Variational Bayesian Methods for Inferring Spatial Statistics and Nonlinear Dynamics

Antonio Khalil Moretti

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

ABSTRACT

Variational Bayesian Methods for Inferring Spatial Statistics and Nonlinear Dynamics

Antonio Khalil Moretti

This thesis discusses four novel statistical methods and approximate inference techniques for analyzing structured neural and molecular sequence data. The main contributions are new algorithms for approximate inference and learning in Bayesian latent variable models involving spatial statistics and nonlinear dynamics. First, we propose an amortized variational inference method to separate a set of overlapping signals into spatially localized source functions without knowledge of the original signals or the mixing process. In the second part of this dissertation, we discuss two approaches for uncovering nonlinear, smooth latent dynamics from sequential data. Both algorithms construct variational families on extensions of nonlinear state space models where the underlying systems are described by hidden stochastic differential equations. The first method proposes a structured approximate posterior describing spatially-dependent linear dynamics, as well as an algorithm that relies on the fixed-point iteration method to achieve convergence. The second method proposes a variational backward simulation technique from an unbiased estimate of the marginal likelihood defined through a subsampling process. In the final chapter, we develop connections between discrete and continuous variational sequential search for Bayesian phylogenetic inference. We propose a technique that uses sequential search to construct a variational objective defined on the composite space of non-clock phylogenetic trees. Each of these techniques are motivated by real problems within computational biology and applied to provide insights into the underlying structure of complex data.

Table of Contents

List of Figures

Acknowledgments

There are so many people to thank for the opportunity and ability to continue my education. My academic advisor Dr. Itsik Pe'er has provided constant guidance and support over the past several years. Itsik has been a true educator and role model both as a scientist and an academic. I owe a huge thanks to Dr. Ansaf Salleb-Aouissi, my first academic advisor who has continued to be an invaluable mentor and a friend. I would also like to thank professors Alex Andoni, David Knowles and David Blei for serving on my committee, for being generous with their time in providing helpful feedback and career advice.

I was lucky to encounter and to know a number of people who acted as mentors before my decision to return to school. I want to thank Iftekhar Hasan, Mark Brennan-Ing, Virpi Ranta, Jose Gonzales-Brenes and Katherine McKnight for encouraging me to pursue a PhD. I would also like to express my gratitude to my teachers Robert Mason, Charles Rice, Michelle Viard, Stanley Rosenberg, Sage Morillo and Jerome Taylor among others for their encouragement and support when I was younger. Jessica Rosa, Cindy Meekins, Twinkle Edwards, Elias Tesfaye, Raluca Joanta each deserve a big thanks in helping me navigate the process of life as a graduate student in the department.

I was fortunate to be a part of a productive research group. Thank you Tyler Joseph, Jie Yuan, Ryan Bernstein, Shuo Yang as well as Sitara Persaud, Raiyan Khan and Daniel Li for listening and offering support throughout lab meetings. I have always thought that good research is result of fruitful collaborations. The publications described in this dissertation were done with incredibly talented coauthors Daniel Hernandez-Diaz, Andrew Atkinson-Stirn, Gabriel Marks, Shreya Saxena, Zizhao Wang, Luhuan Wu and Liyi Zhang. I also want to give a huge thanks to Evan Archer and Christian Naesseth for their technical expertise. Thank you professors Iddo Drori, Jose Blanchett, Liam Paninski and John Cunningham for general research advice. I appreciate the opportunity to learn from each of you.

The friends I made at Columbia and at CUNY including Sakhar Al-Khereyf, Hooshmand Shokri Razhagi, Michael Iannelli, Ayman Zeidan, Basak Taylan, Satesh Ramdhani, Justin Agbata, Plapa Koukpamou, Claire Olsson among others helped me navigate the challenges of life as a graduate student. Professors Robert Haralick and Lucas Parra from CUNY provided encouragement and guidance in the early days of my graduate studies. I want to thank Jerée Matherson for supporting me through difficult times in my life, and to thank to Weusi Berry, Jerard Matherson and Mrs Cherry for their care as well. Thank you Kai Perry Parker, Kofi Afful, Benjamin Saah, Kimone Antoine, Jordan Smalls, Deshaun Mars, Nnaemeka Echibiri, Clarence Agbi, David Alade, Nadia Abouzaid, Aimee Elivert, Jubilant Moy, Lorenzo Shabazz Sidberry and others for helping me stay grounded. Thank you Ijeoma and to Patience and Ethelbert Ekeocha.

Finally but not least of all I would like to thank my siblings Nkrirote, Niccolo, Kathurima, my cousins, my uncle Michael and Aunt Toni for hosting me in Berkeley and my entire family. If I have not named anyone explicitly please forgive me. I would like to acknowledge my three parents, Cheryl Mwaria, Yusif Simaan and Frank Moretti for the many sacrifices they have made and for their endless kindness, wisdom and love.

For my family

Chapter 1

Introduction

1.1 The Rise of Bayesian Inference

Bayesian statistical inference has seen a rapid growth in popularity over the past several decades. This development is due to advancements in approximate inference techniques coinciding with increases in computational resources. As transistor counts across microprocessors have skyrocketed, what were once theoretically appealing methodologies applicable only to textbook problems are now the predominant approach to modern machine learning. Computational statistics and Bayesian machine learning play a central role within the natural sciences, however the life sciences and the field of biology is uniquely positioned to undergo a historical period of discovery analogous to that of the early 20^{th} century for the physical sciences. Arguably, the driver of this progress is a host of new experimental tools for collecting massive amounts of data, which in turn has elicited a demand for new computational and statistical techniques to interpret this data. This thesis attempts to meet this demand by proposing novel Bayesian statistical methods and approximate inference techniques for analyzing structured neural and molecular sequence data.

1.1.1 Thesis Overview

In the first part of this thesis, we summarize preliminaries on approximate inference and address the problem of nonlinear blind source separation. Chapter 2 proposes an amortized variational inference method to separate a set of overlapping signals into spatially localized source functions without knowledge of the original signals or the mixing process. We show that under this setup, model parameters scale independently of dataset size making it possible to perform inference on large temporal sequences of functional magnetic resonance imaging data. In the second part of this dissertation, we discuss two approaches for uncovering nonlinear, smooth latent dynamics from sequential data. Both algorithms construct variational families on extensions of nonlinear state space models where the underlying systems are described by hidden stochastic differential equations. The first method in Chapter 3 utilizes a structured approximate posterior describing spatially-dependent linear dynamics, as well as an algorithm that relies on the fixed-point iteration method to achieve convergence. The second method in Chapter 4 proposes a variational backward simulation technique from an unbiased estimate of the marginal likelihood defined through a subsampling process. In Chapter 5, we develop connections between discrete and continuous variational sequential search for Bayesian phylogenetic inference. We propose a technique that uses sequential search to construct a variational objective defined on the composite space of non-clock phylogenetic trees. Chapter 6 offers directions for future work and concluding thoughts.

The remainder of the introduction is organized as follows. Section 1.2 provides a review of recent advances in approximate Bayesian inference including variational inference, Sequential Monte Carlo and Markov Chain Monte Carlo methods. Section 1.3 provides an

outline of this thesis and summarizes the contributions of each subsequent chapter.

1.2 Recent Advances in Approximate Bayesian Inference

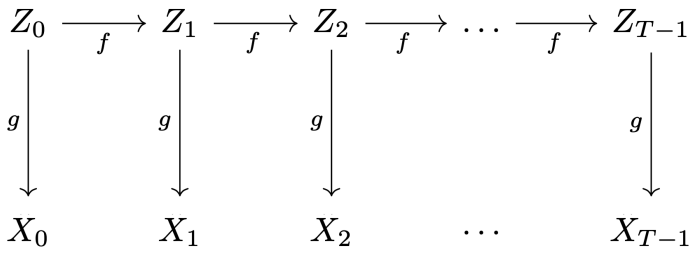


Figure 1.1: Graphical model for the HMM with transition and emission functions f and g denoted. Closed-form inference is not possible when f and g are non-conjugate or nonlinear.

1.2.1 Inference in State Space Models

Let $\mathbf{X} \equiv \{\mathbf{x}_1, \dots \mathbf{x}_T\}$ denote a sequence of T observations of a $\mathbb{R}^{d_\mathbf{x}}$-dependent random variable. State space models (SSMs) posit a generating process for $\mathbf{X}$ through a sequence $\mathbf{Z} \equiv \{\mathbf{z}_1, \dots \mathbf{z}_T\}$, $\mathbf{z}_t \in \mathbb{R}^{d_\mathbf{z}}$ of unobserved latent variables, that transitions according to a stochastic evolution law. The joint density then factorizes:

$$p_\theta(\mathbf{X}, \mathbf{Z}) = F_\theta(\mathbf{Z}) \cdot \prod_{t=1}^{T} g_\theta(\mathbf{x}_t|\mathbf{z}_t)\,, \tag{1.1}$$

where $g_\theta(\mathbf{x}|\mathbf{z})$ is an observation model, and $F_\theta(\mathbf{Z})$ is a prior representing the evolution in the latent space. Here we focus on the case of Markov evolution with Gaussian conditionals:

$$F_\theta(\mathbf{Z}) = f_1(\mathbf{z}_1) \prod_{t=2}^{T} f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1})\,, \tag{1.2}$$

$$f_1 = \mathcal{N}\big(\psi_1, \mathbf{Q}_1\big)\,, \qquad \mathbf{z}_t \sim \mathcal{N}\big(\psi_\theta(\mathbf{z}_{t-1}),\, \mathbf{Q}\big)\,. \tag{1.3}$$

Inference in SSMs requires marginalizing the joint distribution with respect to the hidden variables $\mathbf{Z}$,

$$\log p_\theta(\mathbf{X}) = \int \log p_\theta(\mathbf{X}, \mathbf{Z})\, d\mathbf{Z}. \tag{1.4}$$

Eq. (1.4) is intractable when $\psi_\theta(\mathbf{z}_t)$ is a nonlinear function or when $g_\theta(\mathbf{x}_t|\mathbf{z}_t)$ is non-Gaussian.

1.2.2 Inference and Learning

We are often interested in two distinct tasks for nonlinear SSMs which we define below.

1. **Inference** (*marginalization*) requires sampling latent trajectories $\mathbf{Z}_{1:T}$ to compute an intractable marginal likelihood: $p_\theta(\mathbf{X}_{1:T})$.

2. **Learning** (*optimization*) requires recovering *transition* $f(\cdot)$ and *emission* $g(\cdot)$ functions by maximizing a lower bound to Eq. (1.4).

We define the filtering posterior $p_\theta(\mathbf{Z}_{1:t}|\mathbf{X}_{1:t})$ by the use of information only up to the current time point to estimate the latent state. In contrast, the smoothing posterior $p_\theta(\mathbf{Z}_{1:t}|\mathbf{X}_{1:T})$ uses information from the complete time-ordered sequence of observations to estimate the latent state. A variety of methods have been proposed to address each of these tasks on the premise that smoothing improves the quality of learned nonlinear dynamics.

