> this paper：Score-based Data Assimilation　F.Rozet
>
> thanks for Mathpix translate this pdf to markdown.

Score-based Data Assimilation

> Data assimilation, in its most comprehensive form, addresses the Bayesian inverse problem of identifying plausible state trajectories that explain noisy or incomplete observations of stochastic dynamical systems. Various approaches have been proposed to solve this problem, including particle-based and variational methods. However, most algorithms depend on the transition dynamics for inference, which becomes intractable for long time horizons or for high-dimensional systems with complex dynamics, such as oceans or atmospheres. In this work, we introduce score-based data assimilation for trajectory inference. We learn a score-based generative model of state trajectories based on the key insight that the score of an arbitrarily long trajectory can be decomposed into a series of scores over short segments. After training, inference is carried out using the score model, in a nonautoregressive manner by generating all states simultaneously. Quite distinctively, we decouple the observation model from the training procedure and use it only at inference to guide the generative process, which enables a wide range of zero-shot observation scenarios. We present theoretical and empirical evidence supporting the effectiveness of our method.

1. Introduction

Data assimilation (DA) $[1-9]$ is at the core of many scientific domains concerned with the study of complex dynamical systems such as atmospheres, oceans or climates. The purpose of DA is to infer the state of a system evolving over time based on various sources of imperfect information, including sparse, intermittent, and noisy observations.

Formally, let $x_{1:L} = (x_1, x_2, \dots, x_L) \in \mathbb{R}^{L \times D}$ denote a trajectory of states in a discrete-time stochastic dynamical system and $p\left(x_{i+1} \mid x_i\right)$ be the transition dynamics from state x_i to state x_{i+1}. An observation $y \in \mathbb{R}^M$ of the state trajectory $x_{1:L}$ follows an observation process $p\left(y \mid x_{1:L}\right)$, generally formulated as $y = \mathcal{A}\left(x_{1:L}\right) + \eta$, where the measurement function $\mathcal{A} : \mathbb{R}^{L \times D} \mapsto \mathbb{R}^M$ is often non-linear and the observational error $\eta \in \mathbb{R}^M$ is a stochastic additive term that accounts for instrumental noise and systematic uncertainties. In this framework, the goal of DA is to solve the inverse problem of inferring plausible trajectories $x_{1:L}$ given an observation y, that is, to estimate the trajectory posterior

$$p\left(x_{1:L} \mid y\right) = \frac{p\left(y \mid x_{1:L}\right)}{p(y)} p\left(x_1\right) \prod_{i=1}^{L-1} p\left(x_{i+1} \mid x_i\right)$$

where the initial state prior $p\left(x_1\right)$ is commonly referred to as background [5-9]. In geosciences, the amount of data available is generally insufficient to recover the full state of the system from the observation alone [8]. For this reason, the physical model underlying the transition dynamics is of paramount importance to fill in spatial and temporal gaps in the observation.

State-of-the-art approaches to data assimilation are based on variational assimilation [1, 2, 5-7]. Many of these approaches formulate the task as a maximum-a-posteriori (MAP) estimation problem and solve it by maximizing the $\log$-posterior density $\log p\left(x_{1:L} \mid y\right)$ via gradient ascent. Although this approach only produces a point estimate of the trajectory posterior, its cost can already be substantial for problems of the size and complexity of geophysical systems, since it requires differentiating through the physical model. The amount of data that can be assimilated is

therefore restricted because of computational limitations. For example, only a small volume of the available satellite data is exploited for operational forecasts and yet, even with these restrictions, data assimilation accounts for a significant fraction of the computational cost for modern numerical weather prediction [10, 11]. Recent work has shown that deep learning can be used in a variety of ways to improve the computational efficiency of data assimilation, increase the reconstruction performance by estimating unresolved scales after data assimilation, or integrate multiple sources of observations [12-19].

Contributions In this work, we propose a novel approach to data assimilation based on scorebased generative models. Leveraging the Markovian structure of dynamical systems, we train a score network from short segments of trajectories which is then capable of generating physically consistent and arbitrarily-long state trajectories. The observation model is decoupled from the score network and used only during assimilation to guide the generative process, which allows for a wide range of zero-shot observation scenarios. Our approach provides an accurate approximation of the whole trajectory posterior - it is not limited to point estimates - without simulating or differentiating through the physical model. The code for all experiments is made available at https://github.com/francois-rozet/sda.

2. Background

Score-based generative models have recently shown remarkable capabilities, powering many of the latest advances in image, video or audio generation [20-27]. In this section, we review score-based generative models and outline how they can be used for solving inverse problems.

Continuous-time score-based generative models

Adapting the formulation of Song et al. [28], samples $x \in \mathbb{R}^D$ from a distribution $p(x)$ are progressively perturbed through a continuous-time diffusion process expressed as a linear stochastic differential equation (SDE)

$$\mathrm{d}x(t) = f(t)x(t)\mathrm{d}t + g(t)\mathrm{d}w(t)$$

where $f(l) \in \mathbb{R}$ is the drift coefficient, $g(t) \in \mathbb{R}$ is the diffusion coefficient, $w(t) \in \mathbb{R}^D$ denotes a Wiener process (standard Brownian motion) and $x(t) \in \mathbb{R}^D$ is the perturbed sample at time $t \in [0, 1]$. Because the SDE is linear with respect to $x(t)$, the perturbation kernel from x to $x(t)$ is Gaussian and takes the form

$$p(x(t) \mid x) = \mathcal{N}(x(t) \mid \mu(t)x, \Sigma(t))$$

where $\mu(t)$ and $\Sigma(t) = \sigma(t)^2 I$ can be derived analytically from $f(t)$ and $g(t)$[29, 30]. Denoting $p(x(l))$ the marginal distribution of $x(t)$, we impose that $\mu(0) = 1$ and $\sigma(0) \ll 1$, such that $p(x(0)) \approx p(x)$, and we chose the coefficients $f(t)$ and $g(t)$ such that the influence of the initial sample x on the final perturbed sample $x(1)$ is negligible with respect to the noise level - that is, $p(x(1)) \approx \mathcal{N}(0, \Sigma(1))$. The variance exploding (VE) and variance preserving (VP) SDEs [28,31, 32] are widespread examples satisfying these constraints.
Crucially, the time reversal of the forward $\mathrm{SDE}(2)$ is given by a reverse SDE[28, 33]

$$\mathrm{d}x(t) = \left[f(t)x(t) - g(t)^2 \nabla_{x(t)} \log p(x(t))\right]\mathrm{d}t + g(t)\mathrm{d}w(t)$$

That is, we can draw noise samples $x(1) \sim \mathcal{N}(0, \Sigma(1))$ and gradually remove the noise therein to obtain data samples $x(0) \sim p(x(0))$ by simulating the reverse SDE from $t = 1$ to 0 . This requires access to the quantity $\nabla_{x(t)} \log p(x(l))$ known as the score of $p(x(l))$.

Denoising score matching

In practice, the score $\nabla_{x(t)} \log p(x(t))$ is approximated by a neural network $s_\phi(x(t), t)$, named the score network, which is trained to solve the denoising score matching objective [28, 34, 35]

$$\arg\min_\phi \mathbb{E}_{p(x)p(t)p(x(t)|x)} \left[\sigma(t)^2 \left\| s_\phi(x(t), t) - \nabla_{x(t)} \log p(x(t) \mid x) \right\|_2^2 \right]$$

where $p(t) = \mathcal{U}(0, 1)$. The theory of denoising score matching ensures that $s_\phi(x(t), t) \approx \nabla_{x(t)} \log p(x(t))$ for a sufficiently expressive score network. After training, the score network is plugged into the reverse SDE (4), which is then simulated using an appropriate discretization scheme [28, 30, 36, 37].
In practice, the high variance of $\nabla_{x(t)} \log p(x(t) \mid x)$ near $t = 0$ makes the optimization of (5) unstable [30]. To mitigate this issue, a slightly different parameterization $\epsilon_\phi(x(t), t) = -\sigma(t) s_\phi(x(t), t)$ of the score network is often used, which leads to the otherwise equivalent objective $[30, 32, 36]$

$$\arg\min_\phi \mathbb{E}_{p(x)p(t)p(\epsilon)} \left[\left\| \epsilon_\phi(\mu(t)x + \sigma(t)\epsilon, t) - \epsilon \right\|_2^2 \right]$$

where $p(\epsilon) = \mathcal{N}(0, I)$. In the following, we keep the score network notation $s_\phi(x(t), t)$ for convenience, even though we adopt the parameterization $\epsilon_\phi(x(l), t)$ and its objective for our experiments.

Zero-shot inverse problems

With score-based generative models, we can generate samples from the unconditional distribution $p(x(0)) \approx p(x)$. To solve inverse problems, however, we need to sample from the posterior distribution $p(x \mid y)$. This could be accomplished by training a conditional score network $s_\phi(x(t), t \mid y)$ to approximate the posterior score $\nabla_{x(t)} \log p(x(t) \mid y)$ and plugging it into the reverse SDE (4). However, this would require data pairs (x, y) during training and one would need to retrain a new score network each time the observation process $p(y \mid x)$ changes. Instead, many have observed [28,38-41] that the posterior score can be decomposed into two terms thanks to Bayes' rule

$$\nabla_{x(t)} \log p(x(t) \mid y) = \nabla_{x(t)} \log p(x(t)) + \nabla_{x(t)} \log p(y \mid x(t))$$

Since the prior score $\nabla_{x(t)} \log p(x(t))$ can be approximated with a single score network, the remaining task is to estimate the likelihood score $\nabla_{x(t)} \log p(y \mid x(l))$. Assuming a differentiable measurement function $\mathcal{A}$ and a Gaussian observation process $p(y \mid x) = \mathcal{N}(y \mid \mathcal{A}(x), \Sigma_y)$, Chung et al. [41] propose the approximation

$$p(y \mid x(t)) = \int p(y \mid x) p(x \mid x(t)) \mathrm{d}x \approx \mathcal{N}\left(y \mid \mathcal{A}(\hat{x}(x(t))), \Sigma_y\right)$$

where the mean $\hat{x}(x(t)) = \mathbb{E}_{p(x|x(t))}[x]$ is given by Tweedie's formula $[42, 43]$

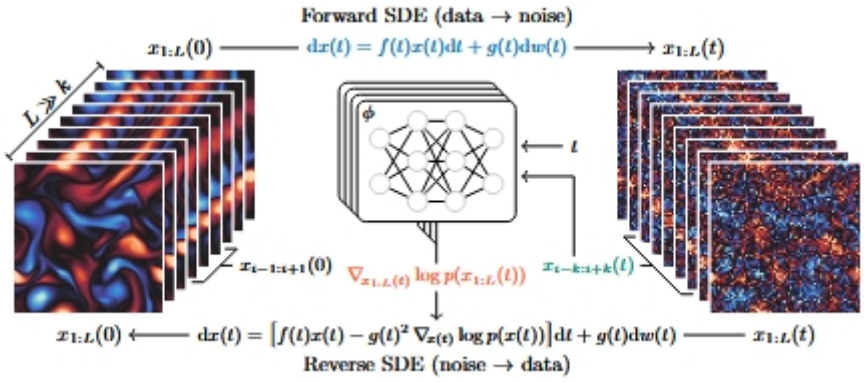
$$\begin{aligned} \mathbb{E}_{p(x|x(t))}[x] &= \frac{x(l) + \sigma(t)^2 \nabla_{x(t)} \log p(x(t))}{\mu(t)} \\ &\approx \frac{x(t) + \sigma(t)^2 s_\phi(x(t), t)}{\mu(t)} \end{aligned}$$

As the log-likelihood of a multivariate Gaussian is known analytically and $s_\phi(x(t), t)$ is differentiable, we can compute the likelihood score $\nabla_{x(t)} \log p(y \mid x(l))$ with this approximation in zero-shot, that is, without training any other network than $s_\phi(x(l), l)$.

