LUNA: GENERALIZABLE AND TRUSTWORTHY STATE ESTIMATION WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

State estimation infers latent states from noisy and partial observations to provide both state estimates and uncertainty quantification (UQ). Classical Kalman-family filters (KFs) heavily depend upon the adopted system transition and observation dynamic models, suffering from low generalization. Pretraning-model based learnable state estimators aim for high generalization, yet with untrustworthy state estimates due to missed UQ. Particularly, offering UQ in LLM-based state estimators is hard due to missing UQ sources. To this end, in this paper, we propose an LLM-based UNcertainty-aware stAte estimation framework, namely LUNA, to offer trustworthy estimates with high generalization. Moreover, LUNA develops three stages to obtain the principled UQ as classic estimators from both observed systems and LLMs. Experiments on six synthetic and real-world benchmarks show that LUNA achieves the lowest RMSE on five benchmarks and the lowest NLL on four under in-domain evaluation, reducing RMSE by 71.8% on average relative to LLMFilter. Under cross-system evaluation, LUNA reduces RMSE by 41.3% on average relative to the best-performing external baseline. The code is available at an anonymous repository[1].

1 INTRODUCTION

State estimation infers latent states from noisy and partial observations. It is central to many applications such as robotics, autonomous driving, and weather prediction (Talbot et al., 2025; Dahal et al., 2024; Allen et al., 2025). As illustrated in Fig. 1, the robot tracking application takes the observation of noisy position measurements as input and then estimates its positions and velocities. Beyond a single state value, uncertainty quantification (UQ) of the estimated states, e.g., the covariance of the estimated position or velocity, is important for reliable control and informed decision-making.

Classic state estimators, e.g., Kalman Filtering, adopt recursive prediction-update steps to obtain the state distribution with the mean $\hat{\mathbf{x}}_{t|t}$ and covariance $\mathbf{\Sigma}_{t|t}$. As shown in Figure 1, the *prediction step* takes the previous state $(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{\Sigma}_{t-1|t-1})$ as input to obtain the current state $(\hat{\mathbf{x}}_{t|t-1}, \mathbf{\Sigma}_{t|t-1})$ via a transition model. The *update step* then employs the current observation $\mathbf{y}_t$ through an observation model to obtain the corrected estimate $(\hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t})$. Yet, these filters require expert knowledge to build the transition and observation models of the observed systems, and do not perform well on partially or fully systems (e.g., cross-domain state estimation) if without the associated transition and observation models. For better readability, we use $(\hat{\mathbf{x}}_{t-1}, \mathbf{\Sigma}_{t-1})$ to denote the previous state $(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{\Sigma}_{t-1|t-1})$, and $(\hat{\mathbf{x}}_t, \mathbf{\Sigma}_t)$ to the current state.

To offer high generalization, the literature has developed learnable state estimators. Data-driven neural filters, such as KalmanNet (Revach et al., 2022), learn the transition and observation models of observed systems, but still with two issues: the learned models are heavily tied to training data, and do not offer UQ. Inspired by the powerful capability of pretrained models to perform the next-token prediction, some recent work has explored pretrained sequential mode or language models for current state estimation. For example, FM-UKF (Holtmann et al., 2025) exploits a pretrained sequential transformer model to replace the original prediction step within the classic Unscented Kalman Filter (UKF) framework. However, this pretraining model based prediction step does not provide any UQ, and the UQ of the entire FM-UKF completely depends upon the UKF update

[1] https://anonymous.4open.science/r/LUNA-V1AC

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

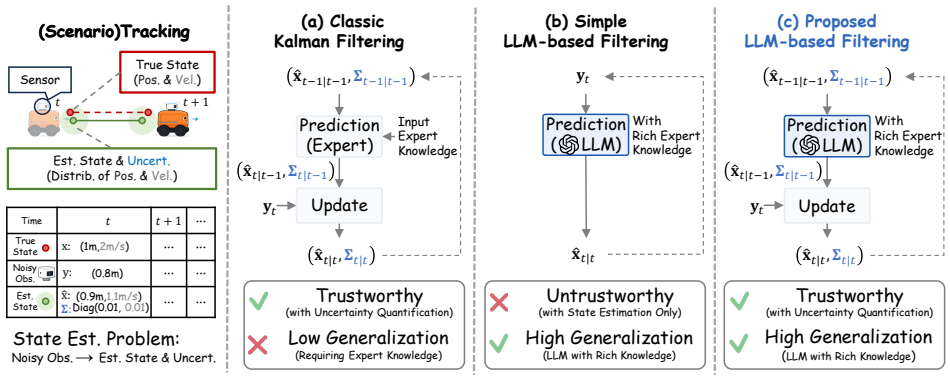


Figure 1: Motivation of LUNA

step alone. LLMFILTER (Liu et al., 2025) exploits large-language models (LLMs) to predict state estimates at the current time step, but missing the update step and UQ. Although with the few-short or even zero-shot prediction capability, FM-UKF and LLM-Filter may not obtain trustworthy estimates due to insufficient or missed UQ.

Providing UQ in LLM-based state estimators is hard. For example, when LLMFILTER takes history observations as input to LLMs for current state estimates, we may follow literature works of LLM UQ, e.g., (Manakul et al., 2023), by multiple-rounds of state estimation to obtain the associated state estimates and then have the covariance of such estimates. Such UQ suffers from the following challenges. Firstly, the obtained covariance does not differentiate the UQ sources from either observation systems or LLMs. Secondly, history observations are essentially point measurements at certain time steps. Thus, such covariance is insufficient to identify the UQ of measurement noise or observed systems. Finally, existing LLMs are still difficult to explain and interpret continuous data distributions and data patterns (Zhong, 2025). When given the previous state estimate $(\hat{\mathbf{x}}_{t-1}, \mathbf{\Sigma}_{t-1})$ as input to LLMs, how to represent and process it within LLMs is non-trivial.

To overcome the challenges above, in this paper, we propose an LLM-based state estimation framework, namely LUNA. Specifically, LUNA exploits the LLM backbone to predict the estimates of the current step state with both mean and uncertainty in the prediction step, and the update step then corrects the predicted states to finally obtain the current state $(\hat{\mathbf{x}}_t, \mathbf{\Sigma}_t)$. Hence, LUNA can benefit from LLMs with high generalization. More importantly, as classic KF estimators do, LUNA provides the principled UQ from three sources by the following stages. In the 1st stage of *state transition UQ*, LUNA deterministically samples Sigma-points from the previous state distribution $(\hat{\mathbf{x}}_{t-1}, \mathbf{\Sigma}_{t-1})$, then exploits LLM backbones to perform state transition over such data points. In this way, LUNA does not require multiple-rounds of processing overhead to verbalize the continuous state distribution as (Zhong, 2025). Meanwhile, the sampled Sigma-points can capture the UQ of the previous state distribution $\mathbf{\Sigma}_{t-1}$. Next, in the 2nd stage of *LLM UQ*, LUNA then estimates the uncertainty induced by LLM itself, involving two sources of LLM uncertainties: aleatoric and epistemic components $\mathbf{\Sigma}^{\text{ale}}_{t|t-1}$ and $\mathbf{\Sigma}^{\text{epi}}_{t|t-1}$. Third, by the Law of Total Covariance (LTC), this stage incorporates the three UQ together to reconstruct the final predictive uncertainty $\mathbf{\Sigma}_{t|t-1}$. After the three-stage prediction step, LUNA performs the update step to obtain the final corrected state estimate $(\hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t})$. In summary, we make the following contributions.

