This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TCYB.2025.3632756, IEEE Transactions on Cybernetics

Hybrid-Driven State Estimation with Adaptive Cross-Coupled Priors: Enhancing Data Representation and Model Robustness

Lizhang Wang, Zidong Wang, and Qinyuan Liu

***Abstract*—This paper addresses the integration of model-driven and data-driven approaches for robust hybrid-driven state estimation under limited data and model uncertainties. An unsupervised hybrid estimation framework, termed *A*daptive *M*odel-driven and *D*ata-driven (AMD), is proposed. AMD employs an adaptive cross-coupled prior mechanism within the Bayesian inference paradigm to integrate prior information. A two-stage fusion strategy is introduced: an initial hard fusion of model pseudo-measurements and data-driven priors, followed by an adaptive soft fusion that adjusts model influence based on reconstruction discrepancies, thereby enhancing robustness to imperfect model priors. To capture complex nonlinear transition dynamics, a dynamic bilinear recurrent module has been developed, tailored to the system's underlying behavior. The AMD framework adopts a non-identical training-testing strategy and an unsupervised hybrid learning objective inspired by the information bottleneck principle, enabling accurate parameter learning without access to ground-truth states. Extensive experiments on multiple nonlinear chaotic systems have demonstrated that AMD consistently achieves competitive or superior estimation accuracy compared to state-of-the-art model-based and hybrid approaches, particularly under underdetermined estimation, model mismatch, and dynamic disturbances. These results demonstrate AMD's capability to effectively leverage limited information through complementary fusion, thereby enhancing both data representation and model robustness. This adaptability positions AMD as a powerful solution for challenging state estimation problems.**

***Index Terms*—hybrid-driven state estimation, cross-coupled Bayesian inference, unsupervised learning, dynamic bilinear data driven, adaptive weight adjustment, complementary prior fusion**

I. Introduction

Accurate state estimation is regarded as fundamental to the effective analysis, prediction, and control of complex dynamic systems across various engineering domains [2], [25], [46], [61]. In state estimation methods, prior knowledge derived from models or data is combined with real-time sensor measurements to reconstruct unobservable internal system states [7], [14], [18]. Such methods have been widely applied in fields such as signal processing and communications [55], [64], dynamic scene monitoring, robotics [26], [56], and automatic control [51], [59].

Traditional state estimation approaches are based on two primary forms of prior knowledge: model-driven system modeling and data-driven evolutionary fitting. In model-driven methods, prior dynamical knowledge is employed to construct system models and identify model parameters, with estimation accuracy and stability ensured through precise system representation. However, these methods are typically characterized by a strong dependence on model robustness. They implicitly assume access to a complete mathematical characterization of system dynamics, including accurate transition and observation models as well as well-defined noise statistics, assumptions that are rarely satisfied in practice. Limited domain knowledge, unmodeled dynamics, or oversimplified noise assumptions inevitably introduce modeling bias and parameter uncertainty, which in turn lead to systematic errors and reduced estimation reliability. Moreover, in high-dimensional, nonlinear, or chaotic systems, conventional models typically capture only partial aspects of the underlying processes. Such limitations can critically undermine the effectiveness of model-driven state estimation and, under extreme conditions, may even cause complete estimation failure, thereby restricting the practical applicability of purely model-driven estimators.

Data-driven methods, on the other hand, are designed to exploit historical data to infer the internal system evolution as a data prior, thereby enabling the complex characteristics of the system to be captured through large-scale, high-quality datasets. However, historical data are often unaccompanied by ground-truth state annotations and are corrupted by measurement noise. These limitations in data representation capacity prevent data-driven methods from achieving supervised and interpretable learning of the system's internal dynamics. Furthermore, scenarios involving restricted data representations (e.g. underdetermined estimation problems where high-dimensional latent states must be inferred from low-dimensional noisy measurements [22]) further exacerbate estimation uncertainty and degrade learning performance.

Given the limitations of both model-driven and data-driven approaches, hybrid-driven state estimation has been explored, in which complementary priors are integrated to leverage the advantages of both strategies while mitigating their respective drawbacks. In existing hybrid-driven methods, one prior is typically used to refine or supplement the other; for example, data-driven techniques may be employed to capture system dynamics that cannot be explicitly modeled, while model-

This work was supported in part by the National Science Foundation of China under Grants 62222312 and 62473285, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. *(Corresponding author: Qinyuan Liu)*

Lizhang Wang and Qinyuan Liu are with the School of Computer Science and Technology, Tongji University, Shanghai 201804, China. (Email: liuqy@tongji.edu.cn).

Zidong Wang is with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (Email: Zidong.Wang@brunel.ac.uk).

driven methods may be used to enhance or constrain data-driven estimation results. Although such fusion strategies have improved estimation performance by addressing the limitations of individual approaches, the inherent weaknesses of both paradigms are inevitably inherited, thereby introducing new challenges rather than fundamentally resolving the complexities of real-world estimation tasks.

Motivated by the challenges in state estimation and the limitations of existing hybrid-driven approaches, a novel *A*daptive *M*odel-driven and *D*ata-driven Cross-coupled Estimator (AMD) is proposed. The key contributions are summarized as follows.

1) AMD is proposed as an adaptive cross-coupled framework that integrates model priors and data priors through a 'conditional hard fusion + adjustment + adaptive soft fusion' strategy. Known model knowledge is exploited to constrain data representation, while data fitting is simultaneously employed to compensate for model incompleteness and uncertainty. Through the complementary fusion of these limited information sources, estimation accuracy is enhanced in a principled and efficient manner, with improvements also achieved in data representation and model robustness.
2) A dynamic bilinear-driven component is introduced to address complex system nonlinearities. To dynamically regulate the contributions of model-driven and data-driven components in hybrid estimation, an adaptive weight adjustment strategy is developed. This strategy employs a non-identical training and testing mechanism to balance modeling constraints and estimation performance.
3) An unsupervised hybrid learning framework is formulated from an information bottleneck perspective, inspired by variational autoencoder (VAE) optimization. This design ensures the interpretability and accuracy of hybrid-driven estimation even in the absence of ground-truth state labels.
4) Comprehensive comparative experiments are conducted to evaluate the robustness of AMD. The results validate its effectiveness and broad applicability across various state estimation scenarios, including those involving limited observational data.

A. Related Work

Current mainstream state estimation methods can be broadly categorized into three types according to the utilized prior knowledge: model-driven, data-driven, and hybrid-driven approaches.

Traditional model-driven methods are primarily based on comprehensive mathematical models constructed from domain-specific expertise to characterize system dynamics. State estimation is performed through explicitly formulated state-space models, such as the classical Kalman Filter (KF) [58], the unscented KF (UKF) [35], [42], [62], the sampling-based Particle Filter (PF) [1], [54], and the Adaptive Kalman Filter (AKF) [20], which is capable of identifying unknown system parameters.

Emerging data-driven methods, by contrast, rely on implicit system behaviors captured from extensive historical datasets to replace explicit model knowledge. These methods typically employ deep neural networks to construct both transition and observation models implicitly [17], [30]. Examples include kernel-based approaches for learning latent representations of high-dimensional systems [4], and variational inference techniques for identifying underlying system models [23].

Hybrid-driven methods are developed to integrate the strengths of both model-driven and data-driven approaches. By jointly exploiting their respective priors, these approaches enhance state estimation performance, combining the robustness and interpretability of models with the flexibility and expressiveness of data, while offering improved generalization across varying conditions. These hybrid strategies can be classified into two categories based on the integration sequence:

- *Model-driven enhancements of data-driven methods (MD-DD)*: These strategies are designed to capitalize on the robustness and low complexity of model-driven techniques to refine or constrain initial data-driven estimations. One group of such approaches incorporates data-driven filtering or smoothing into Bayesian (model-driven) frameworks [13], [21], [22], [32], [53], [60]. In these approaches, a data-driven predictor (such as a neural network) is initially employed to produce coarse state estimates. These estimates are then refined within a Bayesian inference framework to enforce consistency with the established measurement model. For example, in Danse [21], a recurrent neural network is employed to generate prior predictions, which are subsequently refined through Bayesian updating using known measurement models. Another category relies on physics-informed machine learning principles [12], [29], [33], [41], [43], [47], in which established physical models are embedded either internally within the network architecture or enforced through constraints in the loss functions. This integration enhances interpretability and reduces the search space for the data-driven methods.
- *Data-driven enhancements of model-driven methods (DD-MD)*: Conversely, these strategies employ either internal or external fusion mechanisms, whereby the generalization capabilities of data-driven techniques are leveraged to enhance the performance of model-driven estimation. One class of such approaches is based on residual complementarity, in which data-driven components are trained to learn the discrepancies between preliminary model-driven estimates and the actual states, thereby refining the results [8], [19], [52], [63]. For instance, in [19], a graph neural network framework is utilized to construct data-driven residual messages that complement the initial model-driven outcomes. Another approach focuses on addressing unknown parameters by embedding data-driven components within model-driven frameworks, where learned modules are used to estimate uncertain parameters in physical models [3], [11], [45], [48]–[50]. The notable KalmanNet [48], for example, integrates the data-driven learning of the Kalman gain within a classical Kalman filtering framework.

Table I presents a comparative summary of attributes for

Attribute	Methods				
	Model-driven	Data-driven	Hybrid-driven		**AMD**
			MD-DD	DD-MD	
	e.g. UKF, etc.	e.g. RNNs, etc.	e.g. Danse, etc.	e.g. KalmanNet, etc.	
Domain knowledge					
Transition model	✓	✗	✓/ ✗	✓	✓
Measurement model	✓	✗	✓	✓ / ✗	✓
Problems with model priors					
Model uncertainties	✗	✓	✓	✓	✓
Model incompleteness	✗	✓	✓	✗	✓
Problems with data priors					
Real state labels	✗	✗	✗	✓ / ✗	✓
Underdetermined estimation	✓	✗	✗	✓ / ✗	✓
Closed-form posterior	✓	✗	✓ / ✗	✓ / ✗	✓

TABLE I
Visual comparison of method characteristics.'✓' = required/addressed;'✗' = not required/unaddressed; ' ✓' = partially required/partially addressed.

model-driven, data-driven, hybrid-driven, and AMD methods.

B. Notations

Throughout this paper, scalars and vectors are denoted by regular and bold lowercase letters, respectively, while matrices are represented by bold uppercase letters. The transpose of a matrix is denoted as $(\cdot)^\top$. The ℓ_2-norm of a vector is represented as $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{x}$, and similarly, a weighted Euclidean norm is defined as $\|\mathbf{A}\|_{\mathbf{B}}^2 = \mathbf{A}^\top\mathbf{B}\mathbf{A}$. The identity matrix of size $n \times n$ is denoted by $\mathbf{I}_n$. The notation $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a Gaussian distribution, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote its mean and covariance, respectively. The operator $\mathbb{E}_{p(x)}[\cdot]$ denotes the expectation with respect to the distribution $p(x)$. A sequence $\{\mathbf{x}_1, \mathbf{x}_1, \cdots, \mathbf{x}_t\}$ is denoted as $\mathbf{x}_{1:t}$. The symbol $\mathbb{R}$ denotes the set of real numbers.