3. Score-based data assimilation

Coming back to our initial inference problem, we want to approximate the trajectory posterior $p\left(x_{1:L} \mid y\right)$ of a dynamical system. To do so with score-based generative modeling, we need to estimate the posterior score $\nabla_{x_{1:L}(t)} \log p\left(x_{1:L}(l) \mid y\right)$, which we choose to decompose into prior and likelihood terms, as in (7), to enable a wide range of zero-shot observation scenarios.

In typical data assimilation settings, the high-dimensionality of each state x_i (e.g. the state of atmospheres or oceans) combined with potentially long trajectories would require an impractically large score network $s_\phi\left(x_{1:L}(t), t\right)$ to estimate the prior score $\nabla_{x_{1:L}(t)} \log p\left(x_{1:L}(t)\right)$ and a proportional amount of data for training, which could be prohibitive if data is scarce or if the physical model is expensive to simulate. To overcome this challenge, we leverage the Markovian structure of dynamical systems to approximate the prior score with a series of local scores, which are easier to learn, as explained in Section 3.1. In Section 3.2, we build upon diffusion posterior sampling (DPS) [41] to propose a new approximation for the likelihood score $\nabla_{x_{1:L}(t)} \log p\left(y \mid x_{1:L}(t)\right)$, which we find more appropriate for posterior inference. Finally, in Section 3.3, we describe our sampling procedure inspired from predictor-corrector sampling [28]. Our main contribution, named score-based data assimilation (SDA), is the combination of these three components.



> Figure 1: Trajectories $x_{1:L}$ of a dynamical system are transformed to noise via a diffusion process. Reversing this process generates new trajectories, but requires the score of $p\left(x_{1:L}(l)\right)$. We approximate it by combining the outputs of a score network over subsegments of $x_{1:L}(l)$

3.1 How is your blanket?

Given a set of random variables $x_{1:L} = \{x_1, x_2, \ldots, x_L\}$, it is sometimes possible to find a small Markov blanket $x_{b_i} \subseteq x_{\neq i}$ such that $p\left(x_i \mid x_{\neq i}\right) = p\left(x_i \mid x_{b_i}\right)$ for each element x_i using our knowledge of the set's structure. It follows that each element $\nabla_{x_i} \log p\left(x_{1:L}\right)$ of the full score $\nabla_{x_{1:L}} \log p\left(x_{1:L}\right)$ can be determined locally, that is, only using its blanket;

$$\begin{aligned}\nabla_{x_i} \log p\left(x_{1:L}\right) &= \nabla_{x_i} \log p\left(x_i \mid x_{\neq i}\right) + \nabla_{x_i} \log p\left(x_{\neq i}\right) \\ &= \nabla_{x_i} \log p\left(x_i \mid x_{b_i}\right) + \nabla_{x_i} \log p\left(x_{b_i}\right) = \nabla_{x_i} \log p\left(x_i, x_{b_i}\right)\end{aligned}$$

This property generally does not hold for the diffusion-perturbed set $x_{1:L}(l)$ as there is no guarantee that $x_{b_i}(l)$ is a Markov blanket of the element $x_i(l)$. However, there exists a set of indices $\overline{b}_i \supseteq b_i$ such that

$$\nabla_{x_i(t)} \log p\left(x_{1:L}(t)\right) \approx \nabla_{x_i(t)} \log p\left(x_i(t), x_{\overline{b}_i}(t)\right)$$

is a good approximation for all $t \in [0,1]$. That is, $x_{b_i}(t)$ is a "pseudo" Markov blanket of $x_i(t)$. In the worst case, $\overline{b}_i$ contains all indices except i, but we argue that, for some structures, there is a set $\overline{b}_i$ not much larger than b_i that satisfies (13). Our rationale is that, since we impose the initial noise to be negligible, we know that $x_{b_i}(t)$ becomes indistinguishable from x_{b_i} as t approaches 0 . Furthermore, as l grows and noise accumulates, the mutual information between elements $x_i(l)$ and $x_j(t)$ decreases to finally reach 0 when $t=1$. Hence, even if $\overline{b}_i = b_i$, the pseudo-blanket approximation (13) already holds near $t=0$ and $t=1$. In between, even though the approximation remains unbiased (see Appendix A), the structure of the set becomes decisive. If it is known and present enough regularities/symmetries, (13) could and should be exploited within the architecture of the score network $s_\phi\left(x_{1:L}(\ell), t\right)$.

In the case of dynamical systems, the set $x_{1:L}$ is by definition a first-order Markov chain and the minimal Markov blanket of an element x_i is $x_{b_i} = \{x_{i-1}, x_{i+1}\}$. For the perturbed element $x_i(t)$, the pseudo-blanket $x_{\overline{b}_i}(t)$ can take the form of a window surrounding $x_i(t)$, that is $\overline{b}_i = \{i-k, \ldots, i+k\} \backslash \{i\}$ with $k \geq 1$. The value of k is dependent on the problem, but we argue, supported by our experiments, that it is generally much smaller than the chain's length L. Hence, a fully convolutional neural network (FCNN) with a narrow receptive field is well suited to the task, and any long-range capabilities would be wasted resources. Importantly, if the receptive field is $2k+1$, the network can be trained on segments $x_{i-k:i+k}$ instead of the full chain $x_{1:L}$, thereby drastically reducing training costs. More generally, we can train a local score network (see Algorithm 1)

$$s_\phi\left(x_{i-k:i+k}(t), t\right) \approx \nabla_{x_{i-k:i+k}(t)} \log p\left(x_{i-k:i+k}(t)\right)$$

such that its $k+1$-th element approximates the score of the i-th state $\nabla_{x_i(t)} \log p\left(x_{1:L}(l)\right)$. We also have that the k first elements of $s_\phi\left(x_{1:2k+1}(t), t\right)$ approximate the score of the k first states $\nabla_{x_{1:k}(t)} \log p\left(x_{1:L}(t)\right)$ and the k last elements of $s_\phi\left(x_{L-2k:L}(t), t\right)$ approximate the score of the k last states $\nabla_{x_{L-k:L}(t)} \log p\left(x_{1:L}(t)\right)$. Hence, we can apply the local score network on all sub-segments $x_{i-k:i+k}(t)$ of $x_{1:L}(t)$, similar to a convolution kernel, and combine the outputs (see Algorithm 2) to get an approximation of the full score $\nabla_{x_{1:L}(t)} \log p\left(x_{1:L}(t)\right)$. Note that we can either condition the score network with i or assume the statistical stationarity of the chain, that is $p\left(x_i\right) = p\left(x_{i+1}\right)$.

Algorithm 1 Training $\epsilon_\phi\left(x_{i-k:i+k}(t), t\right)$

for $i = 1$ **to** N **do**
  $x_{1:L} \sim p\left(x_{1:L}\right)$
  $i \sim \mathcal{U}(\{k+1, \ldots, L-k\})$
  $t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0, I)$
  $x_{i-k:i+k}(l) \leftarrow \mu(t) x_{i-k:i+k} + \sigma(l)\epsilon$
  $\ell \leftarrow \left\|\epsilon_\phi\left(x_{i-k:i+k}(t), l\right) - \epsilon\right\|_2^2$
  $\phi \leftarrow \mathrm{GRADIENTDESCENT}\left(\phi, \nabla_\phi \ell\right)$

Algorithm 2 Composing $s_\phi\left(x_{i-k:i+k}(t), t\right)$

function $s_\phi\left(x_{1:L}(t), t\right)$
  $s_{1:k+1} \leftarrow s_\phi\left(x_{1:2k+1}(t), t\right)[:k+1]$
  for $i = k+2$ **to** $L-k-1$ **do**
    $s_i \leftarrow s_\phi\left(x_{i-k:i+k}(t), t\right)[k+1]$
  $s_{L-k:L} \leftarrow s_\phi\left(x_{L-2k:L}(t), t\right)[k+1:]$
  return $s_{1:L}$

3.2 Stable likelihood score

Due to approximation and numerical errors in $\hat{x}(x(l))$, computing the score $\nabla_{x(t)} \log p(y \mid x(l))$ with the likelihood approximation (8) is very unstable, especially in the low signal-to-noise regime, that is when $\sigma(t) \gg \mu(t)$. This incites Chung et al. [41] to replace the covariance Σ_y by the identity I and rescale the likelihood score with respect to $\|y - \mathcal{A}(\hat{x}(x(t)))\|$ to stabilize the sampling process. These modifications introduce a significant error in the approximation as they greatly affect the norm of the likelihood score.

We argue that the instability is due to (8) being only exact if the variance of $p(x \mid x(t))$ is null or negligible, which is not the case when $t > 0$. Instead, Adam et al. [40] and Meng et al. [44] approximate the covariance of $p(x \mid x(t))$ with $\Sigma(t)/\mu(t)^2$, which is valid as long as the prior $p(x)$ is Gaussian with a large diagonal covariance Σ_x. We motivate in Appendix B the more general covariance approximation $\sigma(t)^2/\mu(t)^2\Gamma$, where the matrix Γ depends on the eigendecomposition of Σ_x. Then, taking inspiration from the extended Kalman filter, we approximate the perturbed likelihood as

$$p(y \mid x(t)) \approx \mathcal{N}\left(y \mid \mathcal{A}(\hat{x}(x(t))), \Sigma_y + \frac{\sigma(t)^2}{\mu(t)^2} A\Gamma A^T\right)$$

where $A = \partial_x \mathcal{A}|_{\hat{x}(x(t))}$ is the Jacobian of $\mathcal{A}$. In practice, to simplify the approximation, the term $A\Gamma A^T$ can often be replaced by a constant (diagonal) matrix. We find that computing the likelihood score $\nabla_{x(t)} \log p(y \mid x(t))$ with this new approximation (see Algorithm 3) is stable enough that rescaling it or ignoring Σ_y is unnecessary.

3.3 Predictor-Corrector sampling

To simulate the reverse SDE, we adopt the exponential integrator (EI) discretization scheme introduced by Zhang et al. [30]

$$x(t - \Delta t) \leftarrow \frac{\mu(t - \Delta t)}{\mu(t)} x(t) + \left(\frac{\mu(t - \Delta t)}{\mu(t)} - \frac{\sigma(t - \Delta t)}{\sigma(t)}\right)\sigma(t)^2 s_\phi(x(t), t)$$

which coincides with the deterministic DDIM [36] sampling algorithm when the variance preserving SDE [32] is used. However, as we approximate both the prior and likelihood scores, errors accumulate along the simulation and cause it to diverge, leading to low-quality samples. To prevent errors from accumulating, we perform (see Algorithm 4) a few steps of Langevin Monte Carlo (LMC) [45, 46]

$$x(t) \leftarrow x(t) + \delta s_\phi(x(t), t) + \sqrt{2\delta}\epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$, between each step of the discretized reverse SDE (16). In the limit of an infinite number of LMC steps with a sufficiently small step size $\delta \in \mathbb{R}_+$, simulated samples are guaranteed to follow the distribution implicitly defined by our approximation of the posterior score at each time t, meaning that the errors introduced by the pseudo-blanket (13) and likelihood (15) approximations do not accumulate. In practice, we find that few LMC steps are necessary. Song et al. [28] introduced a similar strategy, named predictor-corrector (PC) sampling, to correct the errors introduced by the discretization of the reverse SDE.

4. Result

We demonstrate the effectiveness of score-based data assimilation on two chaotic dynamical systems: the Lorenz 1963 [47] and Kolmogorov flow [48] systems. The former is a simplified mathematical model for atmospheric convection. Its low dimensionality enables posterior inference using classical sequential Monte Carlo methods [49, 50] such as the bootstrap particle filter [51]. This allows us to compare objectively our posterior approximations against the ground-truth posterior. The second system considers the state of a two-dimensional turbulent fluid subject to Kolmogorov forcing [48]. The evolution of the fluid is modeled by the Navier-Stokes equations, the same equations that underlie the models of oceans and atmospheres. This task provides a good understanding of how SDA would perform in typical data assimilation applications, although our analysis is primarily qualitative due to the unavailability of reliable assessment tools for systems of this scale.

