UNCERTAINTY IN DATA-DRIVEN KALMAN FILTERING FOR PARTIALLY KNOWN STATE-SPACE MODELS

Itzik Klein, Guy Revach, Nir Shlezinger, Jonas E. Mehr, Ruud J. G. van Sloun, and Yonina. C. Eldar

ABSTRACT

Providing a metric of uncertainty alongside a state estimate is often crucial when tracking a dynamical system. Classic state estimators, such as the Kalman filter (KF), provide a time-dependent uncertainty measure from knowledge of the underlying statistics; however, deep learning based tracking systems struggle to reliably characterize uncertainty. In this paper, we investigate the ability of KalmanNet, a recently proposed; hybrid; model-based; deep state tracking algorithm, to estimate an uncertainty measure. By exploiting the interpretable nature of KalmanNet, we show that the error covariance matrix can be computed based on its internal features, as an uncertainty measure. We demonstrate that when the system dynamics are known, KalmanNet—which learns its mapping from data without access to the statistics—provides uncertainty similar to that provided by the KF; and while in the presence of evolution model-mismatch, KalmanNet provides a more accurate error estimation.

Index Terms— Kalman filter, deep learning, uncertainty

1. INTRODUCTION

Tracking a hidden state vector from noisy observations in real-time is at the core of many signal processing applications. A leading approach to the task is the Kalman filter (KF) [1], which operates with low complexity and achieves a minimum mean-squared error (MMSE) in setups characterized by linear Gaussian state space (SS) models. A key merit of the KF, which is of paramount importance in safety critical applications, e.g., autonomous driving, aviation, and medical, is its ability to provide uncertainty alongside state estimation [2, Ch. 4]. KF and its variants are model-based (MB) algorithms, and are therefore sensitive to inaccuracies in modeling the system dynamics using the SS model.

The recent success of deep learning architectures, such as recurrent neural networks (RNNs) [3] and attention mechanisms [4], in learning from complex time-series data in unstructured environments and in a model-agnostic manner, evoked interest in using them for tracking dynamical systems. However, the lack of interpretability of deep architectures and their inability to capture uncertainty [5], together with the fact that they require many trainable parameters and large data sets even for simple setups [6], limit their applicability for safety-critical applications in hardware-limited systems.

Characterizing uncertainty in deep neural networks (DNNs) is an active area of research [7–9]. One approach is to use Bayesian DNNs [10], which, when combined with SS models, enables extraction of the uncertainty. See, e.g., [11–13]. However, doing so relies on variational inference, which makes learning more complex and less scalable, and cannot be used directly for state estimation [5]. Alternatively, one can use deep learning techniques to estimate the SS model parameters and then plug them into a variant of a KF that provides uncertainty; e.g., [14–17]. These approaches are limited in accuracy and complexity, requiring linearization of the second-order moments to compute the error covariance as required by the MB KF. The KF-inspired RNN proposed [5] was trained to predict both the state and the error. However, [5] focused on specific factorizable SS models with partially observable states, for which the KF is simplified to parallel scalar operations, and the proposed architecture does not naturally extend to general SS models.

The recently proposed KalmanNet [18] utilizes RNNs to enhance the robustness of the KF for complex and mismatched models as a form of MB deep learning [19]. KalmanNet was shown to operate reliably in practical hardware-limited systems [20] due to its principled incorporation of partial domain knowledge. In this work we show how KalmanNet can be extended to provide uncertainty measures, by exploiting its interpretable architectures and the fact that it preserves the internal flow of the KF. In particular, we build upon the identification of an internal feature as the estimated Kalman gain (KG), and show that when combined with partial domain knowledge, it can be used to compute the time-dependent error covariance matrix. We numerically show that the extracted uncertainty indeed reflects the performance of KalmanNet, providing similar state estimates and error measures as the KF—which knows the SS model—while being

I. Klein is with the Hatter Department of Marine Technologies, University of Haifa, Israel (email: kitzik@univ.haifa.ac.il). G. Revach and J. E. Mehr are with the Institute for Signal and Information Processing (ISI), D-ITET, ETH Zürich, (email: grevach@ethz.ch, mehrjo@student.ethz.ch). N. Shlezinger is with the School of ECE, Ben-Gurion University of the Negev, Beer Sheva, Israel (e-mail: nirshl@bgu.ac.il). R. J. G. van Sloun is with the EE Dpt., Eindhoven University of Technology, and with Phillips Research, Eindhoven, The Netherlands (e-mail: r.j.g.v.sloun@tue.nl). Y. C. Eldar is with the Faculty of Math and CS, Weizmann Institute of Science, Rehovot, Israel (e-mail: yonina.eldar@weizmann.ac.il). The authors thank Prof. Hans-Andrea Loeliger for his helpful comments and discussion.

notably more reliable in terms of tracking and uncertainty in the presence of mismatched models.

