LEARNING META-DYNAMICS FOR OFF-DYNAMICS KALMAN FILTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recursive state estimation becomes challenging when the transition model itself evolves during filtering: each recursion provides only one new noisy observation, yet the transition must be revised before state and covariance propagation. We formulate this setting as *off-dynamics Kalman filtering* and view the causal evolution of the transition model as *meta-dynamics*. We introduce the Dynamics-Adaptive Kalman Network (DyAKNet), which represents dynamics adaptation explicitly in a sparse coefficient space. DyAKNet converts the diagnostic innovation under the preceding model into coefficient-space evidence, aggregates evidence across recursions, and separately updates signed coefficient values and active support before prediction. The revised transition is then immediately reused within the same filtering recursion. Under local smoothness, boundedness, and contraction conditions, we derive a finite-horizon mean-square error bound that characterizes the contributions of vector-field mismatch and stochastic noise. Experiments span coefficient evolution, structural changes within a prescribed candidate representation, unseen noise and switching patterns, and real-world localization. Across these settings, DyAKNet consistently improves state-estimation accuracy over the evaluated adaptive and learning-based baselines, while retaining an explicit and interpretable dynamics representation. These results support same-recursion dynamics adaptation as an effective strategy for filtering under evolving model mismatch. Code is available at `https://anonymous.4open.science/r/DyAKNet-191A`.

1 INTRODUCTION

Recursive filters rely on a transition model at every prediction step. When physical parameters drift, interaction structures switch, or previously inactive admissible effects become active, a transition that was accurate earlier can become systematically mismatched even while measurements continue to arrive O'Connell et al. (2022); Kumar et al. (2022); Lv et al. (2021). We study this regime under known observation and noise models but unknown, evolving sparse transition dynamics, and refer to it as *off-dynamics Kalman filtering*. The difficulty is distinct from ordinary process noise: the model used for state and covariance propagation is itself changing during operation.

Existing dynamics-learning approaches operate under a different information pattern. Data-driven predictors and sparse equation discovery learn from trajectories containing many transitions Pathak et al. (2018); Lu et al. (2018); Patel et al. (2021); Patel & Ott (2023); Brunton et al. (2016), whereas end-to-end neural filters absorb the transition into an implicit learned representation Jonschkowski et al. (2018); Zhang et al. (2024). Online variants such as OL-DPF adapt at deployment Li et al. (2023), but the changing dynamics remain implicit. Adaptive and dual filters track parameters within prescribed structures, while IMM filters select among predefined modes Mehra (1972); Kitagawa (1998); Kantas et al. (2015); Blom & Bar-Shalom (1988). What remains difficult is the combination that matters here: one new noisy observation per recursion, possible changes in both coefficient values and active terms, and the need to revise the transition *before* the current state prediction.

This creates an underdetermined inference problem. The current innovation can indicate that the preceding model is inconsistent with the observation, but it mixes model mismatch with state uncertainty and stochastic noise and is expressed in measurement space rather than in the space of candidate dynamics. One observation therefore provides only a partial constraint on the possible coefficient changes. The key question is not merely whether to adapt online, but *how to turn sparse, noisy filtering evidence into a structured transition update that can be used immediately*.

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.

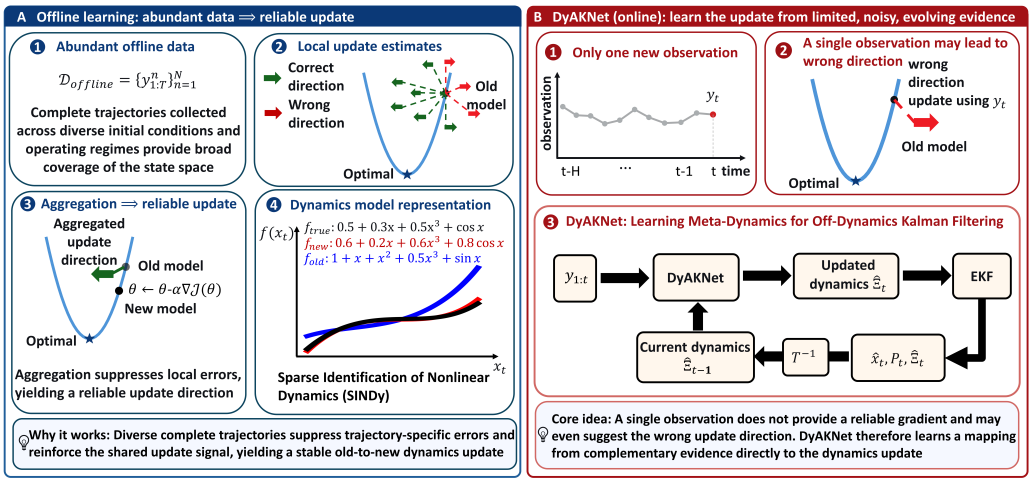

Figure 1: Offline dynamics learning versus same-recursion meta-dynamics learning. Offline trajectories provide many transitions before model fitting; off-dynamics filtering receives one new noisy observation per recursion. DyAKNet turns this local evidence into an explicit sparse dynamics update before the current Kalman prediction and correction.

We address this question by viewing the causal evolution of the transition model itself as *meta-dynamics* and introducing the Dynamics-Adaptive Kalman Network (DyAKNet). Fig. 1 highlights the distinction from offline dynamics learning. DyAKNet first diagnoses the preceding transition without correcting the state and then maps the resulting innovation sensitivity into the explicit coefficient space of a prescribed sparse dynamics library. Because this instantaneous evidence is incomplete, recurrent memory aggregates evidence across recursions and operating points. A structured updater separately revises signed coefficient values and soft support, allowing the same mechanism to represent coefficient drift and activation or suppression of admissible terms. The revised transition is then used immediately for state and covariance propagation before the single Kalman correction of the current recursion.

This design makes the dynamics update an explicit intermediate representation rather than an implicit side effect of a learned filter. It directly connects the evidence available to the filter with the structure being updated: the local score attributes measurement discrepancy to candidate dynamics terms, temporal memory aggregates otherwise incomplete one-step evidence, and value/support factorization preserves the distinction between coefficient changes and support changes. Our local analysis further characterizes how mismatch in the resulting vector field enters the filtering recursion under finite-horizon regularity and contraction conditions.

Our contributions are threefold. **First**, we formulate off-dynamics filtering as observation-conditioned sparse meta-dynamics learning, where the transition is revised causally and reused within the same recursion. **Second**, we introduce a history-conditioned coefficient-space update that connects innovation sensitivity to candidate dynamics terms and separates value adaptation from support adaptation within a prescribed representation. **Third**, we provide a finite-horizon filtering-error analysis and evaluate the method through coefficient evolution, structural evolution, unseen noise and switching patterns, and NCLT localization. DyAKNet achieves the lowest mean MSE in all nine coefficient-evolution settings, in both target-tracking tests, across the reported unseen-noise and unseen-switching tests among non-oracle methods, and in the reported NCLT comparison.

2 Related Work

Data-Driven Dynamics Learning and Sparse Equation Discovery. Data-driven dynamics models learn state-transition rules from informative trajectories. Reservoir computing and recurrent models have been used to reproduce chaotic attractors, non-stationary dynamics, and regime transitions from sequential state data Pathak et al. (2018); Lu et al. (2018); Patel et al. (2021); Patel & Ott (2023). Learned state-space models and differentiable particle filters similarly represent system transitions

through neural parameterizations Jonschkowski et al. (2018); Chen et al. (2021). These methods are typically trained from trajectories containing many state transitions and are primarily designed for forecasting, dynamics modeling, or end-to-end filtering.

Sparse Identification of Nonlinear Dynamics (SINDy) instead represents a vector field as a sparse combination of candidate functions, providing an explicit and interpretable dynamics representation Brunton et al. (2016). Subsequent work has extended this formulation to controlled, time-varying, model-selection, and online settings Champion et al. (2019); Mangan et al. (2019); Shen et al. (2022). However, these approaches generally rely on informative state trajectories or derivative estimates and treat dynamics identification separately from recursive state estimation. In contrast, our setting requires the transition model to be revised from the newly arriving observation and recursive filtering history before the state prediction at the same time step.

Filtering under Uncertain and Evolving Dynamics. Adaptive and dual filters estimate unknown parameters or noise statistics within a prescribed transition structure Mehra (1972); Kitagawa (1998); Kantas et al. (2015), while IMM filters maintain predefined motion models and infer their mode probabilities online Blom & Bar-Shalom (1988); Fan et al. (2021). These methods remain constrained by the assumed parameterization or candidate model set.

