DVSI: DYNAMICAL VARIATIONAL SYSTEM IDENTIFICATION WITH HIGH-DIMENSIONAL STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

We address the problem of identifying a nonlinear stochastic Markov dynamical system that has many state variables represented together by a high-dimensional state vector. We focus on the case where the state-transition model (STM) of the system has no known analytical structure due to lack of physics knowledge, is modeled using a neural network, and is identified from noisy measurements through a known measurement model. Particle-filter-based system identification, including differentiable particle filters (DPFs), suffers from the curse of dimensionality in this setting. To address the curse of dimensionality, we develop Dynamical Variational System Identification (DVSI): transformer-based dynamical variational inference methods that jointly learn the STM and an amortized posterior of the state trajectory by maximizing an evidence lower bound. We develop four DVSI methods, spanning autoregressive versus factorized and smoothing versus filtering posteriors. Through experiments on stochastic Lorenz-63 and Lorenz-96 systems and real magnetoencephalography (MEG) data, we evaluate the DVSI methods for identifying dynamical systems that have up to 200 state variables, where DPFs have significant limitations due to the curse of dimensionality.

1 INTRODUCTION

The identification of dynamical systems from measurements is a fundamental learning problem arising across control, robotics, neuroscience, and the physical sciences. A common representation is a stochastic state-space model, in which a latent state evolves according to a state-transition model (STM) and is observed through noisy measurements. System identification seeks to recover the unknown dynamics governing this evolution from observed trajectories. While classical approaches often exploit analytical structure motivated by prior physical knowledge, modern applications increasingly involve complex systems whose dynamics are only partially known, or for which a flexible neural model of the STM is desired. This becomes particularly challenging when the physical state itself is high-dimensional, as occurs in spatially distributed, biological, and geophysical processes. We therefore ask: *how can an unknown nonlinear stochastic STM be learned from noisy measurements when the underlying physical state is high-dimensional?*

The key difficulty in this setting is that the state trajectories required for learning the dynamics are not directly observed. If the latent states were available, learning a parameterized transition model from consecutive state pairs would reduce to a standard supervised learning problem. Instead, system identification from noisy measurements inherently requires reasoning about the posterior distribution of the hidden state trajectory jointly with learning the dynamics. Sequential Monte Carlo (SMC) methods, commonly known as particle filters (PFs), provide a general mechanism for approximating such posterior distributions in nonlinear and non-Gaussian state-space models, and have consequently been widely used for state estimation and system identification (Arulampalam et al., 2002; Doucet & Johansen, 2011; Särkkä, 2013; Schön et al., 2015). Their generality, however, comes at a severe cost in high dimensions: particle weights tend to degenerate as the dimension of the inferred state grows, and maintaining an informative particle approximation can require a rapidly increasing, and in canonical settings exponentially growing, number of particles (Daum & Huang, 2003; Bengtsson et al., 2008). Thus, the latent-state inference mechanism itself becomes a principal bottleneck for learning high-dimensional dynamics.

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

Recent differentiable PFs (DPFs) combine this probabilistic inference machinery with learnable neural transition and measurement models, enabling state estimation and model learning to be optimized end-to-end (Jonschkowski et al., 2018; Corenflos et al., 2021; Kloss et al., 2021; Chen & Li, 2025b). Yet differentiability does not remove the underlying particle representation, and DPFs therefore inherit the unfavorable dimensional scaling of SMC. Ensemble-based alternatives can operate in higher-dimensional regimes by replacing a general posterior representation with estimates of low-order statistics; for example, auto-differentiable ensemble Kalman filtering has been used to jointly learn unknown dynamics and latent states in high-dimensional Lorenz-96 systems (Chen et al., 2022). Such scalability, however, relies on restrictive Gaussian and ensemble approximations that may poorly represent nonlinear posterior distributions. These limitations motivate a different approach to learning latent dynamics: one that retains probabilistic inference over the hidden trajectory while avoiding an explicit particle approximation whose fidelity deteriorates with state dimension.

Rather than approximating the latent-state posterior through a population of particles, we propose to learn this posterior directly using amortized variational inference. Specifically, we introduce *Dynamical Variational System Identification (DVSI)*, in which a parameterized inference model maps a sequence of noisy measurements into a distribution over plausible state trajectories, while a second learnable model represents the unknown stochastic STM. The two models are trained jointly by maximizing a variational lower bound on the measurement likelihood. This objective couples two complementary requirements: the inferred states must explain the observed measurements through the known measurement model, while their evolution must be consistent with the learned STM. Unlike generic dynamical latent-variable models (Chung et al., 2015; Krishnan et al., 2017), the inferred states are not arbitrary learned representations: the known measurement model anchors them to the physical state space. Consequently, the learned transition distribution constitutes an identified model of the underlying dynamics rather than merely a latent generative representation.

The DVSI formulation gives rise to a family of inference architectures that trade posterior expressiveness and temporal context for computational efficiency. We consider two complementary design axes: whether state inference is *autoregressive*, explicitly conditioning each inferred state on the preceding one, or *factorized* across time; and whether inference is *smoothing*, exploiting the complete measurement sequence, or *filtering*, using only measurements available up to the current time. Combining these choices yields four representative DVSI realizations. We instantiate their variational posteriors using transformer-based inference networks and model the unknown STM by a neural transition model. In contrast to particle-based learning, these realizations do not maintain an explicit particle population to represent the state posterior, while the factorized variants additionally admit parallel computation across time.

Our main contributions are summarized as follows:

- We introduce DVSI, a variational framework for identifying unknown nonlinear stochastic dynamics from noisy measurements. By jointly learning an amortized posterior over physical state trajectories and the unknown STM, DVSI replaces particle-based latent-state inference with a parametric variational representation grounded by the known measurement model.
- We develop a family of DVSI algorithms spanning autoregressive versus factorized and smoothing versus filtering posterior structures, exposing explicit tradeoffs between inference expressiveness, available temporal context, and computational parallelism.
- We evaluate DVSI on nonlinear synthetic systems (stochastic Lorenz-63 and Lorenz-96) and high-dimensional real-world trajectories (MEG data). The experiments demonstrate accurate identification under nonlinear noisy measurements and scalability to stochastic dynamics with up to 200 state variables, including regimes in which DPFs become computationally prohibitive or fail to train reliably.

2 Related Work

System identification with latent states. System identification in state-space models is classically formulated as parameter estimation when the underlying states are unobserved (Ljung, 1998; Särkkä, 2013). Likelihood-based approaches couple identification with latent-state inference, commonly using expectation maximization, where state smoothing and parameter estimation are alter-

nated (Schön et al., 2011). For nonlinear and non-Gaussian models, the required smoothing distributions are generally intractable, motivating SMC-based approximations for maximum-likelihood and Bayesian system identification (Schön et al., 2011; 2015). Variational inference has also been considered as an alternative to particle-based parameter estimation: Courts et al. (2023) formulate nonlinear system identification as joint variational optimization over the unknown state trajectory and model parameters, while Dutra (2025) develops posterior parameterizations that enable large-scale and parallelizable variational system identification. These approaches primarily target parameter estimation in prescribed state-space model structures, while we consider learning a structure-free neural STM in a high-dimensional physical state space.

Differentiable sequential inference. Recent works incorporate Bayesian filtering algorithms into differentiable computation graphs, allowing their process, measurement, and uncertainty models to be learned end-to-end (Jonschkowski et al., 2018; Kloss et al., 2021). DPFs retain the nonparametric particle representation of SMC while making filtering differentiable through learned models and differentiable or relaxed resampling mechanisms (Jonschkowski et al., 2018; Karkus et al., 2018; Corenflos et al., 2021; Chen & Li, 2025b;a). This flexibility makes them suitable for nonlinear state estimation, but it does not eliminate the dimensionality dependence of particle inference: representing increasingly high-dimensional posterior distributions can require rapidly growing particle populations and is prone to weight degeneracy (Daum & Huang, 2003; Bengtsson et al., 2008). Differentiable Kalman-type methods provide a complementary direction by maintaining compact parametric beliefs (Shlezinger et al., 2025); in particular, auto-differentiable ensemble Kalman filters combine ensemble-based state recovery with neural learning of unknown dynamics and scale to high-dimensional systems (Chen et al., 2022), but through Gaussian or low-order moment approximations of the filtering distribution. Our approach instead departs from recursive filtering during training and learns a parametric variational posterior over the latent trajectory, avoiding an explicit particle representation while retaining a probabilistic description of the hidden states.

Deep variational dynamical models. Deep latent-variable models have been developed for learning dynamical representations from sequential measurements, including variational recurrent models (Chung et al., 2015), deep Markov models (Krishnan et al., 2017), deep variational Bayes filters (Karl et al., 2017), Kalman-inspired variational models (Fraccaro et al., 2017), and continuous-time formulations such as latent neural ODEs (Rubanova et al., 2019), ODE2VAE (Yildiz et al., 2019), latent SDEs (Li et al., 2020), and continuous recurrent units (Schirmer et al., 2022); see Girin et al. (2021) for a comprehensive review. These approaches generally learn the latent representation together with an measurement decoder, such that the inferred states need not correspond to a prescribed physical state. A complementary line of work studies identifiable representation learning for dynamical systems, seeking to recover the underlying latent states or innovations from temporal measurements under suitable structural assumptions (Morioka et al., 2021; Yao et al., 2022; Song et al., 2023). More recently, Hızlı et al. (2025) established conditions for identifying both latent states and nonlinear latent transition functions from high-dimensional measurements. In contrast to these works, we do not seek to discover a latent representation from an unknown measurement mapping, but consider a prescribed, potentially high-dimensional physical state with a known measurement model, and use variational inference as a scalable mechanism for learning the stochastic STM governing that state from noisy measurements.