The rest of the paper is organized as follows: Section 2 reviews the SS model and recalls the KF and KalmanNet. Section 3 analyzes the extraction of uncertainty in KalmanNet, while Section 4 presents a numerical study.

2. SYSTEM MODEL AND PRELIMINARIES

We review the SS model and briefly recall the MB KF, since its operation serves as the baseline for KalmanNet and for its uncertainty extraction scheme detailed in Section 3. We then recap the data-driven filtering problem and the architecture of KalmanNet. For simplicity, we focus on linear SS models, although the derivations can also be used for non-linear models in the same manner as the extended KF [2, Ch. 10].

2.1. System Model and Model-Based Kalman Filtering

We consider a dynamical system characterized by a linear, Gaussian, continuous evolution model in discrete-time. For $t \in \mathbb{Z}$, this SS model is defined by [21]

$$\mathbf{x}_t = \mathbf{F} \cdot \mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{x}_t \in \mathbb{R}^m, \quad \text{(1a)}$$

$$\mathbf{y}_t = \mathbf{H} \cdot \mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad \mathbf{y}_t \in \mathbb{R}^n. \quad \text{(1b)}$$

In (1a), $\mathbf{x}_t$ is the latent state vector of the system at time t, which evolves by a linear state evolution matrix $\mathbf{F}$ and by an additive white Gaussian noise (AWGN) $\mathbf{w}_t$ with noise covariance $\mathbf{Q}$. In (1b), $\mathbf{y}_t$ is the vector of observations at time t, $\mathbf{H}$ is the measurement matrix, and $\mathbf{v}_t$ is an AWGN with measurement noise covariance $\mathbf{R}$. The filtering problem deals with real-time state estimation; i.e., the recovery of $\mathbf{x}_t$ from $\{\mathbf{y}_\tau\}_{\tau \le t}$ for each time instance t [2].

The KF is a two-step, low complexity, recursive algorithm that produces a new estimate $\hat{\mathbf{x}}_t$ from a new observation $\mathbf{y}_t$ based on the previous estimate $\hat{\mathbf{x}}_{t-1}$ as a sufficient statistic. In the first step it predicts the statistical moments based on the previous *a posteriori* estimates:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F} \cdot \hat{\mathbf{x}}_{t-1}, \quad \boldsymbol{\Sigma}_{t|t-1} = \mathbf{F} \cdot \boldsymbol{\Sigma}_{t-1} \cdot \mathbf{F}^\top + \mathbf{Q}, \quad \text{(2a)}$$

$$\hat{\mathbf{y}}_{t|t-1} = \mathbf{H} \cdot \hat{\mathbf{x}}_{t|t-1}, \quad \mathbf{S}_{t|t-1} = \mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top + \mathbf{R}. \quad \text{(2b)}$$

In the second step, the *a posteriori* moments are updated based on the *a priori* moments. The pivot computation for this step is the Kalman gain $\mathcal{K}_t$:

$$\mathcal{K}_t = \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top \cdot \mathbf{S}_{t|t-1}^{-1}. \quad \text{(3)}$$

Given the new observation $\mathbf{y}_t$, the state estimate, i.e., the first-order posterior, is obtained via

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathcal{K}_t \cdot \Delta\mathbf{y}_t, \quad \Delta\mathbf{y}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}. \quad \text{(4)}$$

The second-order posterior, which is the estimation error covariance, is then computed as

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t|t-1} - \mathcal{K}_t \cdot \mathbf{S}_{t|t-1} \cdot \mathcal{K}_t^\top. \quad \text{(5)}$$

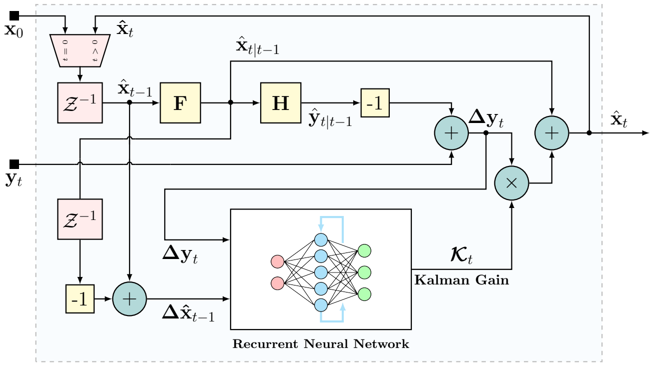

Fig. 1: KalmanNet block diagram.