Data-driven filters relax some of these assumptions. Differentiable particle filters learn filtering components jointly Jonschkowski et al. (2018), Transformer-based trackers learn motion representations from data Zhang et al. (2024), and OL-DPF further adapts a particle filter during deployment Li et al. (2023). Hybrid data-assimilation methods learn forecast corrections or evolving dynamics online Wikner et al. (2021); Bocquet et al. (2021), while EKF-SINDy augments the filter state with selected sparse dynamics coefficients Rosafalco et al. (2025).

DyAKNet differs in both representation and update timing: it maintains a sparse dynamics representation with adaptable signed coefficients and active support, conditions the update on the current innovation and accumulated evidence, and immediately reuses the revised transition for state and covariance propagation within the same filtering recursion.

3 PRELIMINARIES

We consider a nonlinear stochastic system with sparse, time-varying dynamics,

$$x_t = x_{t-1} + \Xi_t^\top \Theta(x_{t-1})\Delta t + w_t, \qquad y_t = h(x_t) + v_t, \tag{1}$$

where $x_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^n$, $w_t \sim \mathcal{N}(0, Q)$ and $v_t \sim \mathcal{N}(0, R)$, and $\Theta(x) = [\phi_1(x), \ldots, \phi_p(x)]^\top$ is a prescribed candidate library as in SINDy Brunton et al. (2016); Champion et al. (2019). The sparse coefficient matrix $\Xi_t \in \mathbb{R}^{p\times m}$ determines both the active terms and their signed values. For compactness, define $a_\Xi(x) = \Xi^\top \Theta(x)$ and $f_\Xi(x) = x + a_\Xi(x)\Delta t$.

We use an EKF as the recursive state estimator. Given a transition f, its Jacobian F_t, and observation Jacobian H_t, the filter forms the innovation $\delta y_t = y_t - h(\hat{x}_t^-)$ with covariance $S_t = H_t P_t^- H_t^\top + R$, then applies the standard Kalman correction. Only these innovation quantities are needed to describe DyAKNet; the full EKF recursion is provided in Appendix B.

4 METHOD

We consider recursive state estimation when the transition dynamics evolve during deployment. Unlike offline system identification, only one new noisy observation becomes available at each filtering step, while the transition used for state and covariance propagation must be determined immediately. We represent the evolving dynamics with a sparse coefficient matrix and introduce the **Dynamics-Adaptive Kalman Network (DyAKNet)**, which updates this representation before the state prediction at each recursion.

DyAKNet consists of three components: **Recurrent Residual Representation**, which summarizes persistent filtering discrepancies and recent dynamics changes; **Geometric Residual Attribution**, which maps the current residual from measurement space to the sparse coefficient space; and **Structured Coefficient Estimation**, which decodes the fused representation into updated coefficient

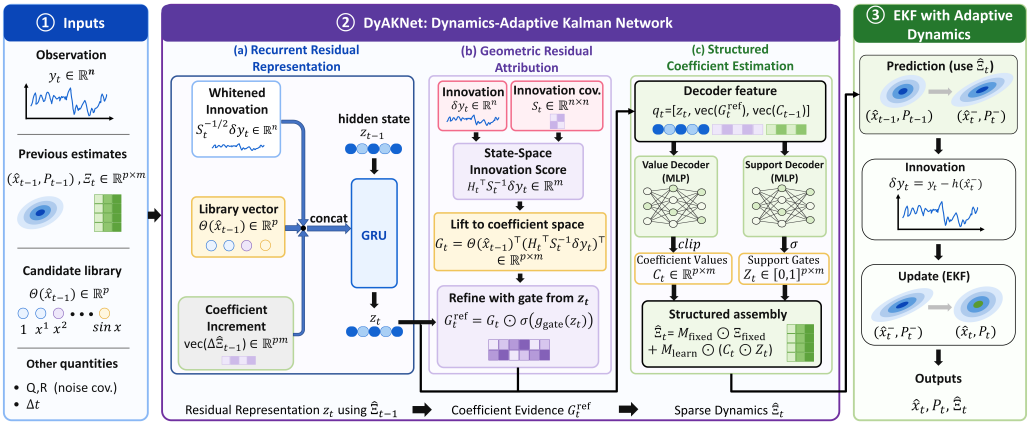

Figure 2: Overview of DyAKNet. **Recurrent Residual Representation** aggregates current filtering discrepancy with historical adaptation context. **Geometric Residual Attribution** maps the residual from measurement space to candidate dynamics coefficients. **Structured Coefficient Estimation** decodes the fused representation into updated coefficient values and support variables, which define the transition used for the current Kalman prediction and correction.

values and support variables. The resulting transition is then used directly for the current Kalman prediction and correction.

4.1 Problem Formulation

Building on Eq. (1), we assume a prescribed candidate-function library $\Theta(x) \in \mathbb{R}^p$ and a sparse, time-varying coefficient matrix $\Xi_t \in \mathbb{R}^{p\times m}$. Both coefficient values and active support may change over time, but changes are restricted to the admissible candidate representation. The observation model h, noise statistics, and candidate library are assumed available, while the current state and dynamics coefficients are unknown.

At recursion t, the new observation y_t becomes available together with the recursive estimates $(\hat{x}_{t-1}, P_{t-1}, \hat{\Xi}_{t-1})$. DyAKNet learns a causal dynamics-update operator

$$\hat{\Xi}_t = \mathcal{U}_\theta\big(y_t, \hat{x}_{t-1}, P_{t-1}, \hat{\Xi}_{t-1}\big), \tag{2}$$

and immediately uses the revised transition $f_{\hat{\Xi}_t}$ for state and covariance propagation at the same filtering step. The training objective minimizes state-estimation error over the filtering horizon while regularizing the estimated dynamics within the prescribed sparse representation.

4.2 Dynamics-Adaptive Kalman Network

DyAKNet updates the transition model before state propagation at each filtering recursion. As illustrated in Fig. 2, the current diagnostic residual is combined with recursive history, attributed to candidate dynamics terms in coefficient space, and decoded into an updated sparse transition model. The architecture consists of three modules: **Recurrent Residual Representation**, **Geometric Residual Attribution**, and **Structured Coefficient Estimation**.

Recurrent Residual Representation. DyAKNet first evaluates the preceding dynamics estimate $\hat{\Xi}_{t-1}$ through a diagnostic prediction without performing a state correction. This produces the diagnostic innovation $\delta y_{t,\mathrm{diag}}$ and its covariance $S_{t,\mathrm{diag}}$, which measure the discrepancy between the current observation and the prediction generated by the preceding transition.

Because the instantaneous residual contains both dynamics mismatch and stochastic uncertainty, DyAKNet combines it with recursive context. The recurrent input includes the whitened diagnostic innovation, the current library excitation, and the coefficient change applied in the preceding recursion. A GRU summarizes these signals as

$$z_t = \mathrm{GRU}_\theta(u_t, z_{t-1}), \tag{3}$$

where z_t summarizes residual history together with the current operating point and recent dynamics adaptation. This temporal representation addresses the fact that a single observation provides only a partial constraint on the higher-dimensional coefficient space.

Geometric Residual Attribution. The diagnostic innovation is expressed in measurement space, while the transition model is parameterized by sparse dynamics coefficients. We therefore construct a local attribution that maps the observed discrepancy directly into coefficient space:

$$G_t = \Theta(\hat{x}_{t-1})\big(H_{t,\mathrm{diag}}^{\top} S_{t,\mathrm{diag}}^{-1} \delta y_{t,\mathrm{diag}}\big)^{\top}. \tag{4}$$

The resulting matrix $G_t \in \mathbb{R}^{p\times m}$ has the same structure as the coefficient matrix and associates each candidate library term with a state-equation component. It provides structured local evidence in coefficient space rather than directly updating the model from a measurement-space residual.

The recurrent representation further modulates this instantaneous attribution as

$$G_t^{\mathrm{ref}} = G_t \odot \sigma(g_{\mathrm{gate}}(z_t)), \tag{5}$$

so that the current coefficient-space evidence is conditioned on the residual history accumulated across previous recursions.

Structured Coefficient Estimation. The final module converts the recurrent representation and geometric attribution into the updated sparse dynamics. DyAKNet separates coefficient estimation into signed values C_t and soft support variables Z_t, allowing coefficient magnitudes and active terms to evolve within the same representation.

The decoder receives the recurrent state, refined attribution, and preceding coefficient values,

$$q_t = \big[z_t, \mathrm{vec}(G_t^{\mathrm{ref}}), \mathrm{vec}(C_{t-1})\big], \tag{6}$$

and predicts the updated value and support variables:

$$C_t = \mathrm{clip}(C_{t-1} + \eta \tanh(g_{\mathrm{val}}(q_t)), -c_{\max}, c_{\max}), \qquad Z_t = \sigma(g_{\mathrm{sup}}(q_t)). \tag{7}$$

The sparse transition is then constructed as

$$\hat{\Xi}_t = M_{\mathrm{fixed}} \odot \Xi_{\mathrm{fixed}} + M_{\mathrm{learn}} \odot (Z_t \odot C_t). \tag{8}$$

This decoder represents both coefficient variation and support changes through a single dynamics-update mechanism while preserving known coefficients and the prescribed admissibility constraints.