- To our best knowledge, LUNA is the first to offer trustworthy state estimates meanwhile with high generalization on LLM-backbones.
- LUNA can perform LLM-based state transition over sampled Sigma-points and obtain uncertainty quantification from both observed systems and LLMs.
- Experiments on six synthetic and real-world tasks demonstrate the supervisory of LUNA in term of estimation accuracy, probabilistic consistency, and computational cost.

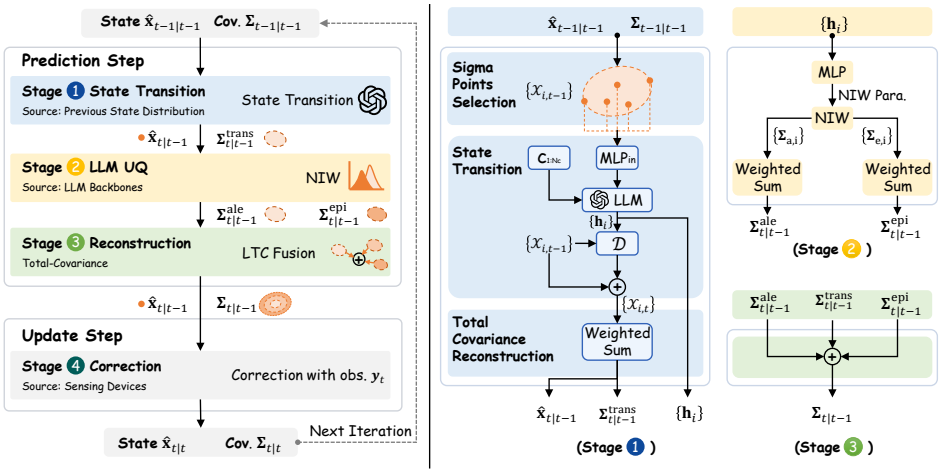


Figure 2: Left: (a) Overall Workflow; Right: (b) Three Detailed Stages.

2 Related Work

Explicit Model-based State Estimation. Bayesian Filters provide a general prediction-update framework for state estimation. The Kalman-Filter (KF) family exploits the recursive steps to predict and update states in the form of the mean-covariance pair (Chen et al., 2003), e.g., the linear or locally approximated dynamics (Grewal, 2025), Sigma-point filters (Merwe & Wan, 2004), particle filters (Djuric et al., 2003), and adaptive variants (Li et al., 2016; Zhang et al., 2020). These filters typically require the explicit system transition and observation models, and provide principled uncertainty quantification (UQ). Yet, when the used transition and observation models are fully or partially inapplicable to target systems, the filters may not perform well.

Learnable State Estimation. To overcome the low generalization issue above, literature works have exploited either deep learning or pretrained models for state estimation. The data-driven neural filters, e.g., KalmanNet (Revach et al., 2022) and WALNUT (Zhou et al., 2024), learn such system models from training data. Yet, these neural filters either suffer from low generalization to unknown systems without sufficient training data, or may not offer UQ at all. Recent works explore pretrained models for state estimation, depending upon how such models are used in the prediction-update framework. Within the UKF framework, FM-UKF (Holtmann et al., 2025) replaces the prediction step by a pretrained sequence model. LLMFilter (Liu et al., 2025) directly exploits LLMs to perform the prediction step but without the update step and UQ at all. Despite their generalization capabilities, LLMFilter does not provide explicit UQ, while FM-UKF provides limited UQ without explicitly modeling the uncertainty introduced by its pretrained sequence model.

Uncertainty Quantification (UQ) of LLMs. UQ is important to black-box LLMs and deep neural networks (DNNs). The classic DNN UQ techniques include MC dropout, ensembles and iterative estimation (He et al., 2026; Durasov et al., 2024). To tackle the hallucination issue of LLMs, the literature has developed various UQ methods, e.g., verbalizing methods (Xiong et al., 2024), information-theoretic methods (Manakul et al., 2023), consistency-based methods (Manakul et al., 2023) and recent semantic clustering methods (Farquhar et al., 2024). However, such LLM UQ approaches may not be comfortably used for the LLM-based state estimation problem. For example, as shown in the work (Zhong, 2025), LLMs are hard to explain and interpret the continuous data distribution that is required in our state estimation.

3 Methodology

3.1 Framework Overview

As shown in Figure 2a, LuNA follows the recursive prediction-update steps. The prediction step involves three stages. The LLM backbone firstly takes the previous state estimate $(\hat{\mathbf{x}}_{t-1}, \mathbf{\Sigma}_{t-1})$

as inputs, and obtains the transition state $(\hat{\mathbf{x}}_{t|t-1}, \mathbf{\Sigma}^{\text{trans}}_{t|t-1})$. Here, the transition uncertainty $\mathbf{\Sigma}^{\text{trans}}_{t|t-1}$ is dominated by the previous one $\mathbf{\Sigma}_{t-1}$. Secondly, LUNA then estimates the uncertainty induced by LLM itself, involving two sources of LLM uncertainties: aleatoric and epistemic components $\mathbf{\Sigma}^{\text{ale}}_{t|t-1}$ and $\mathbf{\Sigma}^{\text{epi}}_{t|t-1}$. Thirdly, by the Law of Total Covariance (LTC), this stage incorporates the three UQ together to reconstruct the final predictive uncertainty $\mathbf{\Sigma}_{t|t-1}$. After the three-stage prediction, LUNA performs the update step as usual to obtain the final corrected state estimate $(\hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t})$.

Consider that the update step performs state correction, mainly by the observation model of the sensing devices to observe target systems. Since these sensing devices are typically provided by the owners of state estimators, we reasonably assume that the sensor observation model is completely known to state estimators, consistent with many previous works, e.g., FM-UKF (Holtmann et al., 2025). Moreover, the prediction step mainly depends upon the transition model of the observed target systems, and mostly these systems are either completely or partially unknown to state estimators. Thus, we focus on the prediction step in this paper, and give the details of three prediction stages in the rest of this section.

