An Enhanced Generalised Emulation Framework for the Lorenz-96 Inverse Problem

Weiting Yi

St Catherine's College

University of Oxford

A thesis submitted for the degree of

Master of Science in Statistical Science

Michaelmas Term 2022

Abstract

Many challenges are detected in Bayesian inverse problems for numerical weather models. Firstly, there are two significant sources of uncertainty in numerical weather models, one is initial conditions, and the other one is in model representations. Secondly, in such problems, the parameter-to-data forward models are usually computationally expensive to evaluate. A new generalised emulation framework for this type of inverse problem is proposed in this study. It leverages the state-of-art stochastic parameterisation idea in numerical weather modeling to improve computational efficiency as well as enhances preceding data-driven emulators by keeping the partial physics-informed structure of the accurate natural model. Inverse problems in the famous test bed model, the Lorenz-96 system, are introduced to deploy this methodology.

Contents

1 Introduction and literature review **1**

2 Bayesian inverse problems in the Lorenz-96 system **5**

2.1 The idealised continuous-time system 5

2.1.1 ODEs for the Lorenz-96 system 5

2.1.2 An idealised Bayesian inverse problem 6

2.2 Bayesian inverse problems in the discrete-time L96 7

2.2.1 Finite-time averaged data 8

2.2.2 Uncertain initial conditions 8

2.3 The stochastic forecast model 9

2.3.1 An additive AR(1) parameterisation 10

2.3.2 Hyperparameter training framework for autoregressive parameterisations 12

2.4 Approximate Bayesian inversion with stochastic parameterisations 14

2.4.1 Bayesian inversion with the surrogate forward model 14

2.4.2 Learning of the surrogate forward model 15

3 An enhanced CES with stochastic parameterisations **17**

3.1 Overview 17

3.2 Calibration - EKS 20

3.3 Sub-grid progress emulation - Probabilistic Autoregressive models . . 22

3.3.1 A general form of probabilistic autoregressive models for sub-grid processes 23

3.3.2 Probabilistic autoregressive model examples 24

3.3.2.1 Normal-distributed one-step Markov models 24

3.3.2.2 Extensions with deep learning models 25

3.3.3 Training framework 28

3.4 Posterior sampling with the surrogate model 30

4 Numerical experiments with the Lorenz-96 system 32

4.1 Experiment design 33

4.1.1 Synthetic data generation process 33

4.1.2 Parameter settings 34

4.1.3 Inverse problem settings 34

4.2 Important implementation methods 35

4.2.1 Ensemble Kalman Sampling in L96 35

4.2.2 Surrogate forward model evaluation 35

4.3 Numerical results 37

4.3.1 Synthetic data overview 37

4.3.2 Calibration 38

4.3.3 Emulation 39

5 Conclusions and future work 42

A A review of Bayesian inverse problems 44

Bibliography 46

Chapter 1

Introduction and literature review

In the summer of 2022, the United Kingdom experienced record-breaking heatwaves surpassing 40C for the first time. More and more unprecedented extreme weather events like the heatwaves happening in recent years are seen as one of the major effects of climate change, which has drawn the attention of many researchers to weather and climate science. One of the most active research topics in this broad area is numerical weather prediction (NWP) based on mathematical models. NWP has a long-standing history since the early 20th century. The first try with NWP is the manual experiment done by Lewis Fry Richardson in 1921, Britain and the early computer-based forecasts were firstly produced in 1950, Norway ([1]). It also has made great progress lately, enjoying the booming of computational techniques ([2]). Making good weather predictions with NWP is nevertheless a still challenging task due to the chaotic nature of the atmosphere. There exist two main sources of uncertainty in numerical atmospheric models for weather prediction, uncertainty in initial conditions and intrinsic uncertainty of the complex dynamical systems ([3], [4]). NWP is highly sensitive to initial conditions because the atmosphere is a chaotic system following the famous chaos theory by Edward Lorenz [5]. A metaphor for the chaotic phenomena is the butterfly effect stating a butterfly flapping its wings in Brazil can cause a tornado in Texas. Thus the uncertainty in initial conditions needs special attention and is usually handled by data assimilation with ensemble methods which is a particular class of probabilistic inverse problems where the unknown parameters are initial conditions ([6] [7] ,[8]). Intrinsic model uncertainty arises from mathematical representations of the atmosphere with deterministic differential equations. These atmospheric models with differential equations usually have a finite resolution due to computational constraints, and small-scale sub-grid processes in the atmosphere are represented by oversimplified mathematical equations or are even not represented. For each state of resolved variables, there exist many possible states of the unsolved sub-grid processes,

which turns into a significant resource of model uncertainty. This motivates recent development of stochastic parameterisation scheme for sub-grid processes in NWP ([3], [9], [10]).

Although Bayesian inversion and stochastic parametrisation seem like two distinct research areas in NWP, they are indispensable to producing good weather forecasts. And we further find they are also connected.

- **Approximate Bayesian inversion (ABI) tools in NWP**. One of the most challenging issues is numerical weather models, as forward models are usually expensive to elevate, which makes posterior sampling based on exact data likelihood prohibitively expensive. This issue is broadly known as the doubly intractable problem in Bayesian inference, where both exact likelihood and posterior density are intractable. The class of doubly intractable problems in Bayesian inference is usually handled by approximate Bayesian computation (ABC) methods ([11], [12]). Another common stream for approximately solving Bayesian inverse problems with expensive forward models like NWP is to use a cheap surrogate model (emulator) to replace the expensive NWP simulator and construct an approximate posterior distribution that can be efficiently sampled from with the surrogate models such as the Gaussian Process emulator [13]. There are also advanced techniques combining the best of both worlds - adopting the cheap surrogate models in ABC- rejection sampling ([14],[15]).
- **Stochastic parameterisations in NWP**. Recent work such as [3] has accessed the impact of this idea in weather prediction and even climate models for successfully improving initialised forecast reliability and reflecting large-scale climate variability. And deep-learning based techniques such as GAN ([16]) are applied to find pure data-driven parameterisations for sub-grid processes in NWP and thus further improve prediction reliability with the power of data.

However, we notice that seldom work on model calibration for stochastic NWP has been done. On the other hand, there are some connections between emulator-based posterior sampling and stochastic parameterisation for NWP. Firstly, the models are both trained from experimental data based on based on mathematical models instead of real-world data. Secondly, using stochastic parameterisation for NWP bypasses solving differential equations for high-dimensional small-scale variables, which also largely improves computational efficiency as the cheap emulator in approximate Bayesian inversion does.

Spotting these gaps and connections, we innovatively develop an idea bridging stochastic parameterisation and approximate Bayesian inversion with emulation in NWP, that is, to use the atmospheric model with stochastic parameterisations as the cheap surrogate model for approximate Bayesian inversion in NWP. We propose an original ABI solution for NWP with the help of stochastic parameterisations and embed it into a practical emulator-based ABI framework called *Calibrate, Emulate, Sample* (CES) proposed by [13]. The core idea of CES is: firstly, to use the derivative-free ensemble Kalman methods with the exact forward model to find some most likely parameters based on actual data, acting like a data-driven experiment step; secondly, train a Gaussian Process emulator for the exact forward model based on parameters and synthetic data obtained from the first step; thirdly, replace the exact forward model with the GP emulator in MCMC where massive evaluations of the forward model are required. Compared to the original CES, the contribution of our enhanced framework contains,

- We replace the completely data-driven Gaussian Process emulator in the original CES framework with a surrogate forward model based on the class of stochastic atmospheric models with both deterministic and stochastic parameterisations. The new surrogate forward model becomes cheaper to evaluate by replacing computationally intensive computations of solving for high-dimensional sub-grid processes. It also enhances classic data-driven emulators such as the GP emulator used in the original CES work of [13] by keeping part of the physics-informed structure of the original expensive exact forward model.
- We design a Bayesian hierarchical structure (BHM) to represent the Bayesian inverse problem with the surrogate forward model with stochastic parameterisations by treating the stochastic approximations for sub-grid processes as latent variables. With the BHM structure, the posterior density for Bayesian inverse problems with our proposed surrogate forward model is explicitly defined.
- We propose a general form of probabilistic autoregressive models as stochastic parameterisations for modeling sub-grid processes, with which likelihood-based posterior inference is tractable. We give several examples of probabilistic autoregressive models, including some state-of-art deep learning methods.

This study aims to introduce the new general emulator-based ABC framework for NWP. While the ultimate goal is to apply this framework to a broad class of numerical weather and climate models, such as the state-of-art full general circulation model (GCM), as a proof of concept, we set the Bayesian inverse problem in a two-scale

toy model for the atmosphere proposed in Edward Lorenz in 1996 ([3]). Numerical experiments are also done with the Lorenz- 96 system. This toy model has been popularly used as a test bed in research in stochastic parameterisation schemes ([3], [9], [17]) and Bayesian inverse problems ([18], [19], [13]) for NWP.

The following sections are organised as: Chapter 2 introduces Bayesian inverse problems in the Lorenz-96 system; Chapter 3 introduces the enhanced CES framework with stochastic parameterisations; Chapter 4 presents our numerical experiments; Chapter 6 contains conclusions and discussions of future work.

Chapter 2

Bayesian inverse problems in the Lorenz-96 system

2.1 The idealised continuous-time system

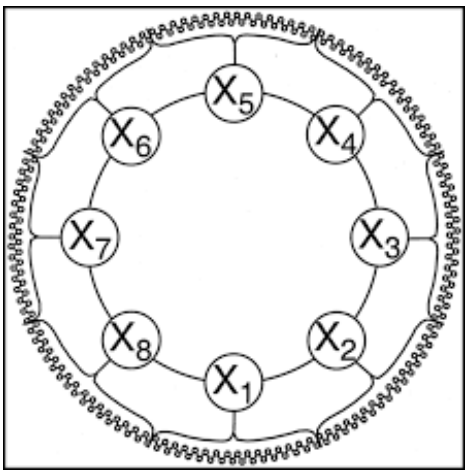
2.1.1 ODEs for the Lorenz-96 system

The Lorenz 96 system (L96, in short) is a simple toy model of the atmosphere, incorporating interaction of multi-scale variables. The ordinary differential equations for L96 given below describe a coupled dynamical system of equations for two types of variables arranged along a latitude circle. We let X denote the large-scale low-frequency variables and Y denote the small-scale high-frequency variables and it follows,

$$\begin{aligned} \frac{\mathrm{d}X_k}{\mathrm{d}t} &= -X_{k-1}\left(X_{k-2} - X_{k+1}\right) - X_k + F - hc\overline{Y}_k; \quad k = 1, \ldots, K \\ \frac{1}{c}\frac{\mathrm{d}Y_{l,k}}{\mathrm{d}t} &= -bY_{l+1,k}\left(Y_{l+2,k} - Y_{l-1,k}\right) - Y_{l,k} + \frac{h}{L}X_k; \quad j = 1, \ldots, JK \end{aligned} \tag{2.1}$$

where $\overline{Y}_k = \sum_{l=1}^{L} Y_{l,k}$ for $k = 1, \ldots, K$ are the sub-grid tendencies. Each one of the fast variable $X_k, (k = 1, \ldots, K)$ is coupled with L fast variables $Y_{l,k}, (l = 1, \ldots, L)$ and they all have cyclic boundary conditions: $X_{k+K} = X_k, Y_{l,k+K} = Y_{l,k}$, and $Y_{l+L,k} = Y_{l.k+1}$. [20] gives some possible geophysical interpretations for the system such as the fast variables can represent convective events while the slow variables can represent larger scale synoptic event. The model includes some other parameters: the number of slow variables K, the number of fast variables L, coupling constant h, force term F, spatial-scale ratio b, and time-scale ratio c. The first two parameters are fixed as K = 8 and L = 32, consistent with the numerical experiments done in [21] A schematic of this system is shown in Figure 2.1. For convenience, we use $\theta := (h, F, \log c, b)'$ to denote the vector of parameters of our interest in inverse problems. The logarithmic scale is used for the time-scale ratio c because it is restricted to be positive.

Figure 2.1: The schematic of the L96 system with K = 8 slow variables and L = 32 fast variables ([3])



2.1.2 An idealised Bayesian inverse problem

For inverse problems in climatology, data usually is only available in time-averaged form. We suppose the Bayesian inversion for L96 is performed on data $\mathbf{y}$ averaged across K locations and over time windows of length τ. (more descriptions of the data) The above dynamical system consists of several continuous-time dynamical processes $\{X_k(t)\}$ and $\{Y_{l,k}(t)\}$ for $k \in \{1, \ldots, K\}$ and $l \in \{1, \ldots, L\}$. To formulate the inverse problem with time-averaged data, we need to introduce a new observation operator that links the dynamical processes with the data. For ease of notation, we use $\{Z_k(t)\}$ to denote the dynamical processes for the slow variable and fast variables at location k with $Z_k(t) = (X_k(t), \ldots, Y_{1,k}(t), \ldots, Y_{L,k}(t))^T \in \mathbb{R} \times \mathbb{R}^L$. The K vectors $Z_k(t)$ for $k = 1, \ldots, K$ together contain all information at time t of the dynamical system. The observation vector $\phi_k(t) \in \mathbb{R}^5$ at location k and time t is then defined with the observation operator $\varphi : \mathbb{R} \times \mathbb{R}^L \longrightarrow \mathbb{R}^5$ as,

$$\phi_k(t) := \varphi\left(z_k(t)\right) = \left(X_k(t), \overline{Y_k(t)}, X_k(t)^2, X_k(t)\overline{Y_k(t)}, \overline{Y_k(t)^2}\right)^T \tag{2.2}$$

where $\overline{Y_k(t)} = \sum_{l=1}^{L} Y_{l,k}(t)$ and $\overline{Y_k(t)^2} = \sum_{l=1}^{L} Y_{l,k}(t)^2$. Let z_0 denote the initial state of $\{Z_k(t)\}$ for $k = 1, \ldots, K$ in the L96 system, the forward operator for continuous-time L96 can be defined as,

$$\mathcal{G}_\tau(\theta; z_0) = \frac{1}{\tau} \int_{T_0}^{T_0+\tau} \left(\frac{1}{K} \sum_{k=1}^{K} \Phi(Z_k(s)) \right) ds \tag{2.3}$$

where T_0 is a predefined spinup time, τ is the time horizon for the time-averaging data, and z_0 is the initial state of the system.

Under idealised settings, the time horizon τ is long enough such that the initial state z_0 can be forgotten. Following the analogue of the Law of Large Number for ergodic dynamical systems (given in the appendix), we can define an infinite-horizon operator based on Equation 2.3,

$$\mathcal{G}_\infty(\theta) := \lim_{\tau \longrightarrow \infty} \mathcal{G}_\tau(\theta; z_0) = \frac{1}{K} \sum_{k=1}^{K} \int_{\mathcal{A}} \Phi(z_k) \mu(dz_k; \theta) \tag{2.4}$$

where $\mathcal{A}$ is the compact attractor for L96 and is assumed to be invariant with respect to the measure $\mu(dz_k; \theta)$ for $k = 1, \ldots, K$.