1.2.3 The Monte Carlo Principle

One of the main challenges in Bayesian inference is numerical integration. Monte Carlo simulation is a straightforward method for approximating integrals or intractable summations via tractable sums. The idea is to draw a set of i.i.d. samples $\{\mathbf{z}^i\}_{i=1}^N$ to evaluate a target density $\pi(\mathbf{z})$ defined on a high dimensional space $\mathcal{X}$. The target measure is then

approximated using the empirical mass function,

$$\pi_N(\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{z}^i}, \tag{1.5}$$

where $\delta_{\mathbf{z}^i}$ denotes the Dirac delta. Convergence of the approximation I_N to the integral I is established by the Law of Large Numbers:

$$I_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{z}^i) \underset{N\to\infty}{\overset{a.s.}{\longrightarrow}} \int_{\mathcal{X}} \pi(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}\,. \tag{1.6}$$

The estimate $I_N(f)$ is unbiased and converges almost surely to $I(f)$ by the Strong Law of Large Numbers (SLLN). If the variance of $f(\mathbf{z})$ is finite and satisfies $\sigma_f^2 = \underset{\pi(\mathbf{z})}{\mathbb{E}}(f^2(\mathbf{z})) - I^2(f) < \infty$, the variance decrease as a function of N so that Var $I_N(f) = \sigma^2(f)/N$. The Central Limit Theorem (CLT) then provides a convergence in distribution of the error:

$$\sqrt{N}\left((I_N(f) - I(f))\right) \underset{N\to\infty}{\Longrightarrow} N(0, \sigma_f^2). \tag{1.7}$$

1.2.4 Variational Bayesian Inference

VI describes a family of techniques for approximating $\log p_\theta(\mathbf{X})$ when marginalization is analytically impossible. The idea is to define a tractable distribution $q_\phi(\mathbf{Z}|\mathbf{X})$ and then optimize a lower bound to the log-likelihood:

$$\log p_\theta(\mathbf{X}) \geq \mathcal{L}_{\text{ELBO}}(\theta, \phi, \mathbf{X}) = \underset{q}{\mathbb{E}}\left[\log \frac{p_\theta(\mathbf{X}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X})}\right]. \tag{1.8}$$

Rewrite the likelihood by introducing a density over hidden variables $q_\phi(\mathbf{Z})$ and marginalizing over $\mathbf{Z}$:

$$\log p_\theta(\mathbf{X}) = \int q_\phi(\mathbf{Z}) p_\theta(\mathbf{X}) \frac{q_\phi(\mathbf{Z})}{q_\phi(\mathbf{Z})} d\mathbf{Z} \tag{1.9}$$

Replace $p_\theta(\mathbf{X})$ with $p_\theta(\mathbf{Z}, \mathbf{X})/p_\theta(\mathbf{Z}|\mathbf{X})$:

$$\log p_\theta(\mathbf{X}) = \int q_\phi(\mathbf{Z}) \frac{\log p_\theta(\mathbf{Z}, \mathbf{X})}{q_\phi(\mathbf{Z})} d\mathbf{Z} + \int q_\phi(\mathbf{Z}) \frac{p_\theta(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{Z})} d\mathbf{Z} \tag{1.10}$$

$$\log p_\theta(\mathbf{X}) = \underset{q}{\mathbb{E}}\left[\log p_\theta(\mathbf{Z}, \mathbf{X})\right] + \mathcal{H}\left(q_\phi(\mathbf{Z})\right) + \mathcal{D}_{KL}\left(q_\phi(\mathbf{Z}) || p_\theta(\mathbf{Z}|\mathbf{X})\right) \tag{1.11}$$

Maximizing the ELBO $\mathcal{L}_{ELBO}$ is equivalent to minimizing the KL divergence $\mathcal{D}_{KL}$:

$$\underset{\theta,\phi}{\operatorname{argmin}} \mathcal{D}_{KL}\left(q_\phi(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})\right) \equiv \underset{\theta,\phi}{\operatorname{argmax}} \mathcal{L}_{ELBO} \tag{1.12}$$

In maximizing Eq. (1.8), VI simultaneously performs both inference and learning.

1.2.4.1 Coordinate Ascent Variational Inference

Tractability and expressiveness of the variational approximation $q_\phi(\mathbf{Z}|\mathbf{X})$ are contrasting goals. A simple choice is to consider a variational family in which each variable is independent:

$$q_\phi(\mathbf{z}_1, \cdots, \mathbf{z}_K) = \prod_{j=1}^{K} q_\phi(\mathbf{z}_j). \tag{1.13}$$

This formulation is referred to as the *mean field variational family*. The name *mean field* originates within statistical mechanics in the analysis of phase transitions when relaxing a problem by ignoring second order effects by averaging over degrees of freedom [51, 93]. Coordinate Ascent Variational Inference (CAVI) is a technique to update each factor $q_k(\cdot)$ while fixing the remaining $K-1$ factors by performing coordinate ascent to optimize $\mathcal{L}_{ELBO}$ in Eq. (5.7). Factorizing the joint and entropy terms:

$$\mathcal{L}_{ELBO} = \log p_\theta(\mathbf{x}_{1:n}) + \sum_{j=1}^{K} \mathbb{E}\left[\log p_\theta(\mathbf{z}_j|\mathbf{z}_{1:(j-1)}, \mathbf{x}_{1:n})\right] - \mathbb{E}_j[\log q_j(\mathbf{z}_j)] \tag{1.14}$$

Writing the objective as a function of factor $q(\mathbf{z}_k)$:

$$\mathcal{L}_k = \int q(\mathbf{z}_k)\mathbb{E}_{-k}[\log p(\mathbf{z}_k|\mathbf{z}_{-k}, \mathbf{x})]d\mathbf{z}_k - \int q(\mathbf{z}_k)\log q(\mathbf{z}_k)d\mathbf{z}_k \tag{1.15}$$

Each factor $q(\mathbf{z}_k)$ is a functional. Making use of the Euler-Lagrange equation to write the functional derivative with respect to $q(\mathbf{z}_k)$:

$$\frac{d\mathcal{L}_j}{dq(\mathbf{z}_k)} = \mathbb{E}_{-k}[\log p(\mathbf{z}_k|\mathbf{z}_{-k}, \mathbf{x})] - \log q(\mathbf{z}_k) - 1 = 0 \tag{1.16}$$

The coordinate ascent update is given by:

$$q_k(\mathbf{z}_k) \propto \exp\left(\mathbb{E}_{q_{-k}}[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \mathbf{x})]\right) \tag{1.17}$$

This simplifies given that the denominator of the conditional does not depend on $\mathbf{z}_j$:

$$q_k(\mathbf{z}_k) \propto \exp\left(\mathbb{E}_{q_{-k}}[\log p(\mathbf{z}_j, \mathbf{z}_{-j}, \mathbf{x})]\right) \tag{1.18}$$

CAVI requires analytically evaluating the expectation and renormalizing (1.18) with respect to ϕ_k. This can be done when the complete conditional distribution belongs to a class of exponential family distributions.

Definition 1.2.1. Let X be a random variable with sample space $\mathcal{X} \subset \mathbb{R}^n$ and probability P_θ. The class of models $\mathcal{P} = \{P_\theta, \theta \in \Theta\}$ is an exponential family if the density can be written as follows:

$$p(x|\theta) = h(x) \exp\left(\eta(\theta)T(x) - B(\theta)\right) \tag{1.19}$$

where $h : \mathcal{X} \to \mathbb{R}$, $\eta : \Theta \to \mathbb{R}$, and $B : \Theta \to \mathbb{R}$ and $T(X)$ is the natural sufficient statistic.

1.2.4.2 Stochastic Gradient Variational Inference

When either the model $p(\mathbf{x}, \mathbf{z})$ or the variational approximation $q(\mathbf{z})$ do not meet the requirements for CAVI, it is possible to use stochastic gradients to optimize the ELBO [94, 110]. Stochastic gradient descent iteratively solves for a parameter λ by performing the update

$$\lambda^n = \lambda^{n-1} + \gamma_n \hat{g}(\lambda^{n-1}), \tag{1.20}$$

where the step sizes $\gamma_n \geq 0$, $\sum_n \gamma_n = \infty$, $\gamma_n^2 < \infty$ [106]. Monte Carlo can be used to estimate the gradient of the ELBO via its expectation. The ELBO gradient can be reformulated using a log-derivative trick where,

$$\nabla_\lambda q_\lambda(\mathbf{z}) = q_\lambda(\mathbf{z}) \nabla_\lambda \log q_\lambda(\mathbf{z}). \tag{1.21}$$

Given that the expectation of the score function $\nabla_\lambda \log q_\lambda(\mathbf{z})$ is zero the ELBO gradient can be estimated as follows:

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{N} \sum_{i=1}^{N} \big(\log p(\mathbf{x}, \mathbf{z}^i) - \log q(\mathbf{z}_i)\big) \cdot \nabla_\lambda \log q_\lambda(\mathbf{z}^i) \tag{1.22}$$

Monte Carlo estimates of gradients have several advantages that follow from the law of large numbers. These estimators are simple to simulate, unbiased and consistent. The score function estimator however produces high variance gradient estimates [102].

1.2.4.3 Autoencoding Variational Bayes and Importance Weighted Autoencoders

Auto Encoding Variational Bayes [57] (AEVB) is a method to simultaneously train $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}, \mathbf{z})$. The expectation value in Eq. (5.7) is approximated by summing over samples from the recognition distribution; which in turn are drawn by evaluating a deterministic function of a ϕ-independent random variable (the reparameterization trick).

$$\mathbf{z} = \mu + \Sigma^{1/2}\epsilon \qquad \epsilon \sim \mathcal{N}(0, \mathbb{I}) \tag{1.23}$$

Importance Sampling (IS) is closely related to the ELBO in VI. Consider $R = \frac{p(\mathbf{z},\mathbf{x})}{q(\mathbf{z})}$ where $z \sim q(\mathbf{z})$. It is easy to see that $\mathbb{E}[R] = p(\mathbf{x})$. By Jensen's inequality, $\log p(\mathbf{x}) \geq \mathbb{E}[\log R]$. In AEVB, Monte Carlo samples are used to approximate $\mathbb{E}[\log R]$ analogous to IS estimates of $\mathbb{E}[R]$. When R is concentrated around its mean $p(\mathbf{x})$, Jensen's inequality produces a tighter bound. It is possible to construct estimators with the same mean that are more concentrated, for example the sample average $\frac{1}{M} \sum_{m=1}^{M} R_m$. It follows that $\log p(\mathbf{x}) \geq \mathbb{E}[\log R_m]$. Building upon this, the Importance Weighted Auto Encoder [11, 21] (IWAE) constructs tighter bounds than the AEVB through mode averaging as opposed to mode matching. The idea to achieve a better estimate of the log-likelihood is to draw K samples from the

proposal and to average probability ratios.

$$\begin{aligned}
&\log p(\mathbf{x}) = \\
&\log \mathbb{E}_{z\sim q_\phi(\mathbf{z}_j)}\left[\frac{1}{K}\sum_{j=1}^{K}\frac{p_\theta(\mathbf{x},\mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}\right] \geq \mathbb{E}_{\mathbf{z}_j\sim q_\phi(\mathbf{z})}\left[\log\frac{1}{K}\sum_{j=1}^{K}\frac{p_\theta(\mathbf{x},\mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}\right] \\
&\qquad \coloneqq \mathcal{L}_K(\theta,\phi)
\end{aligned}$$

Weighting samples by the ratio p/q effectively corrects for the approximation by biasing the proposal towards the true posterior. It can be shown that $Y_K \coloneqq \log\frac{1}{K}\sum_{j=1}^{K}\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}$ is a biased estimator for $\log p_\theta(\mathbf{x})$ where the bias is $\mathcal{O}(K^{-1})$.