For both systems, we employ as diffusion process the variance preserving SDE with a cosine schedule [52], that is $\mu(t) = \cos(\omega t)^2$ with $\omega = \arccos \sqrt{10^{-3}}$ and $\sigma(t) = \sqrt{1 - \mu(t)^2}$. The score networks are trained once and then evaluated under various observation scenarios. Unless specified otherwise, we estimate the posterior score according to Algorithm 3 with $\Gamma = 10^{-2} I$ and simulate the reverse SDE (4) according to Algorithm 4 in 256 evenly spaced discretization steps.

4.1 Lorenz 1963

The state $x = (a, b, c) \in \mathbb{R}^3$ of the Lorenz system evolves according to a system of ordinary differential equations

$$\begin{aligned} \dot{a} &= \sigma(b - a) \\ \dot{b} &= a(\rho - c) - b \\ \dot{c} &= ab - \beta c \end{aligned}$$

where $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$ are parameters for which the system exhibits a chaotic behavior. We denote $\tilde{a}$ and $\tilde{c}$ the standardized (zero mean and unit variance) versions of a and c, respectively. As our approach assumes a discrete-time stochastic dynamical system, we consider a transition process of the form $x_{i+1} = \mathcal{M}(x_i) + \eta$, where $\mathcal{M} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the integration of the differential equations (18) for $\Delta = 0.025$ time units and $\eta \sim \mathcal{N}(0, \Delta I)$ represents Brownian noise.

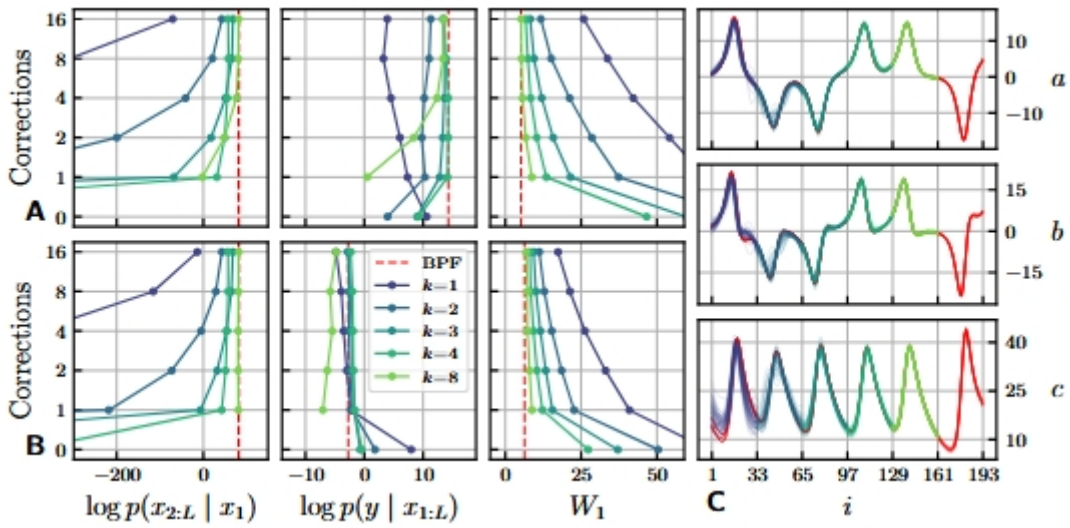


Figure 2: Average posterior summary statistics over 64 observations from the low (A) and high (B) frequency observation processes. We observe that, as k and the number of corrections C increase, the statistics of the approximate posteriors get closer to the ground-truth, in red, which means they are getting more accurate. However, increasing k and C improves the quality of posteriors with decreasing return, such that all posteriors with $k \geq 3$ and $C \geq 2$ are almost equivalent. This is visible in $\mathbf{C}$, where we display trajectories inferred $(C = 2)$ for an observation of the low frequency observation process. For readability, we allocate a segment of 32 states to each k instead of overlapping all 192 states. Note that the Wasserstein distance between the ground-truth posterior and itself is not zero as it is estimated with a finite number (1024) of samples.

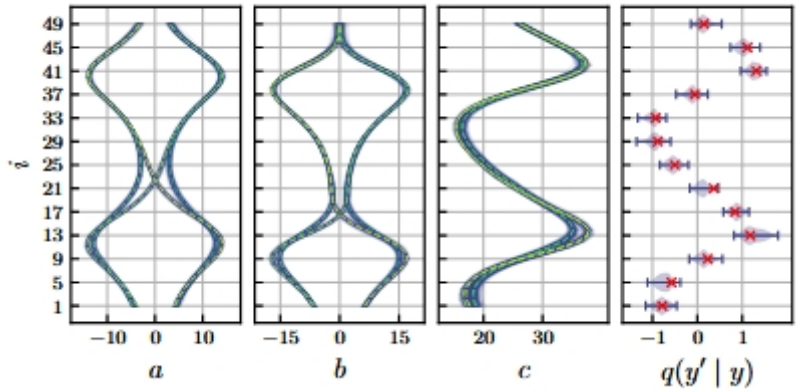


Figure 3: Example of multi-modal posterior inference with SDA. We identify four modes (dashed lines) in the inferred posterior. All modes are consistent with the observation (red crosses), as demonstrated by the posterior predictive distribution $q\left(y' \mid y\right) = \mathbf{E}_{q(x_{1:L}|y)}\left[p\left(y' \mid x_{1:L}\right)\right]$.

We generate 1024 independent trajectories of 1024 states, which are split into training (80%), validation (10%) and evaluation (10%) sets. The initial states are drawn from the statistically stationary regime of the system. We consider two score network architectures: fully-connected local score networks for small $k(k \leq 4)$ and fully-convolutional score networks for large k. Architecture and training details for each k are provided in Appendix D.

We first study the impact of k (see Section 3.1) and the number of LMC corrections (see Section 3.3) on the quality of the inferred posterior. We consider two simple observation processes $\mathcal{N}(y \mid \tilde{a}_{1:L:8}, 0.05^2 I)$ and $\mathcal{N}\left(y \mid \tilde{a}_{1:L}, 0.25^2 I\right)$. The former observes the state at low frequency (every eighth step) with low noise, while the latter observes the state at high frequency (every step) with high noise. For both processes, we generate an observation y for a trajectory of the evaluation set (truncated at $L = 65$) and apply the bootstrap particle filter (BPF) to draw 1024 trajectories $x_{1:L}$ from the ground-truth posterior $p\left(x_{1:L} \mid y\right)$. We use a large number of particles $\left(2^{16}\right)$ to ensure convergence. Then, using SDA, we sample 1024 trajectories from the approximate posterior $q\left(x_{1:L} \mid y\right)$ defined by each score network. We compare the approximate and ground-truth posteriors with three summary statistics: the expected log-prior $\mathbb{E}_{q(x_{1:L}|y)}\left[\log p\left(x_{2:L} \mid x_1\right)\right]$, the expected loglikelihood $\mathbb{E}_{q(x_{1:L}|y)}\left[\log p\left(y \mid x_{1:L}\right)\right]$ and the Wasserstein distance $W_1(p, q)$ in trajectory space. We repeat the procedure for 64 observations and different number of corrections ($\tau = 0.25$, see Algorithm 4) and present the results in Figure 2. To paraphrase, SDA is able to reproduce the ground-truth posterior accurately. Interestingly, accuracy can be traded off for computational efficiency: fewer corrections leads to faster inference at the potential expense of physical consistency.

Another advantage of SDA over variational data assimilation approaches is that it targets the whole posterior distribution instead of point estimates, which allows to identify when several scenarios are plausible. As a demonstration, we generate an observation from the observation process $p(y \mid x_{1:L}) = \mathcal{N}\left(y \mid \tilde{c}_{1:L:4}, 0.1^2 I\right)$ and infer plausible trajectories with SDA $(k = 4, C = 2)$. Several modes are identified in the posterior, which we illustrate in Figure 3.

4.2 Kolmogorov flow

Incompressible fluid dynamics are governed by the Navier-Stokes equations

$$\begin{aligned} \dot{\boldsymbol{u}} &= -\boldsymbol{u}\nabla\boldsymbol{u} + \frac{1}{Re}\nabla^2\boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f} \\ 0 &= \nabla \cdot \boldsymbol{u} \end{aligned}$$

where $\boldsymbol{u}$ is the velocity field, Re is the Reynolds number, ρ is the fluid density, p is the pressure field and $\boldsymbol{f}$ is the external forcing. Following Kochkov et al. [53], we choose a two-dimensional domain $[0, 2\pi]^2$ with periodic boundary conditions, a large Reynolds number $Re = 10^3$, a constant density $\rho = 1$ and an external forcing f corresponding to Kolmogorov forcing with linear damping [48,54]. We use the jax-cfd library [53] to solve the Navier-Stokes equations (19) on a 256×256 domain grid. The states x_i are snapshots of the velocity field $\boldsymbol{u}$, coarsened to a 64×64 resolution, and the integration time between two such snapshots is $\Delta = 0.2$ time units. This corresponds to 82 integration steps of the forward Euler method, which would be expensive to differentiate through repeatedly, as required by gradient-based data assimilation approaches.

We generate 1024 independent trajectories of 64 states, which are split into training (80%), validation (10%) and evaluation (10%) sets. The initial states are drawn from the statistically stationary regime of the system. We consider a local score network with $k = 2$. As states take the form of 64×64 images with two velocity channels, we use a U-Net [55] inspired network architecture. Architecture and training details are provided in Appendix D.

We first apply SDA to a classic data assimilation problem. We take a trajectory of length $L = 32$ from the evaluation set and observe the velocity field every four steps, coarsened to a resolution 8×8 and perturbed by a moderate Gaussian noise ($\Sigma_y = 0.1^2 I$). Given the observation, we sample a trajectory with SDA ($C = 1, \tau = 0.5$) and find that it closely recovers the original trajectory, as illustrated in Figure 4. A similar experiment where we modify the amount of spatial information is presented in Figure 8. When data is insufficient to identify the original trajectory, SDA extrapolates a physically plausible scenario while remaining consistent with the observation, which can also be observed in Figure 6 and 7.

Finally, we investigate whether SDA generalizes to unlikely scenarios. We design an observation process that probes the vorticity of the final state x_L in a circle-shaped sub-domain. Then, we sample a trajectory $(C = 1, \tau = 0.5)$ consistent with a uniform positive vorticity observation in this sub-domain, which is unlikely, but not impossible. The result is discussed in Figure 5.

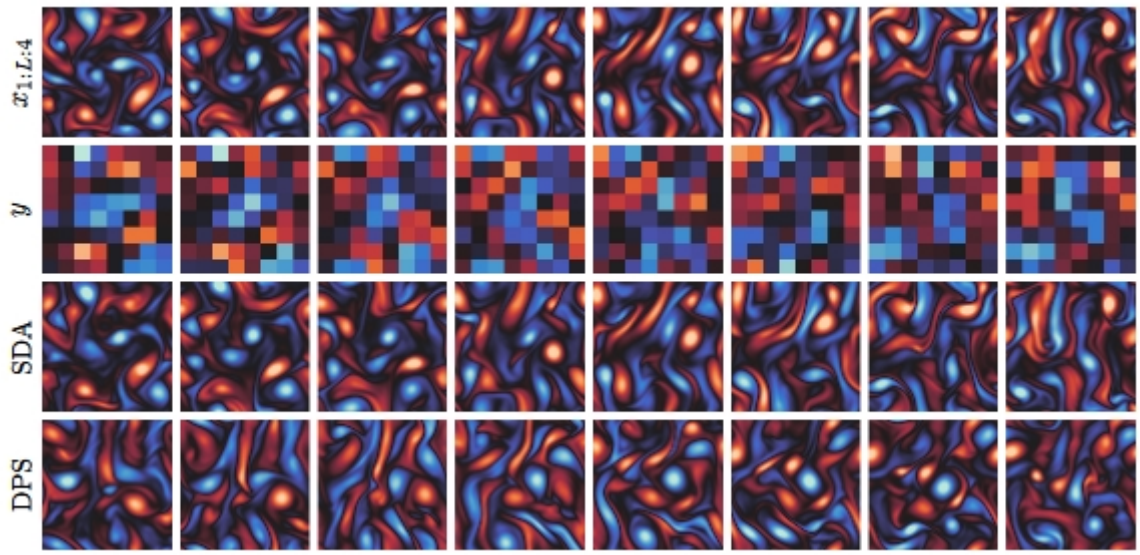


Figure 4: Example of sampled trajectory from coarse, intermittent and noisy observations. States are visualized by their vorticity field $\omega = \nabla \times \boldsymbol{u}$, that is the curl of the velocity field. Positive values (red) indicate clockwise rotation and negative values (blue) indicate counter-clockwise rotation. SDA closely recovers the original trajectory, despite the limited amount of available data. Replacing SDA's likelihood score approximation with the one of DPS [41] yields trajectories inconsistent with the observation.

