



Brief paper

Universal approximation capacity of bidirectional recurrent neural networks with noisy sequential inputs[☆]

Yangtiance Tao^{a,1}, Jiayi Kang^{b,1}, Xiuqiong Chen^{c,*}, Stephen Shing-Toung Yau^{a,*}

^a Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China

^b Hetao Institute of Mathematics and Interdisciplinary Sciences (HIMIS), Shenzhen, Guangdong, 518045, China

^c School of Mathematics, Renmin University of China, Beijing 100872, China

ARTICLE INFO

Article history:

Received 6 February 2025

Received in revised form 1 April 2026

Accepted 27 July 2026

Available online 22 September 2026

Keywords:

State-space model

Bidirectional recurrent neural networks

Kalman smoother

Approximation capacity

ABSTRACT

In this paper, we prove that, under certain reasonable assumptions, bidirectional recurrent neural networks (BRNNs) can well approximate a class of bidirectional dynamical systems. As a special case, we prove that the forward recursions of BRNNs are capable of accurately approximating any filtering dynamics, while the bidirectional recursions of BRNNs satisfactorily approximate the corresponding smoothing dynamics. We present explicit estimation bounds and utilize these theoretical results to construct a BRNN-based smoother, which offers a synthetic resolution to the smoothing problem. By training a BRNN on sequences of observations and their corresponding states produced by computer simulations or actual experiments, we demonstrate that the BRNN-based smoother can closely approximate the optimal smoother, including approximating the optimal filter as its forward recursion. Finally, we support our theoretical findings through a numerical example and compare our results numerically against the extended Kalman smoother and particle smoother.

© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

1. Introduction

Recurrent neural networks (RNNs) with various structures, including vanilla RNNs (Elman, 1990), bidirectional RNNs (BRNNs) (Schuster & Paliwal, 1997), long short-term memory (LSTM) networks (Hochreiter & Schmidhuber, 1997), gate recurrent unit (GRU) networks (Chung et al., 2014), etc (Yu et al., 2019), have been widely applied in sequence modeling (Chiu et al., 2018; Sutskever et al., 2011), image generation (Gregor et al., 2015; Van Den Oord et al., 2016), natural language processing (Goldberg, 2016; Mikolov et al., 2010), and reinforcement learning (Perusquia & Yu, 2021; Santoro et al., 2018). Despite their success, theoretical aspects require further investigation. Schäfer and Zimmermann (2006) proved the universal approximation theorem for RNNs based on the Feedforward neural networks (FNNs) result by Hornik et al. (1989). Recent work by Chen et al. (2023), Tao et al. (2024) extended these results to stochastic settings. However,

all existing work focuses on unidirectional models, leaving the approximation capacity of BRNNs unexplored.

In this work, we rigorously prove that under reasonable assumptions, BRNNs can approximate a class of bidirectional dynamical systems. Furthermore, we find that this result can be used in smoothing problems for the state-space models (SSMs).

We consider the discrete-time SSM:

$$\begin{aligned} x_t &= f(x_{t-1}) + w_t, \\ y_t &= h(x_t) + v_t, \end{aligned} \quad (1)$$

where $x_t \in \mathbb{R}^n$ is the state, $y_t \in \mathbb{R}^m$ is the observation, f and h are potentially nonlinear functions, time $t = 1, 2, \dots, T$ with fixed $T \in \mathbb{N}^+$, and w_t, v_t are zero-mean Gaussian noises with covariances $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$. Let $y_{1:t} := \{y_1, y_2, \dots, y_t\}$. Filtering estimates $p(x_t | y_{1:t})$ via forward pass, while smoothing estimates $p(x_t | y_{1:T})$ via backward pass after filtering, analogous to BRNN's bidirectional structure.

Current smoothing algorithms rely on model-based methodologies such as the Rauch–Tung–Striebel smoother (Rauch et al., 1965) and particle smoother (PS) (Carvalho et al., 2010; Klaas et al., 2006). However, these methods have limitations in data-only scenarios and nonlinear SSMs. Model-based methods require exact knowledge of the underlying SSMs, restricting their applicability when this information is unavailable. Nonlinear variants like the extended Kalman smoother (EKS) are not minimum mean-squared error (MMSE) optimal (Jazwinski, 2007), while

[☆] This work is supported by the National Natural Science Foundation of China (No. 42450242 and 12201631), and Tsinghua University Education Foundation. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dario Piga under the direction of Editor Alessandro Chiuso.

* Corresponding authors.

E-mail addresses: tytz19@tsinghua.org.cn (Y. Tao), kangjiayi@himis-sz.cn (J. Kang), cxq0828@ruc.edu.cn (X. Chen), yau@uic.edu (S.S.-T. Yau).

¹ Yangtiance Tao and Jiayi Kang contributed equally to this work.

PS algorithms require large particle numbers for highly non-linear systems, making them computationally expensive. Data-driven smoothing algorithms based on the Kalman framework have been developed (Ni et al., 2022). In scenarios where explicit SSM knowledge is unavailable but historical observation-state data exist, synthetic approaches leveraging this data become necessary.

In this work, motivated by the similar framework of BRNN and smoothing, we develop a BRNN-based smoother trained on simulation or experimental data using backpropagation through time (Werbos, 1990) to minimize mean-squared error between states and estimates. The forward recursion yields an RNN-based filter (Chen et al., 2023), extending the neural filter concept (Lo, 1994).

The key contributions of this paper are summarized as follows:

- We establish a theoretical result showing that BRNNs can effectively approximate a broad class of bidirectional dynamical systems. As a special case, we prove that the forward and backward recursions of BRNNs can respectively approximate the filtering and smoothing dynamics, with explicit error bounds. This gives the first quantitative analysis of the approximation capability of BRNNs for noisy sequential data.
- We develop the BRNN-based smoother for SSMs, integrating BRNN forward recursions into an RNN-based filter. Numerical experiments demonstrate superior performance over EKS and PS in nonlinear cases.

The paper is organized as follows. In Section 2, we review SSM estimation problems, including filtering and smoothing, as well as relevant neural network architectures such as FNNs, RNNs, and BRNNs. Section 3 presents the RNN-based filter and introduces the architecture of the BRNN-based smoother. In Section 4, we analyze the approximation capabilities of BRNNs and quantify the corresponding error in the BRNN-based smoother. Section 5 provides numerical experiments comparing the BRNN-based smoother with EKS and PS. Finally, Section 6 concludes the paper and outlines directions for future research.

Notations: We use $\cdot^\top$ for transpose, and $\mathbb{E}[\cdot]$ for expectation. The probability space with filtration $\mathcal{F}_t$ is $(\Omega, \mathcal{F}_t, \mathbb{P})$. Gaussian distribution with mean μ and covariance P is $\mathcal{N}(\mu, P)$. $\mathbb{R}$ ($\mathbb{Z}$) denotes real (integer) numbers; $\mathbb{R}^n$ and $\mathbb{R}^{m \times n}$ denote n -dimensional Euclidean space and $m \times n$ real matrices, respectively. We denote by $L^1(\mathbb{R}^n)$, $L^2(\mathbb{R}^n)$, and $L^\infty(\mathbb{R}^n)$ the spaces of integrable, square-integrable, and essentially bounded functions on $\mathbb{R}^n$, with norms $\|\cdot\|_p$ for $p = 1, 2, \infty$. The symbol $|\cdot|$ denotes the Euclidean norm.

2. Preliminaries

This section introduces the filtering and smoothing problems for SSMs and their optimization formulations, followed by brief descriptions of FNNs, RNN and BRNN.