4.3 State Inference with Updated Dynamics

After estimating $\hat{\Xi}_t$, DyAKNet immediately replaces the preceding transition with the revised dynamics for the current filtering step. The prediction is

$$\hat{x}_t^{-} = \hat{x}_{t-1} + \hat{\Xi}_t^{\top} \Theta(\hat{x}_{t-1}) \Delta t, \tag{9}$$

and the corresponding transition Jacobian is used for covariance propagation. A single standard Kalman measurement correction is then performed using the recomputed prediction and innovation.

The ordering is central to the formulation: the current observation first provides evidence for updating the transition, and the revised transition is then used for both state and covariance prediction before the standard measurement correction. The posterior state determines the operating point for the next dynamics update, closing the recursion between dynamics adaptation and state estimation.

4.4 Local Filtering Analysis

The role of the learned dynamics update can be characterized through the local vector-field mismatch

$$\varepsilon_{a,t} = \left\| (\Xi_t - \hat{\Xi}_t)^{\top} \Theta(\hat{x}_{t-1}) \right\|_2. \tag{10}$$

Proposition 4.1 (Local finite-horizon filtering error)**.** *Under local smoothness, bounded covariance and gain, and contraction of the linearized closed-loop filtering dynamics, the posterior error satisfies*

$$\mathbb{E}\|x_t - \hat{x}_t\|_2^2 \le \rho\, \mathbb{E}\|x_{t-1} - \hat{x}_{t-1}\|_2^2 + c_a \Delta t^2 \mathbb{E}\varepsilon_{a,t}^2 + c_q\, \mathrm{tr}(Q) + c_r\, \mathrm{tr}(R), \tag{11}$$

for $\rho \in (0,1)$ and positive constants c_a, c_q, c_r.

The bound makes the role of dynamics mismatch explicit: error in the learned vector field enters the filtering recursion as a perturbation term alongside process and observation noise. The result is local and finite-horizon; the complete assumptions and proof are provided in Appendix D.

4.5 Training Objective

DyAKNet is trained over complete trajectories. We use an innovation-consistency term together with sparse-dynamics regularization and the supervision available in each dataset:

$$\mathcal{L} = \lambda_{\mathrm{NLL}}\mathcal{L}_{\mathrm{NLL}} + \mathcal{L}_{\mathrm{dyn}} + \lambda_{\mathrm{sup}}\mathcal{L}_{\mathrm{sup}}, \tag{12}$$

where $\mathcal{L}_{\mathrm{dyn}}$ groups the sparsity and temporal-smoothness regularizers on the estimated coefficient sequence. The supervision term uses coefficient reconstruction when ground-truth dynamics are available and state-estimation error otherwise. Objective terms are provided in Appendix C.

The innovation term encourages consistency between the updated transition and the observations, while $\mathcal{L}_{\mathrm{dyn}}$ regularizes sparsity and temporal variation in the coefficient representation. The supervision term allows the same recursive inference architecture to use coefficient supervision in synthetic experiments and state supervision in the real-world experiment.

5 Experiments

The experiments evaluate the same dynamics-update mechanism under different sources of model mismatch. We first test *parametric evolution*, where coefficients vary while the structure is retained. We then test *structural evolution*, including motion-regime switches and activation of an admissible term that is absent from the nominal motion model. Third, we evaluate *distribution shift* using observation-noise levels and switching patterns not seen during training. Finally, NCLT tests the method on localization data when coefficient supervision is unavailable. All reported accuracy results use trajectory-averaged state MSE in dB; protocols are provided in Appendices E and F.

Benchmarks and baselines. Synthetic tests use Selkov, Hopf, Lorenz, and maneuvering-target systems SEL'KOV (1968); Hassard et al. (1981); Lorenz (1963); Rong Li & Jilkov (2003); real-world evaluation uses NCLT Carlevaris-Bianco et al. (2015). We compare with IMM, EKF-SINDy, FoT, OL-DPF, and RC–EKF Blom & Bar-Shalom (1988); Rosafalco et al. (2025); Li et al. (2025; 2023); Patel & Ott (2023); an oracle EKF is shown only as a privileged reference. Implementation and modeling details are deferred to the appendix.

5.1 Coefficient Evolution in Nonlinear Systems

This experiment evaluates state estimation when the dynamics coefficients evolve over time while the prescribed candidate structure remains available.

Experimental Setup We evaluate DyAKNet on three canonical nonlinear systems used in state estimation: the Selkov, Hopf, and Lorenz models. In each system, the dominant dynamics coefficients evolve over time, reflecting changes in the system's motion. DyAKNet is trained at the nominal level $1/\mathrm{q}^2 = 1/\mathrm{r}^2 = 0$ dB. We test it at unseen levels $1/\mathrm{r}^2 \in \{-10, 10, 30\}$ dB, with $Q = \mathrm{q}^2 I$ and $R = \mathrm{r}^2 I$. Detailed descriptions of the evolving dynamics and baselines are in Appendix E.1.

Results and Analysis Table 1 summarizes state-estimation performance across the three nonlinear systems and three unseen measurement-noise levels. DyAKNet achieves the lowest reported mean MSE in all nine settings, with RC–EKF ranking second throughout. For example, on Lorenz at $1/\mathrm{r}^2 = 30$ dB, DyAKNet attains -27.25 dB, compared with -26.65 dB for RC–EKF, -26.36 dB for EKF-SINDy, and -25.62 dB for OL-DPF. Across these controlled tests, the same observation-conditioned coefficient update remains effective as the underlying coefficients evolve and the measurement-noise level differs from training.

5.2 Adaptation to Target Tracking

This experiment isolates two forms of structural evolution in target tracking: switches among prescribed motion regimes and activation of an admissible candidate term that is absent from

Table 1: State-estimation performance (MSE [dB]) on nonlinear systems under different measurement-noise levels, parameterized by $1/r^2$ [dB].

Method	Selkov			Hopf			Lorenz		
	-10	10	30	-10	10	30	-10	10	30
Oracle Filter	7.86±0.42	-12.11±0.48	-32.12±0.46	7.56±0.47	-12.13±0.46	-32.36±0.44	7.77±0.37	-12.05±0.38	-32.20±0.40
IMM	11.94±3.45	-7.26±3.81	-25.06±3.89	11.56±3.84	-7.43±3.56	-25.78±3.96	12.90±3.40	-7.01±3.42	-23.36±3.97
FoT	11.33±3.86	-8.33±3.87	-28.28±4.07	10.97±3.84	-8.44±3.90	-28.63±3.85	11.09±3.45	-8.24±3.43	-26.18±3.78
EKF-SINDy	9.61±1.33	-11.15±1.37	-28.07±1.03	8.79±1.24	-11.62±1.26	-28.65±1.03	9.06±1.39	-10.87±1.40	-26.36±1.03
OL-DPF	10.93±0.64	-9.17±0.66	-27.42±1.25	10.14±0.59	-8.98±0.69	-27.96±1.26	9.78±0.57	-9.35±0.67	-25.62±1.22
RC–EKF	9.52±0.53	-11.26±0.64	-28.49±1.48	8.61±0.57	-11.96±0.71	-28.94±1.22	8.89±0.54	-11.58±0.67	-26.65±1.30
DyAKNet	**9.03±0.38**	**-11.98±0.39**	**-29.04±1.11**	**8.14±0.38**	**-12.45±0.39**	**-29.47±1.11**	**8.61±0.38**	**-11.64±0.39**	**-27.25±1.11**

Table 2: Target-tracking performance (MSE [dB]) under motion-model switches and activation of an admissible non-nominal dynamics term.

Method	IMM	FoT	EKF-SINDy	OL-DPF	RC–EKF	DyAKNet
Structural Changes	-9.61±1.08	-0.24±2.98	-6.37±1.14	-1.96±2.06	-8.45±1.12	**-9.76±1.08**
New Term	-18.46±3.50	-21.40±0.64	-26.46±1.04	-20.38±2.18	-27.63±1.01	**-28.41±0.96**

the nominal motion models. The latter tests support adaptation within the prescribed candidate representation rather than discovery of an arbitrary out-of-library term. Distribution shifts in noise level and switching pattern are evaluated separately in Section 5.3. Detailed system definitions and switching protocols are provided in Appendix E.2.

