Self-Organizing Score-based Data Assimilation

Yuma Yamaoka, Seiichi Uchida, Shoji Toyota
Department of Advanced Information Technology, Kyushu University,
Fukuoka, Japan
{yuma.yamaoka@human., uchida@, toyota@}ait.kyushu-u.ac.jp

***Abstract*—A state-space model is a statistical framework for inferring latent states from observed time-series data. However, inference with nonlinear and high-dimensional state-space models remains challenging. To this end, an approach based on diffusion models—a powerful class of deep generative models—has been developed, known as *Score-based Data Assimilation* (SDA) [1]. However, SDA cannot be directly applied when the latent-state transition depends on unknown parameters that must be inferred jointly with the latent states. To overcome this limitation, we propose a framework that enables SDA to handle latent states with unknown parameters. A key feature of the proposed method is the incorporation of the self-organization technique [2], which has been used in classical state-space modeling for the joint estimation of latent states and parameters. By integrating this classical technique into modern SDA, our method enables joint inference of latent states and unknown parameters while maintaining the high training efficiency of SDA. The effectiveness of the proposed approach is validated through numerical experiments on dynamical systems arising in neuroscience and atmospheric science. In addition, its scalability is demonstrated using a high-dimensional Kolmogorov flow, with the data dimension on the order of several hundred thousand.**

***Index Terms*—Data Assimilation, State-Space Modeling, Diffusion Model, Numerical Simulation, Differential Equations**

I. Introduction

Given a time series of observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$, recovering the underlying latent states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ is a central inference problem in many scientific and engineering applications. For example, in vehicle navigation, one seeks to infer the vehicle trajectory $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ from noisy observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$ such as signals obtained from GPS sensors [3].

State-space models provide a fundamental framework for addressing this problem. This model consists of two components: (i) a latent state process that describes the temporal evolution of unobserved states $\boldsymbol{x}_t$ (illustrated by the transition chain $\cdots \to \boldsymbol{x}_{t-1} \to \boldsymbol{x}_t \to \boldsymbol{x}_{t+1} \to \cdots$ in the upper left of Fig. 1), and (ii) an observation process that maps each latent state $\boldsymbol{x}_t$ to an observable measurement $\boldsymbol{y}_t$ (illustrated by $\boldsymbol{x}_t \to \boldsymbol{y}_t$ in the same figure). By regarding a target system as the state-space model, one can infer the latent states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ from noisy observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$.

Nevertheless, the applicability of existing state-space modeling approaches remains limited. Classical (Extended and Ensemble) Kalman filters [4]–[6] fundamentally rely on linear-Gaussian assumptions in the latent state transition $\boldsymbol{x}_t \to \boldsymbol{x}_{t+1}$ and the observation model $\boldsymbol{x}_t \to \boldsymbol{y}_t$, which restricts their applicability to systems with complex dynamics. While particle filters [7] alleviate these modeling assumptions, their performance is known to degrade severely in high-dimensional systems.

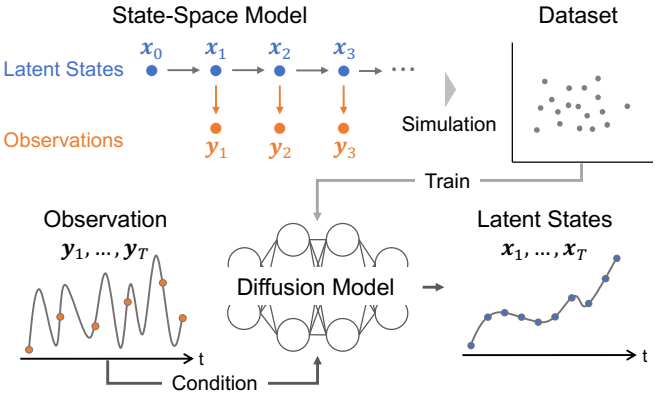

Fig. 1. Overview of the training and inference procedure in Score-based Data Assimilation (SDA) [1]. We first simulate samples from a given state-space model. These samples are used as a training dataset to train a diffusion model for sampling latent states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ conditioned on a given time-series observation $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$.

To address this challenge, Rozet et al. [1] proposed *Score-based Data Assimilation* (SDA), a latent-state inference method for state-space models with high-dimensional and complex transition dynamics, leveraging diffusion models [8]. Diffusion models are a powerful class of deep generative models demonstrating remarkable success in generating and sampling extremely high-dimensional data, such as images and videos. SDA applies diffusion models to latent-state inference in state-space models. Specifically, the method trains a diffusion model on simulated data from the state-space model, and uses this model to sample latent states that are consistent with given observations (Fig. 1). Since this approach relies solely on simulated data from a given state-space model and does not require an explicit form of the latent-state transition, it is applicable to a broad class of state-space models in which assumptions such as linearity and Gaussianity do not hold. Moreover, by inheriting the ability of diffusion models to handle high-dimensional data, SDA naturally scales to high-dimensional state-space models.