C. Structure of the Article

The remainder of this paper is structured as follows. Section II provides a comprehensive overview of the problem formulation and offers a detailed description of the proposed AMD architecture, including its methodology and associated operations. In Section III, experimental results are presented along with a comparative analysis of AMD's performance against other related methods. Finally, in Section IV, the main findings are summarized, and potential directions for future research are discussed.

II. Proposed AMD

A. Problem Formulation

In this study, we focus on state estimation within a finite discrete-time horizon to facilitate theoretical derivations, though the framework can be readily extended to the infinite-horizon setting. Let $\mathbf{x}_t \in \mathbb{R}^m$ denote the state vector of a dynamical system at time step t, which evolves over time following an underlying transition process:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_{1:t}) + \boldsymbol{\xi}_{t+1}, t = 1, 2, \ldots, T. \tag{1}$$

where $f(\cdot)$ represents the state transition process, which is typically complex and difficult to fully characterize through explicit modeling, and is often treated as *unknown*. The term $\boldsymbol{\xi}_{t+1}$ denotes the system transition noise, capturing the stochastic perturbations in the state evolution. During inference, the true system states remain inaccessible. Instead, we rely on available linear measurements $\mathbf{y}_t \in \mathbb{R}^n$ to infer the latent states:

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t, t = 1, 2, \ldots, T. \tag{2}$$

Here, $\mathbf{H}_t \in \mathbb{R}^{n\times m}$ represents the known measurement matrix. The term $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ denotes zero-mean Gaussian measurement noise with covariance $\mathbf{R}_t$. It is worth noting that such measurements may fail to fully capture the underlying system states, particularly under limited data representation conditions, such as in underdetermined estimation scenarios (i.e., $n < m$), where $\mathbf{H}_t$ is a fat matrix.

In many real-world scenarios, only partial knowledge of the system's transition dynamics is available. To effectively leverage this partial model-based prior, we introduce a model-representable latent state, denoted as $\mathbf{z}_t \in \mathbb{R}^r$, which characterizes the tractable portion of the system's internal evolution. In contrast to the system state $\mathbf{x}_t$, which is the ultimate target of estimation, the intermediate state $\mathbf{z}_t$ captures a projection or a model-aligned representation of the true system state. It reflects the component that can be described or approximated by an available model. The known component of the dynamics can be modeled as follows:

$$\begin{aligned} \mathbf{z}_{t+1} &= g(\mathbf{z}_t), t = 1, 2, \ldots, T. \\ \mathbf{z}_t &= \mathbf{M}_t\mathbf{x}_t + \mathbf{w}_t \end{aligned} \tag{3}$$

The mapping matrix $\mathbf{M}_t \in \mathbb{R}^{r\times m}$ projects the full state $\mathbf{x}_t$ into a latent subspace that reflects the model-representable or known component of the system dynamics. The term $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ represents the modeling noise associated with the latent state $\mathbf{z}_t$, characterized by covariance $\mathbf{Q}_t$. It captures the residual uncertainty introduced by projection or model

approximation. For numerical stability and subsequent derivations, the noise covariance $\mathbf{Q}_t$ is assumed to be a positive-definite matrix. While related to $\boldsymbol{\xi}_t$, $\mathbf{w}_t$ is not identical; rather, it can be interpreted as the simplified, modelable projection of $\boldsymbol{\xi}_t$ onto the model-representable subspace. Unlike the full but typically unknown dynamics of the true state $\mathbf{x}_t$ characterized by Eq. (1), the function $g(\cdot)$ represents a known (though possibly coarse or incomplete) transition function that governs the temporal evolution of the model-representable latent state $\mathbf{z}_t$. This provides a principled way of incorporating partial model knowledge into the Bayesian framework. This evolution is modeled as a Markov process, implying that the future latent state $\mathbf{z}_{t+1}$ depends solely on the current state $\mathbf{z}_t$ and is conditionally independent of the past states $\mathbf{z}_{1:t-1}$. When $g(\cdot)$ is nonlinear, it can be locally linearized using its Jacobian matrix, defined as $\mathbf{G}_t = \frac{\partial g}{\partial \mathbf{z}_t}|_{\mathbf{z}_t=\hat{\mathbf{z}}_t}$.

The *objective* of state estimation is to reconstruct the latent system state $\mathbf{x}_t$ using historical measurements $\mathbf{y}_{1:t}$ by inferring the conditional probability distribution $p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t}\right)$. Traditional model-driven approaches within the Bayesian inference framework rely on complete and accurate transition models. In contrast, classical data-driven methods depend on well-annotated datasets and strong feature representation capabilities. However, in this study, the transition model is inherently uncertain and incomplete. Moreover, the true states are inaccessible, rendering historical data unlabeled. Compounding this, measurement noise and underdetermined measurement further limit the available data representational capacity. These constraints pose significant challenges for both standard model-driven Bayesian estimation frameworks and conventional data-driven inference models, as system states must be estimated under uncertain transition models and limited historical data.

To address the aforementioned challenges, we propose AMD, a hybrid-driven state estimation framework that integrates data-driven and model-driven approaches through the use of adaptive cross-coupled priors. Despite the inherent uncertainties in both model priors and historical measurements, AMD effectively consolidates all available information through a complementary fusion mechanism. This enables precise state estimation from noisy and limited dynamic measurements while significantly enhancing both data representation and model robustness.

B. AMD System

The proposed AMD framework enables an adaptive cross-coupling of data-driven and model-driven priors by leveraging a complementary fusion mechanism, thereby integrating the strengths of both paradigms. Specifically, within a two-stage Bayesian inference framework, accurate state estimation is accomplished via two sequential steps: prior predictive fusion and posterior estimate update. In the first step, the prior distribution over the latent state $\mathbf{x}_t$ is inferred based on historical observations $\mathbf{y}_{1:t-1}$, denoted as $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$. In the second step, the posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ is computed by incorporating the newly received observation $\mathbf{y}_t$ via Bayes' rule, thereby completing the state estimation process.

We begin by analyzing the prior predictive fusion step. In traditional state estimation approaches, the prior distribution is obtained through a known state transition model as follows:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$$

However, in the context of our estimation problem, the underlying transition model is unknown. Consequently, the prior distribution can be inferred via a latent state representation $\mathbf{z}_t$, which is accessible through a model-representable abstraction. This leads to the following formulation:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int_{\mathbf{z}_t} p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t) p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1}) \mathrm{d}\mathbf{z}_t \quad (4)$$

where $p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t\right)$ denotes the conditional prior of the current state given the latent representation and historical observations, and $p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})$ captures the distribution over the latent variable based on historical measurements. The latter can be recursively computed from the previous posterior and known model priors as follows:

$$\begin{aligned}
&p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1}) \\
&= \int_{\mathbf{z}_{t-1}} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) p(\mathbf{z}_{t-1} \mid \mathbf{y}_{1:t-1}) \mathrm{d}\mathbf{z}_{t-1} \\
&= \int_{\mathbf{z}_{t-1}} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) \\
&\quad \left(\int_{\mathbf{x}_{t-1}} p(\mathbf{z}_{t-1} \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) \mathrm{d}\mathbf{x}_{t-1} \right) \mathrm{d}\mathbf{z}_{t-1} \\
&= \int_{\mathbf{z}_{t-1}} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) \mathcal{N}\left(\mathbf{z}_{t-1}; \boldsymbol{\mu}^z_{t-1|t-1}, \boldsymbol{\Sigma}^z_{t-1|t-1}\right) \mathrm{d}\mathbf{z}_{t-1} \\
&= \mathcal{N}\left(\mathbf{z}_t; \boldsymbol{\mu}^z_{t|t-1}, \boldsymbol{\Sigma}^z_{t|t-1}\right)
\end{aligned} \quad (5)$$

where $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$ represents the posterior distribution at the previous time step, assumed to be Gaussian: $\mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}\right)$. Given a known mapping and transition process between the state space and latent space, the mean and covariance of the latent state distributions $p(\mathbf{z}_{t-1} \mid \mathbf{y}_{1:t-1})$ and $p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})$ are computed as follows:

$$\begin{aligned}
\boldsymbol{\mu}^z_{t-1|t-1} &= \mathbf{M}_{t-1} \boldsymbol{\mu}_{t-1|t-1} \\
\boldsymbol{\Sigma}^z_{t-1|t-1} &= \mathbf{M}_{t-1} \boldsymbol{\Sigma}_{t-1|t-1} \mathbf{M}^\top_{t-1} + \mathbf{Q}_{t-1} \\
\boldsymbol{\mu}^z_{t|t-1} &= g(\boldsymbol{\mu}^z_{t-1|t-1}) \\
\boldsymbol{\Sigma}^z_{t|t-1} &= \mathbf{G}_t \boldsymbol{\Sigma}^z_{t-1|t-1} \mathbf{G}^\top_t
\end{aligned} \quad (6)$$

It is important to note that the local linearization employed in Eq. 6 is introduced solely for the tractability of covariance propagation, in a manner analogous to the treatment in the Extended Kalman Filter (EKF). Although this approximation may introduce minor errors when representing the nonlinear latent transition, the underlying transition process of the latent state $\mathbf{z}_t$, denoted as $g(\cdot)$, does not require complete accuracy. The inherent incompleteness of the model prior is adaptively compensated by the fusion strategy proposed in

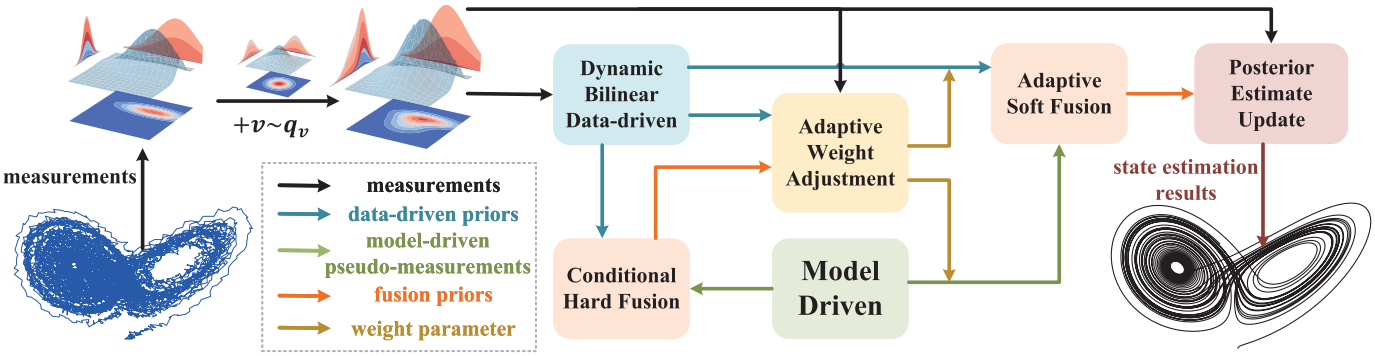


Fig. 1. The block diagram of AMD.