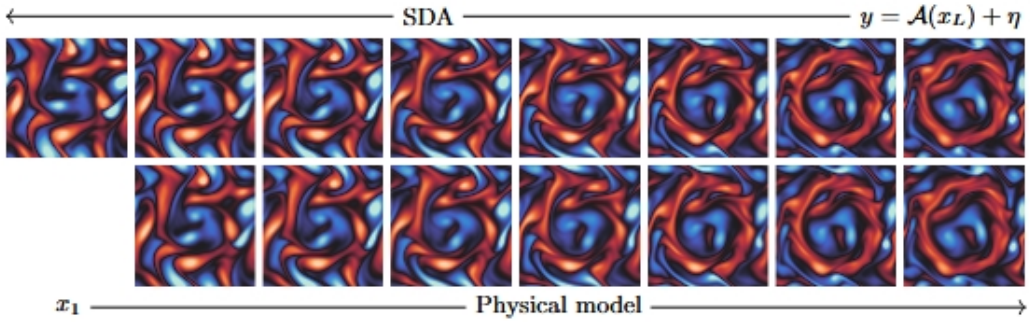


Figure 5: A trajectory consistent with an unlikely observation of the final state x_L is generated with SDA. To verify whether the trajectory is realistic and not hallucinated, we plug its initial state x_1 into the physical model and obtain an almost identical trajectory. This indicates that SDA is not simply interpolating between observations, but rather propagates information in a manner consistent with the physical model, even in unlikely scenarios.

5. Conclusion

Impact

In addition to its contributions to the field of data assimilation, this work presents new technical contributions to the field of score-based generative modeling.

First, we provide new insights on how to exploit conditional independences (Markov blankets) in sets of random variables to build and train score-based generative models. Based on these findings, we are able to generate/infer simultaneously all the states of arbitrarily long Markov chains $x_{1:L}$, while only training score models on short segments $x_{i-k:i+k}$, thereby reducing the training costs and the amounts of training data required. The decomposition of the global score into local scores additionally allows for better parallelization at inference, which could be significant depending on available hardware. Importantly, the pseudo-blanket approximation (13)

is not limited to Markov chains, but could be applied to any set of variables $x_{1:L}$, as long as some structure is known.

Second, we motivate and introduce a novel approximation (15) for the perturbed likelihood $p(y \mid x(t))$, when the likelihood $p(y \mid x)$ is assumed (linear or non-linear) Gaussian. We find that computing the likelihood score $\nabla_{x(t)} \log p(y \mid x(l))$ with this new approximation leads to accurate posterior inference, without the need for stability tricks [41]. This contribution can be trivially adapted to many tasks such as inpainting, deblurring, super-resolution or inverse problems in scientific fields [39-41].

Limitations

From a computational perspective, even though SDA does not require simulating or differentiating through the physical model, inference remains limited by the speed of the simulation of the reverse SDE. Accelerating sampling in score-based generative models is an active area of research $[30, 36, 37, 56]$ with promising results which would be worth exploring in the context of our method.

Regarding the quality of our results, we empirically demonstrate that SDA provides accurate approximations of the whole posterior, especially as k and the number of LMC corrections C increase. However, our approximations (13) and (15) introduce a certain degree of error, which precise impact on the resulting posterior remains to be theoretically quantified. Furthermore, although the Kolmogorov system is high-dimensional (tens of thousands of dimensions) with respect to what is approachable with classical posterior inference methods, it remains small in comparison to the millions of dimensions of some operational DA systems. Whether SDA would scale well to such applications is an open question and will present serious engineering challenges.

Another limitation of our work is the assumption that the dynamics of the system are shared by all trajectories. In particular, if a parametric physical model is used, all trajectories are assumed to share the same parameters. For this reason, SDA is not applicable to settings where fitting the model parameters is also required, or at least not without further developments. Some approaches [57-60] tackle this task, but they remain limited to low-dimensional settings. Additionally, if a physical model is used to generate synthetic training data, instead of relying on real data, one can only expect SDA to be as accurate as the model itself. This is a limitation shared by any model-based approach and robust assimilation under model misspecification or distribution shift is left as an avenue for future research.

Finally, posterior inference over entire state trajectories is not always necessary. In forecasting tasks, inferring the current state of the dynamical system is sufficient and likely much less expensive. In this setting, data assimilation reduces to a state estimation problem for which classical methods such as the Kalman filter [61] or its nonlinear extensions [62,63] provide strong baselines. Many deep learning approaches have also been proposed to bypass the physical model entirely and learn instead a generative model of plausible forecasts from past observations only [64-67].

Related work

A number of previous studies have investigated the use of deep learning to improve the quality and efficiency of data assimilation. Mack et al. [12] use convolutional auto-encoders to project the variational data assimilation problem into a lower-dimensional space, which simplifies the optimization problem greatly. Frerix et al. [14] use a deep neural network to predict the initial state of a trajectory given the observations. This prediction is then used as a starting point for traditional (4D-Var) variational data assimilation methods, which proves to be more effective than

starting at random. This strategy is also possible with SDA (using a trajectory sampled with SDA as a starting point) and could help cover multiple modes of the posterior distribution. Finally, Brajard et al. [17] address the problem of simultaneously learning the transition dynamics and estimating the trajectory, when only the observation process is known.

Beyond data assimilation, SDA closely relates to the broader category of sequence models, which have been studied extensively for various types of data, including text, audio, and video. The latest advances demonstrate that score-based generative models achieve remarkable results on the most demanding tasks. Kong et al. [26] and Goel et al. [27] use score-based models to generate long audio sequences non-autoregressively. Ho et al. [23] train a score-based generative model for a fixed number of video frames and use it autoregressively to generate videos of arbitrary lengths. Conversely, our approach is non-autoregressive which allows to generate and condition all elements (frames) simultaneously. Interestingly, as part of their method, Ho et al. [23] introduce "reconstruction guidance" for conditional sampling, which can be seen as a special case of our likelihood approximation (15) where the observation y is a subset of x. Lastly, Ho et al. [25] generate low-frame rate, low-resolution videos which are then up-sampled temporally and spatially with a cascade [24] of super-resolution diffusion models. The application of this approach to data assimilation could be worth exploring, although the introduction of arbitrary observation processes seems challenging.

> 感谢Claude3.5的翻译

基于分数的数据同化

摘要

数据同化在其最全面的形式中,处理贝叶斯逆问题,即识别可以解释随机动力系统中噪声或不完整观测的可行状态轨迹。已经提出了各种方法来解决这个问题,包括基于粒子和变分方法。然而,大多数算法依赖于转移动力学进行推理,这对于长时间范围或具有复杂动力学的高维系统(如海洋或大气)来说变得难以处理。在这项工作中,我们引入了基于分数的数据同化来进行轨迹推理。我们基于一个关键洞察来学习状态轨迹的基于分数的生成模型,即任意长轨迹的分数可以分解为短片段分数的序列。训练后,使用分数模型进行推理,通过同时生成所有状态的非自回归方式进行。更独特的是,我们将观测模型与训练过程解耦,仅在推理时使用它来指导生成过程,这使得能够处理广泛的零样本观测场景。我们提供了支持我们方法有效性的理论和实证证据。

1. 引言

数据同化(DA)是许多涉及研究复杂动力系统(如大气、海洋或气候)的科学领域的核心。DA的目的是基于各种不完美信息源(包括稀疏、间歇和噪声观测)来推断随时间演化的系统状态。

具体而言,让 $x_{1:L} = (x_1, x_2, \dots, x_L) \in \mathbb{R}^{L\times D}$ 表示离散时间随机动力系统中的状态轨迹,并且 $p(x_{i+1}|x_i)$ 是从状态 x_i 到状态 x_{i+1} 的转移动力学。状态轨迹 $x_{1:L}$ 的观测 $y \in \mathbb{R}^M$ 遵循观测过程 $p(y|x_{1:L})$,通常表述为 $y = \mathcal{A}(x_{1:L}) + \eta$,其中测量函数 $\mathcal{A} : \mathbb{R}^{L\times D} \mapsto \mathbb{R}^M$ 通常是非线性的,观测误差 $\eta \in \mathbb{R}^M$ 是一个随机加性项,用于说明仪器噪声和系统性不确定性。在此框架中,DA的目标是解决给定观测 y 推断可能轨迹 $x_{1:L}$ 的逆问题,即估计轨迹后验概率

$$p\left(x_{1:L} \mid y\right) = \frac{p\left(y \mid x_{1:L}\right)}{p(y)} p\left(x_1\right) \prod_{i=1}^{L-1} p\left(x_{i+1} \mid x_i\right)$$

其中初始状态先验 $p(x_1)$ 通常被称为背景[5-9]。在地球科学中,可用数据量通常不足以仅从观测中恢复系统的完整状态[8]。因此,转移动力学背后的物理模型对于填补观测中的空间和时间间隙至关重要。

最先进的数据同化方法基于变分同化[1,2,5-7]。许多这些方法将任务表述为最大后验概率(MAP)估计问题,并通过梯度上升最大化对数后验密度 $\log p(x_{1:L}|y)$ 来求解。尽管这种方法只产生轨迹后验的点估计,但对于地球物理系统规模和复杂性的问题来说,其成本已经相当可观,因为它需要对物理模型进行微分。因此,由于计算限制,可以同化的数据量受到限制。例如,现代数值天气预报中只利用了少量可用的卫星数据,然而即使有这些限制,数据同化仍占用了现代数值天气预报计算成本的很大一部分[10,11]。最近的研究表明,深度学习可以通过多种方式用来提高数据同化的计算效率,通过估计数据同化后未解决的尺度来提高重建性能,或整合多个观测源[12-19]。

贡献 在这项工作中,我们提出了一种基于基于分数的生成模型的新型数据同化方法。利用动力系统的马尔可夫结构,我们从短的轨迹段训练分数网络,然后能够生成物理上一致且任意长度的状态轨迹。观测模型与分数网络解耦,仅在同化过程中用于指导生成过程,这允许广泛的零样本观测场景。我们的方法在不模拟或对物理模型进行微分的情况下,提供了整个轨迹后验的准确近似 - 它不限于点估计。所有实验的代码可在 https://github.com/francois-rozet/sda 获取。

2. 背景

基于分数的生成模型最近展现出了卓越的能力,推动了图像、视频或音频生成的许多最新进展[20-27]。在本节中,我们回顾基于分数的生成模型并概述如何使用它们来解决逆问题。

