also leveraged learnable modules to address the challenges of filtering and state tracking. Algorithmic priors have been utilized to improve the learning efficiency of DPFs, and adversarial methods have been used for posterior estimation [26]. However, the gain mechanism in the traditional Kalman filter is not differentiable and has not been incorporated into the learning procedure of the DFs mentioned earlier. In a recent study [9], the effectiveness of DFs in training and modeling uncertainty with noise profiles has been demonstrated. Typically, multi-layer perceptrons integrated with an RNN layer are employed in the implementation of DFs. Performance on real-world tasks has shown considerable improvement in state tracking accuracy [9, 18, 20], and results indicate that the adoption of end-to-end learning is crucial for accurately learning noise models. However, as noted in [8], the use of RNN has been shown to be "a limiting factor for learning accurate models" and may "lead to a non-Markovian state-space". Furthermore, the traditional Kalman gain for DFs in [9, 18, 20] remains non-learnable, despite the numerous advancements made thus far in differentiable filters and particle filters. While DFs have demonstrated considerable promise as differentiable sequential models, their application in multimodal settings has thus far received limited attention. One

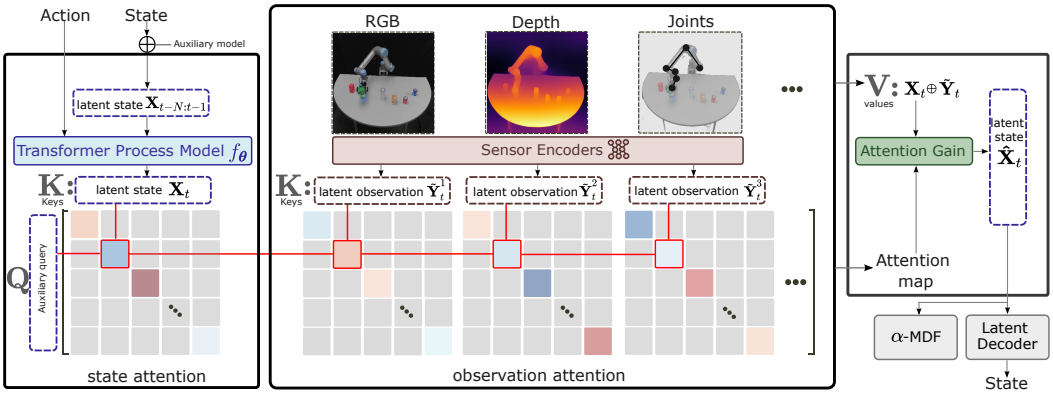

Figure 2: The α-MDF framework consists of a transformer process model, sensor encoders, and an attention gain update model. The transformer process model predicts the following latent state, while the sensor encoders learn latent representations from observations. The attention gain model then corrects the predicted latent state using the learned representations.

notable exception is the recent work in [18], which proposes sensor fusion strategies associated with DFs. Employing multimodal environments often necessitates the adoption of representation learning methodologies [27], which require learning a latent representation for capturing intricate static and dynamic features [8, 12, 28]. Studies such as [11] have highlighted the effectiveness of blending networks in learning a shared representation based on Conditional Neural Processes [29]. Other studies, including [17, 12], have adopted a self-supervised approach involving variational autoencoders (VAE) [30] to discover latent representations for stable manipulation policies. The majority of multimodal latent representations are obtained through policy learning. However, in certain situations, a gating technique [31] that focuses on specific modalities can be employed to enhance policy robustness. In light of this, we propose an alternative approach which utilizes modern attention mechanisms to learn multimodal latent representations *at both the level of modalities and observations*. The novel approach fuses multiple types of modalities and variables in a *dynamic, context-dependent* fashion, thereby enabling synergies between their respective qualities, structures, relevance, and degrees of redundancy.

3 Multimodal Differentiable Filters

We introduce a novel approach called **a**ttention-based **M**ultimodal **D**ifferentiable **F**ilters (α-**MDF**), which combines differentiable filters with insights from transformer models. This approach performs state estimation and fusion of multiple sensor modalities in a unified and differentiable manner. We first discuss Recursive Bayesian filtering as the general technique used to estimate the state $\mathbf{x}_t$ of a discrete-time dynamical system. Thereafter, we provide the details of our specific algorithm. 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:

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

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:

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

where η is a normalization factor, $p(\mathbf{y}_t|\mathbf{x}_t)$ 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.