With the above notations, the inverse problem for the idealised continuous-time L96 is given as,

$$\mathbf{y} = \mathcal{G}_\infty(\theta) + \eta \tag{2.5}$$

where the observational noise η is assumed to be normally distributed with $\eta \sim N(0, \Gamma_\mathbf{y})$ and θ follows a prior distribution $N(0, \Gamma_\theta)$. The data vector $\mathbf{y} \in \mathbb{R}^5$ now refers to an observation at a single time points, however, usually data measured at multiple time points are available. The only source of uncertainty in the forward model $\mathcal{G}_\infty \theta$is θ, the vector of parameters to be inferred via Bayesian inversion. The posterior distribution for θ given data $\mathbf{y}$ is simply given as,

$$\pi^{(a)}(\theta|\mathbf{y}) \propto \exp(-\Phi_R^{(a)}(\theta|\mathbf{y})), \quad \text{with } \Phi_R^{(a)}(\theta|\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathcal{G}_\infty(\theta)\|^2_{\Gamma_m} + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}, \tag{2.6}$$

where $\Phi_R^{(a)}(\theta|\mathbf{y})$ is known as the potential of the dynamical system seeing the function behaves like a potential energy. The derivation of the posterior density function for Bayesian inverse problems is shown in Appendix A.

Equation 2.5 and Equation 2.6 are, however, just some idealised results. Firstly, the L96 system is usually solved numerically so we do not have access to the forward evaluations of analytical Equation 2.3 and Equation 2.4. Secondly, it is impractical to obtain time-averaging data over infinite-time horizons and thus initial conditions usually play a role in forward model evaluations. In the following sections, we will introduce some practical formulations of Bayesian inverse problems for the L96 system.

2.2 Bayesian inverse problems in the discrete-time L96

In practice, the Lorenz-96 system does not have analytical solutions and can only solved with numerical methods such as the adaptive fourth-order Runge–Kutta time-stepping scheme (RK4). Thus we only have access to discrete trajectories of the

dynamical system instead of the continuous $X_k(t)$ and $Y_{l,k}(t)$. Thus we adopt discrete notations in the following sections. The discrete time series obtained by numerically integrating Equation 2.1 with RK4 using a small time step of δt model time units (MTU) are denoted as $\{X_{t_i,k}\}$ for $k = 1, \ldots, K$ and $\{Y_{t_i,l,k}\}$ for $k = 1, \ldots, K, l = 1, \ldots, L$ with a subscript t_i as the discrete time index such that $\delta t = t_{i+1} - t_i$.

2.2.1 Finite-time averaged data

Consistent with the continuous case, the inversions are also performed on noisy time-averaged data. Due to computational constraints in reality, the forward model evaluation can only be done using a finite number of discrete points. We suppose the data $\mathbf{y}$ for inversions is averaged across K locations and over time windows of length T, during which N_T discrete time points indexed by t_i are available. The observation operator Equation 2.2 in discrete case is redefined as,

$$\begin{aligned} \phi_{t_i,k} &= \varphi_k(Z_{t_i,k}) := \varphi\left(X_{t_i,k}, Y_{t_i,1,k}, \ldots, Y_{t_i,L,k}\right) \\ &= \left(X_{t_i,k}, \overline{Y_{t_i,k}}, X^2_{t_i,k}, X_{t_i,k}\overline{Y_{t_i,k}}, \overline{Y^2_{t_i,k}}\right)^T \quad i = 1, \ldots, N_T \end{aligned} \tag{2.7}$$

where $Z_{t_i,k}, \overline{Y_{t_i,k}}, \overline{Y^2_{t_i,k}}$ are respectively the discrete counterparts of $Z_k(t_i), \overline{Y_k}(t_i), \overline{Y^2_k}(t_i)$ in Equation 2.2.

Approximating the integral of continuous-time processes in Equation 2.3 with the sum of discrete-time variables, the finite-time forward operator for discrete L96 is defined as,

$$\mathcal{G}_T(\theta; z_0) = \frac{1}{N_T}\frac{1}{K}\sum_{i=1}^{N_T}\sum_{k=1}^{K}\Phi(Z_{t_i,k}) \tag{2.8}$$

with which, we obtain the standard form of inverse problem same as Equation A.1,

$$\mathbf{y} = \mathcal{G}_T(\theta; z_0) + \eta \tag{2.9}$$

where the measurement noise $\eta \sim N(0, \Gamma_m)$. As the time horizon is finite, the initial state z_0 plays a role in the forward model as an input. When z_0 is known and fixed, the posterior distribution for θ is given as,

$$\pi^{(1)}(\theta|\mathbf{y}) \propto \exp(-\Phi_R^{(1)}(\theta|\mathbf{y})), \quad \text{with } \Phi_R^{(1)}(\theta|\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathcal{G}_T(\theta; z_0)\|^2_{\Gamma_\mathbf{y}} + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}, \tag{2.10}$$

2.2.2 Uncertain initial conditions

There is a non-negligible issue with the time-averaged data with unknown initial states. In the infinite-horizon setting, the initial condition is trivial to the system.

But when the data is averaged for a finite time horizon, uncertain initial values may give noisy forward model evaluations, which is not of our intrinsic interest. By an analogy of central limit theorem, we can replace the forward map $\mathcal{G}_T(\theta; z_0)$ conditioned on the initial conditions $z0$ with the ergodic average defined in Equation 2.4 as,

$$\mathcal{G}_T(\theta; z_0) \approx \mathcal{G}_\infty(\theta) + \sigma, \quad \text{with } \sigma \sim N(0, \Sigma(\theta)) \tag{2.11}$$

where $\Sigma(\theta) \in \mathbb{R}^5 \times \mathbb{R}^5$ is concerned with the internal variability of the system with unknown conditions, and $\mathcal{G}_\infty(\theta)$ is the operator defined in Equation 2.4.

With Equation 2.9 and Equation 2.11 together, we can combine the measurement noise and uncertainty of the initial condition into a covariance matrix $\Gamma_{\mathbf{y}} = \Gamma_m + \Sigma(\theta)$. The inverse problem becomes,

$$\mathbf{y} = \mathcal{G}_\infty(\theta) + \epsilon, \quad \text{with } \epsilon \sim N(0, \Gamma_{\mathbf{y}}) \tag{2.12}$$

and we have the posterior distribution as Equation A.4 with $\mathcal{G}(\theta)$ replaced by $\mathcal{G}_\infty(\theta)$

$$\pi^{(2)}(\theta|\mathbf{y}) \propto \exp(-\Phi_R^{(2)}(\theta)), \quad \text{with } \Phi_R^{(2)}(\theta) = \frac{1}{2}\|\mathbf{y} - \mathcal{G}_\infty(\theta)\|^2_{\Gamma_y} + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}, \tag{2.13}$$

The above representation for uncertainty in initial conditions has been used in [13]. We will tackle this uncertainty in initial conditions in a different manner with the new methodology framework.

2.3 The stochastic forecast model

The main source of internal uncertainty with the Lorenz-96 system is the approximation of unsolved sub-grid processes through deterministic parameterisations. Traditional deterministic parameterisation schemes are able to represent the sub-grid processes at a given resolvable scale. In the natural model we present, only two scales of variables X and Y are resolved. Due to limiting computing power, model errors can only be reduced to a certain degree by using parameterisations at smaller and finer scales, but they cannot be eliminated [16]. Stochastic parameterisations are motivated to produce robust forecasts for a lot of chaotic models against dynamical systems with internal uncertainties that cannot be captured by their deterministic ODE parameterisations. Stochastic parameterisations for the Lorenz-96 system are firstly introduced by [3] and further developed in [16] with the state-of-art GAN tools. Besides representing the model internal uncertainty, stochastic parameterisations for L96 can help to improve computational efficiency when making forecasts. Massive

computations are needed for solving the fast variables at small time steps with the deterministic parameterisation, which can be avoided when using stochastic approximations for the sub-grid tendencies. We also notice, for Bayesian inversions, a large number of forward model evaluations are required, making the computation burden even larger when using the deterministic parameterisation. We call the deterministic system above "the natural model", and the model with stochastic parameterisations "the stochastic forecast model".

More explicitly, stochastic parameterisations are applied to model the sub-grid tendencies for L96 so that the extensive computations for solving these fast variables $Y_{l,k}$ can be bypassed. Let $X_k^*(t)$ and $U_k^*(t)$ denote the predicted values for $X_k(t)$ and $U_k(t) = \frac{hc}{b}\sum_l Y_{l,k}(t)$ with the stochastic forecast model. Parallel to the natural model for slow variables in Equation 2.1, a generic form of the stochastic forecast model is given as follows,

$$\frac{\mathrm{d}X_k^*(t)}{\mathrm{d}t} = -X_{k-1}^*(t)\left[X_{k-2}^*(t) - X_{k+1}^*(t)\right] - X_k^*(t) + F - U^p(X_k^*(t)); \quad k = 1, \ldots, K \tag{2.14}$$

where $U^p(\cdot)$ is some stochastic parameterisation for the sub-grid tendencies $U_k^*(t)$. We notice Equation 2.14 is a stochastic differential equation (SDE) for the large-scale variables X_k^*. Because X_k is low-frequency compared to $Y_{l,k}$, we only need to integrate Equation 2.14 with some lower order numerical methods and larger time steps, requiring less computing power. A piece-wise deterministic, adaptive second-order Runge-Kunata scheme with a larger time step Δt is adopted, in which the stochastic noise term is held constant over the time step. The RK2 scheme has been shown to converges to the traditional Stratonovich forward integration scheme for stochastic differential equations, see [22] and [23] for further details. We then move to the discrete time notation with time series $\{X_{t'_i,k}^*\}$ and $\{U_{t'_i,k}^*\}$ indexed by a different time subscript t'_i such that $\Delta t = t'_{i+1} - t'_i$. Different time indexes are applied to the trajectories of the natural model and stochastic forecast model because they are integrated with different time steps, thus numerical trajectories are with different time intervals δt and Δt.

2.3.1 An additive AR(1) parameterisation

To complete the stochastic forecast model, we need to find the proper stochastic parameterisations $U_p(\cdot)$. The stochastic parameterisations introduced by [21] are generally decomposed into a deterministic mean characterised by functions of $X_{t',k}^*$

and a zero-mean stochastic component,

$$U^*_{t'_i,k} = U^d(X^*_{t'_i,k}) + r_{t'_i,k} \quad \text{for } t'_i = t'_1, t'_2, \ldots \text{and } k = 1, \ldots, K \tag{2.15}$$

The deterministic part $U^d(\cdot)$ is set as a polynomial function in $X^*_{t'_i,k}$ with a vector of unknown parameters $\beta = (b_0, b_1, b_2, b_3)^T$,

$$U^d(X; \beta) = b_0 + b_1 X + b_2 X^2 + b_3 X^3 \tag{2.16}$$

where we drop the indexes t'_i and k of $X^*_{t'_i,k}$ without loss of generality. There are many ways to model the stochastic component $r_{t'_i,k}$ while a common structural assumption is it is serial-correlated. Spatial correlations have been discussed in related literature ([9], [21]) but they choose to disregard them in the L96 system for simplicity and argue in more complicated systems including both spatial correlations and serial correlations can be helpful to explain the sub-gridscale variability. The first-order auto-regressive model is used in to characterise the serial correlations of the stochastic component. The AR(1) stochastic component together with the polynomial deterministic component compose the PolyAR1 parameterisation. It has been verified to be satisfactory for the L96 through numerical experiments and selected as a stringent benchmark in the related literature. The full PolyAR1 parameterisation is given as,

$$\begin{aligned} U^*_{t'_i,k} &= U^d(X^*_{t'_i,k}; \beta) + \epsilon_{t'_i,k} \\ \epsilon_{t'_i,k} &= \gamma \epsilon_{t'_i-1,k} + \sigma_\epsilon (1-\gamma^2)^{1/2} z_{t'_i,k} \end{aligned} \tag{2.17}$$

where $z_{t'_i,k} \sim N(0,1)$. This is nothing special, just a polynomial regression model with serial correlations of order 1. The unknown parameters β in $U^d(X^*_{t'_i})$ can be simply estimated with ordinary least square (OLS) and the estimates for γ and σ_ϵ can be obtained by fitting the residuals of the polynomial regression with the AR(1) model. We notice Equation 2.17 can be re-arranged as,

$$U^*_{t'_i,k} = \gamma U^*_{t'_i-1,k} + U^d(X^*_{t'_i,k}; \beta) - \gamma U^d(X^*_{t'_i-1,k}; \beta) + \sigma_\epsilon (1-\gamma^2)^{1/2} z_{t'_i,k} \tag{2.18}$$

where it specifies a normal distribution model for the conditional density for the sub-grid tendencies,

$$U^*_{t'_i,k} | U^*_{t'_i-1,k}, X^*_{t'_i,k}, X^*_{t'_i-1,k} \sim N\left(\gamma U^*_{t'_i-1,k} + U^d(X^*_{t'_i,k}; \beta) - \gamma U^d(X^*_{t'_i-1,k}; \beta), \sigma^2_\epsilon (1-\gamma^2)\right) \tag{2.19}$$

This provides us with another perspective to understand stochastic parameterisations for L96. With the serial correlation assumption, stochastic parameterisations for the sub-grid tendencies can be seen as some auto-regressive generative models

$p_\psi(\cdot)$ parameterised by ψ for the time series $\{U^*_{t'_i,k}\}$ with a series of covariates $\{X^*_{t'_i,k}\}$. To not be confused with the parameter vector θ for Bayesian inversions, we call the vector of parameters ψ in the stochastic parameterisations as "hyper-parameters". Note that, hyperparameters here are not same as hyperparameters in machine learning which can not be learned through training. In the PolyAR(1) parameterisation, ψ includes $\beta, \gamma, \sigma_\epsilon$. Other powerful auto-regressive generative models can be flexibly replaced in the stochastic forecast model of Equation 2.14, corresponding to different stochastic parameterisations. We will further discuss other applications of other autoregressive models in later sections.

2.3.2 Hyperparameter training framework for autoregressive parameterisations

With the above formulation, parameter estimation for stochastic parameterisations of L96 can be seen as hyper-parameter training for corresponding auto-regressive models $p_\psi(\cdot)$. Here, we use the PolyAR(1) parameterisation as an example to show how to learn the hyper-parameters.