the subsequent framework. The conditional prior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t)$ can be reformulated as:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t) &= \frac{p(\mathbf{x}_t, \mathbf{z}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})\, p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{y}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})\, p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})} \\ &= K \underbrace{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}_{\text{data prior}} \underbrace{p(\mathbf{z}_t \mid \mathbf{x}_t)}_{\text{model prior}} \end{aligned} \tag{7}$$

where K is a normalization constant independent of $\mathbf{x}_t$, and is therefore omitted in subsequent analysis as it does not affect the optimization or modeling of the state distribution. Assuming that the system state $\mathbf{x}_t$ encapsulates all the information relevant to predicting future states and measurements, it renders past observations conditionally independent of the latent state $\mathbf{z}_t$ given $\mathbf{x}_t$. This Markov property enables us to factorize the conditional prior using Bayes' rule as the product of a data-driven prior and a model-driven prior:

$$\begin{aligned} &\text{Data-Driven Prior}: \tilde{p}(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_t^{\text{data}}, \boldsymbol{\Sigma}_t^{\text{data}}\right), \\ &\text{Model-Driven Prior}: p(\mathbf{z}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; \mathbf{M}_t \mathbf{x}_t, \mathbf{Q}_t). \end{aligned} \tag{8}$$

where the notation $\tilde{p}(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$ denotes the purely data-driven prior distribution inferred directly from historical measurements, which will be elaborated in Subsection II-C. The tilde is used to distinguish this candidate prior from the final fused prior $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$, obtained by integrating $\tilde{p}(\cdot)$ with the model-driven prior through the fusion stage. Accordingly, $\tilde{p}(\cdot)$ reflects the initial data-driven perspective, whereas $p(\cdot)$ represents the hybrid prior employed in the Bayesian update. Specifically, in Eq. (7), $p(\cdot)$ represents a generalized Markov factorization, where the subsequent computation of Eq. (7) is carried out using $\tilde{p}(\cdot)$. In the context of model-driven priors, $\mathbf{z}_t$ is derived as a prior prediction from the modeled state transition process g based on historical model-representable latent states $\mathbf{z}_{t-1}$.

Unlike conventional Bayesian filtering, our setting does not allow the prior prediction of the full state to be obtained solely from the transition function. As shown in Eq. (8), we instead construct a data-driven prior as the baseline prediction for the complete state, and then embed the model-representable component $\mathbf{z}_t$ as a constraint derived from the model prior. At each time step $t \in T$, AMD recursively integrates available knowledge through an adaptive cross-coupled process. This process follows a "fusion–adjustment–refusion" paradigm, where the conditional prior is adaptively constructed and fused before proceeding to posterior estimation. This high-level framework is illustrated in Fig. 1.

*a) **Conditional Hard Fusion**:* In this process, we assume that the transition noise can be correctly modeled. The conditional state distribution is computed based on both the data-driven prior and model-driven prior:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t) &= \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_t^{\text{hard}}, \boldsymbol{\Sigma}_t^{\text{hard}}\right) \\ \mathcal{K}_t^{\text{hard}} &= \boldsymbol{\Sigma}_t^{\text{data}} \mathbf{M}_t^\top \left(\mathbf{M}_t \boldsymbol{\Sigma}_t^{\text{data}} \mathbf{M}_t^\top + \mathbf{Q}_t\right)^{-1} \\ \boldsymbol{\mu}_t^{\text{hard}} &= \boldsymbol{\mu}_t^{\text{data}} + \mathcal{K}_t^{\text{hard}} (\mathbf{z}_t - \mathbf{M}_t \boldsymbol{\mu}_t^{\text{data}}) \\ \boldsymbol{\Sigma}_t^{\text{hard}} &= \boldsymbol{\Sigma}_t^{\text{data}} - \mathcal{K}_t^{\text{hard}} \mathbf{M}_t \boldsymbol{\Sigma}_t^{\text{data}} \end{aligned} \tag{9}$$

Using the Woodbury matrix identity [28], we obtain a form similar to that of the Kalman filter, which demonstrates that the model prior can be treated as a pseudo-observation, thereby constraining the parameter space of the data-driven prior. During the fusion process, the projection matrix $\mathbf{M}_t$ explicitly maps the model subspace onto the full state $\mathbf{x}_t$, thereby ensuring dimensional alignment and enabling a consistent integration across dimensions.

*b) **Adaptive Soft Fusion**:* Since the modeling of the state transition process may be incomplete or based on simplified assumptions that overlook certain system complexities, estimation errors in $\mathbf{z}_t$ can result from such model inaccuracies or simplifications. Furthermore, biases or misspecification in the state transition noise $\mathbf{Q}_t$ may introduce substantial uncertainty into the prior fusion process. As a consequence, the formulation in Eq. (9) cannot be directly applied in its original form.

To address the above issue, the integration of model-driven and data-driven priors should be performed through a soft fusion strategy, whereby each source of prior knowledge contributes adaptively according to its assessed reliability. In this process, $\mathbf{Q}_t$ is not explicitly estimated; instead, an exponential penalty is applied to govern the weighted fusion. The adaptive adjustment of the penalty (weighting) parameter α_t provides flexible control over the respective contributions of the model-

driven and data-driven priors. Specifically, independent information extracted from noise measurements is used to evaluate the performance discrepancy between the hard fusion prior and the data-driven prior, thereby enabling the adaptive inference of an appropriate weighting parameter. Further details of this procedure are provided in Subsection II-D.

In the Bayesian framework, the soft fusion process exponentially weights the model prior:

$$\begin{aligned} &p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t\right) \\ &\quad \approx K \tilde{p}\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}\right) \exp\left(-\frac{\alpha_t}{2}\|\mathbf{M}_t \mathbf{x}_t - \mathbf{z}_t\|_2^2\right) \end{aligned} \tag{10}$$

To incorporate model-based priors under model uncertainties, the original likelihood is reformulated as an exponentially weighted penalty term, serving as a "pseudo-likelihood" that encodes the preference for imposing prior constraints on the data. The adoption of an exponential form not only naturally preserves the Gaussian property of the fused prior but also ensures theoretical consistency of the update when treated as a pseudo-likelihood. In addition, this design provides enhanced interpretability as well as computational stability and scalability in high-dimensional settings. Furthermore, by applying the technique of completing the square [5], a closed-form representation of the fused prior distribution can be derived.

$$\begin{aligned} p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t\right) &= \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_t^{\text{soft}}, \boldsymbol{\Sigma}_t^{\text{soft}}\right) \\ \boldsymbol{\mu}_t^{\text{soft}} &= \boldsymbol{\Sigma}_t^{\text{soft}}\left(\left(\boldsymbol{\Sigma}_t^{\text{data}}\right)^{-1} \boldsymbol{\mu}_t^{\text{data}} + \alpha_t \mathbf{M}_t^{\top} \mathbf{z}_t\right) \\ \boldsymbol{\Sigma}_t^{\text{soft}} &= \left(\left(\boldsymbol{\Sigma}_t^{\text{data}}\right)^{-1} + \alpha_t \mathbf{M}_t \mathbf{M}_t^{\top}\right)^{-1} \end{aligned} \tag{11}$$

The above process adaptively weights the regularization of the model prior's constraint by adjusting the re-fusion strategy, thus mitigating the impact of estimation errors in $\mathbf{z}_t$ and model biases in $\mathbf{Q}_t$. Specifically, a scaled identity matrix term $\alpha_t \mathbf{I}_r$ is introduced to serve a role analogous to $\mathbf{Q}_t$, providing a controlled form of regularization. This adaptive adjustment reduces the uncertainty that would otherwise arise during prior fusion due to model misalignment or approximation errors. Accordingly, the covariance term $\mathbf{Q}_{t-1}$ in Eq. (6) should be replaced by the adaptive regularization term $\alpha_{t-1}\mathbf{I}_r$, reflecting the calibrated treatment of model uncertainty at the preceding time step.

It is important to note that the order of the fusion process is not interchangeable. Conditional hard fusion must be performed first to establish a principled Bayesian baseline, which is then refined through adaptive soft fusion. The adaptive weighting depends on the discrepancy between the data-driven prior distribution and the hard-fused distribution. Reversing this order would eliminate this reference, rendering the adaptive procedure ill-posed and potentially unstable.

*c) **Posterior Estimate Update**:* In this stage, the system updates the posterior distribution of the current state $p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t}\right)$ by incorporating the fused prior $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}, \mathbf{z}_t)$, obtained from the integration of data-driven and model-driven priors, and the real-time observation $\mathbf{y}_t$, and available domain knowledge. The fusion prior is first computed according to Eq. (4) as follows:

$$\begin{aligned} &p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) \\ &= \int_{\mathbf{z}_t} \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_t^{\text{soft}}, \boldsymbol{\Sigma}_t^{\text{soft}}\right) \mathcal{N}\left(\mathbf{z}_t; \boldsymbol{\mu}_{t|t-1}^{z}, \boldsymbol{\Sigma}_{t|t-1}^{z}\right) \mathrm{d}\mathbf{z}_t \\ &= \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}\right) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1} &= \boldsymbol{\Sigma}_t^{\text{soft}}\left(\left(\boldsymbol{\Sigma}_t^{\text{data}}\right)^{-1} \boldsymbol{\mu}_t^{\text{data}} + \alpha_t \mathbf{M}_t^{\top} \boldsymbol{\mu}_{t|t-1}^{z}\right) \\ \boldsymbol{\Sigma}_{t|t-1} &= \boldsymbol{\Sigma}_t^{\text{soft}} + \boldsymbol{\Sigma}_t^{\text{soft}} \mathbf{M}_t^{\top} \boldsymbol{\Sigma}_{t|t-1}^{z} \mathbf{M}_t \boldsymbol{\Sigma}_t^{\text{soft}\top} \end{aligned} \tag{12}$$

Finally, the posterior distribution of the current state, $p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t}\right)$, is updated by integrating the prior distribution with the real-time observation. This update is expressed as:

$$\begin{aligned} p\left(\mathbf{x}_t \mid \mathbf{y}_{1:t}\right) &= \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}\right) \\ \mathcal{K}_t &= \boldsymbol{\Sigma}_{t|t-1} \mathbf{H}_t^{\top}\left(\mathbf{H}_t \boldsymbol{\Sigma}_{t|t-1} \mathbf{H}_t^{\top} + \mathbf{R}_t\right)^{-1} \\ \boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathcal{K}_t(\mathbf{y}_t - \mathbf{H}_t \boldsymbol{\mu}_{t|t-1}) \\ \boldsymbol{\Sigma}_{t|t} &= \boldsymbol{\Sigma}_{t|t-1} - \mathcal{K}_t \mathbf{H}_t \boldsymbol{\Sigma}_{t|t-1} \end{aligned} \tag{13}$$

where $\boldsymbol{\mu}_{t|t}$ and $\boldsymbol{\Sigma}_{t|t}$ represent the posterior mean and covariance matrix, respectively, which reflect the updated state estimation.