However, as mentioned in the original paper (Section 5 in [1]), SDA cannot be applied when the latent-state transition depends on unknown parameters. Many state-space models involve such unknown parameters in their latent-state dynamics. A representative example is a state-space model in which the latent state evolves according to a differential equation, $\frac{d\boldsymbol{x}(t)}{dt} = f\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big)$. Such systems often involve unknown

parameters $\boldsymbol{\theta}$ that govern the dynamics. The original SDA formulation treats them as fixed and known and thus does not directly support joint inference of states and parameters.

This limitation is problematic in practical data assimilation, where unknown parameters often encode physical constants, operating conditions, and environment-specific properties. Reliable parameter estimates, together with latent states, enable simulator calibration and uncertainty quantification, improve forecasting by correcting transition dynamics, and provide interpretable quantities tied to domain knowledge [9]–[11].

In this study, we propose a framework that enables SDA to handle latent states with unknown parameters. A key ingredient of the proposed framework is the incorporation of a self-organization method [2], which has been primarily utilized in classical inference methods for state-space models, into SDA. By combining this classical technique with a modern SDA framework, it becomes possible to jointly infer latent states and parameters while maintaining the high training efficiency of SDA. We demonstrate the effectiveness of the proposed approach through the FitzHugh–Nagumo model, the Lorenz-63 system, and a high-dimensional Kolmogorov flow.

II. Background

A. State-Space Modeling

We formally describe the problem setting of state-space modeling. Let $\boldsymbol{x}_{1:T} := (\boldsymbol{x}_1, \dots, \boldsymbol{x}_T) \in \mathbb{R}^{d_{\mathcal{X}} \times T}$ denote a trajectory of latent states evolving according to discrete-time stochastic dynamics, $\boldsymbol{x}_{t+1} \sim p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$. In state-space modeling, observations are also given as a time series, $\boldsymbol{y}_{1:T} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_T) \in \mathbb{R}^{d_{\mathcal{Y}} \times T}$, assumed to be generated from a stochastic observation model, $\boldsymbol{y}_t \sim p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$. The goal of state-space modeling is to sample a latent trajectory $\boldsymbol{x}_{1:T}$ from the posterior distribution given observations $\boldsymbol{y}_{1:T}$. Specifically, we aim to obtain a sample from the posterior

$$p(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}) = \frac{\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\, p(\boldsymbol{x}_{1:T})}{p(\boldsymbol{y}_{1:T})}, \tag{1}$$

which represents the distribution over latent state trajectories conditioned on the observed data.

The Kalman filter and particle filters are well-established methods in state-space modeling, enabling the sampling of latent trajectories from the posterior distribution (1). However, they are known to be difficult to scale to high-dimensional and nonlinear systems. The Kalman filter and its extensions (e.g., the extended and ensemble Kalman filters) [4]–[6] rely on linearization or Gaussian assumptions. Particle filters [7] provide a more general framework, but they often suffer from severe performance degradation in high-dimensional settings.

B. Diffusion Models

Diffusion models have emerged as a powerful tool for generating samples from target data distributions. We review the technical background of diffusion models based on the formulation in [8].

a) Technical Background of Diffusion Models: Diffusion models consist of a forward diffusion process that gradually adds noise to data and a reverse diffusion process that gradually removes noise in order to recover the data distribution. In the forward diffusion process, noise is added to the target data $\boldsymbol{x} = \boldsymbol{x}^0$ from timestep $a = 0$ to $a = A$[1] according to the stochastic differential equation

$$d\boldsymbol{x}^a = f(a)\, \boldsymbol{x}^a\, da + g(a)\, dW, \tag{2}$$

where W is a standard Wiener process. The drift and diffusion coefficients $f(a)$ and $g(a)$ are chosen such that the SDE (2) transforms any initial distribution into a normal distribution $\mathcal{N}(\mu_A, I_A)$ at $a = A$. Sampling from the target distribution $p(\boldsymbol{x}) = p(\boldsymbol{x}^0)$ can then be performed by solving the corresponding reverse-time process [12]

$$d\boldsymbol{x}^a = \left\{ f(a) - g^2(a)\, \nabla_{\boldsymbol{x}^a} \log p_a(\boldsymbol{x}^a) \right\} da + g(a)\, d\bar{W}, \tag{3}$$

backward in time from $a = A$ to $a = 0$. Here, $\bar{W}$ denotes a backward Wiener process. The reverse-time process requires the score function $\nabla_{\boldsymbol{x}^a} \log p_a(\boldsymbol{x}^a)$, whose analytical form is generally intractable. To address this issue, we employ denoising score matching [13] and train a score network $s_\phi(\boldsymbol{x}^a, a)$ using a dataset $\mathcal{D}$ consisting of samples drawn from the target distribution, i.e., $\mathcal{D} \sim p(\boldsymbol{x})$.

b) Conditional Generation: Diffusion models can be readily extended to conditional generation. Given a forward process (2) with an initial distribution $p(\boldsymbol{x} \mid \boldsymbol{y})$, the corresponding reverse process (3) involves the conditional score $\nabla_{\boldsymbol{x}^a} \log p(\boldsymbol{x}^a \mid \boldsymbol{y})$. This conditional score can be estimated in essentially the same manner as described above. During training, we require a dataset drawn from the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$, i.e., $\mathcal{D} \sim p(\boldsymbol{x}, \boldsymbol{y})$.