2.1. Filtering and smoothing

We focus on estimating the expectations of test functions φ under the filtering and smoothing distributions, denoted by $\hat{\varphi}_{t|t} := \mathbb{E}[\varphi(x_t) | y_{1:t}]$ and $\hat{\varphi}_t := \mathbb{E}[\varphi(x_t) | y_{1:T}]$, respectively, where $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$ is a test function. Let $L^2_{y_{1:t}}(\Omega; \mathbb{R}^r)$ denote the family of $\mathbb{R}^r$ -valued $\mathcal{F}_t$ -measurable random variables ξ with $\mathbb{E}[\|\xi\|_2^2] < \infty$, where $\mathcal{F}_t$ is σ algebra generated by $y_{1:t}$. The filtering problem is formulated as:

$$\hat{\varphi}_{t|t} = \arg \min_{\psi_t \in L^2_{y_{1:t}}(\Omega; \mathbb{R}^r)} \mathbb{E}[\|\psi_t - \varphi(x_t)\|_2^2], \quad (2)$$

and the smoothing problem as:

$$\hat{\varphi}_t = \arg \min_{\psi_t \in L^2_{y_{1:T}}(\Omega; \mathbb{R}^r)} \mathbb{E}[\|\psi_t - \varphi(x_t)\|_2^2]. \quad (3)$$

2.2. Neural networks

We first introduce the FNNs.

Definition 2.1 (FNNs). Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be an activation function. Define

$$\Sigma^{n_1, n_2}(\sigma) = \left\{ NN : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_2} \mid NN(x) = V\sigma(Wx - \theta) \right\},$$

where $V \in \mathbb{R}^{n_2 \times n_3}$, $W \in \mathbb{R}^{n_3 \times n_1}$, $\theta \in \mathbb{R}^{n_3}$ and σ is applied component-wise.

A vanilla RNN processes sequential data with the following structure.

Definition 2.2 (RNN). For $n_1, n_2, n_3 \in \mathbb{N}^+$ and activation function σ , the class $\text{RNN}^{n_1, n_2, n_3}(\sigma)$ consists of functions defined by

$$\begin{aligned} \vec{s}_t &= \sigma \left(W_s^{(f)} \vec{s}_{t-1} + W_\alpha^{(f)} \alpha_t + b_h^{(f)} \right), \\ \beta_t &= W_{hq}^{(f)} \vec{s}_t + b_q^{(f)}, \end{aligned}$$

where $\alpha_t \in \mathbb{R}^{n_1}$ is the input, $\vec{s}_t \in \mathbb{R}^{n_2}$ is the hidden state, $\beta_t \in \mathbb{R}^{n_3}$ is the output, and the weights $W_s^{(f)}$, $W_\alpha^{(f)}$, $W_{hq}^{(f)}$ together with the biases $b_h^{(f)}$, $b_q^{(f)}$ are the model parameters.

A BRNN extends the vanilla RNN with forward and backward hidden states.

Definition 2.3 (BRNN). For $n_1, n_2, n_3 \in \mathbb{N}^+$ and activation σ , the class $\text{BRNN}^{n_1, n_2, n_3}(\sigma)$ consists of functions defined by

$$\begin{aligned} \vec{s}_t &= V^{(f)} \sigma(W_s^{(f)} \vec{s}_{t-1} + W_\alpha^{(f)} \alpha_t + b^{(f)}), \\ \overleftarrow{s}_t &= V^{(b)} \sigma(W_s^{(b)} \overleftarrow{s}_{t+1} + W_\alpha^{(b)} \alpha_t + b^{(b)}), \\ s_t &= [\vec{s}_t; \overleftarrow{s}_t], \quad \beta_t = M_\beta \sigma(W_\beta s_t + b_\beta), \end{aligned} \quad (4)$$

where $\alpha_t \in \mathbb{R}^{n_1}$ is the input, $\vec{s}_t, \overleftarrow{s}_t \in \mathbb{R}^{n_2}$ are the forward and backward hidden states, $\beta_t \in \mathbb{R}^{n_3}$ is the output, and the weights $W_s^{(f)}$, $W_\alpha^{(f)}$, $V^{(f)}$, $W_s^{(b)}$, $W_\alpha^{(b)}$, $V^{(b)}$, W_β , M_β together with the biases $b^{(f)}$, $b^{(b)}$, b_β are the model parameters.

3. RNN-based filter and BRNN-based smoother

In this section, we introduce the architecture of the BRNN-based smoother, which incorporates an RNN-based filter (Chen et al., 2023) as its forward recursion.

3.1. RNN-Based filter

We begin with the RNN-based filter, which consists of a dynamical encoder and an output decoder. Let us denote the dynamical encoder and the output decoder by **Forward** – **Encoder** ($\cdot; \mathbf{w}_1$) and **F** – **Decoder** ($\cdot; \mathbf{w}_2$), respectively. For brevity, we denote them as **FE** ($\cdot; \mathbf{w}_1$), and **FD** ($\cdot; \mathbf{w}_2$) in the sequel. Their dynamics are given below:

$$\begin{aligned} \vec{s}_t &= \text{FE}(\vec{s}_{t-1}, y_t; \mathbf{w}_1), \\ \hat{\varphi}^{\mathbf{w}}(x_t)(y_{1:t}) &= \text{FD}(\vec{s}_t; \mathbf{w}_2), \end{aligned} \quad (5)$$

where $\hat{\varphi}^{\mathbf{w}}(x_t)(y_{1:t})$ is the estimation of $\varphi(x_t)$ using observations $y_{1:t}$ by RNN. Typically, the dynamical encoder **FE** ($\cdot; \mathbf{w}_1$) and the output decoder **FD** ($\cdot; \mathbf{w}_2$) are parameterized by FNNs with parameter $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2]$, respectively.

3.2. BRNN-Based smoother

After introducing the RNN-based filter, we turn to the BRNN-based smoother, which still consists of a dynamical encoder and an output decoder. Compared to RNN-based filter, a backward recursion of the hidden state is added into the dynamical encoder. Hence this dynamical encoder is divided into forward recursion and backward recursion. Let us denote the forward recursion of the dynamical encoder, the backward recursion of dynamical encoder and the output decoder by **Forward – Encoder**($\cdot$; θ_1), **Backward – Encoder**($\cdot$; θ_2) and **S – Decoder**($\cdot$; θ_3), respectively. For brevity, we denote them as **FE**($\cdot$; θ_1), **BE**($\cdot$; θ_2), and **SD**($\cdot$; θ_3) in the sequel. Their dynamics are summarized as follows:

$$\begin{aligned}\vec{s}_t &= \mathbf{FE}(\vec{s}_{t-1}, y_t; \theta_1), \\ \overleftarrow{s}_t &= \mathbf{BE}(\overleftarrow{s}_{t+1}, y_{t+1}; \theta_2), \\ \hat{\varphi}^{\mathbf{W}}(x_t)(y_{1:T}) &= \mathbf{SD}(s_t; \theta_3),\end{aligned}\quad (6)$$

where $s_t = [\vec{s}_t; \overleftarrow{s}_t]$ be the vector concatenating $\vec{s}_t$, $\overleftarrow{s}_t$, and $\hat{\varphi}^{\mathbf{W}}(x_t)(y_{1:T})$ is the estimation of $\varphi(x_t)$ using observations $y_{1:T}$ by BRNN. Usually, the dynamical encoder **FE**($\cdot$; θ_1), **BE**($\cdot$; θ_2) and the output decoder **SD**($\cdot$; θ_3) are parameterized by BRNN and FNN with parameter $\theta = [\theta_1, \theta_2, \theta_3]$ respectively. It is necessary to note that **FE**($\cdot$; θ_1) and **BE**($\cdot$; θ_2) can be any NNs with recurrent architecture such as LSTM (Hochreiter & Schmidhuber, 1997), GRU (Chung et al., 2014). Since the following theoretical analysis are based on continuous assumptions of **FE**($\cdot$; θ_1), **BE**($\cdot$; θ_2), and **SD**($\cdot$; θ_3), we only consider the original architecture in Schuster and Paliwal (1997).

Remark 3.1. Notice that in (6), $\vec{s}_t$ contains all the history up to time t , i.e., it can be viewed as the function of $y_1, \dots, y_t$ in (6). And $\overleftarrow{s}_t$ contains the information from time t to the terminate time T , i.e., it can be viewed as the function of $y_{t+1}, \dots, y_T$. s_t defined as the concatenation of $\vec{s}_t$ and $\overleftarrow{s}_t$, which contains information about all time-step observations.