C. Dynamic Bilinear Data-driven Modeling

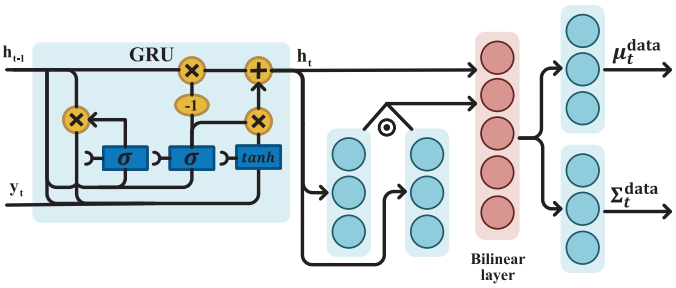


Fig. 2. Dynamic bilinear data-driven block diagram.

Inspired by the Bayesian inference framework, AMD aims to leverage the generalization capability of data-driven methods to capture complex transition dynamics within the dynamic state-space system. To achieve this, AMD models the data-driven prior $\tilde{p}\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}\right)$ as a fully informed Gaussian distribution based on historical observations, as previously defined in Eq. (8). For the neural network architecture, we take into account the temporal characteristics of data and employ recurrent neural networks (RNNs) [40] and their variants to address nonlinear dynamical challenges in the state-space representation. Specifically, in this study, to balance the complexity of data-driven modeling with the requirements of stability and efficiency, we adopt the Gated Recurrent Unit (GRU) architecture [15] as the core component of the data-driven module. This choice ensures high scalability and robust resistance to overfitting.

It should be pointed out that conventional RNN-based approaches primarily capture system nonlinearity through nonlinear activation functions [31]. The direct combination of

nonlinear activations with linear transformations often fails to efficiently approximate highly nonlinear state-space models, such as the Lorenz dynamic system. More specifically, physical dynamical systems commonly exhibit bilinear nonlinearity, which characterizes the multiplicative interaction between two physical variables [16], [34], [39]. This characteristic is structurally similar to polynomial decomposition and aligns with the classical model-driven approach of approximating nonlinearities via Taylor series expansion. However, conventional activation function-based methods tend to introduce overly complex approximations [57], making it difficult to provide physically interpretable dynamic representations for such systems.

To address these challenges, AMD integrates a bilinear neural network architecture, in which two parallel linear layers learn distinct components of the variables, followed by element-wise multiplication to capture their multiplicative interactions, and a final linear mapping to produce the bilinear representation. This bilinear architecture enhances interpretability and computational efficiency in dynamic operators, allowing it to effectively capture complex dynamics using only dominant linear activation functions. The framework of the dynamic bilinear data-driven module is illustrated in Fig. 2 and Eq. (14), the DBDD module utilizes a GRU to predict the current hidden state $\mathbf{h}_t$ based on historical measurement data, effectively capturing the intrinsic dynamic transition process of the system. Building upon $\mathbf{h}_t$, a bilinear neural network is employed to integrate fully connected layers with state-product operators, explicitly constructing a second-order polynomial representation. This representation enables the system to approximate complex intrinsic nonlinear transition dynamics more effectively.

$$\begin{aligned} \mathbf{h}_t &= \mathrm{GRU}(\mathbf{y}_t, \mathbf{h}_{t-1}), \quad \boldsymbol{\varphi}_t = \mathrm{FC}_1\left(\mathrm{FC}_2(\mathbf{h}_t) \odot \mathrm{FC}_3(\mathbf{h}_t), \mathbf{h}_t\right) \\ \boldsymbol{\mu}_t^{\text{data}} &= \mathrm{FC}_\mu(\boldsymbol{\varphi}_t), \quad \boldsymbol{\Sigma}_t^{\text{data}} = \mathrm{FC}_\Sigma(\boldsymbol{\varphi}_t) \end{aligned} \tag{14}$$

where $\mathrm{FC}(\cdot)$ denotes a fully connected neural network layer, the operator $\odot$ denotes the element-wise multiplication between vectors, and φ denotes the intermediate hidden state.

To facilitate practical implementation, we model the covariance matrix in a constrained diagonal form, ensuring both numerical stability and ease of computation:

$$\boldsymbol{\Sigma}_t^{\text{data}} = \mathrm{FC}_\Sigma(\boldsymbol{\varphi}_t) = \mathrm{diag}\left(\boldsymbol{\sigma}_0^2 \exp\left(\beta \tanh \boldsymbol{\varphi}_t\right)\right) \tag{15}$$

where $\boldsymbol{\sigma}_0$ denotes the initial covariance estimate, and $\beta \in \mathbb{R}_{>0}$ is a scaling coefficient. The operator $\mathrm{diag}(\cdot)$ transforms a vector into a diagonal matrix of the corresponding size, while $\tanh(\cdot)$ represents the hyperbolic tangent function. Through this formulation, the neural network constrains the covariance values within the range $\boldsymbol{\sigma}_0^2 e^{-\beta}$ to $\boldsymbol{\sigma}_0^2 e^{\beta}$, thereby preventing numerical issues such as divergence caused by excessively large variances or gradient vanishing due to overly small variances, ultimately stabilizing the learning process. Moreover, the bounded interval admits a clear physical interpretation. The exponential-tanh structure ensures that the data-driven module converges in a stable and physically meaningful manner, even when the response is negligible during the initial training phase or when state estimates are not yet available.

Now, to prevent the DBDD module from overfitting to noisy measurements and to account for the inherent uncertainty in the measurement process, we introduce observation perturbations [36] during training, as illustrated in Fig. 1. Specifically, Gaussian noise $v \sim q_v$ is added to the original measurements to generate perturbed observations. This strategy improves model robustness and enhances generalization, enabling the model to better capture measurement uncertainty and to learn more representative prior distributions, thereby strengthening the data representation capability under limited measurements. This mechanism is introduced only during training. During inference, this perturbation is disabled, and the model directly processes the true observations to ensure consistency with the actual measurement data.

D. Adaptive Weight Adjustment Strategy

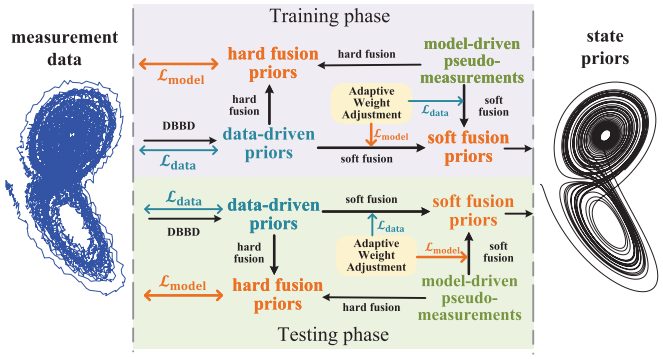


Fig. 3. Adaptive Weight Adjustment strategy block diagram.

This subsection introduces an adaptive method for computing appropriate weighting parameters α_t to balance the contributions of data-driven and model-driven components in the hybrid framework. Since the true system states are generally unobservable during the estimation process, the system cannot be explicitly corrected through direct supervision based on state discrepancies. Instead, it can only infer internal states indirectly from noisy measurements. Furthermore, incomplete or inaccurate model-driven representations pose significant challenges for adaptive weighting, as an incomplete state cannot be fully assessed through measurements alone.

To address the above identified challenges, we replace model prior with initial fusion prior and evaluate it using independent information provided by $\mathbf{y}_t$. Specifically, the weighting parameter is iteratively updated by comparing the reconstruction errors of the model prior and data prior against the measurements:

$$Loss_{\text{model}} = \|\mathbf{H}_t \boldsymbol{\mu}_t^{\text{hard}} - \mathbf{y}_t\|_2^2, Loss_{\text{data}} = \|\mathbf{H}_t \boldsymbol{\mu}_t^{\text{data}} - \mathbf{y}_t\|_2^2$$

where $Loss_{\text{model}}$ and $Loss_{\text{data}}$ represent the reconstruction errors of model prior and data prior, respectively.

It should be noted that a larger error indicates a poorer ability to explain the measurement, which indirectly reflects a greater deviation from the true state. Therefore, based on the relative performance of the priors in reconstructing

measurements, α_t is adaptively updated. However, since the objectives and underlying rationale for weighting adjustments differ between the training and testing phases, distinct yet complementary update strategies are employed, as illustrated in Fig. 3.

- Training Phase: During training, the estimator aims to encourage the data-driven module to internalize as much model knowledge as possible. When $Loss_{\text{model}}$ is large, it indicates that model-driven prior performs poorly in reconstruction. However, rather than reducing trust in the model prior, we further increase its weight to encourage learning in areas where the current representation is insufficient.
- Testing Phase: The objective in testing is to estimate the system's true state as accurately as possible, without necessarily adhering to the system behavior. When $Loss_{\text{model}}$ is large, it implies that model-driven prior has poor representation capability. Consequently, the model-driven weight should be reduced to mitigate the impact of erroneous model priors and improve estimation accuracy.

This complementary design ensures that AMD acquires strong physical inductive biases during training, while preserving robustness and flexibility during testing. To implement this adaptive strategy, we employ a logistic function-based update mechanism:

$$\begin{aligned}\text{Training}: \alpha_t &= \alpha_{t-1}\exp\left(\gamma\cdot\phi\left(\frac{Loss_{\text{model}}+\epsilon}{Loss_{\text{data}}+\epsilon}\right)\right)\\ \text{Testing}: \alpha_t &= \alpha_{t-1}\exp\left(\gamma\cdot\phi\left(\frac{Loss_{\text{data}}+\epsilon}{Loss_{\text{model}}+\epsilon}\right)\right)\\ \phi(r) &= \frac{2}{1+e^{-(r-1)/\delta}}-1\end{aligned} \tag{16}$$

where ϵ is a small positive constant to prevent division by zero, ensuring numerical stability. The logistic function $\phi(\cdot)$ structure smooths the variations in α, preventing abrupt changes and guaranteeing saturation under extreme error rates, which preserves flexibility. The parameter γ controls the scaling rate, while δ determines the sensitivity of ϕ to error differences. By tuning these two parameters, the sensitivity of α updates can be flexibly adapted to different scenarios, thereby enhancing both stability and adaptability.