In particular, when the explicit form of $p(\boldsymbol{y} \mid \boldsymbol{x})$ is specified, conditional generation can be performed by training only the unconditional score function $\nabla_{\boldsymbol{x}^a} \log p(\boldsymbol{x}^a)$. This follows from the identity

$$\nabla_{\boldsymbol{x}^a} \log p(\boldsymbol{x}^a \mid \boldsymbol{y}) = \nabla_{\boldsymbol{x}^a} \log p(\boldsymbol{x}^a) + \nabla_{\boldsymbol{x}^a} \log p(\boldsymbol{y} \mid \boldsymbol{x}^a),$$

together with the fact that the second term can be evaluated using Tweedie's lemma [14] when $p(\boldsymbol{y} \mid \boldsymbol{x})$ is known.

C. State-Space Modeling with Diffusion Models

Score-based Data Assimilation (SDA) [1] is a method for latent state estimation (1) that leverages recent advances in diffusion models. In SDA, we first simulate a dataset from a given state-space model (upper part of Fig. 1). Using this dataset $\mathcal{D}$ as training data, we train a score function $\nabla_{\boldsymbol{x}^a_{1:T}} \log p(\boldsymbol{x}^a_{1:T} \mid \boldsymbol{y}_{1:T})$ to run the reverse diffusion process (3). Finally, given an observation sequence $\boldsymbol{y}_1, \dots, \boldsymbol{y}_T$, we obtain samples from the posterior distribution (1) using the trained score function or diffusion model (lower part of Fig. 1).

[1] Although it is common to use t to denote time in diffusion processes, we use a to distinguish it from the time index of the state-space model.

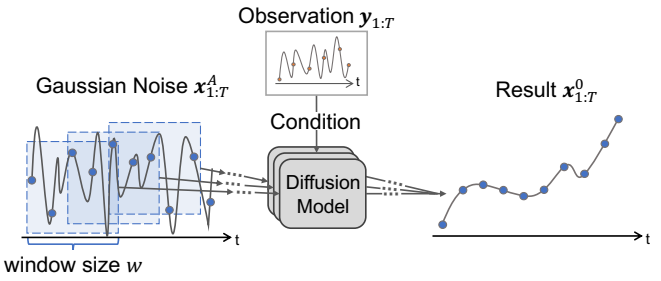

Fig. 2. We train the diffusion model on trajectories that are shorter than observations $\boldsymbol{y}_{1:T}$. We refer to the training trajectory length as the *window size* w. The initial states $\boldsymbol{x}^A_{1:T}$ at $a = A$ of the reverse diffusion process are divided into short trajectories of length w, and each short trajectory is individually fed into the diffusion model. The final inferred latent state $\boldsymbol{x}^0_{1:T}$ is obtained by aggregating the outputs of these short trajectories.

In SDA, we assume that the observation process $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$ follows a normal distribution

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t|\mathcal{A}(\boldsymbol{x}_t), \Sigma), \tag{4}$$

for a given observation operator $\mathcal{A}$, as is standard in data assimilation and state-space modeling [4], [6]. In this setting, we have access to an analytical form of the observation process. Therefore, for conditional generation $p(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T})$, it is sufficient to simulate the latent states $\boldsymbol{x}_{1:T}$ and use them as training data to train the unconditional score $\nabla_{\boldsymbol{x}^a_{1:T}} \log p(\boldsymbol{x}^a_{1:T})$ (Section II.B.b).

A key advantage of SDA is that it does not require assumptions on an explicit model form, as long as simulation from the given state-space model is possible. This is in contrast to the (Extended or Ensemble) Kalman filter, which imposes restrictive assumptions such as linearity and Gaussianity. Moreover, by inheriting strong sampling performance of diffusion models for high dimensional data, SDA is expected to scale well to high-dimensional state-space modeling. This stands in contrast to particle filters [7], which are known to suffer from severe scalability issues in high-dimensional settings.