4. BRNN approximation and error analysis for nonlinear smoothing

In this section, we first establish the approximation capability of BRNNs. Then we prove that the nonlinear smoother can be well approximated by a class of bidirectional system. Consequently, it can be further approximated by BRNN.

4.1. Approximation property of BRNN

We first give the approximation ability of FNN defined in Definition 2.1.

Theorem 4.1 (Universal Approximation Theorem Leshno et al., 1993). Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be a locally bounded piecewise continuous function that is not a polynomial. Then, for integers $n_1, n_2 \in \mathbb{N}^+$, the class $\Sigma^{n_1, n_2}(\sigma)$ is uniformly dense in C^{n_1, n_2} in any compact domain.

Using this theorem, we can prove the approximation ability of BRNN defined in (4) and the detailed proof is put in Appendix A.1.

Theorem 4.2 (Approximation Property of BRNNs). Let η_f, η_b, ξ be continuous functions and assume the input sequence $\{\alpha_t \in \mathbb{R}^{n_1} : t = 1, 2, \dots, T\}$ is bounded. Consider the system

$$\begin{cases} \vec{s}_t = \eta_f(\vec{s}_{t-1}, \alpha_t), \\ \overleftarrow{s}_t = \eta_b(\overleftarrow{s}_{t+1}, \alpha_t), \\ s_t = [\vec{s}_t; \overleftarrow{s}_t], \quad \beta_t = \xi(s_t). \end{cases}$$

Then for every $\varepsilon > 0$, there exists a $\text{BRNN}^{n_1, n_2}(\sigma)$ with output $\tilde{\beta}_t$ such that $|\beta_t - \tilde{\beta}_t| < \varepsilon$ for all $t = 1, 2, \dots, T$.

Remark 4.1. Theorem 4.2 extends naturally to the case where the backward recursion depends on α_{t+1} instead of α_t , i.e., $\overleftarrow{s}_t = \eta_b(\overleftarrow{s}_{t+1}, \alpha_{t+1})$. This follows by augmenting the backward state as $\overleftarrow{v}_t = [\overleftarrow{s}_t; \alpha_t]$ and applying Theorem 4.2 to the equivalent system.

4.2. Smoothing problem

The smoothing distribution satisfies

$$p(x_t | y_{1:T}) \propto p(x_t | y_{1:t}) p(y_{t+1:T} | x_t), \quad (7)$$

where $p(x_t | y_{1:t})$ is obtained from forward filtering and $p(y_{t+1:T} | x_t)$ from backward recursion.

Define the prediction, update, and backward transition operators, respectively, as:

$$\mathcal{F}_1(u)(x) \triangleq \int_{\mathbb{R}^n} p_{x_t | x_{t-1}}(x | z) u(z) dz, \quad (8)$$

$$\mathcal{F}_2(u, y)(x) \triangleq q(y | x) u(x), \quad (9)$$

$$\mathcal{F}_3(v)(x) \triangleq \int_{\mathbb{R}^n} p_{x_t | x_{t-1}}(z | x) v(z) dz, \quad (10)$$

where $q(y | x) = \exp[-\frac{1}{2}(y - h(x))^T R^{-1}(y - h(x))]$ is the unnormalized likelihood. Intuitively, $\mathcal{F}_1$ propagates the density forward in time (prediction), $\mathcal{F}_2$ applies the Bayesian measurement update, and $\mathcal{F}_3$ performs the backward Chapman–Kolmogorov integration.

The unnormalized filtering densities $\{u_t\}$ (proportional to $p(x_t | y_{1:t})$) and backward messages $\{v_t\}$ (representing $p(y_{t+1:T} | x_t)$) satisfy the recursions:

$$u_{t+1} = \mathcal{F}_2(\mathcal{F}_1(u_t), y_{t+1}), \quad v_t = \mathcal{F}_3(\mathcal{F}_2(v_{t+1}, y_{t+1})), \quad (11)$$

with $u_0 = p(x_0)$ and $v_T \equiv 1$. The smoothing distribution is then given by $p(x_t | y_{1:T}) \propto u_t(x)v_t(x)$.

Assumption 4.1. There exists $L > 0$ such that $y_t \in \mathcal{B}_L := \{y \in \mathbb{R}^m : |y| \leq L\}$ for all $1 \leq t \leq T$.

Remark 4.2. This assumption can be relaxed to cases where the observation process remains within a sufficiently large ball with high probability, in which case a similar result can be established in the expectation sense. For the sake of clarity and intuition in the proof, we adopt the slightly stronger Assumption 4.1 here.

4.3. Finite-dimensional coordinate representation

By Kolmogorov n -width theory (Pinkus, 2012), for any $\varepsilon > 0$ there exist N orthonormal functions $\{\phi_j\}_{j=1}^N \subset L^\infty(\mathbb{R}^n)$ such that

$$\max_{y \in \mathcal{B}_L} \left\| q(y | \cdot) - \sum_{j=1}^N \alpha_j(y) \phi_j(\cdot) \right\|_\infty < \varepsilon, \quad (12)$$

where $\alpha_j(y) \triangleq \int q(y | x) \phi_j(x) dx$. Define

$$\tilde{q}(y | x) = \sum_{j=1}^N \alpha_j(y) \phi_j(x).$$

For the remainder of this section, we assume that $0 < \varepsilon < \varepsilon_0$, where ε_0 is an arbitrary but fixed positive constant.

4.3.1. Forward approximation

Define the approximated operator

$$\tilde{\mathcal{F}}_2(u, y)(x) \triangleq \tilde{q}(y | x) u(x). \quad (13)$$

and the approximated densities $\{\tilde{u}_t\}$ evolve by

$$\tilde{u}_{t+1} = \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1}), \quad \tilde{u}_0 = u_0. \quad (14)$$

Define the *multiplication operator* $M_\phi(u)(x) \triangleq \phi(x)u(x)$. Since $\tilde{\mathcal{F}}_2(u, y) = \sum_{j=1}^N \alpha_j(y) M_{\phi_j}(u)$ admits the separated representation, the sequence $\{\tilde{u}_t\}$ evolves in a finite-dimensional subspace of $L^1(\mathbb{R}^n)$ given by

$$\mathcal{V}_N \triangleq \text{span} \left\{ \left(\bigcirc_{j=0}^{k-1} (M_{\phi_{i_j}} \circ \mathcal{F}_1) \right) u_0 : 1 \leq k \leq T, 1 \leq i_0, \dots, i_{k-1} \leq N \right\}, \quad (15)$$

Here $\bigcirc$ denotes right-to-left operator composition. Although each element in the spanning set corresponds to selecting a single basis function at each step, the actual recursion uses the full linear combination $\tilde{q}(y_{t+1}|\cdot) = \sum_{j=1}^N \alpha_j(y_{t+1})\phi_j$. Thus, the dependence on y_{t+1} only affects the coefficients $\{\alpha_j(y_{t+1})\}$, while the generating family of operators remains fixed. Consequently, the elements in (15) enumerate all single-basis “skeleton” trajectories, and their linear span contains every trajectory of $\{\tilde{u}_t\}$.

