Signal Processing 230 (2025) 109856

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

RKFNet: A novel neural network aided robust Kalman filter

Pengcheng Hao [a,*], Oktay Karakuş [b], Alin Achim [a]

[a] Visual Information Laboratory, University of Bristol, Bristol, UK
[b] School of Computer Science and Informatics, Cardiff University, Cardiff, UK

ARTICLE INFO

Keywords:
Robust Kalman filter
Heavy-tailed noise
Deep learning

ABSTRACT

Driven by the filtering challenges in linear systems disturbed by non-Gaussian heavy-tailed noise, robust Kalman filters (RKFs) leveraging diverse heavy-tailed distributions have been introduced. However, the RKFs rely on precise noise models, and large model errors can degrade their filtering performance. Also, the posterior approximation by the employed variational Bayesian (VB) method can further decrease the estimation precision. Here, we introduce an innovative RKF method, the RKFNet, which combines the heavy-tailed-distribution-based RKF framework with the deep learning technique and eliminates the need for the precise parameter estimation of the heavy-tailed distributions. To reduce the VB approximation error, the mixing-parameter-based function and the scale matrix are estimated by the incorporated neural network structures. Also, the stable training process is achieved by our proposed unsupervised scheduled sampling (USS) method, where a loss function based on the Student's t (ST) distribution is utilised to overcome the disturbance of the noise outliers and the filtering results of the traditional RKFs are employed as reference sequences. Furthermore, the RKFNet is evaluated against various RKFs and recurrent neural networks (RNNs) under three kinds of heavy-tailed measurement noises, and the simulation results showcase its efficacy in terms of estimation accuracy and efficiency.

1. Introduction

Kalman filtering is an algorithm used for estimating the state of a dynamic system from a series of noisy measurements over time [1]. It takes into account the uncertainties associated with both the measurements and the dynamic model, and recursively updates an estimate of the state, minimising the mean squared error (MSE). Due to the versatility and effectiveness of the Kalman filter (KF), it has been a valuable tool in various fields, including target tracking [2], the interest rates and inflation prediction in economics [3,4], the GPS navigation system [5], etc. While the KF yields optimal estimates under linear Gaussian models, its filtering performance may be undermined by heavy-tailed noise, which deviates from the Gaussian assumption.

Various robust Kalman filters (RKFs) have been developed to address scenarios involving heavy-tailed noise. For the M-estimator-based and outlier-detection-based RKFs, several M-estimators and outlier detectors are leveraged within the KF framework to improve the robustness of state estimation in the presence of outliers. For example, one introduces the RKFs based on the Huber function [6,7], correntropy [8,9] and statistical similarity measure [10]. Also, in [11,12], the Chi-Square test and variational Bayesian (VB) strategies are utilised to detect outliers and formulate novel RKF frameworks. However, their filtering performance is constrained as the stochastic characteristics of the noise are not leveraged. By contrast, the heavy-tailed-distribution-based RKFs offer a solution for the exploitation of the noise stochastic nature and provide more accurate estimation. For instance, in [13], the robust Student's t-distribution-based Kalman filter (RSTKF) writes the prediction and likelihood PDFs in hierarchical Gaussian forms, and the joint posterior distribution is approximated by the VB approach [14]. Also, considering the skewed noise, the RKF based on the Gaussian scale mixture (GSM) distribution is proposed in [15], where the one-step prediction and likelihood PDFs are formulated as various skewed GSM distributions. Nonetheless, the employed VB method updates all variational parameters concurrently within the same iteration, potentially introducing instability to the estimation process. Instead, employing a heuristic approach, [16] presents an elliptically-contoured-distribution-based RKF framework, where however the uncertainty of the scale matrices is not considered. Besides, in response to filtering challenges presented by the stable noise, the RKF based on sub-Gaussian α-stable (SGαS) distribution [17] (RKF-SGαS) is developed in [18], and the efficient estimators of the mixing parameter are discussed. With precise noise models, the heavy-tailed-distribution-based RKFs provide robust filtering results. However, large model errors can degrade the performance of these model-based algorithms significantly.

* Corresponding author.
E-mail addresses: ju18422@bristol.ac.uk (P. Hao), karakuso@cardiff.ac.uk (O. Karakuş), alin.achim@bristol.ac.uk (A. Achim).

https://doi.org/10.1016/j.sigpro.2024.109856
Received 14 May 2024; Received in revised form 20 September 2024; Accepted 10 December 2024
Available online 19 December 2024

Benefiting from the deep learning techniques [19], various data-driven filters have been presented, and accurate knowledge of the state-space model is not required. A prevalent method is to utilise the recurrent neural networks (RNNs), including the vanilla RNN [20], long short-term memory (LSTM) networks [21,22], gated recurrent units (GRUs) [23], attention mechanisms [24] and so on, to process the observations and generate state estimates sequentially. For example, [25–27] employ the LSTM, GRU and transformers for filtering or prediction tasks, respectively. However, to cope with the complexity of uncertain real-world data and achieve better belief approximation, a long latent vector is needed, which thus increases the number of network parameters and the amount of training data. By contrast, the hybrid frameworks combining RNNs with traditional filters can alleviate this drawback. For instance, the particle filter recurrent neural networks [28] combine the strengths of the RNNs and the particle filter, approximating the latent state distribution as a set of particles. Without lengthening the latent vector, the required data amount is reduced. In comparison, the hybrid LSTM-KF [29] learns the parameters of the KF by the LSTM and outperforms both the standalone KF and LSTM. Besides, utilising the partially known dynamics, [30] proposes a data-driven and model-based (DM) neural network, KalmanNet, where the structural state-space model with a dedicated RNN module is embedded in the flow of the extended KF updating framework. Compared with traditional RNNs, the KalmanNet is more interpretable and can be trained with a smaller dataset. Also, due to the embedded RNN structure, the KalmanNet can accurately characterise the state dynamics and obtain more precise estimates than the model-based filters. Nevertheless, due to the Gaussian assumption of the incorporated extended KF method, the performance of the KalmanNet is limited under the heavy-tailed noise.

In this work, we focus on a new DM RKF framework, and the motivation of this work relies on the complementary advantages of the DM KalmanNet and the RKFs. Specifically, the performance of the heavy-tailed-distribution-based RKFs relies on precise noise models, and extra estimation errors are caused by the posterior approximation of the employed VB method. By contrast, the KalmanNet has no requirement for precise models and can produce more accurate estimates than the model-based RKFs. In addition, the KalmanNet is developed for light-tailed Gaussian noise scenarios, whereas the RKF framework can overcome the influence of outliers in the data. We propose a hybrid framework, where the benefits of both the DM strategy and the RKF framework can be leveraged. Specifically, the main contributions of this work consist in:

1. The DM RKFNet is presented, combining the heavy-tailed-distribution-based RKF framework and deep learning techniques. Particularly, we consider a state-space model, where the heavy-tailed distribution of the measurement noise is expressed in a hierarchical Gaussian form without prior assumptions on its mixing density and scale matrix. Then the posterior state estimation is achieved by a KF update step with the aid of the incorporated neural networks.
2. To improve the stability of the training process, we present an unsupervised scheduled sampling (USS) training method. In particular, the loss function based on the Student's t (ST) distribution is exploited, and the reference sequences are provided by the traditional RKFs.
3. In the simulations, we evaluate the influence of many factors on the performance of the RKFNet. Also, a comparison between the proposed method and various benchmark filters is conducted under different types of heavy-tailed measurement noise.

The remainder of this paper is structured as follows: We begin, in Section 2, with an introduction to the theoretical background, including the linear discrete-time state-space model and deep neural networks. Subsequently, Section 3 describes the architecture of the proposed RKFNet, and the USS technique is elucidated in Section 4. Furthermore, the developed framework is assessed in target tracking scenarios in Section 5, whilst Section 6 wraps up this study with a summary.

Table 1
Mathematical notations.

Notations	Definitions
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathcal{N}(.; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$\mathbf{I}_n$	$n \times n$ identity matrix.
$\lvert \cdot \rvert$	Element-wise absolute value operation.
$\det(\mathbf{X})$	Determinant of square matrix $\mathbf{X}$.
$(\cdot)^T, (\cdot)^{-1}$	Transpose and inverse operation.
$\text{sgn}(\cdot)$, $\lvert\lvert \cdot \rvert\rvert^2$	Signum function and $L2$ norm.
$\exp(\cdot)$, $\log(\cdot)$	Exponential and logarithm function.
$\tanh(\cdot)$	Hyperbolic tangent function.
$st(.; \upsilon, \sigma)$	One-dimensional ST PDF with the dof value υ and scale parameter σ.
$\frac{\partial(.)}{\partial(.)}$, $\propto$	Partial differential operation and proportional symbol.
$\max(a, b)$	Maximum of a and b.

2. Theoretical preliminaries

In this section, we provide brief, essential details, on the key concepts, which serve as the foundation for the main developments presented in the following sections. We start by introducing in Table 1 the main mathematical notations employed throughout this study.

2.1. Linear discrete-time state-space model

A linear discrete-time state-space model comprises two distinct models. The signal model characterises the state transition dynamics over time, whereas the measurement model elucidates the association between the states and their corresponding measurements. A typical linear discrete-time state-space model is expressed as follows:

$$\begin{cases} \mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{z}_k \in \mathbb{R}^m$ represent the hidden state and the measurement at time k, respectively. Also, $\mathbf{F}_k \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_k \in \mathbb{R}^{m \times n}$ are the state transition and measurement matrices, respectively. Besides, $\mathbf{w}_k \in \mathbb{R}^n$ denotes the process noise, whilst the measurement noise is represented as $\mathbf{v}_k \in \mathbb{R}^m$. In this work, we assume $\mathbf{w}_k$ follows a Gaussian distribution and $\mathbf{v}_k$ is heavy-tailed.