3.1 α-MDF

We utilize an ensemble method for Bayesian filtering wherein each ensemble member represents a compact robot state. Figure 2 shows the procedural steps of how this compact representation, known as the latent state, is obtained and get updated. The filtering process includes two essential steps, namely *prediction* and *update*, both of which are also implemented through neural networks. Most importantly, we replace the Kalman gain step with an attention mechanism, which is trained to weigh observations against predictions based on the current context. Additionally, we demonstrate that attention can be used to balance and weigh different modalities, e.g., video, depth, inertial measurements, against each other. We will see that both steps can be naturally integrated into a single **attention gain** (AG) module.

Let $\mathbf{X}_{0:N}$ denote the latent states with dimension d_x of N steps in t with number of E ensemble members, $\mathbf{X}_{0:N} = [\mathbf{x}^1_{0:N}, \ldots, \mathbf{x}^E_{0:N}]$, where $E \in \mathbb{Z}^+$.

Prediction step: In this step, the state transition model takes the previous states with the current action, and predicts the next state. To this end, we leverage the capabilities of transformer-style neural networks [32]. In addition, we generate a probability distribution over the posterior by implementing the state transition model as a stochastic neural network. Therefore, we can use the following prediction step to update each ensemble member, given a sequence of latent states $\mathbf{X}_{t-N:t-1}$:

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

Where $f_{\boldsymbol{\theta}}$ is a transformer-style neural network with multiple attention layers. In our framework, the latent state and the action at t are processed by positional and type embedding layers [32] prior to being fed into $f_{\boldsymbol{\theta}}$. Matrix $\mathbf{X}_{t|t-N:t-1} \in \mathbb{R}^{d_x \times E}$ holds the updated ensemble members which are propagated one step forward in latent space. For simplicity, we represent $\mathbf{X}_{t|t-N:t-1}$ as $\mathbf{X}_t$ to denote the predicted state. Further elaboration on positional embeddings, type embeddings, and filter initialization can be found in Appendix A.1 for more comprehensive details.

Update step: A crucial step of the filtering process is the update step, which involves calculating the gain value. Traditional KF uses the Kalman gain to correct the state by comparing the uncertainty or covariance obtained from state space and observation space. However, it is important to note that the traditional Kalman gain is designed to handle a single observation space, posing a limitation in scenarios involving multiple sensor modalities. Consequently, sensor fusion becomes an additional step within the filtering framework, enabling the incorporation of observations from diverse modalities and their fusion into a unified observation. Instead, the proposed **attention gain** (AG) module for KF update and sensor fusion utilizes attention mechanisms to simultaneously perform measurement updates and leverage the information from each modality within the system. Instead of using one sensor encoder, we use multiple sensor encoders $[s^1(\cdot), s^2(\cdot), \cdots, s^M(\cdot)]$ to learn latent observations from each modality:

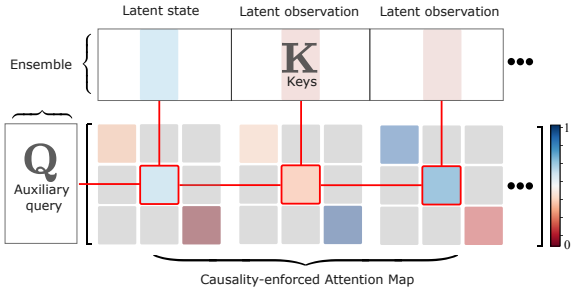

Figure 3: Attention gain (AG) module uses a learned causality-enforced attention map to replace Kalman gain.

$$\tilde{\mathbf{y}}^{(i,m)}_t \sim s^m(\tilde{\mathbf{y}}^{(i,m)}_t | \mathbf{y}^m_t), \ \forall \ i \in E, \ m \in M. \tag{4}$$