Let $\{\psi_i\}_{i=1}^{N_1}$ be a basis of $\mathcal{V}_N$. Then every $\tilde{u}_t \in \mathcal{V}_N$ admits a unique representation

$$\tilde{u}_t(x) = \sum_{i=1}^{N_1} a_i(t) \psi_i(x).$$

Since $\mathcal{V}_N$ is invariant under the approximated recursion (14), $\tilde{u}_{t+1}$ also belongs to $\mathcal{V}_N$. Therefore, there exists a uniquely determined coefficient vector $a(t+1) \in \mathbb{R}^{N_1}$ depending on $a(t)$ and y_{t+1} . Moreover, since $\mathcal{V}_N$ is finite-dimensional and both $\mathcal{F}_1$ and $\tilde{\mathcal{F}}_2$ are continuous operators, there exists a continuous map

$$\lambda : \mathbb{R}^{N_1} \times \mathcal{B}_L \rightarrow \mathbb{R}^{N_1}$$

such that

$$a(t+1) = \lambda(a(t), y_{t+1}). \quad (16)$$

4.3.2. Backward approximation

Similarly, define

$$\tilde{v}_t = \mathcal{F}_3(\tilde{\mathcal{F}}_2(\tilde{v}_{t+1}, y_{t+1})) \quad \tilde{v}_T = v_T \equiv 1. \quad (17)$$

By the same separation argument, $\{\tilde{v}_t\}$ evolves in a finite-dimensional subspace. Let $\{\tilde{\psi}_j\}_{j=1}^{N_2}$ be a basis of this space. Then every $\tilde{v}_t$ admits a unique representation

$$\tilde{v}_t(x) = \sum_{j=1}^{N_2} b_j(t) \tilde{\psi}_j(x).$$

Since the finite-dimensional subspace is invariant under the approximated backward recursion (17), $\tilde{v}_t$ is uniquely determined by $\tilde{v}_{t+1}$ and y_{t+1} . Therefore, there exists a uniquely determined coefficient vector $b(t) \in \mathbb{R}^{N_2}$ depending on $b(t+1)$ and y_{t+1} . Moreover, since the space is finite-dimensional and both $\mathcal{F}_3$ and $\tilde{\mathcal{F}}_2$ are continuous operators, there exists a continuous map

$$\Lambda : \mathbb{R}^{N_2} \times \mathcal{B}_L \rightarrow \mathbb{R}^{N_2}$$

such that

$$b(t) = \Lambda(b(t+1), y_{t+1}). \quad (18)$$

4.3.3. Smoothing approximation

The approximated smoothing distribution is

$$\tilde{p}_t(x) = \frac{\tilde{u}_t(x) \tilde{v}_t(x)}{\int \tilde{u}_t \tilde{v}_t} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} c_{i,j}(a(t), b(t)) \psi_i(x) \tilde{\psi}_j(x), \quad (19)$$

$$\text{where } c_{i,j}(a, b) = \frac{a_i b_j}{\sum_{k=1}^{N_1} \sum_{\ell=1}^{N_2} a_k b_\ell \langle \psi_k, \tilde{\psi}_\ell \rangle}.$$

4.4. Approximation error bounds

Lemma 4.1 (Forward Error Bound). Suppose [Assumptions 4.1](#) holds. Then for all $t = 1, \dots, T$,

$$\|u_t - \tilde{u}_t\|_1 \leq C_1 \varepsilon, \quad (20)$$

where $C_1 = T(1 + \varepsilon_0)^{T-1} \|u_0\|_1$.

The proof is provided in [Appendix A.2](#).

Lemma 4.2 (Backward Error Bound). Suppose [Assumption 4.1](#) holds. Then for all $t = 1, \dots, T$,

$$\|v_t - \tilde{v}_t\|_\infty \leq C_2 \varepsilon, \quad (21)$$

where $C_2 = T(1 + \varepsilon_0)^{T-1}$.

The proof can be found in [Appendix A.3](#).

Assumption 4.2. There exists $c > 0$ such that $Z_t \triangleq \int \rho_t(x) dx \geq c$ for all $1 \leq t \leq T$.

[Assumption 4.2](#) prevents density degeneracy. With these preliminary results in place, we can bound the error of the nonlinear smoother.

Theorem 4.3 (Smoothing Approximation Error). Suppose [Assumptions 4.1–4.2](#) hold. Then for all $t = 1, \dots, T$ and any test function $\varphi \in L^\infty$,

$$\left| \mathbb{E}[\varphi(x_t) | y_{1:T}] - \int \varphi(x) \tilde{p}_t(x) dx \right| \leq C_3 \|\varphi\|_\infty \varepsilon,$$

where C_3 depends on C_1, C_2, c , and

$$\begin{aligned} \int \varphi(x) \tilde{p}_t(x) dx &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} c_{i,j}(a(t), b(t)) \int \varphi \psi_i \tilde{\psi}_j \\ &\triangleq \Gamma(a(t), b(t)). \end{aligned} \quad (22)$$

The detailed derivation is provided in [Appendix A.4](#).

Finally, combining the approximation of the forward and backward dynamics with the universal approximation property of BRNNs, we obtain the following result.

Theorem 4.4 (BRNN Approximation of Smoothing Problem). Under the assumptions of [Theorem 4.3](#), for any test function $\varphi \in L^\infty$ and $\varepsilon > 0$, there exists a BRNN $^{n_1, n_2}(\sigma)$ with input y_t and output $\tilde{\beta}_t$ such that

$$|\tilde{\beta}_t - \mathbb{E}[\varphi(x_t) | y_{1:T}]| < \varepsilon, \quad \forall t = 1, \dots, T.$$

The proof, which follows by applying [Theorem 4.2](#) to the system (22) and combining the forward and backward error bounds, is provided in [Appendix A.5](#).

Remark 4.3. By following a similar proof as in [Theorem 4.4](#), a corresponding result can be established for filtering problems. That is, the forward recursion of BRNNs can also accurately approximate the optimal filtering dynamics.

5. Experiments

In this section, we aim to assess the performance of our proposed BRNN-based smoother in comparison to other existing smoothers on Nonlinear SSM. Additionally, we seek to validate the main results presented in [Theorem 4.4](#). For brevity, we will refer to the RNN-based filter as RNNF and the BRNN-based smoother as BRNNS. The subsequent numerical experiments exclusively involve synthetic data. Initially, we will train our model on the training set until convergence. Subsequently, we will evaluate the performance of our trained model on the test set.

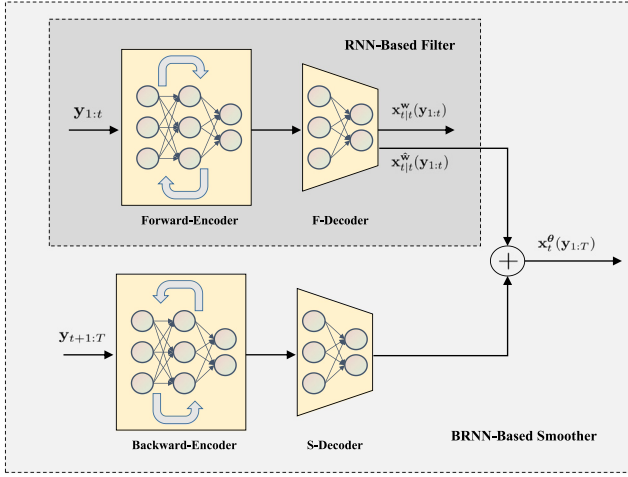


Fig. 1. The high-level architecture of BRNN-based smoother.

5.1. Training method

We use supervised learning with N paired data paths $(Y_{(i)}, X_{(i)})_{i=1}^N$, where $Y_{(i)} = [y_1^{(i)}, \dots, y_T^{(i)}]$ and $X_{(i)} = [x_1^{(i)}, \dots, x_T^{(i)}]$ are observations and corresponding states from system (1).

For brevity, denote the RNN-based filter output (5) as $\hat{x}_{t|t}^w$ and BRNN-based smoother output (6) as $\hat{x}_t^\theta$.