3 High-Dimensional System Identification

3.1 Problem Formulation

We consider a discrete-time stochastic dynamical system with state $\mathbf{x}_t \in \mathbb{R}^m$, whose dynamics follow a first-order Markov model $\mathbf{x}_t \sim p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$, where $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the STM and $\boldsymbol{\phi}$ denotes its unknown parameters. We focus on the challenging setting in which the STM has no prescribed analytical structure and is represented by a neural probabilistic model. Specifically, we consider

$$p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1})\right), \tag{1}$$

where the conditional mean and covariance are parameterized by neural networks. Our formulation also applies when part or all of the STM is analytically structured, in which case $\boldsymbol{\phi}$ represents only

its unknown components. Known exogenous inputs can similarly be incorporated by conditioning $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ on $\mathbf{u}_t$; we omit them for notational clarity.

The state is not observed directly. Instead, at each time t we observe $\mathbf{y}_t \in \mathbb{R}^n$ through a known measurement model

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}), \quad \text{or equivalently} \quad p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{h}(\mathbf{x}_t), \sigma_w^2 \mathbf{I}\right). \tag{2}$$

Both the measurement function $\mathbf{h}$ and the measurement-noise statistics are assumed known.

Let $\mathbf{x}_{1:T} \triangleq (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and $\mathbf{y}_{1:T} \triangleq (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ denote a state trajectory and its corresponding measurement sequence. Assuming a known initial-state distribution $p(\mathbf{x}_1)$, their joint distribution factorizes as

$$p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = p(\mathbf{x}_1) \prod_{t=2}^{T} p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t). \tag{3}$$

Given a dataset of measurement trajectories $\mathcal{D} = \left\{\mathbf{y}_{1:T}^{(i)}\right\}_{i=1}^{N}$, the system-identification objective is to learn $\boldsymbol{\phi}$ by maximizing the marginal likelihood of the measurements,

$$\boldsymbol{\phi}^\star \in \arg\max_{\boldsymbol{\phi}} \sum_{i=1}^{N} \log p_{\boldsymbol{\phi}}\left(\mathbf{y}_{1:T}^{(i)}\right). \tag{4}$$

For clarity, in the sequel we develop the formulation for a single sequence $\mathbf{y}_{1:T}$; extension to the dataset $\mathcal{D}$ follows by summation as in (4).

3.2 The Latent-State Inference Bottleneck

If the state trajectory $\mathbf{x}_{1:T}$ were observed, learning the STM would amount to fitting the conditional model $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ from consecutive state pairs. With noisy measurements, however, the states must be marginalized out, such that

$$p_{\boldsymbol{\phi}}(\mathbf{y}_{1:T}) = \int p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})\, \mathrm{d}\mathbf{x}_{1:T}. \tag{5}$$

Consequently, evaluating the identification objective requires integrating over an mT-dimensional latent trajectory. For nonlinear neural STMs and nonlinear measurement models, this integral is generally intractable.

More fundamentally, state inference is not merely an auxiliary task needed to evaluate (5); it is intrinsically coupled to learning the dynamics. Indeed, by Fisher's identity, the gradient of the log-evidence can be expressed as

$$\begin{aligned} \nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\phi}}(\mathbf{y}_{1:T}) &= \mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}\left[\nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})\right] \\ &= \sum_{t=2}^{T} \mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{y}_{1:T})}\left[\nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\right]. \end{aligned} \tag{6}$$

Thus, updating the STM requires expectations with respect to the latent-state smoothing posterior

$$p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \propto p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}), \tag{7}$$

which itself depends on the unknown STM. System identification therefore entails repeatedly inferring the hidden state trajectory while simultaneously updating the model that governs its evolution.

This coupling becomes particularly problematic as the state dimension m grows. Particle-based methods approximate (7) using a finite collection of weighted state samples, but maintaining a representative particle population becomes increasingly difficult in high-dimensional state spaces. Moreover, this inference must be carried out repeatedly as $\boldsymbol{\phi}$ changes throughout training. The central challenge is thus not merely to parameterize a sufficiently expressive high-dimensional STM, but to obtain a scalable approximation of the latent-state posterior through which the STM can be learned.

4 Dynamical Variational System Identification

We now introduce DVSI, which addresses the latent-state inference bottleneck identified in Section 3 by replacing particle-based inference with an amortized variational approximation: rather than maintaining an explicit population of state hypotheses while the STM is being learned, DVSI jointly trains a parametric distribution over latent state trajectories and the unknown transition model.

Table 1: Variational posterior families considered in DVSI.

	measurement context	
State dependence	Smoothing, $\mathbf{c}_t = \mathbf{y}_{1:T}$	Filtering, $\mathbf{c}_t = \mathbf{y}_{1:t}$
Autoregressive, (11)	DVSI-I	DVSI-II
Factorized, (13)	DVSI-III	DVSI-IV

4.1 Variational System Identification

Let $q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ be a parameterized approximation of the intractable smoothing posterior in (7), with parameters $\boldsymbol{\theta}$. Introducing this distribution into the evidence in (5) yields

$$\log p_{\boldsymbol{\phi}}(\mathbf{y}_{1:T}) = \log \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}\left[\frac{p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T},\mathbf{y}_{1:T})}{q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}\right] \geq \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}\left[\log \frac{p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T},\mathbf{y}_{1:T})}{q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}\right] \triangleq \mathcal{L}_{\mathrm{DVSI}}(\mathbf{y}_{1:T};\boldsymbol{\theta},\boldsymbol{\phi}), \tag{8}$$

where the inequality follows from Jensen's inequality. Using the factorization in (3), the resulting evidence lower bound (ELBO) can be written as

$$\mathcal{L}_{\mathrm{DVSI}} = \underbrace{\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log p(\mathbf{y}_t|\mathbf{x}_t)\right]}_{\text{measurement consistency}} - \underbrace{D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \,\|\, p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T})\right)}_{\text{dynamical consistency}}, \tag{9}$$

where $p_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}) = p(\mathbf{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$. The two terms in (9) play complementary roles. The measurement-consistency term encourages the inferred states to explain the observed sequence through the known measurement model, while the dynamical-consistency term encourages their evolution to agree with the learned STM. The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are therefore jointly optimized as

$$(\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star) \in \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{i=1}^{N} \mathcal{L}_{\mathrm{DVSI}}(\mathbf{y}_{1:T}^{(i)};\boldsymbol{\theta},\boldsymbol{\phi}). \tag{10}$$

The role of the known measurement model is important in interpreting (10) as system identification. In generic dynamical latent-variable models, the latent representation and measurement decoder are typically learned jointly, and the resulting latent coordinates need not correspond to a prescribed physical state. Here, $p(\mathbf{y}_t|\mathbf{x}_t)$ is fixed, and therefore constrains the variational variables $\mathbf{x}_t$ to represent states that are consistent with the prescribed measurement process. Subject to the observability and identifiability properties of this process, the transition model learned in (10) is consequently associated with the evolution of the physical state itself. The variational model $q_{\boldsymbol{\theta}}$ serves as an inference mechanism for learning this transition law; once training is complete, the identified system is represented by $p_{\boldsymbol{\phi}^\star}(\mathbf{x}_t|\mathbf{x}_{t-1})$.

4.2 Scalable Variational State Inference

The general objective in (8) allows different choices of the variational family. We consider two design axes that control the information available to the inference model and its computational structure: (i) *smoothing* inference allows the estimate of $\mathbf{x}_t$ to depend on the complete sequence, i.e., on the context $\mathbf{c}_t = \mathbf{y}_{1:T}$, whereas *filtering* inference restricts it to the measurements available up to time t, i.e., $\mathbf{c}_t = \mathbf{y}_{1:t}$; (ii) an *autoregressive* posterior explicitly conditions each inferred state on its predecessor, whereas a factorized (mean-field) posterior removes this conditional dependence, enabling the states at different time indices to be sampled in parallel. Combining these choices gives the four DVSI realizations summarized in Table 1.

For the autoregressive variants (DVSI-I and II), we use

$$q_{\boldsymbol{\theta}}^{\mathrm{AR}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) = q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)\prod_{t=2}^{T} q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t). \tag{11}$$

Let $\bar{q}^{\mathrm{AR}}_{\boldsymbol{\theta},t-1}$ denote the marginal distribution of $\mathbf{x}_{t-1}$ induced by (11). Substituting (11) into (8) gives

$$\mathcal{L}_{\mathrm{AR}} = \sum_{t=1}^{T} \mathbb{E}_{q^{\mathrm{AR}}_{\boldsymbol{\theta}}} \left[\log p(\mathbf{y}_t|\mathbf{x}_t)\right] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)\|p(\mathbf{x}_1)\right) - \sum_{t=2}^{T} \mathbb{E}_{\bar{q}^{\mathrm{AR}}_{\boldsymbol{\theta},t-1}} \left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)\|p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\right)\right]. \tag{12}$$

The autoregressive construction can capture dependence between consecutive states directly in the variational posterior, but sampling $\mathbf{x}_{1:T}$ must proceed sequentially.