2.2. Deep neural networks for filtering tasks

Neural networks are a class of machine learning models inspired by the structure of the human brain and consist of interconnected nodes, called neurons, organised into layers. Each neuron takes inputs, performs computations and produces an output. Deep neural networks [19] are neural networks with multiple hidden layers between the input and output layers. These hidden layers enable the network to learn hierarchical representations of the input data, capturing complex patterns and relationships. An multilayer neural network follows a feedforward architecture. The information flows from the input layer through the hidden layers to the output layer, and each layer contains multiple neurons. For a fully connected neural network (FCNN), each neuron in one layer is connected to every neuron in the subsequent layer. To properly adjust the weights of neural networks, a training process based on an objective function is required, and there are two steps: forward pass and backpropagation. During the forward pass, the input data is fed into the network, and the outputs of each layer are computed sequentially. By contrast, during the backpropagation step, the gradients of the model parameters with respect to the loss function are calculated by the chain rule. Then, the gradient-based optimisation algorithms, such as the stochastic gradient descent and the Adam algorithm, are employed to update the parameters.

Among various deep neural networks, RNNs [19] are a set of neural networks dealing with time series data and are thus often employed for

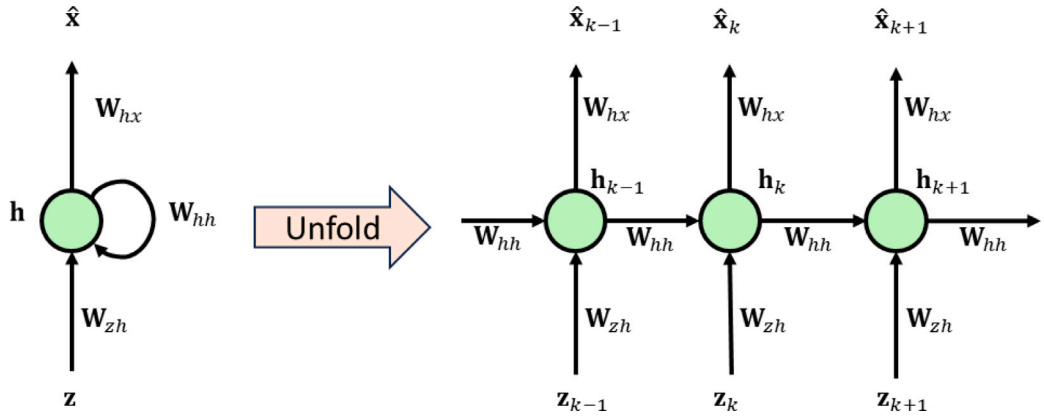

Fig. 1. A filtering framework based on RNNs.

filtering tasks. Fig. 1 illustrates a filtering framework based on RNNs. At any time k, the RNN processes the observation $\mathbf{z}_k$ and maintains the hidden states $\mathbf{h}_{k-1}$, containing past information. The recurrent nature allows the network to capture temporal dependencies in the sequential data. The vanilla RNN is the simplest form of RNNs, and its computational framework can be expressed as

$$\mathbf{h}_k = \tanh\left(\mathbf{W}_{hh}\mathbf{h}_{k-1} + \mathbf{W}_{zh}\mathbf{z}_k\right)$$
$$\hat{\mathbf{x}}_k = \mathbf{W}_{xh}\mathbf{h}_k,$$

where $\hat{\mathbf{x}}_k$ represents the estimate of $\mathbf{x}_k$. Also, the parameters, $\mathbf{W}_{hh}$, $\mathbf{W}_{zh}$ and $\mathbf{W}_{xh}$, are shared across all time steps, allowing the vanilla RNN to capture the temporal patterns in the sequential data. However, its performance is limited in learning long-term dependencies due to the vanishing gradient problem. Specifically, the network is trained using backpropagation through time, and then the gradients can diminish or explode exponentially when they propagate through multiple time steps.

To alleviate the vanishing gradient problem, the LSTM structure [21] introduces a memory cell and three gating mechanisms: input gate, forget gate, and output gate. These gates regulate the information flow within the LSTM cell, allowing it to retain important information over long sequences and forget irrelevant information. To simplify the LSTM structure, the GRU structure [23] is proposed, combining the forget and input gates into a single update gate and merging the cell state and hidden state. The GRU has fewer parameters than the LSTM and hence is more computationally efficient. To evaluate our proposed method, these classic RNNs are compared with the RKFNet in Section 5.

3. RKFNet architecture

This section explains our proposed RKFNet structure. In Section 3.1, we describe a state-space model, which is based on a linear model with Gaussian signal noise and unknown heavy-tailed measurement noise. Also, the proposed filtering framework is detailed in Section 3.2.

3.1. Hierarchical Gaussian state-space model with arbitrary heavy-tailed noise distributions

We assume the zero-mean process noise follows the Gaussian distribution and the symmetric heavy-tailed distribution of the measurement noise can be written in a hierarchical Gaussian form, i.e.,

$$p(\mathbf{w}_{k-1}) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \tag{2}$$
$$p(\mathbf{v}_k; \mathbf{R}) = \int_0^{+\infty} \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \lambda_k\mathbf{R})\pi(\lambda_k)d\lambda_k, \tag{3}$$

where $\mathbf{Q}_{k-1} \in \mathbb{R}^{n\times n}$ is the covariance matrix of the state noise at time $k-1$. Also, $\mathbf{R} \in \mathbb{R}^{m\times m}$ is an unknown scale matrix, and λ_k is a scalar mixing parameter and follows the unknown mixing density $\pi(\lambda_k)$. Then the prediction and likelihood PDF can be expressed as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1}) \tag{4}$$
$$p(\mathbf{z}_k|\mathbf{x}_k; \mathbf{R}) = \int_0^{+\infty} \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \lambda_k\mathbf{R})\pi(\lambda_k)d\lambda_k \tag{5}$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the posterior mean vector at time $k-1$ and the prior error covariance matrix $\mathbf{P}_{k|k-1}$ can be calculated based on the posterior error covariance matrix $\mathbf{P}_{k-1|k-1}$, i.e.,

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_{k-1}. \tag{6}$$

Remark 1. In the traditional RKF frameworks, various heavy-tailed distributions have been utilised to fit the measurement noise, and the corresponding likelihood PDFs can be seen as an approximation of Eq. (5). Specifically, $\pi(\lambda_k)$ is approximated by a fully skewed mixing density, and the uncertainty about the scale matrix $\mathbf{R}$ is represented by an inverse Wishart distribution [15]. Although showing robustness and efficiency in many scenarios, the performance of the RKFs degrades when the model error is large. By contrast, the model approximation is not required in our work.

3.2. The proposed RKFNet filtering framework

Based on the forecast and likelihood PDFs in (4) and (5), the joint posterior distribution can be expressed as

$$\begin{aligned} p(\tilde{\boldsymbol{\Theta}}_k|\mathbf{z}_{1:k}; \mathbf{R}) &\propto p(\mathbf{z}_k|\tilde{\boldsymbol{\Theta}}_k; \mathbf{R})p(\tilde{\boldsymbol{\Theta}}_k|\mathbf{z}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \lambda_k\mathbf{R}) \times \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1}) \times \pi(\lambda_k), \end{aligned} \tag{7}$$

where $\tilde{\boldsymbol{\Theta}}_k = \{\mathbf{x}_k, \lambda_k\}$ and then an RKF framework can be derived as explained in Proposition 1:

Proposition 1 (*A Similar Proof Can Be Seen in Appendix C of [15]*). *Given the posterior distribution of* λ_k,

$$p(\lambda_k|\mathbf{z}_{1:k}; \mathbf{R}) = \int p(\tilde{\boldsymbol{\Theta}}|\mathbf{z}_{1:k}; \mathbf{R})d\mathbf{x}_k, \tag{8}$$

the marginal posterior distribution of $\mathbf{x}_k$ *can be approximated as a Gaussian distribution, i.e.,*

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}; \mathbf{R}) \approx \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\right),$$

where

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}^T\mathbf{H}_k^T + \tilde{\mathbf{R}}_k\right)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) \\ \mathbf{P}_{k|k} &= \left(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1} \end{aligned} \tag{9}$$

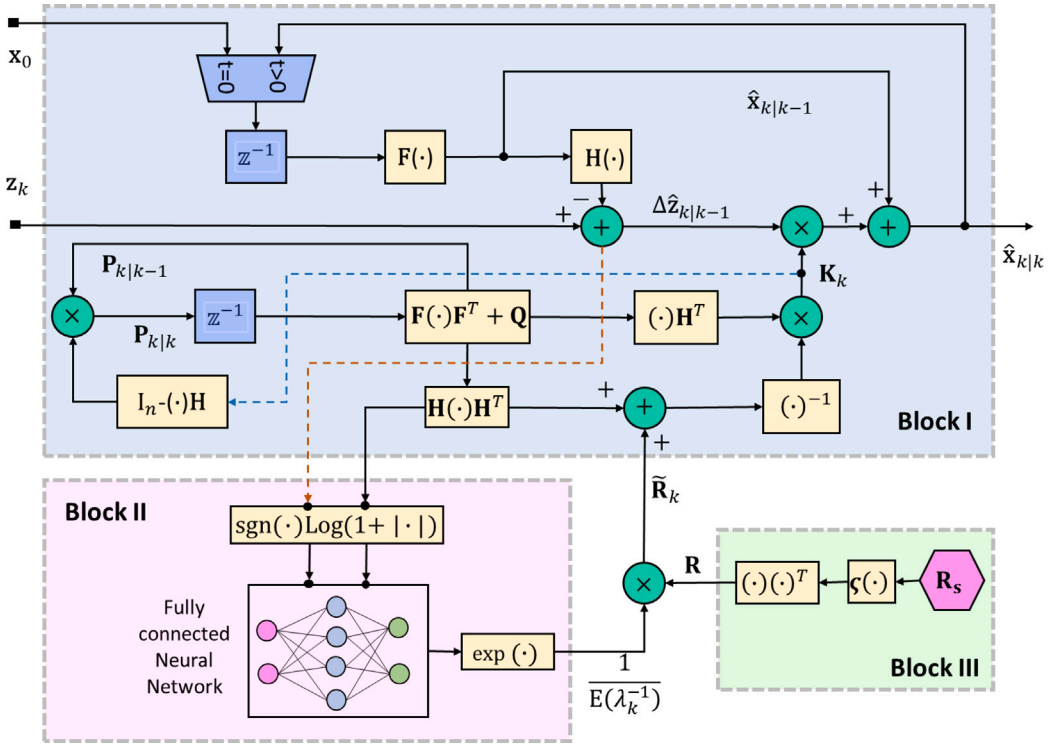

Fig. 2. The RKFNet structure.