Next, to prevent excessively large or small weight parameters from causing numerical instability in the prior fusion process, we apply a clamping operation to constrain α_t within a predefined range $[\alpha_{\min}, \alpha_{\max}]$. To maintain consistency in the fusion strategy, when α_t reaches its minimum or maximum threshold, the adaptive soft fusion transitions to either no fusion (using only the data prior) or conditional hard fusion. This aligns with the intended role of the weighting parameter α_t.

E. Unsupervised Hybrid Learning Framework

In this subsection, we introduce how AMD adaptively performs hybrid-driven learning and training in the absence of ground-truth state annotations. As an unsupervised state estimation task, our approach is inspired by the optimization principles of the Variational Autoencoder (VAE) [23], which seeks to maximize the likelihood of the measurement distribution $p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$. This is equivalent to minimizing the negative log marginal likelihood, i.e., $-\log p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$, which can be reformulated as:

$$\begin{aligned}&-\log p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})\\ &= -\log \iint p(\mathbf{y}_t, \mathbf{x}_t, \mathbf{z}_t \mid \mathbf{y}_{1:t-1})\mathrm{d}\mathbf{z}_t\mathrm{d}\mathbf{x}_t\\ &= -\log \iint p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{z}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})\mathrm{d}\mathbf{z}_t\mathrm{d}\mathbf{x}_t\\ &= -\log \iint p(\mathbf{x}_t \mid \mathbf{y}_{1:t})p(\mathbf{z}_t \mid \mathbf{y}_{1:t})\\ &\quad\times\frac{p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{z}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t})p(\mathbf{z}_t \mid \mathbf{y}_{1:t})}\mathrm{d}\mathbf{z}_t\mathrm{d}\mathbf{x}_t\\ &\le \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{z}_t|\mathbf{y}_{1:t})}\Big[-\log p(\mathbf{y}_t \mid \mathbf{x}_t) - \log p(\mathbf{z}_t \mid \mathbf{x}_t)\\ &\quad+\log\frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}+\log p(\mathbf{z}_t \mid \mathbf{y}_{1:t})\Big]\\ &= \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{z}_t|\mathbf{y}_{1:t})}\Big[-\log p(\mathbf{y}_t \mid \mathbf{x}_t) - \log p(\mathbf{z}_t \mid \mathbf{x}_t)\\ &\quad+\log\frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}+\log\frac{p(\mathbf{z}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})}\\ &\quad+\log p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})\Big]\end{aligned} \tag{17}$$

where the third line above is derived under the conditional independence assumption of observations. From the third to the fourth line, we apply Jensen's inequality [44] to transform the intractable integral into an upper-bound expectation decomposition. The final term of last line, $\log p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1})$, is a model-derived prior based solely on historical states in Eq. (5). This term does not influence the location of optima or the gradient trajectory and can thus be regarded as a constant with respect to learnable parameters. Therefore, it is omitted in the following analysis.

- $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}\left[-\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right]$ corresponds to the negative log-likelihood of current measurement $\mathbf{y}_t$, serving as the reconstruction loss. This term guides the data-driven component to indirectly learn the system's internal dynamic state evolution from independent measurements.
- $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{z}_t|\mathbf{y}_{1:t})}\left[-\log p(\mathbf{z}_t \mid \mathbf{x}_t)\right]$ represents the reconstruction loss for the model-representable latent state $\mathbf{z}_t$, which penalizes deviations between the learned latent dynamics and known model-based transition knowledge. It encourages the data-driven process to conform to established model priors.
- $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}\log\frac{p(\mathbf{x}_t|\mathbf{y}_{1:t})}{p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}$ can be rewritten as a Kullback-Leibler (KL) divergence term between the posterior and the prior of the state $\mathbf{x}_t$, which constrains the learned posterior to remain close to the one-step predictive prior, thereby mitigating covariance inflation and stabilizing the estimation process.
- $\mathbb{E}_{p(\mathbf{z}_t|\mathbf{y}_{1:t})}\log\frac{p(\mathbf{z}_t|\mathbf{y}_{1:t})}{p(\mathbf{z}_t|\mathbf{y}_{1:t-1})}$ similarly quantifies the discrepancy between the current inferred latent state and its model-based prediction from the previous timestep. This term adjusts for inconsistencies between model-driven and data-driven priors at the latent representation level.

Further interpreting the perspective of the Information Bottleneck principle [27], we formulate the hybrid-driven loss

function as:

$$\mathcal{L} = \mathcal{L}_y + \lambda_z \mathcal{L}_z + \beta_x \mathrm{KL}_x + \beta_z \mathrm{KL}_z \tag{18}$$

where λ_z governs the strength of the model-based physical constraint on the latent state $\mathbf{z}_t$. When the physical model is poorly specified but measurement information is reliable, λ_z is reduced to avoid overfitting to potentially inaccurate model knowledge. Conversely, in the presence of a well-defined model prior, a larger λ_z enforces a stronger bottleneck, guiding the learning process toward compact and structured latent representations, thereby improving convergence and stability. The weights β_x and β_z control the contributions of the KL divergence terms for the estimated states and latent states, respectively. These can be scheduled to increase linearly over training, following a β-VAE-inspired annealing strategy.

The first two terms in (18) aim to jointly minimize both measurement error and model residual, striking a dynamic balance between data-driven and model-driven trustworthiness. The latter two terms thus form a dual information bottleneck, designed not only to stabilize the state estimation process but also to suppress variance inflation in the latent state space, helping preserve long-term dynamics and structural memory.

The explicit formulations of each loss component are as follows:

$$\begin{aligned}
\mathcal{L}_y &= \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}\left[-\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] \\
&= -\frac{1}{2}\Big[\|\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t}\|^2_{\mathbf{R}_t^{-1}} + \mathrm{tr}(\mathbf{R}_t^{-1}\mathbf{H}_t\boldsymbol{\Sigma}_{t|t}\mathbf{H}_t^\top) \\
&\quad + n\log 2\pi + \log\det\mathbf{R}_t\Big] \\
\mathcal{L}_z &= \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{z}_t|\mathbf{y}_{1:t})}\left[-\log p(\mathbf{z}_t \mid \mathbf{x}_t)\right] \\
&= -\frac{1}{2}\Big[\mathrm{tr}(\alpha_t^{-1}\boldsymbol{\Sigma}^z_{t|t}) + \mathrm{tr}(\alpha_t^{-1}\mathbf{M}_t\boldsymbol{\Sigma}^z_{t|t}\mathbf{M}_t^\top) \\
&\quad + r\log 2\pi + \log r\alpha_t\Big] \\
\mathrm{KL}_x &= \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}\log\frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})} \\
&= \frac{1}{2}\left[\|\boldsymbol{\mu}_{t|t} - \boldsymbol{\mu}_{t|t-1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t|t-1}} - \log\frac{\det\boldsymbol{\Sigma}_{t|t-1}}{\det\boldsymbol{\Sigma}_{t|t}}\right. \\
&\quad \left. - m + \mathrm{tr}(\boldsymbol{\Sigma}^{-1}_{t|t-1}\boldsymbol{\Sigma}_{t|t})\right] \\
\mathrm{KL}_z &= \mathbb{E}_{p(\mathbf{z}_t|\mathbf{y}_{1:t})}\log\frac{p(\mathbf{z}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})} \\
&= \frac{1}{2}\left[\|\boldsymbol{\mu}^z_{t|t} - \boldsymbol{\mu}^z_{t|t-1}\|^2_{\boldsymbol{\Sigma}^z_{t|t-1}{}^{-1}} - \log\frac{\det\boldsymbol{\Sigma}^z_{t|t-1}}{\det\boldsymbol{\Sigma}^z_{t|t}}\right. \\
&\quad \left. - r + \mathrm{tr}(\boldsymbol{\Sigma}^z_{t|t-1}{}^{-1}\boldsymbol{\Sigma}^z_{t|t})\right]
\end{aligned} \tag{19}$$

where m, n and r denote the dimensions of the state, measurement, and state model-representable latent state, respectively. The term $\mathrm{tr}(\cdot)$ represents the trace of a matrix.

F. Discussion

A central challenge in traditional state estimation methods lies in the limitations imposed by limited data on data-driven approaches and the interference caused by model uncertainties in model-driven techniques. Conventional hybrid-driven methods fail to fundamentally address these issues. The proposed AMD framework tackles these challenges through a 'conditional hard fusion + adjustment + adaptive soft fusion' adaptive cross-coupled fusion strategy, enabling a complementary integration of data priors and model priors.

AMD leverages known domain model priors to constrain data representations while simultaneously exploiting the generalization capabilities of data-driven methods to compensate for model uncertainties. This adaptive and complementary interaction allows the framework to overcome the inherent limitations of each paradigm. Moreover, AMD introduces an adaptive weight adjustment mechanism that dynamically balances model constraints and estimation performance across the training and testing phases, further enhancing data representation capability and model robustness.

To further address the nonlinear complexities inherent in real-world systems, AMD incorporates a dynamic bilinear-driven module, which enhances both the expressiveness and robustness of the data-driven representation. Additionally, to handle scenarios where ground-truth states are inaccessible, AMD employs an unsupervised hybrid learning strategy that ensures both learning efficiency and model interpretability.

Overall, AMD achieves a cross-coupled fusion of model-driven and data-driven paradigms by adaptively leveraging their respective strengths in an unsupervised manner. Through the integration of multiple strategies, AMD effectively addresses the complex challenges faced by conventional hybrid estimation methods, while robustly and efficiently enhancing both data representation capability and model robustness. The training procedure is detailed in Algorithm 1.