For the factorized variants (DVSI-III and IV), we instead use

$$q^{\mathrm{MF}}_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) = \prod_{t=1}^{T} q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t). \tag{13}$$

In this case, (8) becomes

$$\mathcal{L}_{\mathrm{MF}} = \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t)} \left[\log p(\mathbf{y}_t|\mathbf{x}_t)\right] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)\|p(\mathbf{x}_1)\right) - \sum_{t=2}^{T} \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{c}_{t-1})} \left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t)\|p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\right)\right]. \tag{14}$$

Although the variational posterior in (13) is factorized across time, the learned prior remains Markovian through $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$. Consequently, the STM is still learned from transitions between consecutive inferred states. The key computational difference is that the parameters and samples of all factors in (13) can be obtained simultaneously, allowing the terms in (14) to be evaluated in a batched manner. The derivations of (12) and (14) are given in Appendix A.

4.3 Neural Parameterization

We parameterize the variational distributions using Gaussian inference networks. For each t, the relevant posterior factor is represented as

$$q_{\boldsymbol{\theta}}(\mathbf{x}_t|\cdot) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta},t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\theta},t})\right), \tag{15}$$

where its conditioning arguments are determined by (11) or (13). The measurement sequence is first mapped into a sequence of embeddings and processed by a Transformer encoder (Vaswani et al., 2017). For the filtering variants, causal attention restricts the representation at time t to $\mathbf{y}_{1:t}$; for the smoothing variants, the representation may additionally incorporate future measurements in $\mathbf{y}_{t+1:T}$. A Gaussian output head maps the resulting representation to $\boldsymbol{\mu}_{\boldsymbol{\theta},t}$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}$. For the autoregressive variants, this head additionally receives the preceding state sample $\mathbf{x}_{t-1}$.

The STM in (1) is implemented by a feed-forward neural network that maps $\mathbf{x}_{t-1}$ to $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1})$ and a diagonal covariance $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1})$. Positive posterior and transition variances are enforced using a softplus output transformation. Since both the variational factors and transition distributions are Gaussian, the conditional KL divergences in (12) and (14) admit closed-form expressions.

Training is carried out using the reparameterization trick. In particular, samples from (15) are generated as

$$\mathbf{x}_t = \boldsymbol{\mu}_{\boldsymbol{\theta},t} + \boldsymbol{\sigma}_{\boldsymbol{\theta},t} \odot \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{16}$$

which permits gradients of the ELBO to propagate through the sampled state trajectory. For the autoregressive variants, the samples are generated sequentially and each $\mathbf{x}_t$ conditions on the realized $\mathbf{x}_{t-1}$. For the factorized variants, all state samples can be generated in parallel. In our implementation, a single Monte Carlo trajectory is used per measurement sequence in each stochastic-gradient update; additional samples can be used to reduce the variance of the ELBO estimator. Algorithm 1 in Appendix B summarizes the resulting training procedure, which applies to all four realizations.

4.4 Discussion

Particle-free inference and scalability. DVSI replaces an explicit particle representation of the latent-state posterior with an amortized parametric model. Its computational requirements therefore depend on the inference-network architecture and state dimension, rather than on maintaining a particle population sufficiently large to represent the posterior. This does not make inference independent of dimensionality, but avoids the particle degeneracy mechanism that motivates the present work. Among the proposed realizations, DVSI-III and DVSI-IV further remove sequential state sampling and permit batched computation across time, whereas DVSI-I and DVSI-II trade this parallelism for a more expressive autoregressive posterior.

Filtering versus smoothing. The smoothing variants exploit measurements from the complete trajectory when inferring each state and are thus naturally suited to offline system identification. The filtering variants restrict inference to information available up to the current time and are consequently applicable when causal state estimates are also required. Importantly, when the objective is solely system identification, the variational inference network is needed during training but need not be retained after learning: the output of DVSI is the identified transition model $p_{\boldsymbol{\phi}^\star}$.

Grounding, observability, and model structure. The grounding of the inferred states in the physical state by the known measurement model requires sufficient observability; this, and extensions to analytically structured STMs and richer variational families, are discussed in Appendix C.

5 Experiments

In this section, we implement the four proposed DVSI methods and compare them against DPF, whose relevant analytical and structural aspects are given in Appendix D. All methods learn a structure-free DNN-based STM as in (1). We perform three sets of experiments as follows:

1. Stochastic Lorenz-63 system with state dimension $m \triangleq \dim(\mathbf{x}_t) = 3$ and a highly nonlinear camera measurement model with measurement dimension $n \triangleq \dim(\mathbf{y}_t) = 64 \gg m$. For $m = 3$, we can implement DPF and benchmark the four DVSI methods against DPF.
2. Stochastic Lorenz-96 system with high state dimension, $m = 20$ to 200, to evaluate how the DVSI methods scale with m.
3. Real MEG data recorded through 144 channels, that is, $m = 144$.

5.1 Performance Measures and Implementation

Performance measures: We evaluate the learned STM using mean square error (MSE $\downarrow$), log-likelihood (LL $\uparrow$), and Wasserstein 2-distance (WD $\downarrow$) for one-step transitions $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ of test state trajectories. A method is good if MSE and WD are low, and LL is high. For real data, the true STM is not known, hence only MSE and LL are used. The measurement noise variance σ_w^2 in (2) is determined by the signal-to-measurement noise ratio (SMNR). All definitions are given in Appendix E.1. Results show mean $\pm$ std over 10 runs with different seeds.

Implementation: The variational posterior $q_{\boldsymbol{\theta}}$ is implemented with a transformer architecture as in Section 4.3, with a causal mask in the self-attention step for the filtering DVSI methods (DVSI-II and IV), and the STM with a dense feed-forward network. All models are implemented in PyTorch and trained with Adam (Kingma & Ba, 2015) and KL annealing (Fu et al., 2019) to avoid posterior collapse. The architectures, hyperparameters, and training procedure are given in Appendix E.2.

5.2 Experiments for Stochastic Lorenz-63 System

The Lorenz-63 system is a 3-dimensional chaotic system that has been used for many state estimation and system identification methods, including learning the governing equations of the system (Brunton et al., 2016). We use it to study how well the methods handle a highly nonlinear measurement model. We use the discretized stochastic Lorenz-63 system with sampling interval $\Delta\tau = 0.02$ and process noise variance $\sigma_e^2 = -10$ dB, following Revach et al. (2022); Ghosh et al. (2024). A 2D camera model is used as the measurement function, following Buchnik et al. (2024);

Table 2: Lorenz-63 system ($m = 3$): performance of the learned STM. Best in **bold**, second-best underlined.

Method	MSE	LL	WD
SMNR = 10 dB			
DVSI-I	$\mathbf{0.35}_{\pm 0.01}$	$\mathbf{-1.15}_{\pm 0.05}$	$\mathbf{0.29}_{\pm 0.03}$
DVSI-II	$\underline{0.39}_{\pm 0.01}$	$\underline{-1.48}_{\pm 0.09}$	$\underline{0.45}_{\pm 0.05}$
DVSI-III	$\underline{0.39}_{\pm 0.01}$	$-1.77_{\pm 0.04}$	$0.79_{\pm 0.03}$
DVSI-IV	$0.42_{\pm 0.02}$	$-2.07_{\pm 0.05}$	$0.96_{\pm 0.04}$
DPF	$0.44_{\pm 0.15}$	$-2.09_{\pm 0.28}$	$0.74_{\pm 0.11}$
SMNR = 0 dB			
DVSI-I	$\mathbf{0.38}_{\pm 0.02}$	$\mathbf{-1.40}_{\pm 0.12}$	$\mathbf{0.42}_{\pm 0.08}$
DVSI-II	$\underline{0.46}_{\pm 0.03}$	$\underline{-2.09}_{\pm 0.09}$	$\underline{0.85}_{\pm 0.05}$
DVSI-III	$1.34_{\pm 0.13}$	$-3.29_{\pm 0.04}$	$2.27_{\pm 0.09}$
DVSI-IV	$1.44_{\pm 0.16}$	$-3.73_{\pm 0.05}$	$2.69_{\pm 0.09}$
DPF	$6.37_{\pm 3.39}$	$-4.85_{\pm 0.47}$	$3.22_{\pm 0.55}$

Table 3: MEG data ($m = 144$): performance of the learned STM. DNC: did not converge.