When the noise is Gaussian and the SS model parameters (1) are fully known, the KF is the MMSE estimator.

2.2. Data-Driven Filtering with KalmanNet

In practice, the state evolution model (1a) is determined by the complex dynamics of the underlying system, while the observation model (1b) is dictated by the type and quality of the observations. For instance, $\mathbf{x}_t$ can be the location, velocity, and acceleration of a vehicle, while $\mathbf{y}_t$ are measurements obtained from several sensors. For real world problems it is often difficult to characterize the SS model accurately. In data-driven filtering, one relies on a labeled data set—i.e., $\mathbf{y}_t$ with its corresponding ground truth $\mathbf{x}_t$—to fill the information gap.

KalmanNet [18] is a DNN-aided architecture for state estimation and filtering with partial domain knowledge. It considers filtering without knowledge of the noise covariance matrices and with a possibly inaccurate description of the evolution matrix $\mathbf{F}$ and the observation model $\mathbf{H}$, obtained from, e.g., understanding of the system dynamics and the sensing setup. Although $\mathbf{R}$, the observation covariance, is not needed for state estimation with KalmanNet, for simplicity of derivation, here we assume that it is known (or estimated) and thus set $\mathbf{R} = \mathbf{I}_n$, since one can always process $\mathbf{R}^{-1/2}\mathbf{y}_t$ without altering the achievable mean-squared error (MSE).

KalmanNet [18] implements data-driven filtering by augmenting the theoretically solid flow of the MB KF with a RNN. The latter is designed to learn to estimate the KG, whose MB computation encapsulates the missing domain knowledge, from data. The inherent memory of the RNN allows implicit tracking of the second-order statistical moments without requiring knowledge of the underlying noise statistics. In a similar manner as the KF, it predicts the first-order moments $\hat{\mathbf{x}}_{t|t-1}$ and $\hat{\mathbf{y}}_{t|t-1}$, which are then used to estimate $\mathbf{x}_t$ via (4). The overall system, illustrated in Fig. 1, is trained end-to-end to minimize the MSE between $\hat{\mathbf{x}}_t$ and the true $\mathbf{x}_t$.

3. UNCERTAINTY IN KALMANNET

KalmanNet, detailed in Subsection 2.2, is designed and trained to estimate the state variable $\mathbf{x}_t$. Neither its architecture nor the loss measure it uses for training encourage KalmanNet to maintain an estimate of its error covariance, which the KF provides. However, as we show here, the fact that KalmanNet preserves the flow of the KF allows to estimate its error covariance from its internal features.

3.1. Kalman Gain-based Error Covariance

To extend KalmanNet to provide uncertainty, we build on top of the interpretable feature of KalmanNet, which is the estimated KG. Combined with the observation model, the KG can be used to compute the time-dependent, error covariance matrix $\boldsymbol{\Sigma}_t$, thus bypassing the need to explicitly estimate the evolution model, as stated in the following theorem:

Theorem 1. *Consider the SS model* (1) *where* $\mathbf{H}$ *has full column rank; i.e.,* $\tilde{\mathbf{H}} = \left(\mathbf{H}^\top \cdot \mathbf{H}\right)^{-1}$ *exists. Then, filtering with the KG* $\mathcal{K}_t$ (3) *results in estimation with error covariance*

$$\boldsymbol{\Sigma}_t = \left(\mathbf{I}_m - \mathcal{K}_t \cdot \mathbf{H}\right) \cdot \tilde{\mathbf{H}} \cdot \mathbf{H}^\top \left(\mathbf{I}_n - \mathbf{H} \cdot \mathcal{K}_t\right)^{-1} \cdot \mathbf{H} \cdot \mathcal{K}_t \cdot \mathbf{H} \cdot \tilde{\mathbf{H}}. \qquad (6)$$