Another key property of SDA is its ability to reduce the simulation cost required for training diffusion models by leveraging the Markov property of state-space models. Figure 2 illustrates the intuition behind this technique, while we refer the reader to [1] for the mathematical derivation. In a naïve application of SDA, especially when handling observations $\boldsymbol{y}_{1:T}$ with a long time-series length T, costly model simulations over long time horizons are required. Rozet et al. [1] mitigate this issue by exploiting the Markovian structure of the state-space model. Specifically, they show that the target score $\nabla_{\boldsymbol{x}^a_{1:T}} \log p(\boldsymbol{x}^a_{1:T})$ can be factorized into local scores $\nabla_{\boldsymbol{x}^a_{i:i+w}} \log p(\boldsymbol{x}^a_{i:i+w})$, which can be estimated using model simulations over short segments $\boldsymbol{x}_{i:i+w}$ of length w, called the *window size*. This factorization enables efficient sampling for long time-series observations using only short segments of window size $w \ll T$.

However, the current SDA framework is not directly applicable when the state-space model contains unknown parameters. In many cases, state-space models—particularly the latent-state model $p(\boldsymbol{x}_{1:T})$—include parameters such as the initial conditions of the latent dynamics or the coefficients of the underlying differential equations. Jointly estimating both the state-space model parameters and the latent states is a fundamental problem in state-space modeling; however, this remains challenging within existing SDA frameworks. This difficulty in SDA is also discussed in Section 5 of [1].

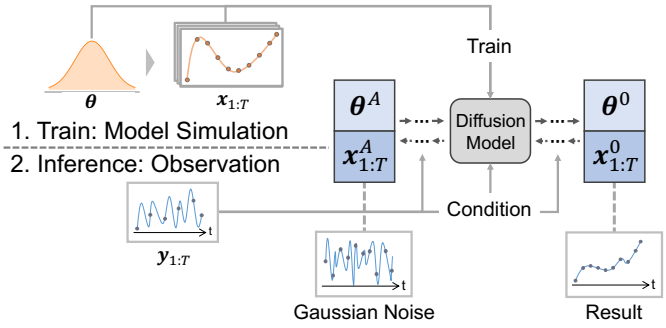

Fig. 3. We simulate the latent states $\boldsymbol{x}_{1:T}$ and the model parameter θ to train the diffusion model. Subsequently, we use the trained model to jointly infer the latent states and model parameter conditioned on the observations $\boldsymbol{y}_{1:T}$.

III. Self-Organizing Score-based Data Assimilation

We propose a method for jointly estimating latent states and model parameters in a state-space model within the framework of Score-based Data Assimilation (SDA).

A. Problem Setting

We consider a state-space model with a state transition density $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{\theta})$, which depends on an unknown parameter $\boldsymbol{\theta}$ governing the system dynamics. The objective of this paper is to infer the joint posterior distribution of the latent states $\boldsymbol{x}_{1:T}$ and the parameter $\boldsymbol{\theta}$ given the observations $\boldsymbol{y}_{1:T}$, namely, obtaining a sample from

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}) = \frac{\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\, p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta})}{p(\boldsymbol{y}_{1:T})}. \tag{5}$$

For simplicity, throughout the study we assume the same Gaussian observation process (4) as in SDA.

B. Joint Estimation of Latent States and Model Parameters via Model Simulation

Building on the learning procedure in SDA, we train a diffusion model to infer latent states and model parameters (5) using simulated data from the target state-space model.

Formally, we first simulate pairs $(\boldsymbol{x}_{1:T}, \boldsymbol{\theta})$ and use them as training data to train a target diffusion model (Fig. 3, training). Here, noting that we assume a Gaussian observation process (4), it suffices to simulate pairs $(\boldsymbol{x}_{1:T}, \boldsymbol{\theta})$ consisting of the latent state and the model parameters for conditional generation. At inference time, conditional generation given real observations is performed using the trained diffusion model, enabling us to infer latent-state trajectories and parameters that are consistent with the observations (Fig. 3, inference).

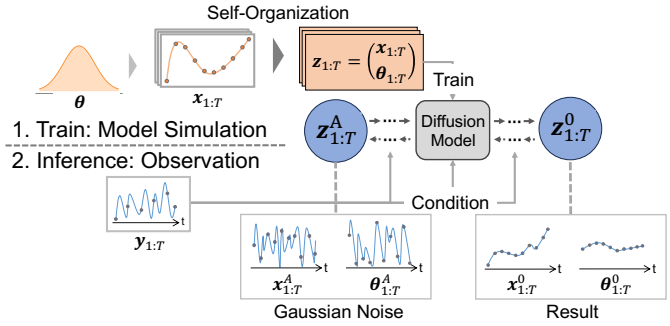

Fig. 4. In contrast to the naive method described in Fig. 3, we simulate the augmented latent states $\boldsymbol{z}_{1:T} = (\boldsymbol{x}_{1:T}, \boldsymbol{\theta}_{1:T})$ to train the diffusion model. As a result, the inferred model parameter is obtained as a trajectory $\boldsymbol{\theta}_{1:T}$ rather than a static parameter $\boldsymbol{\theta}$.