and $\mathbf{K}_k$ *denotes the Kalman gain. Also,* $\tilde{\mathbf{R}}_k$ *is a modified matrix of the true measurement noise covariance matrix* $\mathbf{R}_k$ *at time* k*, and*

$$\tilde{\mathbf{R}}_k = \frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}\mathbf{R}, \quad \mathbf{R}_k = \lambda_k \mathbf{R}, \tag{10}$$

where

$$\mathrm{E}\left(\lambda_k^{-1}\right) = \int_0^{+\infty} \lambda_k^{-1} p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R}) d\lambda_k.$$

Remark 2. Proposition 1 provides an efficient RKF framework for the posterior estimation of $\mathbf{x}_k$. However, in Eq. (10), the calculation of $\tilde{\mathbf{R}}_k$ requires the unknown $\mathbf{R}$ and $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$.

To estimate $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$ and $\mathbf{R}$, and produce the posterior state estimates, we present a new neural network architecture, the RKFNet, combining the RKF framework in Proposition 1 with the deep learning technique. As shown in Fig. 2, the RKFNet consists of three blocks, where Block I produces the posterior state estimation based on the KF framework in Eq. (9). Also, $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$ and $\mathbf{R}$ are estimated by Block II and Block III, respectively. The details of the RKFNet are explained below:

(1) At any time k, Block I receives the new observation $\mathbf{z}_k$ and the estimated $\tilde{\mathbf{R}}_k$, and produces $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$.

(2) To estimate $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$, we investigate the marginal posterior distribution of λ_k in Theorem 1.

Theorem 1. *At any time* k*, given the new observation* $\mathbf{z}_k$*, the stochastic properties of* $p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R})$ *and* $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$ *are determined by* $\Delta\mathbf{z}_{k|k-1}$*,* $\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T$*,* $\mathbf{R}$ *and* $\pi(\lambda_k)$*.*

Proof. According to Eqs. (7) and (8), the marginal posterior distribution of λ_k can be expressed as

$$p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R}) \propto \int \mathcal{N}(\mathbf{z}_k;\mathbf{H}_k\mathbf{x}_k,\lambda_k\mathbf{R}) \times \mathcal{N}(\mathbf{x}_k;\mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1},\mathbf{P}_{k|k-1}) \times \pi(\lambda_k) d\mathbf{x}_k.$$

By integrating $\mathbf{x}_k$, then

$$p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R}) \propto \mathcal{N}(\Delta\mathbf{z}_{k|k-1};\mathbf{0},\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \lambda_k\mathbf{R})\pi(\lambda_k),$$

and hence we have Theorem 1. □

According to Theorem 1, Block II employs an FCNN to estimate $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$, and the input features are selected as $\Delta\mathbf{z}_{k|k-1} = \mathbf{z}_k - \mathbf{H}\mathbf{x}_{k|k-1}$ and $\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T$. As $\mathbf{R}$ and $\pi(\lambda_k)$ are fixed, they are not incorporated as input features. Also, considering the outliers of the input features, $\mathrm{sgn}(\cdot)\log(1+|\cdot|)$ is employed to shrink their values. Moreover, to ensure a positive estimate of $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$ and narrow the range of the FCNN outputs, $\exp(\cdot)$ is connected to the FCNN output layer.

Remark 3. Due to the complexity of $p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R})$, there is no close form of $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$. In the traditional RKF frameworks, an approximated PDF of $p(\lambda_k|\mathbf{z}_{1:k};\mathbf{R})$ is provided by the VB method [15], and then the estimator of $\frac{1}{\mathrm{E}\left(\lambda_k^{-1}\right)}$ can be derived. However, the VB method can introduce large approximation errors. By contrast, the RKFNet employs the neural-network-based estimator in Block II to provide better posterior estimation.

(3) In Block III, we introduce a neural network parameter $\mathbf{R}_s \in \mathbb{R}^{m\times m}$, and the scale covariance matrix $\mathbf{R}$ can be estimated by

$$\hat{\mathbf{R}} = (\varsigma_1\mathbf{R}_s)(\varsigma_1\mathbf{R}_s)^T, \tag{11}$$

where $\varsigma_1 > 1$ is used to increase the gradient of the loss function to $\mathbf{R}_s$. This allows for efficient updates of $\mathbf{R}_s$, particularly when the learning rate is small. Also, Eq. (11) guarantees the symmetry and positive semi-definiteness of $\hat{\mathbf{R}}$.

4. Unsupervised training algorithm

Following the introduction of the RKFNet architecture, an unsupervised training method is proposed in this section, which can benefit various practical applications in scenarios where the ground-truth data is difficult to obtain. In [31], the unsupervised loss function is

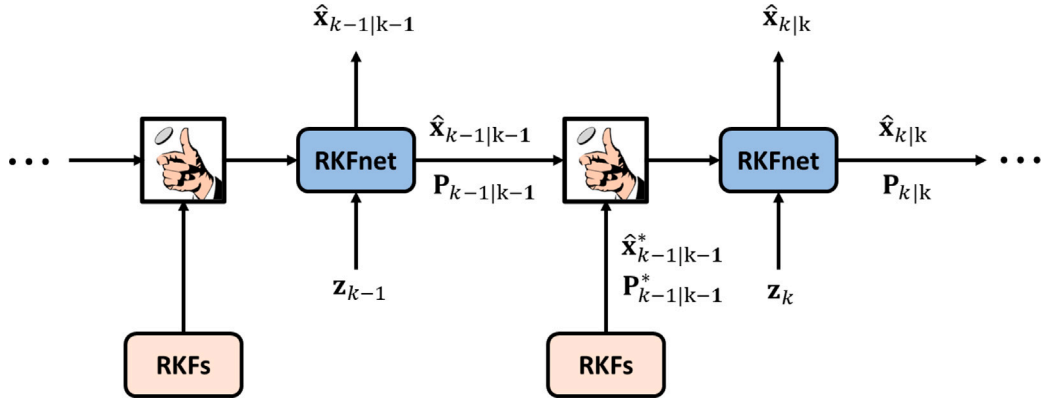

Fig. 3. The unsupervised scheduled sampling technique.

$\|\Delta \mathbf{z}_{k+1|k}\|^2$. Although achieving satisfactory performance in Gaussian noise scenarios, the employed L2 norm is sensitive to the outliers of the observation prediction error $\Delta \mathbf{z}_{k+1|k}$.

To alleviate the drawback, we adopt the function $-\log(st(.;\upsilon,\sigma))$ in the loss function. Given N sequences with length T, the loss function can be written as

$$\mathcal{L}\left(\breve{\boldsymbol{\Theta}}\right) = \frac{1}{N\times T\times m}\sum_{j=1}^{N}\sum_{k=1}^{T}\sum_{i=1}^{m} -\log\left(st\left(\Delta \mathbf{z}^{i,j}_{k+1|k}\left(\widehat{\boldsymbol{\Theta}}\right);\upsilon,\sigma\right)\right) + \gamma_1\left\|\det\left(\hat{\mathbf{R}}\right)-1\right\|^2 + \gamma_2\left\|\breve{\boldsymbol{\Theta}}\right\|^2,$$

where $\Delta \mathbf{z}^{i,j}_{k+1|k}$ is the ith element of the observation error at time k in the jth trajectory. $\widehat{\boldsymbol{\Theta}}$ represents the RKFNet parameters, including the parameters of the employed FCNN and $\mathbf{R}_s$. Also, due to the difficulty of manually giving the parameters of the ST PDF, we set $\upsilon = \exp(\varsigma_2\upsilon')$ and $\sigma = \exp(\varsigma_3\sigma')$, where υ' and σ' are specified as neural network parameters, and $\varsigma_2 > 1$, $\varsigma_3 > 1$ are introduced to increase the loss gradients with respect to υ' and σ', respectively. Besides, if $\hat{\mathbf{R}}$ represents a reasonable estimate, its scaled matrix $\psi\hat{\mathbf{R}}$, where $\psi > 0$, remains reasonable, indicating the existence of an infinite set of solutions. To keep the consistency of the training results, $\left\|\det\left(\hat{\mathbf{R}}\right)-1\right\|^2$ with parameter γ_1 limits the matrix determinant $\det\left(\hat{\mathbf{R}}\right)$ close to 1. Furthermore, $\left\|\breve{\boldsymbol{\Theta}}\right\|^2$ is a penalty term with parameter γ_2, and $\breve{\boldsymbol{\Theta}} = \{\widehat{\boldsymbol{\Theta}}, \upsilon', \sigma'\}$.