Method	MSE	LL/dim
SMNR = 10 dB		
DVSI-I	$0.617_{\pm 0.014}$	$-1.274_{\pm 0.031}$
DVSI-II	$\mathbf{0.573}_{\pm 0.001}$	$\mathbf{-1.104}_{\pm 0.005}$
DVSI-III	$0.620_{\pm 0.049}$	$-1.299_{\pm 0.181}$
DVSI-IV	$\underline{0.579}_{\pm 0.005}$	$\underline{-1.132}_{\pm 0.013}$
DPF	DNC	DNC
SMNR = 0 dB		
DVSI-I	$0.604_{\pm 0.002}$	$\mathbf{-1.127}_{\pm 0.003}$
DVSI-II	$0.606_{\pm 0.002}$	$\underline{-1.142}_{\pm 0.005}$
DVSI-III	$\underline{0.601}_{\pm 0.003}$	$-1.856_{\pm 0.114}$
DVSI-IV	$\mathbf{0.600}_{\pm 0.002}$	$-1.607_{\pm 0.072}$
DPF	DNC	DNC

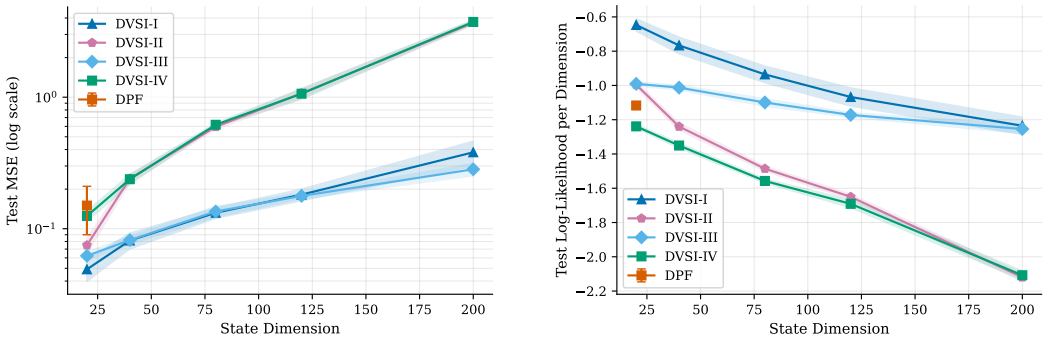
van Nierop et al. (2025). The 2D camera model uses a Gaussian point spread function over a fixed grid of $8 \times 8 = 64$ pixels; the state is projected onto the camera plane according to its first two coordinates, and the third coordinate is perceived as depth. The equations of the Lorenz-63 system and the camera model are given in Appendix E.3. This setup has a low state dimension $m = 3$, but a high measurement dimension $n = 64$. We experiment with two SMNRs, 0 dB and 10 dB.

The results are shown in Table 2. DVSI-I provides the best results for all performance measures at both SMNRs, and DVSI-II the second-best. That is, the autoregressive DVSI methods (DVSI-I and II) outperform the factorized DVSI methods (DVSI-III and IV), in particular in LL and WD, and the gap increases at 0 dB SMNR. We attribute this to the posterior dependence between consecutive states, which is strong at low SMNR and is captured by the autoregressive posterior, but neglected by the factorized posterior. Within both pairs, the smoothing variant outperforms the filtering variant. At 10 dB SMNR, DPF is comparable to the factorized DVSI methods, but with a considerably higher standard deviation; at 0 dB SMNR, DPF degrades considerably. Figure 2 in Appendix F shows trajectories simulated with the learned STMs.

5.3 Experiments for Stochastic Lorenz-96 System

The Lorenz-96 system is a chaotic system that, unlike the Lorenz-63 system, can be defined for any state dimension $m \geq 4$ (Lorenz, 1995). We use it to study how well the methods perform with increasing state dimension. We simulate discretized Lorenz-96 trajectories and make the system stochastic by a random forcing, as described in Appendix E.3. The measurement function is elementwise and nonlinear, $\mathbf{h}(\mathbf{x}_t) = \sin(\mathbf{x}_t/10) + \mathbf{x}_t/10$, with $n = m$; note that $n < m$ would leave state variables unmeasured and lead to a loss of observability, as the methods assume a structure-free STM. We use 10 dB SMNR and $m = 20$, 40, 80, 120, and 200.

The results are shown in Figure 1. DPF achieves good performance at $m = 20$. However, at higher state dimensions, the DPF training diverges, and we are unable to produce any stable result. We believe that the curse of dimensionality is the major reason: at higher dimensions, DPF requires an exponentially growing number of particles (Daum & Huang, 2003; Bengtsson et al., 2008), and using more particles is not feasible due to the memory and computation power of the GPU. Among the four DVSI methods, the smoothing DVSI methods (DVSI-I and III) are the best, and their advantage over the filtering DVSI methods (DVSI-II and IV) grows with m. At 10 dB SMNR, the smoothing DVSI methods also provide the lowest state estimation error and the lowest 5-step prediction error (Appendices G and H). This shows that the DVSI methods scale to high state dimensions, while DPF has limitations. DVSI is not limited to $m = 200$; in our implementation, the network widths grow with m (Table 4 in Appendix E.2), and higher m was not feasible due to the rising computational complexity.



(a) MSE (b) Log-likelihood

Figure 1: Performance of the learned STM on the Lorenz-96 system for different state dimensions m at 10 dB SMNR. DPF fails to converge beyond $m = 20$.

5.4 Experiments for MEG Data

Next, we consider real data: the MEG dataset FaceDetection (Bagnall et al., 2018) of 16 subjects performing a visual discrimination task, with 144 MEG channels, 62 time points per trial, and a predefined split into training (10 subjects) and testing (the remaining 6 subjects); see Appendix E.3. We consider the state $\mathbf{x}_t$ to be the 144 MEG channel recordings at time point t, that means $m = 144$ and $T = 62$. Since we are interested in system identification of Markovian processes, we augment the state vector as $\tilde{\mathbf{x}}_t = (\mathbf{x}_{t-1}^\top, \mathbf{x}_t^\top)^\top$, which makes the system approximately Markov. The noisy measurements are given by $\mathbf{y}_t = \mathbf{h}(\tilde{\mathbf{x}}_t) + \mathbf{w}_t$, where $\mathbf{h} : \tilde{\mathbf{x}}_t \mapsto \mathbf{x}_t$ and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$, with σ_w^2 determined by the SMNR. The task is to learn the STM $p_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}_t | \tilde{\mathbf{x}}_{t-1})$.

The results are shown in Table 3. DPF did not converge (DNC) for any run, as the number of particles is too small to represent a good posterior; using more than 25 particles is not possible on a single V100 GPU, and for fairness, we train each model on one GPU. The four DVSI methods have similar MSE. At 10 dB SMNR, the filtering DVSI methods (DVSI-II and IV) provide the best MSE and LL. At 0 dB SMNR, the autoregressive DVSI methods (DVSI-I and II) provide a considerably higher LL than the factorized DVSI methods (DVSI-III and IV), consistent with the Lorenz-63 results.

5.5 Training Time

We compare the training time on the Lorenz-96 system at $m = 20$, 40, and 120, with the configuration of Section 5.3 and an NVIDIA T4 GPU for all methods. The results are shown in Table 9 in Appendix I. DVSI-IV achieves the lowest total training time for all m (0.04, 0.09, and 0.30 hours for $m = 20$, 40, and 120), and the lowest time per epoch for $m = 40$ and 120. The reason is that the factorized DVSI methods are not autoregressive, and their forward pass can be batched across time on a GPU. DPF has a significantly higher training time per epoch at $m = 20$ (35.92 seconds), about 30 times that of DVSI-I; the bottleneck is the Sinkhorn algorithm in the differentiable resampling.

6 Conclusion

We introduced DVSI for learning a structure-free neural STM of a nonlinear stochastic dynamical system with a high-dimensional state from noisy measurements. DVSI jointly learns the STM and an amortized transformer-based variational posterior of the state trajectory by maximizing an ELBO, where the known measurement model grounds the inferred states in the physical state. On stochastic Lorenz-63 and Lorenz-96 systems and real MEG data, the four DVSI methods learn the STM of systems with up to 200 state variables, whereas DPF fails to converge beyond 20 state variables due to the curse of dimensionality. The autoregressive DVSI methods perform best under the highly nonlinear camera measurement model, the smoothing DVSI methods perform best at high state dimension, and the factorized DVSI methods have the lowest training time. Future work includes learning σ_w^2 jointly with the STM and richer variational families.

AI USE STATEMENT

In this work, we used generative AI tools for interpreting results and for assisting in writing the derivations in Appendix A. We have not used generative AI tools for generating data sets, developing the DVSI framework, formulating mathematical claims or the key steps of the derivations, proposing hypotheses, designing the methodology or experiments, or implementing the methods, and translation, cleaning datasets and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for editing the manuscript text, including improving formulations, proposing structure, formatting, and in plotting results and facilitating the experiments. We have reviewed all AI-assisted work, where the derivations were checked by hand, and the AI-assisted code and plots were verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 4 describes the DVSI methods, Appendix A gives the derivations of the DVSI lower bounds, and Algorithm 1 in Appendix B summarizes the training procedure. Appendix E gives the performance measures, the SMNR, the architectures and hyperparameters, the training procedure, the KL annealing strategy, and the Lorenz-63, Lorenz-96, and MEG datasets. The DPF baseline is described in Appendix D. Code will be made available upon acceptance.

REFERENCES

M.S. 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*, 50(2):174–188, 2002. doi: 10.1109/78.978374.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*, volume 2 of *IMS Collections*, pp. 316–334. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008. doi: 10.1214/193940307000000518.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1517384113.

Itay Buchnik, Guy Revach, Damiano Steger, Ruud J. G. van Sloun, Tirza Routtenberg, and Nir Shlezinger. Latent-kalmannet: Learned kalman filtering for tracking from high-dimensional signals. *IEEE Transactions on Signal Processing*, 72:352–367, 2024. doi: 10.1109/TSP.2023.3344360.