Structural Changes Structural changes here correspond to switches among constant velocity (CV), constant acceleration (CA), and coordinated turn (CT) models over time Rong Li & Jilkov (2003). Table 2 summarizes the resulting tracking performance. DyAKNet obtains the lowest reported mean MSE at -9.76 dB, closely matching IMM at -9.61 dB within the reported variability. RC–EKF reaches -8.45 dB, whereas OL-DPF yields -1.96 dB. These results show that DyAKNet remains effective under temporal model switches without receiving the switching sequence in advance.

Emergence of New Dynamic Terms We next activate a candidate dynamics term that is absent from the nominal motion model, using the linear damping term in the CA phase described in Appendix E.2.2. This setting tests structural mismatch within the prescribed candidate representation: the term is inactive in the nominal model but admissible to the DyAKNet update. As shown in Table 2, DyAKNet achieves the lowest reported mean MSE at -28.41 dB. RC–EKF, EKF-SINDy, and OL-DPF obtain -27.63, -26.46, and -20.38 dB, respectively. The result provides state-estimation evidence for adapting to activation of an admissible non-nominal term; it is not used as a quantitative claim of exact coefficient recovery.

5.3 Generalization under Distribution Shift

Table 3: Generalization comparison in MSE [dB]. The left block varies unseen noise intensity ($1/r^2$ [dB]); the right block varies unseen dynamics switching.

Method	Unseen Noise Intensity				Unseen Dynamics Switching		
	-10	10	20	30	Sequence	Timing	Time+Seq
Oracle Filter	-0.40±1.25	-19.31±1.11	-28.92±1.12	-39.24±1.16	-29.29±1.14	-29.22±1.15	-29.33±1.27
IMM	-0.24±1.05	-19.00±1.12	-28.50±1.87	-21.21±4.39	-29.10±1.14	-28.06±3.69	-29.21±1.25
FoT	6.72±0.92	-13.14±0.88	-22.97±1.02	-32.97±0.94	-22.43±0.98	-21.84±1.01	-22.75±0.89
EKF-SINDy	1.38±1.21	-15.47±1.19	-19.46±1.22	-21.04±1.21	-20.13±1.09	-19.47±1.21	-20.17±1.08
OL-DPF	2.56±1.18	-14.63±1.16	-21.61±1.32	-23.95±1.22	-16.08±1.12	-15.43±1.26	-15.56±1.31
RC–EKF	1.07±1.15	-17.25±1.16	-25.04±1.19	-29.62±1.19	-22.26±1.12	-21.43±1.19	-22.48±1.10
DyAKNet	**-0.28±1.05**	**-19.06±1.15**	**-28.81±1.10**	**-39.03±1.13**	**-29.28±1.13**	**-28.46±1.07**	**-29.27±1.24**

We next test whether the same learned update remains effective when either the observation statistics or the temporal dynamics pattern differs from training. Table 3 reports both shifts. Across all four unseen noise levels, DyAKNet has the lowest reported mean MSE among non-oracle methods. At $1/\mathrm{r}^2 = 30$ dB, it reaches -39.03 dB compared with the oracle value of -39.24 dB, while IMM, RC–EKF, and OL-DPF obtain -21.21, -29.62, and -23.95 dB, respectively. This result shows that the observation-conditioned dynamics update remains effective across the tested unseen noise levels; the comparison does not isolate which individual architectural component accounts for the differences among methods.

Under unseen switching sequences and timings, DyAKNet also has the lowest reported mean MSE among non-oracle methods in all three tests. In the combined timing-and-sequence test, it obtains -29.27 dB, compared with -29.21 dB for IMM, -22.48 dB for RC–EKF, -20.17 dB for EKF-SINDy, and -15.56 dB for OL-DPF. The small mean difference from IMM should not be interpreted as a statistically significant gain. Taken together, the two sets of tests show that the same update mechanism remains effective when either measurement conditions or switching patterns change beyond those used for training.

5.4 Evaluation on the Real-World NCLT Dataset

This experiment evaluates DyAKNet for localization state estimation on the real-world NCLT dataset. The true system dynamics and coefficient labels are unavailable, so training uses state supervision rather than explicit dynamics supervision.

Experimental Setup We evaluate DyAKNet on three nonlinear systems used in state estimation: the Selkov, Hopf, and Lorenz models. In each system, the dominant dynamics coefficients evolve over time, reflecting changes in the system's motion. DyAKNet is trained at the nominal level $1/\mathrm{q}^2 = 1/\mathrm{r}^2 = 0$ dB. We test it at unseen levels $1/\mathrm{r}^2 \in \{-10, 10, 30\}$ dB, with $Q = \mathrm{q}^2 I$ and $R = \mathrm{r}^2 I$. Descriptions of the evolving dynamics and baselines are in Appendix E.1.

Results Results in Table 4 show that DyAKNet achieves the lowest reported MSE at 9.70 dB. OL-DPF is second at 10.69 dB, while RC–EKF obtains 15.43 dB. Thus, the same recursive dynamics-update mechanism used in the synthetic experiments remains effective on this real-world localization benchmark when training relies on state rather than coefficient supervision. Trajectory comparisons in Fig. 3 are consistent with the numerical results. EKF, EKF-SINDy, IMM, and FoT accumulate larger visible deviations; OL-DPF and RC–EKF reduce these deviations; and DyAKNet produces the trajectory closest to the ground truth among the evaluated methods while retaining an explicit coefficient representation.

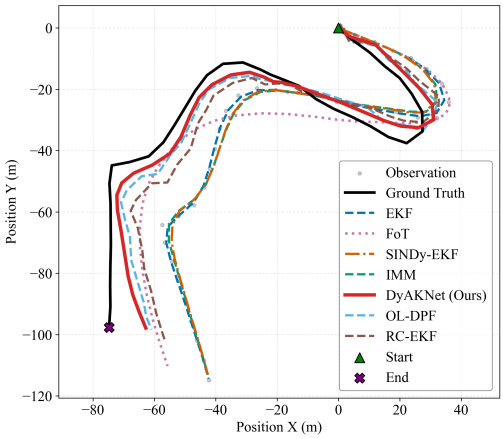

Figure 3: Estimated trajectory on NCLT.

Figure 4 shows the estimated sparse coefficients for $\dot{p}_x$. The coefficient magnitude decreases as robot motion shifts from the x-direction to the y-direction. The coefficient on v_x remains close to one with small fluctuations, consistent with the prescribed kinematic relation. A position-related term p_x is transiently active at the beginning and vanishes as more measurements arrive. Because the true dynamics coefficients are unavailable on NCLT, these traces provide a qualitative consistency check rather than coefficient-recovery evidence.

Table 4: State-estimation error and runtime on the NCLT experiment.

Metric	EKF	IMM	FoT	SINDy-EKF	OL-DPF	RC-EKF	DyAKNet
MSE [dB]	21.90	21.95	22.41	21.87	10.69	15.43	**9.70**
Runtime [s]	**0.30**	0.68	0.32	0.34	3.09	0.42	0.75

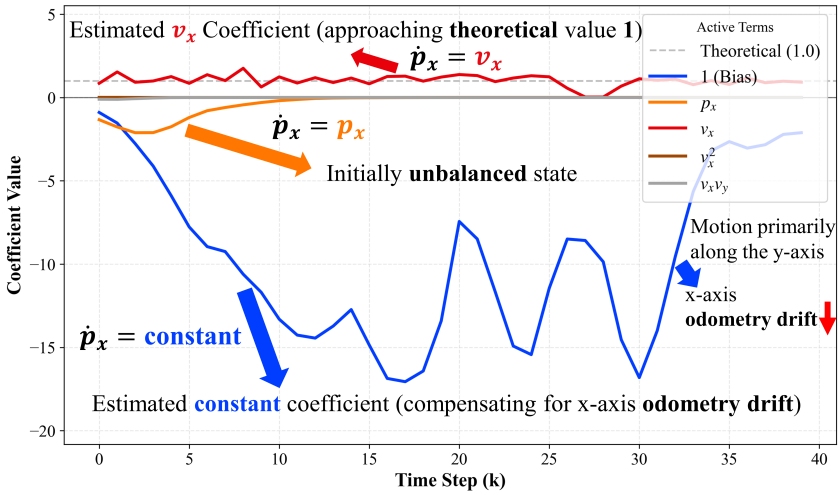

Figure 4: Estimated sparse dynamics coefficients for $\dot{p}_x$ on NCLT, adaptation to odometry drift.

5.5 Ablation Studies and Runtime

We evaluate the coefficient-space diagnostic score through three variants: removing the attribution term (w/o Attr) and replacing the score with the reported LSTM Hochreiter & Schmidhuber (1997) or Transformer Vaswani et al. (2017) alternatives. Table 5 shows that the full DyAKNet design has the lowest reported MSE on Lorenz, target tracking, and NCLT. The strongest alternative reaches -8.16, -15.09, and 13.71 dB, compared with -11.75, -19.26, and 9.70 dB for DyAKNet; the variant without attribution also produces no finite target-tracking result. Under these tested variants, retaining the proposed coefficient-space attribution is associated with the strongest state-estimation performance.