Besides the ST-based loss function, we also introduce the USS method to stabilise the training process. The proposed RKFNet can be seen as a special RNN structure, where the temporally dependent information is delivered through Block I. Due to the error accumulation over time, the training process of the RKFNet is not stable, especially when the noise is highly heavy-tailed. Although the scheduled sampling technique [32] can improve the convergence stability of the sequence-to-sequence models, it cannot be directly employed in our framework due to its requirement for ground-truth data. This drawback can be overcome by our proposed USS training method, of which the structure is shown in Fig. 3. At any time k, the input of the RKFNet is chosen from either $\left[\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right]$ or the filtering result of a selected traditional RKF, $\left[\hat{\mathbf{x}}^*_{k-1|k-1}, \mathbf{P}^*_{k-1|k-1}\right]$, based on a coin toss. This increases the training stability but causes exposure bias simultaneously due to the differences between the training and inference processes. To reduce the bias, the probability of selecting the RKF estimates, p_t, decreases linearly, i.e.,

$$p_t = \max(p_{min}, p_{max} - \Delta p * t)$$

where t is the sequence iteration number and Δp represents the decreasing rate. Also, p_{max} and p_{min} are the maximum and minimum probability values of p_t, respectively.

5. Numerical simulations

5.1. The target tracking model and noises

This section introduces the employed target tracking models and measurement noises in the subsequent simulations. Sections 5.1.1 and 5.1.2 explain the constant-velocity (CV) model [33] and the Singer model [33,34], respectively. Also, three measurement noises are described in Section 5.1.3.

5.1.1. The constant-velocity model

The CV model aligns with the experimental setup in [18] and is specified as Eq. (1), where

$$\mathbf{F}_k = \begin{bmatrix}\mathbf{I}_2 & \Delta\, t\mathbf{I}_2\\ \mathbf{0} & \mathbf{I}_2\end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix}\mathbf{I}_2 & \mathbf{0}\end{bmatrix}, \tag{12}$$

where $\Delta\ t = 1$ is the observation interval. Also, the initial target state follows $\mathcal{N}(\mathbf{x}_{0|0}, \mathbf{P}_{0|0})$, where $\mathbf{x}_{0|0} = [0,0,10,10]^T$ and $\mathbf{P}_{0|0} = \mathrm{diag}([25,25,2,2])$. Furthermore, the covariance matrix of the zero-mean Gaussian process noise is given by

$$\mathbf{Q}_k = 0.1 * \begin{bmatrix}\frac{\Delta t^3}{3}\mathbf{I}_2 & \frac{\Delta t^2}{2}\mathbf{I}_2\\ \frac{\Delta t^2}{2}\mathbf{I}_2 & \Delta\, t\mathbf{I}_2\end{bmatrix}. \tag{13}$$

5.1.2. The Singer model

The Singer model is likewise defined by Eq. (1), and

$$\mathbf{F}_k = \mathrm{diag}\left(\mathbf{F}_{11}, \mathbf{F}_{22}\right), \quad \mathbf{H}_k = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix},$$

where

$$\mathbf{F}_{11} = \mathbf{F}_{22} = \begin{bmatrix}1 & \Delta\, t & \frac{1}{\rho^2}\left[-1+\rho\,\Delta\, t+\exp\left(-\rho\,\Delta\, t\right)\right]\\ 0 & 1 & \frac{1}{\rho}\left[1-\exp\left(-\rho\,\Delta\, t\right)\right]\\ 0 & 0 & \exp\left(-\rho\,\Delta\, t\right)\end{bmatrix}.$$

Following the setup of the CV model, we specify $\Delta\ t = 1$, $\rho = 0.2$, $\mathbf{x}_{0|0} = [0,10,0,0,10,0]^T$ and $\mathbf{P}_{0|0} = \mathrm{diag}([25,25,2,2,0.1,0.1])$. Also, for the zero-mean process noise, we have

$$\mathbf{Q}_k = \mathrm{diag}\left(\mathbf{Q}_{11}, \mathbf{Q}_{22}\right),$$

$$\mathbf{Q}_{11} = \mathbf{Q}_{22} = 2\rho\sigma_\rho^2\begin{bmatrix}q_{11} & q_{12} & q_{13}\\ q_{12} & q_{22} & q_{23}\\ q_{13} & q_{23} & q_{33}\end{bmatrix},$$

$$\begin{cases} q_{11} = \frac{1}{2\rho^5}\left[1-\exp(-2\rho\vartriangle t)+2\rho\vartriangle t+\frac{2\rho^3\vartriangle t^3}{3}-2\rho^2\vartriangle t^2-4\rho\vartriangle t\exp(-\rho\vartriangle t)\right] \\ q_{12} = \frac{1}{2\rho^4}\left[1+\exp(-2\rho\vartriangle t)-2\exp(-\rho\vartriangle t)+2\rho\vartriangle t\exp(-\rho\vartriangle t)-2\rho\vartriangle t+\rho^2\vartriangle t^2\right] \\ q_{13} = \frac{1}{2\rho^3}\left[1-\exp(-2\rho\vartriangle t)-2\rho\vartriangle t\exp(-\rho\vartriangle t)\right] \\ q_{22} = \frac{1}{2\rho^3}\left[4\exp(-\rho\vartriangle t)-3-\exp(-2\rho\vartriangle t)+2\rho\vartriangle t\right] \\ q_{23} = \frac{1}{2\rho^2}\left[\exp(-2\rho\vartriangle t)+1-2\exp(-\rho\vartriangle t)\right] \\ q_{33} = \frac{1}{2\rho}\left[1-\exp(-2\rho\vartriangle t)\right], \end{cases}$$

where $\sigma_\rho^2 = 0.5$.

5.1.3. Measurement noises

This work chooses three types of heavy-tailed measurement noises: Gaussian Mixture (GM), ST, and SGαS noises. The formulation of the GM noise can be written as follows:

$$\mathbf{GM} = \begin{cases} \mathcal{N}(\mathbf{0}, \overline{\mathbf{R}}) & \text{with probability} \quad 0.9 \\ \mathcal{N}(\mathbf{0}, U\overline{\mathbf{R}}) & \text{with probability} \quad 0.1, \end{cases} \tag{14}$$

where the nominal covariance matrix $\overline{\mathbf{R}} = 10\mathbf{I}_2$ and the augmentation factor U is selected from [5, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8]. By contrast, both the SGαS and ST noises utilise the scale matrix $\overline{\mathbf{R}}$, and their shape parameters are chosen from [0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.85] and [0.3, 0.5, 0.7, 0.9, 1.2, 1.7, 2.5, 3.5, 6], respectively.

5.2. Benchmark filters

The proposed RKFNet is compared with 3 kinds of heavy-tailed-distribution-based RKFs—the robust Kalman filter based on the slash distribution (RKF-SL) [18], RSTKF, and RKF-SGαS. We also evaluate the performance of the RKF with true shape parameters and scale matrices of the heavy-tailed noise (RKF-TSS) and the standard KF with true noise covariance matrices (KFTNCM) [16] for reference. These RKFs follow the same parameter set as in [18]. Specifically, for the RKF-SGαS, its GSIS-based variant (RKF-SGαS-GSIS) is employed with the particle number 100, the maximal series number $\Xi = 30$, the threshold for series convergence test $\varepsilon_1 = 10^{-2}$ and the number of latest items for series convergence test $\tau_1 = 4$. Besides, for the fixed-point iteration of all the RKFs, we set the maximum number of iterations $M = 50$, the threshold for the convergence test $\varepsilon_2 = 10^{-2}$ and the number of the latest items for convergence test $\tau_2 = 4$. Additionally, the parameters of the heavy-tailed distributions employed in the RKFs, including the shape parameters and the scale matrices, are estimated by existing expectation maximisation and maximum likelihood estimation methods (the detailed set is the same as in the section IV-C in [18]).

Except for the traditional RKFs, the benchmark filters also include three model-based RNNs [30], which employ one-layer vanilla RNN, GRU and LSTM units with hidden size 63, 36 and 31, respectively. Hence, their parameter numbers are roughly equal. As shown in Fig. 4, the model-based RNNs imitate the KF operation by first recovering $\hat{\mathbf{x}}_{k|k-1}$ using domain knowledge, i.e., via (9). Also, $\Delta\hat{\mathbf{z}}_{k|k-1}$ is selected as the input feature, and an increment $\Delta\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}$ is estimated by an RNN unit.

For the RKFNet, Block II employs a 3-layer FCNN with hidden-layer sizes of 32, 64, and 32, resulting in the parameter number approximately equal to that of the model-based RNNs. Also, the LeakyReLU function, characterised by a negative slope parameter of 0.1, is applied across all hidden layers. Furthermore, the elements of $\mathbf{R}_s$ are initially drawn from a uniform distribution $U[0, \frac{1}{\varsigma_1}]$, and $\varsigma_1 = 300$. Besides, for the loss function, $\varsigma_2 = \varsigma_3 = 300$, $\gamma_1 = 0.1$ and $\gamma_2 = 0.0001$.

5.3. Training dataset and optimiser

In each tracking scenario, the datasets are composed of 3200 trajectories for training, 200 for cross-validation, and 200 for testing, with a fixed length of $T = 100$ for the CV model and $T = 81$ for the Singer model. Also, there are 2000 iterations in every training process, and we adopt the Adam optimiser, where the learning rate starts from 0.0002 and is halved every 400 iterations. Besides, for the USS, the training batch size is 200, $p_{max} = 1$ and $\Delta p = \frac{1}{600}$.

5.4. Performance evaluation of RKFNet

In this section, under the CV model, we investigate the RKFNet filtering performance across 3 factors: p_{min}, the ST-based loss function and the reference trajectories produced by the traditional RKFs.