M is the number of modalities in the system, $M \in \mathbb{Z}^+$. The encoders generate a series of latent observations, $\tilde{\mathbf{Y}}_t = [\tilde{\mathbf{Y}}^1_t, \cdots, \tilde{\mathbf{Y}}^M_t] \in \mathbb{R}^{Md_x \times E}$, where $\tilde{\mathbf{Y}}^m_t = [\tilde{\mathbf{y}}^{(1,m)}_t, \cdots, \tilde{\mathbf{y}}^{(E,m)}_t] \in \mathbb{R}^{d_x \times E}$. The latent observations are then concatenated with predicted state $\mathbf{X}_t$ as input to the AG model:

$$\hat{\mathbf{X}}_t = \text{softmax}\left(\frac{\boldsymbol{Q}(\mathbf{X}'_t \oplus \tilde{\mathbf{Y}}'_t)^T}{\sqrt{E}} \circ \tilde{\boldsymbol{M}}\right)(\mathbf{X}_t \oplus \tilde{\mathbf{Y}}_t), \tag{5}$$

where "$\oplus$" denotes the concatenation and "$\circ$" is the Hadamard product, and $\hat{\mathbf{X}}_t$ is the final output. In general, an attention module typically receives three sequences of tokens: queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$. In our case, we define $(\mathbf{X}'_t \oplus \tilde{\mathbf{Y}}'_t)$ as the $\boldsymbol{K}$ tokens, where $\mathbf{X}'_t$ and $\tilde{\mathbf{Y}}'_t$ are obtained by zero-centering, and the actual values of $(\mathbf{X}_t \oplus \tilde{\mathbf{Y}}_t)$ are regarded as the $\boldsymbol{V}$ tokens. As illustrated in Fig. 3, the length of the $\boldsymbol{K}$ tokens is denoted as $d_k = (M+1)d_x$, where each token has a dimension of E, representing the distribution along this particular token index.

In a traditional attention mechanism, the proximity of $\boldsymbol{Q}$ and $\boldsymbol{K}$ is measured, and $\boldsymbol{V}$ that is associated with $\boldsymbol{K}$ is utilized to generate outputs. However, we posit that within each latent vector, every index is probabilistically independent, and index i of a latent state should only consider index i of each latent observation. To accomplish this, we utilize matrix $\tilde{\boldsymbol{M}}$ to retain only the diagonal elements of each $(d_x \times d_x)$ attention map, which enforces causality and allows the attention weights to be determined according to the corresponding indices. As depicted in Fig. 3, the red line represents the mapping for a single latent state token index. Auxiliary query tokens $\boldsymbol{Q} \in \mathbb{R}^{d_x \times E}$ are introduced as trainable parameters in the neural network to facilitate learning. It is important to note that both the $\boldsymbol{Q}$ and $\boldsymbol{K}$ tokens undergo positional embedding before being fed into the AG module.

Placing Conditions on the Latent Space: Within the framework of Kalman filters, the update step plays a crucial role in aligning the predicted observation with the observations obtained from sensors. Within the framework of α-MDF, we ensure consistency in the latent space by introducing a decoder model $\mathcal{D}$. This decoder model, implemented using multilayer perceptrons, projects the latent space onto the actual state space. By doing so, we resolve the alignment challenges in multimodal learning [33], and gain meaningful comparisons when conducting sensor fusion and measurement update. Let $\boldsymbol{x}_t$ be the ground truth state at t, the loss functions are defined as:

$$\mathcal{L}_{f_{\boldsymbol{\theta}}} = \|\mathcal{D}(f_{\boldsymbol{\theta}}(\mathbf{X}_t)) - \boldsymbol{x}_t\|_2^2, \;\; \mathcal{L}_{\text{e2e}} = \|\mathcal{D}(\hat{\mathbf{X}}_t) - \boldsymbol{x}_t\|_2^2, \;\; \mathcal{L}_s = \|\mathcal{D}(s^m(\mathbf{y}_t^m)) - \boldsymbol{x}_t\|_2^2. \tag{6}$$

The final loss function is $\mathcal{L} = \mathcal{L}_{f_{\boldsymbol{\theta}}} + \mathcal{L}_{\text{e2e}} + \mathcal{L}_s$, where $\mathcal{L}_{\text{e2e}}$ is the end-to-end loss. $\mathcal{L}_{f_{\boldsymbol{\theta}}}$ is used to supervise the state transition model, the latent observation conditioning is provided with $\mathcal{L}_s$ during the training process. The modular architecture of α-MDF provides a key advantage in facilitating training and testing with masked modalities. The attention matrix $\tilde{\boldsymbol{M}}$ can be disabled based on different input sensor modalities, thus improving the model's resilience to missing modalities.