Xiongjie Chen and Yunpeng Li. An overview of differentiable particle filters for data-adaptive sequential Bayesian inference. *Foundations of Data Science*, 7(4):915–943, 2025a.

Xiongjie Chen and Yunpeng Li. Normalizing flow-based differentiable particle filters. *IEEE Transactions on Signal Processing*, 73:493–507, 2025b. doi: 10.1109/TSP.2024.3521338.

Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. Autodifferentiable ensemble kalman filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, 2022.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in Neural Information Processing Systems*, 28, 2015.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pp. 2100–2111. PMLR, 2021.

Jarrad Courts, Adrian G Wills, Thomas B Schön, and Brett Ninness. Variational system identification for nonlinear state-space models. *Automatica*, 147:110687, 2023.

Fred Daum and Jim Huang. Curse of dimensionality and particle filters. In *2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652)*, volume 4, pp. 4_1979–4_1993. IEEE, 2003.

Julie Delon, Agnes Desolneux, and Antoine Salmona. Gromov–wasserstein distances between gaussian distributions. *Journal of Applied Probability*, 59(4):1178–1198, 2022. doi: 10.1017/jpr.2022.16.

Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In Dan Crisan and Boris Rozovskii (eds.), *The Oxford handbook of nonlinear filtering*, Oxford handbooks in mathematics, pp. 656–705. Oxford University Press, Oxford ; N.Y., 2011. ISBN 9780199532902. URL http://webcat.warwick.ac.uk/record=b2490036~S1.

Dimas Abreu Archanjo Dutra. Parameterizations for large-scale variational system identification using unconstrained optimization. *Automatica*, 173:112086, 2025.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems*, 30, 2017.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 240–250, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1021. URL https://aclanthology.org/N19-1021/.

Anubhab Ghosh, Antoine Honoré, and Saikat Chatterjee. DANSE: Data-Driven Non-Linear State Estimation of Model-Free Process in Unsupervised Learning Setup. *IEEE Transactions on Signal Processing*, 72:1824–1838, 2024. doi: 10.1109/TSP.2024.3383277.

Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends in Machine Learning*, 15(1-2):1–175, 2021.

Çağlar Hızlı, Çağatay Yıldız, Matthias Bethge, ST John, and Pekka Marttinen. Identifying latent state transitions in non-linear dynamical systems. In *International Conference on Learning Representations*, pp. 71286–71315, 2025.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. *Robotics: Science and Systems XIV*, 2018.

Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Conference on robot learning*, pp. 169–178. PMLR, 2018.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyTqHL5xg.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, 45:561–578, 2021. doi: https://doi.org/10.1007/s10514-021-09990-9.

R. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

E.N. Lorenz. *Predictability: a problem partly solved*. PhD thesis, Shinfield Park, Reading, 1995 1995.

Hiroshi Morioka, Hermanni Hälvä, and Aapo Hyvarinen. Independent innovation analysis for nonlinear vector autoregressive process. In *International Conference on Artificial Intelligence and Statistics*, pp. 1549–1557. PMLR, 2021.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International conference on artificial intelligence and statistics*, pp. 968–977. PMLR, 2018.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022. doi: 10.1109/TSP.2022.3158588.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems*, 32, 2019.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning*, pp. 19388–19405. PMLR, 2022.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Thomas B Schön, Fredrik Lindsten, Johan Dahlin, Johan Wågberg, Christian A Naesseth, Andreas Svensson, and Liang Dai. Sequential Monte Carlo methods for system identification. *IFAC-PapersOnLine*, 48(28):775–786, 2015.

Nir Shlezinger, Guy Revach, Anubhab Ghosh, Saikat Chatterjee, Shuo Tang, Tales Imbiriba, Jindrich Dunik, Ondrej Straka, Pau Closas, and Yonina C Eldar. Artificial intelligence-aided kalman filters: Ai-augmented designs for kalman-type algorithms. *IEEE Signal Processing Magazine*, 42 (3):52–76, 2025.

Xiangchen Song, Weiran Yao, Yewen Fan, Xinshuai Dong, Guangyi Chen, Juan Carlos Niebles, Eric Xing, and Kun Zhang. Temporally disentangled representation learning under unknown nonstationarity. *Advances in Neural Information Processing Systems*, 36:8092–8113, 2023.

Wessel L. van Nierop, Nir Shlezinger, and Ruud J.G. van Sloun. Deep variational sequential monte carlo for high-dimensional observations. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889044.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Weiran Yao, Yuewen Sun, Alex Ho, Changyin Sun, and Kun Zhang. Learning temporally causal latent processes from general temporal data. In *International Conference on Learning Representations*, 2022.

Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. ODE2VAE: Deep generative second order odes with bayesian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

A Derivation of the DVSI Lower Bounds

Using the factorization in (3), the ELBO in (8) is

$$\mathcal{L}_{\mathrm{DVSI}} = \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\theta}}} \left[\log p(\mathbf{y}_t|\mathbf{x}_t)\right] - \mathbb{E}_{q_{\boldsymbol{\theta}}} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})}{p(\mathbf{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})}\right], \tag{17}$$

where $\mathbb{E}_{q_{\boldsymbol{\theta}}}$ denotes the expectation with respect to $q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$. The two DVSI lower bounds follow by substituting the two posterior families into (17).

Autoregressive posterior (DVSI-I and II): Substituting (11) into the second term of (17) gives

$$\mathbb{E}_{q_{\boldsymbol{\theta}}^{\mathrm{AR}}} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)}{p(\mathbf{x}_1)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q_{\boldsymbol{\theta}}^{\mathrm{AR}}} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)}{p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})}\right]. \tag{18}$$

The tth term of the sum depends only on $(\mathbf{x}_{t-1}, \mathbf{x}_t)$. The conditional factors of $\mathbf{x}_{t+1:T}$ in (11) integrate to one, and integrating out $\mathbf{x}_{1:t-2}$ gives the marginal $\bar{q}^{\mathrm{AR}}_{\boldsymbol{\theta},t-1}$ of $\mathbf{x}_{t-1}$. Hence,

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\theta}}^{\mathrm{AR}}} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)}{p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] &= \mathbb{E}_{\bar{q}^{\mathrm{AR}}_{\boldsymbol{\theta},t-1}} \left[\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)}{p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})}\right]\right] \\ &= \mathbb{E}_{\bar{q}^{\mathrm{AR}}_{\boldsymbol{\theta},t-1}} \left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{c}_t)\|p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\right)\right]. \end{aligned} \tag{19}$$

Similarly, the first term of (18) is $D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)\|p(\mathbf{x}_1))$. Inserting both into (17) gives $\mathcal{L}_{\mathrm{AR}}$ in (12).

Factorized posterior (DVSI-III and IV): Under (13), $\mathbf{x}_t$ depends only on $\mathbf{c}_t$ and is independent of $\mathbf{x}_{t-1}$. Hence, $\mathbb{E}_{q_{\boldsymbol{\theta}}^{\mathrm{MF}}}[\log p(\mathbf{y}_t|\mathbf{x}_t)] = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t)}[\log p(\mathbf{y}_t|\mathbf{x}_t)]$, and

$$\mathbb{E}_{q_{\boldsymbol{\theta}}^{\mathrm{MF}}} \left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t)}{p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{c}_{t-1})} \left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}_t)\|p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\right)\right], \quad t \geq 2. \tag{20}$$

Inserting this and $D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{x}_1|\mathbf{c}_1)\|p(\mathbf{x}_1))$ into (17) gives $\mathcal{L}_{\mathrm{MF}}$ in (14).

B DVSI Algorithm

Algorithm 1 summarizes the resulting training procedure. The same procedure applies to all four realizations, with the selected posterior family determining whether state inference is autoregressive or factorized and whether the measurement encoder operates in smoothing or filtering mode.

Algorithm 1 Dynamical Variational System Identification

Require: Measurement trajectories $\mathcal{D}$; known measurement likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$; variational model $q_{\boldsymbol{\theta}}$; transition model $p_{\boldsymbol{\phi}}$.

1: Initialize $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.
2: **repeat**
3: Sample a mini-batch of measurement sequences $\mathbf{y}_{1:T}$ from $\mathcal{D}$.
4: Sample $\mathbf{x}_{1:T}$ using (16); sample sequentially for DVSI-I/II and in parallel for DVSI-III/IV.
5: Evaluate the measurement-consistency and dynamical-consistency terms using (12) for DVSI-I/II or (14) for DVSI-III/IV.
6: Update $(\boldsymbol{\theta}, \boldsymbol{\phi})$ by stochastic gradient ascent on the mini-batch estimate of $\mathcal{L}_{\mathrm{DVSI}}$.
7: **until** convergence
8: **return** Identified STM $p_{\boldsymbol{\phi}^\star}(\mathbf{x}_t|\mathbf{x}_{t-1})$.