Table 5: Ablation of the coefficient-space diagnostic score (MSE [dB]).

Method	Lorenz	Target Tracking	NCLT
DyAKNet	**-11.75**	**-19.26**	**9.70**
w/o Attr	0.68	NaN	18.41
LSTM	-6.94	-12.45	14.85
Transformer	-8.16	-15.09	13.71

Runtime is reported in Table 4. On NCLT, DyAKNet requires 0.75 s, compared with 0.68 s for IMM, 0.42 s for RC–EKF, and 3.09 s for OL-DPF. Thus the method adds computation relative to IMM and RC–EKF while remaining faster than OL-DPF on this benchmark.

6 Conclusion

We introduced DyAKNet for off-dynamics Kalman filtering, where the transition model is updated from the current observation and recursive history before state propagation. DyAKNet maps diagnostic innovations into coefficient-space evidence and adapts coefficient values and active support within a prescribed candidate representation. Experiments across coefficient evolution, structural changes, distribution shifts, and real-world localization support the effectiveness of the proposed same-recursion dynamics update across the tested settings. The current formulation assumes that the candidate representation contains the locally relevant dynamics, while future work will explore adaptive dynamics libraries and richer uncertainty modeling.

AI use statement

We use LLMs only for writing assistance. Specifically, LLMs are used to refine grammar, clarity, and presentation. No experimental results, methods, or scientific conclusions are generated by LLMs.

REFERENCES

Mohammad Al-Jarrah, Niyizhen Jin, Bamdad Hosseini, and Amirhossein Taghvaei. Nonlinear filtering with brenier optimal transport maps. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Henk AP Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988. doi: 10.1109/9.1299.

Marc Bocquet, Alban Farchi, and Quentin Malartic. Online learning of both state and dynamics using ensemble kalman filters. *Foundations of Data Science*, 3(3):305–330, 2021. doi: 10.3934/fods.2020015.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2015. doi: 10.1177/0278364915614638.

Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

Changhao Chen, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. Dynanet: Neural kalman dynamical model for motion estimation and prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5479–5491, 2021.

Xuxiang Fan, Gang Wang, Jiachen Han, and Yinghui Wang. Interacting multiple model based on maximum correntropy kalman filter. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(8):3017–3021, 2021.

Brian D Hassard, Nicholas D Kazarinoff, and Yieh-Hei Wan. *Theory and applications of Hopf bifurcation*, volume 41. CUP Archive, 1981.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. *arXiv preprint arXiv:1805.11122*, 2018.

Nikolas Kantas, Arnaud Doucet, Sumeetpal S. Singh, Jan M. Maciejowski, and Nicolas Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015. doi: 10.1214/14-STS511.

Genshiro Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, 1998. doi: 10.2307/2669862.

Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1161–1168. IEEE, 2022.

Jiaxi Li, Xiongjie Chen, and Yunpeng Li. Learning differentiable particle filter on the fly. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 1355–1361. IEEE, 2023. doi: 10.1109/IEEECONF59524.2023.10476913.

Tiancheng Li, Huimin Chen, Shudong Sun, and Juan M Corchado. Joint smoothing and tracking based on continuous-time target trajectory function fitting. *IEEE transactions on Automation Science and Engineering*, 16(3):1476–1483, 2018.

Tiancheng Li, Jingyuan Wang, Guchong Li, and Dengwei Gao. From target tracking to targeting track — part iii: Stochastic process modeling and online learning. *IEEE Transactions on Signal Processing*, 73, 2025.

Tiancheng Li, Yan Song, Hongqi Fan, and Jingdong Chen. From target tracking to targeting track — part i: a metric for spatio-temporal trajectory evaluation. *Information Fusion*, 127: 103890, 2026. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2025.103890. URL https://www.sciencedirect.com/science/article/pii/S1566253525009522.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130 – 141, 1963. doi: 10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2.

Zhixin Lu, Brian R. Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, 2018. doi: 10.1063/1.5039508.

Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2894–2901, 2021.

Niall M Mangan, Travis Askham, Steven L Brunton, J Nathan Kutz, and Joshua L Proctor. Model selection for hybrid dynamical systems via sparse regression. *Proceedings of the Royal Society A*, 475(2223):20180534, 2019.

Raman Mehra. Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 17(5): 693–698, 1972. doi: 10.1109/TAC.1972.1100100.

Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022.

Dhruvit Patel and Edward Ott. Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(2):023143, 2023. doi: 10.1063/5.0131787.

Dhruvit Patel, Daniel Canaday, Michelle Girvan, Andrew Pomerance, and Edward Ott. Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(3):033149, 2021. doi: 10.1063/5.0042598.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018. doi: 10.1103/PhysRevLett.120.024102.

Konrad Reif, Stefan Gunther, Engin Yaz, and Rolf Unbehauen. Stochastic stability of the discrete-time extended kalman filter. *IEEE Transactions on Automatic control*, 44(4):714–728, 1999.

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. doi: 10.1109/TSP.2022.3158588.

X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003. doi: 10.1109/TAES.2003.1261132.

Luca Rosafalco, Paolo Conti, Andrea Manzoni, Stefano Mariani, and Attilio Frangi. Online learning in bifurcating dynamic systems via sindy and kalman filtering. *Nonlinear Dynamics*, pp. 1–21, 2025.

Evgeny Evgenievich SEL'KOV. Self-oscillations in glycolysis 1. a simple kinetic model. *European Journal of Biochemistry*, 4(1):79–86, 1968.

Tan Shen, YunLong Dong, DingXin He, and Ye Yuan. Online identification of time-varying dynamical systems for industrial robots based on sparse bayesian learning. *Science China Technological Sciences*, 65(2):386–395, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Rui Wang, Robin Walters, and Rose Yu. Meta-learning dynamics forecasting using task inference. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21640–21653, 2022.

Alexander Wikner, Jaideep Pathak, Brian R. Hunt, Istvan Szunyogh, Michelle Girvan, and Edward Ott. Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5): 053114, 2021. doi: 10.1063/5.0048050.

Yushu Zhang, Gang Li, Xiao-Ping Zhang, and You He. A deep learning model based on transformer structure for radar tracking of maneuvering targets. *Information Fusion*, 103:102120, 2024. doi: 10.1016/j.inffus.2023.102120.

APPENDIX CONTENTS

- **A.** Extended Related Work
- **B.** Kalman Filtering Details
- **C.** DyAKNet Derivations and Training Details
- **D.** Finite-Horizon Filtering Error under Vector-Field Mismatch
- **E.** Experimental Descriptions for Synthetic Data
  - E.1 Coefficient Evolution in Nonlinear Systems
    - E.1.1 Model Descriptions of Evolving Dynamics
    - E.1.2 Baseline Implementation Details
  - E.2 Model Description of Target Tracking
    - E.2.1 State-Space Model for Target Tracking
    - E.2.2 Emergence of New Dynamic Terms Setting
    - E.2.3 Target Tracking Generalization Setting
  - E.3 RC–EKF Baseline Implementation
- **F.** Experimental Description for NCLT
  - F.1 Dataset Processing
  - F.2 System Dynamics Modeling
- **G.** Additional Discussion

A EXTENDED RELATED WORK

Black-box dynamics learning. Reservoir computing and recurrent models learn state-transition maps from informative trajectories. Early work demonstrated scalable model-free prediction of spatiotemporal chaos Pathak et al. (2018); subsequent studies connected recurrent dynamics to attractor reconstruction and long-term statistics Lu et al. (2018), and extended them to nonstationary processes and regime transitions Patel et al. (2021); Patel & Ott (2023). Meta-learning adapts dynamics predictors across tasks using offline trajectory collections Wang et al. (2022). End-to-end differentiable particle filters and Transformer-based trackers similarly represent transitions implicitly Jonschkowski et al. (2018); Zhang et al. (2024). These methods are expressive, but their learned dynamics and uncertainty representations can be coupled to the training distribution.

Sparse equation discovery. SINDy identifies governing equations by selecting active terms from a prescribed function library Brunton et al. (2016). Its explicit coefficient representation has been extended to controlled, time-dependent, model-selection, and online settings Champion et al. (2019); Mangan et al. (2019); Shen et al. (2022). Most equation-discovery methods use informative state trajectories or derivative estimates and separate identification from recursive filtering. An identified equation therefore does not generally receive immediate innovation feedback.