Algorithm 1: Training the Adaptive Model-driven and Data-driven Cross-coupled Estimator (AMD)

while *not converged* **do**
  for *all trajectories* $\mathbf{y}_{1:T}$ **do**
    Sample Gaussian noise $v \sim q_v$ to generate observation perturbations;
    for $t = 1 : T$ **do**
      `// Compute Data Prior`
      Compute $\boldsymbol{\mu}_t^{\text{data}}$ and $\boldsymbol{\Sigma}_t^{\text{data}}$ by Eq. (14) ;
      `// Compute Model Prior`
      Compute $\mathbf{z}_t$ by Eq. (3) ;
      `// Conditional Hard Fusion`
      Compute $\boldsymbol{\mu}_t^{\text{hard}}$ and $\boldsymbol{\Sigma}_t^{\text{hard}}$ by Eq. (9) ;
      `// Adaptive Weight Adjustment`
      Compute α_t by Eq. (16) ;
      `// Adaptive Soft Fusion`
      Compute $\boldsymbol{\mu}_t^{\text{soft}}$ and $\boldsymbol{\Sigma}_t^{\text{soft}}$ by Eq. (11) ;
      `// Posterior Estimate Update`
      Compute $\boldsymbol{\mu}_{t|t}$ and $\boldsymbol{\Sigma}_{t|t}$ by Eq. (13) ;
    end
  end
  Update trainable parameters θ by Eq. (18) ;
end

Remark 1. In comparison with existing literature, this paper exhibits the following distinctive novelties.

1) *Development of an Adaptive Cross-Coupled Hybrid Framework*: A novel hybrid-driven state estimation

framework, termed AMD, has been proposed, which adaptively integrates model-based and data-driven priors through a two-stage process: an initial hard fusion followed by a data-aware soft fusion. This structure addresses the limitations of fixed-weight or heuristic fusion seen in existing hybrid estimators such as KalmanNet and DANSE.

2) *Formulation of an Unsupervised Variational Learning Objective*: Unlike most data-driven and hybrid methods that rely on supervised learning or access to ground-truth states, AMD adopts an unsupervised training approach inspired by the information bottleneck principle. This enables effective parameter learning under unknown states and noisy observations, broadening applicability in practical scenarios.
3) *Design of a Dynamic Bilinear Recurrent Module*: To capture complex nonlinear transition dynamics more effectively than standard RNNs or GRUs, a dynamic bilinear recurrent structure has been designed. This component improves modeling fidelity by incorporating bilinear interactions reflective of system physics, enhancing the data-driven prior construction.
4) *Implementation of an Adaptive Weight Update Mechanism*: A distinct training-testing weight adaptation mechanism has been introduced, allowing AMD to dynamically adjust the contribution of model and data priors based on observed reconstruction errors. This improves robustness under model uncertainty and measurement noise, under which traditional fusion strategies often fail.
5) *Principled Bayesian Fusion with Model Uncertainty Handling*: A reformulation of model priors as pseudo-likelihoods using exponential penalties has been proposed, allowing AMD to perform soft prior fusion without requiring direct estimation of transition noise covariance. This principled Bayesian integration enhances flexibility and theoretical interpretability.

III. Experiments and Results

This section presents an extensive set of numerical experiments conducted to comprehensively evaluate the proposed AMD framework. To rigorously assess its capability in addressing complex dynamics under limited prior knowledge, we adopt chaotic dynamical systems as challenging benchmark nonlinear systems. The performance of AMD is evaluated under various experimental conditions and compared against several established state estimation approaches, including:

- Model-based EKF and UKF
- Unsupervised hybrid-driven KalmanNet and Danse.

Model-driven approaches generally require full knowledge of the underlying state-space transition and measurement models. EKF [6], [24] approximates nonlinear dynamics via first-order Taylor expansion, while UKF [35] employs an unscented transformation to construct sigma points for nonlinear moment estimation. Both approaches are highly dependent on accurate noise parameter estimation and represent the most widely used extensions of the Kalman filter for nonlinear state estimation. These methods serve as strong baselines for evaluating AMD.

For fair comparison, we adopt unsupervised versions of hybrid methods, excluding purely data-driven models which are less applicable when partial physical knowledge is available. KalmanNet assumes access to a linearized model similar to EKF but does not require prior knowledge of noise characteristics. Danse assumes linear measurement systems with Gaussian noise without requiring knowledge of transition models.

Inspired by experimental setups in [22], [48], we use synthetic datasets generated by numerically integrating classical chaotic systems. This design ensures access to ground-truth state-space models, enabling controlled comparison with traditional model-based methods and robustness evaluation under model perturbations. Moreover, access to true states facilitates quantifiable benchmarking of estimation accuracy.

Experiments are conducted on representative nonlinear chaotic systems, including the Lorenz attractor [37], Chen attractor [10], and Rössler attractor [9]. To enhance realism, we introduce moderate process noise into the dynamical systems, rendering them uncertain while preserving their stochastic statistical properties.

Training is performed on an unlabeled dataset $\mathcal{D}_{\text{train}} := \left\{ \mathbf{y}^{(i)}_{1:T^{(i)}_{\text{train}}} \right\}_{i=1}^{N_{\text{train}}}$, while evaluation is carried out on a labeled test set $\overline{\mathcal{D}}_{\text{test}} := \left\{ \mathbf{x}^{(i)}_{1:T^{(i)}_{\text{test}}}, \mathbf{y}^{(i)}_{1:T^{(i)}_{\text{test}}} \right\}_{i=1}^{N_{\text{test}}}$.

To assess the accuracy, robustness, and generalization capabilities of each method, we report performance using Mean Squared Error (MSE) in decibel (dB) scale and Negative Log-Likelihood (NLL), along with their respective standard deviations. The evaluation metrics are defined as follows:

$$\begin{aligned} \text{MSE} &= \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} 10 \log_{10} \sum_{t=0}^{T^{(i)}} \|\mathbf{x}_t^{(i)} - \boldsymbol{\mu}_t\|_2^2 \\ \text{NLL} &= \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{1}{T^{(i)}_{\text{test}}} \sum_{j=1}^{T^{(i)}_{\text{test}}} -\log p(\mathbf{x}_j^{(i)} \mid \mathbf{y}_{1:j}^{(i)}) \end{aligned} \tag{20}$$

In the results tables, the best values for each metric are highlighted in bold, and the second-best results are italicized.

A. Experimental Setting

The AMD framework is trained using mini-batch gradient descent with the Adam optimizer. A learning rate scheduler and early stopping strategy are employed to prevent overfitting and enhance generalization performance. For the DBDD component, AMD adopts a two-layer GRU model with 40 hidden units per layer. The final fully connected layer (FC) is implemented as a single layer with 24 hidden units. The covariance scaling factor β is set to 3. Through experimental evaluation, we inject perturbations during training with a magnitude equivalent to 10% of the observation noise amplitude, in order to simulate perturbed measurement noise. This perturbation level effectively enhances robustness while ensuring stable training outcomes.

In the experiments, we set $\alpha_{\min} = 10^{-6}$ and $\alpha_{\max} = 10^{6}$ as empirical bounds. This range is sufficiently broad to cover most estimation scenarios while preventing the fusion

TABLE II
Lorenz attractor with fully measurements.

	MSE (dB)	NLL ($\times 10^3$)	Time ($\times 10^{-1}$ s)
EKF	*3.042 ± 0.352*	8.987 ± 8.792	1.257
UKF	**3.041 ± 0.352**	8.452 ± 8.815	2.759
Danse	3.105 ± 0.357	**0.070 ± 0.009**	**0.005**
KalmanNet	21.212 ± 1.206	0.137 ± 0.032	0.869
AMD no	3.061 ± 0.351	*0.564 ± 0.056*	*0.007*
AMD	3.052 ± 0.347	0.814 ± 0.078	0.019

mechanism from diverging or collapsing, thereby suppressing numerical instabilities. In practice, these thresholds can be adjusted according to the specific task, the reliability of the model priors, and the quality of the available data.

Unless otherwise specified, AMD employs an incomplete transition process model, with a known component ratio of $r/m = 2/3$. The process mapping matrix is fixed as

$$\mathbf{M}_t = \mathbf{M} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

For baseline comparisons, we adopt the official implementations of KalmanNet [49] and DANSE [21] as provided in the literature.

B. Lorenz Attractor

In this experiment, we evaluate the performance of the proposed AMD framework on a nonlinear chaotic dynamical system, namely, the Lorenz attractor, to demonstrate its competitiveness across multiple dimensions. The Lorenz attractor [37] is a classical benchmark known for its strong chaos and high nonlinearity, posing significant challenges for conventional state estimation methods. The discrete-time formulation of the Lorenz system is given by:

$$\mathbf{x}_t = f_{\text{lorenz}}(\mathbf{x}_{t-1}) + \mathbf{w}_t \in \mathbb{R}^3 \tag{21}$$

where the nonlinear transition function is defined as:

$$f_{\text{lorenz}}(\mathbf{x}_t) = \exp\left(\begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -\mathbf{x}_{t,1} \\ 0 & \mathbf{x}_{t,1} & -8/3 \end{bmatrix} \Delta t\right) \mathbf{x}_t \tag{22}$$

with a time step of $\Delta t = 0.02$ seconds. The process noise $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$ and the measurement noise $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$ are modeled as independent Gaussian white noise processes, where $\mathbf{Q} = q^2\mathbf{I}_m$ and $\mathbf{R} = r^2\mathbf{I}_n$.

*a) **Fully Measurements**:* We first evaluate the AMD framework under a fully measurement setting to highlight its superiority in capturing latent system dynamics, as well as its ability to integrate model- and data-driven estimation approaches. In this configuration, the measurement-to-state dimension ratio is $n/m = 3/3$, indicating that the measurements provide a complete representation of the state. The measurement matrix is defined as:

$$\mathbf{H}_t = \mathbf{H} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

The process and measurement noise covariances are set to $q^2 = 0.01$ and $r^2 = 0.1$, respectively. Performance comparisons are summarized in Table II. To further investigate the estimation capability of the data-driven component alone, we introduce a variant of AMD with no knowledge of the state transition model (i.e., the transition function is entirely unknown). Results show that even under this unsupervised setting, AMD demonstrates competitive real-time estimation performance, outperforming other hybrid methods. This suggests that the proposed DBDD module effectively captures nonlinear transition behavior.

When partial knowledge of the transition process is available, a significant improvement in AMD's performance is observed, approaching that of fully model-driven methods. This demonstrates that available model priors are effectively leveraged by AMD and validates the advantage of the proposed adaptive cross-coupled fusion strategy.

TABLE III
Lorenz attractor with underdetermined measurements.

	MSE (dB)	NLL ($\times 10^3$)	Time ($\times 10^{-1}$ s)
EKF	9.484 ± 3.939	10.306 ± 15.363	1.250
UKF	**3.168 ± 0.388**	9.577 ± 15.272	2.626
Danse	22.977 ± 0.295	96.414 ± 86.820	**0.005**
KalmanNet	21.858 ± 0.767	**0.155 ± 0.024**	0.866
AMD full	*8.141 ± 3.821*	*1.560 ± 1.505*	0.019
AMD	12.320 ± 3.750	3.010 ± 2.763	*0.018*

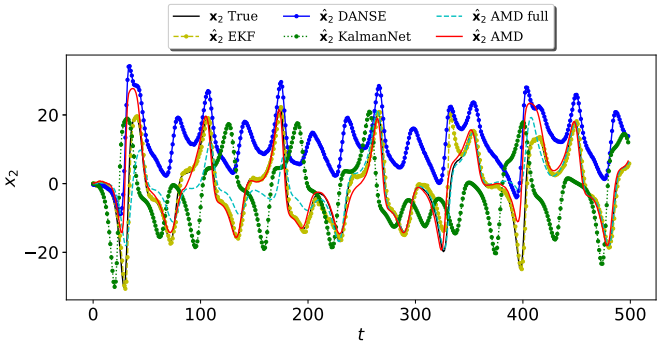


Fig. 4. Visualization of Lorenz attractor with underdetermined measurements.