(1) Within the context of the target-tracking model, we evaluate the impact of p_{min}, ranging from 0.0 to 1.0 in the fixed increment of 0.1. Also, the SGαS measurement noise is considered, and the shape parameter is selected from [0.3, 0.7, 1.1, 1.5, 1.85]. Besides, the filtering results of the RKF-SGαS-GSIS are employed for the USS training method. To analyse the stability and consistency of the training results, 5 Monte Carlo experiments are run. As shown in Fig. 5, due to the exposure errors, large p_{min} can cause inaccurate and unstable filtering results, especially in the highly heavy-tailed noise scenario ($\alpha = 0.3$). By contrast, for small p_{min}, the RKFNet can produce more precise and reliable estimation.

In the following simulations, p_{min} is set as 0, and the RKFNet is always trained through 5 independent Monte Carlo run. Also, The model corresponding to the best cross-validation results is selected as the final RKFNet model, which is then evaluated using the testing dataset.

(2) The ST-based loss function is compared with the classical L1 and L2 loss functions under the three measurement noises introduced in Section 5.1. For light-tailed noise, the estimation errors based on the loss functions are comparable as depicted in Fig. 6. However, when the noise contains heavy outliers, the ST-based loss function yields the most accurate estimation due to its lower sensitivity to outliers. Also, Fig. 7 shows that the RKFNet based on the ST loss can achieve the highest convergence rate, whereas the L1/L2 losses suffer from the overfitting problem as their corresponding training results are better than the cross-validation results.

(3) Based on the filtering results of three RKFs, the influence of reference trajectories is assessed under the three types of heavy-tailed noises. Fig. 8 shows that the RKFNets trained based on different reference sequences achieve similar estimation results in all scenarios. This implies that the RKFNet has no requirement for highly precise reference sequences, which are only employed to stabilise the convergence during the initial stage of the training process. When the p_{min} gradually reduces towards zero, the RKFNet can converge without the assistance of the RKF estimates.

5.5. RKFNet vs. Benchmark filters

In this experiment, our proposed RKFNet is compared with the benchmark filters introduced in Section 5.2. To train the RKFNet and the model-based RNNs, the RKF-SGαS is employed to produce the reference sequences. Also, we train both the RKFNet and the model-based RNNs 5 times, and the final model achieves the best cross-validation results.

From Fig. 9, when the noise is light-tailed, the RKFNets and the other benchmark filters achieve similar filtering results. Nevertheless, the comparison under the heavy-tailed noise scenarios is more complicated. (1) Compared with traditional RKFs, the RKFNet can produce more precise results as it has no reliance on precise noise models. (2) The advantages of the RKFNets over the RKFs become more pronounced under the GM noise. The mixing density of the GM distribution follows a delta mixing distribution, which cannot be accurately approximated by the fully skewed mixing densities employed by the traditional RKFs. (4) Despite the true model parameters, the RKF-TSS performs worse than the RKFNet, indicating that the marginal-approximation-based VB method yields less accurate estimation than

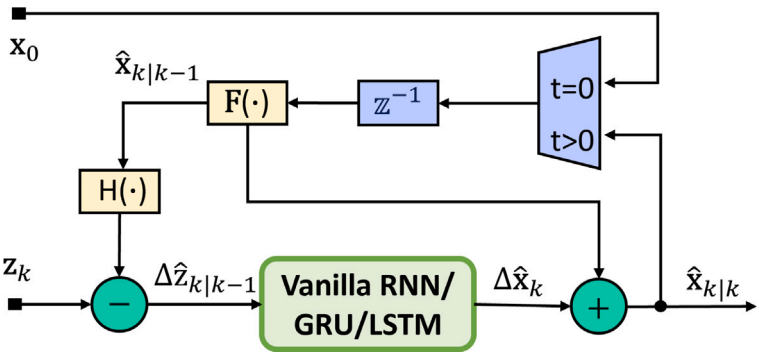

Fig. 4. The structure of the model-based RNNs.

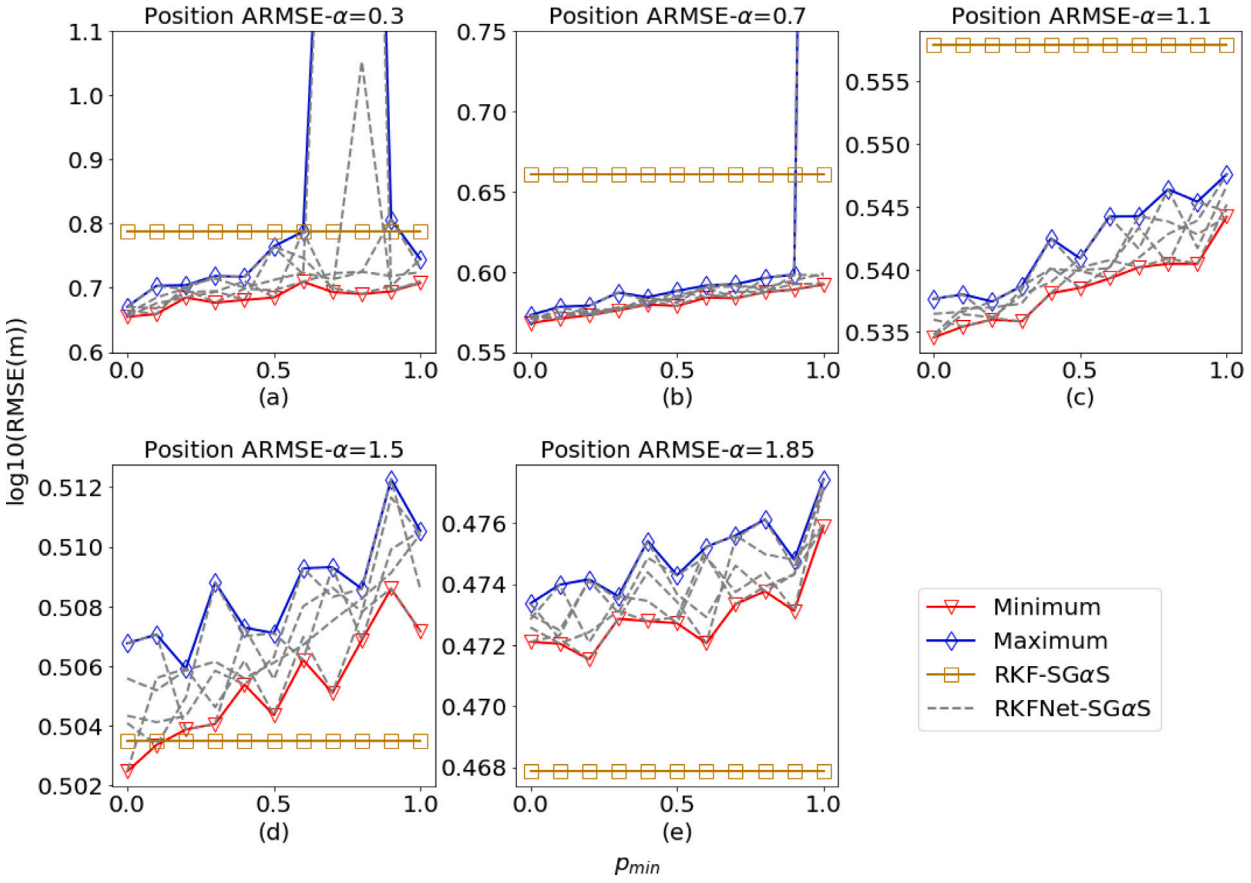

Fig. 5. The tracking results of the RKFNet under the SGαS measurement noise and the CV model. In every subplot, the title is the shape parameter α of the SGαS distribution, the x-axis represents p_{min}, and the y-axis is the position estimation log10(ARMSE(m)). Also, the "RKFNet-SGαS" represents the 5 independent Monte Carlo results of the RKFNet, and the corresponding maximal and minimal errors are depicted by "Maximum" and "Minimum", respectively. Besides, the filtering performance of the RKF-SGαS-GSIS is shown as a reference.

the neural-network-based method introduced in this study. (5) Compared with the model-based RNNs, the RKFNet can provide better filtering results under heavy-tailed noise. Integrating the traditional RKF framework, the RKFNet diminishes the necessary neural network parameter number and achieves greater interpretability.

Fig. 10 illustrates the estimation precision of the $\mathbf{R}_k$. Considering the heavy-tailed measurement noise, we evaluate the estimation error by $\log(|\tilde{\mathbf{R}}_k| - |\mathbf{R}_k|)$. Under the GM noise, the RKFNet performs better than the other benchmark methods. However, all the methods produce comparable estimation errors under the ST and SGαS noises. This indicates that the filtering performance is susceptible to the estimation of $\mathbf{R}_k$ and even a slight discrepancy in its estimation can lead to significant variations in performance.

The training processes of the RKFNet and the model-based RNNs under SGαS noise with $\alpha = 0.7$ are shown in Fig. 11. The RKFNet presents a higher convergence rate and lower estimation errors than the model-based RNNs.

Fig. 12 presents the filtering error dynamics of the position estimation under the SGαS noise with characteristic exponent $\alpha = 0.7$. All the filters produce stable estimation results over time, and the RKFNet achieves the best performance.