4 Experiments

We conduct a series of experiments to evaluate the efficacy of the α-MDF framework. Specifically, we aim to answer the following questions: (a) Can the α-MDF framework generalize across various tasks? (b) To what extent does the new filtering mechanism improve state tracking performance when compared to the current state-of-the-art? (c) How does the use of multiple modalities compare to a subset of modalities for state estimation with differentiable filters? Therefore, we evaluate the effectiveness of α-MDF across multiple robotics tasks, each with distinct setups: (1) **Visual odometry** for autonomous driving, (2) **Robot manipulation** employing multi-modalities in both real-world and simulation, and (3) **Soft robot modeling** task. Our study examines two categories of baselines: (a) DF baselines such as those proposed in [9, 24, 22], including dEKF [9], DPF [24], and dPF-M-lrn [9]; and (b) sensor fusion baselines proposed in [18]. Additional details on the baselines can be found in Appendix A.3.

4.1 Visual Odometry Task

In this experiment, we evaluate the performance of α-MDF on the popular KITTI Visual Odometry dataset [34]. Since the visual odometry task uses a single modality, we only consider RGB images as the input modality in order to make a fair comparison with the baselines [9, 24, 22]. The actual state is defined as a 5-dimensional vector $\boldsymbol{x} = [x, y, \theta, v, \dot{\theta}]^T$, including the position and orientation of the vehicle, and the linear and angular velocity. We use the latent state $\mathbf{x} \in \mathbb{R}^{256}$ for α-MDF. In comparison to dEKF, DPF, and dPF-M-lrn, we observe a reduction in the translational error of approximately 88%, 83%, and 79% for Test 100/200/400/800. The results also reflect a considerable

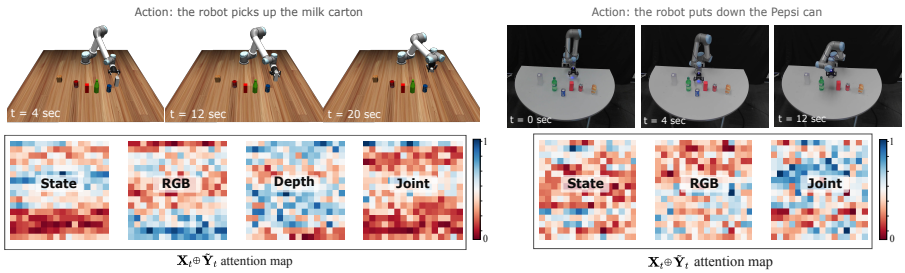

Figure 4: Learned attention gain. **Left**: manipulation in a simulated environment with modalities $[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3]$, and **right**: real robot manipulation with modalities $[\mathbf{y}^1, \mathbf{y}^3]$. The attention maps indicate the attention weights assigned to each modality during model inference. In the visualization, regions in red correspond to low attention values, while those in blue indicate high attention values.

reduction in rotational error of approximately 64%, 54%, and 46% as compared to each of the baselines. We report a detailed experimental setup and thorough results in Appendix B.1.

4.2 Multimodal Manipulation Task

This experiment aims to evaluate the effectiveness of the α-MDF framework in a robot manipulation scenario. Specifically, we use α-MDF for monitoring the state of a UR5 robot during tabletop arrangement tasks. Similar to behavioral cloning from observation tasks [35], actions are not available as inputs for this study. Instead, we train α-MDF to learn how to propagate the state of the robot over time. The evaluation involves three manipulation tasks, namely: (1) estimating the state of the robot in a simulated environment, (2) estimating the state of the real-world robot, and (3) estimating the joint state of the robot and the object being manipulated.