*b) **Underdetermined Measurements**:* To further evaluate the robustness of AMD under data scarcity and its ability to complement limited model priors, we conduct experiments in a underdetermined measurement setting, following the setup described in [21]. The measurement matrix is given by:

$$\mathbf{H}_t = \mathbf{H} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{24}$$

In this setting, measurements do not provide a full representation of the system state as multiple latent states can produce identical observations. This inherent ambiguity poses significant challenges for both data-driven and traditional model-based estimators. The process and observation noise covariances are set to $\mathbf{Q} = 0.01\mathbf{I}_3$ and $\mathbf{R} = 0.01\mathbf{I}_2$, respectively.

As shown in Table III, underdetermined measurements substantially degrade the performance of existing hybrid estimation methods. In contrast, AMD maintains competitive estimation accuracy by effectively leveraging its adaptive cross-coupled fusion strategy, even when both data-driven and model-driven components operate under limited information.

TABLE IV
Lorenz Attractor with mismatched transition dynamics.

	MSE (dB)	NLL ($\times 10^3$)	Time ($\times 10^{-1}$ s)
EKF	3.469 ± 0.463	4.457 ± 6.740	1.075
UKF	*3.442 ± 0.462*	4.414 ± 6.917	2.095
Danse	3.546 ± 0.449	**0.040 ± 0.004**	**0.005**
KalmanNet	22.478 ± 0.492	0.178 ± 0.020	0.690
AMD	**3.415 ± 0.466**	*0.085 ± 0.010*	*0.017*

TABLE V
Lorenz Attractor with erroneous transition dynamics.

	MSE (dB)	NLL ($\times 10^3$)	Time ($\times 10^{-1}$ s)
EKF	4.831 ± 0.353	0.198 ± 0.019	1.239
UKF	*4.638 ± 0.369*	0.126 ± 0.011	2.739
Danse	**4.539 ± 0.373**	**0.018 ± 0.011**	**0.007**
KalmanNet	19.038 ± 1.609	*0.085 ± 0.026*	0.845
AMD	4.668 ± 0.634	0.182 ± 0.029	*0.012*

We also include a fully model-informed variant of AMD, where the transition dynamics are entirely known. Under this configuration, AMD achieves even better estimation accuracy than the model-based EKF. This indicates that, even when the model is fully specified, AMD does not degenerate into a purely model-driven method. Instead, the data-driven prior remains effective in capturing residual dynamics induced by noise, model simplifications, or subtle nonlinearities, thereby providing flexible distributional corrections. In this process, AMD not only integrates prior model knowledge but also enhances it through adaptive data-driven mechanisms. Fig. 4 provides a visual comparison of the posterior state estimates from different methods, illustrating AMD's superior performance in reconstructing system trajectories under constrained measurement conditions.

*c) **Mismatched Transition Dynamics**:* To assess the robustness of AMD under model uncertainty, we further evaluate its performance in the presence of mismatched transition dynamics. In this experiment, instead of providing the true nonlinear transition function, we approximate the system evolution using a second-order Taylor expansion, which serves as a coarse approximation of the true dynamics. The measurement model follows the full observation setup defined in Eq. (23). The process and measurement noise covariances are set to $q^2 = 0.1$ and $r^2 = 0.1$, respectively.

As shown in Table IV, the performance of model-driven methods deteriorates significantly due to the mismatch in the transition function. In contrast, AMD remains robust under such structural model mismatch, effectively leveraging data-driven priors to compensate for the uncertainty in the model and thereby enhancing the model robustness. Even with only approximate or incomplete transition information, AMD consistently outperforms traditional model-driven estimators, demonstrating its strong adaptability in uncertain environments.

*d) **Erroneous Transition Dynamics**:* Beyond partial or mismatched knowledge, model uncertainty may also manifest as errors in the known prior dynamics. In this experiment, we simulate such errors by introducing a slight rotational perturbation to the transition matrix, rotating it by $\theta = 1^\circ$. This small perturbation introduces a model bias of approximately 0.55% [48]. The measurement model again follows the full observation configuration in Eq. (23). The noise covariances are set to $\mathbf{Q} = 0.1\mathbf{I}_3$ for the process noise and $\mathbf{R} = 0.01\mathbf{I}_3$ for the observation noise.

As shown in Table V, such model inaccuracies lead to degradation in the performance of model-driven estimators. In contrast, AMD is able to partially compensate for the erroneous model through data-driven learning, outperforming the model-based EKF. However, its performance is slightly inferior to the hybrid DANSE method. This is attributed to the fact that DANSE operates independently of an explicit transition model, while the hybrid-driven AMD incorporates (potentially inaccurate) model information, which cannot be fully disregarded during inference.

TABLE VI
Chen and Rössler Attractor with subsampled measurements.

System	Method	MSE (dB)	NLL ($\times 10^2$)	Time ($\times 10^{-1}$ s)
Chen Attractor	EKF	0.574 ± 1.752	1.481 ± 0.272	1.255
	UKF	-1.115 ± 0.730	0.383 ± 0.076	2.665
	Danse	24.430 ± 0.487	31472.000 ± 12099.058	0.004
	AMD full	0.456 ± 2.166	3.055 ± 1.322	0.010
	AMD	2.519 ± 3.999	4.086 ± 3.429	0.010
Rössler Attractor	EKF	-16.811 ± 0.309	9.962 ± 0.001	2.489
	UKF	-16.981 ± 0.315	9.960 ± 0.001	0.380
	Danse	-2.674 ± 4.161	10.737 ± 9.367	0.005
	AMD full	-12.356 ± 1.575	0.057 ± 0.045	0.008
	AMD	-12.988 ± 0.480	-0.007 ± 0.004	0.008

C. Another Two More Chaotic Dynamical Systems

As a final experiment, we evaluate the generalization ability of AMD on two additional chaotic dynamical systems: the Chen attractor [10] and the Rössler attractor [9]. The measurement matrix follows the underdetermined configuration defined in Eq. (24). The process and observation noise covariances are set to $q^2 = 0.01$ and $r^2 = 0.1$, respectively.

The results, summarized in Table VI, demonstrate that AMD consistently achieves competitive estimation performance across different chaotic systems, effectively enhancing both data representation and model robustness. This highlights the framework's adaptability and robustness in handling complex, nonlinear dynamical processes beyond the Lorenz system.

IV. Conclusion

A novel state estimation framework, AMD, has been proposed in this paper. This framework adopts a hybrid-driven architecture that has adaptively cross-coupled prior information and has effectively addressed challenges posed by limited data and model uncertainty. By complementarily integrating the strengths of both model-driven and data-driven approaches while mitigating their respective limitations, AMD has enhanced data representation and model robustness, thereby improving estimation accuracy across a wide range of conditions. Extensive evaluations have been carried out on three representative nonlinear chaotic dynamical systems under diverse state estimation scenarios. The results have demonstrated that AMD has consistently delivered competitive performance, highlighting its effectiveness and adaptability in complex and challenging environments. Future work will

focus on extending the AMD framework to estimation tasks involving non-sensor modalities or partially known observation models [38]. In addition, the robustness of AMD under mismatched training and testing conditions will be explored, along with its applicability to real-world industrial control scenarios.

REFERENCES

[1] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[2] X. Bai, G. Li, M. Ding, L. Yu and Y. Sun, Recursive strong tracking filtering for power harmonic detection with outliers-resistant event-triggered mechanism, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 4, art. no. 100023, Dec. 2024.

[3] T. Bao, Y. Zhao, S. Zaidi, S. Xie, P. Yang and Z. Zhang, A deep Kalman filter network for hand kinematics estimation using sEMG, *Pattern Recognition Letters*, vol. 143, pp. 88–94, 2021.

[4] P. Becker, H. Pandya, G. Gebhardt, C. Zhao, C. Taylor and G. Neumann, Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces, *International Conference on Machine Learning (ICML)*, pp. 544–552, 2019.

[5] C. Bishop and N. Nasrabadi, Pattern recognition and machine learning, *Springer*, vol. 4, no. 4, 2006.

[6] R. Caballero-Águila, J. Hu and J. Linares-Pérez, Filtering and smoothing estimation algorithms from uncertain nonlinear observations with time-correlated additive noise and random deception attacks, *International Journal of Systems Science*, vol. 55, no. 10, pp. 2023–2035, 2024.

[7] R. Caballero-Águila and J. Linares-Pérez, Centralized fusion estimation in networked systems: addressing deception attacks and packet dropouts with a zero-order hold approach, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 4, art. no. 100021, Dec. 2024.

[8] K. Cao, J. Li, R. Song and Y. Li, HE2LM-AD: Hierarchical and efficient attitude determination framework with adaptive error compensation module based on ELM network, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 418–431, 2023.

[9] S. Čelikovskỳ and G. Chen, On the generalized Lorenz canonical form, *Chaos, Solitons & Fractals*, vol. 26, no. 5, pp. 1271–1276, 2005.

[10] G. Chen and T. Ueta, Yet another chaotic attractor, *International Journal of Bifurcation and Chaos*, vol. 9, no. 07, pp. 1465–1466, 1999.

[11] H. Chen, Q. Chen, B. Shen and Y. Liu, Parameter learning of probabilistic Boolean control networks with input-output data, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 1, art. no. 100005, Mar. 2024.

[12] J. Chen and Y. Liu, Probabilistic physics-guided machine learning for fatigue data analysis, *Expert Systems with Applications*, vol. 168, pp. 114316, 2021.

[13] K. Course and P. Nair, State estimation of a physical system with unknown governing equations, *Nature*, vol. 622, no. 7982, pp. 261–267, 2023.

[14] D. Dai, J. Li, Y. Song and F. Yang, Event-based recursive filtering for nonlinear bias-corrupted systems with amplify-and-forward relays, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2332419, 2024.

[15] R. Dey and F. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, 2017.

[16] R. Fablet, S. Ouala and C. Herzet, Bilinear residual neural network for the identification and forecasting of geophysical dynamics, *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1477–1481, 2018.