C Further Discussion

Physical state grounding and observability. The known measurement model distinguishes the considered setting from generic latent sequence modeling and ties the inferred variables to a prescribed physical state. It does not, by itself, guarantee unique recovery of that state or its transition

law: the measurement process and dynamics must still provide sufficient observability and excitation, and symmetries in the measurement model may lead to equivalent solutions. In particular, the relation between the measurement dimension n and state dimension m alone does not determine observability. Unknown finite-dimensional parameters of the measurement model, such as σ_w^2, can also be incorporated into (10) and learned jointly. In contrast, replacing the entire known measurement model with an unrestricted learned decoder would return to the latent-coordinate ambiguities characteristic of generic dynamical latent-variable models.

Model structure and variational approximation. Although we focus on a structure-free neural STM, the same variational formulation applies when analytical knowledge of the dynamics is available, by parameterizing only its unknown components. Such structure can reduce the amount of data required to learn the system. Likewise, the diagonal-Gaussian variational family used here is a computationally convenient realization of DVSI, rather than a restriction of the framework itself; richer variational families may be employed when strongly correlated or multimodal state posteriors must be represented.

D On DPF Implementation

The DPF is an end-to-end PF offering greater modeling flexibility than a model-driven PF (Jonschkowski et al., 2018; Corenflos et al., 2021). Similar to the PF, the underlying STM is assumed to be Markovian. DPF involves modeling the unknown STM (and the measurement model, if required) using DNNs. The unknown parameters are then learned in either an unsupervised, supervised, or even a semi-supervised learning framework (Chen & Li, 2025a).

The STM $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ is typically parameterized either as a Gaussian (Jonschkowski et al., 2018; Corenflos et al., 2021) or a Gaussian mixture model (GMM) (van Nierop et al., 2025), with the distribution parameters obtained using a DNN with parameters $\boldsymbol{\phi}$. Along with the STM, for sampling particles at every time step, a proposal distribution $q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ is also parameterized using DNNs. Since the sampling operation is required to be differentiable, a suitable family of distributions is chosen for the proposal distribution, e.g., a multivariate Gaussian or GMM. As in Section 3.1, we assume that the measurement model $p(\mathbf{y}_t|\mathbf{x}_t)$ is known a priori.

The prediction step of DPF propagates the particle approximation of the filtering distribution at time $t-1$, $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx \sum_{j=1}^{J} w_{t-1}^{(j)} \delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{(j)})$, using the current measurement $\mathbf{y}_t$ as

$$\mathbf{x}_t^{(j)} \sim q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}, \mathbf{y}_t), \quad j = 1, \ldots, J. \tag{21}$$

Thereafter, the weights $\{w_{t-1}^{(j)}\}_{j=1}^{J}$ are updated as (van Nierop et al., 2025)

$$\tilde{w}_t^{(j)} = w_{t-1}^{(j)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(j)})\, p_{\boldsymbol{\phi}}(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)})}{q_{\boldsymbol{\theta}}(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)}, \mathbf{y}_t)}, \qquad w_t^{(j)} = \frac{\tilde{w}_t^{(j)}}{\sum_{k=1}^{J} \tilde{w}_t^{(k)}}, \qquad j = 1, \ldots, J. \tag{22}$$

Thereafter, the resampling step is carried out in a differentiable manner, e.g., using the soft-resampling scheme (Karkus et al., 2018) or the optimal transport-based resampling scheme (Corenflos et al., 2021). The posterior mean and covariance are computed as weighted averages of the particles,

$$\mathbf{m}_{t|1:t} = \sum_{j=1}^{J} w_t^{(j)} \mathbf{x}_t^{(j)}, \qquad \mathbf{L}_{t|1:t} = \sum_{j=1}^{J} w_t^{(j)} \big(\mathbf{x}_t^{(j)} - \mathbf{m}_{t|1:t}\big)\big(\mathbf{x}_t^{(j)} - \mathbf{m}_{t|1:t}\big)^\top. \tag{23}$$

For learning the unknown parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, DPF also follows a maximum-likelihood approach. However, since the log-likelihood of the measurements is analytically intractable, a lower bound of the log-likelihood is maximized instead. This was proposed as part of the variational SMC method (Naesseth et al., 2018), where the objective is

$$\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \sum_{i=1}^{N} \mathbb{E}\left[\log \hat{p}_{\boldsymbol{\phi},\boldsymbol{\theta}}\big(\mathbf{y}_{1:T}^{(i)}\big)\right], \qquad \log \hat{p}_{\boldsymbol{\phi},\boldsymbol{\theta}}\big(\mathbf{y}_{1:T}^{(i)}\big) = \sum_{t=1}^{T} \log \sum_{j=1}^{J} \tilde{w}_t^{(j)}, \tag{24}$$

where $\hat{p}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{y}_{1:T}^{(i)})$ is the particle estimate of the likelihood of the ith measurement sequence, and $w_{t-1}^{(j)}$ in (22) are the normalized weights after resampling. In our experiments, DPF uses the optimal transport-based resampling scheme (Corenflos et al., 2021), which is computed by the Sinkhorn algorithm (threshold of 10^{-3} with maximum of 100 iterations, as per (Corenflos et al., 2021)), and $J = 25$ particles.

E Experimental Details

E.1 Performance Evaluation Measures

A synthetic test dataset is given by $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_{t-1}, \mathbf{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\Lambda}_t)\}$ with N_{test} tuples, where $\mathbf{x}_t$ is a realization of the true STM $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Lambda}_t)$, and $t \geq 2$ is a time index of a stochastic trajectory. This means that at the time of testing, we assume that we know the states and the true Markovian STM. For the stochastic Lorenz-63 and Lorenz-96 systems, we simulate 10 trajectories of length 1001. This yields a test dataset of $N_{\text{test}} = 10^4$ tuples, obtained by extracting consecutive states and their corresponding true transition distributions. A real test dataset is given by $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_{t-1}, \mathbf{x}_t)\}$ with N_{test} tuples, since we do not have access to the true STM. For the MEG data, we extract the tuples from the test trajectories of the unseen subjects.

Let $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{t|t-1}, \mathbf{S}_{t|t-1})$ be the learned STM to be evaluated, with $\boldsymbol{\mu}_{t|t-1} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1})$ and $\mathbf{S}_{t|t-1} = \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1})$ as in (1), and let $\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \boldsymbol{\mu}_{t|t-1}$. Sums below run over the N_{test} tuples of $\mathcal{D}_{\text{test}}$. The MSE is given by

$$\text{MSE} = \frac{1}{N_{\text{test}}} \sum \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 = \frac{1}{N_{\text{test}}} \sum \|\mathbf{x}_t - \boldsymbol{\mu}_{t|t-1}\|_2^2. \tag{25}$$

The LL is given by

$$\begin{aligned} \text{LL} &= \frac{1}{N_{\text{test}}} \sum \log \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{t|t-1}, \mathbf{S}_{t|t-1}) \\ &= -\frac{1}{2N_{\text{test}}} \sum \left(m \log(2\pi) + \log|\mathbf{S}_{t|t-1}| + (\mathbf{x}_t - \boldsymbol{\mu}_{t|t-1})^\top \mathbf{S}_{t|t-1}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{t|t-1}) \right), \end{aligned} \tag{26}$$

where m is the dimension of $\mathbf{x}_t$. For the Lorenz-96 system and the MEG data, we report the LL per dimension, LL/m.

For the synthetic test dataset, the true distribution of the next state is known, and we compare it to the learned distribution using the Wasserstein 2-distance (Delon et al., 2022). The WD represents the optimal transport cost for transforming one distribution into another. For Gaussian distributions, there is a closed-form expression of the WD (Delon et al., 2022), and we report

$$\text{WD} = \frac{1}{N_{\text{test}}} \sum \left(\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t|t-1}\|_2^2 + \text{tr}\left(\boldsymbol{\Lambda}_t + \mathbf{S}_{t|t-1} - 2\big(\boldsymbol{\Lambda}_t^{1/2} \mathbf{S}_{t|t-1} \boldsymbol{\Lambda}_t^{1/2}\big)^{1/2} \right) \right)^{1/2}. \tag{27}$$

Note that WD is a metric (Delon et al., 2022).

The signal-to-measurement noise ratio (SMNR) is defined as

$$\text{SMNR} = \frac{1}{N} \sum_{i=1}^{N} 10 \log_{10} \left(\frac{\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\|\mathbf{h}(\mathbf{x}_t^{(i)}) - \mathbb{E}\{\mathbf{h}(\mathbf{x}_t^{(i)})\}\|_2^2\}}{n\sigma_w^2} \right). \tag{28}$$

E.2 Training Configuration

Architectures: The variational posterior $q_{\boldsymbol{\theta}}$ consists of a projection layer mapping the input measurements to a higher dimension, which is followed by a transformer encoder layer (Vaswani et al., 2017) and two dense layers, where the last dense layer maps to the mean and diagonal covariance in (15). For the filtering DVSI methods (DVSI-II and IV), a causal mask is applied in the self-attention step. The STM consists of three dense layers, where the first two dense layers have an equal number of nodes, and the last layer maps to the mean and diagonal covariance in (1). Elementwise softplus is used to ensure positive variances. The architectures and training hyperparameters are given in Table 4.

Training procedure: The optimization objectives are (12) for DVSI-I and II, and (14) for DVSI-III and IV. In addition, we clip the gradients of the transformer to a maximum norm of 5. Training is performed for at most 3000 epochs with early stopping, where training is stopped if the validation loss has not decreased for the last 100 epochs. Furthermore, a learning rate scheduler is used, where the learning rate is decreased by 10% every 500 epochs; this procedure was adopted from Ghosh et al. (2024).