Task Setup and Data: For α-MDF, we define the latent state $\mathbf{x} \in \mathbb{R}^{256}$ for all the sub-tasks. The actual state of the UR5 robot is defined by $\boldsymbol{x}_R$, which consists of the joint angles (J_1-J_7) and the Cartesian coordinates (x, y, z) of the robot's end-effector (EE). $\boldsymbol{x}_O$ denotes the state of the object, which only includes the location (x, y, z) of the object. The complete set of modalities comprises $[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4]$, where $\mathbf{y}^1 \in \mathbb{R}^{224\times224\times3}$ represents RGB images, $\mathbf{y}^2 \in \mathbb{R}^{224\times224}$ represents depth maps, $\mathbf{y}^3 \in \mathbb{R}^7$ represents proprioceptive inputs (joint angles), and $\mathbf{y}^4 \in \mathbb{R}^6$ represents Force/torque (F/T) sensor readings. However, the input modalities for each of the three tasks may differ; for task (1), it involves $[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3]$, for task (2), it comprises $[\mathbf{y}^1, \mathbf{y}^3]$, while task (3) has $[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4]$. A more detailed description of the task setup and data collection is supplied in Appendix B.2.

Results: Using the same comparison protocol as in [9, 24, 22], Table 1 compares the proposed framework's performance with other DF baselines. Note that all baselines perform tracking in actual space, therefore, we use a pretrained sensor encoder to process RGB modality for all DFs and supplied the latent embedding to α-MDF, as DF baselines only take one modality. α-MDF outperforms dEKF and DPF, reducing errors by 33% and 25% in real-world and 45% and 30% in simulation, with an average MAE of 3.81cm and a deviation of 1.06cm from ground truth for end-effector positions. Additionally, α-MDF exhibits a 42% and 26% improvement in estimating joint angles compared to dPF-M-Lrn. In the case of filtering with *multiple modalities*, results presented in Table 2 show clear improvements achieved by α-MDF in comparison to other sensor fusion techniques. The baselines are reproduced following the procedure of [18] by providing the same pretrained sensor encoder to each modality. α-MDF outperforms all other methods across all three manipulation tasks. In particular, it cuts the positional error of the end-effector (EE) in half when compared to the crossmodal fusion strategy [18] on the real-robot (7.35cm $\rightarrow$ 3.04cm). In simulation tasks, it achieves an even better reduction in

Table 1: Result evaluations on UR5 manipulation task

Method	Real-world (MAE)		Simulation (MAE)	
	Joint (deg)	EE (cm)	Joint (deg)	EE (cm)
dEKF [9]	16.08±0.1	5.67±0.1	4.93±0.2	1.91±0.1
DPF [24]	15.93±0.1	5.08±0.3	4.46±0.2	1.51±0.2
dPF-M-lrn [9]	12.83±0.1	3.95±0.4	3.82±0.2	1.26±0.1
α-MDF	**7.49±0.1**	**3.81±0.2**	**2.84±0.1**	**1.06±0.1**

Means±standard errors.