As we do not have exact solutions to the idealised continuous L96 system, the numerical trajectories obtained from the solving the natural model can be seen as some close approximations to the continuous system. In other words, the time series $\{X_{t_i,k}\}$ and $\{Y_{t_i,l,k}\}$ obtained by solving the natural model of Equation 2.1 with RK4 can be treated as some groundtruth series when learning hyper-parameters for the stochastic forecast models. More specifically, realisations of sub-grid tendencies $\{U_{t'_i,k}\}$ in the stochastic forecast model with the PolyAR1 parameterisation can be closely approximated with discretised trajectories $\{X_{t'_i,k}\}$ of the natural model by the finite difference method as,

$$U_{t'_i,k} \approx \left[-X_{t'_i,k-1}\left(X_{t'_i,k-2} - X_{t'_i,k+1}\right) - X_{t'_i,k} + F\right] - \left(\frac{X_{t'_{i+1},k} - X_{t'_i,k}}{\Delta t}\right) \tag{2.20}$$

where $X_{t'_i,k}$ is the ground truth slow variable obtained by solving the natural model at time index t'_i from the stochastic forecast model. As mentioned earlier, the time interval Δt between t'_i and t'_{i+1} is larger than δt between t_i and t_{i+1}. The time series $\{U_{t'_i,k}\}$ and $\{X_{t'_i,k}\}$ can then be used to estimate the unknown parameters $\psi = (\beta, \gamma, \sigma_\epsilon)^T$ in PolyAR1. Estimation can be done by fitting the first equation of Equation 2.17 with OLS and fitting the regression residuals with the AR(1) respectively.

An alternative way to estimate ψ is using the maximum likelihood method (MLE). The conditional density specified in Equation 2.19 can be used to construct the likelihood function for the hyperparameters. Suppose there are N training points of $\{U_{t'_i,k}\}$

and $\{X_{t'_i,k}\}$ available. Conditioning on $X_{t_0,k}$ and $U_{t_0,k}$, the joint density for $\{U^*_{t'_i,k}\}$ is given as,

$$p(U^*_{t'_1:t'_N,k}|X_{t_0,k}, U_{t_0,k}, \psi) = \prod_i^N N(U^*_{t'_i,k}|\mu_{t'_i}, \sigma^2_{t'_i}) \tag{2.21}$$

where $N(\cdot|\mu, \sigma^2)$ is the density function for a normal distribution with mean μ and variance σ^2. The functions for the conditional mean $\mu_{t'_i}$ and variance $\sigma^2_{t'_i}$ are specified by PolyAR1 and parameterised by ψ as,

$$\begin{aligned} \mu_{t'_i} &= \gamma U^*_{t'_i-1,k} + U^d(X^*_{t'_i,k}; \beta) - \gamma U^d(X^*_{t'_i-1,k}; \beta) \\ \sigma^2_{t'_i} &= \sigma^2_\epsilon(1-\gamma^2) \end{aligned} \tag{2.22}$$

We hence obtain the negative log-likelihood function for ψ with respect to the realised series $\{U_{t'_i,k}\}$, $\{X_{t'_i,k}\}$ as,

$$\begin{aligned} -\log \mathcal{L}(\psi) = &\frac{1}{2\sigma^2_\epsilon(1-\gamma^2)} \sum_{i=1}^N \sum_{k=1}^K \left[U_{t'_i,k} - \gamma U_{t'_i-1,k} + U^d(X_{t'_i,k}) - \gamma U^d(X_{t'_i-1,k})\right]^2 \\ &+ \frac{KN}{2} \log(\sigma^2_\epsilon(1-\gamma^2)) + Const \end{aligned} \tag{2.23}$$

MLE estimates for hyperparameters ψ in PolyAR can be found by minimised the above log-likelihood function.

In the context of machine learning, parameter estimation via fitting statistical models is called model training and the negative log-likelihood function in MLE can be seen as the loss function for training. Hence estimating ψ in PolyAR1 boils down to training the autoregressive model specified by Equation 2.21 and Equation 2.22 with the loss function in Equation 2.23. And training data are available from numerical trajectories of the natural model. This provides us with a practical framework to find the stochastic parameterisations for L96 and other climate models through the idea of hyperparameter training,

- Step1. Solve the natural model in Equation 2.1 with small time-steps δt and high-order numerical methods such as RK4 and take the resulted trajectories $\{X_{t_i,k}\}$ as ground truth series.

- Step2. Construct realisations $\{U_{t'_i,k}\}$ and $\{X_{t'_i,k}\}$ for $\{U^*_{t'_i,k}\}$ and $\{X^*_{t'_i,k}\}$ in the stochastic forecast model with the ground truth series $\{X_{t_i,k}\}$ as Equation 2.20

- Step3. Train the autoregressive model $p_\psi(\cdot)$ specified by the stochastic parameterisation with a corresponding loss function and the training data. (With PolyAR(1), this step is to find the minimiser $\hat{\psi}$ for Equation 2.23).

This hyperparameter training framework has been used in stochastic parameterisation literature such as [21] [16]. We summarise it differently in the language of statistical learning, for convenience of applying it to Bayesian inverse problems later. The framework offers flexibility to replace the autoregressive model specified by PolyAR1 with other powerful deep autoregressive models. Moreover, it provides explicit joint density for sub-grid tendencies which is of great value to posterior sampling in Bayesian inverse problems. We will come back to how to apply and implement this framework in Bayesian inverse problems in the later sections.

2.4 Approximate Bayesian inversion with stochastic parameterisations

Approximate Bayesian inference tools are commonly used in Bayesian inverse problems for L96 because its forward model evaluation requires expensive computations for of solving for sub-grid processes. Especially, massive forward evaluations are needed in the posterior sampling step, making the computation for Bayesian inversions prohibitively expensive and even intractable. We come up with a new approximate Bayesian inversion idea of replacing the forward model based on the natural model with some surrogate models based on the stochastic forecast model. For simplicity, we call the forward model Equation 2.8 constructed with the natural model the natural forward model and the one based on the stochastic forecast model of Equation 2.14 the surrogate forward model, enhanced by stochastic parameterisations of the L96 system.

2.4.1 Bayesian inversion with the surrogate forward model

We firstly introduce a Bayesian inverse problem with the enhanced surrogate forward model $\mathcal{G}_T^*(\theta; z_0)$. Leaving the sub-gird processes unsolved, $\{\overline{Y}_{t_i,k}\}$ and $\{\overline{Y^2}_{t_i,k}\}$ in the observational series $\{\varphi_{t_i,k}\}$ defined in Equation 2.7 are replaced by the stochastic appoximations $\{U^*_{t'_i,k}\}$ and $\{V^*_{t'_i,k}\}$ obtained by solving the stochastic forecast model. Although $\{V^*_{t'_i,k}\}$ has not been introduced in the previous stochastic model, the training framework for $\{U^*_{t'_i,k}\}$ can be directly used to train an autoregressive parameterisation $V_p(\cdot)$ without extra formulation. A new stochastic observational vector is then defined as,

$$\phi^*_{t'_i,k} := \varphi(X^*_{t'_i,k}, U^*_{t'_i,k}, V^*_{t'_i,k}) = (X^*_{t'_i,k}, U^*_{t'_i,k}, X^{*2}_{t'_i,k}, X^*_{t'_i,k}U^*_{t'_i,k}, V^*_{t'_i,k}) \in \mathbb{R}^5 \tag{2.24}$$

where $X^*_{t'_i,k}$ is obtained by numerically solving the stochastic differential equation in Equation 2.14 conditional on $X^*_{t'_{i-1},k}$ and $U^*_{t'_{i-1},k}$.

Suppose there are N'_T time intervals indexed by t'_i within a time window of length T. Parallel to Equation 2.8 and Equation 2.9 in the deterministic case, we formally define the surrogate forward model and rewrite the inverse problem as,

$$\mathbf{y} = \mathcal{G}^*_T(\theta; z_0) + \eta, \quad \text{with } \mathcal{G}^*_T(\theta; z_0) = \frac{1}{N'_T}\frac{1}{K}\sum_{i=1}^{N'_T}\sum_{k=1}^{K}\Phi(X^*_{t'_i,k}, U^*_{t'_i,k}, V^*_{t'_i,k}) \tag{2.25}$$

where $\eta \sim N(0, \Gamma_m)$ and $\mathcal{G}^*_T(\theta; z_0)$ is the cheap surrogate forward model for $\mathcal{G}_T(\theta; z_0)$.

Albeit $\mathcal{G}^*_T(\theta; z_0)$ is efficient to evaluate, it is a stochastic approximation of the natural forward model, casting the challenge to directly define the posterior distribution for θ. There are three resources of uncertainty within the surrogate model $G^*_T(\theta; z_0)$: (1) uncertain θ, (2) uncertain z_0 (3) internal uncertainty driven by $U^*_{t'_i,k}$ and $V^*_{t'_i,k}$. We notice the latter two sources of uncertainty are not interesting to posterior sampling of our inverse problem and can be regarded as latent variables. With a Bayesian hierarchical representation with latent variables for the inverse problem, the posterior function can still be explicitly defined as long as probability density functions in all hierarchical layers are explicitly defined. We will come back to a derivation of the posterior distribution with a Bayesian hierarchical structure later.

2.4.2 Learning of the surrogate forward model

The surrogate forward model with stochastic parameterisations can be decomposed into three parts: (i) stochastic parameterisations for $U^*_{t'_i,k}$ and $V^*_{t'_i,k}$; (ii) the stochastic differential equation for $X^*_{t'_i,k}$; (iii) the observation operator mapping $X^*_{t'_i,k}$, $U^*_{t'_i,k}$ and $V^*_{t'_i,k}$ to the observational vector. The latter two parts are deterministic and respectively defined by Equation 2.14 and Equation 2.24. Hence learning the surrogate forward model boils down to learning the stochastic parameterisations, equivalently, autoregressive models for $\{U^*_{t'_i,k}\}$ and $\{V^*_{t'_i,k}\}$. The only difference is in inverse problems, the model parameters θ is the one to be inferred and the forward model takes θ as an input. Thus θ should always enter as an input in the stochastic parameterisations as well. The modification is simply done by adding θ as a vector of covariates in the autoregressive models. For example, Equation 2.21 can be modified as,

$$\begin{aligned} &p_\psi(U^*_{t'_1:t'_N,k}|X_{t_0,k}, U_{t_0,k}, \theta) = \prod_i^N N(U^*_{t'_i,k}|\mu_{t'_i}, \sigma^2_{t'_i}) \\ &\text{with} \quad \mu_{t'_i} = \mu_\psi(U^*_{t'_{i-1},k}, X^*_{t'_i,k}, X^*_{t'_{i-1},k}, \theta) \quad \text{and,} \quad \sigma_{t'_i} = \sigma_\psi(U^*_{t'_{i-1},k}, X^*_{t'_i,k}, X^*_{t'_{i-1},k}, \theta) \end{aligned} \tag{2.26}$$

where $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ are two deterministic functions $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ specified by the stochastic parameterisations.

We notice our surrogate forward model can be applied to *Calibrate, Emulate, Sample* (CES, in short), a practical framework for approximate Bayesian computation proposed by [13]. Unlike traditional ABC solutions learning complete data-driven surrogate models for $\mathcal{G}_T(\theta; z_0)$, we only need to learn a data-driven autoregressive model for $\{U^*_{t'_i,k}\}$ and $\{V^*_{t'_i,k}\}$. The other parts of $\mathcal{G}^*_T(\theta; z_0)$ remain the same as $\mathcal{G}_T(\theta; z_0)$. It thus keeps the structural information when solving for large-scale variables while improves computational efficiency largely by replacing the sub-grid tendencies with stochastic approximations. Hence it can be seen as enhanced version of the traditional CES framework. Moreover, the hyper-parameter training framework for the autoregressive models introduced earlier can be perfectly embedded into a modified CES framework. This completes a new methodology of ABC for the Lorenz-96 system, which can be also applied to broad class of weather and climate models with stochastic parameterisations. We are thus motivated to introduce a general form of our methodology in the next chapter.

Chapter 3

An enhanced CES with stochastic parameterisations

In this section we step out to a more general NWP setting. The Lorenz-96 system can be seen a special case of the general framework and we will discuss links with this enhanced framework with the Lorenz-96 system again in Section 4.

3.1 Overview

Consider a general Bayesian inverse problem in weather modeling with data $\mathbf{y} \in \mathbb{R}^d$ and a vector of unknown parameters $\theta \in \mathbb{R}^p$,

$$\mathbf{y} = \mathcal{G}(\theta; z_0) + \eta \tag{3.1}$$

where z_0 is the initial state and the measurement noise $\eta \in \mathbb{R}^d \sim N(0, \Gamma_{\mathbf{y}})$ and the prior $\theta \in \mathbb{R}^p \sim N(0, \Gamma_\theta)$. The forward model $\mathcal{G} : \mathbb{R}^p \to \mathbb{R}^d$ is some numerical climate or weather model that is expensive to evaluate.

The original *Calibrate, Emulate, Sample* of [13] largely improves the computational efficiency of posterior sampling in Bayesian problems with expensive-to-evaluate forward models such as numerical weather or climate models. The CES framework is composed of three steps:

- **Calibration**: find an approximate posterior sample of unknown parameters θ in the inverse problem with the ensemble Kalman sampler (EKS) using the real data and the (expensive) exact forward model;

- **Emulation**: emulate a computationally-cheap surrogate model such as *Gaussian Processes* with the sample of θ and corresponding synthetic forward model evaluations obtained at the calibration step;

- **Sampling**: replace the exact forward model with the surrogate model and sample from the posterior distribution through MCMC or other sampling methods

The core idea of CES is to keep a high-resolution but expensive forward model at the calibration step because ensemble Kalman sampling only requires a few evaluations of $\mathcal{G}(\theta)$ and does not need to calculate the derivatives of $\mathcal{G}(\theta)$. The forward evaluation values and approximate posterior sample of θ obtained at the calibration step can be used to train a cheap GP emulator for the forward model. In the sampling step, which may need many iterations, only requires evaluations of the GP emulator obtained in the emulation step. Moreover, Gaussian Processes emulators learned in second step can emulate inherent noise or uncertainty regarding the forward model evaluation. Assuming prior and observational noise both Gaussian, their structure demonstrates a cheap and elegant form of surrogate likelihoods for MCMC by enjoying the Gaussian conjugacy properties. But we notice some major drawbacks: (1)The Gaussian assumptions are generally strong and not as practical as they may appear; (2)Emulation for the whole noisy forward model with machine learning models sacrifices all information.