1.2.5 Structured Generative Models for Smooth Dynamics

A large body of state space models (SSMs) posit a set of time-evolving latent trajectories $\mathbf{z}_{rt} \in \mathbb{R}^m$ governed by linear dynamics [48, 61, 62, 3, 27]:

$$\mathbf{z}_1 \sim \mathcal{N}(\mu_1, \mathbf{Q}_1)\,, \tag{1.24}$$

$$\mathbf{z}_{t+1}|\mathbf{z}_t \sim \mathcal{N}(\mathbf{A}\mathbf{z}_t, \mathbf{Q})\,, \tag{1.25}$$

where $\mathbf{A}$ is an $m\times m$ linear dynamics matrix, and the matrices $\mathbf{Q}_1$ and $\mathbf{Q}$ are the covariances of the initial states and Gaussian noise. Consider an observation model specified by a deterministic rate function $[f(\mathbf{z}_t)]_i$ where the i^{th} element of the rate function and $\mathcal{P}_\lambda(\lambda)$ is a noise model with parameter λ; $f_\psi : \mathbb{R}^m \longrightarrow \mathbb{R}^n$:

$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{P}_\lambda(\lambda_{ti} = [f(\mathbf{z}_t)_i])\,. \tag{1.26}$$

Under this setup, when $\mathcal{P}_\lambda$ is Gaussian with mean parameter λ and linear rate function f, the model reduces to the classical Kalman filter. When $\mathcal{P}_\lambda$ is non-Gaussian or f is nonlinear, conjugacy is broken and inference is intractable.

1.2.6 Linear Dynamical Systems with Nonlinear Observations

The idea of fLDS [27, 3] is to relax the assumption that $f_\psi(\cdot)$ is a linear function and to parameterize $f_\psi(\cdot)$ using a feed forward neural network. Each observation now has a separate nonlinear dependence on the latent variable through the function $f(\mathbf{z}_{rt})_i$:

$$\mathbf{x}_{rti}|\mathbf{z}_{rt} \sim \mathcal{P}_\lambda(\lambda_{rti} = [f(\mathbf{z}_{rt})_i])\,. \tag{1.27}$$

When the noise model is Poisson or Gaussian the setup is referred to as PfLDS and GfLDS respectively. Model fitting is performed via AEVB with a temporally correlated Gaussian approximate posterior:

$$q_\phi(\mathbf{z}_r|\mathbf{x}_r) = \mathcal{N}\left(\mu_\phi(\mathbf{x}_r), \Sigma_\phi(\mathbf{x}_r)\right) \tag{1.28}$$

$$\propto \prod_{t=1}^{T} q_\phi(\mathbf{z}_{rt}|\mathbf{z}_{r(t-1)}) q_\phi(\mathbf{z}_{rt}|\mathbf{x}_{rt}) q_\phi(\mathbf{z}_{r1}) \tag{1.29}$$

where the mean $\mu_r(\mathbf{x}_r)$ is an $mT \times 1$ vector and $\Sigma_\phi(\mathbf{x}_r)$ is an $mT \times mT$ covariance matrix. In the above setup,

$$q_\phi(\mathbf{z}_{r1}) \sim \mathcal{N}(\tilde{\mu_1}, \tilde{Q}_1), \tag{1.30}$$

$$q_\phi(\mathbf{z}_{rt}|\mathbf{z}_{r(t-1)}) \sim \mathcal{N}(\tilde{A}\mathbf{z}_{r(t-1}, \tilde{Q}), \tag{1.31}$$

$$q_\phi(\mathbf{z}_{rt}|\mathbf{x}_{rt}) \sim \mathcal{N}(m_{\tilde{\psi}}(\mathbf{x}_{rt}), c_{\tilde{\psi}}(\mathbf{x}_{rt}))\,, \tag{1.32}$$

where the matrices $\tilde{A}, \tilde{Q}$ and $\tilde{Q}_1$ are $m \times m$ trainable parameters with $m_{\tilde{\psi}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ and $c_{\tilde{\psi}}(\cdot) : \mathbb{R}^m \to \mathbb{R}^{m \times m}$ defined as nonlinear functions of the observations. Specifically, the covariance matrix $c_{\tilde{\psi}}(\mathbf{x}_{rt}) = \left(r_{\tilde{\psi}}(\mathbf{x}_{rt}) c_{\tilde{\psi}}(\mathbf{x}_{rt})^T\right)^{-1}$ is defined as the product as two matrix valued functions. All factors in Eq. (3.7), Eq. (3.8) and Eq. (3.9) are Gaussian so that $q_\phi(\mathbf{z}_{r(1:T)}|\mathbf{x}_{r(1:T)})$ retains Gaussian functional form. Deducing all terms needed for

Eq. (3.3) via normalization yields:

$$\Sigma_\phi(\mathbf{x}_r) = \left(\mathbf{D}^{-1} + \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\right)^{-1} \tag{1.33}$$

$$\mu_\phi(\mathbf{x}_r) = \left(\mathbf{D}^{-1} + \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\right)^{-1} \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\mathbf{M}_\phi(\mathbf{x}_r)\,. \tag{1.34}$$

In the above, $\mathbf{D} = (\mathbb{I} - \mathbf{A})^{-T}\mathbf{Q}(\mathbb{I} - \mathbf{A})^{-1}$ with

$$\mathbf{Q} = \mathbb{I}_{T\times T} \otimes Q = \begin{bmatrix} \tilde{Q}_1 & & & \\ & \tilde{Q} & & \\ & & \ddots & \\ & & & \tilde{Q} \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 0 & & & \\ \tilde{A} & 0 & & \\ & & \ddots & \\ & & \tilde{A} & 0 \end{bmatrix}, \tag{1.35}$$

$$\mathbf{C}_{\tilde{\psi}}(\mathbf{x}_r) = \begin{bmatrix} c_{\tilde{\psi}}(\mathbf{x}_{r1}) & & & \\ & c_{\tilde{\psi}}(\mathbf{x}_{r2}) & & \\ & & \ddots & \\ & & & c_{\tilde{\psi}}(\mathbf{x}_{rT}) \end{bmatrix}, \qquad \mathbf{M}_{\tilde{\psi}}(\mathbf{x}) = \begin{bmatrix} c_{\tilde{\psi}}(\mathbf{x}_{r1}) \\ \vdots \\ c_{\tilde{\psi}}(\mathbf{x}_{rT}) \end{bmatrix} \in \mathbb{R}^{mT} \tag{1.36}$$

Here Σ is a dense matrix whose inverse Σ^{-1} is parameterized as block tri-diagonal. Matrix inversion and sampling is accomplished via Cholesky decomposition. The computation of the lower-triangular factor r is linear in the length of the time series T [116].

1.2.7 Particle Filtering and Sequential Monte Carlo

1.2.7.1 Filtering and Autoencoding SMC

SMC is a family of techniques for inference applicable to SSMs with an intractable joint distribution. The generative model defined in Eq. (1.1) and Eq. (1.2) imply that the likelihood and the posterior satisfy the following recursions,

$$p_\theta(\mathbf{z}_{0:t}, \mathbf{x}_{0:t}) = p_\theta(\mathbf{z}_{0:t-1}|\mathbf{x}_{0:t-1}) \times \frac{f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1})g_\theta(\mathbf{x}_t|\mathbf{z}_t)}{p_\theta(\mathbf{x}_t|\mathbf{x}_{0:t-1})}\,, \tag{1.37}$$

and

$$p_\theta(\mathbf{x}_t|\mathbf{x}_{0:t-1}) = \int f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1})g_\theta(\mathbf{x}_t|\mathbf{z}_t)p_\theta(\mathbf{z}_{t-1}|\mathbf{x}_{0:t-1})d\mathbf{z}_{t-1:t}. \tag{1.38}$$

Given a proposal distribution $q_\phi(\mathbf{z}|\mathbf{x})$, these methods operate sequentially, approximating $p_\theta(\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ (the *target* measure) and its normalization constant $p_\theta(\mathbf{x}_{t:t})$ for each t by performing inference on a sequence of increasing probability spaces. K samples (*particles*) are drawn from the proposal distribution and used to compute importance weights:

$$\mathbf{z}_t^k \sim q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k, \mathbf{x}_t) \ , \quad w_t^k \coloneqq \frac{f_\theta(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k) g_\theta(\mathbf{x}_t|\mathbf{z}_t^k)}{q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k, \mathbf{x}_t)}. \tag{1.39}$$

It is clear that the proposal should be chosen as close as possible to the *optimal form*, $q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}) \propto f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}) g_\theta(\mathbf{x}_t|\mathbf{z}_t)$. While this presents challenges due to intractability, a large body of techniques have been established for developing good approximations. SMC methods make use of a resampling strategy to ensure that particles remain on regions of high probability mass. Without resampling, the variance of the unnormalized importance weights is independent across iterations and increases exponentially with the time index. SMC exploits Markovian assumptions to mitigate sample degeneracy by resampling the particle indices (*ancestors*) according to their weights at the previous time step:

$$a_{t-1}^k \sim \textsc{Categorical}(\cdot|\bar{w}_{t-1}^1, \cdots, \bar{w}_{t-1}^K) \ , \qquad w_t^k \coloneqq \frac{f_\theta(\mathbf{z}_t^k|\mathbf{z}_{t-1}^{a_{t-1}^k}) g_\theta(\mathbf{x}_t|\mathbf{z}_t^{a_{t-1}^k})}{q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^{a_{t-1}^k}, \mathbf{x}_t)}. \tag{1.40}$$

The posterior can be evaluated at the final time step. The functional integral is approximated below where $\delta_{\mathbf{z}_{1:T}^k}(\mathbf{z}_{1:T})$ is the Dirac measure:

$$\sum_{k=1}^K \bar{w}_T^k \delta_{\mathbf{z}_{1:T}^k}(\mathbf{z}_{1:T}) \quad \text{where} \quad \bar{w}_T^k = w_T^k / \sum_{j=1}^K w_T^j. \tag{1.41}$$

Intuitively, it seems inefficient to re-sample particles at iteration $t-1$ without looking at incoming observation $\mathbf{x}_t$. The Auxiliary Particle Filter (APF) [99] aims to guide the proposal into promising regions of state space by sampling an auxiliary variable that weights each particle in terms of compatibility with the current observation. This is accomplished by weighting $\hat{p}(\mathbf{x}_t|\mathbf{z}_{t-1})$ as an approximation to $p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \int g(\mathbf{x}_t|\mathbf{z}_t) f(\mathbf{z}_t|\mathbf{z}_{t-1}) d\mathbf{z}_t$. The use

of the exact $p(\mathbf{x}_t|\mathbf{z}_{t-1})$ is referred to as a *fully adapted* APF. Other approximations and choices of the importance function q_ϕ define many cases of particle algorithms including:

- The Bootstrap particle filter [32], in which the transition function is used to define the proposal distribution:

$$q_\phi(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{t-1}) = f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}) \tag{1.42}$$

- Sequential Importance Sampling-Resampling [22], in which resampling based on importance samples at time $t-1$ is used to remove particles with low weights.