By (2) and (3), we define MSE objective functions with regularization coefficients λ_1, λ_2 :

$$\begin{aligned} \mathcal{L}_{\text{Filtering}}(\mathbf{w}) &= \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\left| \hat{x}_{t|t}^w(y_{1:t}) - x_t \right|_2^2 \right] + \lambda_1 \|\mathbf{w}\|_2^2, \\ \mathcal{L}_{\text{Smoothing}}(\theta) &= \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\left| \hat{x}_t^\theta(y_{1:T}) - x_t \right|_2^2 \right] + \lambda_2 \|\theta\|_2^2. \end{aligned} \quad (23)$$

The empirical (Monte Carlo) approximations are:

$$\begin{aligned} \hat{\mathcal{L}}_{\text{filtering}}(\mathbf{w}, N) &= \sum_{i=1}^N \frac{1}{T} \sum_{t=1}^T \left| \hat{x}_{t|t}^w(y_{1:t}^{(i)}) - x_t^{(i)} \right|_2^2 + \lambda_1 \|\mathbf{w}\|_2^2, \\ \hat{\mathcal{L}}_{\text{smoothing}}(\theta, N) &= \sum_{i=1}^N \frac{1}{T} \sum_{t=1}^T \left| \hat{x}_t^\theta(y_{1:T}^{(i)}) - x_t^{(i)} \right|_2^2 + \lambda_2 \|\theta\|_2^2. \end{aligned} \quad (24)$$

Following standard filtering-then-smoothing procedure, we first train the RNN-based filter to obtain $\hat{\mathbf{w}} = \arg \min \mathcal{L}_{\text{Filtering}}(\mathbf{w})$, then integrate it into the BRNN-based smoother with fixed $\hat{\mathbf{w}}$:

$$\begin{aligned} \overleftarrow{s}_t &= \mathbf{BE}(\overleftarrow{s}_{t+1}, y_{t+1}; \theta_2), \\ \hat{x}_t^\theta(y_{1:T}) &= \mathbf{SD}(\overleftarrow{s}_t; \theta_3) + \hat{x}_{t|t}^{\hat{\mathbf{w}}}(y_{1:t}), \end{aligned} \quad (25)$$

where

$$\mathbf{SD}(\overleftarrow{s}_t; \theta_3) = \tilde{W}_{hq} \overleftarrow{s}_t + \tilde{b}_q, \quad (26)$$

with $\theta_3 = [\tilde{W}_{hq}, \tilde{b}_q]$, $\tilde{W}_{hq} \in \mathbb{R}^{q \times h}$ and $\tilde{b}_q \in \mathbb{R}^{q \times 1}$. The architecture is illustrated in Fig. 1.

5.2. Experimental setup

Evaluation Metrics. We adopt the average Mean Squared Error (MSE) and average Standard Deviation (STD) as evaluation metrics in this study. These metrics, denoted by MSE and STD, respectively, are computed over 100 test paths and can be

calculated as follows:

$$\begin{aligned} \text{MSE} &:= \frac{1}{100} \sum_{j=1}^{100} \frac{1}{T} \sum_{t=1}^T \left| \hat{x}_t^{(j)} - x_t^{(j)} \right|_2^2, \\ \text{STD} &:= \frac{1}{T} \sum_{t=1}^T \sqrt{\frac{\sum_{j=1}^{100} \left| \hat{x}_t^{(j)} - \frac{\sum_{j=1}^{100} \hat{x}_t^{(j)}}{100} \right|_2^2}{100}}. \end{aligned} \quad (27)$$

Here, $x_t^{(j)}$ represents the true state at time step t in the j th path, while $\hat{x}_t^{(j)}$ denotes the estimated value of $x_t^{(j)}$.

Implementation Details. The implementations of RNNF and BRNNS were performed using the PyTorch framework, while EKS, and PS were implemented using Numpy. All the models mentioned above employ the Rectified Linear Unit (ReLU) as the activation function and the number of layers for both RNN and BRNN is set to 2. Gaussian initialization was adopted for parameter initialization in the NNs model with standard setting. Adam with mini-batch gradient descent was used as the optimizer during model training. The training process for all the experiments took approximately 20 min on a single Tesla P100, with the inference process executed on an Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz. The hyper-parameters of the models used are summarized in Table 1.

5.3. Nonlinear SSM

In this subsection, we examine a non-linear SSM characterized by a cubic sensor function. The non-linear SSM is defined as follows:

$$\begin{aligned} x_{t+1} &= (I_3 + \kappa_1 A) x_t + \kappa_2 \cos(x_t) + w_t, \\ y_t &= x_t^3 + v_t, \end{aligned} \quad (28)$$

where $x_0 \sim \mathcal{N}(0, I_3)$, $\mathbb{E}[w_t^\top w_t] = 0.5I_3$ and $\mathbb{E}[v_t^\top v_t] = 0.5I_3$. $\kappa_1 = \kappa_2 = 0.05$ are constants controlling the state dynamics, and $A \in \mathcal{M}_{3 \times 3}(\mathbb{R})$ is defined as follows:

$$A = \begin{bmatrix} -1 & 0.2 & 0 \\ 0.2 & -1 & 0.2 \\ 0 & 0.2 & -1 \end{bmatrix}.$$

Unless otherwise specified, all hyperparameters follow Table 1. It is important to highlight that based on the main theorems presented in Crisan and Doucet (2002), the PS attains convergence to the target distribution function in terms of L^p -convergence, where $p \geq 2$, as the number of particles approaches infinity. Hence, with a sufficiently large number of particles, the PS can be regarded as a state-of-the-art (SOTA) benchmark algorithm in terms of accuracy. Here, we set the number of particles in PS to be 3000. Fig. 2 illustrates the MSE of BRNNS as a function of the number of hidden neurons, training paths, and training steps. The results demonstrate that estimation performance improves significantly as the number of hidden neurons increases, ultimately surpassing both EKS and PS. Similar performance gains are observed when increasing the number of training paths and training steps. Furthermore, in Fig. 3, we present the true state versus the estimate of EKS, PS, and BRNNS over time. We observe that in every state dimension, the estimate of BRNNS (dashed line) outperforms the estimation of EKS (dash-dotted line).

The specific comparison results, including the MSE between EKS, PS, BRNNS, and the true state, as well as their respective inference times, are summarized in Table 2. Note that BRNNS outperforms both the EKS and PS, and it has the same inference speed as EKS. Although PS also gives good estimates, its excessive inference time significantly limits its efficiency. Hence these arguments show that BRNNS is a very efficient smoother that can be applied to nonlinear smoothing problems.

Table 1
The hyper-parameters for BRNNS on nonlinear SSM.

Hidden neurons	Epochs	Batch size	Learning rate	Training paths	Training steps
48	500	256	0.005	1024	300

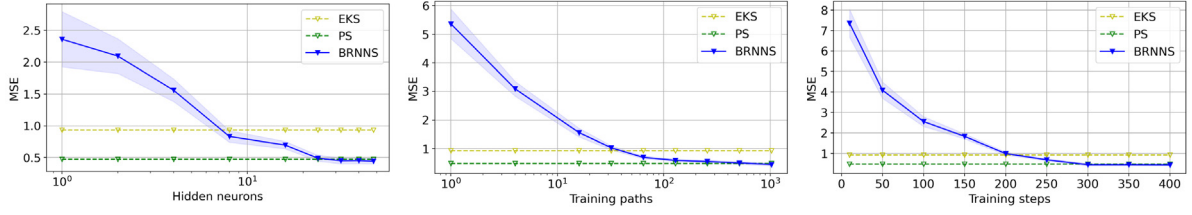


Fig. 2. MSE comparisons with respect to the number of hidden neurons (left), training paths (middle) and training steps (right) for the nonlinear SSM. All other parameters in the three figures are kept consistent with Table 1.

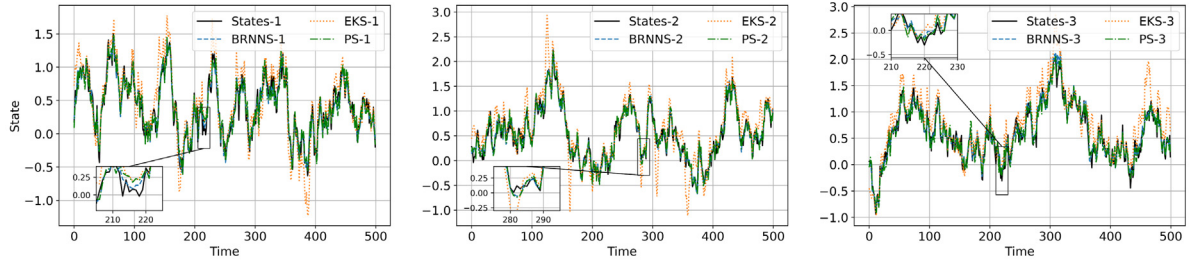


Fig. 3. Comparison of the true state versus the estimates obtained from EKS, PS, and BRNNS over time.