基于分数的连续时间生成模型

采用Song等人[28]的表述,来自分布 $p(x)$ 的样本 $x \in \mathbb{R}^D$ 通过连续时间扩散过程逐渐扰动,表示为线性随机微分方程(SDE)

$$\mathrm{d}x(t) = f(t)x(t)\mathrm{d}t + g(t)\mathrm{d}w(t)$$

其中 $f(t) \in \mathbb{R}$ 是漂移系数, $g(t) \in \mathbb{R}$ 是扩散系数, $w(t) \in \mathbb{R}^D$ 表示维纳过程(标准布朗运动), $x(t) \in \mathbb{R}^D$ 是时间 $t \in [0, 1]$ 处的扰动样本。因为SDE对于 $x(t)$ 是线性的,从 x 到 $x(t)$ 的扰动核是高斯的,形式为

$$p(x(t) \mid x) = \mathcal{N}(x(t) \mid \mu(t)x, \Sigma(t))$$

其中 $\mu(t)$ 和 $\Sigma(t) = \sigma(t)^2 I$ 可以从 $f(t)$ 和 $g(t)$ 解析推导[29,30]。表示 $p(x(t))$ 为 $x(t)$ 的边缘分布,我们要求 $\mu(0) = 1$ 和 $\sigma(0) \ll 1$,使得 $p(x(0)) \approx p(x)$,并且我们选择系数 $f(t)$ 和 $g(t)$ 使得初始样本 x 对最终扰动样本 $x(1)$ 的影响相对于噪声水平可以忽略不计 - 即 $p(x(1)) \approx \mathcal{N}(0, \Sigma(1))$。方差爆炸(VE)和方差保持(VP)SDE[28,31,32]是满足这些约束的广泛例子。

至关重要的是,前向SDE(2)的时间反演由反向SDE给出[28,33]

$$\mathrm{d}x(t) = \big[f(t)x(t) - g(t)^2 \nabla_{x(t)} \log p(x(t))\big]\mathrm{d}t + g(t)\mathrm{d}w(t)$$

也就是说,我们可以绘制噪声样本 $x(1) \sim \mathcal{N}(0, \Sigma(1))$ 并通过从 $t = 1$ 到 0 模拟反向SDE逐渐移除其中的噪声来获得数据样本 $x(0) \sim p(x(0))$。这需要访问称为 $p(x(t))$ 的分数的量 $\nabla_{x(t)} \log p(x(t))$。

去噪分数匹配

在实践中,分数 $\nabla_{x(t)} \log p(x(t))$ 由神经网络 $s_\phi(x(t), t)$ 近似,称为分数网络,其被训练以解决去噪分数匹配目标[28,34,35]

$$\arg\min_\phi \mathbb{E}_{p(x)p(t)p(x(t)|x)} \left[\sigma(t)^2 \left\| s_\phi(x(t), t) - \nabla_{x(t)} \log p(x(t) \mid x) \right\|_2^2\right]$$

其中 $p(t)=\mathcal{U}(0,1)$。去噪分数匹配的理论确保对于足够表达的分数网络,$s_\phi(x(t),t)\approx\nabla_{x(t)}\log p(x(t))$。训练后,分数网络被插入到反向SDE(4)中,然后使用适当的离散化方案进行模拟[28,30,36,37]。

在实践中,接近 $t=0$ 时 $\nabla_{x(t)}\log p(x(t)|x)$ 的高方差使得优化(5)不稳定[30]。为了缓解这个问题,通常使用分数网络的略微不同的参数化 $\epsilon_\phi(x(t),t)=-\sigma(t)s_\phi(x(t),t)$,这导致等效的目标[30,32,36]

$$\arg\min_\phi \mathbb{E}_{p(x)p(t)p(\epsilon)}\left[\|\epsilon_\phi(\mu(t)x+\sigma(t)\epsilon,t)-\epsilon\|_2^2\right]$$

其中 $p(\epsilon)=\mathcal{N}(0,I)$。在下文中,为了方便起见,我们保留分数网络的表示法 $s_\phi(x(t),t)$,即使我们采用参数化 $\epsilon_\phi(x(t),t)$ 及其目标进行实验。

零样本逆问题

使用基于分数的生成模型,我们可以从无条件分布 $p(x(0))\approx p(x)$ 生成样本。然而,要解决逆问题,我们需要从后验分布 $p(x|y)$ 采样。这可以通过训练条件分数网络 $s_\phi(x(t),t|y)$ 来近似后验分数 $\nabla_{x(t)}\log p(x(t)|y)$ 并将其插入反向SDE(4)来完成。然而,这需要在训练期间提供数据对 (x,y),并且每次观测过程 $p(y|x)$ 改变时都需要重新训练新的分数网络。相反,许多人观察到[28,38-41]由于贝叶斯规则,后验分数可以分解为两项

$$\nabla_{x(t)}\log p(x(t)\mid y)=\nabla_{x(t)}\log p(x(t))+\nabla_{x(t)}\log p(y\mid x(t))$$

由于先验分数 $\nabla_{x(t)}\log p(x(t))$ 可以用单个分数网络近似,剩下的任务是估计似然分数 $\nabla_{x(t)}\log p(y|x(t))$。假设测量函数 $\mathcal{A}$ 可微且观测过程为高斯 $p(y|x)=\mathcal{N}(y|\mathcal{A}(x),\Sigma_y)$,Chung等人[41]提出近似

$$p(y\mid x(t))=\int p(y\mid x)p(x\mid x(t))\mathrm{d}x\approx\mathcal{N}\left(y\mid\mathcal{A}(\hat{x}(x(t))),\Sigma_y\right)$$

其中均值 $\hat{x}(x(t))=\mathbb{E}_{p(x|x(t))}[x]$ 由Tweedie公式给出[42,43]

$$\begin{aligned}\mathbb{E}_{p(x|x(t))}[x]&=\frac{x(l)+\sigma(t)^2\nabla_{x(t)}\log p(x(t))}{\mu(t)}\\&\approx\frac{x(t)+\sigma(t)^2 s_\phi(x(t),t)}{\mu(t)}\end{aligned}$$

由于多元高斯的对数似然可以解析知道,且 $s_\phi(x(t),t)$ 可微,我们可以在零样本情况下用这个近似计算似然分数 $\nabla_{x(t)}\log p(y|x(t))$,即无需训练除 $s_\phi(x(t),t)$ 以外的任何网络。

> **基本概念介绍**
> 这段文字主要介绍了基于分数的生成模型（Score-based Generative Models）的核心概念和数学原理。这类模型近期在图像、视频、音频生成等领域取得了显著成果。
>
> **连续时间生成模型的数学框架**
>
> - 使用随机微分方程(SDE)描述扰动过程
> - 关键方程：$\mathrm{d}x(t)=f(t)x(t)\mathrm{d}t+g(t)\mathrm{d}w(t)$
> - 包含漂移系数f(t)和扩散系数g(t)
> - 扰动核是高斯分布
>
> **重要特性**
>
> - 从t=0到t=1的前向过程是逐渐增加噪声
> - 初始状态p(x(0))近似原始分布
> - 最终状态p(x(1))近似标准高斯分布

- 存在可逆的反向SDE过程

去噪分数匹配（Denoising Score Matching）

- 使用神经网络s_φ(x(t),t)来近似分数函数
- 优化目标是最小化预测分数与真实分数的差异
- 提供了替代参数化方法来提高训练稳定性

零样本逆问题求解

- 利用贝叶斯规则分解后验分数
- 创新性地使用Tweedie公式估计条件期望
- 实现了无需额外训练就能解决新的逆问题

技术亮点

- 将连续时间生成模型与逆问题求解结合
- 提供了一个通用框架，可以处理各种观测过程
- 实现了零样本（zero-shot）逆问题求解

实际应用优势

- 不需要针对每个新的观测过程重新训练模型
- 可以处理可微的测量函数
- 适用于高斯观测过程

理论创新

- 将SDE理论应用到生成模型中
- 结合了分数匹配和扩散模型的优点
- 提供了理论上合理的近似方法

这对于实际应用具有重要意义，因为它大大降低了解决新逆问题的计算成本和数据需求。

3. 基于分数的数据同化

回到我们最初的推理问题,我们想要近似动力系统的轨迹后验 $p(x_{1:L}|y)$。为此使用基于分数的生成建模,我们需要估计后验分数 $\nabla_{x_{1:L}(t)} \log p(x_{1:L}(t)|y)$,我们选择将其分解为先验和似然项,如(7)所示,以启用广泛的零样本观测场景。

在典型的数据同化设置中,每个状态 x_i 的高维性(例如大气或海洋的状态)与潜在的长轨迹相结合,需要一个不切实际的大型分数网络 $s_\phi(x_{1:L}(t), t)$ 来估计先验分数 $\nabla_{x_{1:L}(t)} \log p(x_{1:L}(t))$ 和相应数量的训练数据,如果数据稀缺或物理模型模拟昂贵,这可能难以实现。为了克服这个挑战,我们利用动力系统的马尔可夫结构用一系列更容易学习的局部分数来近似先验分数,如3.1节所述。在3.2节中,我们基于扩散后验采样(DPS)[41]提出了一个新的似然分数 $\nabla_{x_{1:L}(t)} \log p(y|x_{1:L}(t))$ 的近似,我们发现这更适合后验推理。最后,在3.3节中,我们描述了受预测器-校正器采样[28]启发的采样程序。我们的主要贡献，命名为基于分数的数据同化(SDA),是这三个组件的组合。

问题背景：参考传统的基于分数的生成建模在高纬度大气海洋模型中不适用，需要极大规模的分数网络、需要大量训练数据、物理模型模拟成本高昂、相关数据可能稀缺

F.Rozet提出了解决方案，SDA。基本假设是动力系统是马尔可夫结构

1. **利用马尔可夫结构**

- 将大规模先验分数分解为多个局部分数
- 简化学习难度
- 利用系统的时序依赖性

2. **改进的似然分数近似**
   - 基于扩散后验采样(DPS)
   - 针对后验推理进行优化
   - 提高了推理准确性
3. **预测器-校正器采样程序**
   - 受现有采样方法启发
   - 用于实际执行采样过程

通过马尔可夫分解降低复杂度、改进了似然分数的估计方法、提供了可实现的采样策略、降低了计算复杂度、减少了数据需求、更适合实际应用场景、保持了理论框架的完整性

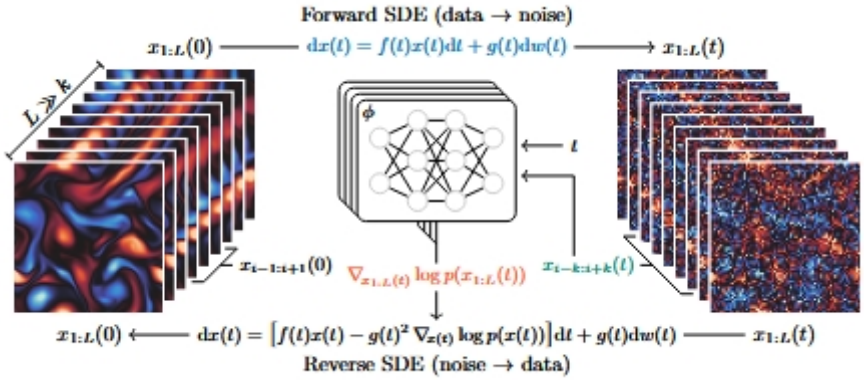


图1：动力系统的轨迹 $x_{1:L}$ 通过扩散过程转换为噪声。逆转这个过程可以生成新的轨迹，但需要 $p\left(x_{1:L}(l)\right)$ 的分数。我们通过组合分数网络在 $x_{1:L}(l)$ 子段上的输出来近似它。

图像呈现了一个时序过程；显示了前向SDE和反向SDE两个过程

1. 前向SDE的过程：从原始数据x₁:L(0)开始⇒通过方程 dx(t) = f(t)x(t)dt + g(t)dw(t) 进行扩散⇒逐渐将清晰的数据模式转换为噪声
2. 反向SDE的过程：从噪声开始⇒使用方程 dx(t) = [f(t)x(t) - g(t)²∇x(t)log p(x(t))]dt + g(t)dw(t) ⇒通过分数函数指导，将噪声逐步转换回有意义的数据