We come up with an enhanced version of CES with the help of stochastic parameterisations for numerical weather and climate models to overcome the two drawbacks of the original CES. As mentioned earlier, stochastic parameterisations are popular with numerical weather and climate models for capturing uncertainties in unresolved processes as well as improving computational efficiency. Models with deterministic physical parameterisations can be run with high resolutions and known as natural models. Models with stochastic parameterisations only need to be solved for large-scale major processes and use stochastic approximations for the unsolved sub-grid processes. For simplicity, we call the forward model constructed with natural models the natural forward model, denoted by $\mathcal{G}^N(\theta)$ and the one based on hybrid-mode stochastic forecast models the surrogate forward model, denoted by $\mathcal{G}^S(\theta)$. Our enhanced CES uses the exact but expensive $\mathcal{G}^N(\theta)$ in the calibration step and the efficient surrogate model $\mathcal{G}^S(\theta)$ in the posterior sampling step.

With $\mathcal{G}^N(\theta)$, the inverse problem in Equation 3.1 is simply given by replacing the forward model with $\mathcal{G}^N(\theta)$. The corresponding posterior distribution for θ is given as,

$$\pi^N(\theta|\mathbf{y}) \propto \exp(-\Phi_R^N(\theta|\mathbf{y})), \quad \text{with } \Phi_R^N(\theta|\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathcal{G}^N(\theta)\|^2_{\Gamma_\mathbf{m}} + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta} \tag{3.2}$$

with the same Gaussian prior and measurement noise assumptions. The calibration step with $\mathcal{G}^N(\theta)$ is able to produce an approximate sample of the posterior distribution

defined in Equation 3.2. The inverse problem with the surrogate forward model is written as,

$$\mathbf{y} = \mathcal{G}^S(\theta; z_0) + \eta \tag{3.3}$$

while the posterior distribution for θ is not as simple as before because stochastic parameterisations introduce unwanted uncertainty to the surrogate forward model. Luckily, the internal uncertainty is purely driven by the stochastic forecasts for unsolved sub-grid progresses, the joint density of which can be specified by the stochastic parameterisations. Suppose we have q unsolved sub-grid processes. We thus treat the q-dimensional series for stochastic unsolved sub-grid processes as latent variables denoted by $\mathbf{V}_{1:T}$ which contains q univariate series for each sub-grid process denoted by $\mathbf{v}^1_{1:T}, \ldots, \mathbf{v}^q_{1:T}$. We can write the Bayesian inverse problem in a Bayesian hierarchical structure as,

$$\begin{aligned}
&\text{Layer I: } \mathbf{y}|\mathbf{V}_{1:T}, \theta, z_0 \sim N\left(\cdot|\mathcal{F}(\mathbf{V}_{1:T}, \theta, z_0)\,, \Gamma_m\right) \\
&\text{Layer II: } \mathbf{V}_{1:T}|\theta, z_0 \sim p_\psi(\cdot|\theta, z_0) \\
&\text{Layer III: } \theta \sim \pi_\theta(\cdot), z_0 \sim \pi_{z_0}(\cdot)
\end{aligned} \tag{3.4}$$

with which we are able to derive the full (surrogate) posterior distribution for θ,

$$\pi(\theta, z_0, \mathbf{V}_{1:T}|\mathbf{y}) \propto N(\mathbf{y}|\mathcal{F}\left(\mathbf{V}_{1:T}, \theta, z_0\right), \Gamma_m)\, p(\mathbf{V}_{1:T}|\theta, z_0)\pi(\theta)\pi(z_0) \tag{3.5}$$

[1] The density function in Layer I is specified by the statistical model for measurement noise and a deterministic function $\mathcal{F}(\mathbf{V}_{1:T}, \theta, z_0)$ which maps latent variables $\mathbf{V}_{1:T}$ to forward evaluations $\mathcal{G}^s(\theta; z_0)$. Construction of $\mathcal{F}(\cdot)$ usually depends on deterministic parameterisations of stochastic forecast models and does not include extra randomness. In Layer II, $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$ is the joint conditional density function of latent $\mathbf{V}_{1:T}$ which can be specified by stochastic parameterisations for sub-grid processes. Noticing the unsolved sub-grid processes $\mathbf{V}_{1:T}$ are normally time series with serial correlations, we apply autoregressive parameterisations to specify their joint density. In that sense, $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$ is treated as products of a series of autoregressive model parameterised by ψ for sub-grid processes with covariates θ and z_0. Layer III includes prior distributions for θ and initial conditions z_0.

With the Bayesian hierarchical structure, we notice the only unknown part of the surrogate forward model and posterior distribution is the probabilistic model with density function $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$. Hence learning the surrogate forward model at the emulation step of CES boils down to learning the autoregressive model $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$.

[1] When z_0 is deterministic, Equation 3.4 and Equation 3.5 can be easily modified to the case with known initial conditions.

Recall the hyper-parameter training framework 2.3.2 for stochastic parameterisations mentioned earlier, training data is numerical trajectories obtained by solving the natural model. It coincides with the calibration step of CES with natural forward models where corresponding natural models are solved with different theta. Hence the hyperparameter training framework can be perfectly embeded CES without extra computational effort.

Taken together, our enhanced CES framework also compromises *Calibration, Emulation and Sampling* three steps. The first calibration step remains the same as the original CES using natural forward models and real data. Calibrated approximate samples concentrate near the main support of the posteior in Equation 3.3. The emulation step is enhanced with stochastic parameterisations. We only emulate subprogresses with autoregressive models and using numerical trajectories obtained in the first step to train the models. The sampling step uses the surrogate forward models with the learned autoregressive parameterisations in sampling methods such as MCMC. It gives samples converging to the full posterior distribution in Equation 3.5. The marginal posterior distribution of θ can be approximated from samples of the full posterior distribution. Compared to the original CES, our modifications make it possible to keep part of structural information of natural models and only emulate dynamics of sub-grid processes. Moreover, GP emulators are replaced by deep generative models which are capable of representing a broad class of distributions and suitable for time series data. It relaxes the distributional requirements for the emulator so that it more accurately characterises physical systems. We tackle the challenge to study more complex posterior distributions with modified computational sampling tools at the last step. For completeness, we give a review of the ensemble Kalman inversion and ensemble Kalman sampling methods used in [13] for calibration, and further introduce the two modified steps *Emulation* and *Sampling* in the following sections.

3.2 Calibration - EKS

We firstly use ensemble Kalman sampling (EKS) to calibrate some θ with the noisy data from the approximate posterior distribution. As discussed earlier, EKS is a derivative-free approximate sampler for Bayesian inversion. It proceeds ensemble Kalman inversion which serves a derivative-free optimizer for deterministic inverse problems and works as an approximate posterior sampler by incorporating prior information. EKI is proposed in [24] and EKS is proposed in [16].

EKI is obtained by time-discretization of the following interacting particle system,

$$\frac{d\theta^{(j)}}{dt} = -\frac{1}{J}\sum_{k=1}^{J}\left\langle \mathcal{G}\left(\theta^{(k)}\right) - \overline{\mathcal{G}}, \mathcal{G}\left(\theta^{(j)}\right) - \mathbf{y}\right\rangle_{\Gamma_{\mathbf{y}}}\left(\theta^{(k)} - \bar{\theta}\right) \tag{3.6}$$

where $\langle\cdot,\cdot\rangle_{\Gamma_y}$ denotes the inner product and $\bar{\theta}$ and $\bar{\mathcal{G}}$ denote the sample means given by,

$$\bar{\theta} = \frac{1}{J}\sum_{k=1}^{J}\theta^{(k)}, \quad \overline{\mathcal{G}} = \frac{1}{J}\sum_{k=1}^{J}\mathcal{G}\left(\theta^{(k)}\right)$$

The above dynamical system drives the particles to fit the data and thus solve the deterministic inverse problem.

For EKS, we add a prior related damping term and some θ-related noise to Equation 3.6,

$$\frac{d\theta^{(j)}}{dt} = -\frac{1}{J}\sum_{k=1}^{J}\left\langle \mathcal{G}\left(\theta^{(k)}\right) - \overline{\mathcal{G}}, \mathcal{G}\left(\theta^{(j)}\right) - \mathbf{y}\right\rangle_{\Gamma_{\mathbf{y}}}\left(\theta^{(k)} - \bar{\theta}\right) - \mathrm{C}(\Theta)\Gamma_{\theta}^{-1}\theta^{(j)} + \sqrt{2\mathrm{C}(\Theta)}\frac{dW^{(j)}}{dt} \tag{3.7}$$

where $\{W^{(j)}\}$ are a collection of i.i.d standard Brownian motions in the parameter space and $\mathrm{C}(\Theta)$ is given as the outer product below,

$$\mathrm{C}(\Theta) = \frac{1}{J}\sum_{k=1}^{J}\left(\theta^{(k)} - \bar{\theta}\right) \otimes \left(\theta^{(k)} - \bar{\theta}\right)$$

The system defined by Equation 3.7 approximates a mean-field Langevin-McKean diffusion process which is invariant to the posterior distribution defined by Equation A.4 (see [16] for theoretical details).

The algorithm for EKS implemented in this study time-discretises the interacting particle system in Equation 3.7 by means of a linearly implicit split-step scheme, which is given as,

Algorithm 1 Ensemble Kalman Sampling

1: Set the maximum number of iterations N and the ensemble size J
2: Initialize $\theta_0^{(j)}$ for $j = 1 \dots J$ with the prior $\pi(\theta)$
3: **for** $n = 1, \dots, N$ **do**
4: Calculate:

$$\mathrm{C}(\Theta) = \frac{1}{J} \sum_{k=1}^{J} \left(\theta^{(k)} - \bar{\theta}\right) \otimes \left(\theta^{(k)} - \bar{\theta}\right)$$

5: For $j \in \{1, \dots, J\}$, set

$$\theta_{n+1}^{(*,j)} = \theta_n^{(j)} - \Delta t_n \frac{1}{J} \sum_{k=1}^{J} \left\langle \mathcal{G}\left(\theta_n^{(k)}\right) - \overline{\mathcal{G}}, \mathcal{G}\left(\theta_n^{(j)}\right) - \mathbf{y} \right\rangle_{\Gamma_\mathbf{y}} \theta_n^{(k)} - \Delta t_n \mathrm{C}\left(\Theta_n\right) \Gamma_\theta^{-1} \theta_{n+1}^{(*,j)}$$

$$\theta_{n+1}^{(j)} = \theta_{n+1}^{(*,j)} + \sqrt{2\Delta t_n \mathrm{C}\left(\Theta_n\right)} W_n^{(j)}, \text{ with } W_n^{(j)} \sim N(0, I_p)$$

▷ Δt_n is an adaptive timestep suggested by [24]

6: **end for**

3.3 Sub-grid progress emulation - Probabilistic Autoregressive models

The calibration step should be able to produce an approximate posterior sample of θ which is denoted as $\{\theta^{(m)}\}_{m=1}^{M}$ with sample size M. Calibration methods such as ensemble Kalman sampling also evaluate natural forward models on approximate samples of θ and we denote these natural forward evaluations by $\{\mathcal{G}^N(\theta^{(m)})\}_{m=1}^{M}$. Evaluating natural forward models requires solving natural models and thus producing numerical trajectories for all revolved progresses by the deterministic parameterisations. We let $\mathbf{X}_{1:T}$ denote the time series for major large-scale processes which are still resolved by deterministic parameterisations of stochastic forecast models, apposed to the series for sub-grid processes $\mathbf{V}_{1:T}$. Hence calibration produces trajectories major processes and sub-grid processes evaluated on different θ, respectively denoted by $\{\mathbf{X}_{1:T}^{(m)}\}_{m=1}^{M}$ and $\{\mathbf{V}_{1:T}^{(m)}\}_{m=1}^{M}$. For simplicity, we use subscript t to index $\mathbf{V}_t^{(m)} \in \mathbb{R}^q$ at time t in the trajectories. As we disregard correlations across different series, every individual sub-grid process series $\mathbf{v}_{1:T}^{i,(m)}$ can be modelled separately with the same method and we then drop the index i and (m) without loss of generality.

Motivated by deterministic parameterisations that large-scale processes can help to explain small-scale processes, a series of covariates for sub-grid processes $\mathbf{v}_{1:T}$ can be constructed from $\mathbf{x}_{1:T}$, denoted by $\mathbf{x}_{v,1:T}$. We make an assertion, the covariate $\mathbf{x}_{v,t}$ is integrated with deterministic parameterisations and hence can be written as a

deterministic function of $\mathbf{v}_{0:t-1}$ and $\mathbf{x}_{v,0}$ as

$$\mathbf{x}_{v,t} = \mathcal{H}(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0}) \tag{3.8}$$

so they does not introduce extra randomness to the joint density of $\mathbf{v}_{1:T}$ and can be treated as deterministic covariates. We will illustrate the assertion with the Lorenz-96 system later.

The initial conditions $\mathbf{x}_{v,0}$ and $\mathbf{v}_0$ are included in z_0.

3.3.1 A general form of probabilistic autoregressive models for sub-grid processes

Our goal is to learn autoregressive models for sub-grid processes that can be used to specify the joint density $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$ in Layer II of Equation 3.4 and in the posterior distribution in Equation 3.5. Without loss of generality, a single series $\mathbf{v}_{1:T} \in \mathbf{V}_{1:T}$ is considered in this section. More specifically, the joint density for an univariate series $p_\psi(\mathbf{v}_{1:T}|\theta, z_0)$ can be decomposed into a series of conditionals with the serial-correlated assumption,

$$\begin{aligned} p_\psi(\mathbf{v}_{1:T}|\theta, z_0) = p_\psi(\mathbf{v}_T|\mathbf{v}_{0:T-1}, \mathbf{x}_{v,0:T}, \theta) \times \cdots \times p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) \\ \times \cdots \times p_\psi(\mathbf{v}_1|\mathbf{v}_0, \mathbf{x}_{v,0:1}, \theta) = \prod_{t=1}^{T} p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) \end{aligned} \tag{3.9}$$

where we assume only $\mathbf{x}_{v,0}$ and $\mathbf{v}_0$ in z_0 are related to the sub-grid process $\mathbf{v}_{1:T}$, and $\mathbf{x}_{v,0:t}$ is the series of covriates $\mathbf{x}_v$ available until time t. The autoregressive model for conditional distribution $p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ corresponds to the stochastic parameterisation for sub-grid processes. It is autoregressive in the sense that the distribution of $\mathbf{v}_t$ depends historical observations $\mathbf{v}_{0:t-1}$. We choose to let the conditional p.d.f $p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ belong to a parametric distributional class so that the conditional p.d.f is usually explicitly defined. Suppose the distributional class is $\mathcal{P} = \{P_\omega : \omega \in \Omega\}$ where ω are the vector of parameters of the class, the conditional p.d.f can be rewritten as,

$$p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) = p(\mathbf{v}_t; \omega_t), \text{with } \omega_t = \omega_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) \tag{3.10}$$

where $p(\mathbf{v}_t; \omega_t)$ is the p.d.f of the distributional class $\mathcal{P}$ with parameter ω_t and $\omega_\psi(\cdot)$ is deterministic mapping parameterised by unknown ψ. Equation 3.10 is a general form of probabilistic autoregressive models for sub-grid processes $\mathbf{v}_{1:T}$ in our study. We call them probabilistic autoregressive models because they are defined both by

probabilistic models of some distributional class $\mathcal{P}$ and autogressive mappings $\omega_t = \omega_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$. Any autoregressive models with the general form of Equation 3.10 can always provide explicitly conditional p.d.f $p_\psi(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$, which is essential to posterior inference for θ with MCMC. We will explain it further in the next section.