There are abundant connections between SMC and VI. The SMC algorithm is deterministic conditioning on $(\mathbf{z}_{1:T}^{1:K}, a_{1:T-1}^{1:K})$ [77, 66]. This implies that the proposal density can be reparameterized to act as a variational distribution that can be encoded:

$$Q_{\text{SMC}}(\mathbf{Z}_{1:T}^{1:K}, \mathbf{A}_{1:T-1}^{1:K}) \coloneqq \left(\prod_{k=1}^{K} q_{1,\phi}(\mathbf{z}_1^k)\right) \times \prod_{t=2}^{T}\prod_{k=1}^{K} q_{t,\phi}(\mathbf{z}_t^k|\mathbf{z}_{1:t-1}^{a_{t-1}^k}) \cdot \text{CATEGORICAL}(a_{t-1}^k|\bar{w}_{t-1}^{1:K}).$$

The idea of variational and autoencoding SMC methods [77, 66, 90] is to simultaneously train proposal and target distributions where SMC is used to construct the lower bound to the likelihood via Jensen's inequality. SMC is used to approximate the expectation in Eq. (1.8) which is used to define an objective for learning. An unbiased estimate for the marginal likelihood and the corresponding variational objective are defined below:

$$\hat{\mathcal{Z}}_{\text{SMC}} \coloneqq \prod_{t=1}^{T}\left[\frac{1}{K}\sum_{k=1}^{K} w_t^k\right], \quad \mathcal{L}_{\text{SMC}} \coloneqq \mathop{\mathbb{E}}_{Q_{\text{SMC}}}\left[\log \hat{Z}_{\text{SMC}}\right]. \tag{1.43}$$

SMCs resampling step introduces challenges for standard AEVB-style reparameterization due to the CATEGORICAL distribution. This results in gradient estimates which suffer from high variance. One solution is to drop the discrete terms from the gradient estimates, introducing bias to mitigate high variance of the gradient estimator. The trade-off between

bias and variance of the ELBO gradients is explored both theoretically and empirically in subsequent sections of this work.

1.2.7.2 Particle Smoothing with Backward Simulation

Forward Filtering Backward Simulation (FFBSI) [31] is an approach to approximate the smoothing posterior which admits the following factorization

$$p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = p(\mathbf{z}_T|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} p(\mathbf{z}_t|\mathbf{z}_{t+1:T}, \mathbf{x}_{1:T}) \ , \tag{1.44}$$

where, by Markovian assumptions, the conditional backward kernel can be written as:

$$p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T}) = p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:t}) \tag{1.45}$$

$$= \frac{p(\mathbf{z}_t|\mathbf{x}_{1:t}) f(\mathbf{z}_{t+1}|\mathbf{z}_t)}{p(\mathbf{z}_{t+1}|\mathbf{x}_t)} \tag{1.46}$$

$$\propto p(\mathbf{z}_t|\mathbf{x}_{1:t}) f(\mathbf{z}_{t+1}|\mathbf{z}_t). \tag{1.47}$$

FFBSI begins by performing filtering SMC to obtain $\{\mathbf{z}_{1:T}^{1:K}, w_{1:T}^{1:K}\}$ which provides a particulate approximation to the backward kernel:

$$p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \approx \sum_{i=1}^{K} w_{t|t+1}^k \delta_{\mathbf{z}_t^i}(\mathbf{z}_t), \tag{1.48}$$

$$\text{where} \qquad w_{t|t+1}^i = \frac{w_t^i f(\mathbf{z}_{t+1}|\mathbf{z}_t^i)}{\sum_{j=1}^{K} w_t^j f(\mathbf{z}_{t+1}|\mathbf{z}_t^j)}. \tag{1.49}$$

Backward simulation generates states in the reverse-time direction conditioning on future states by choosing $\tilde{\mathbf{z}}_t = \mathbf{z}_t^i$ with probability $w_{t|T}^i$. This corresponds to a *discrete* resampling step in the backward pass. As a result the backward kernel is approximated from particles that are drawn from the proposal $q(\mathbf{z}_t|\mathbf{z}_{t-1})$ in the forward pass. The FFBSI can only generate trajectories suppported by the forward filtering particles, thus limiting the expressiveness of a variational distribution that might be defined using the algorithm.

1.2.8 Markov Chain Monte Carlo

Markov-Chain Monte Carlo (MCMC) describes an alternative family of approximate inference techniques for indirectly sampling from a target distribution $\mathbf{Z}_n \sim \pi(\mathbf{Z}) \propto h(\mathbf{Z})$ where the normalization constant may be unknown. MCMC constructs a Markov Chain $\{\mathbf{Z}_n\}_{n=0}^{\infty}$ whose stationary distribution is $\pi(\mathbf{Z})$. This is done by drawing a *candidate* state from a proposal $\mathbf{Z}' \sim q(\mathbf{Z}|\mathbf{Z}_{n-1})$. The candidate state is then accepted, in which case $\mathbf{Z}_n = \mathbf{Z}'$ or rejected, in which case $\mathbf{Z}_n = \mathbf{Z}_{n-1}$. The probability of accepting a new state $\mathbf{Z}'$ given current state $\mathbf{Z}$ is given by

$$\alpha(\mathbf{Z}, \mathbf{Z}') = min\left(1, \frac{p(\mathbf{Z})q(\mathbf{Z}|\mathbf{Z}')}{p(\mathbf{Z})q(\mathbf{Z}'|\mathbf{Z})}\right), \tag{1.50}$$

which ensures that the chain has detailed balance with respect to π

$$\pi(\mathbf{Z}')K_{MH}(\mathbf{Z}', \mathbf{Z}) = \pi(\mathbf{Z})K_{MH}(\mathbf{Z}, \mathbf{Z}'), \tag{1.51}$$

for the Metropolis-Hastings kernel K_{MH}. The transition kernel is defined below

$$K_{MH}(\mathbf{Z}_{n+1}|\mathbf{Z}_n) = q(\mathbf{Z}_{n+1}|\mathbf{Z}_n)\alpha(\mathbf{Z}_n, \mathbf{Z}_{n+1}) + \delta_{\mathbf{Z}_n}(\mathbf{Z}_{n+1})r(\mathbf{Z}_n) \tag{1.52}$$

where $r(\mathbf{Z}_n)$ is the term corresponding to the rejection of the proposed move,

$$r(\mathbf{Z}_n) = \int_{\mathcal{X}} q(\mathbf{Z}'|\mathbf{Z}_n)\left(1 - \alpha(\mathbf{Z}_n, \mathbf{Z}')\right) d\mathbf{Z}'. \tag{1.53}$$

We summarize the Metropolis-Hastings MCMC algorithm below.

1. Start with initial state $\mathbf{Z}_0$ for $n = 0$.

2. Generate state $\mathbf{Z}' \sim q(\mathbf{Z}|\mathbf{Z}_{n-1})$ and sample $U \sim \text{UNIFORM}(0, 1)$

3. Check if $U \leq \alpha(\mathbf{Z}, \mathbf{Z}')$:

(a) If $U \leq \alpha(\mathbf{Z}, \mathbf{Z}')$, set $\mathbf{Z}_{n+1} = \mathbf{Z}'$

(b) Else if $U > \alpha(\mathbf{Z}, \mathbf{Z}')$, set $\mathbf{Z}_{n+1} = \mathbf{Z}_n$.

4. Set $n = n + 1$ and return to step 2.

Intuitively, for any starting state, the n-th run of K (denoted K^n) has a chance of $\pi(\mathbf{Z})$ being close to $\mathbf{Z}'$ if n is large. This is formalized in the following Theorem.

Theorem 1. *(Fundamental Theorem of Markov Chains) Let $\mathcal{X}$ be a finite set and let $K(\mathbf{Z}, \mathbf{Z}')$ be a Markov Chain indexed by $\mathcal{X}$. If there exists an n_0 such that $K^n(\mathbf{Z}, \mathbf{Z}') \geq n_0$ for all $n > n_0$, then K has unique stationary distribution π, and as $n \to \infty$,*

$$K^n(\mathbf{Z}, \mathbf{Z}') \to \pi(\mathbf{Z}') \quad \textit{for each} \quad (\mathbf{Z}, \mathbf{Z}') \in \mathcal{X}$$

MCMC methods must be run for an infinite amount of time in order to guarantee convergence of $K^n \to \pi$. In practice, it is common to discard initial runs up until a burn-in time. The rate of convergence of $K^n(\mathbf{Z}, \mathbf{Z}') \to \pi(\mathbf{Z}')$ can be studied via the total variation distance between two probabilities,

$$\|K_x^n - \pi\|_{TV} \coloneqq \frac{1}{2} \sum_{\mathbf{Z}'} \|K^n(\mathbf{Z}, \mathbf{Z}') - \pi(\mathbf{Z}')\| \equiv \max_{A \in \mathcal{X}} |K^n(X, A) - \pi(A)| \tag{1.54}$$

where, given K, n, x and $\epsilon > 0$, we seek to find n such that $\|K_x^n - \pi\|_{TV} < \epsilon$. The answer to this question is highly domain specific and has been studied in certain special cases, for a review see [107, 111, 17]. While MCMC is typically used for inference, it can also be used for learning by sampling by generating parameters θ from the Markov chain.

1.2.8.1 Particle MCMC

It is often the case that a likelihood term in the MCMC acceptance ratio is difficult to evaluate. For example, when using MCMC to sample from Eq. (1.1), the likelihood requires

marginalizing Eq. (1.4). The idea of Particle MCMC algorithms (PMCMC) is to use SMC as an unbiased estimate of the marginal likelihood as specified in Eq. (1.43) to define a proposal for MCMC [1]. We summarise Particle Marginal Metropolis Hastings below.

1. Propose a new set of parameters $\theta' \sim q(\theta'|\theta)$

2. Compute approximation of marginal likelihood using SMC:

$$\mathcal{Z}'_{SMC} = \prod_{t=1}^{T} \frac{1}{K} \sum_{k=1}^{K} w_t^k \tag{1.55}$$

3. Form the acceptance ratio:

$$\alpha = \frac{p(\theta')}{p(\theta)} \cdot \frac{\mathcal{Z}'_{SMC}}{\mathcal{Z}^{j-1}_{SMC}} \cdot \frac{q(\theta|\theta')}{q(\theta'|\theta)} \tag{1.56}$$

4. Sample $U \sim \text{UNIFORM}(0, 1)$ and check if $U \leq \alpha$:

   (a) If accepted, set $(\mathcal{Z}^j, \theta^j) \leftarrow (\mathcal{Z}', \theta')$.

   (b) If rejected, set $(\mathcal{Z}^j, \theta^j) \leftarrow (\mathcal{Z}^{j-1}, \theta^{j-1})$.

It is possible to show that Particle Marginal Metropolis Hastings (PMMH) is equivalent to a standard Metropolis-Hastings algorithm on an extended space [1].

1.3 Thesis Outline and Summary of Contributions

The main contributions of this work are in presenting new techniques for approximate inference and learning in Bayesian latent variable models. Each of these techniques are motivated by real problems within computational biology and applied to provide insights into the underlying structure of complex data.

- **Chapter 2: Autoencoding Topographic Factors**

  Topographic factor methods separate a set of overlapping singals into spatially localized source functions without knowledge of the original signals or the mixing process. These methods require underlying structure of the generative model to be held fixed implying parameters that scale linearly with dataset size. We propose Auto-Encoding Topographic Factors (AETF), an amortized variational inference method and structured approximate posterior that does not require sources to be held constant across locations on the lattice. Model parameters scale independently of dataset size making it possible to perform inference on temporal sequences of large 3D image matrices. AETF is evaluated on both simulations and on deep generative models of functional magnetic resonance imaging data. AETF significantly improves upon existing Topographic factor models in computational efficiency and in reconstruction error.