3.1 你的毯子如何？马尔可夫结构

给定一组随机变量 $x_{1:L} = \{x_1, x_2, \ldots, x_L\}$，有时可以利用我们对集合结构的了解，为每个元素 x_i 找到一个小的马尔可夫毯 $x_{b_i} \subseteq x_{\neq i}$，使得 $p\left(x_i \mid x_{\neq i}\right) = p\left(x_i \mid x_{b_i}\right)$。

因此，完整分数 $\nabla_{x_{1:L}} \log p\left(x_{1:L}\right)$ 的每个元素 $\nabla_{x_i} \log p\left(x_{1:L}\right)$ 都可以局部确定，即只使用其毯子：

$$\begin{aligned}\nabla_{x_i} \log p\left(x_{1:L}\right) &= \nabla_{x_i} \log p\left(x_i \mid x_{\neq i}\right) + \nabla_{x_i} \log p\left(x_{\neq i}\right) \\ &= \nabla_{x_i} \log p\left(x_i \mid x_{b_i}\right) + \nabla_{x_i} \log p\left(x_{b_i}\right) = \nabla_{x_i} \log p\left(x_i, x_{b_i}\right)\end{aligned}$$

该属性通常不适用于扩散扰动集 $x_{1:L}(l)$，因为没有保证 $x_{b_i}(l)$ 是元素 $x_i(l)$ 的马尔可夫毯。然而，存在一组索引 $\bar{b}_i \supseteq b_i$，使得

$$\nabla_{x_i(t)} \log p\left(x_{1:L}(t)\right) \approx \nabla_{x_i(t)} \log p\left(x_i(t), x_{\bar{b}_i}(t)\right)$$

对于所有 $t \in [0,1]$，这是一个良好的近似。也就是说，$x_{b_i}(t)$ 是 $x_i(t)$ 的"伪"马尔可夫毯。在最坏的情况下，$\bar{b}_i$ 包含除 i 以外的所有索引，但我们认为，对于某些结构，存在一个集合 $\bar{b}_i$，其大小并不比 b_i 大多少，且满足 (13)。我们的理由是，由于我们假设初始噪声可以忽略不计，我们知道当 t 接近 0 时，$x_{b_i}(t)$ 变得与 x_{b_i} 无法区分。此外，随着 l 的增长和噪声的积累，元素 $x_i(l)$ 和 $x_j(t)$ 之间的互信息减少，最终在 $t=1$ 时达到 0。因此，即使 $\bar{b}_i = b_i$，伪毯近似 (13) 在 $t=0$ 和 $t=1$ 附近已经成立。在这两者之间，尽管近似保持无偏（见附录 A），但集合的结构变得至关重要。如果已知并存在足够的规律/对称性，(13) 可以并且应该在评分网络 $s_\phi\left(x_{1:L}(\ell), t\right)$ 的架构中加以利用。

在动态系统的情况下，集合 $x_{1:L}$ 根据定义是一个一阶马尔可夫链，元素 x_i 的最小马尔可夫毯是 $x_{b_i} = \{x_{i-1}, x_{i+1}\}$。对于扰动元素 $x_i(t)$，伪毯 $x_{\bar{b}_i}(t)$ 可以呈现为围绕 $x_i(t)$ 的窗口，即 $\bar{b}_i = \{i-k, \ldots, i+k\} \backslash \{i\}$，其中 $k \geq 1$。k 的值取决于问题，但我们认为，基于我们的实验，通常远小于链的长度 L。因此，具有狭窄感受野的全卷积神经网络（FCNN）非常适合这个任务，任何长程能力都将是浪费资源。重要的是，如果感受野是 $2k+1$，网络可以在段 $x_{i-k:i+k}$ 上进行训练，而不是整个链 $x_{1:L}$，从而大幅降低训练成本。更一般地，我们可以训练一个局部评分网络（见算法 1）

$$s_\phi\left(x_{i-k:i+k}(t), t\right) \approx \nabla_{x_{i-k:i+k}(t)} \log p\left(x_{i-k:i+k}(t)\right)$$

使得其 $k+1$-th 元素近似于第 i 个状态的得分 $\nabla_{x_i(t)} \log p\left(x_{1:L}(l)\right)$。我们还知道，$s_\phi\left(x_{1:2k+1}(t), t\right)$ 的前 k 个元素近似于前 k 个状态的得分 $\nabla_{x_{1:k}(t)} \log p\left(x_{1:L}(t)\right)$，而 $s_\phi\left(x_{L-2k:L}(t), t\right)$ 的后 k 个元素近似于后 k 个状态的得分 $\nabla_{x_{L-k:L}(t)} \log p\left(x_{1:L}(t)\right)$。因此，我们可以在所有子段 $x_{i-k:i+k}(t)$ 上应用局部得分网络，类似于卷积核，并结合输出（见算法 2）以获得完整得分的近似值 $\nabla_{x_{1:L}(t)} \log p\left(x_{1:L}(t)\right)$。请注意，我们可以用 i 来条件化得分网络，或者假设链的统计平稳性，即 $p\left(x_i\right) = p\left(x_{i+1}\right)$。

> 提出了基于局部评分网络的训练方法
>
> 使用全卷积神经网络(FCNN)处理局部依赖关系
>
> 可以在较小的序列片段上训练，而不需要处理整个序列
>
> 显著降低了计算复杂度；保持了模型的准确性；可以利用序列的统计平稳性

Algorithm 1 Training $\epsilon_\phi\left(x_{i-k:i+k}(t), t\right)$

1 for $i = 1$ to N do
2 $x_{1:L} \sim p\left(x_{1:L}\right)$
3 $i \sim \mathcal{U}(\{k+1, \ldots, L-k\})$
4 $t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0, I)$
5 $x_{i-k:i+k}(l) \leftarrow \mu(t) x_{i-k:i+k} + \sigma(l)\epsilon$
6 $\ell \leftarrow \left\|\epsilon_\phi\left(x_{i-k:i+k}(t), l\right) - \epsilon\right\|_2^2$
7 $\phi \leftarrow \mathrm{GRADIENTDESCENT}\left(\phi, \nabla_\phi \ell\right)$

Algorithm 2 Composing $s_\phi\left(x_{i-k:i+k}(t), t\right)$

1 function $s_\phi\left(x_{1:L}(t), t\right)$
2 $s_{1:k+1} \leftarrow s_\phi\left(x_{1:2k+1}(t), t\right)[:k+1]$
3 for $i = k+2$ to $L-k-1$ do
4 $s_i \leftarrow s_\phi\left(x_{i-k:i+k}(t), t\right)[k+1]$
5 $s_{L-k:L} \leftarrow s_\phi\left(x_{L-2k:L}(t), t\right)[k+1:]$
6 return $s_{1:L}$

3.2 稳定的似然分数

该属性通常不适用于扩散扰动集 $x_{1:L}(l)$，因为没有保证 $x_{b_i}(l)$ 是元素 $x_i(l)$ 的马尔可夫毯。然而，存在一组索引 $\bar{b}_i \supseteq b_i$，使得

$$\nabla_{x_i(t)} \log p\left(x_{1:L}(t)\right) \approx \nabla_{x_i(t)} \log p\left(x_i(t), x_{\bar{b}_i}(t)\right)$$

对于所有 $t \in [0,1]$，这是一个良好的近似。也就是说，$x_{b_i}(t)$ 是 $x_i(t)$ 的"伪"马尔可夫毯。在最坏的情况下，$\bar{b}_i$ 包含除 i 以外的所有索引，但我们认为，对于某些结构，存在一个集合 $\bar{b}_i$，其大小并不比 b_i 大多少，且满足 (13)。我们的理由是，由于我们假设初始噪声可以忽略不计，我们知道当 t 接近 0 时，$x_{b_i}(t)$ 变得与 x_{b_i} 无法区分。此外，随着 l 的增长和噪声的积累，元素 $x_i(l)$ 和 $x_j(t)$ 之间的互信息减少，最终在 $t=1$ 时达到 0。因此，即使 $\bar{b}_i = b_i$，伪毯近似 (13) 在 $t=0$ 和 $t=1$ 附近已经成立。在这两者之间，尽管近似保持无偏（见附录 A），但集合的结构变得至关重要。如果已知并存在足够的规律/对称性，(13) 可以并且应该在评分网络 $s_\phi\left(x_{1:L}(\ell), t\right)$ 的架构中加以利用。

> 对于扰动后的元素 $x_i(t)$，其伪马尔可夫毯扩展为窗口形式；窗口范围：{i-k, ..., i+k}{i}，k ≥ 1
>
> 窗口大小(k)通常远小于整个序列长度(L)

在动态系统的情况下，集合 $x_{1:L}$ 根据定义是一个一阶马尔可夫链，元素 x_i 的最小马尔可夫毯是 $x_{b_i} = \{x_{i-1}, x_{i+1}\}$。对于扰动元素 $x_i(t)$，伪毯 $x_{\bar{b}_i}(t)$ 可以呈现为围绕 $x_i(t)$ 的窗口，即 $\bar{b}_i = \{i-k, \ldots, i+k\} \backslash \{i\}$，其中 $k \geq 1$。k 的值取决于问题，但我们认为，基于我们的实验，通常远小于链的长度 L。因此，具有**狭窄感受野的全卷积神经网络（FCNN）非常适合这个任务**，任何长程能力都将是浪费资源。重要的是，如果感受野是 $2k+1$，网络可以在段 $x_{i-k:i+k}$ 上进行训练，而不是整个链 $x_{1:L}$，从而大幅降低训练成本。更一般地，我们可以训练一个局部评分网络（见算法 1）

$$s_\phi\left(x_{i-k:i+k}(t), t\right) \approx \nabla_{x_{i-k:i+k}(t)} \log p\left(x_{i-k:i+k}(t)\right)$$

使得其 $k+1$-th 元素近似于第 i 个状态的得分 $\nabla_{x_i(t)} \log p\left(x_{1:L}(l)\right)$。我们还知道，$s_\phi\left(x_{1:2k+1}(t), t\right)$ 的前 k 个元素近似于前 k 个状态的得分 $\nabla_{x_{1:k}(t)} \log p\left(x_{1:L}(t)\right)$，而 $s_\phi\left(x_{L-2k:L}(t), t\right)$ 的后 k 个元素近似于后 k 个状态的得分 $\nabla_{x_{L-k:L}(t)} \log p\left(x_{1:L}(t)\right)$。因此，我们可以在所有子段 $x_{i-k:i+k}(t)$ 上应用局部得分网络，类似于卷积核，并结合输出（见算法 2）以获得完整得分的近似值 $\nabla_{x_{1:L}(t)} \log p\left(x_{1:L}(t)\right)$。请注意，我们可以用 i 来条件化得分网络，或者假设链的统计平稳性，即 $p\left(x_i\right) = p\left(x_{i+1}\right)$。

> 局部评分网络的工作机制：
>
> - 输入：长度为2k+1的局部序列片段
> - 输出：该片段的似然分数估计
> - 中心位置(k+1)的输出对应目标位置的评分
> - 可以通过滑动窗口方式处理整个序列

3.3 预测器-校正器采样

为了模拟反向 SDE，我们采用了 Zhang 等人提出的指数积分器（EI）离散化方案[30]。

$$x(t-\Delta t) \leftarrow \frac{\mu(t-\Delta t)}{\mu(t)} x(t) + \left(\frac{\mu(t-\Delta t)}{\mu(t)} - \frac{\sigma(t-\Delta t)}{\sigma(t)}\right) \sigma(t)^2 s_\phi(x(t), t)$$