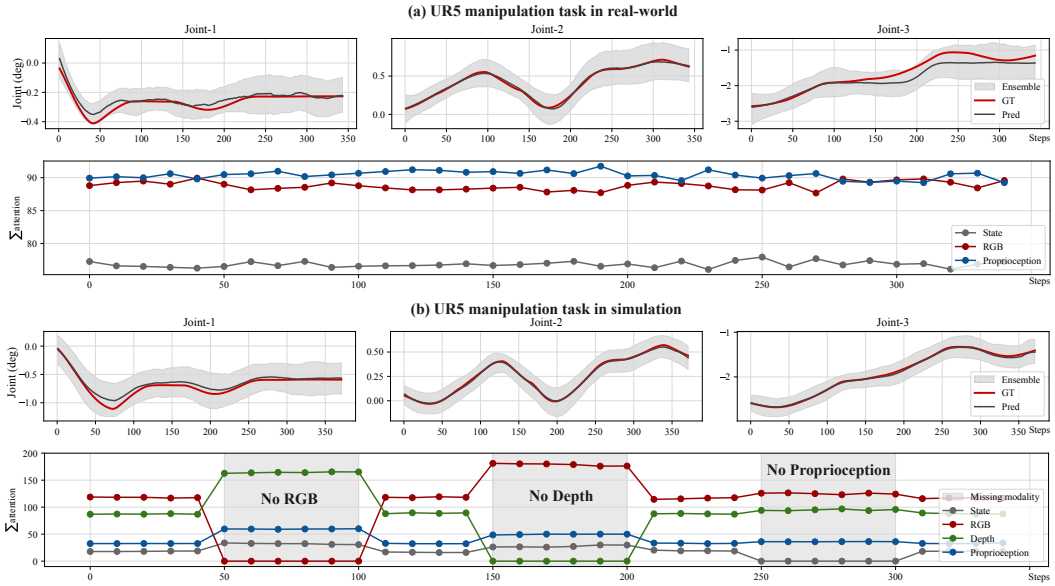

Figure 5: Predicted joint angle trajectories and the corresponding accumulated attention values for each modality. (a) represents the results attained from the actual robot, whereas (b) illustrates attention values for all modalities both with and without masking certain modalities.

Table 2: Result evaluations on UR5 manipulation task with multimodal sensor fusion baselines.

Method	Simulation (MAE)		Real-world (MAE)		Simulation with F/T (MAE)		
	Joint (deg)	EE (cm)	Joint (deg)	EE (cm)	Joint (deg)	EE (cm)	Obj (cm)
Feature Fusion [18]	7.58±0.12	3.15±0.16	11.25±1.17	5.65±0.01	3.62±0.09	2.72±0.02	8.36±0.06
Unimodal [18]	7.46±0.32	3.18±0.03	11.02±0.08	9.52±0.07	3.97±0.08	3.63±0.05	10.23±0.10
Crossmodal [18]	3.64±0.34	1.91±0.04	5.98±0.08	7.35±0.05	3.12±0.02	3.25±0.02	5.54±0.02
α-MDF	**2.19±0.09**	**0.75±0.01**	**5.24±0.04**	**3.04±0.01**	**1.41±0.04**	**0.90±0.01**	**1.65±0.01**

Means±standard errors.

tracking error (3.25cm $\rightarrow$ 0.90cm in simulation with F/T sensing). We present a visualization of the learned attention gain in the filtering process (Fig. 4) and state tracking results with and without certain modalities (Fig. 5). Despite the attention values changing when certain modalities are missing, α-MDF still achieves stable results. Further results and explanations can be found in AppendixB.2.

4.3 Soft robot Modeling

Tensegrity structures [36] have become popular in recent years since they bridge the gap between an inherently flexible system and the ability to use rigid components [37, 38, 39]. However, modeling such a complex system continues to pose considerable challenges due to the high nonlinearity. This experiment involves implementing the α-MDF to model the dynamics of a soft robot system, especially Tensegrity robot [39].

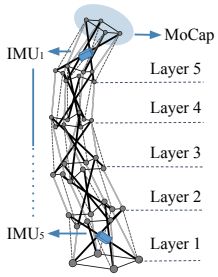

Figure 6: The tensegrity robot structure.

Task Setup and Data: The robot structure is shown in Fig. 6 with 5 layers of tensegrity. The actual state of a soft robot at time t is represented by a 7-dimensional vector $\boldsymbol{x}_t = [x, y, z, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_w]^T$, which denotes the position and orientation of the robot's hand tip. The quaternion vector $\mathbf{q}$ represents the posture of the robot w.r.t the base (layer 1's bottom). In this task, we define $\mathbf{x} \in \mathbb{R}^{256}$ as the latent state. The complete set of modalities comprises $[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3]$, where $\mathbf{y}^1 \in \mathbb{R}^{224\times224\times3}$ represents RGB images, $\mathbf{y}^2 \in \mathbb{R}^{224\times224}$ is depth maps, and $\mathbf{y}^3 \in \mathbb{R}^{30}$ is proprioceptive inputs (IMUs). The action $\mathbf{a}_t$ of the system is the pressure vector of the 40 pneumatic cylinder actuators, where $\mathbf{a}_t \in \mathbb{R}^{40}$. In this experiment, synthetic depth maps are generated offline using the DPT model [40]. Figure. 7 shows the recordings of RGB and the depth modalities, further details regarding the task setup and data collection is in Appendix B.3.