C. Handling Long Time-Series Observations

However, the method described in Fig. 3 requires costly long time-series model simulations to cope with large-scale, long time-series observations. The original SDA addresses this issue by exploiting the Markovian structure of the state-space model (Section II.C and Fig. 2). This factorization relies on the assumption that the generation target $\boldsymbol{x}_{1:T}$ is Markovian. In the present setting, however, the generation target is $(\boldsymbol{x}_{1:T}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is shared across all time steps; therefore, the SDA technique cannot be applied directly. To address this issue, we leverage a self-organization technique [2] developed in the context of classical state-space modeling.

1) Self-Organization in State-Space Modeling: Self-organization [2] is a classical approach, primarily used in particle filtering, for the joint estimation of the latent state $\boldsymbol{x}_t$ and an unknown parameter $\boldsymbol{\theta}$ in state-space models. In this method, the model parameter $\boldsymbol{\theta}$ is extended to a time series $\boldsymbol{\theta}_{1:T}$ and embedded in the latent state $\boldsymbol{x_t}$ at each time step t:

$$\boldsymbol{z}_t = \begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{\theta}_t \end{pmatrix} \tag{6}$$

We treat $\boldsymbol{z}_t$ as a temporally evolving latent state, where $\boldsymbol{\theta}_t$ is either constant or follows a simple random walk, while $\boldsymbol{x}_t$ evolves according to the original latent dynamics $\boldsymbol{x}_t \to \boldsymbol{x}_{t+1}$:

$$\begin{aligned} \boldsymbol{x}_{t+1} &\sim p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{\theta}_t), \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t \quad \big(+\boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})\big). \end{aligned}$$

For the augmented latent state $\boldsymbol{z}_t$, we define the observation process $\boldsymbol{z}_t \to \boldsymbol{y}_t$ as $p(\boldsymbol{y}_t \mid \boldsymbol{z}_t) := p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$, which again yields a state-space model. By applying a latent-state estimation method, such as particle filtering, to this augmented state-space model, one can estimate the augmented latent state $\boldsymbol{z}_t = (\boldsymbol{x}_t, \boldsymbol{\theta}_t)^\top$, thereby obtaining both the model parameters and the latent states aligned with the observations $\boldsymbol{y}_{1:T}$.

2) Incorporating Self-Organization into SDA: By utilizing self-organization, we can extend the simulation cost reduction technique proposed in SDA to our setting of joint inference for the latent states and model parameters $(\boldsymbol{x}_{1:T}, \boldsymbol{\theta})$.

We first augmented latent space $\boldsymbol{x}_t$ to $\boldsymbol{z}_t := (\boldsymbol{x}_t, \boldsymbol{\theta}_t)^\top$ in the manner of self-organization described in the last section (6). We next collect a large number of simulated datasets $\boldsymbol{z}_{1:T} = (\boldsymbol{x}_t, \boldsymbol{\theta}_t)^\top$ (Fig. 4, training). Using these as training data, we train a diffusion model that generates the augmented latent-state trajectory $\boldsymbol{z}_{1:T}$ conditioned on observations, achieving joint inference of parameters and latent states (Fig. 4, inference). As the model parameters are finally obtained in the form of a time series $\boldsymbol{\theta}_{1:T}$, summary statistics of this sequence (e.g., the mean) are used as the final parameter estimates.

Unlike the naive method described in Fig. 3, the approach shown in Fig. 4 considers a Markovian time series, $\cdots \to \boldsymbol{z}_{t-1} \to \boldsymbol{z}_t \to \boldsymbol{z}_{t+1} \to \cdots$, as the generation target. Therefore, we can directly apply the simulation-budget reduction techniques in SDA, relying on the Markov property of the generation target. This enables joint inference of the latent state $\boldsymbol{x}_t$ and the parameter $\boldsymbol{\theta}$ for large-scale time-series observations with fewer simulation budgets, which is the desired outcome. We will examine in the next section how short a simulation length is actually sufficient.

IV. Experiments

We evaluate the proposed method on the FitzHugh–Nagumo [15], [16] and the Lorenz-63 system [17], which are ordinary differential equation models in neuroscience and atmospheric science, respectively. Finally, we demonstrate its scalability using a high-dimensional Kolmogorov flow [18].