Fig. 13 illustrates the execution time of the RKFNet and the benchmark filters based on the same CPU and GPU devices under different heavy-tailed noises. Compared with RKF-SL and RSTKF, the RKFNet is more efficient as the fix-point iterations of the RKFs require more

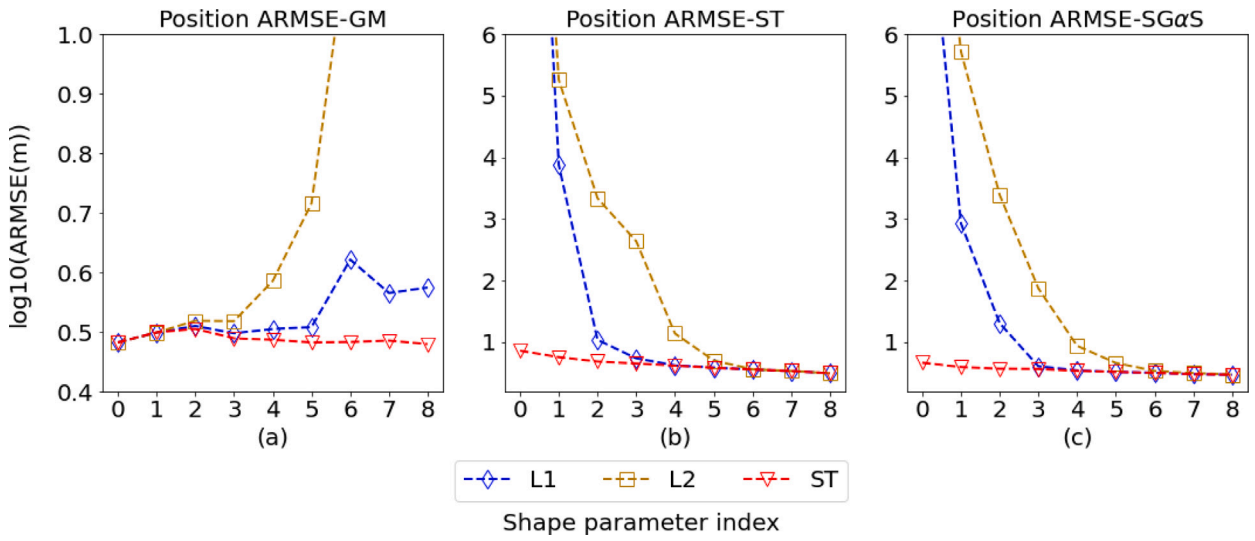

Fig. 6. The filtering results of the RKFNet with different loss functions under the CV model. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation log10(RMSE(m)). (a) plots the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b) and (c), respectively.

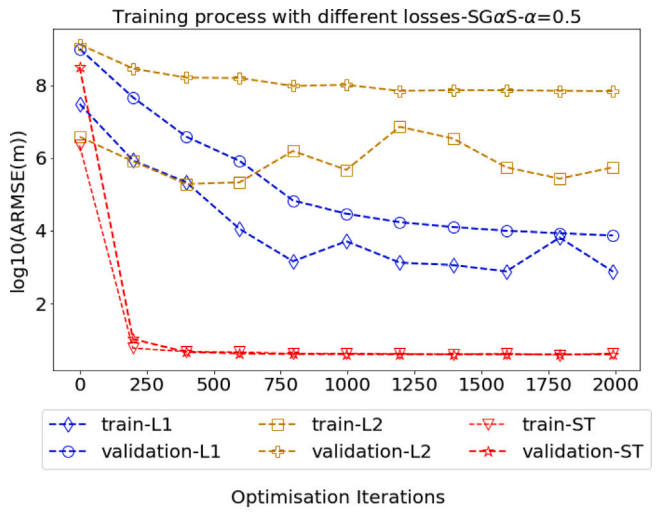

Fig. 7. The learning curves of the RKFNet with different loss functions under SGαS noise ($\alpha = 0.5$) and the CV model. The x-axis is the optimisation iterations, and the y-axis represents the position estimation log10(ARMSE(m)).

execution time. Also, as the RKF-SGαS is more computationally expensive than the other traditional RKFs [18], its performance is not shown here. Besides, the execution time of the RKFNet is higher than model-based RNNs as the KF update in Block I consists of serial operations and decreases the computational efficiency.

6. Conclusion

In this work, we presented a DM RKF framework, RKFNet, combining the traditional heavy-tailed-distribution-based RKFs with the deep-learning technique. Specifically, the mixing-parameter-based function and the scale matrix are estimated by an FCNN and an introduced neural network parameter, respectively, and then a KF update produces the posterior state estimates. Also, the USS training method is proposed to improve the stability of the training process.

Two sets of experiments under three kinds of heavy-tailed noises are implemented to validate the proposed framework. The first simulation shows the advantages of the ST-based loss function over L1/L2 loss functions. Also, the performance of the RKFnet does not hinge on accurate reference sequences from the traditional RKF filtering results. By contrast, the second experiment set conducts a comparison between the RKFNet and the various benchmark filters. The experimental results show that in the light-tailed noise scenarios, all the filters produce similar results. However, the RKFNet achieves the best performance under heavy-tailed noises.

In the future, we will consider applying the proposed framework to more complicated scenarios. In this work, we focus on linear models with symmetric heavy-tailed measurement noise. However, an extension to linear models with skewed heavy-tailed signal and measurement noises can be considered, which can benefit more practical applications.

CRediT authorship contribution statement

Pengcheng Hao: Writing – original draft, Methodology. **Oktay Karakuş:** Writing – review & editing, Supervision. **Alin Achim:** Writing – review & editing, Supervision.

Code availability

All the code needed to reproduce the results presented in this work are available at: https://github.com/PengchengH/RKFNet.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Dr Pengcheng Hao's work was supported by the China Scholarship Council (CSC).

Data availability

Data will be made available on request.

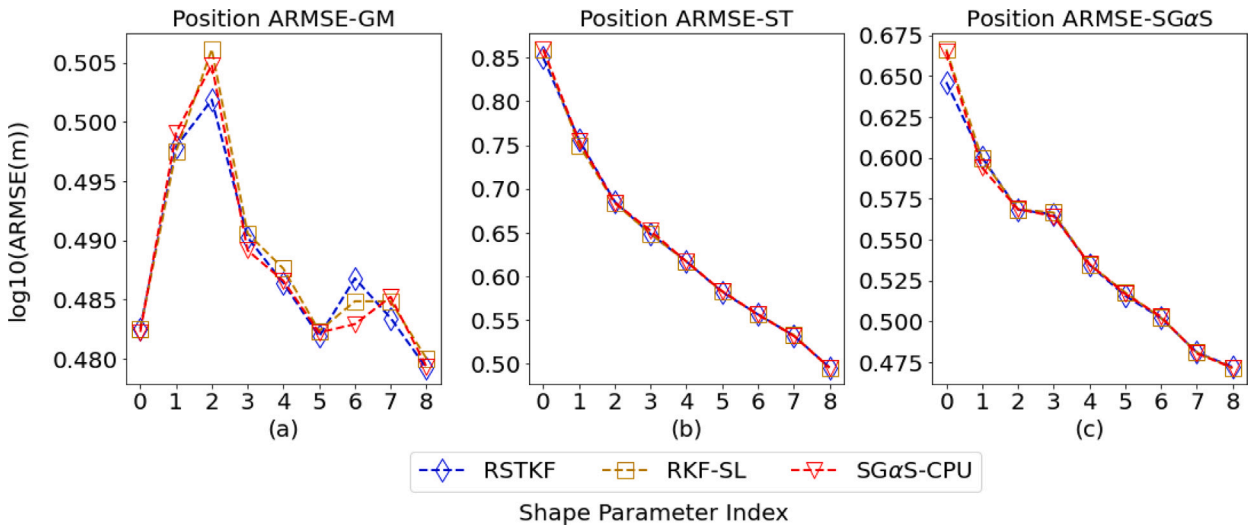

Fig. 8. The filtering performance comparison among the RKFNets trained based on different traditional RKFs under the CV model. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation log10(ARMSE(m)). (a) plots the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b) and (c), respectively.