3.2 STAGE 1: STATE TRANSITION

This stage aims to transit the previous state distribution into the next one by LLMs. Though powerful for statistical text analysis with derived sequences of next-token probability distributions, LLMs are still hard to explain continuous data distributions and data patterns. Thus, in Figure 2b, LUNA does not explicitly process the state distribution within LLMs, and exploits the following pipeline to deterministically sample Sigma-points from the given distribution, perform the LLM-based state transition over such data points, and finally give the transition states. In this way, LUNA can perform state estimation well with various data distribution to offer high generalization and trustworthiness.

(*1*) **Sigma-Point Selection.** Firstly, we adopt the Sigma-Point method (Merwe & Wan, 2004) to select data points from the previous state distribution $(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{\Sigma}_{t-1|t-1})$. The Sigma-Point method involves deterministic sampling based approximations of the relevant Gaussian statistics to achieve "second-order" or higher accuracy. That is, LUNA deterministically samples $(2d+1)$ data points $\{\boldsymbol{\chi}_{i,t-1}\}_{i=0}^{2d} \in \mathbb{R}^d$ from the given distribution. The weighted mean and covariance of such data points then match the given distribution. Each point $\boldsymbol{\chi}_{i,t-1} \in \mathbb{R}^d$ is thus a deterministic state-space query, and LLM then performs the state transition over the point query $\boldsymbol{\chi}_{i,t-1}$.

(*2*) **Sigma-Point State Transition.** This transition involves (*a*) the contextualization of a point query $\boldsymbol{\chi}_{i,t-1}$ and a global system context into token embeddings, and (*b*) LLM transition to map the token embeddings into a one-step state change $\mathcal{D}(\cdot)$ between the previous state and current one. LUNA then exploits this change $\mathcal{D}(\cdot)$ and input query $\boldsymbol{\chi}_{i,t-1}$ to obtain the transition state $\boldsymbol{\chi}_{i,t}$.

(*2a*) CONTEXTUALIZATION: Following (Liu et al., 2025), the system context comprises task instructions and in-context examples (see Appendix B.3), and we denote its token embeddings by $\mathbf{C}_{1:N_c}$. After each sigma point query $\boldsymbol{\chi}_{i,t-1}$ is mapped to one soft token, LUNA then appends it to $\mathbf{C}_{1:N_c}$. The LLM backbone next takes these tokens as input to give the output embeddings $\mathbf{h}_{i,t|t-1}$:

$$\mathbf{h}_{i,t|t-1} = \left[\text{LLM}\left([\mathbf{C}_{1:N_c}; \text{MLP}_{\text{in}}(\boldsymbol{\chi}_{i,t-1})]\right)\right]_{N_c+1}. \tag{1}$$

(*2b*) TRANSITION: Given the embeddings above $\mathbf{h}_{i,t|t-1}$ and the query point $\boldsymbol{\chi}_{i,t-1}$, this step exploits a residual network $\mathcal{D}(\cdot)$ to generate the point $\boldsymbol{\chi}_{i,t}$ at the next time step.

$$\boldsymbol{\chi}_{i,t} = \boldsymbol{\chi}_{i,t-1} + \mathcal{D}\left(\boldsymbol{\chi}_{i,t-1}, \mathbf{h}_{i,t|t-1}\right), \tag{2}$$

Instead of directly giving $\boldsymbol{\chi}_{i,t}$, this network $\mathcal{D}$ learns the one-step state change between $\boldsymbol{\chi}_{i,t-1}$ and $\boldsymbol{\chi}_{i,t}$ for better performance optimization (He et al., 2016). LUNA implements $\mathcal{D}$ as follows.

$$\Delta\boldsymbol{\chi}^{\text{raw}}_{i,t|t-1} = \text{MLP}_{\text{inc}}(\mathbf{h}_{i,t|t-1}), \qquad \boldsymbol{\delta}_{i,t|t-1} = \text{MLP}_{\text{corr}}\left(\boldsymbol{\chi}_{i,t-1}, \text{sg}[\Delta\boldsymbol{\chi}^{\text{raw}}_{i,t|t-1}]\right), \tag{3a}$$

$$\mathcal{D}(\boldsymbol{\chi}_{i,t-1}, \mathbf{h}_{i,t|t-1}) = \Delta\boldsymbol{\chi}^{\text{raw}}_{i,t|t-1} + \boldsymbol{\delta}_{i,t|t-1}. \tag{3b}$$

In Eq. (3a), the delta $\Delta\boldsymbol{\chi}^{\text{raw}}_{i,t|t-1} \in \mathbb{R}^d$ is first decoded from $\mathbf{h}_{i,t|t-1}$ by MLP_{inc}, and the correction $\boldsymbol{\delta}_{i,t|t-1} \in \mathbb{R}^d$ is then produced by MLP_{corr} from the query point $\boldsymbol{\chi}_{i,t-1}$ and the delta $\Delta\boldsymbol{\chi}^{\text{raw}}_{i,t|t-1}$.

Here, to prevent representation collapse, a stop-gradient operator $\mathrm{sg}[\cdot]$ is used to pass $\Delta\boldsymbol{\chi}^{\mathrm{raw}}_{i,t|t-1}$ unchanged during the forward pass but blocks gradients from flowing back through it during back-propagation. Eq. (3b) lastly yields the output by the addition of $\Delta\boldsymbol{\chi}^{\mathrm{raw}}_{i,t|t-1}$ and $\boldsymbol{\delta}_{i,t|t-1}$.

(*3*) **Transition State Reconstruction.** This step reconstructs the LLM transition state estimate by the weighted sum of LLM transition sigma-points $\boldsymbol{\chi}_{i,t}$.

$$\hat{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2d} W_i^{(x)} \boldsymbol{\chi}_{i,t}, \qquad \boldsymbol{\Sigma}^{\mathrm{trans}}_{t|t-1} = \sum_{i=0}^{2d} W_i^{(\Sigma)} \left(\boldsymbol{\chi}_{i,t} - \hat{\mathbf{x}}_{t|t-1}\right)\left(\boldsymbol{\chi}_{i,t} - \hat{\mathbf{x}}_{t|t-1}\right)^{\top}. \tag{4}$$

Here, $W_i^{(\Sigma)}$ and $W_i^{(x)}$ are two deterministic covariance weights given by the previous 1st step, i.e., Sigma-Point method. Because the weights $W_i^{(\Sigma)}$ and $W_i^{(x)}$ and Sigma-Points $\boldsymbol{\chi}_{i,t}$ all heavily depend upon the previous state distribution, it is not hard to find that the reconstructed uncertainty $\boldsymbol{\Sigma}^{\mathrm{trans}}_{t|t-1}$ is mainly dominated by the previous one $\boldsymbol{\Sigma}_{t-1|t-1}$. Yet, the transition uncertainty caused by LLM itself is still unknown, and will be given in the following subsection.