Table 2
The comparison of different methods on the nonlinear SSM.

Methods	MSE	STD	Inference time [s]
EKS	0.9079	0.1137	0.0372
PS	0.4445	0.0212	685.1445
BRNNS	0.4116	0.0205	0.0466

6. Conclusion

In this paper, we establish the universal approximation capability of BRNNs. As a special case, they can well approximate optimal smoothers under some mild conditions. Our work offers significant insights into the quantitative analysis of the approximation capacities of BRNNs with noisy sequential inputs. Building on this result, we create a new smoothing algorithm, the BRNN-based smoother, which we compare to the EKS and PS for efficiency. Our theoretical analysis concentrates on the original BRNN, but examining the approximation capacity of its variations like Bidirectional LSTM and Bidirectional GRU in SSM form can be an exciting avenue for future research.

Appendix. Proofs of main results

A.1. Proof of Theorem 4.2

The proof follows from the universal approximation theorem and uniform continuity.

Step 1. For any $\delta > 0$, there exist neural networks NN_1, NN_2 such that

$$\sup_{(\vec{s}, \alpha) \in U} |\eta_f(\vec{s}, \alpha) - NN_1(\vec{s}, \alpha)| < \delta,$$

$$\sup_{(\vec{s}, \alpha) \in U} |\eta_b(\vec{s}, \alpha) - NN_2(\vec{s}, \alpha)| < \delta,$$

where U is compact.

Define $\vec{s}_t = NN_1(\vec{s}_{t-1}, \alpha_t)$ with $\vec{s}_0 = \vec{s}_0$, and let $e_t = |\vec{s}_t - \tilde{s}_t|$. Then

$$e_{t+1} \leq \omega_f(e_t) + \delta, \quad e_0 = 0,$$

where ω_f is the modulus of continuity of η_f .

Define $E_0(\delta) = 0$ and $E_{t+1}(\delta) = \omega_f(E_t(\delta)) + \delta$. Then $e_t \leq E_t(\delta)$ for all $t \leq T$, and

$$\max_{1 \leq t \leq T} E_t(\delta) \rightarrow 0 \quad \text{as } \delta \rightarrow 0.$$

The same argument applies to the backward recursion, yielding $|s_t - \tilde{s}_t| \leq \tilde{E}(\delta)$ with some $\tilde{E}(\delta) \rightarrow 0$.

Step 2. Since ξ is continuous on a compact set, there exists a neural network NN_3 such that

$$\sup_s |\xi(s) - NN_3(s)| < \delta.$$

Thus

$$|\beta_t - \tilde{\beta}_t| \leq \omega_\xi(\tilde{E}(\delta)) + \delta,$$

where ω_ξ is the modulus of continuity of ξ . Letting $\delta \rightarrow 0$ yields the result.

A.2. Proof of Lemma 4.1

We first prove the boundedness of $\mathcal{F}_1$. Using Tonelli's theorem, we have, for any $u \in L^1$,

$$\|\mathcal{F}_1 u\|_1 \leq \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} p_{x_t|x_{t-1}}(x|z) |u(z)| dz dx = \|u\|_1. \quad (\text{A.1})$$

For $t = 0$, we have $u_0 = \tilde{u}_0$, so $\|u_0 - \tilde{u}_0\|_1 = 0$.

For $t + 1$, using the recursions (11) and (14), we have

$$\begin{aligned} & \|u_{t+1} - \tilde{u}_{t+1}\|_1 \\ &= \|\mathcal{F}_2(\mathcal{F}_1(u_t), y_{t+1}) - \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1 \\ &\leq \|\mathcal{F}_2(\mathcal{F}_1(u_t), y_{t+1}) - \mathcal{F}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1 \\ &\quad + \|\mathcal{F}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1}) - \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1. \end{aligned}$$

For the first term, note that $q(y | x) = \exp[-\frac{1}{2}(y - h(x))^T R^{-1}(y - h(x))]$ with $R^{-1} > 0$, so the exponent is non-positive and $0 \leq q(y | x) \leq 1$ for all $x \in \mathbb{R}^n, y \in \mathcal{B}_L$. Hence $\mathcal{F}_2(u, y)(x) = q(y | x)u(x)$ is multiplicative by a function bounded by 1, giving

$$\begin{aligned} & \|\mathcal{F}_2(\mathcal{F}_1(u_t), y_{t+1}) - \mathcal{F}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1 \\ & \leq \|\mathcal{F}_1(u_t) - \mathcal{F}_1(\tilde{u}_t)\|_1 \leq \|u_t - \tilde{u}_t\|_1. \end{aligned}$$

where we use (A.1).

For the second term, by (12), we have

$$\begin{aligned} \|\mathcal{F}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1}) - \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1 & \leq \varepsilon \|\mathcal{F}_1(\tilde{u}_t)\|_1 \\ & \leq \varepsilon \|\tilde{u}_t\|_1. \end{aligned}$$

Now we need to bound $\|\tilde{u}_t\|_1$. From (14), we have

$$\|\tilde{u}_{t+1}\|_1 = \|\tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_t), y_{t+1})\|_1.$$

Since $\tilde{\mathcal{F}}_2(u, y)(x) = (\sum_{j=1}^N \alpha_j(y) \phi_j(x))u(x)$ and $|\sum_{j=1}^N \alpha_j(y) \phi_j|$ is bounded by $\|q(y | \cdot)\|_\infty + \varepsilon \leq 1 + \varepsilon$, we obtain

$$\|\tilde{u}_{t+1}\|_1 \leq (1 + \varepsilon) \|\mathcal{F}_1(\tilde{u}_t)\|_1 \leq (1 + \varepsilon) \|\tilde{u}_t\|_1.$$

By induction, $\|\tilde{u}_t\|_1 \leq (1 + \varepsilon)^t \|u_0\|_1$ (using $\tilde{u}_0 = u_0$).

Let $e_t = \|u_t - \tilde{u}_t\|_1$ denote the error at step t . Combining the estimates above, we obtain the following recurrence relation for the error:

$$e_{t+1} \leq e_t + \varepsilon(1 + \varepsilon)^t \|u_0\|_1.$$

Unrolling this recurrence from $t = 0$ (noting that $e_0 = 0$), we get:

$$e_t \leq \sum_{k=0}^{t-1} \varepsilon(1 + \varepsilon)^k \|u_0\|_1 = ((1 + \varepsilon)^t - 1) \|u_0\|_1.$$

Using the mean value theorem, we can bound $(1 + \varepsilon)^t - 1 \leq t\varepsilon(1 + \varepsilon)^{t-1}$ for $\varepsilon > 0$. Therefore, the error at any time $t \leq T$ is bounded by:

$$\|u_t - \tilde{u}_t\|_1 \leq C_1 \varepsilon, \quad \text{where } C_1 = T(1 + \varepsilon_0)^{T-1} \|u_0\|_1.$$

A.3. Proof of Lemma 4.2

We first prove the boundedness of $\mathcal{F}_3$. For $w \in L^\infty$,

$$|(\mathcal{F}_3 w)(x)| \leq \int_{\mathbb{R}^n} p_{x_{t+1}|x_t}(z | x) |w(z)| dz \leq \|w\|_\infty$$

Thus $\|\mathcal{F}_3 w\|_\infty \leq \|w\|_\infty$.

Then the proof is completely analogous to that of Lemma 4.1, adapted for the backward recursion. We start from the terminal condition $v_T = \tilde{v}_T \equiv 1$, which trivially gives $v_T - \tilde{v}_T = 0$.