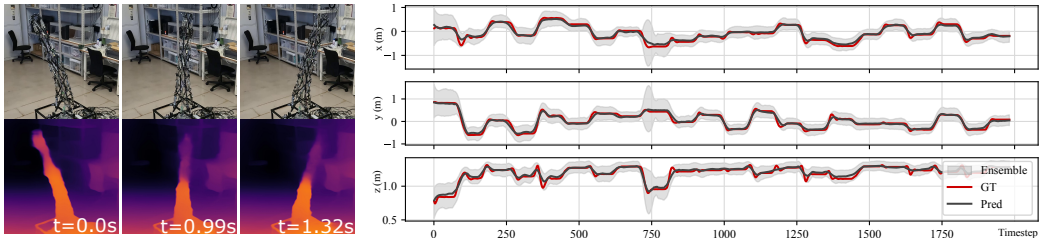

Figure 7: Estimated end-effector (EE) positions for tensegrity robot. **Left**: the RGB and depth modalities $[\mathbf{y}^1, \mathbf{y}^2]$, and **right**: state estimation results with ensemble distribution.

Results: The soft robot modeling task is evaluated using 10-fold cross-validation and the mean absolute error (MAE) metric, and the results are presented in Table 3. Our results demonstrate that α-MDF outperforms the state-of-the-art methods in terms of DFs, achieving a MAE of 8.99cm. Specifically, our approach yields an MAE on the end-effector (EE) position estimation that is 45%, 34%, and 29% lower than that obtained by dEKF, DPF, and dPF-M-lrn, respectively. Of the sensor fusion baselines, cross-modal fusion [18] exhibits marginally better outcomes than others, although it do not show any advantages over α-MDF in predicting EE positions (2.14cm→1.67cm). Notably, α-MDF surpasses the feature fusion strategy by a significant margin of 4-fold. Additionally, appendix B.3 delves into an exploration of the potential benefits of modality selection for state estimation, where optimal combinations can be selected to achieve even higher accuracy. The results presented in Fig. 7 demonstrate the efficacy of α-MDF in accurately estimating the state of soft robots in a multimodal setting, the ensemble distribution is indicated by gray shade representing the model uncertainty. With stable performance achieved over an extended duration of inference, α-MDF has shown the potential in modeling dynamics for various complex non-linear systems.

Table 3: Result evaluation on soft robot modeling task.

	RGB	Depth	IMUs	EE (cm)	$\mathbf{q}$ (10^1)
dEKF [9]			✓	16.38±0.10	1.01±0.03
DPF [24]			✓	13.68±0.02	0.96±0.03
dPF-M-lrn [9]			✓	12.66±0.09	1.10±0.03
α-MDF			✓	**8.99±0.02**	**0.79±0.03**
Feature Fusion [18]	✓	✓	✓	8.35±0.22	0.60±0.03
Unimodal [18]	✓	✓	✓	2.78±0.05	0.25±0.02
Crossmodal [18]	✓	✓	✓	2.14±0.05	0.15±0.02
α-MDF	✓	✓	✓	**1.67±0.09**	**0.12±0.01**

Means±standard errors.

5 Conclusion

This paper illustrates how utilizing attention as a gain mechanism in differentiable Bayesian filtering and multimodal learning can significantly enhance the accuracy of robot state estimation in numerous tasks. Proposed α-MDF is a unique differentiable filter that conducts filtering on a compressed multimodal latent representation, while preserving the integrity of the Kalman filter algorithm component. Our experiments demonstrate that α-MDF is appropriate for learning both rigid body and soft robot dynamics, exceeding baseline performance by up to 4-fold. Moving forward, we plan to investigate the value of incorporating additional modalities, such as sound, temperature, and proximity sensing, into α-MDF.

Limitation: An obvious difference of α-MDF when compared to traditional filters is the required learning process – this typically takes multiple hours of training on current machines. In a similar vain, an inherent assumption is that the training and test distributions do not differ substantially, i.e., the problem of concept drift. To date, we have successfully tested the algorithm with latent states consisting of several hundred variables. We use 256 dimensions in the experiments for consistency. However, more research is required to understand α-MDF's performance when filtering over thousands of variables. As with any deep learning approaches, hyper-parameter tuning may be required to produce high-performing models. Another practical observation is that utilizing more modalities does not always translate to improved performance, which is consistent with findings in [41]. Including redundant modalities can impose longer training times and pose greater difficulty for the model in extracting valuable information from the input modalities. A pre-processing step for feature selection may be advisable.