[17] M. Fraccaro, S. Kamronn, U. Paquet and O. Winther, A disentangled recognition and nonlinear dynamics model for unsupervised learning, *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] P. Gao, C. Jia and A. Zhou, Encryption-decryption-based state estimation for nonlinear complex networks subject to coupled perturbation, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2357796, 2024.

[19] V. Garcia Satorras, Z. Akata and M. Welling, Combining generative and discriminative models for hybrid inference, *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] Q. Ge, Y. Li, Y. Wang, X. Hu, H. Li and C. Sun, Adaptive Kalman filtering based on model parameter ratios, *IEEE Transactions on Automatic Control*, vol. 69, no. 9, pp. 6230–6237, 2024.

[21] A. Ghosh, A. Honoré and S. Chatterjee, DANSE: Data-driven non-linear state estimation of model-free process in unsupervised learning setup, *IEEE Transactions on Signal Processing*, 2024.

[22] A. Ghosh, Y. Eldar and S. Chatterjee, Data-driven bayesian state estimation with compressed measurement of model-free process using semi-supervised learning, *arXiv preprint arXiv:2407.07368*, 2024.

[23] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber and X. Alameda-Pineda, Dynamical variational autoencoders: A comprehensive review, *arXiv preprint arXiv:2008.12595*, 2020.

[24] M. Gruber, An approach to target tracking, *MIT Lincoln Laboratory*, 1967.

[25] Y. Guo, Z. Wang, J.-Y. Li and Y. Xu, An impulsive approach to state estimation for multirate singularly perturbed complex networks under bit rate constraints, *IEEE Transactions on Cybernetics*, vol. 55, no. 3, pp. 1197–1209, Mar. 2025.

[26] A. Hasan, I. Kuncara, A. Widyotriatmo, O. Osen and R. T. Bye, Secure state estimation and control for autonomous ships under cyberattacks, *Systems Science & Control Engineering*, vol. 13, no. 1, art. no. 2518964, 2025.

[27] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed and A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework, *International Conference on Learning Representations*, 2017.

[28] N. Higham, Accuracy and stability of numerical algorithms, *SIAM*, 2002.

[29] G. Karniadakis, I. Kevrekidis, L. Lu, P. Perdikaris, S. Wang and L. Yang, Physics-informed machine learning, *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[30] R. Krishnan, U. Shalit and D. Sontag, Structured inference networks for nonlinear state space models, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[31] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] A. Li, P. Wu and M. Kennedy, Replay overshooting: Learning stochastic latent dynamics with the extended kalman filter, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 852–858, 2021.

[33] H. Li, Z. Zhang, T. Li and X. Si, A review on physics-informed data-driven remaining useful life prediction: Challenges and opportunities, *Mechanical Systems and Signal Processing*, vol. 209, pp. 111120, 2024.

[34] T. Lin, A. RoyChowdhury and S. Maji, Bilinear CNN models for fine-grained visual recognition, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457, 2015.

[35] T. Liu, Z. Wang, Y. Liu and R. Wang, Unscented-Kalman-filter-based remote state estimation for complex networks with quantized measurements and amplify-and-forward relays, *IEEE Transactions on Cybernetics*, vol. 54, no. 11, pp. 6819–6831, Nov. 2024.

[36] Z. Liu, Z. Guo, Z. Cen, H. Zhang, J. Tan, B. Li and D. Zhao, On the robustness of safe reinforcement learning under observational perturbations, *arXiv preprint arXiv:2205.14691*, 2022.

[37] E. Lorenz, Deterministic nonperiodic flow 1, *Universality in Chaos, 2nd Edition*, pp. 367–378, 2017.

[38] X. Luo, H. Wu and Z. Li, NeuLFT: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors, *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 6148–6166, Jun. 2023.

[39] N. Mangan, J. Kutz, S. Brunton and J. Proctor, Model selection for dynamical systems via sparse regression and information criteria, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2204, pp. 20170009, 2017.

[40] L. Medsker and L. Jain, Recurrent neural networks, *Design and Applications*, vol. 5, no. 64-67, pp. 2, 2001.

[41] C. Meng, S. Seo, D. Cao, S. Griesemer and Y. Liu, When physics meets machine learning: A survey of physics-informed machine learning, *arXiv preprint arXiv:2203.16797*, 2022.

[42] X. Meng, H. Wang, Y. Li and Y. Shen, Unscented Kalman filtering for nonlinear systems with stochastic nonlinearities under FlexRay protocol, *International Journal of Network Dynamics and Intelligence*, vol. 4, no. 2, art. no. 100010, Jun. 2025.

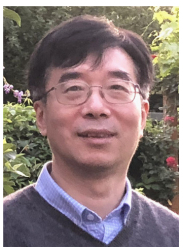
[43] R. Nascimento and F. Viana, Fleet prognosis with physics-informed recurrent neural networks, *arXiv preprint arXiv:1901.05512*, 2019.

[44] T. Needham, A visual explanation of Jensen's inequality, *The American Mathematical Monthly*, vol. 100, no. 8, pp. 768–771, 1993.

[45] X. Ni, G. Revach, N. Shlezinger, R. Van Sloun and Y. Eldar, RTSNet: Deep learning aided Kalman smoothing, *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5902–5906, 2022.

[46] B. Qu, D. Peng, Y. Shen, L. Zou and B. Shen, A survey on recent advances on dynamic state estimation for power systems, *International Journal of Systems Science*, vol. 55, no. 16, pp. 3305–3321, 2024.
[47] M. Raissi, P. Perdikaris and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
[48] G. Revach, N. Shlezinger, X. Ni, A. Escoriza, R. Van Sloun and Y. Eldar, KalmanNet: Neural network aided Kalman filtering for partially known dynamics, *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.
[49] G. Revach, N. Shlezinger, T. Locher, X. Ni, R. Van Sloun and Y. Eldar, Unsupervised learned Kalman filtering, *2022 30th European Signal Processing Conference (EUSIPCO*, pp. 1571–1575, 2022.
[50] D. Ruhe and P. Forré, Self-supervised inference in state-space models, *arXiv preprint arXiv:2107.13349*, 2021.
[51] K. Saoudi, K. Bdirina and K. Guesmi, Robust estimation and control of uncertain affine nonlinear systems using predictive sliding mode control and sliding mode observer, *International Journal of Systems Science*, vol. 55, no. 7, pp. 1480–1492, 2024.
[52] N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin and P. Otero, Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter, *Sensors*, vol. 21, no. 4, pp. 1149, 2021.
[53] Z. Shi, Incorporating Transformer and LSTM to Kalman Filter with EM algorithm for state estimation, *arXiv preprint arXiv:2105.00250*, 2021.
[54] W. Song, Z. Wang, Z. Li and H. Dong, Multi-sensor particle filtering for nonlinear complex networks with heterogeneous measurements under non-Gaussian noises, *IEEE Transactions on Cybernetics*, in press, DOI: 10.1109/TCYB.2025.3623631.
[55] W. Song, Z. Wang, Z. Li, J. Wang and Q.-L. Han, Nonlinear filtering with sample-based approximation under constrained communication: progress, insights and trends, *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 7, pp. 1539–1556, Jul. 2024.
[56] W. Wang, L. Ma, Q. Rui and C. Gao, A survey on privacy-preserving control and filtering of networked control systems, *International Journal of Systems Science*, vol. 55, no. 11, pp. 2269–2288, 2024.
[57] Y. Wang and C. Lin, Runge-Kutta neural network for identification of dynamical systems in high accuracy, *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 294–307, 1998.
[58] Y. Wang, C. Wen and X. Wu, Fault detection and isolation of floating wind turbine pitch system based on Kalman filter and multi-attention 1DCNN, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2362169, 2024.
[59] M. Yuan and W. Qian, Adaptive output feedback tracking control for nonlinear systems with unknown growth rate, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 1, art. no. 100002, Mar. 2024.
[60] Y. Zhan, Z. Li, M. Niu, Z. Zhong, S. Nobuhara, K. Nishino and Y. Zheng, KFD-NeRF: Rethinking Dynamic NeRF with Kalman Filter, *European Conference on Computer Vision* pp. 1–18, 2024.
[61] R. Zhang, H. Liu, Y. Liu and H. Tan, Dynamic event-triggered state estimation for discrete-time delayed switched neural networks with constrained bit rate, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2334304, 2024.
[62] Y. Zhang, G. Liu and X. Song, Unscented recursive three-step filter based unbiased minimum-variance estimation for a class of nonlinear systems, *International Journal of Systems Science*, vol. 56, no. 2, pp. 227–236, 2025.
[63] C. Zhao, L. Sun, Z. Yan, G. Neumann, T. Duckett and R. Stolkin, Learning Kalman Network: A deep monocular visual odometry for on-road driving, *Robotics and Autonomous Systems*, vol. 121, pp. 103234, 2019.
[64] L. Zou, Z. Wang, H. Dong, X. Yi and Q.-L. Han, Recursive filtering under probabilistic encoding-decoding schemes: handling randomly occurring measurement outliers, *IEEE Transactions on Cybernetics*, vol. 54, no. 6, pp. 3378–3391, Jun. 2024.

Lizhang Wang received the bachelor's degree from College of Intelligence and Computing, Tianjin University, Tianjin, China, in 2024. He is currently working toward the M.Sc. degree in computer science and technology at Tongji University, Shanghai, China. His general research interests are in topics related to state estimation methods and multi-sensor fusion approaches.



Zidong Wang (SM'03-F'14) received the B.Sc. degree in mathematics in 1986 from Suzhou University, Suzhou, China, and the M.Sc. degree in applied mathematics in 1990 and the Ph.D. degree in electrical engineering in 1994, both from Nanjing University of Science and Technology, Nanjing, China.

He is currently Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the UK. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published a number of papers in international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for *International Journal of Systems Science*, the Editor-in-Chief for *Neurocomputing*, the Editor-in-Chief for *Systems Science & Control Engineering*, and an Associate Editor for 12 international journals including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics-Part C. He is a Member of the Academia Europaea, a Member of the European Academy of Sciences and Arts, an Academician of the International Academy for Systems and Cybernetic Sciences, a Fellow of the IEEE, a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.

Qinyuan Liu received the B.Eng. degree in measurement and control technology and instrumentation from Huazhong University of Science and Technology, Wuhan, China, in 2012, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2017.

He is currently a Professor in the Department of Computer Science and Technology, Tongji University, Shanghai, China. From Jul. 2015 to Sep. 2016, he was a Researcher Assistant in the Department of Electronic & Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. From Jan. 2016 to Jan. 2017, he was an international researcher in the Department of Computer Science, Brunel University London, UK. His research interests include networked control systems, multi-agent systems, and distributed filtering. He is an active reviewer for many international journals.