Using the exact and approximate recursions $v_t = \mathcal{F}_3(\mathcal{F}_2(v_{t+1}, y_{t+1}))$ and $\tilde{v}_t = \mathcal{F}_3(\tilde{\mathcal{F}}_2(\tilde{v}_{t+1}, y_{t+1}))$, we decompose the error via the triangle inequality and obtain the backward recurrence for the error $e_t = \|v_t - \tilde{v}_t\|_\infty$:

$$e_t \leq e_{t+1} + \varepsilon \|\tilde{v}_{t+1}\|_\infty.$$

This recurrence is identical in mathematical form to the forward case in Lemma 4.1. Similarly, we have $\|\tilde{v}_{t+1}\|_\infty \leq (1 + \varepsilon)^{T-t-1} \|v_T\|_\infty$. Following the exact same induction and unrolling arguments, with time unrolled backwards from T instead of forwards from 0, we deduce $\|v_t - \tilde{v}_t\|_\infty \leq C_2 \varepsilon$, where $C_2 = T(1 + \varepsilon_0)^{T-1}$.

A.4. Proof of Theorem 4.3

Let $\rho_t = u_t v_t$ and $\tilde{\rho}_t = \tilde{u}_t \tilde{v}_t$. We first bound $\|\rho_t - \tilde{\rho}_t\|_1$. Using the decomposition

$$\rho_t - \tilde{\rho}_t = u_t v_t - \tilde{u}_t \tilde{v}_t = u_t(v_t - \tilde{v}_t) + (u_t - \tilde{u}_t)\tilde{v}_t,$$

and applying Hölder's inequality along with Lemmas 4.1 and 4.2, we obtain

$$\begin{aligned} \|\rho_t - \tilde{\rho}_t\|_1 & = \|u_t(v_t - \tilde{v}_t) + (u_t - \tilde{u}_t)\tilde{v}_t\|_1 \\ & \leq \|u_t(v_t - \tilde{v}_t)\|_1 + \|(u_t - \tilde{u}_t)\tilde{v}_t\|_1 \\ & \leq \|u_t\|_1 \|v_t - \tilde{v}_t\|_\infty + \|u_t - \tilde{u}_t\|_1 \|\tilde{v}_t\|_\infty \\ & \leq \|u_t\|_1 \cdot C_2 \varepsilon + C_1 \varepsilon \cdot \|\tilde{v}_t\|_\infty \leq C' \varepsilon, \end{aligned}$$

where $C' = \max_t(\|u_t\|_1 C_2 + C_1 \|\tilde{v}_t\|_\infty)$ is finite by the boundedness established in the proofs of Lemmas 4.1 and 4.2.

Now consider the normalized densities $p_t = \rho_t / Z_t$ and $\tilde{p}_t = \tilde{\rho}_t / \tilde{Z}_t$, where $Z_t = \int \rho_t(x) dx$ and $\tilde{Z}_t = \int \tilde{\rho}_t(x) dx$. Moreover, $\tilde{Z}_t = \|\tilde{\rho}_t\|_1 \leq \|\tilde{u}_t\|_1 \|\tilde{v}_t\|_\infty$. From the proofs of Lemmas 4.1 and 4.2 we have $\tilde{Z}_t \leq (1 + \varepsilon)^T \|u_0\|_1 =: M_Z < \infty$. Note that $|Z_t - \tilde{Z}_t| = |\int(\rho_t - \tilde{\rho}_t) dx| \leq \|\rho_t - \tilde{\rho}_t\|_1 \leq C' \varepsilon$.

We have

$$\begin{aligned} \|p_t - \tilde{p}_t\|_1 & = \left\| \frac{\rho_t}{Z_t} - \frac{\tilde{\rho}_t}{\tilde{Z}_t} \right\|_1 \\ & = \left\| \frac{\rho_t}{Z_t} - \frac{\tilde{\rho}_t}{Z_t} + \frac{\tilde{\rho}_t}{Z_t} - \frac{\tilde{\rho}_t}{\tilde{Z}_t} \right\|_1 \\ & \leq \frac{1}{Z_t} \|\rho_t - \tilde{\rho}_t\|_1 + \left| \frac{1}{Z_t} - \frac{1}{\tilde{Z}_t} \right| \|\tilde{\rho}_t\|_1 \\ & = \frac{1}{Z_t} \|\rho_t - \tilde{\rho}_t\|_1 + \frac{|Z_t - \tilde{Z}_t|}{Z_t \tilde{Z}_t} \|\tilde{\rho}_t\|_1. \end{aligned}$$

By Assumption 4.2, we have $Z_t \geq c$ for all t . For sufficiently small ε , we also have $\tilde{Z}_t \geq c/2$ (since $|Z_t - \tilde{Z}_t| \leq C' \varepsilon$ can be made smaller than $c/2$). Moreover, $\|\tilde{\rho}_t\|_1 = \tilde{Z}_t$ is bounded. Therefore,

$$\|p_t - \tilde{p}_t\|_1 \leq \frac{C' \varepsilon}{c} + \frac{C' \varepsilon \cdot 2\tilde{Z}_t}{c^2} \leq C_3 \varepsilon,$$

where $C_3 = C'(1/c + 2M_Z/c^2)$, with $M_Z = (1 + \varepsilon)^T \|u_0\|_1$ bounding $\tilde{Z}_t$ uniformly in t .

Finally, for the smoothing expectation, we have

$$\begin{aligned} & \left| \mathbb{E}[\varphi(x_t) | y_{1:T}] - \int \varphi(x) \tilde{p}_t(x) dx \right| \\ & = \left| \int \varphi(x) [p_t(x) - \tilde{p}_t(x)] dx \right| \leq \|\varphi\|_\infty \|p_t - \tilde{p}_t\|_1 \leq C_3 \|\varphi\|_\infty \varepsilon. \end{aligned}$$

A.5. Proof of Theorem 4.4

By Theorem 4.3, the smoothing expectation can be expressed as

$$|\mathbb{E}[\varphi(x_t) | y_{1:T}] - \Gamma(a(t), b(t))| \leq O(\varepsilon_1),$$

where ε_1 is the approximation error from the finite-dimensional representation, $(a(t), b(t))$ are the coefficient vectors defined in (16) and (18), and Γ is the continuous function defined in (22).

The coefficients $(a(t), b(t))$ satisfy the continuous recursions

$$a(t+1) = \lambda(a(t), y_{t+1}), \quad b(t) = \Lambda(b(t+1), y_{t+1}),$$

established in (16) and (18). The functions $\lambda: \mathbb{R}^{N_1} \times \mathbb{R}^m \rightarrow \mathbb{R}^{N_1}$ and $\Lambda: \mathbb{R}^{N_2} \times \mathbb{R}^m \rightarrow \mathbb{R}^{N_2}$ are continuous by construction (they involve inner products and sums of continuous functions). Moreover, since $\{y_t\} \subset \mathcal{B}_L$ is bounded by Assumption 4.1, the recursions evolve in a compact domain.

Define the system

$$\begin{cases} \vec{s}_t = a(t), \\ \vec{\zeta}_t = b(t), \\ s_t = [\vec{s}_t; \vec{\zeta}_t], \\ \beta_t = \Gamma(\vec{s}_t, \vec{\zeta}_t). \end{cases}$$

Note that this system has the structure in [Theorem 4.2](#) and satisfies the conditions of [Remark 4.1](#) with $\eta_f = \lambda$, $\eta_b = \Lambda$, $\xi = \Gamma$, and $\alpha_t = y_t$.

Therefore, by [Remark 4.1](#), for any $\varepsilon_2 > 0$, there exists a $BRNN^{n_1, n_2}(\sigma)$ with input $\alpha_t = y_t$ and output $\tilde{\beta}_t$ such that

$$|\tilde{\beta}_t - \beta_t| < \varepsilon_2, \quad \forall t = 1, \dots, T.$$

Combining this with the error from [Theorem 4.3](#), we have $|\tilde{\beta}_t - \mathbb{E}[\varphi(x_t) | y_{1:T}]| \leq |\tilde{\beta}_t - \beta_t| + |\beta_t - \mathbb{E}[\varphi(x_t) | y_{1:T}]| < \varepsilon_2 + C_3 \|\varphi\|_{\infty} \varepsilon_1$.