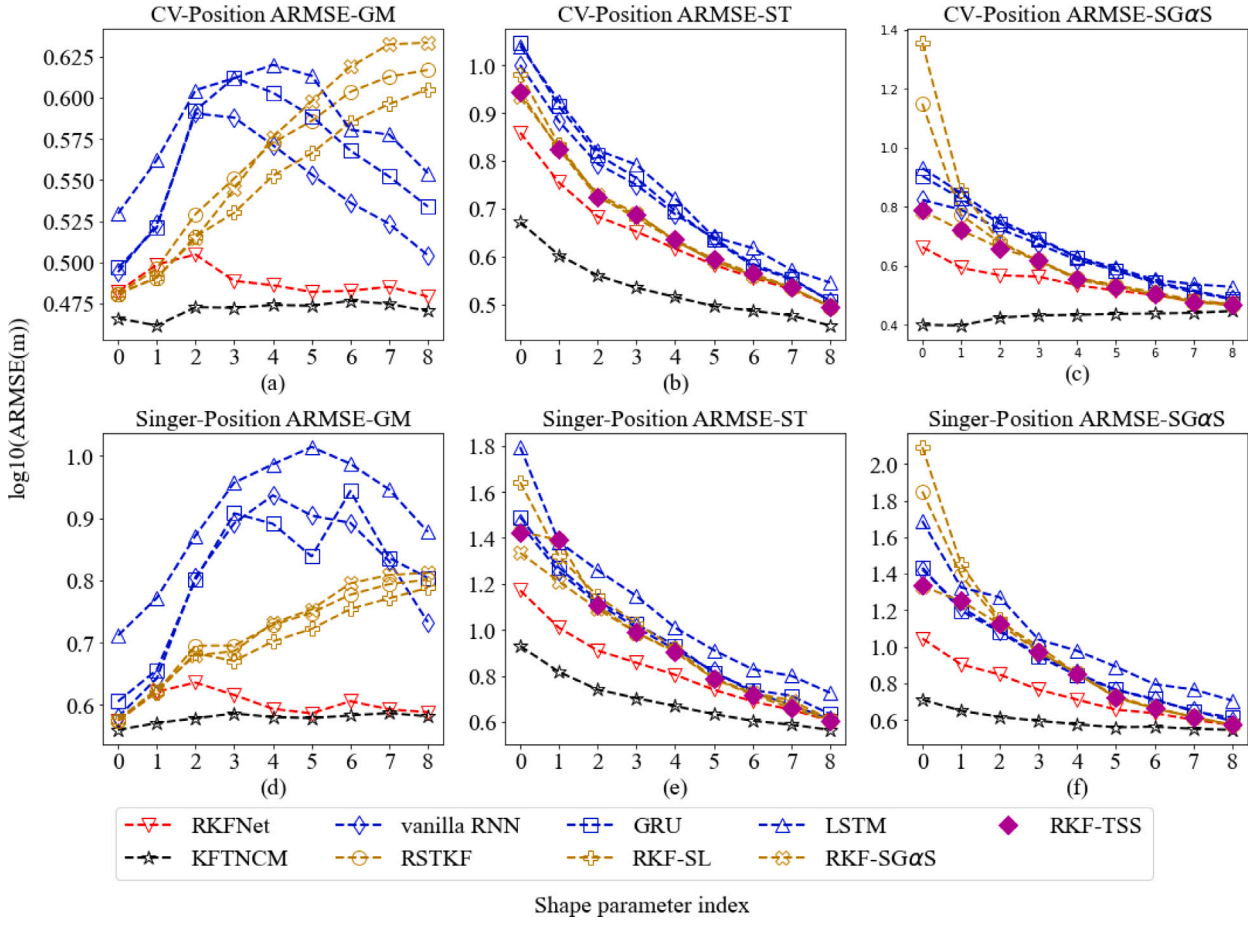

Fig. 9. The filtering performance comparison between the RKFNet and the benchmark filters under the CV and the Singer models. The titles of the subplots consist of the tracking models and the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation log10(ARMSE(m)). (a) and (d) plot the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b, e) and (c, f), respectively.

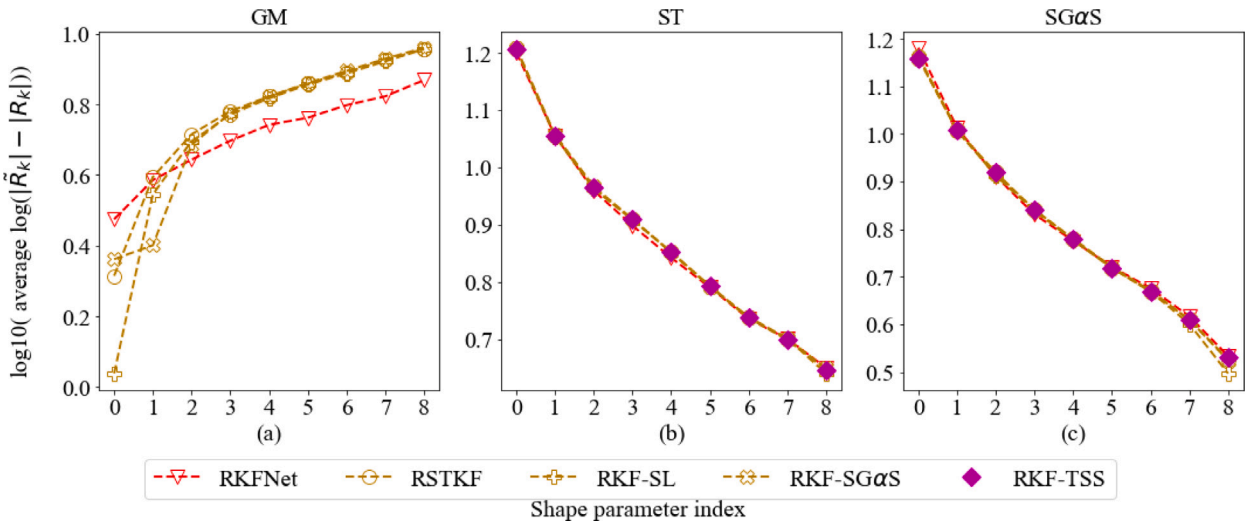

Fig. 10. The $\mathbf{R}_k$ estimation comparison between the RKFNet and the traditional RKFs under the Singer model. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the average log10(estimation error). (a) plots the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b) and (c), respectively.

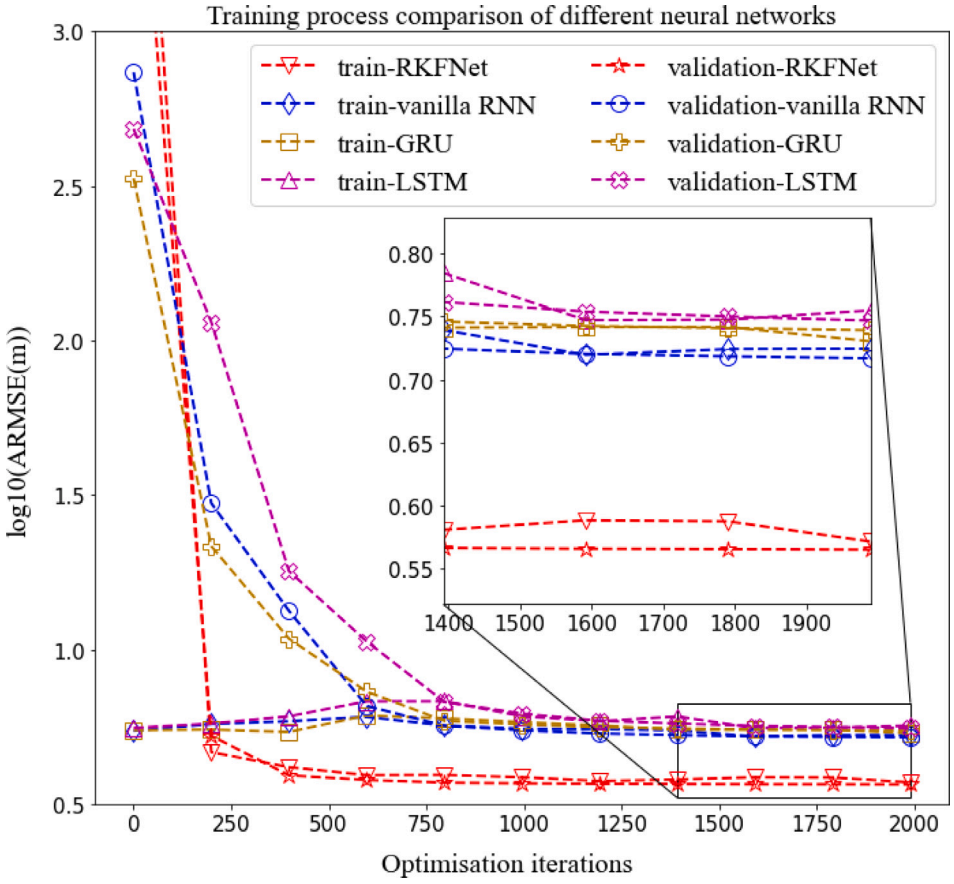

Fig. 11. The learning curves of the RKFNet and 3 types model-based RNNs under SGαS noise ($\alpha = 0.7$) and the CV model. The x-axis is the optimisation iterations, and the y-axis represents the position estimation log10(ARMSE(m)).

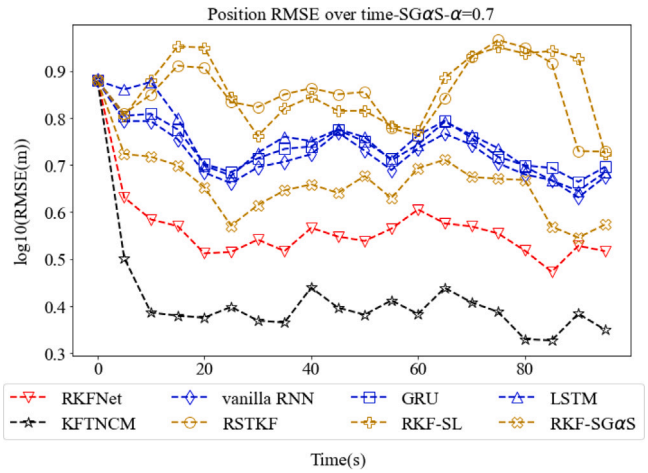

Fig. 12. The position estimation error dynamics over time under SGαS noise (α = 0.7) and the CV model. The x-axis is the tracking time, and the y-axis represents the position estimation log10(RMSE(m)).