A. FitzHugh-Nagumo Model and Lorenz-63 System

We first demonstrate the proposed approach on two ODE models: FitzHugh–Nagumo (FHN) model

$$\begin{aligned} \dot{u} &= u - \frac{u^3}{3} - v + I, \\ \tau\, \dot{v} &= u + a - bv \end{aligned} \tag{7}$$

and Lorenz–63 defined as

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z. \end{aligned} \tag{8}$$

We consider the dynamics $\boldsymbol{x}_t = (u_t, v_t)^\top \in \mathbb{R}^2$ and $\boldsymbol{x}_t = (x_t, y_t, z_t)^\top \in \mathbb{R}^3$, obtained by numerically solving the ODEs (7) and (8) with step sizes of 0.4 (FHN) and 0.05 (Lorenz–63), respectively. By treating them as latent states, we perform inference on both the latent states and the unknown parameters that govern these equations. While ODEs (7) and (8) include three unknown parameters, $\boldsymbol{\theta} = (a, b, \tau)$ and $\boldsymbol{\theta} = (\rho, \sigma, \beta)$, respectively, for simplicity we consider a single unknown model parameter and fix the remaining two. Specifically, for FHN model, we fix $b = 0.2$ and $\tau = 1.0$ and infer a. For the Lorenz–63 system, we fix $\sigma = 10$ and $\beta = 8/3$ and infer ρ. In the observation process, only the first component (u_t for FHN and x_t for Lorenz–63) is observed every 8 time steps, corrupted by Gaussian noise $\mathcal{N}(0, 0.05^2)$.

We generate a dataset of 4096 pairs $(\boldsymbol{x}_{1:T}, \theta)$ by sampling model parameters from the prior ($a \sim \mathrm{Uniform}(0, 1.0)$ for FHN and $\rho \sim \mathcal{N}(28.0, 4.0^2)$ for Lorenz-63) and simulating the system for 1024 steps. We split the dataset into 80% training, 10% validation, and 10% test sets. In our method, we train on

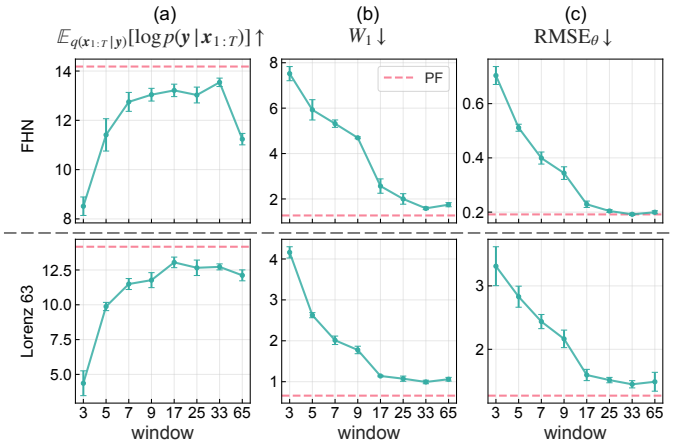

Fig. 5. Evaluation metrics across window sizes $w \in \{3, 5, 7, 9, 17, 25, 33, 65\}$, compared with the particle filter (PF; red dashed line), for FHN (top row) and Lorenz-63 (bottom row). Panels show: (a) expected log-likelihood, (b) Wasserstein distance for the latent states, and (c) RMSE of the model parameter θ.

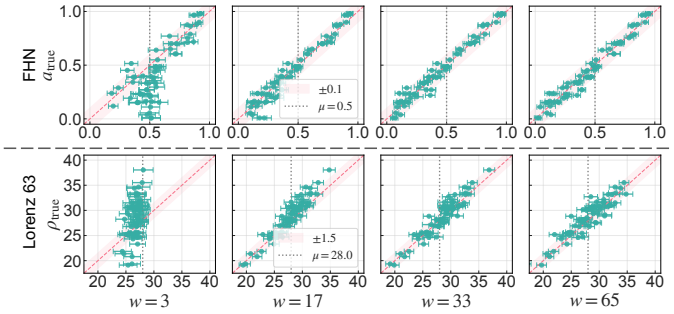

Fig. 6. Comparison between the ground-truth and inferred model parameters across window sizes $w \in \{3, 17, 33, 65\}$, for FHN (top row) and Lorenz-63 (bottom row). The red dashed line represents the ideal estimation ($y = x$), and the vertical dotted line marks the mean of the training distribution ($\mu = 0.5$ for FHN and $\mu = 28.0$ for Lorenz-63).

shorter trajectories than those used during inference; we refer to the training-trajectory length as the window size w (Fig. 2). We evaluate our method using 64 observations generated by applying the observation process to $T = 65$ trajectory segments extracted from test-set trajectories.

Since the latent states of both systems are low-dimensional, we obtain a reliable reference posterior $q(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}_{1:T} \mid \boldsymbol{y}_{1:T})$ using a particle filter (PF) [7]. We then quantitatively evaluate our method by comparing the inferred posterior $p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}_{1:T} \mid \boldsymbol{y}_{1:T})$ with the reference posterior $q(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}_{1:T} \mid \boldsymbol{y}_{1:T})$. We use three metrics: a) the expected log-likelihood $\mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})}[\log p(\boldsymbol{y}_{1:T} \mid \boldsymbol{x}_{1:T})]$; b) the Wasserstein distance $W_1(p, q)$; c) the root mean squared error (RMSE). For the expected log-likelihood, larger is better, whereas for $W_1(p, q)$ and RMSE, lower is better. We also evaluate the performance of model parameter estimation by visualizing the true and inferred model parameters in two dimensions.