References

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005. ISBN 0262201623 9780262201629.

[2] L. Wang, G. Wang, S. Jia, A. Turner, and S. Ratchev. Imitation learning for coordinated human–robot collaboration based on hidden state-space models. *Robotics and Computer-Integrated Manufacturing*, 76:102310, 2022. ISSN 0736-5845. doi:https://doi.org/10.1016/j.rcim.2021.102310.

[3] S. Chen. Kalman filter for robot vision: a survey. *IEEE Transactions on industrial electronics*, 59(11):4409–4420, 2011.

[4] J. Reher, W.-L. Ma, and A. D. Ames. Dynamic walking with compliance on a cassie bipedal robot. In *2019 18th European Control Conference (ECC)*, pages 2589–2595. IEEE, 2019.

[5] L. Ljung. *System identification*. Springer, 1998.

[6] G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

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

[8] 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*, 34, 2021.

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

[10] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, 34(6):96–108, 2017.

[11] M. Y. Seker, A. Ahmetoglu, Y. Nagai, M. Asada, E. Oztop, and E. Ugur. Imitation and mirror systems in robots through deep modality blending networks. *Neural Networks*, 146:22–35, 2022.

[12] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[13] T. Xue, W. Wang, J. Ma, W. Liu, Z. Pan, and M. Han. Progress and prospects of multimodal fusion methods in physical human–robot interaction: A review. *IEEE Sensors Journal*, 20(18): 10355–10370, 2020.

[14] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[15] Y. Zhou, S. Sonawani, M. Phielipp, S. Stepputtis, and H. B. Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. *arXiv preprint arXiv:2212.04573*, 2022.

[16] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[17] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.

[18] 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)*, pages 10444–10451. IEEE, 2020.

[19] A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pages 2100–2111. PMLR, 2021.

[20] N. A. Piga, U. Pattacini, and L. Natale. A differentiable extended kalman filter for object tracking under sliding regime. *Frontiers in Robotics and AI*, 8:686447, 2021.

[21] W. Li, X. Chen, W. Wang, V. Elvira, and Y. Li. Differentiable bootstrap particle filters for regime-switching models. *arXiv preprint arXiv:2302.10319*, 2023.

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

[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)*, pages 1–6. IEEE, 2021.

[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. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[28] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. In *Conference on Robot Learning*, pages 249–261. PMLR, 2017.

[29] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018.

[30] H. Van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3928–3934. IEEE, 2016.

[31] J. Hansen, F. Hogan, D. Rivkin, D. Meger, M. Jenkin, and G. Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE, 2022.

[32] 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*, 30, 2017.

[33] P. P. Liang, A. Zadeh, and L.-P. Morency. Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions. *arXiv preprint arXiv:2209.03430*, 2022.

[34] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[35] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[36] R. E. Skelton and M. C. Oliveira. *Tensegrity Systems*. Springer Nature, 2009. ISBN 978-1-4419-4491-7.

[37] 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)*, pages 5561–5566. IEEE, 2018.

[38] K. Kim, A. K. Agogino, and A. M. Agogino. Rolling locomotion of cable-driven soft spherical tensegrity robots. *Soft robotics*, 7(3):346–361, 2020.

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

[40] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[41] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020.

[42] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17 (2):29–48, 2022. doi:10.1109/MCI.2022.3155327.

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

[44] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

[45] K. Lenac, J. Ćesić, I. Marković, and I. Petrović. Exactly sparse delayed state filter on lie groups for long-term pose graph slam. *The International Journal of Robotics Research*, 37(6): 585–610, 2018.

[46] J. Zhang and S. Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.

[47] C.-C. Chou and C.-F. Chou. Efficient and accurate tightly-coupled visual-lidar slam. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[48] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović. Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of field robotics*, 35(4):578–595, 2018.

[49] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[50] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[51] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[52] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.