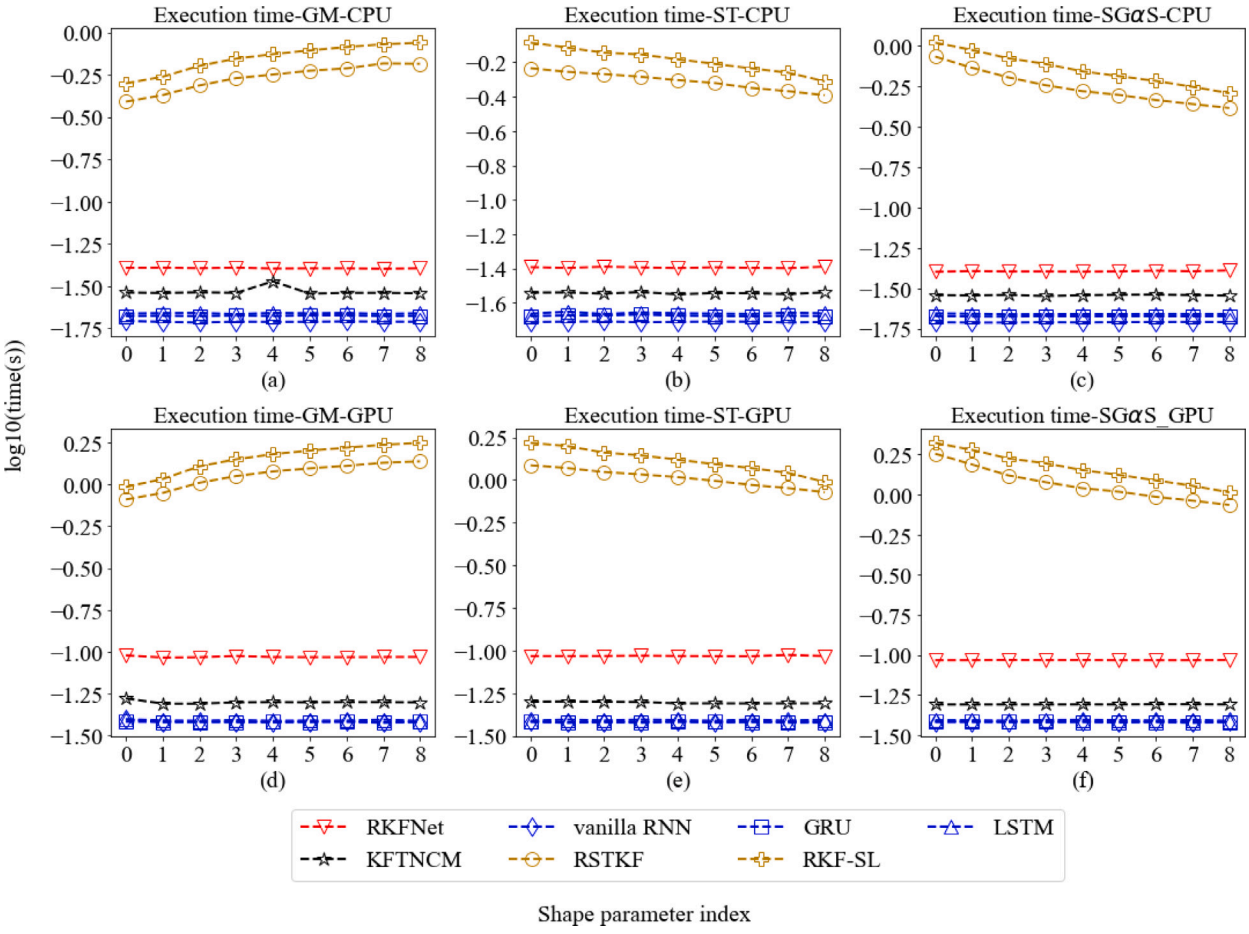

Fig. 13. The filtering efficiency comparison between the RKFNet and the benchmark filters under the CV model. The titles of the subplots contain the distributions of the heavy-tailed measurement noise and the execution device. The x-axis is the shape parameter index, and the y-axis represents the execution time log10(time(s)).

References

[1] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME–J. Basic Eng. 82 (1) (1960) 35–45, http://dx.doi.org/10.1115/1.3662552.
[2] R. Zhan, J. Wan, Iterated unscented Kalman filter for passive target tracking, IEEE Trans. Aerosp. Electron. Syst. 43 (3) (2007) 1155–1163.
[3] M.R. Barassi, G.M. Caporale, S.G. Hall, Interest rate linkages: a Kalman filter approach to detecting structural change, Econ. Model. 22 (2) (2005) 253–284.
[4] L.-H. Chen, Inflation and real short-term interest rates-a Kalman filter analysis of the term structure, Appl. Econ. 33 (7) (2001) 855–861.
[5] D.-J. Jwo, S.-H. Wang, Adaptive fuzzy strong tracking extended Kalman filtering for GPS navigation, IEEE Sens. J. 7 (5) (2007) 778–789, http://dx.doi.org/10.1109/JSEN.2007.894148.
[6] C.D. Karlgaard, H. Schaub, Huber-based divided difference filtering, J. Guid. Control Dyn. 30 (3) (2007) 885–891, http://dx.doi.org/10.2514/1.27968.
[7] W. Wan, J. Feng, B. Song, X. Li, Huber-based robust unscented Kalman filter distributed drive Electric Vehicle State Observation, Energies 14 (3) (2021) 750, http://dx.doi.org/10.3390/en14030750.
[8] R. Izanloo, S.A. Fakoorian, H.S. Yazdi, D. Simon, Kalman filtering based on the maximum correntropy criterion in the presence of non-Gaussian Noise,

in: 2016 Annual Conference on Information Science and Systems, CISS, 2016, http://dx.doi.org/10.1109/ciss.2016.7460553.
[9] G. Wang, Y. Zhang, X. Wang, Iterated maximum correntropy unscented Kalman filters for non-Gaussian Systems, Signal Process. 163 (2019) 87–94, http://dx.doi.org/10.1016/j.sigpro.2019.05.015.
[10] Y. Huang, F. Zhu, Y. Zhang, Y. Zhao, P. Shi, J. Chambers, 4 outlier-robust Kalman filtering framework based on statistical similarity measure, in: Outliers in Control Engineering, 2022, pp. 61–98, http://dx.doi.org/10.1515/9783110729122-004.
[11] G. Gao, S. Gao, G. Hong, X. Peng, T. Yu, A robust INS/SRS/CNS integrated navigation system with the chi-square test-based robust Kalman filter, Sensors 20 (20) (2020) 5909.
[12] H. Li, D. Medina, J. Vilà-Valls, P. Closas, Robust variational-based Kalman filter for outlier rejection with correlated measurements, IEEE Trans. Signal Process. 69 (2020) 357–369.
[13] Y. Huang, Y. Zhang, N. Li, Z. Wu, J.A. Chambers, A novel robust Student's t-based Kalman filter, IEEE Trans. Aerosp. Electron. Syst. 53 (3) (2017) 1545–1554, http://dx.doi.org/10.1109/taes.2017.2651684.
[14] D.G. Tzikas, A.C. Likas, N.P. Galatsanos, The variational approximation for Bayesian inference, IEEE Signal Process. Mag. 25 (6) (2008) 131–146, http://dx.doi.org/10.1109/msp.2008.929620.
[15] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, J.A. Chambers, Robust Kalman filters based on Gaussian scale mixture distributions with application to target tracking, IEEE Trans. Syst. Man Cybern.: Syst. 49 (10) (2019) 2082–2096, http://dx.doi.org/10.1109/tsmc.2017.2778269.
[16] C. Xue, Y. Huang, F. Zhu, Y. Zhang, J.A. Chambers, An outlier-robust Kalman filter with adaptive selection of elliptically contoured distributions, IEEE Trans. Signal Process. 70 (2022) 994–1009, http://dx.doi.org/10.1109/tsp.2022.3151199.
[17] G. Samorodnitsky, M.S. Taqqu, Stable non-Gaussian random processes, 2017, http://dx.doi.org/10.1201/9780203738818.
[18] P. Hao, O. Karakuş, A. Achim, Robust Kalman filters based on the sub-Gaussian α-stable distribution, Signal Process. 224 (2024) 109574.
[19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.
[20] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, S. Valaee, Recent advances in recurrent neural networks, 2018, URL https://arxiv.org/abs/1801.01078.
[21] A. Graves, A. Graves, Long short-term memory, in: Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012, pp. 37–45.
[22] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D 404 (2020) 132306.
[23] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems, MWSCAS, IEEE, 2017, pp. 1597–1600.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[25] C. Gao, J. Yan, S. Zhou, B. Chen, H. Liu, Long short-term memory-based recurrent neural networks for nonlinear target tracking, Signal Process. 164 (2019) 67–73, http://dx.doi.org/10.1016/j.sigpro.2019.05.027.
[26] J. Chen, H. Jing, Y. Chang, Q. Liu, Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process, Reliab. Eng. Syst. Saf. 185 (2019) 372–382, http://dx.doi.org/10.1016/j.ress.2019.01.006.
[27] G. Zhao, Z. Wang, Y. Huang, H. Zhang, X. Ma, Transformer-based maneuvering target tracking, Sensors 22 (21) (2022) 8482.
[28] Y. Liu, J. Cheng, H. Zhang, H. Zou, N. Xiong, Long short-term memory networks based on particle filter for object tracking, IEEE Access 8 (2020) 216245–216258, http://dx.doi.org/10.1109/access.2020.3041294.
[29] H. Coskun, F. Achilles, R. DiPietro, N. Navab, F. Tombari, Long short-term memory Kalman filters: Recurrent neural estimators for pose regularization, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, http://dx.doi.org/10.1109/iccv.2017.589.
[30] G. Revach, N. Shlezinger, X. Ni, A.L. Escoriza, R.J. van Sloun, Y.C. Eldar, Kalmannet: Neural network aided Kalman filtering for partially known dynamics, IEEE Trans. Signal Process. 70 (2022) 1532–1547, http://dx.doi.org/10.1109/tsp.2022.3158588.
[31] G. Revach, N. Shlezinger, T. Locher, X. Ni, R.J. van Sloun, Y.C. Eldar, Unsupervised learned Kalman filtering, in: 2022 30th European Signal Processing Conference, EUSIPCO, 2022, http://dx.doi.org/10.23919/eusipco55093.2022.9909801.
[32] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled sampling for sequence prediction with recurrent neural networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[33] X.R. Li, V.P. Jilkov, Survey of maneuvering target tracking. Part I. Dynamic models, IEEE Trans. Aerosp. Electron. Syst. 39 (4) (2003) 1333–1364.
[34] S. Jia, Y. Zhang, G. Wang, Highly maneuvering target tracking using multi-parameter fusion Singer model, J. Syst. Eng. Electron. 28 (5) (2017) 841–850.