3.3.2 Probabilistic autoregressive model examples

3.3.2.1 Normal-distributed one-step Markov models

Noticing our data is usually real-valued and continuous, the class of normal distributions can be selected to model the conditional distribution. With the normal assumption, ω_t in Equation 3.10 is a two-dimensional vector with mean μ_t and standard deviation σ_t. The general form of conditional p.d.f in Equation 3.10 can specified as,

$$\begin{aligned} p(\mathbf{v}_t; \mu_t, \sigma_t) &= (2\pi\sigma_t^2)^{-\frac{1}{2}} \exp\left(-(\mathbf{v}_t - \mu_t)^2/(2\sigma_t^2)\right) \\ \text{with } \mu_t = \mu_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta), \sigma_t &= \sigma_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) \end{aligned} \tag{3.11}$$

where $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ are two deterministic functions parameterised by ψ, giving model parameters μ_t and σ_t for the normal distribution.

Specifications of $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ are also given by different model assumptions. One of the most common class of autoregressive models are the one-step Markov models which assume the future state of the system depends only upon the present state but not the past states. The assumption does feature selection for estimating $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ by removing variables which are more than one step earlier than the current step and factorise the joint distribution. Combining the one-step Markovian assumption with the normal assumption, the joint density can be factorised as,

$$\begin{aligned} p_\psi(\mathbf{v}_{1:T}|\theta, z_0) &= p_\psi(\mathbf{v}_T|\mathbf{v}_{T-1}, \mathbf{x}_{v,T}, \mathbf{x}_{v,T-1}, \theta) \times \cdots \times p_\psi(\mathbf{v}_1|\mathbf{v}_0, \mathbf{x}_{v,1}, \mathbf{x}_{v,0}, \theta) \\ &= \prod_{t=1}^{T} p_\psi(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta) \end{aligned} \tag{3.12}$$

where $p_\psi(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta)$ is assumed to be normal distributed with μ_t and σ_t as

$$\begin{aligned} p(\mathbf{v}_t; \mu_t, \sigma_t) &= (2\pi\sigma_t^2)^{-\frac{1}{2}} \exp\left(-(\mathbf{v}_t - \mu_t)^2/(2\sigma_t^2)\right) \\ \text{with } \mu_t = \mu_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta), &\text{and } \sigma_t = \sigma_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta) \end{aligned} \tag{3.13}$$

which generalises the conditional density given by the PolyAR1 parameterisation for L96 in Equation 2.21 and Equation 2.22 with an extra covariate θ and consistent with the modification in Equation 2.26 by replacing $U^*_{t'_i,k}$ with $\mathbf{v}_t$ and $X^*_{t'_i,k}$ with $\mathbf{x}_{v,t}$. The specifications for $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ given by PolyAR1 are,

$$\begin{aligned} \mu_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta) &= \gamma\mathbf{v}_{t-1} + U^d(\mathbf{x}_{v,t}, \theta) - \gamma(U^d\mathbf{x}_{v,t-1}, \theta) \\ \sigma_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta) &= \sigma_\epsilon^2(1 - \gamma^2) \end{aligned} \tag{3.14}$$

where $U^d(\mathbf{x}_{v,t}, \theta) = b_0 + b_1\mathbf{x}_{v,t} + b_2\mathbf{x}_{v,t}^2 + b_3\mathbf{x}_{v,t}^3 + \alpha^T\theta$ and $\psi = (\alpha, b_0, b_1, b_2, b_3, \sigma_\epsilon, \gamma)^T$. Specifically, we notice σ_t is same for all t with PolyAR(1).

3.3.2.2 Extensions with deep learning models

Although PolyAR(1) works well for parameterising sub-grid tendencies in the Lorenz-96 system, numerical weather and climate systems are much more complicated where more flexible autoregressive models are desired. Compared to traditional statistical tools such as polynomial regression and AR(1), deep learning techniques require less assumptions about statistical models and potentially represent a broader class of distributions. Here we show three deep learning based models for probabilistic autoregressive modelling to our end. They all can be embedded in the general framework of Equation 3.10 and thus can used to define the joint distribution $p_\psi(\mathbf{v}_{1:T}|\theta, z_0)$ for latent series $\mathbf{v}_{1:T}$. Brief comparisons of autoregressive models will be discussed with PolyAR(1) are shown in Table 3.1, where we show four main assumptions of PolyAR(1) are lifted up by different deep autoregressive models. Detailed descriptions of each are followed. [2]

Table 3.1

Assumptions	(1) $\mu(\cdot)$ is polynomial	(2) $\sigma(\cdot)$ is constant	(3) one-state Markovian	(4) normal distribution
PolyAR	✓	✓	✓	✓
Probablistic ANN	✗	✗	✓	✓
DeepAR	✗	✗	✗	✓
RNN+Normalising Flow	✗	✗	✗	✗

(1) Probabilistic ANN

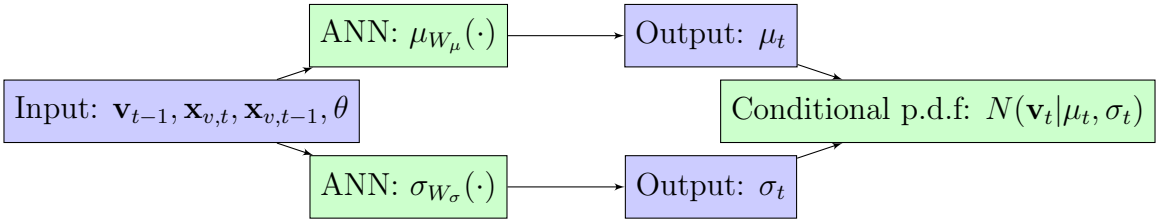
The first idea is to use feedforwad artificial neural networks (ANN) to find specifications for $\mu_t = \mu_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta)$ and $\sigma_t = \sigma_\psi(\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta)$ in the normal distribution model with one-step Markovian assumption in Equation 3.13. We call the idea Probabilistic ANN for simplicity.

A feedforward neural network ([25]) is a fully connected neural network composed of multiple layers with some non-linear activation functions. The architecture of a feedforward neural network includes two types of layers: a flatten input layer and some fully connected (or dense) layers. Fully connected layers are used for the hidden layers and the output layer. Normal non-linear activation functions for hidden layers

[2]Deep-learning based probabilistic autoregressive models will not be used in our numerical experiments with the Lorenz-96 system, because the L96 system is a simple toy model for the atmosphere. Sophisticated deep learning parameterisations are unnecessary for this toy model.

include $reLU$, $tanh$, and $sigmoid$ ([26]). The motivation of using ANN for $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ comes from the universal approximation theorem of artificial neural networks ([27] , [28]) which states neural networks can realise an arbitrary mapping from one vector space onto another vector space. As $\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta$ are different types of variables and may have different scales, we have little knowledge about how to construct feature vectors from them or how do $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ look like. Hence ANN can be a robust solution to recover unknown nonlinearity in $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ where the functions in Equation 3.14 given by PolyAR(1) can also be realised.

We treat $\mathbf{v}_{t-1}, \mathbf{x}_{v,t}, \mathbf{x}_{v,t-1}, \theta$ as input features and use two separate feedforward neural networks $\mu_{W_\mu}(\cdot)$ and $\sigma_{W_\sigma}(\cdot)$ with weights $W_\mu, W_\sigma \in \psi$. The normal p.d.f $p(\mathbf{v}_t; \mu_t, \sigma_t)$ in Equation 3.13 is denoted by $N(\mathbf{v}_t; \mu_t, \sigma_t)$ for simplicity. The outputs are μ_t and σ_t which are used to compute the conditional density $N(\mathbf{v}_t|\mu_t, \sigma_t)$. A schematic of Probabilistic ANN is given as below,



(2) DeepAR

One-state Markovian models are restrictive that the prediction is only made based its one-order lag and might lose some nontrivial historical information. Recurrent Neural Network ([29]), RNN in short, is class of neural networks that provides the flexibility to look into any longer history of the time series when making predictions.

DeepAR is a new deep learning framework proposed by [30], which applies RNN to probabilistic forecasting in time series. Without loss of generality, we still assume the conditional distribution follows normal distribution as $p(\mathbf{v}_t; \omega_t) = N(\mathbf{v}_t; \mu_t, \sigma_t)$ as Equation 3.11. Utilising DeepAR, $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$ for μ_t and σ_t are rewritten as,

$$\mu_t = \mu_\psi(\mathbf{h}_t, \theta), \quad \sigma_t = \sigma_\psi(\mathbf{h}_t, \theta) \tag{3.15}$$

where $\mathbf{h}_t$ is the output of an RNN such that,

$$\mathbf{h}_t = h_\psi(\mathbf{h}_{t-1}, \mathbf{v}_{t-1}, \mathbf{x}_{v,t}) \tag{3.16}$$

where $h_\psi(\cdot)$ is a function implemented by the autoregressive RNN with LTSM cells and $\mathbf{h}_t \in \mathbb{R}^h$ for $t = 1, \ldots, T$ is a vector of h latent variables predicted with information

up to t-1. The dimension of the latent vector h is a tuning variable which needs determining in the training stage. In the sequence-to-sequence settings as [31], $\mathbf{h}_t$ is the output of an encoder network, and μ and σ in Equation 3.15 are outputs of decoder networks. With Equation 3.15 and Equation 3.9, the joint density can rewritten as,

$$p_\psi(\mathbf{v}_{1:T}|\theta, z_0) = \prod_{t=1}^{T} p_\psi(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_{v,t}, \theta) \tag{3.17}$$

which has some extra latent variables $\mathbf{h}_{t-1}$ when compared to the one-step Markovian model Equation 3.12. The latent variables provide additional information for time series prediction than the one-step Markovian model, allowing the model incorporating all past information, while the joint density is still factorisable with respect to $\mathbf{v}_{t-1}$, $\mathbf{h}_{t-1}$ and $\mathbf{x}_{v,t}$, making the model training computationally tractable.

(3) A further extension with normalising flow

Nevertheless, there are two main drawbacks for the DeepAR method. Firstly, each individual time series needs to be modelled separately because using multivariate normal distribution needs to model a full covariance matrix. In numerical weather models, we usually have multiple sub-grid processes to be modeled. Assume we have q sub-grid processes, it is prohibitively expensive to estimate $q \times q$ parameters for the covariance matrix through neural networks. Secondly, the univariate normal distribution assumption is restrictive in many scenarios despite its simple expression. An advanced method, proposed by [32], deals with the two issues of DeepAR by combining the auto-regressive RNN with normalizing flows ([33]; [34]), and still explicitly defines conditional density functions for multivariate time series without assuming any distributional classes.

Background: Normalising flows with RealNVP. Normalizing flows are invertible mappings from $\mathbb{R}^D$ to $\mathbb{R}^D$ such that some complex densities $p_\mathcal{X}(\cdot)$ on the input space $\mathcal{X} = \mathbb{R}^D$ will be transformed to some simple distributions $p_\mathcal{Z}(\cdot)$ on the output space $\mathcal{Z} = \mathbb{R}^D$ (e.g. Gaussian distribution). The normalizing flow mappings $f : \mathcal{X} \to \mathcal{Z}$ are normally compositions of a sequence of bijections and simple invertible functions. With the formula of change of variables, we can write,

$$p_\mathcal{X}(x) = p_\mathcal{Z}(z) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \tag{3.18}$$

and

$$\log\left(p_\mathcal{X}(x)\right) = \log\left(p_\mathcal{Z}(f(x))\right) + \log\left(\left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \right) \tag{3.19}$$

where $\partial f(x)/\partial x^T$ is the Jacobian of f at x. Normalising flow models should have specific properties for their bijections such that $f^{-1}(\cdot)$ is easy to evaluate and the calculation of the Jacobian $\partial f(x)/\partial x^T$ is tractable.

A bijection satisfied the requirements and popularly used in density estimation with normalising flows is RealNVP ([35]). The bijective function is called the coupling layer in the learning framework and it is given as,

$$\begin{cases} y^{1:d} = x^{1:d} \\ y^{d+1:D} = x^{d+1:D} \odot \exp\left(s\left(\mathbf{x}^{1:d}\right)\right) + t\left(\mathbf{x}^{1:d}\right) \end{cases} \tag{3.20}$$

With normalising flow, we are able to transfer the joint distribution $p(\mathbf{v}_{1:T}; \omega_t)$ in Probabilistic ANN and DeepAR into more complex multivariate distributions joint p.d.f of which are still explicitly defined.

.

3.3.3 Training framework

The probabilistic autoregressive model $p(\mathbf{v}_t; \omega_t)$ with $\omega_t = \omega_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ defined in Equation 3.10 also corresponds to the stochastic parameterisation for sub-grid progresses, as we argue earlier. Hence the training framework of stochastic parameterisations for L96 in Section 2.3.2 can also used to be train the probabilistic autoregressive model with parameters ψ here. The first step of the framework in in Section 2.3.2 is to solve natural models and obtain resulted trajectories as ground truth. This step has handled by the calibration step of the CES framework where we also solve natural models with θ from an approximate posterior sample from the calibration step and obtain M pairs $\{\theta^{(m)}, \mathbf{V}_{\mathcal{T}}^{(m)}, \mathbf{X}_{v,\mathcal{T}}^{(m)}\}$ for $m = 1, \ldots, M$ where $\mathcal{T}$ is an index set for numerically solving natural forward models. $\mathbf{V}_{\mathcal{T}}^{(m)}$ and $\mathbf{X}_{v,\mathcal{T}}^{(m)}$ all contain q individual series denoted by $\mathbf{v}_{\mathcal{T}}^{1,(m)}, \ldots \mathbf{v}_{\mathcal{T}}^{q,(m)}$ and $\mathbf{x}_{v,\mathcal{T}}^{1,(m)}, \ldots \mathbf{x}_{v,\mathcal{T}}^{q,(m)}$ again. The M pairs as well as the q univariate series are regarded as ground truth for training the probabilistic autoregressive model $p(\mathbf{v}_t; \omega_t)$ with $\omega_t = \omega_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$.