Filtering with uncertain transitions. Adaptive and dual filters track unknown parameters or noise statistics within prescribed transition structures Mehra (1972); Kitagawa (1998); Kantas et al. (2015). IMM filtering estimates probabilities over a finite bank of predefined transitions Blom & Bar-Shalom (1988); Fan et al. (2021). These approaches are effective when uncertain parameter locations, their evolution laws, or candidate modes are known, but do not discover admissible terms outside the assumed parameterization.

Data-driven filtering and online adaptation. Neural state-space models and differentiable particle filters typically learn latent transitions offline Jonschkowski et al. (2018); Chen et al. (2021). OL-DPF instead updates a differentiable particle filter during deployment Li et al. (2023), while fitting-of-trajectories methods reduce dependence on an explicit process model Li et al. (2018; 2026; 2025). Hybrid data-assimilation approaches learn corrections to imperfect forecasts or update dynamics through ensemble filtering Wikner et al. (2021); Bocquet et al. (2021). EKF-SINDy augments the filter state with selected SINDy coefficients Rosafalco et al. (2025). DyAKNet differs by inferring

both signed coefficient values and active terms from recursive history and a current local score, then using the revised transition within the same recursion.

B KALMAN FILTERING DETAILS

For completeness, given transition f and observation function h, the EKF prediction and innovation are

$$\hat{x}_t^- = f(\hat{x}_{t-1}), \quad P_t^- = F_t P_{t-1} F_t^\top + Q, \quad \delta y_t = y_t - h(\hat{x}_t^-), \quad S_t = H_t P_t^- H_t^\top + R. \tag{13}$$

The Kalman gain and posterior update are $K_t = P_t^- H_t^\top S_t^{-1}$, $\hat{x}_t = \hat{x}_t^- + K_t \delta y_t$, and $P_t = (I - K_t H_t) P_t^-$. Up to an additive constant, the per-step Gaussian innovation negative log-likelihood is $\ell_t^{\mathrm{NLL}} = \frac{1}{2} \delta y_t^\top S_t^{-1} \delta y_t + \frac{1}{2} \log |S_t|$.

C DYAKNET DERIVATIONS AND TRAINING DETAILS

Diagnostic pass. Using the preceding dynamics estimate, DyAKNet forms the diagnostic prediction and uncertainty $\hat{x}_{t,\mathrm{diag}}^- = f_{\hat{\Xi}_{t-1}}(\hat{x}_{t-1})$ and $P_{t,\mathrm{diag}}^- = F_{t,\mathrm{diag}} P_{t-1} F_{t,\mathrm{diag}}^\top + Q$, followed by $\delta y_{t,\mathrm{diag}} = y_t - h(\hat{x}_{t,\mathrm{diag}}^-)$ and $S_{t,\mathrm{diag}} = H_{t,\mathrm{diag}} P_{t,\mathrm{diag}}^- H_{t,\mathrm{diag}}^\top + R$. The realized preceding dynamics change used by the recurrent input is $\Delta\hat{\Xi}_{t-1} = \hat{\Xi}_{t-1} - \hat{\Xi}_{t-2}$, with zero initialization.

Coefficient-space score. For a local candidate coefficient matrix, define $\ell_t^{\mathrm{innov}}(\Xi) = \frac{1}{2} \delta y_{t,\mathrm{diag}}(\Xi)^\top S_{t,\mathrm{diag}}^{-1} \delta y_{t,\mathrm{diag}}(\Xi)$ while holding the diagnostic covariance fixed. Differentiating this mean-sensitive term yields the coefficient-space sensitivity used to construct G_t in Eq. (4). Because the covariance derivative is omitted, G_t is treated as structured diagnostic evidence rather than as the exact gradient of the full innovation likelihood. DyAKNet does not form the full instantaneous Fisher matrix; the local rank structure only motivates aggregating evidence across recursions and does not assert coefficient identifiability.

Training losses. The filtering-consistency term is $\mathcal{L}_{\mathrm{NLL}} = T^{-1} \sum_t \frac{1}{2} (\delta y_t^\top S_t^{-1} \delta y_t + \log |S_t|)$. The structural term in Eq. (12) uses $\mathcal{L}_{\mathrm{spar}} = T^{-1} \sum_t \|\hat{\Xi}_t\|_1$ and $\mathcal{L}_{\mathrm{smooth}} = (T-1)^{-1} \sum_{t=2}^{T} \|\hat{\Xi}_t - \hat{\Xi}_{t-1}\|_F^2$. Coefficient-supervised training uses $\mathcal{L}_{\mathrm{sup}} = T^{-1} \sum_t \|\hat{\Xi}_t - \Xi_{\mathrm{GT},t}\|_F^2$, whereas state-supervised training uses $\mathcal{L}_{\mathrm{sup}} = T^{-1} \sum_t \|\hat{x}_t - x_{\mathrm{GT},t}\|_2^2$.

D FINITE-HORIZON FILTERING ERROR UNDER VECTOR-FIELD MISMATCH

This appendix proves Proposition 4.1. The argument follows local stochastic EKF analyses Reif et al. (1999) while treating the learned vector-field mismatch as an observation-adapted perturbation of the error recursion. In particular, $\hat{\Xi}_t$ and the resulting mismatch may depend on the current diagnostic observation through $\mathcal{F}_t$.

D.1 PREDICTION-ERROR DECOMPOSITION

Let $e_t = x_t - \hat{x}_t$ and let Ξ_t denote the true coefficient matrix. The prediction error generated by the adapted transition is

$$e_t^- = f_{\Xi_t}(x_{t-1}) - f_{\hat{\Xi}_t}(\hat{x}_{t-1}) + w_t = F_t^\star e_{t-1} + d_t + r_t^f + w_t. \tag{14}$$

where $F_t^\star$ is the Jacobian of f_{Ξ_t} evaluated at $\hat{x}_{t-1}$,

$$d_t = \Delta t \, (\Xi_t - \hat{\Xi}_t)^\top \Theta(\hat{x}_{t-1}), \tag{15}$$

and r_t^f is the transition linearization remainder. The perturbation d_t is not assumed independent of the current process or measurement noise. The smoothness assumptions imply

$$\|r_t^f\|_2 \le L_f \|e_{t-1}\|_2^2, \qquad \|d_t\|_2 = \Delta t \, \varepsilon_{a,t}. \tag{16}$$

Linearizing the observation function around $\hat{x}_t^-$ gives

$$h(x_t) - h(\hat{x}_t^-) = H_t e_t^- + r_t^h, \qquad \|r_t^h\|_2 \le L_h \|e_t^-\|_2^2. \tag{17}$$

Using the EKF measurement update, the posterior error satisfies

$$e_t = A_t e_{t-1} + B_t d_t + B_t r_t^f + B_t w_t - K_t v_t - K_t r_t^h. \tag{18}$$

where

$$A_t = (I_m - K_t H_t) F_t^\star, \qquad B_t = I_m - K_t H_t. \tag{19}$$

Here, $e_t, e_t^-, d_t, r_t^f, w_t \in \mathbb{R}^m$, while $v_t, r_t^h \in \mathbb{R}^n$. Moreover, $F_t^\star, A_t, B_t \in \mathbb{R}^{m\times m}$. Equation (18) shows directly that vector-field mismatch enters the filter through d_t before the measurement correction.

D.2 Proof of Proposition 4.1

Uniform bounds on P_t, P_t^-, H_t, and K_t imply uniform bounds on B_t and K_t. By the assumed local contraction of the closed-loop linearization, there exists an equivalent quadratic norm and a constant $\rho_0 < 1$ such that

$$\mathbb{E}\|A_t e_{t-1}\|_2^2 \le \rho_0 \mathbb{E}\|e_{t-1}\|_2^2. \tag{20}$$

Uniform equivalence between this quadratic norm and the Euclidean norm is absorbed into the constants below. Applying Young's inequality to every cross term in (18), together with the uniform matrix bounds and the finite second moments of w_t and v_t, yields constants $c_a, c_q, c_r, c_f, c_h > 0$ such that

$$\mathbb{E}\|e_t\|_2^2 \le \rho_0 \mathbb{E}\|e_{t-1}\|_2^2 + c_a \mathbb{E}\|d_t\|_2^2 + c_q \operatorname{tr}(Q) + c_r \operatorname{tr}(R) + c_f \mathbb{E}\|e_{t-1}\|_2^4 + c_h \mathbb{E}\|e_t^-\|_2^4. \tag{21}$$

Within a sufficiently small neighborhood, the fourth-order remainder terms can be absorbed into the contracting quadratic term. Hence, for some $\rho \in (\rho_0, 1)$,

$$\mathbb{E}\|e_t\|_2^2 \le \rho \mathbb{E}\|e_{t-1}\|_2^2 + c_a \Delta t^2 \mathbb{E}\varepsilon_{a,t}^2 + c_q \operatorname{tr}(Q) + c_r \operatorname{tr}(R). \tag{22}$$

which proves (11) without requiring independence between d_t and the current noise. Iterating this recursion gives the corresponding finite-horizon bound. □

E Experimental Descriptions for Synthetic Data

This section provides detailed descriptions of the synthetic-data experiments considered in this work, including the *Coefficient Evolution* experiments in Subsection 5.1 and the *Target Tracking* experiments in Subsection 5.2. The focus is on the underlying nonlinear models, their state-space formulations, and the associated experimental configurations. For all synthetic experiments, we adopt the linear observation model in (1), specifically $h(\cdot) = I_n$.