3.3 Stage 2: LLM-induced Uncertainty Estimate

Given the LLM transition embeddings $\mathbf{h}_{i,t|t-1}$ generated from the $(2d+1)$ Sigma-Points by Eq. (1), this stage exploits Normal-Inverse-Wishart (NIW) evidential regression model (Meinert et al., 2023) to give the uncertainty caused by this LLM transition. Unlike standard regression with a point estimate output, this evidential regression model learns the distribution with the mean and variance parameters of a Gaussian reconstruction likelihood. To this end, this 2nd stage first learns the NIW parameters from each embedding $\mathbf{h}_{i,t|t-1}$. The NIW parameters are then used to compute the local aleatoric and epistemic uncertainty estimates regarding this $\mathbf{h}_{i,t|t-1}$ (a.k.a the Sigma-Point $\boldsymbol{\chi}_{i,t-1}$). After that, for the $(2d+1)$ embeddings (and Sigma-Points), this stage marginalizes the associated local uncertainty estimates to give the overall uncertainty estimates $\boldsymbol{\Sigma}^{\mathrm{ale}}_{t|t-1}$ and $\boldsymbol{\Sigma}^{\mathrm{epi}}_{t|t-1}$.

Local NIW Parameterization. From each Sigma-Point embedding $\mathbf{h}_{i,t|t-1}$, LUNA first learns three local NIW parameters and computes its local aleatoric and epistemic uncertainty estimate:

$$(\kappa_i, \nu_i, \mathbf{L}_i) = \mathrm{MLP}_{\mathrm{NIW}}(\mathbf{h}_{i,t|t-1}), \quad \boldsymbol{\Psi}_i = \mathbf{L}_i \mathbf{L}_i^{\top}, \tag{5a}$$

$$\boldsymbol{\Sigma}^{\mathrm{ale}}_{i,t|t-1} = \frac{\boldsymbol{\Psi}_i}{\nu_i - d - 1}, \qquad \boldsymbol{\Sigma}^{\mathrm{epi}}_{i,t|t-1} = \frac{\boldsymbol{\Psi}_i}{\kappa_i(\nu_i - d - 1)}. \tag{5b}$$

Here, the NIW parameters κ_i and ν_i indicate the mean-precision and degrees-of-freedom, respectively. Note that the standard NIW regression model directly gives the output scale matrix $\boldsymbol{\Psi}_i$. This $\mathrm{MLP}_{\mathrm{NIW}}$ instead gives the matrix $\mathbf{L}_i$ and then computes $\boldsymbol{\Psi}_i = \mathbf{L}_i\mathbf{L}_i^{\top}$ with the main purpose: by imposing $\kappa_i > 0$ and $\nu_i > d+1$, LUNA can parameterize $\mathbf{L}_i$ as a lower-triangular matrix with a positive diagonal to ensure $\boldsymbol{\Psi}_i \succ 0$. In addition, $\boldsymbol{\Sigma}^{\mathrm{ale}}_{i,t|t-1}$ represents data-dependent transition variability, while $\boldsymbol{\Sigma}^{\mathrm{epi}}_{i,t|t-1}$ represents the uncertainty in the learned transition (Meinert et al., 2023).

Sigma-Point Marginalization. Now, given the local uncertainty estimates from $(2d+1)$ Sigma-Point embeddings, LUNA marginalizes them using the state-estimate weights:

$$\boldsymbol{\Sigma}^{\mathrm{ale}}_{t|t-1} = \sum_{i=0}^{2d} W_i^{(x)} \boldsymbol{\Sigma}^{\mathrm{ale}}_{i,t|t-1}, \qquad \boldsymbol{\Sigma}^{\mathrm{epi}}_{t|t-1} = \sum_{i=0}^{2d} W_i^{(x)} \boldsymbol{\Sigma}^{\mathrm{epi}}_{i,t|t-1}. \tag{6}$$

Since $\boldsymbol{\Sigma}^{*}_{i,t|t-1} \succeq 0$ for all i, and $W_i^{(x)} \geq 0$ with $\sum_{i=0}^{2d} W_i^{(x)} = 1$, the weighted covariances $\boldsymbol{\Sigma}^{\mathrm{ale}}_{t|t-1}$ and $\boldsymbol{\Sigma}^{\mathrm{epi}}_{t|t-1}$ remain positive semidefinite.

3.4 Stage 3: Reconstruct Total-Covariance

In order to give the complete state estimate of the prediction step, besides the mean state estimate $\hat{\mathbf{x}}_{t|t-1}$ in Eq. (4), in this stage, we follow the Law of Total Covariance to incorporate the state transition uncertainty and LLM-induced one and reconstruct the final prediction covariance $\boldsymbol{\Sigma}_{t|t-1}$ as follows (The proof refers to Appendix C.2).

$$\boldsymbol{\Sigma}_{t|t-1} = \underbrace{\boldsymbol{\Sigma}^{\mathrm{trans}}_{t|t-1}}_{\text{Transition covariance}} + \underbrace{\boldsymbol{\Sigma}^{\mathrm{ale}}_{t|t-1} + \boldsymbol{\Sigma}^{\mathrm{epi}}_{t|t-1}}_{\text{LLM-induced covariance}}. \tag{7}$$

3.5 TRAINING OBJECTIVE

Given a sequence of ground truth states of a target system, $\mathbf{x}_0 \cdots \mathbf{x}_{t-1}$, $\mathbf{x}_t \cdots$, we define the state change $\Delta\mathbf{x}_t^* = \mathbf{x}_t - \mathbf{x}_{t-1}$. Let $\Delta\boldsymbol{\chi}_{\mathrm{raw},t}$, $\boldsymbol{\delta}_t$, and $(\kappa_t, \nu_t, \boldsymbol{\Psi}_t)$ denote the outputs of Eqs. (3a), and (5a), respectively. We define the final corrected increment as $\mathcal{D} = \Delta\boldsymbol{\chi}_{\mathrm{raw},t} + \boldsymbol{\delta}_t$. Since the transition model is unknown, supervision is provided by the observed state pairs rather than ground-truth transitions queried at arbitrary sigma points. During filtering, the same learned transition is evaluated at each deterministic query $\boldsymbol{\chi}_{i,t-1}$. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{NIW}}(\mathcal{D}, \kappa_t, \nu_t, \boldsymbol{\Psi}_t), \qquad \mathcal{L}_{\mathrm{NIW}} = \mathcal{L}_{\mathrm{NLL}} + \lambda\mathcal{L}_{\mathrm{reg}}, \tag{8a}$$

$$\mathcal{L}_{\mathrm{reg}} = \frac{\kappa_t + \nu_t}{2}\left(\|\Delta\mathbf{x}_t^* - \Delta\boldsymbol{\chi}_{\mathrm{raw},t}\|_2 + \|\Delta\mathbf{x}_t^* - \mathrm{sg}[\Delta\boldsymbol{\chi}_{\mathrm{raw},t}] - \boldsymbol{\delta}_t\|_2\right). \tag{8b}$$

where the minibatch objective is obtained by averaging the loss over the training transitions. Since $\mathrm{LLM}_{\mathrm{core}}$ is frozen, the UQ reconstruction in Sec. 3.4 does not require trainable parameters. The corrected increment $\mathcal{D}$ and the NIW parameters $(\kappa_t, \nu_t, \boldsymbol{\Psi}_t)$ define a predictive distribution for $\Delta\mathbf{x}_t^*$, with $\mathcal{D}$ as its mean. We optimize its negative log-likelihood together with an evidence regularizer:

The first term in $\mathcal{L}_{\mathrm{reg}}$ anchors the raw increment prediction, whereas the second term couples the residual after correction to the total evidence $\kappa_t + \nu_t$. The factor $1/2$ averages the two residual terms. The coefficient $\lambda > 0$ controls the strength of this evidence regularization (Amini et al., 2020; Meinert et al., 2023). The appendix provides the full form of the negative log-likelihood $\mathcal{L}_{\mathrm{NLL}}$. The overall objective updates $\mathrm{MLP}_{\mathrm{in}}$, $\mathrm{MLP}_{\mathrm{inc}}$, $\mathrm{MLP}_{\mathrm{NIW}}$, and $\mathrm{MLP}_{\mathrm{corr}}$. The stop-gradient in the second term of $\mathcal{L}_{\mathrm{reg}}$ separates the two regularization signals: the first term updates $\mathrm{MLP}_{\mathrm{in}}$ and $\mathrm{MLP}_{\mathrm{inc}}$ to learn the raw increment, whereas the second treats the raw increment as fixed and updates $\mathrm{MLP}_{\mathrm{corr}}$ to learn the residual correction.

4 EXPERIMENTS

We perform the evaluation with three research questions: **RQ1 (Quality):** Does LUNA improve state estimate quality? **RQ2 (Generalization):** Does LUNA remain robust under cross-domain adaptation? **RQ3 (Ablation):** How the adopted NIW evidential regression model and the key LLM transition components effect the performance of LUNA?

4.1 EXPERIMENTAL SETUP

Benchmarks. We systematically evaluate LUNA on six open benchmark datasets (Liu et al., 2025; Revach et al., 2022). Specifically, the datasets comprise two *low-dimensional linear* systems, including the simulated Tracking (Särkkä & Svensson, 2023) and the real-world NCLT robot-localization dataset (Carlevaris-Bianco et al., 2016), two *low-dimensional nonlinear* models, e.g., Oscillator (Basiri & Khanmohammadi, 2024) and Hopf (Hassard et al., 1981), and two standard *high-dimensional chaotic* systems in atmospheric dynamics, e.g., Lorenz96 (Lorenz, 1996) and VL20 (Vissio & Lucarini, 2020). Besides the in-domain system evaluation, we further study the cross-domain generalization, and thus perform bidirectional adaption across the nonlinear systems (Oscillator $\leftrightarrow$ Hopf) and chaotic systems (Lorenz96 $\leftrightarrow$ VL20), respectively. Moreover, we evaluate the robustness against test-time process-noise mismatch on the Oscillator and Hopf benchmarks.

Baselines. We compare LUNA with three following categories of baseline filters. *Classic Filter:* On simulated benchmarks, the Extended Kalman filter (EKF) (Lei, 2025) and UKF both use the ground-truth transition and observation models, with the same process and observation noise covariances used for data generation. *Neural filters:* KalmanNet (Revach et al., 2022) and WALNUT (Zhou et al., 2024) learn neural networks to represent Kalman Filter with the training data from target systems. *Pretraining Filters:* FM-UKF (Holtmann et al., 2025) couples a pretrained dynamics predictor with a UKF measurement update, and LLMFILTER (Liu et al., 2025) directly maps observation histories to state estimates but without an update step. In addition, we purposely enhance LLMFILTER with the UKF measurement update, namely LLMFILTER+UKF, Regarding ours, we employs standard neural networks (MLP and RNN), time-series foundation models (Google TimesFM-2.5 and -3.0 (Das et al., 2024)) and two language models (GPT-2 (Radford et al., 2019), OPT-1.3B (Zhang et al., 2022), and LLaMA-7B (Touvron et al., 2023)) as state prediction backbones, and further develop three variants of LUNA without three uncertainties (the transition one dominated by previous

Table 1: In-Domain Evaluation with RMSE↓ and NLL↓.

	Low-dimensional nonlinear systems				High-dimensional chaotic systems				Low-dimensional linear systems			
	Oscillator		Hopf		Lorenz96		VL20		Tracking		NCLT (real data)	
	RMSE↓	NLL↓	RMSE↓	NLL↓	RMSE↓	NLL↓	RMSE↓	NLL↓	RMSE↓	NLL↓	RMSE↓	NLL↓
Classical Filters												
EKF(Oracle)	0.605	10.193	0.343	1.957	0.470	42.932	0.324	176.685	0.445	1.099	22.591	513.624
UKF(Oracle)	0.529	2.094	0.369	2.315	**0.404**	**18.059**	0.278	16.969	0.445	1.098	22.413	480.376
Neural Filters												
KalmanNet	1.437	-	0.362	-	NaN	-	NaN	-	0.454	-	19.668	-
WALNUT	0.527	1.530	0.254	0.140	NaN	NaN	NaN	NaN	0.452	1.159	18.380	581.492
Pretraining Filters												
FM-UKF	0.468	3.750	0.248	0.256	0.495	54.251	0.377	87.555	0.570	5.616	25.913	193.579
LLMFilter	1.231	-	0.542	-	2.606	-	0.887	-	4.972	-	17.499	-
LLMFilter + UKF	0.836	2.532	0.577	1.814	1.773	264.353	0.714	77.851	2.649	30.262	16.792	118.006
Ours (Standard neural and time-series foundation models)												
MLP	0.401	0.467	0.210	-0.619	0.549	50.523	0.291	3.029	0.443	1.100	18.741	10.040
RNN	0.408	0.529	0.211	-0.606	0.522	44.234	0.287	-5.130	0.447	1.184	17.041	7.916
Transformer	0.402	0.470	0.213	-0.593	0.508	44.846	0.262	-8.209	0.445	1.205	9.895	4.898
TimesFM-2.5 (231M)	0.387	0.347	**0.207**	**-0.654**	0.513	45.336	0.272	-6.480	0.446	1.182	9.628	4.156
TimesFM-3.0 (337M)	0.385	0.330	0.208	-0.639	0.517	45.345	0.276	-5.821	0.448	1.222	8.662	4.008
Ours (Pretrained language-models)												
GPT-2 (124M)	0.402	0.450	0.209	-0.637	0.531	48.917	0.272	-9.789	0.452	1.404	10.381	4.551
OPT-1.3B	0.395	0.403	**0.207**	-0.644	0.465	37.430	**0.252**	-12.831	0.449	1.310	9.984	4.371
Ours (Uncertainty Ablation)												
LUNA w/o Transition	0.700	24.801	0.301	6.274	0.559	63.580	0.308	-4.581	2.044	718.688	8.310	36.023
LUNA w/o Aleatoric	0.475	5.079	0.229	0.027	0.482	71.424	0.254	1.483	0.718	17.212	8.242	3.374
LUNA w/o Epistemic	0.397	0.406	0.223	-0.245	0.433	28.846	0.253	-8.257	0.445	1.115	8.236	3.313
LUNA (LLaMA-7B)	**0.378**	**0.255**	**0.207**	-0.649	0.431	28.132	**0.252**	**-13.336**	**0.442**	**1.086**	**8.151**	**3.099**

states, the aleatoric and epistemic ones). By default, we use LLaMA-7B as the LLM backbone for both LUNA and LLMFILTER. Appendix B.2 summarizes the network architecture and sigma-point settings of LUNA.