The second step of the framework in Section 2.3.2 is to prepare training data for stochastic forecast models with the ground truth series. The deterministic parameterisations in stochastic forecast models are usually for larges-scale variables. For example, the slow variables X in the Loren-96 system of Equation 2.1 $\sim$ **??**. Thus the realisations of stochastic forecast models are always taken with larger time intervals compared the ground truth trajectories obtained by solving natural models. Hence we need to take a sub-sample with larger time intervals from the ground truth series $\{\mathbf{V}_{\mathcal{T}}^{(m)}\}$ and $\{\mathbf{V}_{v,\mathcal{T}}^{(m)}\}$ for $m = 1, \ldots, M$ and prepare it as the training data for $p(\mathbf{v}_t; \omega_t)$

with $\omega_t = \omega_\psi(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$. Suppose we take a sub-sample of size with every n_t points from the original time index set $\mathcal{T}$ as required by stochastic forecast models. We use $t \in \{1:T\}$ to re-index the new sub sample series, and the training data is then denoted by $\{\theta^{(m)}, \mathbf{V}^{(m)}_{1:T}, \mathbf{X}^{(m)}_{v,1:T}\}$ for $m = 1, \ldots, M$

The last step of the framework in Section 2.3.2 requires a loss function for training. We notice probabilistic autoregressive models with the general form of Equation 3.10 all have explicit p.d.f parameterised by ψ, with which we can construct a likelihood function using the training data with respect to ψ. Hence the negative log-likelihood function can be used as the loss function to train the corresponding probabilistic autoregressive model. A general form of the loss function specified by negative log-likelihood for probabilistic autoregressive models in Equation 3.10 is given as,

$$\mathcal{L}(\psi) = -\sum_{m=1}^{M}\sum_{i=1}^{q}\sum_{t=1}^{T} \log\left[p(\mathbf{v}_t^{i,(m)} \mid \omega(\mathbf{v}_{0:t-1}^{i,(m)}, \mathbf{x}_{v,0:t}^{i,(m)}, \theta; \psi))\right] \tag{3.21}$$

where $p(\cdot|\omega)$ is a p.d.f and $\omega(\cdot|\psi)$ is a deterministic function parameterised by ψ, both specified by the probabilistic autoregressive model. For example, for normal-distributed one-step Markov models as Equation 3.13, the loss function for training is specified as,

$$\begin{aligned}
\mathcal{L}(\psi) &= -\sum_{m=1}^{M}\sum_{i=1}^{q}\sum_{t=1}^{T} \log\left[N\left(\mathbf{v}_t^{i,(m)} \mid \mu(\mathbf{v}_{0:t-1}^{i,(m)}, \mathbf{x}_{v,0:t}^{i,(m)}, \theta; \psi), \sigma^2(\mathbf{v}_{0:t-1}^{i,(m)}, \mathbf{x}_{v,0:t}^{i,(m)}, \theta; \psi)\right)\right] \\
&\propto \frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{q}\sum_{t=1}^{T}\left[\log(\sigma_t^{i,(m)}(\psi)^2) + (\mathbf{v}_t^{i,(m)} - \mu_t^{i,(m)}(\psi))^2/\sigma_t^{i,(m)}(\psi)^2\right] \\
&\text{with } \mu_t^{i,(m)}(\psi) = \mu(\mathbf{v}_{0:t-1}^{i,(m)}, \mathbf{x}_{v,0:t}^{i,(m)}, \theta; \psi), \quad \sigma_t^{i,(m)}(\psi) = \sigma(\mathbf{v}_{0:t-1}^{i,(m)}, \mathbf{x}_{v,0:t}^{i,(m)}, \theta; \psi)
\end{aligned} \tag{3.22}$$

which is a general form of the loss function for PolyAR(1) for Lorenz-96 in Equation 2.23 only with an extra covariate θ.

Optimisation of the loss function in Equation 3.21 is computationally tractable even for complex deep learning models if we choose certain distributional classes for the probabilistic autoregressive model. For example, the likelihood function for normal distribution is concave in μ and σ. Thus the loss function of normal distribution in Equation 3.22 is convex in $\mu_t^{i,(m)}(\psi)$ and $\sigma_t^{i,(m)}(\psi)$ which will not add much computational burden in learning compared to learning of deterministic deep learning models $\mu_\psi(\cdot)$ and $\sigma_\psi(\cdot)$, and all popular learning algorithms can be applied. Broadly, if we choose exponential family distributions as $\mathcal{P}$ and their natural parameters as

ω in the probabilistic autoregressive model, we can always enjoy the fruit of convexity when training the probabilistic autoregressive model via likelihood based loss functions.

3.4 Posterior sampling with the surrogate model

At the sampling step, we use MCMC to study the posterior distribution for θ as [13]. Recall that we need to replace deterministic and expensive natural forward $\mathcal{G}^N(\theta; z_0)$ with cheap surrogate models $\mathcal{G}^S(\theta; z_0)$ with stochastic parameterisations for unsolved sub-grid processes $\mathbf{V}_{1:T}$, and then the Bayesian inverse problem becomes Bayesian hierarchical structure as Equation 3.4. Unlike using natural forward models, we now need to sample from the full posterior distribution defined in Equation 3.5 including θ as well as latent variables $\mathbf{V}_{1:T}, z_0$. A vital ingredient for MCMC is the joint likelihood for $\theta, \mathbf{V}_{1:T}, z_0$ given as,

$$p(\mathbf{y}, \theta, z_0, \mathbf{V}_{1:T}) = N(\mathbf{y}|\mathcal{F}(\mathbf{V}_{1:T}, \theta, z_0), \Gamma_m)\, p_\psi(\mathbf{V}_{1:T}|\theta, z_0)\pi(\theta)\pi(z_0) \tag{3.23}$$

which has to be explicitly defined and easy to evaluated. The deterministic mapping $\mathcal{F}(\cdot)$ is determined by deterministic parameterisations of stochastic forecast models and $\pi(\theta), \pi(z_0)$ are known priors. The conditional joint p.d.f $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$ is constructed with results from the emulation step and MCMC requires it to be explicitly defined.

The emulation step yields a learned probabilistic autoregressive models $p(\mathbf{v}_t; \omega_t)$ with $\omega_t = \omega_{\hat{\psi}}(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ of Equation 3.10 where $\hat{\psi}$ is the learned values for ψ. We fix the values of $\psi = \hat{\psi}$ and ignore it in this section, so $\omega_t = \omega(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ is a known deterministic mapping. Then the conditional p.d.f for $\mathbf{v}_{1:T}$ is defined as,

$$p(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) := p\left(\mathbf{v}_t; \omega(\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)\right) \tag{3.24}$$

Assuming $\hat{\psi}$ is true for all q series in $\mathbf{V}_{1:T}$ and disregard it in notations, we can construct the joint density of $p_\psi(\mathbf{V}_{1:T}|\theta, z_0)$ based on Equation 3.9 as,

$$p(\mathbf{V}_{1:T}|\theta, z_0) = \prod_{i=1}^{q}\prod_{t=1}^{T} p(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta) \tag{3.25}$$

We notice the joint density is only defined if the conditional p.d.f $p(\mathbf{v}_t|\mathbf{v}_{0:t-1}, \mathbf{x}_{v,0:t}, \theta)$ is defined, which again motivates our usage of probabilistic autoregressive models of the form in Equation 3.10.

With the surrogate forward model and the joint likelihood in Equation 3.23, many MCMC methods can be used to study the full posterior distribution of Equation 3.5. Here we employ one of the most popular MCMC methods in practice, the Random Walk Metropolis–Hastings algorithm where the random walk transition kernel is given as,

$$Q(x_{t+1}|x_t) := N(x_{t+1} \mid x_t, 1) \tag{3.26}$$

where $N(\cdot \mid x_t, 1)$ is the p.d.f of normal distribution with mean x_t and standard deviation 1. The pseudo-code of using Metropolis–Hastings algorithm to sample from Equation 3.5 is given as,

1. Make a proposal: $\theta', z_0', \mathbf{V}_{1:T'}$ based on the transition kernel such that

$$\theta', z_0', \mathbf{V}_{1:T} \sim Q(\cdot|\theta^t, z_0^t, \mathbf{V}_{1:T}^t)$$

2. Accept $\theta^{t+1}, z_0^{t+1}, \mathbf{V}_{1:T}^{t+1} = \theta', z_0', \mathbf{V}_{1:T}'$ with probability,

$$\min(1, \frac{N(\mathbf{y}|\mathcal{F}\left(\mathbf{V}_{1:T}', \theta', z_0'\right), \Gamma_m)\, p_\psi(\mathbf{V}_{1:T}'|\theta', z_0')\pi(\theta')\pi(z_0')Q(\theta^t, z_0^t, \mathbf{V}_{1:T}^t|\theta', z_0', \mathbf{V}_{1:T}')}{N(\mathbf{y}|\mathcal{F}\left(\mathbf{V}_{1:T}^t, \theta^t, z_0^t\right), \Gamma_m)\, p(\mathbf{V}_{1:T}^t|\theta^t, z_0^t)\pi(\theta^t)\pi(z_0^t)Q(\theta', z_0', \mathbf{V}_{1:T}'|\theta^t, z_0^t, \mathbf{V}_{1:T}^t)}) \tag{3.27}$$

3. Set $\theta^t, z_0^t, \mathbf{V}_{1:T}^t = \theta^{t+1}, z_0^{t+1}, \mathbf{V}_{1:T}^{t+1}$ and go back to Step 1

However, we have high-dimensions of latent variables $\mathbf{V}_{1:T}$ and the performance of Random Walk Metropolis–Hastings MCMC is not good in high-dimensional sampling. It might be necessary to search more appropriate high-dimensional sampling techniques for this end in the future research. It is out of scope of this study to further discuss advanced sampling methods.

Chapter 4

Numerical experiments with the Lorenz-96 system

In the section, we present applications of the proposed enhanced CES framework to Bayesian inverse problems in the Lorenz-96 system introduced in Section 2 with numerical experiments. A glossary of notations used in the general enhanced CES framework and their specifications in the Lorenz-96 system (L96) is given in Table 4.1. Discrete notations are applied in this section. We construct a matrix $\mathbf{U}_T^* \in \mathbb{R}^{N_T' \times K}$

Table 4.1: Glossary

Name	Notation in CES	Lorenz-96
Natural model	not used	Equation 2.1
Stochastic forecast model	not used	Equation 2.14
Natural forward model	$\mathcal{G}^N(\theta; z_0)$	$\mathcal{G}_T(\theta; z_0)$ in Equation 2.8
Stochastic forward model	$\mathcal{G}^S(\theta; z_0)$	$\mathcal{G}_T^*(\theta; z_0)$ in Equation 2.25
Major large-scale processes in $\mathcal{G}^S(\theta; z_0)$	$\mathbf{X}_t \in \mathbf{X}_{1:T}$	Slow variables $\{X_{t_i',k}^*\}_{k=1}^K$
Unsolved sub-grid processes or latent variables	$\mathbf{V}_t \in \mathbf{V}_{1:T}$	$\{U_{t_i',k}^*\}_{k=1}^K, \{V_{t_i',k}^*\}_{k=1}^K$

with $[\mathbf{U}_{N_T'}^*]_{i,k} = U_{t_i',k}^*$. The same procedure applies to construct $\mathbf{V}_T^* \in \mathbb{R}^{N_T' \times K}$. The Bayesian hierarchical model of Equation 3.4 in for L96 is then given as,

$$\begin{aligned} &\text{Layer I: } \mathbf{y}|\mathbf{U}_T^*, \mathbf{V}_T^*, \theta, z_0 \sim N(\cdot|\mathcal{F}(\mathbf{U}_T^*, \mathbf{V}_T^*, \theta, z_0), \Gamma_m) \\ &\text{Layer II: } \mathbf{U}_T^*, \mathbf{V}_T^*|\theta, z_0 \sim p(\cdot|\theta, z_0) \\ &\text{Layer III: } \theta \sim \pi(\theta), z_0 \sim \pi(z_0) \end{aligned} \tag{4.1}$$

with which we are able to derive the (surrogate) full posterior distribution,

$$\pi(\theta, z_0, \mathbf{U}_T^*, \mathbf{V}_T^*|\mathbf{y}) \propto N(\mathbf{y}|\mathcal{F}(\mathbf{U}_T^*, \mathbf{V}_T^*, \theta, z_0), \Gamma_y) p(\mathbf{U}_T^*, \mathbf{V}_T^*|\theta, z_0) \pi(\theta) \pi(z_0) \tag{4.2}$$

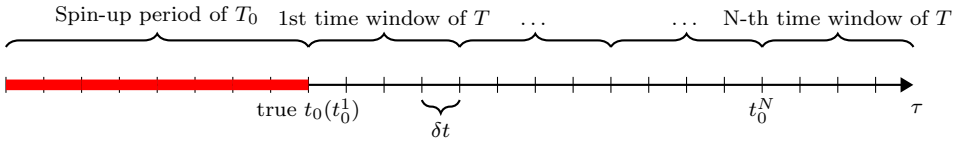
As the same method can be used to model all sub-grid processes in $\mathbf{U}_T^*$ as well as in $\mathbf{V}_T^*$. Without loss of generality, we only consider $U_{t_i',k} \in \mathbf{U}_T^*$ in this study.

4.1 Experiment design

In the numerical experiments, we study a Bayesian inverse problem in the Lorenz-96 system as Section 2.2 with unknown model parameters θ_0 and known initial states z_0. Time-location-averaged data suggested by [13] is used for inversion. We implement the EKS algorithm at the calibrations step and the enhanced surrogate forward emulator at the emulation step. Posterior sampling with MCMC is not included in our numerical experiments because of the untackled deficiency of using standard MCMC algorithms in this case.