  This work, which was published as [82] was done jointly with Andrew Stirn, Gabriel Marks and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/AETF`.

- **Chapter 3: Nonlinear Evolution from Spatially Dependent Dynamics**

  State space models play a central role in the analysis of high frequency time series data generated from experimental neuroscience techniques. A large collection of data from

the Allen Brain Atlas involves voltage recordings from single cells that are thought to be modeled by a set of nonlinear differential equations. The task of inferring latent structure and learning the stochastic dynamics of these systems is an open problem in statistical neuroscience motivating the development of novel techniques in approximate inference. We develop Variational Inference for Nonlinear Dynamics (VIND), a statistical model and variational inference technique that is able to recover nonlinear, smooth hidden dynamics from sequential data. VIND builds upon fLDS by proposing a generative model with nonlinear evolution in the latent space, as well as an approximate posterior with spatially dependent locally linear dynamics. Efficient inference is performed via an algorithm that leverages the fixed-point iteration method to speed up convergence. We apply VIND to single cell voltage data with state-of-the-art results in reconstruction error and explore the geometry of nonlinear spiking dynamics. We quantify the performance of the latent dynamics VIND by predicting future neural activity, substantially outperforming current methods.

Part of the work described in this chapter is published as part of a larger joint work [38] which was done jointly with Daniel Hernandez, Ziqiang Wei, Shreya Saxena, John Cunningham and Liam Paninski. A Python/Tensorflow implementation of our algorithms can be found online at `https://github.com/dhernandd/vind`.

- **Chapter 4: Particle Smoothing Variational Objectives**

  Sequential Monte Carlo (SMC) and Variational Inference (VI) are two families of approximate inference algorithms for Bayesian latent variable models. A body of recent work uses SMC to construct a *filtered* estimate of the log marginal likelihood which is used to specify a variational objective by forming a lower bound. We present a

novel backward simulation technique and a variational objective constructed from a *smoothed* approximate posterior. Our method sub-samples auxiliary random variables to enhance the support of the proposal and increase particle diversity. Recent literature argues that increasing the number of samples K to obtain tighter variational bounds may hurt the proposal learning, due to a signal-to-noise ratio (SNR) of gradient estimators decreasing at the rate $\mathcal{O}(\sqrt{1/K})$. As a second contribution, we develop theoretical and empirical analysis of the SNR in filtering SMC, which motivates our choice of biased gradient estimators. We prove that introducing bias by dropping CATEGORICAL terms from the gradient estimate or using Gumbel-Softmax mitigates the adverse effect on the SNR. We demonstrate our approach on three benchmark latent nonlinear dynamical systems tasks consistently outperforming filtered objectives when given fewer Monte Carlo samples.

This work, which was published as [83, 85, 86] was done jointly with Zizhao Wang, Luhuan Wu, Iddo Drori and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/PSVO`.

- **Chapter 5: Variational Combinatorial Sequential Monte Carlo**

  Bayesian phylogenetic inference is often conducted via local or sequential search algorithms such as random-walk Markov chain Monte Carlo or Combinatorial Sequential Monte Carlo. These methods sample tree topologies and branch lengths to compute the marginal likelihood, however when leveraged to perform optimization or evolutionary parameter learning, MCMC requires long runs with inefficient state space exploration. Here we introduce Variational Combinatorial Sequential Monte Carlo (VCSMC), a novel Variational Inference method that simultaneously performs both

parameter inference and model learning. VCSMC uses sequential search to construct a variational objective defined on the composite space of phylogenetic trees. We show that VCSMC is computationally efficient and explores higher probability spaces when compared with state-of-the-art Hamiltonian Monte Carlo methods.

This work, which was published as [84] was done jointly with Liyi Zhang and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/phylo`.

- **Chapter 6: Summary and Future Work**

  We summarize the contributions of this thesis and discuss opportunity for extensions, open questions and future work.

Chapter 2

Autoencoding Topographic Factors

Topographic factor models separate a set of overlapping singals into spatially localized source functions without knowledge of the original signals or the mixing process. These methods require underlying structure of the generative model to be held fixed implying parameters that scale linearly with dataset size. We propose Auto-Encoding Topographic Factors (AETF), an amortized variational inference method and structured approximate posterior that does not require sources to be held constant across locations on the lattice. Model parameters scale independently of dataset size making it possible to perform inference on temporal sequences of large 3D image matrices. AETF is evaluated on both simulations and on deep generative models of functional magnetic resonance imaging data. AETF consistently outperforms existing Topographic factor models in reconstruction error.

This work, which is published as [82] was done jointly with Andrew Stirn, Gabriel Marks and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/AETF`.

2.1 Introduction and Motivation

The analysis of biomedical images has accelerated in recent years due to domain specific methodologies developed for multiple application areas. Calcium imaging in neurons [100], transcriptome profiling from single cells [115] and functional imaging of various biomarkers [29, 79] are exciting examples. Latent variable models are the predominant method for visualizing and extracting structure in spatial data. This data is characterized by a location vector $\mathbf{x}_i \in \Omega \subseteq \mathbb{R}^d$ parameterizing each observation $\mathbf{y}(\mathbf{x}_i)$. Given a tensor $\mathcal{Y} \equiv \{\mathbf{y}(\mathbf{x}_1), \cdots, \mathbf{y}(\mathbf{x}_m)\}_{n=1}^N$ of N realizations, each a sequence of m correlated random variables $Y(\mathbf{x}_1), \cdots, Y(\mathbf{x}_m)$, a fundamental challenge is to identify a subset of physical locations that define areas of interest. To this end, lattice based models formalize an encoding of a latent probability distribution over $Y(\mathbf{x}_1), \cdots, Y(\mathbf{x}_m)$ to quantify statistical dependencies based on distance. This representation is often used for Gaussian process regression or Kriging methods to predict covariance structure between hidden variables and observed features across physical location in an ensemble [115]. For example, extracting relevant voxels from a collection of functional images to discover a latent hemodynamic response enables comparing baseline vs pathological populations [121].

Techniques such as robust principal component analysis [12], independent components analysis and dictionary learning are commonly applied to blind source separation problems; however they require an inherently linear demixing or deconvolution and may fail if there is no linear mixture that leads to independent outputs [88]. Notably these methods do not learn a distribution on the lattice that can be used to quantify uncertainty or to generate new data. Topographic factor models [30, 29, 79] are a family of Bayesian variational techniques for images that require underlying structure on the set of random variables to

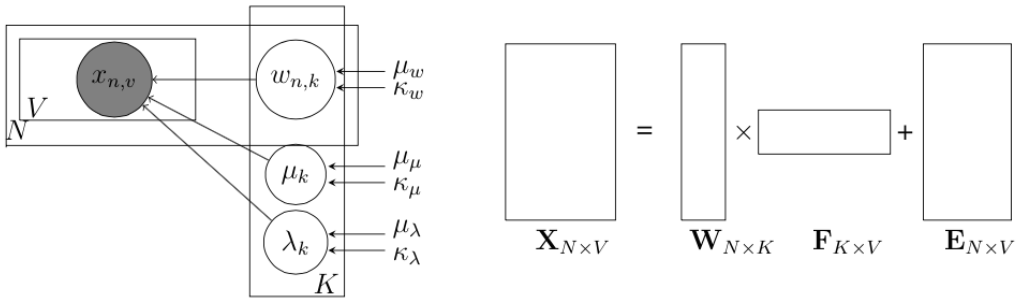


Figure 2.1: Generative graphical model for TFA (left); matrix factorization (right)

be held constant to produce a matrix factorization with spatially interpretable sources.

Here we develop Auto-Encoding Topographic Factors (AETF), a novel Bayesian algorithm to infer spatial dependencies by decomposing observations on a lattice into a weighted set of low rank sources. We are particularly interested in a solution that generalizes to unseen data and that is robust to non-collocated regions of interest. The key insight of AETF is to leverage recent advances in variational inference [29, 103] and Stochastic Gradient Variational Bayes [57, 105] to learn a latent probability model that preserves group variability in spatial structure. Our contributions are to combine two paradigms where convolutional neural networks define the loading matrix and the factor matrix itself maps data to source functions that transform across observations. This is achieved without hard coding hyper parameters that control an a-priori generative model. In doing so, we remove the propensity on initialization of domain specific priors. Experiments on two simulated datasets and on functional imaging data show that our model returns a higher proportion of variance explained than existing Topographic factor models.

2.2 TFA and Standard Lattice Modeling

Following the convention of factor analysis, we assume that our data $\mathbf{Y} \in \mathbb{R}^{N \times V}$ can be decomposed into a set of unobserved weights and latent factors. We use N to denote the number of observations (images), K the number of sources and V the number of lattice positions (voxels). We will be discussing lattices in both 2D as well as 3D for our analysis. Each latent source is defined using a function that assigns a value to each point on the lattice (in voxel space) based on its location. For example, using the MVN:

$$K(\mathbf{x}_i|\mu, \mathbf{\Sigma}) = exp\{ -\frac{1}{2}(\mathbf{x}_i - \mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}_i - \mu)\} \tag{2.1}$$

We posit each observation $\mathbf{y}_n \in \mathbb{R}^{1 \times V}$ has a low rank approximation that is a product of factor loadings $\mathbf{w}_n \in \mathbb{R}^{1 \times K}$ and a factor matrix $\mathbf{F} \in \mathbb{R}^{K \times V}$. The generative distribution of our model factorizes using a Gaussian as follows:

$$P(\mathbf{Y}) = \prod_{n=1}^{N} P(\mathbf{y}_n) \tag{2.2}$$

$$P(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{w}_n\mathbf{F}, \sigma_y^2) \tag{2.3}$$

where σ_y^2 denotes the location or voxel noise. In Manning [80], radial basis source functions $f_k \in \mathbb{R}^V$ are used to generate basis images and to define $\mathbf{F}$, the source image matrix. In general rows of $\mathbf{F}$ are computed by evaluating each of the K source functions at all V lattice points of the voxel space.

While it is common to focus on $\mathbf{\Sigma} = \sigma\mathbb{I}$ or the MVN case in which $\mathbf{\Sigma}$ is full, a larger class of kernels are supported through the Matérn family of covariance functions. Here $K_\nu(\cdot)$ is the modified Bessel function of the second-kind with order parameter ν, where ρ defines correlation length and $\lfloor \nu \rfloor$ describes the smoothness of the process. $\Gamma(\cdot)$ is the gamma

function.

$$K(\mathbf{x}_i|\mu,\nu) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\Big(\frac{\sqrt{2\nu}}{\rho}\cdot\|\mathbf{x}_i-\mu\|\Big)^{\nu}\times K_{\nu}\Big(\frac{\sqrt{2\nu}}{\rho}\|\mathbf{x}_i-\mu\|\Big) \tag{2.4}$$

The above simplifies for half-integer values of ν and reduces to the rational quadratic function with $\nu,\rho>0$ to express a scale mixture of squared exponentials:

$$K(\mathbf{x}_i|\mu,\nu,\rho) = \left(1+\frac{\|\mathbf{x}_i-\mu\|^2}{2\nu\rho^2}\right)^{-\nu} \tag{2.5}$$

Samples from the Gaussian process are $\lfloor\nu-1\rfloor$ times differentiable producing the RBF case when $\nu\to\infty$. As with the above, the choice of distance metric can produce isotropy or anisotropy.