Metrics. We evaluate state-estimation performance in terms of point-estimation accuracy and state probability distribution quality. Point-estimation accuracy is measured by the root mean squared error (RMSE) between point state estimate and the ground-truth state. For those methods to provide the state probability distribution with the additional covariance, we use the negative log-likelihood (NLL) of the corresponding Gaussian state distribution $\mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t})$.

4.2 RQ1: IN-DOMAIN PERFORMANCE

Overall Performance. As shown in Table 1, LUNA achieves the lowest NLL on four benchmarks and the lowest RMSE on five. On average across the six benchmarks, LUNA reduces RMSE by 71.8% and 65.7% relative to LLMFilter and LLMFilter+UKF, respectively. UKF remains best on Lorenz96. KalmanNet and WALNUT diverge during training on both chaotic systems (NaN), consistent with previously reported failures of KalmanNet (Liu et al., 2025).

On NCLT, improvements in point accuracy do not necessarily translate into better state distributions. WALNUT improves RMSE over UKF (18.380 versus 22.413) but worsens NLL (581.492 versus 480.376). LUNA improves both metrics, achieving the lowest RMSE of 8.151 and NLL of 3.099.

We replace only the LLM backbone, retaining the surrounding modules and filtering procedure. Under in-domain evaluation, even the MLP variant outperforms LLMFilter+UKF on both metrics across all five simulated systems. These results show that the framework supports effective filtering with relatively simple transition models learned from target-system data. LLaMA-7B provides further gains in most settings, although TimesFM-2.5 matches its RMSE and achieves slightly lower NLL on Hopf. The tested language models exhibit a favorable scaling trend: from GPT-2 (124M) through OPT-1.3B to LLaMA-7B, NLL decreases on every benchmark, while RMSE decreases or remains unchanged. Backbone choice also affects distribution quality beyond point accuracy: on VL20, OPT-1.3B and LLaMA-7B share the reported RMSE of 0.252, but their NLLs differ (-12.831 vs. -13.336).

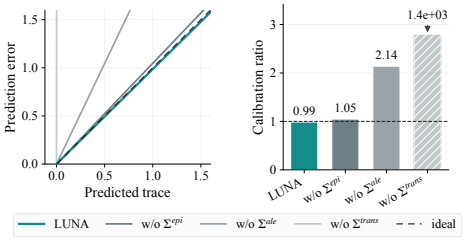


Figure 3: Effect of uncertainty states on state errors. Right: calibration ratio.

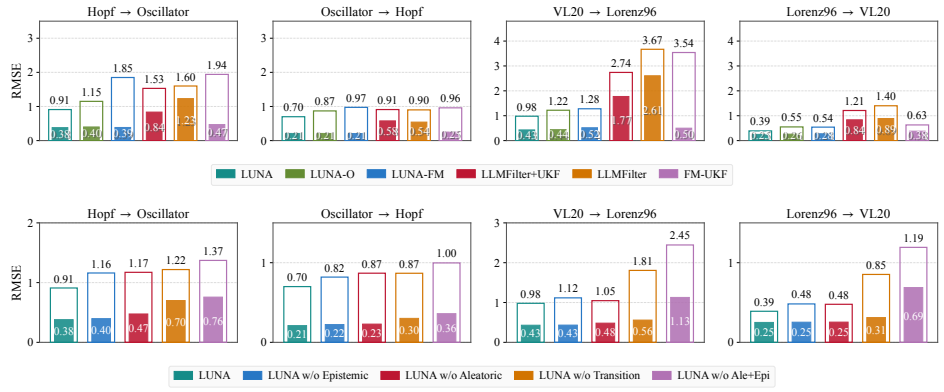


Figure 4: Performance Evaluation (Filled bars: in-domain; Outlined bars: cross-domain).

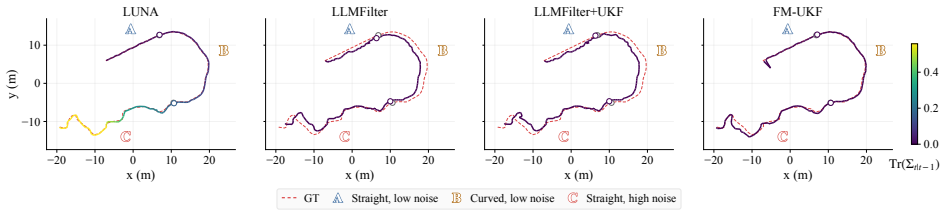


Figure 5: Tracking under changing dynamics and noise across three phases (A, B, and C).

Removing any covariance component worsens both metrics on every benchmark. Removing the propagated covariance raises Tracking RMSE from 0.442 to 2.044, consistent with covariance influencing the Kalman gain and subsequent state updates. Removing epistemic uncertainty barely changes VL20 RMSE (0.252 to 0.253) but worsens NLL (-13.336 to -8.257). NLL additionally evaluates the probability assigned to the true state; RMSE alone therefore understates this component's contribution to distribution quality.

4.3 Uncertainty-Error Correlation.

This experiment tests whether LUNA accurately estimates prediction-step uncertainty. On the Tracking system, Fig. 3 visualizes the relationship between the squared prediction error and the predicted covariance trace in the left panel, and reports their ratio in the right panel; the ratio is defined in Appendix D.1. If the predicted mean and covariance are correct, the expected squared prediction error equals the expected covariance trace, giving an ideal ratio of 1. The full LUNA follows the ideal one-to-one trend and achieves a ratio of 0.99. Removing the propagated, aleatoric, or epistemic covariance source changes the ratio to 1.4×10^3, 2.14, or 1.05, respectively, moving it away from the ideal value. These results support the use of all three covariance sources in prediction-uncertainty estimation.

4.4 RQ2: Generalization and Robustness

Cross-Domain Generalization. Following (Liu et al., 2025), we evaluate cross-domain generalization by training on source systems and testing on different target systems without parameter updates. Evaluation uses the known target observation model and a system context containing a five-state target-system sequence disjoint from the test data (see Appendix B.3). We include LUNA-FM, which replaces only the LLaMA-7B backbone with TimesFM-3.0 while retaining the surrounding modules and filtering procedure. We also include LUNA-O, which removes the system context from LUNA. From Figure 4, LUNA achieves the lowest cross-domain RMSE in all four settings, with an average relative reduction of 41.3% than the runner-up external baseline.