Proof. By combining (3) and (5), the error covariance can be written as $\boldsymbol{\Sigma}_t = \left(\mathbf{I}_m - \mathcal{K}_t \cdot \mathbf{H}\right) \boldsymbol{\Sigma}_{t|t-1}$. Thus, to estimate $\boldsymbol{\Sigma}_t$, we express $\boldsymbol{\Sigma}_{t|t-1}$ using $\mathcal{K}_t$ and the available domain knowledge. To that end, multiplying (3) by $\mathbf{H}$ gives

$$\mathbf{H} \cdot \mathcal{K}_t = \mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top \cdot \mathbf{S}_{t|t-1}^{-1}. \qquad (7)$$

Next, (7) is multiplied by $\mathbf{S}_{t|t-1}$ from the right side, followed by substitution of (2b), which yields

$$\mathbf{H} \cdot \mathcal{K}_t \cdot \left(\mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top + \mathbf{R}\right) = \mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top. \qquad (8)$$

Combining terms in (8) results in

$$\mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top = \left(\mathbf{I}_n - \mathbf{H} \cdot \mathcal{K}_t\right)^{-1} \cdot \mathbf{H} \cdot \mathcal{K}_t. \qquad (9)$$

Multiplying (9) from the left by $\mathbf{H}^\top$ and from the right by $\mathbf{H}$ yields $\mathbf{H}^\top \cdot \mathbf{H} \cdot \boldsymbol{\Sigma}_{t|t-1} \cdot \mathbf{H}^\top \cdot \mathbf{H} = \mathbf{H}^\top \left(\mathbf{I}_n - \mathbf{H} \cdot \mathcal{K}_t\right)^{-1} \cdot \mathbf{H} \cdot \mathcal{K}_t \cdot \mathbf{H}$, which, when $\tilde{\mathbf{H}} = \left(\mathbf{H}^\top \cdot \mathbf{H}\right)^{-1}$ exists, results in

$$\boldsymbol{\Sigma}_{t|t-1} = \tilde{\mathbf{H}} \cdot \mathbf{H}^\top \left(\mathbf{I}_n - \mathbf{H} \cdot \mathcal{K}_t\right)^{-1} \cdot \mathbf{H} \cdot \mathcal{K}_t \cdot \mathbf{H} \cdot \tilde{\mathbf{H}}, \qquad (10)$$

concluding the proof of the theorem. □

Theorem 1 indicates that when the observation model is known and $\mathbf{H}$ has full column rank—requiring that the number of measurements not be smaller than the number of tracked states—then one can extend KalmanNet to predict its error covariance alongside its state estimation. The resulting procedure is summarized as Algorithm 1.

Algorithm 1: KalmanNet with error prediction

```
Init: Trained KalmanNet, previous estimate $\hat{\mathbf{x}}_{t-1}$,
      incoming observations $\mathbf{y}_t$;
Filtering: Apply KalmanNet to estimate $\hat{\mathbf{x}}_t$;
Error prediction: Estimate $\hat{\boldsymbol{\Sigma}}_t$ via (6) with $\mathcal{K}_t$ being
the output of the internal RNN of KalmanNet;
Output: Predicted state $\hat{\mathbf{x}}_t$ and error $\hat{\boldsymbol{\Sigma}}_t$
```

3.2. Discussion

KalmanNet [18] was designed for real-time tracking in non-linear SS models with unknown noise statistics. Algorithm 1 extends it to provide not only $\hat{\mathbf{x}}_t$, an estimate of the state—i.e., for which it was previously derived and trained—but also $\hat{\boldsymbol{\Sigma}}_t$, an estimate of the time-dependent covariance matrix as an uncertainty measure. This is achieved since KalmanNet retains the interpretable flow of the MB KF, where the KG is learned by a dedicated RNN. Numerical evaluations presented in Section 4 indicate that KalmanNet computes the true covariance matrix as it reflects the true empirical error—as expected from the MSE bias variance decomposition theorem [22].

For simplicity and clarity of exposition, Algorithm 1 was derived for linear SS models. In Section 4 we also present results for a non-linear chaotic system, where the extended KF derivation is used; i.e., the SS model matrices are replaced with their respecting Jacobians [2, Ch. 10]. The additional requirement—that observation noise covariance $\mathbf{R}$ is known and needed only for the purpose of estimating the covariance—is often satisfied, as in many applications the main modelling challenge is related to the state evolution rather than the measurement noise. Since we assume that we have access to a labeled data set, one can estimate $\mathbf{R}$ with standard techniques without assumptions regarding the evolution. The case where $\mathbf{H}$ is not full column rank is left for future work.