We are interested in the posterior distribution which involves integrating over the set of possible values for the latent variables:

$$P(\mathbf{W},\mathbf{F}|\mathbf{Y}) = \frac{P(\mathbf{Y},\mathbf{W},\mathbf{F})}{P(\mathbf{Y})}\,, \tag{2.6}$$

where the normalization constant requires marginalizing

$$P(\mathbf{Y}) = \int\int P(\mathbf{Y},\mathbf{W},\mathbf{F})d\mathbf{W}d\mathbf{F} \tag{2.7}$$

The denominator is in general intractable to compute. To perform variational inference, a mean field distribution is defined in which each variable is independent:

$$Q(\mathbf{W},\mathbf{M},\mathbf{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(w_{n,k}|\mathbf{m}_{w_{n,k}},\mathbf{\Lambda}_{w_{n,k}})\mathcal{N}(c_{n,k}|\mathbf{m}_{c_{n,k}},\mathbf{\Lambda}_{c_{n,k}})\mathcal{N}(s_{n,k}|\mathbf{m}_{s_{n,k}},\mathbf{\Lambda}_{s_{n,k}}) \tag{2.8}$$

We introduce notation for the set $\phi_k\in\phi$ to denote hyperparameters where c,s,w denote centers, width scales and weights respectively:

$$\phi_k = \{\mathbf{m}_{c,k},\mathbf{\Lambda}_{c,k},\mathbf{m}_{s,k},\mathbf{\Lambda}_{s,k},\mathbf{m}_{w,k},\mathbf{\Lambda}_{w,k}\} \tag{2.9}$$

These allow drawing corresponding latent random variables for centers, width scales and weights for the kth latent source:

$$Z_k = \{z_{c,k}, z_{s,k}, z_{w,k}\} \tag{2.10}$$

where $z_{\xi,k} \sim \mathcal{N}(\mathbf{m}_{\xi,k}, \mathbf{\Lambda}^2_{\xi,k})$ for $\xi \in \{c, s, w\}$. Note that in the isotropic case $\phi \in \mathbb{R}^{K(D+5)}$ and $\mathbf{Z} \in \mathbb{R}^{K(D+2)}$ where D is the dimensionality of the lattice.

Across all ξ, k one can define $\mathbf{m}_\phi = (\mathbf{m}_{\xi,k})_{\forall \xi,k}$ and $\mathbf{\Sigma}_\phi = \mathbf{\Lambda}_\phi \mathbf{\Lambda}_\phi^T$ for $\mathbf{\Lambda}_\phi = (\mathbf{\Lambda}_{\xi,k})_{\forall \xi,k}$, thus the parameters $\mathbf{m}_\phi, \mathbf{\Sigma}_\phi$ denote the means and covariances which are used to draw $\mathbf{Z}$. $\mathbf{Z}$ then defines $\mathbf{F}$, by f_k being a Gaussian function with parameters $z_{c,k}$ and $z_{s,k}$.

2.3 Auto-Encoding Topographic Factors

The idea of AETF is to replace the fixed latent sources by defining a function that parameterizes $\mathbf{Z}$ using the output of a probabilistic encoder. The encoder creates an implicit mapping from each $\mathbf{y}_n \in \mathbf{Y}$ across the set of observations to a unique factor representation while requiring that ϕ encodes the group variability in spatial structure.

Formally, the variational inference framework states the ELBO for the marginal log likelihood $\mathcal{L}(\mathbf{Y}) \leq \log\ p(\mathbf{Y})$ with respect to the variational approximation $q_\phi(\mathbf{Z}|\mathbf{Y})$:

$$\begin{aligned} \mathcal{L}(\mathbf{Y}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\ p_\theta(\mathbf{Y}, \mathbf{Z})] - \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\ q_\phi(\mathbf{Z}|\mathbf{Y})] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\ p_\theta(\mathbf{Y}|\mathbf{Z})] - D_{KL}(q_\phi(\mathbf{Z}|\mathbf{Y})||p(\mathbf{Z}))] \end{aligned} \tag{2.11}$$

We wish to compute the expectation in (2.11) numerically and differentiate with respect to ϕ.

We now rewrite Equation (2.3) as

$$P(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{w}_n(\mathbf{y}_n)\mathbf{F}(\mathbf{y}_n), \sigma_y^2) \tag{2.12}$$

and decompose $\mathbf{F}$ as

$$\mathbf{F}(\mathbf{y}_n) = \begin{pmatrix} f_1(\mathbf{y}_n) \\ \vdots \\ f_K(\mathbf{y}_n) \end{pmatrix}, \tag{2.13}$$

where $f_k(\mathbf{y}_n)$ is the lattice values of a Gaussian function parameterized by $z_{c,k}(\mathbf{y}_n)$ and $z_{s,k}(\mathbf{y}_n)$. $z_{\xi,k}(\mathbf{y}_n)$ itself is a Gaussian latent variable $z_{\xi,k}(\mathbf{y}_n) \sim \mathcal{N}(\mathbf{m}_{k,\xi,\phi}(\mathbf{y}_n), \mathbf{\Lambda}_{k,\xi,\phi}(\mathbf{y}_n))$ whose parameters are the encoder output.

Employing the "reparameterization trick" [57, 105],samples are drawn from $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and transformed:

$$\mathbf{Z}_c = \mu_c + \epsilon \odot \sigma_c \tag{2.14}$$

$$\mathbf{Z}_s = \mu_s + \epsilon \odot \sigma_s \tag{2.15}$$

One is now free to choose the weights $\mathbf{Z}_w \in \phi$ as variational parameters of the recognition model or parameters with the generative model: $\mathbf{Z}_w \in \theta$. Including the weights in ϕ gives:

$$\mathbf{Z}_w = \mu_w + \epsilon \odot \sigma_w \tag{2.16}$$

When $\mathbf{Z}_w \notin \phi$, we learn the weights as point estimates using the update rule:

$$\mathbf{W}^{i+1} \leftarrow \mathbf{W}^i \odot \mathbf{Y}\mathbf{F}(\mathbf{y}_n)^T \oslash \mathbf{W}^i\mathbf{F}(\mathbf{y}_n)\mathbf{F}(\mathbf{y}_n)^T \tag{2.17}$$

Note that the problem is hard due to the non-convexity in the source image matrix. With the parameters ϕ of the recognition model in hand, we have the full model specification. In contrast to standard autoencoder formalization, where the generative model involves a decoder whose parameters need to be inferred, AETF specifies the generative model. We thus compute the approximation $\hat{\mathbf{y}}_n = \mathbf{W}(\mathbf{y}_n) \cdot \mathbf{F}(\mathbf{y}_n)$.

Standard autoencoders learn the respective encoder/decoder parameters θ, ϕ by maximizing the conditional log likelihood $\mathbb{E}_{q(\mathbf{z}|\mathbf{y}^i)}[log\ p_\theta(\mathbf{y}^i|\mathbf{z})]$ by differentiating through $g \leftarrow$

$\nabla_{\theta,\phi}\mathcal{L}^M(\theta,\phi;\mathbf{Y}^M,\epsilon)$ [57, 105]. AETF only needs to learn the encoder parameters ϕ, which is achieved by analogous maximization of the conditional log likelihood $\mathbb{E}_{q(\mathbf{z}|\mathbf{y}^i)}[log\ p(\mathbf{y}^i|\mathbf{z})]$, differentiating through $g \leftarrow \nabla_\phi \mathcal{L}^M(\phi;\mathbf{Y}^M,\epsilon)$.

2.4 Implementation Details

The encoder takes as input an observation and outputs the parameters of the distributions over latent variables. Two recognition models are implemented, one with isotropic and another with full covariance source functions. The isotropic decoder receives as input the sampled latent space vector $\mathbf{Z} \in \mathbb{R}^{k(d+2)}$ including $\mathbf{Z}_c$, $\mathbf{Z}_s$, and $\mathbf{Z}_w$. Note that in the second case of a full covariance matrix $\mathbf{Z}_{\mathbf{s}_\mathbf{\Sigma}k} = \mathbf{\Lambda}\mathbf{\Lambda}^T$, we learn parameters $\mathbf{Z}_{\mathbf{s}_\mathbf{\Sigma}} \in \mathbb{R}^{kd(d+1)/2}$. The spatial factorization constraints of our probability model are imposed within the decoder. Thus unlike traditional variational autoencoders where both the encoder and decoder are neural networks, AETF uses a neural network only for the encoder. The decoder uses the sampled latent space to reconstitute the input according to our imposed factorization and therefore is not parameterized by a neural network.

The encoder network can be comprised of any number of convolutional layers followed by any number of fully-connected layers before the output layer. The convolutional layer executes a $L^{(1)} \otimes \cdots \otimes L^{(D)}$ convolution along the number of lattice dimensions D (where L is specified for each layer) with k (the number of sources) output channels, a bias addition, a *tanh* non-linearity, and max pooling with a $3^{(1)} \otimes \cdots \otimes 3^{(D)}$ kernel and a stride of 1. Our fully-connected layers begin operating on the flattened output of the last convolutional layer or the flattened image if a convolution layer is not employed. Their output dimensions are specified ratiometrically according their output-to-input dimensions. Like most autoencoders, our encoder seeks to compress information. Thus, we only consider output-to-input ratios for

our fully-connected layers that are all less than or equal to 1. These fully-connected layers invoke an affine transformation followed by a *tanh* non-linearity.

Our final output layer varies according to the latent space parameter class. Those parameters that are means (μ_c, μ_s, and μ_e) have no restrictions on their values except the last one, which must be positive. We handle this exception in the decoder. Therefore, we are free to use a vanilla affine transform as the output layers for these parameters. Conversely, those parameters that are standard deviations (σ_c, σ_s, and σ_w) must be greater than or equal to zero. Thus for those standard deviations that parameterize our latent space, we employ an affine transformation followed by a custom non-linearity we call PostReAct (Positive Real Activation in equation 2.18). This non-linearity is a piece-wise combination of a shifted ReLU and a decaying exponential. In this manner, we benefit from ReLU's positive regime that avoids vanishing gradients that are common with double-saturating activations while avoiding the potential of neuron death associated with ReLU's negative regime.

$$\Psi(\lambda) = \begin{cases} \exp(\lambda) & , \lambda < 0 \\ \lambda + 1 & , \lambda \geq 0 \end{cases} \tag{2.18}$$

Our decoder has two responsibilities. First, it constructs the spatial factors using the $\mathbf{Z}_c$ and $\mathbf{Z}_s$ latent space. However and as aforementioned, $\mathbf{Z}_s$ arrives at the encoder on the incorrect support. The RBF function assumes this number is positive. We convert $\mathbf{Z}_s$ to the correct support in two ways. First, we pass it through a PostReAct non-linearity. Second, we square it in our isotropic implementation. Equation (2.19) captures this process that we use for each of our basis image calculations. Here, $f_k(v)$ represents the value of the kth RBF source at voxel position v. Unlike traditional RBF functions, we add a 1 to the denominator to clamp the source's width in a continuously differentiable fashion. Prior

to this modification, sampled $\mathbf{Z}_s$ that resulted in small source widths produced exploding gradients for our optimizer. Once, the decoder constructs the k basis images it recombines them into a single image via a weighted summation that uses $\mathbf{Z}_w$.

$$f_k(v) = \exp\left(- \frac{||\mathbf{Z}_{c,k} - v||_2^2}{2 \cdot \Psi(\mathbf{Z}_{s,k})^2 + 1} \right) \tag{2.19}$$

We present two encoder network architectures. Our first, uses only a 7×7 convolutional layer followed by the output layer. Our second uses–in order of appearance–a 7×7 convolutional layer, a 5×5 convolutional layer, a $1:1$ output-to-input fully-connected layer, and a $4:3$ output-to-input fully-connected layer followed by the output layer. We then permute these two architectures for differing numbers of latent sources. We note that k modifies the size of the network as it determines the number of output channels for each convolutional layer. Our implementation supports imposing a non-negative factorization in addition to one in which the weights are permitted to take negative values.