Fig. 5 presents the quantitative evaluation of the inferred samples. To show the best performance for each evaluation metric, we also report the results of comparisons between samples drawn from the reference posterior obtained by the particle filter (PF in Fig. 5). Across all metrics, our method achieves performance close to the best performance (PF) for FHN when $w \geq 25$ and for Lorenz-63 when $w \geq 17$. Some results indicate a degradation in performance at $w = 65$, especially for the expected loss in the FHN model; this is likely because increasing the window size makes the problem higher-dimensional and therefore harder to estimate accurately. Fig. 6 visualizes the inferred model parameters and compares them with the corresponding true parameters. For small window sizes (e.g., $w = 3$), the inferred parameters cluster around the mean parameters used for training ($a = 0.5$ for FHN, $\rho = 28.0$ for Lorenz-63); this can be explained by the fact that, with a small window size, sufficient information to infer the parameters could not be obtained from the observations, causing the inferred parameters to be distributed like a prior. Whereas for $w \geq 17$, they concentrate near the ground-truth values. Overall, these results indicate that the proposed framework can correctly infer both latent states and parameters using window sizes $w = 25$ (FHN) or $w = 17$ (Lorenz-63).

B. Kolmogorov Flow

The Kolmogorov flow is governed by the incompressible Navier–Stokes equations:

$$\dot{\mathbf{u}} = -\mathbf{u}\nabla\mathbf{u} + \frac{1}{Re}\nabla^2\mathbf{u} - \frac{1}{\rho_{\text{kol}}}\nabla p + \mathbf{f}, \quad 0 = \nabla \cdot \mathbf{u}, \tag{9}$$

where Re is the Reynolds number, ρ_{kol} the fluid density, p the pressure field, and $\mathbf{f}$ an external forcing. The latent state $\boldsymbol{x}_t$ is defined as a snapshot of the two-dimensional velocity field $\mathbf{u}$ on a 256×256 grid, solved by `jax-cfd` library [19]. We jointly infer $\boldsymbol{x}_t$ and the model parameter $\theta = Re$, while all other parameters are fixed.

We generate a training data of 8192 pairs $(\boldsymbol{x}_{1:T}, Re)$ where $Re \sim \mathrm{Uniform}(10, 200)$. The simulated fields are then downsampled to 64×64 to form the latent states. Each trajectory has length of 64 steps with time step $\Delta = 0.2$. We split the dataset into 80% training, 10% validation, and 10% test sets. We assume that observations are obtained by downsampling the latent state from 64×64 to 8×8 and adding Gaussian noise $\mathcal{N}(0, 0.1^2)$. For evaluation, we extract segments of length $T = 33$ from the test-set trajectories and generate observations by applying the same observation process.

Fig. 7 shows the inferred latent states for observations generated from the ground-truth trajectory $\boldsymbol{x}_{1:T}^{(\text{true})}$. This qualitative result indicates that our method can recover the ground-truth trajectory. Fig. 8 shows the inferred Reynolds numbers compared with the true parameters. The results qualitatively indicate that the proposed method successfully recovers the true parameters. This confirms that the proposed method can successfully infer the true parameters.

For quantitative evaluation, we measure the root mean squared error (RMSE) between the ground truth and the inferred samples for both the latent states (RMSE_x) and the Reynolds number (RMSE_{Re}). We also assess dynamical consistency using the Equation Loss D_{equation} [20]. For all metrics, lower values indicate better performance. Table I summarizes the results across window sizes. We can confirm that smaller windows yield better accuracy. In particular, $w = 5$

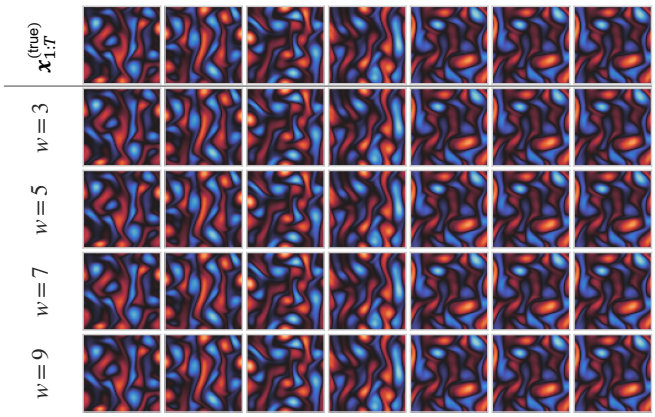

Fig. 7. Visualization of the inferred latent states $\boldsymbol{x}_{1:T}$ for window sizes $w \in \{3, 5, 7, 9\}$, with the ground-truth $\boldsymbol{x}_{1:T}^{(\text{true})}$ (top row). Across all window sizes, our method recovers the ground-truth trajectory.