Table 4: DVSI architecture configurations for different system dimensions. Input/Output denote the input and output dimensions of the networks, where the output contains the mean and the diagonal variance. K is the number of Monte Carlo samples per sequence.

Component	**Lorenz-63**			**Lorenz-96**			**MEG**
	(3D)	**(20D)**	**(40D)**	**(80D)**	**(120D)**	**(200D)**	**(144D)**
Transformer (variational posterior)							
Layers	1	1	1	1	1	1	1
Attention heads	8	8	8	8	8	8	8
d_{model}	64	256	256	512	1024	1024	1024
FFN dimension	32	128	128	256	512	512	512
Input/Output	64/6	20/40	40/80	80/160	120/240	200/400	144/288
Feedforward DNN (STM)							
Hidden layers	[32, 32]	[128, 128]	[128, 128]	[256, 256]	[512, 512]	[512, 512]	[512, 512]
Input/Output	3/6	20/40	40/80	80/160	120/240	200/400	144/288
Training hyperparameters							
Batch size	64	64	64	64	64	64	64
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001	0.001
K (samples)	1	1	1	1	1	1	1

KL annealing: To avoid posterior collapse, where the variational posterior collapses to an uninformative posterior matching the STM (the prior) almost perfectly (Fu et al., 2019), we introduce a KL annealing factor α. For each t, the terms of the ELBO are weighted as

$$\mathbb{E}_{q_{\boldsymbol{\theta}}}[\log p(\mathbf{y}_t|\mathbf{x}_t)] - \alpha\, \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\mathrm{D}_{\mathrm{KL}}\big(q_{\boldsymbol{\theta}}(\mathbf{x}_t|\cdot) \,\|\, p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})\big)\right]. \tag{29}$$

For the synthetic datasets, α is increased linearly over 10 epochs from 0 to 1, and then kept at 1 for the rest of the training; α is updated for every mini-batch. For the MEG dataset that consists of real recordings, linearly increasing α does not suffice to avoid posterior collapse. Instead, we employ a cyclical annealing strategy (Fu et al., 2019) with a cycle length of 50 epochs. Let $e_{\text{norm}} \in [0, 1]$ denote the normalized epoch position within a cycle (e.g., $e_{\text{norm}} = 0.5$ at 25 epochs into a cycle). Then, the KL annealing factor is given by

$$\alpha = 0.1 + 0.4\,\frac{\exp(3e_{\text{norm}}) - 1}{\exp(3) - 1}. \tag{30}$$

Since $\frac{\exp(3e_{\text{norm}})-1}{\exp(3)-1}$ increases from 0 to 1 on $e_{\text{norm}} \in [0, 1]$, α increases exponentially from 0.1 to 0.5 over each cycle of 50 epochs. The parameters herein were found with a small grid search.

E.3 Datasets

Lorenz-63 system: In discretized form, the stochastic Lorenz-63 system can be written as

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t)\mathbf{x}_t + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \sigma_e^2\mathbf{I}), \quad \mathbf{F}(\mathbf{x}_t) = \exp\left[\begin{pmatrix} -10 & 10 & 0 \\ 28 & -1 & -x_{t,1} \\ 0 & x_{t,1} & -8/3 \end{pmatrix} \Delta\tau\right], \tag{31}$$

where $x_{t,j}$ is the jth element of $\mathbf{x}_t$, $\exp[\cdot]$ is the matrix exponential, $\Delta\tau$ is the sampling interval, and σ_e^2 is the variance of the process noise $\mathbf{e}_t$. We set $\Delta\tau = 0.02$ and $\sigma_e^2 = -10$ dB, following Revach et al. (2022); Ghosh et al. (2024). The 2D camera model (Buchnik et al., 2024; van Nierop et al., 2025) uses a Gaussian point spread function over a fixed grid of $8 \times 8 = 64$ pixels, with grid points $\{\mathbf{g}_i\}_{i=1}^{64}$ uniformly distributed in the rectangle from $(-30, -40)$ to $(30, 40)$. The state is projected

onto the camera plane according to its first two coordinates, and the third coordinate is perceived as depth. The measurement function is $\mathbf{h}(\mathbf{x}_t) = (h_{t,1}, \ldots, h_{t,64})^\top$ with

$$h_{t,i} = 10 \exp\left(-\frac{2}{x_{t,3}} \left\| \mathbf{g}_i - \begin{pmatrix} x_{t,1} \\ x_{t,2} \end{pmatrix} \right\|_2^2 \right), \tag{32}$$

where $h_{t,i}$ is the intensity of the measurement at the ith pixel, and the measurements $\mathbf{y}_t$ follow (2).

Lorenz-96 system: The Lorenz-96 system (Lorenz, 1995) is given by the system of differential equations

$$\frac{\mathrm{d}x_j(\tau)}{\mathrm{d}\tau} = \big(x_{j+1}(\tau) - x_{j-2}(\tau)\big)x_{j-1}(\tau) - x_j(\tau) + F_j(\tau), \qquad j = 1, \ldots, m, \tag{33}$$

where $x_j(\tau)$ is the jth state variable at the continuous time τ, $F_j(\tau)$ is the forcing, and $x_0 = x_m$, $x_{-1} = x_{m-1}$, and $x_{m+1} = x_1$. We simulate discretized Lorenz-96 trajectories using a 4th order Runge-Kutta method with step size $\Delta\tau = 0.05$. The system is made stochastic by a random forcing: at each time step, we draw $F_j \sim \mathcal{N}(8, 0.1)$ independently for each j, that is, the standard forcing $F_j = 8$ (Lorenz, 1995) perturbed by Gaussian noise. The measurement function is $\mathbf{h}(\mathbf{x}_t) = \sin(\mathbf{x}_t/10) + \mathbf{x}_t/10$, applied elementwise, with $n = m$.

MEG data: The FaceDetection dataset (Bagnall et al., 2018) originates from a Kaggle competition for decoding human brain activity. It consists of MEG recordings from 16 subjects performing a visual discrimination task, where subjects were presented with either images of faces or scrambled faces. Each trial of the original recordings comprises 1.5 seconds of MEG recording, starting 0.5 seconds before stimulus onset, downsampled to 250 Hz and high-pass filtered at 1 Hz. The FaceDetection dataset contains 144 MEG channels with 62 time points per trial, and approximately 580–590 trials per subject. The dataset has a predefined split into training (10 subjects) and testing (the remaining 6 subjects), without overlap of subjects.

F STM VISUALIZATION

Figure 2 shows trajectories of the Lorenz-63 system simulated with the true STM and with the STMs learned by the four DVSI methods and DPF (panel PF).

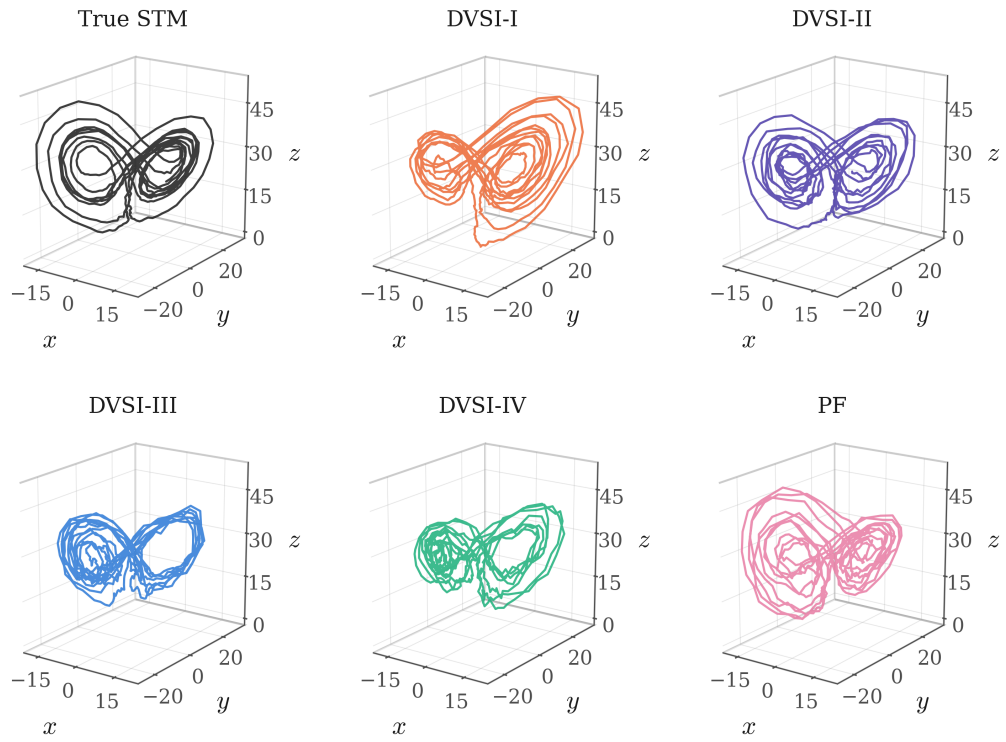


Figure 2: Lorenz-63 trajectories simulated with the true STM and with the learned STMs.