4.1.1 Synthetic data generation process

Data for inversion (denoted by $\mathbf{y}$ in methodology) in this study is synthetic data generated by the natural forward model of L96 with Gaussian measurement noise. The data is constructed to be time-location-averaged across K slow variables (locations) and over time horizon of T, consistent with the setting of numerical experiments in [13]. To construct the data, we run the natural L96 system with the RK4 algorithm with maximum time step of δt and over a time period of τ. An initial spin-up period T_0 is taken to ensure the system reaches equilibrium when constructing the synthetic data. We then divide the time after dropping the spin-up period into N time windows of length T and calculate time-location-averaged within each window. A visualisation of the synthetic data generating process is given as,



where we use t_i to index discrete time points with interval δt as Section 2.2 and set t_0^n as the starting point of the n-th time window. Recall the observational vector of inverse problems in L96 in Equation 2.7 is a 5-dimensional vector of $\phi_{t_i,k} = \left(X_{t_i,k}, \overline{Y_{t_i,k}}, X_{t_i,k}^2, X_{t_i,k}\overline{Y_{t_i,k}}, \overline{Y_{t_i,k}^2}\right)^T$. As we only model $U_{t_i}^*$, the stochastic counterpart of $\overline{Y_{t_i,k}}$, in the surrogate forward model, we should also only keep $\overline{Y_{t_i,k}}$ in the observational vector and drop the other sub-gird processes $\overline{Y_{t_i,k}^2}$. Then within each time window n, the time-location-averaged is calculated as,

$$\mathbf{y}_n = \sum_{k=1}^{K}\sum_{i=0}^{N_T} \begin{pmatrix} X_{t_i^n,k} \\ \overline{Y_{t_i^n,k}} \\ X_{t_i^n,k}^2 \\ X_{t_i^n,k}\overline{Y_{t_i^n,k}} \end{pmatrix} + \eta_n \quad \text{for } n = 1, \ldots, N \tag{4.3}$$

where N_T is the number of discrete points available within T and $\eta_n \in \mathbb{R}^4$ is a vector of measurement noise i.i.d taken from $N(0, \Gamma_m)$ with covariance matrix $\mathbb{R}^{4\times4}$. For simplicity, we use $X, \overline{Y}, X^2, X\overline{Y}$ to denote the four averaged variables in the later section.

4.1.2 Parameter settings

Numerical values for aforementioned parameters and parameters in Equation 2.1 used to generate synthetic data are given in Table 4.2. Parameters for the L96 system are chosen consistent with the numerical experiments done by [13]. All times are in model time unit (MTU) which is approximately equal to 5 atmospheric days. The initial states for the fast variables are set as 0.1 and for slow variables are 1. We construct the measurement noise covariance matrix to be diagonal with entries $\sigma_p^2 = 0.01$ for $p = 1, \ldots, 4$ indexed over variable type (the four averaged variables), that is, $\Gamma_m = \text{diag}(\sigma_p^2)$.

Table 4.2

parameter	symbol	setting	parameter	symbol	setting
number of X variables	K	8	time-scale ratio	c	10
number of Y variables for each X variable	L	32	natural model total running time	τ	402
coupling constant	h	1	time window	T	10
forcing term	F	10	spin-up time	T_0	2
spatial-scale ratio	b	10	time step for natural model	δt	0.001

4.1.3 Inverse problem settings

The parameters used to generate synthetic data in Table 4.2 are treated as ground truth in the experiments. In inverse problems, we assume the ground truth values of h, F, b are unknown to researchers and thus need to be inferred via Bayesian inversion. For simplicity, we let $\theta = (h, F, b)^T$.[1]. The prior distributions for the physical parameters θ are set to be independently Gaussian-distributed. More explicitly, we assume the prior for the vector of parameters $\pi(\theta)$ follows a multivariate Gaussian distribution with the mean $m_\theta = (0, 10, 8)^T$ and diagonal covariance matrix $\Gamma_\theta = \text{diag}(1, 5, 5)$. The uncertainty of initial values is not considered in this study.

[1]We treat c as fixed in the numerical study for simplicity which is different from the notation before in chapter 2 and the numerical experiment in [13]

4.2 Important implementation methods

Many computational mathematics and statistics tools are required in numerical experiments of this study. Due to page limit, we will not one-by-one introduce them here but choose to only further explain some important ones. All implementations in Python are given in the appendix (also on Github).

4.2.1 Ensemble Kalman Sampling in L96

The algorithm for ensemble Kalman sampling is given in Algorithm 1 which can be applied to find an approximate sample for θ in our problem by using L96 natural forward model $\mathcal{G}_T(\theta; z_0)$ as the forward model. One thing worth mentioning is we use long-term empirical covariance matrix calculated over a long time horizon as $\Gamma_{\mathbf{y}}$ in Algorithm 1. In practice, data of long-term empirical covariance matrix is usually available in climatology. Hence in this study, we use the empirical covariance matrix for synthetic data over time τ with the ground truth model parameters as $\Gamma_{\mathbf{y}}$.

4.2.2 Surrogate forward model evaluation

It is worth mentioning again the time step for the stochastic forecast model and surrogate forward model is usually larger than the one for the natural model and natural forward model because we only need to integrate for the slow variable. The time step for the stochastic forecast is set as $\Delta t = 0.005$ and the corresponding time index is $t_{i'}$, consistent with Chapter 2. Recall that the surrogate forward for L96 is constructed with the stochastic forecast model in Equation 2.14 which combines the deterministic parameterisation for slow variables X and stochastic parameterisations for fast variables Y Hence solving it requires both ODE integration and stochastic forecasting, more technically challenging than solving the natural model with only deterministic parameterisations.

With probabilistic autoregressive models with Gaussian and one-step Markovian assumptions of the general form in Equation 3.13, the conditional distribution for sub-grid process $U^*_{t'_i,k}$ is given as,

$$\begin{aligned} U^*_{t'_i,k} \sim N\left(\mu_{t'_i}, \sigma^2_{t'_i}\right) \text{ with } \mu_{t'_i} &= \omega_\psi(U^*_{t'_i-1,k}, X^*_{t'_i,k}, X^*_{t'_i-1,k}, \theta), \\ \sigma_{t'_i} &= \sigma_\psi(U^*_{t'_i-1,k}, X^*_{t'_i,k}, X^*_{t'_i-1,k}, \theta) \end{aligned} \tag{4.4}$$

where ψ are learned in the emulation step and same for all $k = 1, \ldots, K$. Combining Equation 4.7 with the parameterisation in Equation 2.14, the stochastic forecast

model can be solved as Algorithm 2. We notice it provides a recurrent relation between the two series $\{X^*_{t'_i,k}\}$ and $\{U^*_{t'_i,k}\}$. If we represent Step 3 of Algorithm 2 as a deterministic recursive equation of $X^*_{t'_{i+1},k}, X^*_{t'_i,k}$ and $U^*_{t'_i,k}$ as,

$$\begin{aligned} X^*_{t'_{i+1},k} &= \mathcal{D}(X^*_{t'_i,k}, U^*_{t'_i,k}, \theta) \\ X^*_{t'_i,k} &= \mathcal{D}(X^*_{t'_{i-1},k}, U^*_{t'_{i-1},k}, \theta) \\ &\dots \\ X^*_{t'_1,k} &= \mathcal{D}(X_{t'_0,k}, U_{t'_0,k}, \theta), \end{aligned} \tag{4.5}$$

this give a deterministic equation of $\{U^*_{t'_i,k}\}$ and $X_{t'_0,k}, U_{t'_0,k}, \theta$ for $X^*_{t'_{i+1},k}$ as

$$X^*_{t'_{i+1},k} = \mathcal{H}\left(U^*_{t_{N'_T}}, \dots, U^*_{t'_1}, U_{t'_0,k}, X_{t'_0,k}, \theta\right) \tag{4.6}$$

with which we have shown the assertion made in Section 3.3 is true for L96.

Algorithm 2 Solve the stochastic forecast model of Lorenz-96

1: Initialise $\theta, X^*_{t'_0} = X_{t_0,k}$ and $U^*_{t'_0} = U_{t_0,k}$ with some prior distributions
2: **for** $i = 1, \dots, N$ **do**
3: Given $X^*_{t'_{i-1},k}$ and $U^*_{t'_{i-1},k}$, solve for $X^*_{t'_i,k}$ for $k = 1, \dots, K$ by integrating the stochastic forecast model with RK2 using time step Δt,

$$\frac{\mathrm{d}X^*_k(t)}{\mathrm{d}t} = -X^*_{k-1}(t)\left[X^*_{k-2}(t) - X^*_{k+1}(t)\right] - X^*_k(t) + F - U^*_k(t)$$

.
4: With $X^*_{t'_i,k}$, $X^*_{t'_{i-1},k}$, and $U^*_{t'_{i-1},k}$, calculate:

$$\begin{aligned} \mu_{t'_i,k} &= \omega_\psi(U^*_{t'_i-1,k}, X^*_{t'_i,k}, X^*_{t'_i-1,k}, \theta), \\ \sigma_{t'_i,k} &= \sigma_\psi(U^*_{t'_i-1,k}, X^*_{t'_i,k}, X^*_{t'_i-1,k}, \theta) \end{aligned} \tag{4.7}$$

5: Predict $U^*_{t'_i,k}$ with the probabilistic autoregressive model as,

$$\hat{U}^*_{t'_i,k} = \mu_{t'_i,k}$$

6: Set $i = i + 1$ and go back to step 3
7: **end for**

After obtaining numerical trajectories with Algorithm 2, the surrogate forward model can be constructed with the observational operator in Equation 2.24.

4.3 Numerical results

4.3.1 Synthetic data overview

The summary statistics for the non-noisy observations and noisy observations for $X, \overline{Y}, X^2, X\overline{Y}$ are respectively shown in Table 4.3 and Table 4.4 where we have observations for $N = 40$ time windows. We notice they all have close scales within $0 \sim 10$. The variances of non-noisy data reflect the sampling variability due to finite-time averaging with different initial values at t_0 for each time window, which are all relatively small for the synthetic data as shown in Table 4.3.

Table 4.3: Summary statistics for non-noisy synthetic observations for $X, \overline{Y}, X^2, X\overline{Y}$ in the Lorenz-96 system

Variable	X	$\overline{Y}$	X^2	$X\overline{Y}$
count	40	40	40	40
mean	2.620	3.597	6.862	9.422
std	0.004	0.040	0.020	0.097
min	2.613	3.500	6.829	9.188
max	2.637	3.677	6.951	9.609

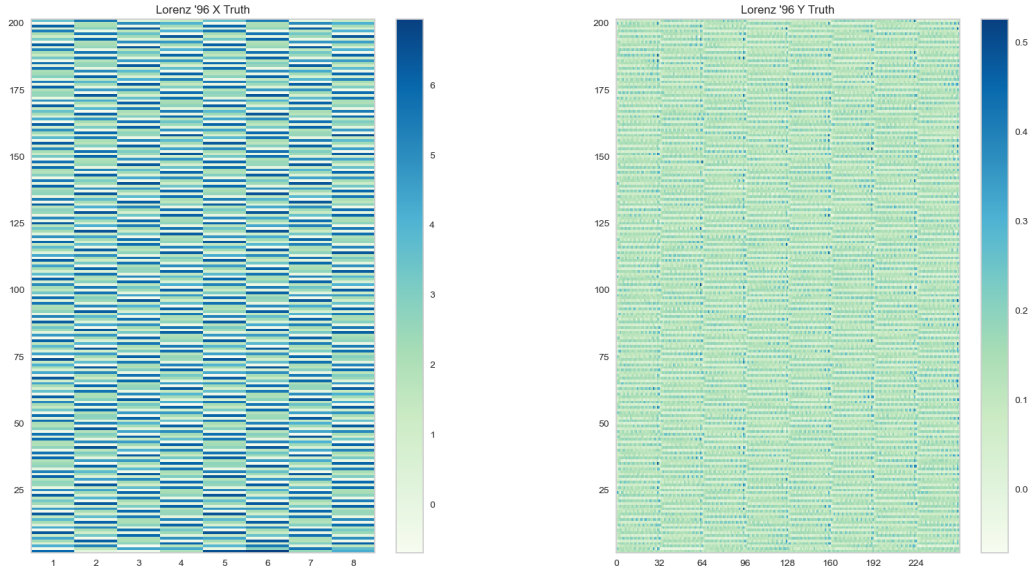
The standard deviations of the noisy synthetic data are all close to 0.01 which is close to σ_p^2 in the covariance matrix for generating the measurement noise. They will be used to construct $\Gamma_{\mathbf{y}}$ in the EKS algorithm. Bayesian inversion is performed on the noisy synthetic data.

Table 4.4: Summary statistics for noisy synthetic observations for $X, \overline{Y}, X^2, X\overline{Y}$ in the Lorenz-96 system

Variable	X	$\overline{Y}$	X^2	$X\overline{Y}$
count	40	40	40	40
mean	2.616	3.608	6.861	9.430
std	0.090	0.090	0.108	0.140
min	2.435	3.448	6.584	9.082
max	2.858	3.784	7.057	9.659

When solving the natural model to generate synthetic data with the ground truth parameters, we are also available to groundtruth trajectories the L96 system. Heatmaps of slow variables and fast variables over time of T = 200 are shown in Figure 4.1, where we notice fast variables have smaller scales than slow variables and change much faster.

Figure 4.1: Heatmaps of slow variable values (left) and fast variable values (right) in the Lorenz-96 system over time T = 200 (y-axis is for times in MTU, x-axis is for variables)



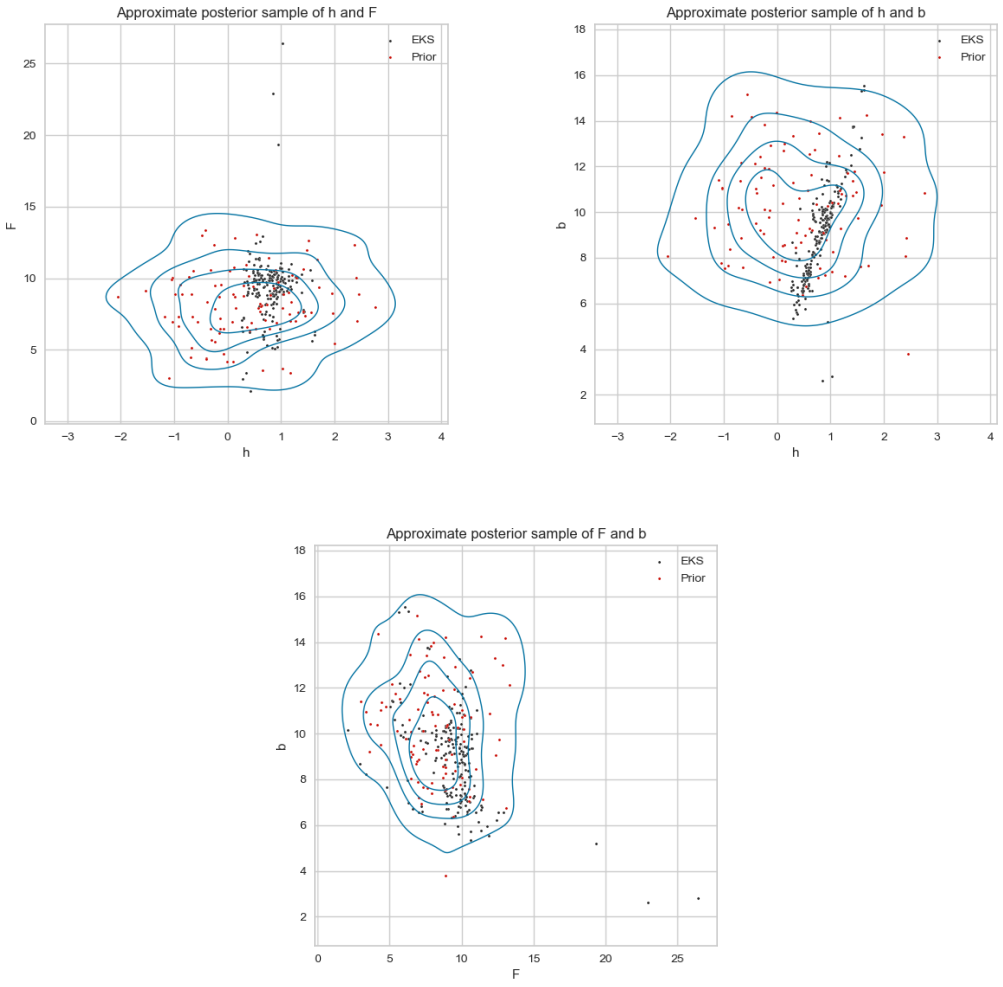
4.3.2 Calibration

First step of the enhanced CES framework is to find an approximate sample of $\theta = (h, F, b)^T$ to be inferred with the EKS algorithm. Although we have 40 observations, they have very small internal variance and the significant source of their variance is the measurement noise. Thus it might not give more information even if we use more data points and running the natural model over a long period within the EKS iterations is computationally expensive. Thus we only use N=2 data points to run the EKS algorithm. We run the EKS for 50 iterations with an ensemble size of J = 80.