当使用方差保持 SDE [32]时，这与确定性 DDIM [36]采样算法相吻合。然而，由于我们同时近似先验和似然分数，*误差在模拟过程中累积并导致其发散，从而产生低质量样本*。为了防止误差累积，我们执行（见算法 4）几步 Langevin 蒙特卡洛（LMC）[45, 46]。

$$x(t) \leftarrow x(t) + \delta s_\phi(x(t), t) + \sqrt{2\delta}\epsilon$$

其中 $\epsilon \sim \mathcal{N}(0, I)$，在离散化的反向随机微分方程 (16) 的每一步之间。在步长足够小且 LMC 步数无限的极限下，模拟样本保证遵循我们在每个时间 t 对后验得分的近似隐式定义的分布，这意味着伪毛毯 (13) 和似然 (15) 近似引入的误差不会累积。实际上，我们发现只需要少量的 LMC 步骤。Song 等人 [28] 提出了一个类似的策略，称为预测-校正 (PC) 采样，以纠正反向 SDE 离散化引入的误差。

> 结合了确定性的SDE离散化和随机的LMC更新，既保持了采样效率，又确保了采样质量。

4. 结果

我们展示了基于评分的数据同化在两个混沌动力系统上的有效性：洛伦兹 1963 [47] 和科尔莫戈罗夫流 [48] 系统。前者是一个简化的气候对流数学模型。其低维度使得可以使用经典的序列蒙特卡洛方法 [49, 50] 进行后验推断，例如自助粒子滤波器 [51]。这使我们能够客观地将我们的后验近似与真实后验进行比较。第二个系统考虑了受科尔莫戈罗夫强迫 [48] 影响的二维湍流流体的状态。流体的演变由纳维-斯托克斯方程建模，这些方程是海洋和大气模型的基础。这项任务提供了对 SDA 在典型数据同化应用中表现的良好理解，尽管由于缺乏可靠的评估工具，我们的分析主要是定性的。

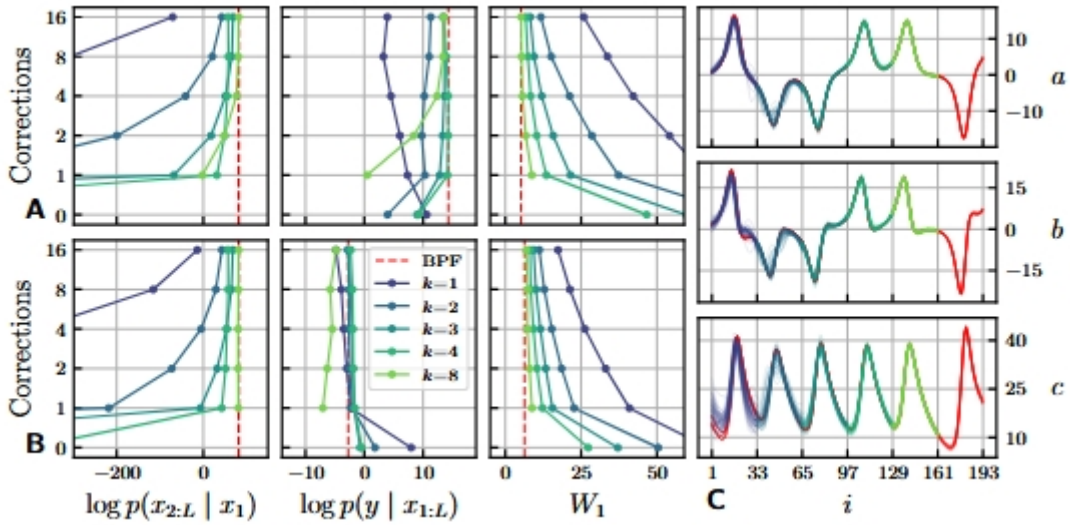
对于这两个系统，我们采用方差保持的随机微分方程作为扩散过程，使用余弦调度 [52]，即 $\mu(t) = \cos(\omega t)^2$，其中 $\omega = \arccos \sqrt{10^{-3}}$，$\sigma(t) = \sqrt{1 - \mu(t)^2}$。评分网络训练一次后，在各种观察场景下进行评估。除非另有说明，我们根据算法 3 估计后验评分，$\Gamma = 10^{-2} I$，并根据算法 4 在 256 个均匀分布的离散化步骤中模拟反向随机微分方程 (4)。

4.1 洛伦兹 1963

洛伦兹系统的状态 $x = (a, b, c) \in \mathbb{R}^3$ 根据一组常微分方程演变。

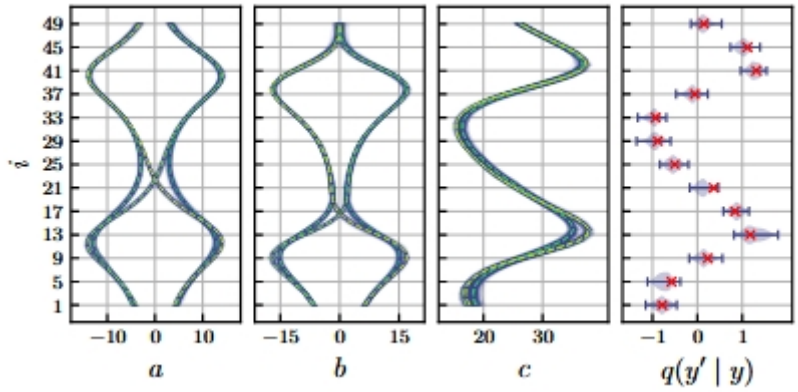
$$\begin{aligned} \dot{a} &= \sigma(b - a) \\ \dot{b} &= a(\rho - c) - b \\ \dot{c} &= ab - \beta c \end{aligned}$$

其中 $\sigma = 10, \rho = 28$ 和 $\beta = \frac{8}{3}$ 是系统表现出混沌行为的参数。我们用 $\tilde{a}$ 和 $\tilde{c}$ 表示 a 和 c 的标准化（零均值和单位方差）版本。由于我们的方法假设为离散时间随机动力系统，我们考虑形式为 $x_{i+1} = \mathcal{M}(x_i) + \eta$ 的转移过程，其中 $\mathcal{M} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ 是对微分方程 (18) 的积分，时间单位为 $\Delta = 0.025$，而 $\eta \sim \mathcal{N}(0, \Delta I)$ 代表布朗噪声。



> 图 2：来自低（A）和高（B）频率观察过程的 64 个观察值的平均后验摘要统计。我们观察到，随着k和修正次数C的增加，近似后验的统计量越来越接近真实值（红色），这意味着它们变得更加准确。然而，增加k和C会提高后验的质量，但收益递减，因此所有$k \geq 3$和$C \geq 2$的后验几乎是等效的。这在$\mathbf{C}$中可见，我们展示了低频观察过程的一个观察值的轨迹推断（$C = 2$）。为了可读

> 性，我们为每个k分配 32 个状态的段，而不是重叠所有 192 个状态。请注意，真实后验与自身之间的 Wasserstein 距离并不为零，因为它是用有限数量（1024）样本估计的。



> 图 3：使用 SDA 进行多模态后验推断的示例。我们在推断的后验中识别出四个模式（虚线）。所有模式与观察结果（红色交叉）一致，正如后验预测分布所示
> $q\left(y' \mid y\right)=\mathbf{E}_{q\left(x_{1:L} \mid y\right)}\left[p\left(y' \mid x_{1:L}\right)\right]$。

我们生成 1024 条独立轨迹，每条轨迹包含 1024 个状态，这些状态被分为训练集（80%）、验证集（10%）和评估集（10%）。初始状态从系统的统计平稳状态中抽取。我们考虑两种评分网络架构：对于小的$k\left(k \leq 4\right)$使用全连接局部评分网络，对于大的k使用全卷积评分网络。每个k的架构和训练细节在附录 D 中提供。

我们首先研究 k（见第 3.1 节）和 LMC 修正次数（见第 3.3 节）对推断后验质量的影响。我们考虑两个简单的观测过程 $\mathcal{N}\left(y \mid \tilde{a}_{1:L:8}, 0.05^2 I\right)$ 和 $\mathcal{N}\left(y \mid \tilde{a}_{1:L}, 0.25^2 I\right)$。前者以低频率（每八步）和低噪声观察状态，而后者以高频率（每一步）和高噪声观察状态。对于这两个过程，我们为评估集的轨迹生成观测 y（截断在 $L=65$），并应用自助粒子滤波器（BPF）从真实后验 $p\left(x_{1:L} \mid y\right)$ 中抽取 1024 条轨迹 $x_{1:L}$。我们使用大量粒子 $\left(2^{16}\right)$ 以确保收敛。然后，使用 SDA，我们从每个评分网络定义的近似后验 $q\left(x_{1:L} \mid y\right)$ 中抽取 1024 条轨迹。 我们使用三个摘要统计量比较近似后验和真实后验：期望的 $\log$-先验 $\mathbb{E}_{q\left(x_{1:L} \mid y\right)}\left[\log p\left(x_{2:L} \mid x_1\right)\right]$，期望的对数似然 $\mathbb{E}_{q\left(x_{1:L} \mid y\right)}\left[\log p\left(y \mid x_{1:L}\right)\right]$ 和轨迹空间中的 Wasserstein 距离 $W_1(p, q)$。我们对 64 个观测值和不同数量的修正（$\tau=0.25$，见算法 4）重复该过程，并在图 2 中呈现结果。换句话说，SDA 能够准确地重现真实后验。有趣的是，准确性可以与计算效率进行权衡：较少的修正会导致推断速度更快，但可能会牺牲物理一致性。

SDA 相对于变分数据同化方法的另一个优势在于它针对整个后验分布，而不是点估计，这使得能够识别出多个情景是合理的。作为演示，我们从观测过程生成一个观测值$p(y \mid x_{1:L})=\mathcal{N}\left(y \mid \tilde{c}_{1:L:4}, 0.1^2 I\right)$并使用 SDA$\left(k=4, C=2\right)$推断合理的轨迹。在后验中识别出多个模式，我们在图 3 中进行了说明。

> 1. 实验设置：
>   - 研究对象：洛伦兹系统（三维混沌系统）
>   - 系统参数：σ=10, ρ=28, β=8/3
>   - 离散化：时间步长Δ=0.025
>   - 数据量：1024条轨迹，每条1024个状态
>   - 数据分割：80%训练，10%验证，10%评估
> 2. 网络架构选择：
>   - 小k值(k≤4)：使用全连接局部评分网络
>   - 大k值：使用全卷积评分网络
> 3. 两种观测过程比较：
>   - 低频率观测：每8步观察一次，低噪声(σ=0.05)

- 高频率观测：每步都观察，高噪声(σ=0.25)

4. 实验结果分析（图2）：
   - A组（低频率）和B组（高频率）显示：
     - k值增加提高准确性
     - 修正次数C增加提高质量
     - k≥3且C≥2时效果趋于稳定
   - C部分显示了具体轨迹推断结果
5. 多模态推断结果（图3）：
   - 成功识别出4种不同模式
   - 所有模式都与观察结果吻合
   - 展示了方法在处理多种可能性时的优势
6. 主要发现：
   - 方法可以准确重现真实后验分布
   - 存在计算效率和准确性的权衡
   - 能够识别多个合理的解释方案
7. 方法优势：
   - 可以处理完整后验分布
   - 不限于单点估计
   - 能识别多个合理场景