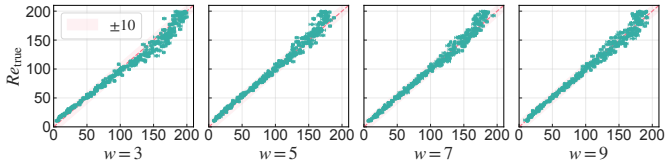

Fig. 8. Comparison of the ground-truth Reynolds number and the inferred Reynolds number across window sizes $w \in \{3, 5, 7, 9\}$. The dashed line represents the identity line ($y = x$).

achieves the best overall performance, attaining the lowest RMSE_x and RMSE_{Re} while keeping D_{equation} competitive (the minimum D_{equation} is achieved at $w = 3$). This tendency may stem from the overwhelmingly high dimensionality of the Kolmogorov flow. Increasing the window size w necessitates a greater amount of simulation data (under the full window setting $w = T$, the data dimension is $64 \times 64 \times 33 = 135{,}168$); consequently, under a fixed simulation budget, a larger window size may result in degraded performance. In contrast, a smaller window size partitions the data into finer segments, thereby increasing the number of available training samples and leading to improved inference performance.

V. Conclusion

We introduced Self-Organizing Score-based Data Assimilation, a method for jointly inferring latent states and parameters for general state-space models, scaling to high-dimensional systems while avoiding strong distributional assumptions. Key ingredient of the proposed method is an incorporation of self-organization technique—originally developed in the context of the particle filter—into the diffusion model, enabling inference on long observation trajectories with high training efficiency. We demonstrated the effectiveness of our approach on three dynamical systems, the FitzHugh–Nagumo, Lorenz-63 systems, and the Kolmogorov flow with the data dimension on the order of several hundred thousand.

Acknowledgment

We used AI tools only for English proofreading; we are responsible for all scientific content and the manuscript. This work was supported by ACT-X-JPMJAX25C, JSPS KAKENHI JP22H05180, JP24K20750 and JP25H01454

TABLE I
Comparison of performance metrics across window sizes w. Values are reported as Means (STDs) over 900 runs.

w	$\mathrm{RMSE}_x \downarrow$	$\mathrm{RMSE}_{Re} \downarrow$	$D_{\text{equation}} \downarrow$
3	0.056 (0.001)	0.144 (0.004)	**4.05** (0.09)
5	**0.052** (0.001)	**0.107** (0.003)	4.16 (0.10)
7	0.054 (0.001)	0.121 (0.003)	4.45 (0.11)
9	0.060 (0.001)	0.134 (0.004)	4.78 (0.12)

References

[1] F. Rozet and G. Louppe, "Score-based data assimilation," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 40 521–40 541.
[2] G. Kitagawa, "A self-organizing state-space model," *Journal of the American Statistical Association*, pp. 1203–1215, 1998.
[3] L. Zhao, W. Y. Ochieng, M. A. Quddus, and R. B. Noland, "An extended kalman filter algorithm for integrating gps and low cost dead reckoning system data for vehicle performance and emissions monitoring," *Journal of Navigation*, vol. 56, p. 257–275, 2003.
[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
[5] G. L. Smith, S. F. Schmidt, and L. A. McGee, *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*, 1962, vol. 135.
[6] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, pp. 10 143–10 162, 1994.
[7] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113, 1993.
[8] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.
[9] P. J. Smith, G. D. Thornhill, S. L. Dance, A. S. Lawless, D. C. Mason, and N. K. Nichols, "Data assimilation for state and parameter estimation: application to morphodynamic modelling," *Quarterly Journal of the Royal Meteorological Society*, vol. 139, pp. 314–327, 2013.
[10] J. J. Ruiz, M. Pulido, and T. Miyoshi, "Estimating model parameters with ensemble-based data assimilation: A review," *Journal of the Meteorological Society of Japan*, vol. 91, pp. 79–99, 2013.
[11] G. Evensen, *Data assimilation: the ensemble Kalman filter*. Springer, 2009.
[12] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, pp. 313–326, 1982.
[13] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching." *Journal of Machine Learning Research*, vol. 6, 2005.
[14] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *International Conference on Learning Representations*, 2023.
[15] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical Journal*, vol. 1, pp. 445–466, 1961.
[16] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, pp. 2061–2070, 1962.
[17] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, pp. 130 – 141, 1963.
[18] G. J. Chandler and R. R. Kerswell, "Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow," *Journal of Fluid Mechanics*, vol. 722, p. 554–595, 2013.
[19] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, "Machine learning–accelerated computational fluid dynamics," *Proceedings of the National Academy of Sciences*, vol. 118, p. e2101784118, 2021.
[20] D. Shu, Z. Li, and A. Barati Farimani, "A physics-informed diffusion model for high-fidelity flow field reconstruction," *Journal of Computational Physics*, vol. 478, p. 111972, 2023.