We modify the loss from equation (2.11). Specifically, we introduce a β term in front of the regularizer as suggested in [69]. Furthermore, they suggest β values less than 1 improve quality. The utilized per-sample loss function for AETF appears in equation (2.20). In our experiments we set β to zero such that our loss reduces to just the reconstruction error. Here, n represents the n^{th} sample and V is the cardinality of our voxel space such that subscript n, i corresponds to the i^{th} voxel of the n^{th} sample.

$$\mathcal{L}(\mathbf{Y}_n) = \frac{1}{V} \sum_{i=1}^{V} \Big[(\hat{\mathbf{Y}}_{n,i} - \mathbf{Y}_{n,i})^2 \Big] + \beta D_{KL}(q_\phi(\mathbf{Z}_i|\mathbf{Y}_i) \;||\; p(\mathbf{Z}_i)) \tag{2.20}$$

2.5 Experiments

Three results are presented, each of which illustrates a strength of the AETF model. We discuss i) fitting in-model synthetic data, ii) fitting non-collocated source functions to smooth,

unmix and localize spatial dependencies in random fields, and *iii*) decomposing thousands of functional images into latent source functions and evaluating our ability to generalize on unseen data.

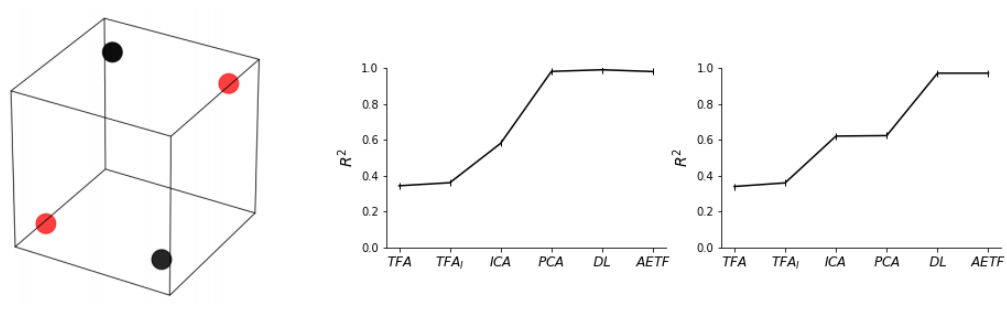


Figure 2.2: Description of the first simulation: (a) Schematic illustrating two source functions located near the vertices of the lattice. Each source transforms across observations drifting between one of two states (denoted with colors red and black); (b) variance explained for different models using two components. AETF outperforms TFA on the train set; (c) TFA, PCA and ICA underperform on the test set.

2.5.1 In-Model Data

We generate a synthetic dataset using $k = 2$ source functions over 1000 observations on a 20× 20×20 lattice. In our experiments, Topographic Factor Analysis (TFA) was unable to run on larger lattice dimensions in $\mathbb{R}^3$. Unlike the generative process specified in TFA [79], the position of each source function may shift across observations and is not restricted to be collocated on the lattice. This design choice is relevant given that the blood oxygen level dependency (BOLD) response is not static and often transforms dynamically as a time series. Figure (2.2a) provides a schematic illustrating the position of two sources located near the vertices of the cube. Each source function is permitted to drift between one of two possible states which are represented using the red and black colors. Figures (2.2b) and (2.2c) provide the variance explained on the training and testing sets respectively using $k = 2$ components. TFA, ICA and PCA underperform relative to Dictionary Learning (DL) and AETF. Unlike DL, AETF is able to parameterize the transforming source functions while maintaining nearly all of the variance explained.

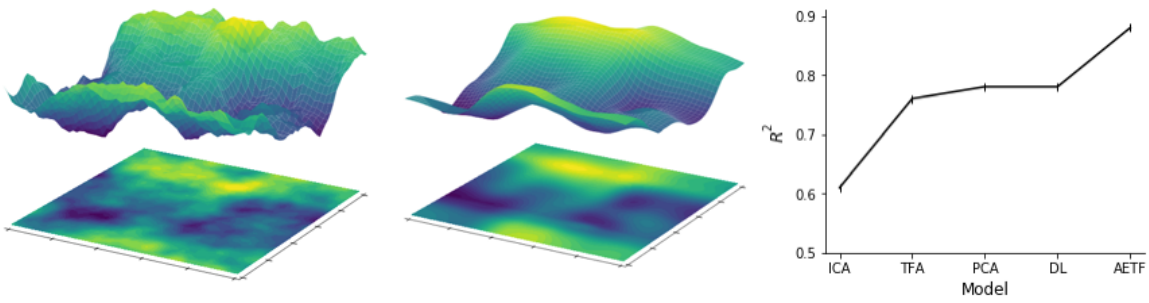


Figure 2.3: Summary of the AETF fit to the GRF simulation: (a) the cross section of a single observation and (b) the cross section of the AETF topographic reconstruction. The surface is shifted above the plane to illustrate the smoothness of the field along with contours presenting the location of the inferred spatial factors; (c) variance explained across models. AETF provides the highest R^2.

2.5.2 Gaussian Random Fields

Gaussian random fields (GRFs) are often used in image analysis to model stochastic processes on a lattice and to introduce noise. We illustrate how Auto-Encoding Topographic Factors recovers autocorrelation structure by filtering a sequence of textscGrfs simulated using spectral methods [2]. The spectral density of a fixed covariance kernel is multiplied with a Fourier transformed white noise field before applying an inverse transform. This process introduces a non-smooth signal in which spatial autocorrelations are not explicitly colocated across observations.

Figure (2.3a) provides a representative sample along with the inferred reconstruction in Figure (2.3b). We fit 10 source functions to 1000 observations on a cubic lattice. As a visualization, the planar cross-section is provided in Figure (2.3). The surface is shifted above the image to illustrate the smoothness of the field along with contours presenting the location of the inferred spatial factors. Figure (2.3c) provides the variance explained across models and fits. Auto-Encoding Topographic Factors outperforms Topographic Factor Analysis both

without and with initialization (denoted TFA and TFA$_I$), ICA, Dictionary Learning (DL), and PCA; the canonical method for Gaussian data. It is clear that Topographic Factor Analysis underperforms when the correlation structure is not held fixed.

2.5.2.1 Image Noise

Spatial factor models learn a smooth statistical map in the presence of noise in which the desired signal extends over several lattice points. A good fit should be robust to variation between observations while preserving correlation structure within the data. To achieve this, Auto Encoding Topographic Factors learns a unique decomposition by simultaneously factorizing the observation matrix, inferring the position of spatial dependencies and introducing flexibility for the location of factors across the lattice. This process is analogous to blurring residual differences in location between comparable areas of activation. When two observations are similar, this is captured in their latent spatial representations. For heterogenous data, AETF parameterizes spatial dynamics.

2.5.2.2 Initializing TFA and HTFA

Heuristics are often suggested to initialize hyper-parameters for Topographic Factor Analysis so that local optima in the source image matrix do not serve as an impediment for non-convex optimization. There exist multiple values of parameters for the location and width of the sources that are equally likely to have generated an observation $\mathbf{y}_n$, due to the rotational invariance of $\mathbf{F}$. One proposed approach is to place hyperparameters a-priori in locations corresponding to high and low activation. Hotspot initialization [79] refers to an iterative process in which the mean image is computed, the mean activation is subtracted and the absolute value is taken of all of the remaining activations. The result is an energy

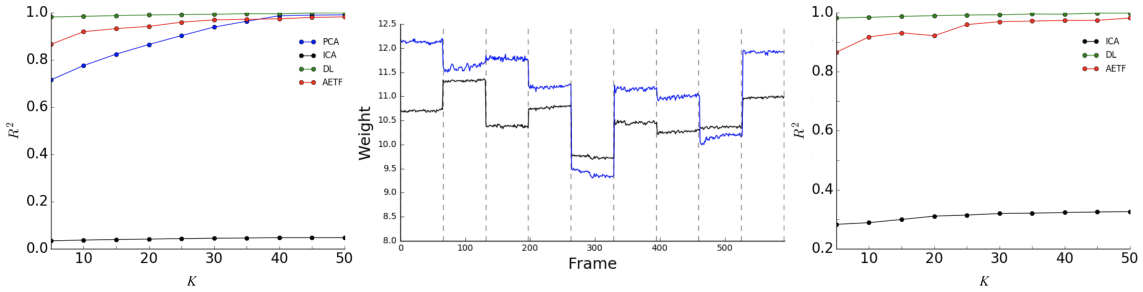


Figure 2.4: Summary of the Sagittal Cross-Section NYU Data: (a) variance explained for various models as a function of number of sources on the training data; (b) two source weights plotted across time frames illustrating strong subject-specific similarities. Dashed vertical lines denote unique subjects; (c) variance explained as a function of number of sources for test data.

landscape in which peaks correspond to extremum. These peaks are iteratively flattened as source centers are placed on these extremum. Values for $\mathbf{m}_{s_{n,k}}$ the mean of the distribution for source $k's$ width scale are then solved for via Newton's method. Once pre-initialized, the source centers and width scales frequently remain fixed. In our experiments, sources for TFA initialized using both hotspot initialization and k-means outperformed experiments with no initialization. Auto-Encoding Topographic Factors outperformed both methods without being contingent upon any such initialization to perform inference successfully.

2.5.3 NYU Dataset

We consider the problem of modeling functional images using the NYU Test-Retest dataset [113]. The data was obtained using a Siemens Allegra 3.0 Tesla scanner. The data consists of twenty six participants each with 3 resting-state scans of 197 continuous EPI functional volumes. Each scan consists of 39 slices of a matrix 64×64 with an acquisition voxel size of $3 \times 3 \times 3$ mm. Scans 2 and 3 were conducted 45 minutes apart roughly 5-16 months after Scan 1.

Slice timing correction, spatial normalization, smoothing and noise stripping were performed using the *Nipype* interface to the FSL software library. The sequential dependency of the time series was not accommodated and each time frame was treated independently. An AETF model was trained using all three sessions reserving 20% for the testing set as a performance criteria to evaluate our fit. To test the significance of the lattice dimensions, models were fit to both sagittal cross-section data and full cubic volumes.