E.1 Coefficient Evolution in Nonlinear Systems

This subsection elaborates on the synthetic datasets used in the **Coefficient Evolution in Nonlinear Systems** experiments introduced in Subsection 5.1. We consider three representative nonlinear dynamical systems—Selkov, Hopf, and Lorenz—which are widely adopted benchmarks for evaluating filtering algorithms under strong nonlinearity, oscillatory behavior, and chaotic dynamics. For each system, a specific model parameter is designed to evolve over time, thereby inducing evolving system dynamics.

E.1.1 Model Descriptions of Evolving Dynamics

Selkov Model The Selkov model SEL'KOV (1968) is a canonical two-dimensional nonlinear biochemical system originally proposed to describe oscillatory behavior in glycolysis. In this work, it is adopted as a benchmark nonlinear state-space model to evaluate filtering performance under strong nonlinear interactions. The system states $x(t)$ and $y(t)$ represent the concentrations of adenosine diphosphate (ADP) and fructose-6-phosphate (F6P), respectively.

The continuous-time dynamics of the Selkov system are given by

$$[\dot{x}(t), \dot{y}(t)]^\top = [-x(t) + a\, y(t) + x^2(t)y(t),\ b(t) - a\, y(t) - x^2(t)y(t)]^\top. \tag{23}$$

where a and $b(t)$ denote model parameters associated with the reaction kinetics. The linear coupling term governed by a influences both the production of $x(t)$ and the depletion of $y(t)$, while the nonlinear interaction $x^2(t)y(t)$ introduces autocatalytic feedback, constituting the primary source of oscillatory and strongly nonlinear behavior.

In the coefficient evolution setting, the parameter $b(t)$ is treated as a time-varying quantity, gradually evolving from 0.3 to 0.9 during filtering. Unless otherwise specified, a is fixed to 0.1, and the system is discretized using a sampling interval of $\Delta t = 0.02$.

Hopf Oscillator The Hopf oscillator Hassard et al. (1981) is a prototypical two-dimensional nonlinear system commonly used to study limit-cycle oscillations and bifurcation phenomena. It serves as a benchmark model for assessing filtering performance under oscillatory and nonstationary dynamics. The system states $x(t)$ and $y(t)$ correspond to the two planar components of the oscillator.

The continuous-time Hopf dynamics are described by

$$[\dot{x}(t), \dot{y}(t)]^\top = [\mu(t)x(t) - \omega y(t) - (x^2(t) + y^2(t))x(t),\ \omega x(t) + \mu(t)y(t) - (x^2(t) + y^2(t))y(t)]^\top. \tag{24}$$

where $\mu(t)$ controls the growth rate of the oscillations and determines the emergence of a stable limit cycle, while ω specifies the intrinsic angular frequency.

The linear terms governed by $\mu(t)$ regulate exponential growth or decay, whereas the cubic nonlinear terms provide amplitude saturation and stabilize the trajectory onto a limit cycle when $\mu(t) > 0$. In the coefficient evolution experiments, $\mu(t)$ is designed to vary over time from -0.5 to 2.0, inducing a gradual transition from stable equilibrium to sustained oscillations. Unless otherwise stated, ω is fixed to 1.0, and the sampling interval is set to $\Delta t = 0.02$.

Lorenz System The Lorenz system Lorenz (1963) is a three-dimensional chaotic dynamical system exhibiting sensitive dependence on initial conditions. Owing to its rich nonlinear and chaotic behavior, it has become a standard benchmark for evaluating state estimation and filtering algorithms in nonlinear systems Al-Jarrah et al. (2024); Revach et al. (2022).

The continuous-time Lorenz dynamics are governed by the following ordinary differential equations:

$$[\dot{x}(t), \dot{y}(t), \dot{z}(t)]^\top = [\sigma(y(t) - x(t)),\ x(t)(\rho(t) - z(t)) - y(t),\ x(t)y(t) - \beta z(t)]^\top. \tag{25}$$

where $x(t) = [x(t), y(t), z(t)]^\top$ denotes the system state, and σ, $\rho(t)$, and β are model parameters.

In the coefficient evolution setting, the Rayleigh parameter $\rho(t)$ is treated as a time-varying coefficient, gradually evolving from 15 to 30, thereby inducing nonstationary and increasingly chaotic system behavior. Unless otherwise specified, the parameters are fixed as $\sigma = 10$ and $\beta = 8/3$. The continuous-time dynamics are discretized with a sampling interval of $\Delta t = 0.02$, and additive process noise is incorporated during discretization.

E.1.2 Baseline Implementations Details

Since our focus is on evolving system dynamics, we assume the current noise statistics are known, i.e., Q and R are given.

For all three nonlinear dynamical systems considered, the parameters governing the *Coefficient Evolution*—denoted as θ—vary over time from θ_{start} to θ_{end}.

For the FoT, each trajectory is treated as a function of time and fitted online using a library consisting of a constant term, t, and t^2. In the IMM, a prior library of motion models is required. Here, we select three models corresponding to the beginning, middle, and end values of the evolving parameter, θ_{start}, θ_{mid}, and θ_{end}, respectively. The model transition probabilities are set to be uniform. For EKF-SINDy, the system is modeled following the standard SINDy approach. The prior library is constructed to include the terms corresponding to the evolving system dynamics, reflecting the anticipated changes in the coefficients over time.

E.2 Model Description of Target Tracking

E.2.1 State-space model for Target Tracking

This subsection provides the state-space formulations for the three canonical target tracking motion models considered in our experiments: Constant Velocity (CV), Constant Acceleration (CA), and Coordinated Turn (CT). All switching experiments use a unified state representation throughout filtering; the equations below use compact regime-specific notation to describe the corresponding motion dynamics.

Constant Velocity (CV) Model The CV model assumes that the target moves with constant velocity along each dimension. Denoting the state as $x(t) = [p_x(t), p_y(t), v_x(t), v_y(t)]^\top$, where p_x, p_y are positions and v_x, v_y are velocities, the continuous-time dynamics are

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + w(t), \tag{26}$$

where $w(t) \sim \mathcal{N}(0, Q)$ is process noise.

Constant Acceleration (CA) Model The CA model extends the CV model by including constant acceleration. The state is $x(t) = [p_x(t), p_y(t), v_x(t), v_y(t), a_x(t), a_y(t)]^\top$, with a_x, a_y as accelerations. The dynamics are

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t) + w(t), \tag{27}$$

where $w(t)$ represents additive process noise.

Coordinated Turn (CT) Model The CT model describes planar motion with constant speed and a constant turn rate ω. The state is $x(t) = [p_x(t), p_y(t), v_x(t), v_y(t), \omega(t)]^\top$, where (p_x, p_y) are positions and (v_x, v_y) are velocity components. The continuous-time dynamics are

$$[\dot{p}_x(t), \dot{p}_y(t), \dot{v}_x(t), \dot{v}_y(t)]^\top = [v_x(t),\ v_y(t),\ -\omega(t)v_y(t),\ \omega(t)v_x(t)]^\top. \tag{28}$$

with $\omega(t)$ as the turn rate. The process noise $w(t)$ is added to account for model uncertainty. In the limit $\omega \to 0$, this model naturally reduces to the Constant Velocity (CV) model.

E.2.2 Emergence of New Dynamic Terms Setting

In certain scenarios, system dynamics may include an admissible effect that is absent from the nominal motion model. A representative example is linear damping (e.g., viscous aerodynamic drag or friction) during the constant-acceleration (CA) phase. In this experiment, the damping term is included in the prescribed candidate representation but is inactive in the nominal CA model; its activation therefore creates a structural mismatch relative to the nominal transition.

We model the damped CA dynamics as follows:

$$[\dot{p}, \dot{v}]^\top = [v,\ a - \mu v]^\top. \tag{29}$$

where $p = [x, y]^\top$ denotes position, $v = [v_x, v_y]^\top$ denotes velocity, $a = [a_x, a_y]^\top$ is the commanded acceleration, and $\mu > 0$ is the linear damping coefficient. The damping term $-\mu v$ introduces velocity-proportional resistive forces, simulating environmental effects such as air or water drag. We set $\mu = 1$ to simulate strong damping, while retaining the nominal acceleration-driven evolution. This setting provides a controlled testbed to evaluate state estimation when a dynamic term absent from the nominal motion model emerges.