G State Estimation Performance

Although the objective of DVSI is system identification, the variational posterior $q_{\boldsymbol{\theta}}$ also provides state estimates. Tables 5 and 6 show the state estimation performance of the DVSI methods in terms of the normalized MSE, $\text{NMSE} = 10\log_{10}\left(\sum_t \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 / \sum_t \|\mathbf{x}_t\|_2^2\right)$, where $\hat{\mathbf{x}}_t$ is the state estimate from $q_{\boldsymbol{\theta}}$, at 10 dB and 0 dB SMNR, respectively. The smoothing DVSI methods (DVSI-I and III) provide the best state estimates, as they use the complete measurement sequence.

Table 5: State estimation performance using NMSE (dB) at SMNR = 10 dB. Mean ± std over 10 runs with different seeds. Best results in **bold**, second-best underlined. For Lorenz-96, the state dimension m is given.

		Lorenz-96				
Method	Lorenz-63	$m = 20$	$m = 40$	$m = 80$	$m = 120$	$m = 200$
DVSI-I	$-30.51_{\pm 1.07}$	$-18.07_{\pm 0.12}$	$-17.62_{\pm 0.17}$	$-16.83_{\pm 0.18}$	$-16.28_{\pm 0.11}$	$-15.80_{\pm 0.15}$
DVSI-II	$-26.90_{\pm 0.73}$	$-16.82_{\pm 0.05}$	$-16.01_{\pm 0.07}$	$-15.29_{\pm 0.09}$	$-14.66_{\pm 0.08}$	$-8.97_{\pm 0.04}$
DVSI-III	$\mathbf{-30.69}_{\pm 0.71}$	$\mathbf{-18.41}_{\pm 0.05}$	$\mathbf{-18.18}_{\pm 0.08}$	$\mathbf{-17.57}_{\pm 0.14}$	$\mathbf{-16.87}_{\pm 0.14}$	$\mathbf{-16.41}_{\pm 0.08}$
DVSI-IV	$-28.51_{\pm 0.63}$	$-16.93_{\pm 0.04}$	$-16.28_{\pm 0.07}$	$-15.47_{\pm 0.05}$	$-14.93_{\pm 0.08}$	$-9.09_{\pm 0.02}$

Table 6: State estimation performance using NMSE (dB) at SMNR = 0 dB. Mean ± std over 10 runs with different seeds. Best results in **bold**, second-best underlined. For Lorenz-96, the state dimension m is given.

		Lorenz-96				
Method	Lorenz-63	$m = 20$	$m = 40$	$m = 80$	$m = 120$	$m = 200$
DVSI-I	$\mathbf{-26.07}_{\pm 0.58}$	$\mathbf{-9.54}_{\pm 0.18}$	$\mathbf{-9.42}_{\pm 0.07}$	$\mathbf{-9.00}_{\pm 0.24}$	$\mathbf{-8.47}_{\pm 0.10}$	$\mathbf{-8.30}_{\pm 0.13}$
DVSI-II	$-23.68_{\pm 0.56}$	$-8.41_{\pm 0.05}$	$-8.00_{\pm 0.02}$	$-7.61_{\pm 0.03}$	$-7.25_{\pm 0.07}$	$-5.10_{\pm 0.11}$
DVSI-III	$\underline{-25.79}_{\pm 0.13}$	$\underline{-9.05}_{\pm 0.04}$	$\underline{-9.03}_{\pm 0.05}$	$\underline{-8.72}_{\pm 0.07}$	$\underline{-8.12}_{\pm 0.32}$	$\underline{-7.99}_{\pm 0.16}$
DVSI-IV	$-24.48_{\pm 0.17}$	$-8.01_{\pm 0.03}$	$-7.73_{\pm 0.04}$	$-7.42_{\pm 0.03}$	$-7.12_{\pm 0.07}$	$-5.13_{\pm 0.02}$

H 5-STEP MSE PERFORMANCE

To evaluate the learned STM beyond one-step transitions, Tables 7 and 8 show the 5-step MSE, i.e., the MSE of predicting $\mathbf{x}_{t+5}$ from $\mathbf{x}_t$ using the learned STM, at 10 dB and 0 dB SMNR, respectively. At 10 dB SMNR, the smoothing DVSI methods (DVSI-I and III) provide the lowest 5-step MSE. At 0 dB SMNR, DVSI-I provides the lowest 5-step MSE for all systems.

Table 7: Performance of the learned STM using 5-step MSE at SMNR = 10 dB. Mean ± std over 10 runs with different seeds. Best results in **bold**, second-best underlined. For Lorenz-96, the state dimension m is given.

		Lorenz-96				
Method	Lorenz-63	$m = 20$	$m = 40$	$m = 80$	$m = 120$	$m = 200$
DVSI-I	$\mathbf{0.68}_{\pm 0.05}$	$\mathbf{0.83}_{\pm 0.14}$	$\underline{1.35}_{\pm 0.20}$	$\underline{2.12}_{\pm 0.22}$	$\underline{2.78}_{\pm 0.28}$	$\underline{5.13}_{\pm 1.04}$
DVSI-II	$0.93_{\pm 0.09}$	$1.16_{\pm 0.15}$	$3.64_{\pm 0.22}$	$6.98_{\pm 0.23}$	$8.11_{\pm 0.08}$	$8.83_{\pm 0.13}$
DVSI-III	$\underline{0.79}_{\pm 0.06}$	$\underline{0.96}_{\pm 0.07}$	$\mathbf{1.24}_{\pm 0.08}$	$\mathbf{1.96}_{\pm 0.26}$	$\mathbf{2.53}_{\pm 0.14}$	$\mathbf{3.73}_{\pm 0.39}$
DVSI-IV	$1.00_{\pm 0.17}$	$1.65_{\pm 0.17}$	$3.22_{\pm 0.33}$	$6.88_{\pm 0.12}$	$7.92_{\pm 0.13}$	$8.87_{\pm 0.08}$

Table 8: Performance of the learned STM using 5-step MSE at SMNR = 0 dB. Mean ± std over 10 runs with different seeds. Best results in **bold**, second-best underlined. For Lorenz-96, the state dimension m is given.

		Lorenz-96				
Method	Lorenz-63	$m = 20$	$m = 40$	$m = 80$	$m = 120$	$m = 200$
DVSI-I	$\mathbf{0.78}_{\pm 0.11}$	$\mathbf{3.25}_{\pm 0.42}$	$\mathbf{3.79}_{\pm 0.30}$	$\mathbf{5.07}_{\pm 0.75}$	$\mathbf{6.84}_{\pm 0.31}$	$\mathbf{7.82}_{\pm 0.54}$
DVSI-II	$\underline{1.10}_{\pm 0.14}$	$\underline{6.59}_{\pm 0.40}$	$8.60_{\pm 0.33}$	$\underline{10.31}_{\pm 0.26}$	$\underline{10.82}_{\pm 0.19}$	$\underline{10.64}_{\pm 0.18}$
DVSI-III	$1.72_{\pm 0.20}$	$7.62_{\pm 0.37}$	$\underline{7.97}_{\pm 0.19}$	$10.53_{\pm 1.48}$	$12.79_{\pm 1.33}$	$12.03_{\pm 0.47}$
DVSI-IV	$2.29_{\pm 0.23}$	$9.29_{\pm 0.46}$	$11.03_{\pm 0.57}$	$12.36_{\pm 0.21}$	$12.78_{\pm 0.40}$	$13.21_{\pm 0.12}$

I TRAINING TIME

Table 9 shows the training time of the methods on the Lorenz-96 system, as discussed in Section 5.5.

Table 9: Training time on the Lorenz-96 system using an NVIDIA T4 GPU. Time per epoch over 1000 trajectories of length 100 (mean $\pm$ std over 3 runs); total time until the maximum number of epochs or early stopping. $\times$: did not converge. Best results in **bold**.

	Lorenz-96 ($m = 20$)		Lorenz-96 ($m = 40$)		Lorenz-96 ($m = 120$)	
Method	Time/epoch (s)	Total (h)	Time/epoch (s)	Total (h)	Time/epoch (s)	Total (h)
DVSI-I	$1.20_{\pm 0.10}$	$0.54_{\pm 0.13}$	$1.15_{\pm 0.07}$	$0.48_{\pm 0.11}$	$2.09_{\pm 0.05}$	$0.56_{\pm 0.12}$
DVSI-II	$1.04_{\pm 0.01}$	$0.30_{\pm 0.05}$	$1.07_{\pm 0.02}$	$0.49_{\pm 0.11}$	$1.21_{\pm 0.04}$	$0.63_{\pm 0.13}$
DVSI-III	$\mathbf{0.14}_{\pm 0.01}$	$0.07_{\pm 0.01}$	$0.91_{\pm 0.21}$	$0.48_{\pm 0.12}$	$1.25_{\pm 0.02}$	$0.46_{\pm 0.05}$
DVSI-IV	$0.15_{\pm 0.01}$	$\mathbf{0.04}_{\pm 0.01}$	$\mathbf{0.16}_{\pm 0.01}$	$\mathbf{0.09}_{\pm 0.02}$	$\mathbf{0.45}_{\pm 0.01}$	$\mathbf{0.30}_{\pm 0.05}$
DPF	$35.92_{\pm 0.12}$	$1.15_{\pm 0.23}$	$\times$	$\times$	$\times$	$\times$