4. NUMERICAL EVALUATIONS

In this section we numerically[1] compare the covarince computed by KalmanNet to the one computed by the KF in the linear and non-linear cases.

We start with the linear Gaussian SS model, for which the KF achieves the MMSE lower bound, and is an unbiased estimator [23]. We generate data from a scalar SS model where $\mathbf{H} = \mathbf{Q} = \mathbf{R} = 1$ and $\mathbf{F} = 0.9$, with sequence length of 100 samples, and compare the error predicted by KalmanNet and its deviation from the true error to those computed by the MB KF that knows the SS model. We observe in Fig. 2 that the theoretical error computed by KF for each time step t and the error predicted by KalmanNet coincide, and that both al-

[1] The source code along with additional information on the numerical study can be found online at `https://github.com/KalmanNet/ERRCOV_ICASSP22`.

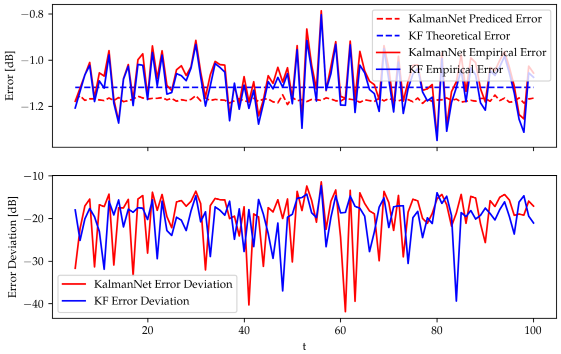

Fig. 2: Average error, linear SS model.

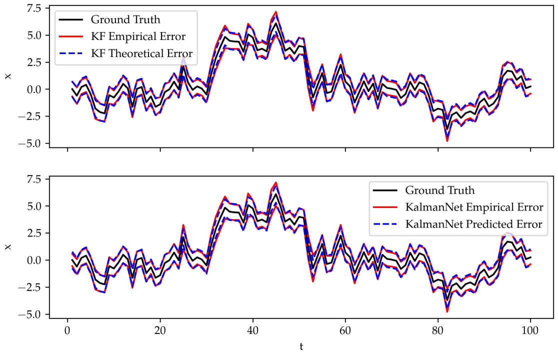

Fig. 3: Single trajectory, linear SS model.

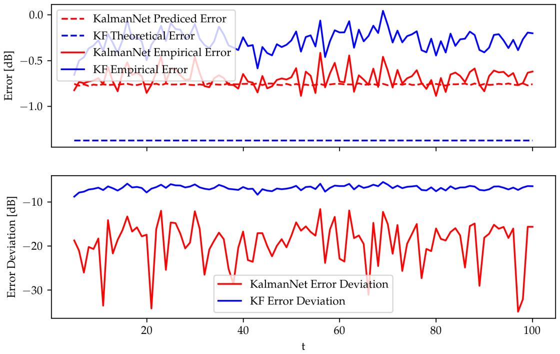

Fig. 4: Average error, linear SS model with mismatch.

gorithms have a similar empirical error. In Fig. 3 we demonstrate that for a single realization of ground truth trajectory, both algorithms produce the same uncertainty bounds. Next, we consider the case where both KalmanNet and the MB KF are plugged in with a mismatched model parameter $\mathbf{F} = 0.5$. In Figs. 4 and 5 we observe that for such a model mismatch, KalmanNet produces uncertainty similar to the empirical error, while the KF underestimates its empirical error.

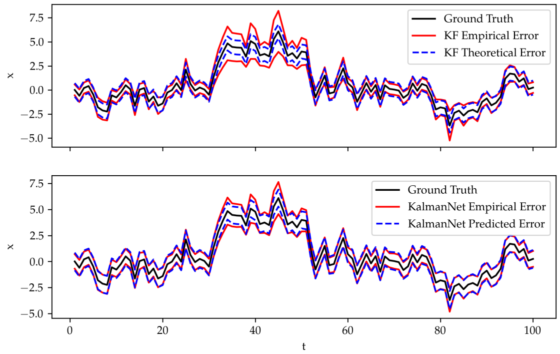

Fig. 5: Single trajectory, linear SS model with mismatch.