Physical Interpretation. The damping coefficient μ represents the magnitude of environmental resistance. When $\mu = 0$, the system reduces to the standard CA model. When $\mu > 0$, the resistive forces act in proportion to the velocity, causing the target to decelerate naturally even in the presence of constant commanded acceleration. This creates a scenario where filters relying solely on the nominal CA model may systematically misestimate the state unless the emerging term $-\mu v$ is properly accounted for.

E.2.3 TARGET TRACKING GENERALIZATION PERFORMANCE SETTING

To test generalization beyond the training conditions, we evaluate DyAKNet under unseen noise levels and switching patterns. The experiments are divided into two categories: *Unseen Noise Intensity* and *Unseen Dynamics Switching*.

Unseen Noise Intensity The network is trained under a nominal noise level of $1/\mathrm{q}^2 = 1/\mathrm{r}^2 = 0$ dB, with noise covariances $Q = \mathrm{q}^2 I$ and $R = \mathrm{r}^2 I$. To test robustness to previously unseen measurement noise, evaluation is performed with $1/\mathrm{r}^2 \in \{-10, 10, 20, 30\}$ dB. This setting probes DyAKNet's ability to generalize under variations in observation noise intensity that were not encountered during training.

Unseen Dynamics Switching Training data consist of target trajectories that switch among motion models in a nominal sequence, e.g., starting with CV, switching to CA at $t = 60$, and then to CT at $t = 80$.

We define three types of *Unseen Dynamics Switching* for testing:

- **Unseen Switching Sequence:** the same switch times as training but with a different order, e.g., CV $\rightarrow$ CT at $t = 60$, then CT $\rightarrow$ CV at $t = 80$.
- **Unseen Switching Timing:** the same switch order as training but different switch times, e.g., CV $\rightarrow$ CA at $t = 50$, then CA $\rightarrow$ CT at $t = 70$.
- **Unseen Switching Time+Seq:** both the switch order and times are changed, e.g., CV $\rightarrow$ CT at $t = 50$, then CT $\rightarrow$ CV at $t = 70$.

These settings evaluate whether the learned dynamics-update mechanism remains effective under temporal switching patterns not used during training.

E.3 RC–EKF BASELINE IMPLEMENTATION

RC–EKF adapts the time-conditioned reservoir-computing predictor of Patel and Ott Patel & Ott (2023) to recursive state estimation. The reservoir computer learns an implicit discrete-time transition from the complete state trajectories in the synthetic training sets. For trajectory i of length T_i, we construct the normalized time signal

$$s_{i,t} = 2\frac{t-1}{T_i - 1} - 1, \qquad t = 1, \ldots, T_i. \tag{30}$$

Let $r_{i,t} \in \mathbb{R}^{d_r}$ denote the reservoir state. Given the preceding true state $x_{i,t-1}$ during training, the reservoir update is

$$r_{i,t} = (1-\alpha) r_{i,t-1} + \alpha \tanh\!\big(A r_{i,t-1} + W_{\mathrm{in}}[x_{i,t-1}^\top, s_{i,t-1}]^\top + b_r 1_{d_r}\big)\,. \tag{31}$$

where $A \in \mathbb{R}^{d_r \times d_r}$ is the fixed reservoir matrix, $W_{\mathrm{in}} \in \mathbb{R}^{d_r \times (m+1)}$ is the fixed input matrix, $\alpha \in (0, 1]$ is the leakage rate, b_r is a bias, and 1_{d_r} is the all-ones vector. The output feature and one-step state prediction are

$$\phi_{i,t} = [r_{i,t}^\top, x_{i,t-1}^\top, s_{i,t-1}, 1]^\top, \qquad \hat{x}_{i,t}^{\mathrm{RC}} = W_{\mathrm{out}} \phi_{i,t}. \tag{32}$$

Here, $\phi_{i,t} \in \mathbb{R}^{d_r+m+2}$ and $W_{\mathrm{out}} \in \mathbb{R}^{m \times (d_r+m+2)}$. Only the output matrix is fitted, using ridge regression over all valid transition pairs:

$$W_{\mathrm{out}} = \arg\min_W \sum_{i,t} \|x_{i,t} - W\phi_{i,t}\|_2^2 + \lambda_{\mathrm{RC}} \|W\|_F^2. \tag{33}$$

The reservoir state is reset for each trajectory, an initial washout segment is discarded, and the remaining transition pairs are pooled when estimating W_{out}.

During testing, the preceding filtered estimate replaces the true state in (31). The resulting deterministic transition f_{RC} provides the EKF prediction

$$\hat{x}_t^- = f_{\text{RC}}(\hat{x}_{t-1}, r_{t-1}, s_{t-1}), \tag{34}$$

with transition Jacobian

$$F_t^{\text{RC}} = \left.\frac{\partial f_{\text{RC}}(x, r_{t-1}, s_{t-1})}{\partial x}\right|_{x=\hat{x}_{t-1}}. \tag{35}$$

The covariance prediction is then

$$P_t^- = F_t^{\text{RC}} P_{t-1} (F_t^{\text{RC}})^\top + Q, \tag{36}$$

after which the standard EKF measurement update described in Appendix B produces $(\hat{x}_t, P_t)$. The posterior state is fed back to the reservoir at the next recursion. The stochastic residual-resampling step used by Patel and Ott for long-term climate generation is omitted because process uncertainty is represented by Q in the EKF. RC–EKF is evaluated in the synthetic experiments reported in Tables 1–3.

F Experimental Description for the NCLT Dataset

F.1 Dataset processing

The University of Michigan North Campus Long-Term Vision and LIDAR (NCLT) dataset is adopted in this study to evaluate performance on real-world data. We utilize the vehicle odometry measurements provided by the dataset, with velocity information serving as the primary input for state estimation. High-precision ground truth trajectories are obtained from the integrated RTK-GPS system.

Our experiments focus on data collected by the Segway mobile robot on January 22, 2012. To mitigate the influence of sensor instability, only the initial portion of the sequence is retained: the first 20,000 samples are selected and uniformly downsampled to 5 Hz, resulting in a sequence of 1,000 time steps. This sequence is segmented into 25 non-overlapping trajectories of length 40. Following standard practice, the trajectories are split into training, validation, and test sets, comprising 15, 5, and 5 trajectories, respectively.

F.2 System Dynamics Modeling

For the NCLT dataset experiments, the system state is defined as $x_t = (p_1, p_2, v_1, v_2)^\top \in \mathbb{R}^4$, where p_1, p_2 denote the planar position and v_1, v_2 the corresponding velocities of the Segway robot.

Model-based baselines, including the Interacting Multiple Model (IMM) filter and EKF-SINDy, require an explicit prior state-space model (SSM). We model the Segway dynamics using a linear Wiener velocity model:

$$x_t = F x_{t-1} + w_t, \tag{37}$$

where the process noise satisfies $w_t \sim \mathcal{N}(0, Q)$, and the state transition matrix and process noise covariance are given by

$$F = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q = q^2 \begin{bmatrix} \frac{\Delta^3}{3} & 0 & \frac{\Delta^2}{2} & 0 \\ 0 & \frac{\Delta^3}{3} & 0 & \frac{\Delta^2}{2} \\ \frac{\Delta^2}{2} & 0 & \Delta & 0 \\ 0 & \frac{\Delta^2}{2} & 0 & \Delta \end{bmatrix}, \tag{38}$$

with $\Delta = 1$ denoting the sampling interval.

In this setting, the objective is to estimate the robot position using odometry measurements, which provide velocity observations only. The corresponding measurement model is

$$y_t = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_t + v_t, \tag{39}$$

where the measurement noise follows $v_t \sim \mathcal{N}(0, R)$ with covariance $R = \mathrm{r}^2 I_2$, consistent with the two-dimensional velocity observations.

Data-driven baselines, including FoT and DyAKNet, do not require the current transition coefficients to be provided at test time. DyAKNet operates with the prescribed candidate representation and the same noise statistics as above while estimating a time-varying coefficient realization from data.

G Additional Discussion

G.1 Limitations and Scope

DyAKNet assumes that a prescribed candidate library contains the locally relevant dynamics. Its filter uses a local Gaussian approximation, and the EKF covariance does not propagate uncertainty in the estimated coefficient matrix. The empirical study covers low-dimensional nonlinear systems, target tracking, and planar robot localization; scaling to substantially larger libraries or high-dimensional latent states may require structured libraries and explicit model-uncertainty propagation. Non-Gaussian observations and abrupt out-of-library dynamics remain important failure modes for future study.