The benefits of the LLM backbone and system context are more pronounced under cross-domain transfer than in-domain evaluation. LUNA achieves lower cross-domain RMSE than both LUNA-FM and LUNA-O in all four settings. For example, on Hopf → Oscillator, the in-domain RMSEs

of LUNA, LUNA-FM, and LUNA-O are similar (0.38, 0.39, and 0.40), whereas their cross-domain RMSEs differ substantially (0.91, 1.85, and 1.15, respectively).

Uncertainty modeling contributes to cross-domain generalization. Removing any covariance component increases cross-domain RMSE in all four settings. On VL20 $\rightarrow$ Lorenz96, removing epistemic uncertainty leaves the reported in-domain RMSE unchanged at 0.43 but increases cross-domain RMSE from 0.98 to 1.12. This contribution is consistent with predicted covariance governing the balance between predictions and observations during the filtering update.

Tracking under Changing Dynamics and Noise. To examine how uncertainty quantification supports generalization, we visualize how predictive uncertainty responds to differences between the training and test systems. We train LUNA on simulated Tracking trajectories with straight motion and low process noise, then evaluate three successive phases: (A) the same conditions as training, (B) curved motion with low process noise, and (C) straight motion with substantially increased process noise, representing a bigger difference from the training system. Fig. 5 shows the estimated trajectories, colored by predictive uncertainty as measured by the covariance trace $\mathrm{tr}(\Sigma_{t|t-1})$. This uncertainty controls how much the update step trusts the prediction: with observation noise fixed, higher predictive uncertainty shifts the balance toward observation. Thus, when system changes make predicted state less reliable, predictive uncertainty should reflect this reduced reliability so that the update step can avoid overreliance on those predictions. LUNA exhibits this response at the transition from B to C: its predictive covariance trace rises markedly, while its updated estimates remain close to the ground-truth trajectory. In contrast, FM-UKF and LLMFilter+UKF show little change in predictive uncertainty and larger tracking errors. These observations suggest that the proposed uncertainty quantification framework helps the update avoid overreliance on unreliable predictions, providing a possible explanation for the generalization benefits observed in the cross-domain uncertainty ablations.

4.5 RQ3: ABLATION STUDY

Finally, we are interested in how the adopted NIW evidential regression model is comparable to other UQ methods. To this end, in Table 2, we choose two alternative UQ methods including iterative UQ (Durasov et al., 2024) and MC-Dropout over the NIW regression model. As shown in this table, the adopted NIW evidential regression model leads to much lower errors and much faster time than the two alternatives. Next, to study how the transition component in Eq. (3) effects the performance of LUNA, we develop the variant, w/o $\mathcal{D}(\cdot)$, that replaces the entire one-step state change network in Eq. (3) with the MLP to directly predict next-step states, and the variant, w/o $\delta(\cdot)$, that removes the correction in Eq. (3a). From this table, the removal of $\mathcal{D}(\cdot)$ leads to rather high errors, revealing that the simple prediction of the next-step state does not work well. Instead, the removal of the correction network $\delta(\cdot)$ in Eq. (3a) leads to trivially higher errors. Nevertheless, the two variants lead to slightly faster inference time.

Table 2: UQ and Transition Evaluation by the average on Hopf, Oscillator, Lorenz96 and VL20.

Method	RMSE↓	NLL↓	ms/step↓
LUNA w/ Iterative	0.365	25.164	165.33
LUNA w/ MCDropout	0.354	22.308	168.69
LUNA w/o change $\mathcal{D}(\cdot)$	0.345	9.658	<u>22.77</u>
LUNA w/o correction $\delta(\cdot)$	0.336	8.249	**22.61**
LUNA	**0.317**	**3.600**	25.23

5 CONCLUSION

To offer trustworthy state estimates with high generalization, we propose an LLM-based uncertainty-aware state estimation framework, LUNA. To obtain the UQ from both observed systems and LLMs, LUNA firstly deterministically selects a fixed number of Sigma-points that follow the state probability distribution. On these selected points, LUNA performs LLM-based state transition, reconstruction and uncertainty quantification without explicit LLM explanation. Experiments on six synthetic and real-world benchmarks demonstrate the effectiveness of LUNA in state estimation and uncertainty quantification: it achieves the best RMSE on five benchmarks and the best NLL on four, reducing average in-domain RMSE by 71.8% relative to LLMFilter. Under cross-system evaluation, LUNA achieves the lowest RMSE in all four transfer settings, with an average 41.3% reduction relative to the best external baseline; the corresponding ablations suggest that predictive uncertainty is linked to cross-system generalization. As future work, we plan to develop a learnable Sigma-point sampler and investigate zero-shot cross-system generalization across state dimensions.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing to improve grammar, clarity, and readability, and to assist in drafting these statements. We did not use these tools to develop the research methodology, formulate hypotheses or mathematical claims, write proofs, generate or process experimental data, implement methods, or interpret results. The use of pretrained LLMs as components of the evaluated models is described separately in the methodology and experimental sections. The authors reviewed all AI-assisted text and checked that it accurately reflects the methods, results, and intended claims. We take full responsibility for the final content of this work, including all AI-assisted text.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 3 describes the proposed framework and training objectives. Appendix B provides the full NIW objective, network architectures, System-as-Prompt context construction, UKF measurement update, and training protocol, while Algorithm 1 summarizes the training and online filtering procedures. Appendix C provides the theoretical derivations and proofs for covariance reconstruction and the analysis of covariance-source omission. Section 4.1 and Appendix D describe the evaluation metrics, benchmark configurations and dynamics, data construction and splits, baseline implementations, and computing environment. Appendix E reports additional experiments.

REFERENCES

Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P. Bruinsma, Tom R. Andersson, Michael Herzog, Nicholas D. Lane, Matthew Chantry, J. Scott Hosking, and Richard E. Turner. End-to-end data-driven weather prediction. *Nat.*, 641(8065):1172–1179, 2025. doi: 10.1038/S41586-025-08897-0. URL https://doi.org/10.1038/s41586-025-08897-0.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/aab085461de182608ee9f607f3f7d18f-Abstract.html.

Mohammad Amin Basiri and Sina Khanmohammadi. Discovering governing equations from graph-structured data by sparse identification of nonlinear dynamical systems. *CoRR*, abs/2409.04463, 2024. doi: 10.48550/ARXIV.2409.04463. URL https://doi.org/10.48550/arXiv.2409.04463.

Nicholas Carlevaris-Bianco, Arash K. Ushani, and Ryan M. Eustice. University of michigan north campus long-term vision and lidar dataset. *Int. J. Robotics Res.*, 35(9):1023–1035, 2016. doi: 10.1177/0278364915614638. URL https://doi.org/10.1177/0278364915614638.

Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003. URL https://automatica.dei.unipd.it/old/public/Schenato/TESI/Tosin_2009/materiale/Nonlinear_filtering_Chen.pdf.

Pragyan Dahal, Simone Mentasti, Luca Paparusso, Stefano Arrigoni, and Francesco Braghin. Robuststatenet: Robust ego vehicle state estimation for autonomous driving. *Robotics Auton. Syst.*, 172:104585, 2024. doi: 10.1016/J.ROBOT.2023.104585. URL https://doi.org/10.1016/j.robot.2023.104585.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Petar M. Djuric, Jayesh H. Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F. Bugallo, and Joaquín Míguez. Particle filtering. *IEEE Signal Process. Mag.*, 20(5):19–38, 2003. doi: 10.1109/MSP.2003.1236770. URL https://doi.org/10.1109/MSP.2003.1236770.

Nikita Durasov, Doruk Öner, Jonathan Donier, Hieu Le, and Pascal Fua. Enabling uncertainty estimation in iterative neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12172–12189. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/durasov24a.html.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nat.*, 630(8017):625–630, 2024. doi: 10.1038/S41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Quanbo Ge, Teng Shao, Zhansheng Duan, and Chenglin Wen. Performance analysis of the kalman filter with mismatched noise covariances. *IEEE Trans. Autom. Control.*, 61(12):4014–4019, 2016. doi: 10.1109/TAC.2016.2535158. URL https://doi.org/10.1109/TAC.2016.2535158.

Mohinder S Grewal. Kalman filtering. In *International encyclopedia of statistical science*, pp. 1285–1289. Springer, 2025. URL https://link.springer.com/content/pdf/10.1007/978-3-662-69359-9_310.pdf.

Brian D Hassard, Nicholas D Kazarinoff, and Yieh-Hei Wan. *Theory and applications of Hopf bifurcation*, volume 41. CUP Archive, 1981. URL https://www.cambridge.org/hk/universitypress/subjects/mathematics/differential-and-integral-equations-dynamical-systems-and-co/theory-and-applications-hopf-bifurcation.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *ACM Comput. Surv.*, 58(7):179:1–179:35, 2026. doi: 10.1145/3786319. URL https://doi.org/10.1145/3786319.

Tobin Holtmann, David Stenger, Andres Felipe Posada-Moreno, Friedrich Solowjow, and Sebastian Trimpe. Sailing towards zero-shot state estimation using foundation models combined with a UKF. In *64th IEEE Conference on Decision and Control, CDC 2025, Rio de Janeiro, Brazil, December 9-12, 2025*, pp. 96–102. IEEE, 2025. doi: 10.1109/CDC57313.2025.11313025. URL https://doi.org/10.1109/CDC57313.2025.11313025.

Ming Lei. Extended kalman filter. In *Optimal Estimation and Information Fusion: Theory and Algorithms*, pp. 181–206. Springer, 2025. URL https://link.springer.com/chapter/10.1007/978-981-96-3173-5_7.

Wenling Li, Shihao Sun, Yingmin Jia, and Junping Du. Robust unscented kalman filter with adaptation of process and measurement noise covariances. *Digit. Signal Process.*, 48:93–103, 2016. doi: 10.1016/J.DSP.2015.09.004. URL https://doi.org/10.1016/j.dsp.2015.09.004.

Shiqi Liu, Wenhan Cao, Chang Liu, Zeyu He, Tianyi Zhang, Yinuo Wang, and Shengbo Eben Li. One filters all: A generalist filter for state estimation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/1ea7c8f5120a620e61425c530644a4c2-Abstract-Conference.html.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pp. 1–18. Reading, 1996. URL https://www.cambridge.org/core/services/aop-cambridge-core/content/view/3221BDE379DEB669BA52C66263AF3206.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.557. URL https://doi.org/10.18653/v1/2023.emnlp-main.557.

Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 9134–9142. AAAI Press, 2023. doi: 10.1609/AAAI.V37I8.26096. URL https://doi.org/10.1609/aaai.v37i8.26096.

Rudolph van der Merwe and Eric A Wan. Sigma-point kalman filters for integrated navigation. In *Proceedings of the 60th annual meeting of the institute of navigation (2004)*, pp. 641–654, 2004. URL https://www.ion.org/publications/abstract.cfm?articleID=5662.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022. ISSN 1941-0476. doi: 10.1109/TSP.2022.3158588. URL https://ieeexplore.ieee.org/abstract/document/9733186.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023. URL https://books.google.com/books?hl=zh-CN&lr=&id=WLe9EAAAQBAJ&oi=fnd&pg=PR11&dq=Bayesian+filtering+and+smoothing&ots=qey-O7wVPY&sig=IDFeaE_XRkgogaCdOznvG1ZWF9M.

William Talbot, Julian Nubert, Turcan Tuna, Cesar Cadena, Frederike Dümbgen, Jesus Tordesillas, Timothy D. Barfoot, and Marco Hutter. Continuous-time state estimation methods in robotics: A survey. *IEEE Trans. Robotics*, 41:4975–4999, 2025. doi: 10.1109/TRO.2025.3593079. URL https://doi.org/10.1109/TRO.2025.3593079.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Gabriele Vissio and Valerio Lucarini. Mechanics and thermodynamics of a new minimal model of the atmosphere. *The European Physical Journal Plus*, 135(10):1–21, 2020. URL https://link.springer.com/article/10.1140/epjp/s13360-020-00814-w.

Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pp. 153–158. Ieee, 2000. URL https://ieeexplore.ieee.org/abstract/document/882463/.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Lingyi Zhang, David Sidoti, Adam Bienkowski, Krishna R. Pattipati, Yaakov Bar-Shalom, and David L. Kleinman. On the identification of noise covariances and adaptive kalman filtering: A new look at a 50 year-old problem. *IEEE Access*, 8:59362–59388, 2020. doi: 10.1109/ACCESS.2020.2982407. URL https://doi.org/10.1109/ACCESS.2020.2982407.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/ARXIV.2205.01068. URL https://doi.org/10.48550/arXiv.2205.01068.

Ruiqi Zhong. *Natural Language Explanations of Dataset Patterns*. University of California, Berkeley, 2025. URL https://search.proquest.com/openview/ddd7f4065c96cc481360061403846f52/1?pq-origsite=gscholar&cbl=18750&diss=y.

Tianlong Zhou, Tianyi Shi, Hongye Gao, and Weixiong Rao. Learning to Optimize State Estimation in Multi-Agent Reinforcement Learning-Based Collaborative Detection. *IEEE Transactions on Mobile Computing*, 23(12):14330–14343, December 2024. ISSN 1558-0660. doi: 10.1109/TMC.2024.3445583. URL https://ieeexplore.ieee.org/document/10638830/.