The results shown in Figure 4.2 are bi-variate plots and kernel density estimates of samples given by the last iteration of EKS and drawn for the prior distributions.The lines are kenerl density estimates of prior distributions. Compared to the samples of prior distributions, we find the EKS samples for h and F are centered closer to the ground truth values 1 and 10 and also have small variance compared to the prior distributions. The EKS samples for h and F are indeed good quality approximate posterior samples. However, the EKS sample for b is less satisfactory which might be due to the settings of our EKS algorithm. The sample means of EKS are

$(0.792, 9.890, 8.761)$ for ground truth $h = 1, F = 10, b = 10$.

Figure 4.2: Calibrated samples for parameter of the Lorenz-96 system with the ensemble Kalman sample algorithm with J=80 ensembles after 50 iterations



4.3.3 Emulation

To evaluate performance of the new surrogate forward model, we take a shortcut from the calibration step noticing the ground truth values for θ are known to us. Instead of using the approximate posterior samples given by EKS, we directly generate a better sample of θ centered around the ground truth. A sample of $J = 80$ different θ is drawn from the Gaussian distribution with mean as the groundtruth θ and standard deviations 1, denoted by $\{\theta_j\}_{j=1}^{J}$. We run the natural model on the sample of θ and obtain J different theta-trajectory pairs. The training data for emulation is obtained

by taking a sub-sample of every 5 points of the trajectories. To avoid overfitting, we only use trajectories over an early time period of 20 MTU with 400 discrete points for training. We fit PolyAR(1) model on the training data.

To check fitting and emulation performance of the PolyAR(1) model, we run the surrogate forward model using Algorithm 2 with ground truth θ and initial values over 40 time windows of T = 10 MTU. The initial values are taken from t_0^1 of the first time window when generating the synthetic data. The running time of the surrogate forward model is about $30 \sim 35$ seconds in Python, significantly faster than the natural forward model which needs to solve for all faster variables.

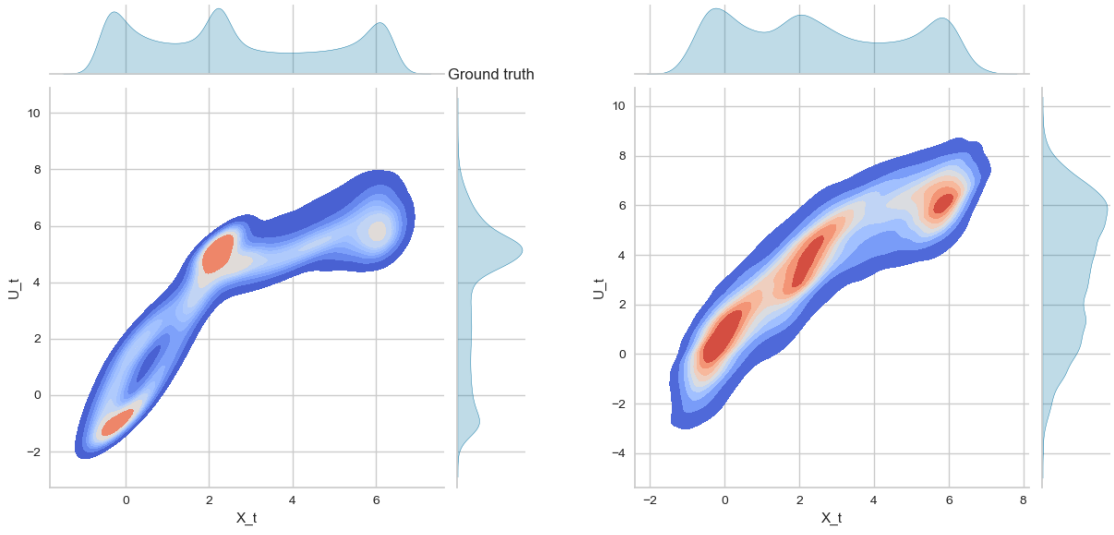
The summary statistics of forecasts given by the emulator are shown in Table 4.5. When compared to synthetic groundtruth data Table 4.3, predicted values of time-location averaged data using the surrogate forward model are close to the groundtruth data. However, the mean of predicted values for variables except for $X\overline{Y}$ are all slightly lower than synthetic ground truth data and the of the standard deviations for the four variables are larger among the forecasts.

Table 4.5: Summary statistics for forecasts of $X, \overline{Y}, X^2, X\overline{Y}, \overline{Y^2}$ with the PolyAR(1) surrogate forward model in the Lorenz-96 system

Variable	X	$\overline{Y}$	X^2	$X\overline{Y}$
count	40	40	40	40
mean	2.547	3.588	6.486	9.612
std	0.026	0.146	0.134	0.523
min	2.485	3.286	6.175	8.536
max	2.592	3.830	6.720	10.409

When looking at trajectories obtained by solving the surrogate forward model, we notice larger discrepancy between the forecasts and the groundtruth series. Figure 4.3 shows the kernel density estimates for the joint density of $X_{t_i',k}$ and $U_{t_i',k}$ over the whole model running period of $T = 40 \times 10$ MTU. Although we observe three color blocks in both panels at similar positions, contours in the two panels are discrepant. The result suggests more suitable and flexible probabilistic autoregressive models than PolyAR(1) should be applied to model the sub-grid process in the Lorenz-96 system. For example the deep learning based models mentioned in Section 3.3.2.

Figure 4.3: Kernel density estimates for the joint distribution of groundtruth (left panel) and forecasting (right panel) trajectories of $X_{t'_i}$ and $U_{t'_i}$ in the Lorenz-96 system over T = 400 MTU



Chapter 5

Conclusions and future work

In this study, we have proposed a generalised framework of emulator-based approximate Bayesian computation which can be applied to a wide range of numerical weather models with stochastic parameterisations based on the work of [13]. The new framework enhances classics complete data-driven emulators with the new surrogate model with stochastic parameterisations because it largely improves computational efficiency of forward evaluations by avoiding numerical computations for solving for high-dimensional sub-grid variables, meanwhile keeps partial structural information of the exact forward model. The application of such methodology has been shown on the famous toy model of the atmosphere, the Lorenz-96 system with derivations and numerical experiments. Moreover, a Bayesian hierarchical representation for inverse problems with the proposed surrogate model is given to make sure likelihood-based posterior inference is still tractable with the new surrogate model. We have shown the probabilistic model at the second layer of the Bayesian hierarchical structure corresponds to the stochastic parameterisation for sub-grid processes, and they share the same training framework.

The current work serves as the formulation and proof of concept for the new proposed methodology. There are many future directions stemming from this work,

- We propose a general class of probabilistic autoregressive models, however, the numerical experiments are only done with the classic PolyAR(1) method. Future research can be done in exploring performance of other probabilistic autoregressive models, especially these based on state-of-art deep learning models
- As discussed earlier, the classic MCMC algorithm is not appropriate for posterior sampling due to the high dimensions of latent variables, thus the research should be continued to search for better sampling methods

- Deploying the methodology to more realistic numerical weather and climate models such as the general circulation model.

Appendix A

A review of Bayesian inverse problems

Inverse problems understood as mapping observational data to model parameters are widely studied in many fields of science and engineering. They are regarded as the reversed process of solving the differential equations, the so-called forward problems. A generic mathematical representation for inverse problems is given as,

$$\mathbf{y} = \mathcal{G}(\theta) + \eta \tag{A.1}$$

where $\mathcal{G} : \mathbb{R}^p \to \mathbb{R}^d$ is a forward model which maps a p-dimensional parameter vector θ to a d-dimensional data vector $\mathbf{y}_0$ without noise. A d-dimensional random vector η is the random noise of the observational data, usually modelled by a Gaussian distribution.

In deterministic case, solving inverse problems is to find the best parameters that gives best data fitness, which boils down to optimisation problems. But in many practical scenarios such weather forecasting, we need not just point estimates for the unknown parameters but also uncertainty qualification for them. This is known as the probabilistic or Bayesian inverse problem where we are interested in the posterior distribution of θ given as,

$$\pi(\theta|y) = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{p(\mathbf{y})}, \quad \text{with } p(\mathbf{y}) = \int_{\mathbb{R}^p} p(\mathbf{y}|\theta)\pi(\theta)d\theta. \tag{A.2}$$

where $p(y|\theta)$ is the data likelihood and $\pi(\theta)$ is the prior distribution. A simple case is, we assume the observational noise follows a Gaussian distribution of $N(0, \Gamma_y)$ and the prior distribution for θ is also Gaussian with $N(0, \Gamma_\theta)$. The data then also follows a non-zero mean Gaussian distribution if the forward model is deterministic. The energy function for the system $\mathcal{G}(\theta)$ is given as,

$$\Phi_R(\theta|\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathcal{G}(\theta)\|^2_{\Gamma_y} + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}, \tag{A.3}$$

with $\|\cdot\|_A^2 = \|A^{-1/2}\cdot\|^2$ is the squared Mahalanobis distance, then the posterior distribution

$$\pi(\theta|y) \propto \exp(-\Phi_R(\theta)). \tag{A.4}$$

With the Gaussian setting, it is possible to get an explicit posterior by Gaussian conjugacy; otherwise, most posteriors are intractable due to the computational difficulty to calculate the marginal density $p(y)$. Massive computational statistics tools are accordingly proposed to sidestep this complication, including Monte Carlo Markov Chain (MCMC) sampling. The idea of MCMC is to construct a Markov Chain which has the target distribution as its equilibrium distribution and it is commonly applied in Bayesian inversion. For MCMC, a necessary ingredient is the data likelihood used to calculate the acceptance probability. The second quandary where even the likelihood is tractable or difficult to evaluate is called "doubly intractable" and tackled by approximation Bayesian computation (ABC) methods.

Bibliography

[1] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of computational physics*, 227(7):3431–3444, 2008.

[2] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[3] HM Arnold, IM Moroz, and TN Palmer. Stochastic parametrizations and model uncertainty in the lorenz'96 system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1991):20110479, 2013.

[4] TN Palmer. Stochastic weather and climate models. *Nature Reviews Physics*, 1(7):463–471, 2019.

[5] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.

[6] Christopher K Wikle and L Mark Berliner. A bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):1–16, 2007.

[7] Geir Evensen et al. *Data assimilation: the ensemble Kalman filter*, volume 2. Springer, 2009.

[8] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214, 2015.

[9] Daniel S Wilks. Effects of stochastic parametrizations in the lorenz'96 system. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(606):389–407, 2005.

[10] MJ Rodwell and TN Palmer. Using numerical weather prediction to assess climate models. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 133(622):129–146, 2007.

[11] Dennis V Lindley. Approximate bayesian methods. *Trabajos de estadística y de investigación operativa*, 31(1):223–245, 1980.

[12] Simon L Cotter, Masoumeh Dashti, and Andrew M Stuart. Approximation of bayesian inverse problems for pdes. *SIAM journal on numerical analysis*, 48(1):322–345, 2010.

[13] Emmet Cleary, Alfredo Garbuno-Inigo, Shiwei Lan, Tapio Schneider, and Andrew M Stuart. Calibrate, emulate, sample. *Journal of Computational Physics*, 424:109716, 2021.

[14] Edward Meeds and Max Welling. Gps-abc: Gaussian process surrogate approximate bayesian computation. *arXiv preprint arXiv:1401.2838*, 2014.

[15] Andrew Stuart and Aretha Teckentrup. Posterior consistency for gaussian process approximations of bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018.

[16] David John Gagne, Hannah M Christensen, Aneesh C Subramanian, and Adam H Monahan. Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz'96 model. *Journal of Advances in Modeling Earth Systems*, 12(3):e2019MS001896, 2020.

[17] Daan Crommelin and Eric Vanden-Eijnden. Subgrid-scale parameterization with conditional markov chains. *Journal of the Atmospheric Sciences*, 65(8):2661–2675, 2008.

[18] Elana J Fertig, John Harlim, and Brian R Hunt. A comparative study of 4d-var and a 4d ensemble kalman filter: Perfect model simulations with lorenz-96. *Tellus A: Dynamic Meteorology and Oceanography*, 59(1):96–100, 2007.

[19] Sam Hatfield, Aneesh Subramanian, Tim Palmer, and Peter Düben. Improving weather forecast skill through reduced-precision data assimilation. *Monthly Weather Review*, 146(1):49–62, 2018.

[20] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

[21] Tim Palmer and Renate Hagedorn. *Predictability of weather and climate*. Cambridge University Press, 2006.

[22] James A Hansen and Cecile Penland. Efficient approximate techniques for integrating stochastic differential equations. *Monthly weather review*, 134(10):3006–3014, 2006.

[23] James A Hansen and Cécile Penland. On stochastic parameter estimation using data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):88–98, 2007.

[24] Nikola B Kovachki and Andrew M Stuart. Ensemble kalman inversion: a derivative-free technique for machine learning tasks. *Inverse Problems*, 35(9):095005, 2019.

[25] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[26] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[27] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[28] Gerald M Maggiora, David W Elrod, and Robert G Trenary. Computational neural networks as model-free mapping devices. *Journal of chemical information and computer sciences*, 32(6):732–741, 1992.

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

[30] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[32] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *arXiv preprint arXiv:2002.06103*, 2020.

[33] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[34] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

[35] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.