4.2 科尔莫戈罗夫流

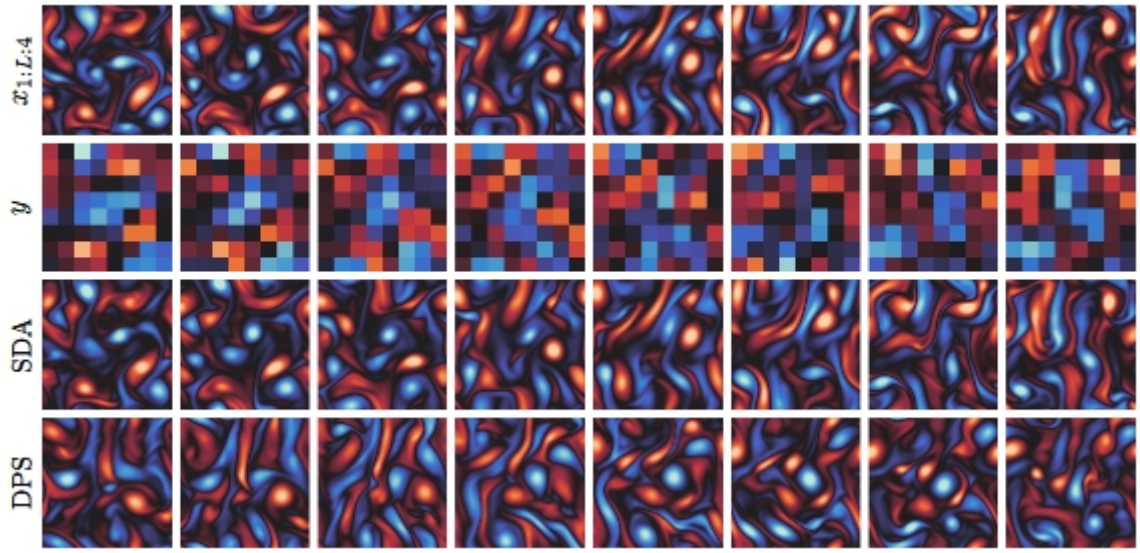
不可压缩流体动力学由纳维-斯托克斯方程支配

$$\dot{\boldsymbol{u}} = -\boldsymbol{u}\nabla\boldsymbol{u} + \frac{1}{Re}\nabla^2\boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f}$$
$$0 = \nabla \cdot \boldsymbol{u}$$

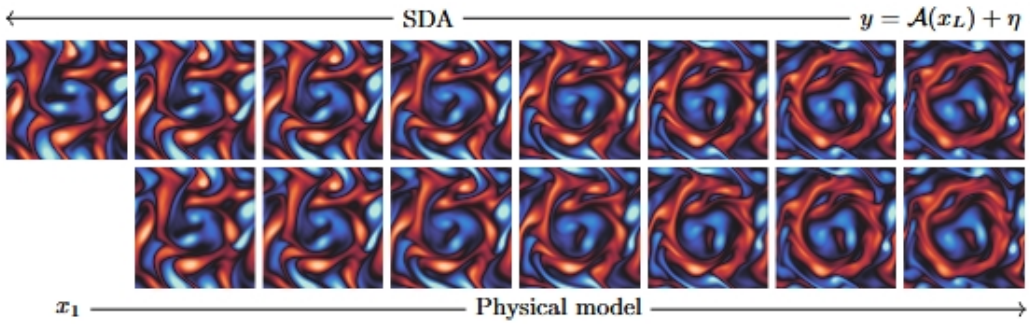
其中 $\boldsymbol{u}$ 是速度场，Re 是雷诺数，ρ 是流体密度，p 是压力场，$\boldsymbol{f}$ 是外部强迫。根据 Kochkov 等人的研究[53]，我们选择一个二维域 $[0, 2\pi]^2$，具有周期性边界条件，较大的雷诺数 $Re = 10^3$，恒定密度 $\rho = 1$，以及对应于 Kolmogorov 强迫的线性阻尼的外部强迫 f [48,54]。我们使用 jax-cfd 库[53]在 256×256 的域网格上求解 Navier-Stokes 方程（19）。状态 x_i 是速度场 $\boldsymbol{u}$ 的快照，粗化到 64×64 的分辨率，两个这样的快照之间的积分时间为 $\Delta = 0.2$ 时间单位。这对应于前向欧拉方法的 82 个积分步骤，这在基于梯度的数据同化方法中需要反复区分，成本较高。
我们生成 1024 条独立的 64 状态轨迹，这些轨迹被分为训练集（80%）、验证集（10%）和评估集（10%）。初始状态来自系统的统计平稳状态。我们考虑一个局部评分网络，$k = 2$。由于状态呈现为具有两个速度通道的64×64图像，我们使用了一个受 U-Net [55]启发的网络架构。架构和训练细节在附录 D 中提供。

我们首先将 SDA 应用于一个经典的数据同化问题。我们从评估集中取出一个长度为$L = 32$的轨迹，并每四步观察一次速度场，分辨率降低到8×8，并受到适度的高斯噪声（$\Sigma_y = 0.1^2 I$）的干扰。给定观察结果，我们使用 SDA（$C = 1, \tau = 0.5$）对轨迹进行采样，发现它与原始轨迹非常接近，如图 4 所示。图 8 中展示了一个类似的实验，其中我们修改了空间信息的量。当数据不足以识别原始轨迹时，SDA 能够推断出一个物理上合理的情景，同时与观察结果保持一致，这在图 6 和 7 中也可以观察到。

最后，我们研究 SDA 是否能够推广到不太可能的情景。我们设计了一个观察过程，探测最终状态x_L在一个圆形子域中的涡度。然后，我们在这个子域中采样一个与均匀正涡度观察结果一致的轨迹（$C = 1, \tau = 0.5$），这虽然不太可能，但并非不可能。结果在图 5 中进行了讨论。



> 图 4：从粗略、不连续和嘈杂的观测中采样的轨迹示例。状态通过其涡度场 $\omega = \nabla \times \boldsymbol{u}$ 可视化，即速度场的旋度。正值（红色）表示顺时针旋转，负值（蓝色）表示逆时针旋转。尽管可用数据有限，SDA 仍能准确恢复原始轨迹。用 DPS [41] 的似然分数近似替代 SDA 的似然分数近似会导致与观测不一致的轨迹。



> 图 5：使用 SDA 生成与最终状态 x_L 不太可能的观测一致的轨迹。为了验证轨迹是否真实而非幻觉，我们将其初始状态 x_1 插入物理模型中，得到几乎相同的轨迹。这表明 SDA 并不仅仅是在观测之间进行插值，而是以与物理模型一致的方式传播信息，即使在不太可能的场景中也是如此。

1. 实验设置：
    - 基于纳维-斯托克斯方程的不可压缩流体模拟
    - 参数设置：
        - 二维周期性边界条件：[0,2π]²
        - 雷诺数：Re=10³
        - 密度：ρ=1
        - 网格：256×256降采样到64×64
        - 时间步长：Δ=0.2
2. 数据集构建：
    - 1024条独立轨迹，每条64个状态
    - 数据分割：80%训练/10%验证/10%测试

- 使用系统的统计平稳状态作为初始状态

3. 网络架构：
   - 局部评分网络(k=2)
   - 基于U-Net的架构设计
   - 输入：64×64分辨率，2个速度通道
4. 实验结果分析（图4）：
   - 从粗糙观测中重建：
     - 每4步观察一次
     - 降采样到8×8
     - 加入高斯噪声(σ=0.1)
   - SDA表现：
     - 准确重建原始轨迹
     - 优于DPS方法
     - 保持物理一致性
5. 特殊场景测试（图5）：
   - 设计了不太可能的观测场景
   - SDA能够：
     - 生成符合物理规律的轨迹
     - 与观测保持一致
     - 通过物理模型验证其合理性
6. 主要发现：
   - SDA能有效处理：
     - 稀疏观测
     - 噪声干扰
     - 低分辨率数据
   - 方法优势：
     - 物理一致性好
     - 适应性强
     - 可处理不太可能的场景
7. 可视化特点：
   - 使用涡度场显示结果
   - 红色表示顺时针旋转
   - 蓝色表示逆时针旋转

5. 结论

影响

除了对数据同化领域的贡献外,本工作还对基于分数的生成建模领域提出了新的技术贡献。

首先,我们提供了关于如何在随机变量集中利用条件独立性(马尔可夫毯)来构建和训练基于分数的生成模型的新见解。基于这些发现,我们能够同时生成/推断任意长马尔可夫链 $x_{1:L}$ 的所有状态,同时仅在短片段 $x_{i-k:i+k}$ 上训练分数模型,从而减少了训练成本和所需的训练数据量。将全局分数分解为局部分数还允许在推理时更好地并行化,这取决于可用硬件可能带来显著提升。重要的是,伪毯近似(13)不限于马尔可夫链,而可以应用于任何变量集 $x_{1:L}$,只要知道某些结构。

其次,当假设似然 $p(y|x)$ 为(线性或非线性)高斯时,我们推导并引入了扰动似然 $p(y|x(t))$ 的新型近似(15)。我们发现使用这个新近似计算似然分数 $\nabla_{x(t)} \log p(y|x(t))$ 可以导致准确的后验推理,而无需使用稳定性技巧[41]。这一贡献可以轻松适用于许多任务,如图像修复、去模糊、超分辨率或科学领域的逆问题[39-41]。

> 利用条件独立性构建生成模型；可在短片段上训练模型；减少训练成本和数据需求；支持并行化处理
>
> 适用于高斯似然情况；无需额外稳定性技巧；可应用于多种任务（图像修复、去模糊等）

局限性

从计算角度来看,尽管SDA不需要模拟或微分物理模型,推理仍然受限于反向SDE模拟的速度。加速基于分数生成模型的采样是一个活跃的研究领域[30,36,37,56],其有前景的结果值得在我们方法的背景下探索。

关于结果质量,我们经验性地证明SDA提供了整个后验的准确近似,特别是随着 k 和LMC校正数量 C 的增加。然而,我们的近似(13)和(15)引入了一定程度的误差,其对结果后验的精确影响仍需理论量化。此外,虽然Kolmogorov系统相对于传统后验推理方法可处理的维度来说是高维的(数万维),但与一些操作性DA系统的数百万维相比仍然很小。SDA是否能很好地扩展到这样的应用是一个开放问题,并将面临严峻的工程挑战。

我们工作的另一个局限性是假设系统的动力学被所有轨迹共享。特别是,如果使用参数化物理模型,假设所有轨迹共享相同的参数。因此,SDA不适用于也需要拟合模型参数的设置,或至少需要进一步开发。一些方法[57-60]解决了这个任务,但它们仍然限于低维设置。此外,如果使用物理模型生成合成训练数据,而不是依赖真实数据,人们只能期望SDA与模型本身一样准确。这是任何基于模型的方法共有的局限性,在模型错误指定或分布偏移下的稳健同化留待未来研究。

最后,并非总是需要对整个状态轨迹进行后验推理。在预测任务中,推断动力系统的当前状态就足够了,而且可能成本要低得多。在这种设置下,数据同化简化为状态估计问题,卡尔曼滤波器[61]或其非线性扩展[62,63]等经典方法提供了强大的基线。也提出了许多深度学习方法,完全绕过物理模型,而是仅从过去的观测中学习可能预测的生成模型[64-67]。

相关工作

之前有许多研究调查了使用深度学习来提高数据同化的质量和效率。Mack等人[12]使用卷积自编码器将变分数据同化问题投影到低维空间,这大大简化了优化问题。Frerix等人[14]使用深度神经网络根据观测预测轨迹的初始状态。这个预测然后用作传统(4D-Var)变分数据同化方法的起点,这比随机起点更有效。这种策略对SDA也是可能的(使用SDA采样的轨迹作为起点),并可能帮助覆盖后验分布的多个模式。最后,Brajard等人[17]在只知道观测过程时,解决了同时学习转移动力学和估计轨迹的问题。

除了数据同化之外,SDA与更广泛的序列模型类别密切相关,这些模型已经针对各种类型的数据(包括文本、音频和视频)进行了广泛研究。最新进展表明,基于分数的生成模型在最具挑战性的任务上取得了显著成果。Kong等人[26]和Goel等人[27]使用基于分数的模型以非自回归方式生成长音频序列。Ho等人[23]训练了固定帧数视频的基于分数的生成模型,并以自回归方式使用它生成任意长度的视频。相反,我们的方法是非自回归的,这允许同时生成和条件化所有元素(帧)。有趣的是,作为他们方法的一部分,Ho等人[23]引入了条件采样的"重建引导",这可以看作是我们似然近似(15)的特例,其中观测 y 是 x 的子集。最后,Ho等人[25]生成低帧率、低分辨率的视频,然后用一系列[24]超分辨率扩散模型在时间和空间上进行上采样。这种方法应用于数据同化值得探索,尽管引入任意观测过程似乎具有挑战性。