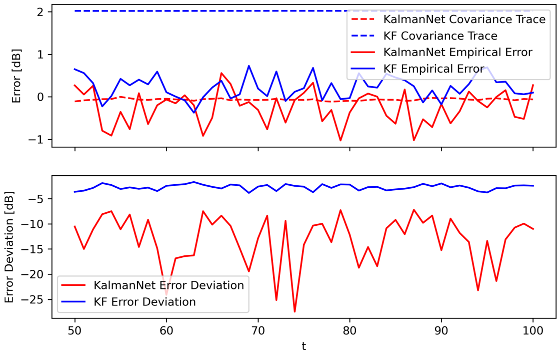

Fig. 6: Average error, non-linear Lorenz attractor model.

Next, we demonstrate the merits of KalmanNet when filtering the Lorenz attractor—a challenging three-dimensional non-linear chaotic system—and compare its performance to the extended KF. See [18] for a detailed description of this setup. The model mismatch in this case is due to sampling a continuous-time system characterized by differential equations to discrete-time. In Fig. 6 we clearly observe that KalmanNet achieves a lower MSE and estimates its error fairly accurately, while the extended KF overestimates it.

5. CONCLUSIONS

In this work we extended the recently proposed KalmanNet state estimator to predict its error alongside the latent state. This is achieved by exploiting the hybrid model-based/data-driven architecture of KalmanNet, which produces the KG as an internal feature. We prove that one can often utilize the learned KG to predict the error covariance as a measure of uncertainty. Our numerical results demonstrate that this extension allows KalmanNet to accurately predict both the state and error, improving upon the KF in the presence of model-mismatch and non-linearities.

6. REFERENCES

[1] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] James Durbin and Siem Jan Koopman, *Time series analysis by state space methods*, Oxford University Press, 2012.

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *preprint arXiv:1412.3555*, 2014.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *preprint arXiv:1706.03762*, 2017.

[5] Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C James Taylor, and Gerhard Neumann, "Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces," in *International Conference on Machine Learning*, 2019, pp. 544–552.

[6] Manzil Zaheer, Amr Ahmed, and Alexander J Smola, "Latent LSTM allocation: Joint clustering and nonlinear dynamic modeling of sequence data," in *International Conference on Machine Learning*, 2017, pp. 3967–3976.

[7] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[8] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia, "Quantitative evaluation of confidence measures in a machine learning world," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5228–5237.

[9] Ian Osband, Zheng Wen, Mohammad Asghari, Morteza Ibrahimi, Xiyuan Lu, and Benjamin Van Roy, "Epistemic neural networks," *preprint arXiv:2107.08924*, 2021.

[10] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun, "Hands-on Bayesian neural networks–a tutorial for deep learning users," *preprint arXiv:2007.06823*, 2020.

[11] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt, "Deep variational Bayes filters: Unsupervised learning of state space models from raw data," *preprint arXiv:1605.06432*, 2016.

[12] Rahul Krishnan, Uri Shalit, and David Sontag, "Structured inference networks for nonlinear state space models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, vol. 31.

[13] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei, "Variational sequential Monte Carlo," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 968–977.

[14] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Y Ng, and Sebastian Thrun, "Discriminative training of Kalman filters," in *Robotics: Science and Systems*, 2005, vol. 2, p. 1.

[15] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel, "Backprop KF: Learning discriminative deterministic state estimators," in *Advances in Neural Information Processing Systems*, 2016, pp. 4376–4384.

[16] Bracha Laufer-Goldshtein, Ronen Talmon, and Sharon Gannot, "A hybrid approach for speaker tracking based on TDOA and data-driven models," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 4, pp. 725–735, 2018.

[17] Liang Xu and Ruixin Niu, "EKFNet: Learning system noise statistics from measurement data," in *Proc. IEEE ICASSP*, 2021, pp. 4560–4564.

[18] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG van Sloun, and Yonina C Eldar, "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *preprint arXiv:2107.10043*, 2021.

[19] Nir Shlezinger, Jay Whang, Yonina C Eldar, and Alexandros G Dimakis, "Model-based deep learning," *preprint arXiv:2012.08405*, 2020.

[20] Adrià López Escoriza, Guy Revach, Nir Shlezinger, and Ruud J G van Sloun, "Data-driven Kalman-based velocity estimation for autonomous racing," in *Proc. IEEE ICAS*, 2021.

[21] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan, *Estimation with applications to tracking and navigation: Theory algorithms and software*, John Wiley & Sons, 2004.

[22] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in Statistics, New York, 2001.

[23] Jeffrey Humpherys, Preston Redd, and Jeremy West, "A fresh look at the Kalman filter," *SIAM Review*, vol. 54, no. 4, pp. 801–823, 2012.