By choosing ε_1 and ε_2 sufficiently small (depending on the desired final accuracy ε), we can ensure that

$$|\tilde{\beta}_t - \mathbb{E}[\varphi(x_t) | y_{1:T}]| < \varepsilon, \quad \forall t = 1, \dots, T.$$

References

- Carvalho, Carlos M., Johannes, Michael S., Lopes, Hedibert F., & Polson, Nicholas G. (2010). Particle learning and smoothing. *Statistical Science*, 25(1), 88–106.
- Chen, Xiuqiong, Tao, Yangtianze, Xu, Wenjie, & Yau, Stephen Shing-Toung (2023). Recurrent neural networks are universal approximators with stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10), 7992–8006.
- Chiu, Chung-Cheng, Sainath, Tara N., Wu, Yonghui, Prabhavalkar, Rohit, Nguyen, Patrick, Chen, Zhifeng, Kannan, Anjuli, Weiss, Ron J., Rao, Kanishka, Gonina, Ekaterina, et al. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 4774–4778). IEEE.
- Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, & Bengio, Yoshua (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
- Crisan, Dan, & Doucet, Arnaud (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.*, 50(3), 736–746.
- Elman, Jeffrey L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211.
- Goldberg, Yoav (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 345–420.
- Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo, & Wierstra, Daan (2015). DRAW: a recurrent neural network for image generation. 37, In *Proceedings of the 32nd International Conference on Machine Learning* Proceedings of Machine Learning Research, (pp. 1462–1471). PMLR, <https://proceedings.mlr.press/v37/gregor15.html>.
- Hochreiter, Sepp, & Schmidhuber, Jürgen (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hornik, Kurt, Stinchcombe, Maxwell, & White, Halbert (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- Jazwinski, Andrew H. (2007). *Stochastic processes and filtering theory*. Courier Corporation.
- Klaas, Mike, Briers, Mark, De Freitas, Nando, Doucet, Arnaud, Maskell, Simon, & Lang, Dustin (2006). Fast particle smoothing: If I had a million particles. In *Proc. mach. learn. res.* (pp. 481–488).
- Leshno, Moshe, Lin, Vladimir Ya., Pinkus, Allan, & Schocken, Shimon (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6), 861–867.
- Lo, J. Ting-Ho (1994). Synthetic approach to optimal filtering. *IEEE Trans. Neural Netw.*, 5(5), 803–811.
- Mikolov, Tomas, Karafiát, Martin, Burget, Lukas, Cernocký, Jan, & Khudanpur, Sanjeev (2010). Recurrent neural network based language model. In *Interspeech: Vol. 2*, (3), (pp. 1045–1048). Makuhari.
- Ni, Xiaoyong, Revach, Guy, Shlezinger, Nir, van Sloun, Ruud J. G., & Eldar, Yonina C. (2022). RTSNet: Deep learning aided Kalman smoothing. In *ICASSP 2022 - 2022 IEEE international conference on acoustics, speech and signal processing* (pp. 5902–5906).
- Perrusquia, Adolfo, & Yu, Wen (2021). Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. *Neurocomputing*, 438, 145–154.
- Pinkus, Allan (2012). *N-widths in approximation theory: Vol. 7*, Springer Science & Business Media.
- Rauch, Herbert E., Tung, F., & Striebel, Charlotte T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8), 1445–1450.
- Santoro, Adam, Faulkner, Ryan, Raposo, David, Rae, Jack, Chrzanowski, Mike, Weber, Theophane, Wierstra, Daan, Vinyals, Oriol, Pascanu, Razvan, & Lillicrap, Timothy (2018). Relational recurrent neural networks. *Proceedings of Advance Neural Information Processing System*, 31.
- Schäfer, Anton Maximilian, & Zimmermann, Hans Georg (2006). Recurrent neural networks are universal approximators. In *Lect. notes comput. sci.* (pp. 632–640). Springer.
- Schuster, Mike, & Paliwal, Kuldip K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11), 2673–2681.
- Sutskever, Ilya, Martens, James, & Hinton, Geoffrey E. (2011). Generating text with recurrent neural networks. In *Proc. mach. learn. res.*
- Tao, Yangtianze, Kang, Jiayi, & Yau, Stephen Shing-Toung (2024). Neural projection filter: Learning unknown dynamics driven by noisy observations. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7), 9508–9522.
- Van Den Oord, Aaron, Kalchbrenner, Nal, & Kavukcuoglu, Koray (2016). Pixel recurrent neural networks. In *Proc. mach. learn. res.* (pp. 1747–1756). PMLR.
- Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560. <http://dx.doi.org/10.1109/5.58337>.
- Yu, Yong, Si, Xiaosheng, Hu, Changhua, & Zhang, Jianxun (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235–1270.



Yangtianze Tao received the B.S. degree in mathematics and applied mathematics from the College of Mathematics, Sichuan University, Sichuan, China, in 2019. He received the Ph.D. degree in applied mathematics from the Department of Mathematical Sciences, Tsinghua University, Beijing, China, in 2024. His research interests include deep learning, machine learning and nonlinear filtering.



Jiayi Kang received his B.S. degree from the College of Mathematics, Sichuan University, Chengdu, China, in 2019 and his Ph.D. degree from the Department of Mathematical Sciences at Tsinghua University, China, in 2024. In July 2024, he joined the Beijing Institute of Mathematics and Applications (BIMSA) as an Assistant Professor. Since November 2025, he has been serving as an Assistant Professor at the Hetao Institute of Mathematics and Interdisciplinary Sciences (HIMIS). His research focuses on the interdisciplinary areas of deep learning, nonlinear filtering, and computational

biology.



Xiuqiong Chen received the B.S. degree from the School of Mathematical Sciences, Beihang University, Beijing, China, in 2014, and the Ph.D. degree in applied mathematics from the Department of Mathematical Sciences, Tsinghua University, Beijing, China, in 2019. From 2019 to 2021, she was a Postdoctoral Scholar at the Yau Mathematical Sciences Center, Tsinghua University. She joined Renmin University of China in 2021, where she is currently an Associate Professor with the School of Mathematics. Her research interests include nonlinear filtering and deep learning.



Stephen S.-T. Yau Stephen Shing-Toung Yau received the Ph.D. degree in mathematics from the State University of New York at Stony Brook, NY, US, in 1976. He was a Member of the Institute of Advanced Study at Princeton, Princeton, NJ, USA from 1976–1977 and 1981–1982, and a Benjamin Pierce Assistant Professor at Harvard University, Cambridge, MA, USA during 1977–1980. During 1980–2011, he joined the Department of Mathematics, Statistics and Computer Science (MSCS), University of Illinois at Chicago (UIC), Chicago, IL, USA. During 2002–2011, he became a joint Professor with the Department of Electrical and Computer Engineering at the MSCS, UIC. After his retirement in 2011, he joined Tsinghua University, Beijing, P. R. China, where he is a full-time professor in Department of Mathematical Sciences. His current research interests include nonlinear filtering, bioinformatics, complex algebraic geometry, CR geometry and singularities theory. He has over 500 publications.

Dr. Yau is the Managing Editor and founder of the *Journal of Algebraic Geometry* since 1992, and the Editor in-Chief and founder of *Communications in Information and Systems* since 2001. He was the General Chairman of the IEEE International Conference on Control and Information, which was held in the Chinese University of Hong Kong in 1995. He was awarded the Sloan Fellowship in 1980, the Guggenheim Fellowship in 2000, the IEEE Fellow Award in 2003, the AMS Fellow Award in 2013, and the IEEE Life Fellow Award in 2022. In 2005, he was entitled the UIC University Distinguished Professor.