2.5.3.1 Sagittal Cross-Sections in 2D

Sagittal cross-section data was fit to the 13 subjects using the first session. Figure (2.4a) provides the variance explained for AETF, PCA, ICA and DL as a function of number of sources on training data. TFA and HTFA implementations are not supported on the 2D lattice. The R^2 approaches 1 the number of sources K increases. Figure (2.4b) plots the weight values for two randomly selected source functions across a subset of time frames. Dashed vertical lines distinguish subjects. Strong per-subject similarities are visible. Figure (2.4c) provides the variance explained by AETF as K increases on the test data. Using $k = 50$ source functions 99% of variance is explained. We find that $K = 25$ anisotropic sources are sufficient for high quality reconstruction. Interestingly, AETF is able to converge without *any* preprocessing to preserve 89% of total variance on the raw NYU data. On the preprocessed dataset 25 source functions preserve 98% of total variance.

2.5.3.2 Functional Imaging with 3D Volumes

The three-session NYU data on the cubic lattice was modeled using AETF, TFA and HTFA. The $64 \times 64 \times 40$ lattice was divided into eight $20 \times 20 \times 20$ cubic volumes. TFA and HTFA were unable to handle larger lattice dimensions on the full set of 7683 frames. We fit $k = 10$

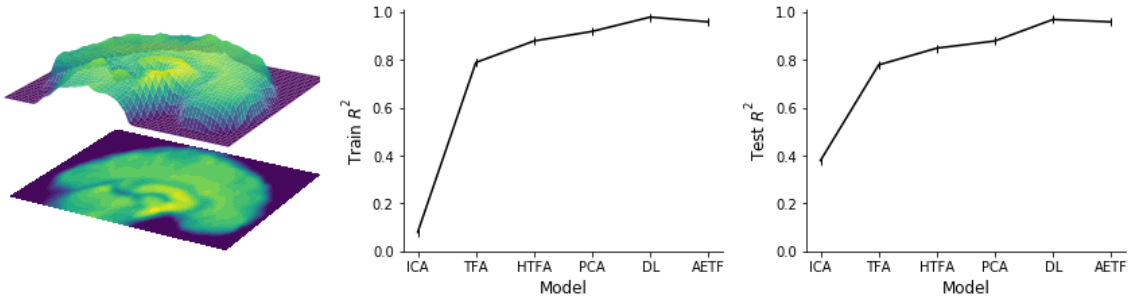


Figure 2.5: Results for the cubic volume NYU data: (a) a cross section of a frame and the surface highlighting source intensities; R^2 values for training (b) and testing (c) for various models averaged across eight cubic volumes using $k = 10$ source functions. AETF consistently outperforms both Topographic Factor Analysis (TFA) and Hierarchical Topographic Factor Analysis (HTFA).

source functions to each cubic volume and average the cost across the total area. For TFA, one model was fit across subjects whereas 39 subjects were fit using HTFA. Figure (2.5a) displays a new frame evaluated using the trained model to illustrate the effect of applying the trained model on unseen data. The surface is plotted above the image to highlight the areas of activation above the corresponding factors on the mesh. Figure (2.5a) and (2.5b) provide the train and test R^2 respectively. It is clear that AETF outperforms both methods. Unlike HTFA, the hierarchical covariance structure is inferred from the data and not specified a-priori.

2.6 Discussion

In the context of functional imaging, a spatial model should be able to extract both global and individual characteristics. In examining how the model parameters for centers, widths and weights varied across testing data, we find source centers are not only similar at the per-subject micro-scale but also marginally similar at the global macro-scale. However, we

see much more global variability with weight values. Compared to a similar factorization in [80] that constricts learning to globally shared sources and individual per-frame weights, our model naturally learns a similar representation. Namely, through a shared encoder mapping, source variability is less pronounced than weight variability.

Auto-Encoding Topographic Factors offers several advantages over unstructured blind source separation techniques. TFA, HTFA, Dictionary Learning, PCA and ICA explicitly learn factor weights (loadings) for each observation. The number of trainable parameters is therefore linear with respect to N, the number of observations. AETF's parameters ϕ are constant with respect to N. This paradigm reduces memory footprint for large N and allows AETF to handle unseen data. By design, the factor images learned by AETF possess lower complexity than the observed images.

AETF can accommodate any priors but is not contingent upon an a-priori choice of generative model hyper-parameters to converge. This is mitigated by choosing uniform priors for the generative model. In this way, AETF is not sensitive to preinitialization issues that plague TFA and HTFA. It is also possible to parameterize the priors of the generative model using a trainable decoder network. Unlike TFA and HTFA, source functions are allowed to transform across individual frames. This is advantageous for time series modeling. In our experiments, Dictionary Learning sometimes provided a comparable R^2. AETF however returns a factorization along with spatially parameterized functions. AETF was written in TensorFlow. The source code and several visualizations are available online.

2.7 Conclusions

We have presented Auto-Encoding Topographic Factors, a novel variational inference scheme for lattice-based measurements in which each observation is given a unique spatial decom-

position. The proposed method is robust to high dimensional data in which sources are not rigidly colocated, introduces non-linearity, supports a family of kernels and the ability to enforce a constrained or non-negative matrix factorization. AETF preserves a large proportion of variance even when factor positions shift dynamically across observations. Highlights include the ability to identify autocorrelation structure in a collection of random fields and the ability to scale to thousands of 3D functional images with a number of training parameters independent of dataset size.

The resultsmotivate an explicitly-hierarchical AETF across individuals, as well as a temporally correlated AETF. A natural extension is to explore the method of normalizing flows [104, 56] as an alternative to defining factors by specifying kernels for source functions. We expect that the approximate posterior would remain simple to compute while each source is permitted to undergo a sequence of transformations giving rise to complex and expressive spatial dependencies.

Chapter 3

Spatially Dependent Locally Linear Dynamics for Single Cell Electrophysiology Data

State space models play a central role in the analysis of high frequency time series data generated from experimental neuroscience techniques. A large collection of data from the Allen Brain Atlas involves voltage recordings from single cells that are thought to be modeled by a set of nonlinear differential equations. The task of inferring latent structure and learning the stochastic dynamics of these systems is an open problem in statistical neuroscience motivating the development of novel techniques in approximate inference. We develop Variational Inference for Nonlinear Dynamics (VIND), a statistical model and variational inference technique that is able to recover nonlinear, smooth hidden dynamics from sequential data. VIND builds upon fLDS by proposing a generative model with nonlinear evolution in the latent space, as well as an approximate posterior with spatially dependent locally linear dynamics. Efficient inference is performed via an algorithm that leverages the fixed-point iteration method to speed up convergence. We apply VIND to single cell

voltage data with state-of-the-art results in reconstruction error and explore the geometry of nonlinear spiking dynamics. We quantify the performance of the latent dynamics VIND by predicting future neural activity, substantially outperforming current methods.

Part of the work described in this chapter is published as part of a larger joint work [38] which was done jointly with Daniel Hernandez, Ziqiang Wei, Shreya Saxena, John Cunningham and Liam Paninski. A Python/Tensorflow implementation of our algorithms can be found online at `https://github.com/dhernandd/vind`.

3.1 Introduction

Conductance based models of excitable cells are widely used in computational neuroscience to describe the spiking activity of individual neurons. One attempt to develop a theory of neural computation is the *dynamical systems hypothesis*, which conjectures that neural computation is explained by dynamics, a branch of mathematics that describes how physical systems change over time [44]. Neuroscientists have long aspired to record from tens of thousands of neurons simoultaneously. Recently, large scale multineuronal neuronal recording technologies such as multielectrode arrays and calcium imaging have opened up avenues for exploration where neural populations as opposed to individual neurons can be studied as the essential units of computation. A fundamental line of research thus involves characterizing the representation and transmission of information recorded from ensembles of neurons. At the other end of the spectrum, there is a collection of high frequency electrophysiological time series data coming from voltage measurements inside single neurons [49]. Here, it is acknowledged that the dynamics are highly nonlinear and multidimensional, although the experimenter only has access to a one-dimensional (1D) voltage measurement. The task, given a 1D or partially observable recording, is thus to approximately recover the

complete latent space paths and dynamics. In each of these situations, the open computational or statistical challenges are how to design algorithms that perform tractable inference on intractable state space models where the underlying dynamics are nonlinear.

We develop Variational Inference for Nonlinear Dynamics (VIND), a statistical model and variational inference technique that is able to recover nonlinear, smooth hidden dynamics from sequential data. VIND builds upon fLDS by proposing a generative model with nonlinear evolution in the latent space, as well as an approximate posterior with spatially dependent locally linear dynamics. Efficient inference is performed via an algorithm that leverages the fixed-point iteration method to speed up convergence. We apply VIND to single cell voltage data with state-of-the-art results in reconstruction error and explore the geometry of nonlinear spiking dynamics. We quantify the performance of the latent dynamics VIND by predicting future neural activity, substantially outperforming current methods.

3.2 Background

3.2.1 Structured Generative Models for Smooth Dynamics

A large body of state space models (SSMs) posit a set of time-evolving latent trajectories $\mathbf{z}_{rt} \in \mathbb{R}^m$ governed by linear dynamics [48, 61, 62, 3, 27]:

$$\mathbf{z}_1 \sim \mathcal{N}(\mu_1, \mathbf{Q}_1)\,, \tag{3.1}$$

$$\mathbf{z}_{t+1}|\mathbf{z}_t \sim \mathcal{N}(\mathbf{A}\mathbf{z}_t, \mathbf{Q})\,, \tag{3.2}$$

where $\mathbf{A}$ is an $m \times m$ linear dynamics matrix, and the matrices $\mathbf{Q}_1$ and $\mathbf{Q}$ are the covariances of the initial states and Gaussian noise. Consider an observation model specified by a deterministic rate function $[f(\mathbf{z}_t)]_i$ where the i^{th} element of the rate function and $\mathcal{P}_\lambda(\lambda)$ is

a noise model with parameter λ; $f_\psi : \mathbb{R}^m \longrightarrow \mathbb{R}^n$:

$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{P}_\lambda(\lambda_{ti} = [f(\mathbf{z}_t)_i]) \,. \tag{3.3}$$

Under this setup, when $\mathcal{P}_\lambda$ is Gaussian with mean parameter λ and linear rate function f, the model reduces to the classical Kalman filter. When $\mathcal{P}_\lambda$ is non-Gaussian or f is nonlinear, conjugacy is broken and inference is intractable.

3.2.2 Linear Dynamical Systems with Nonlinear Observations

The idea of fLDS [27] is to relax the assumption that $f_\psi(\cdot)$ is a linear function and to parameterize $f_\psi(\cdot)$ using a feed forward neural network. Each observation now has a separate nonlinear dependence on the latent variable through the function $f(\mathbf{z}_{rt})_i$:

$$\mathbf{x}_{rti}|\mathbf{z}_{rt} \sim \mathcal{P}_\lambda(\lambda_{rti} = [f(\mathbf{z}_{rt})_i]) \,. \tag{3.4}$$

When the noise model is Poisson or Gaussian the setup is referred to as PfLDS and GfLDS respectively. Model fitting is performed via AEVB with a temporally correlated Gaussian approximate posterior:

$$q_\phi(\mathbf{z}_r|\mathbf{x}_r) = \mathcal{N}\left(\mu_\phi(\mathbf{x}_r), \Sigma_\phi(\mathbf{x}_r)\right) \tag{3.5}$$

$$\propto \prod_{t=1}^{T} q_\phi(\mathbf{z}_{rt}|\mathbf{z}_{r(t-1)}) q_\phi(\mathbf{z}_{rt}|\mathbf{x}_{rt}) q_\phi(\mathbf{z}_{r1}) \tag{3.6}$$

where the mean $\mu_r(\mathbf{x}_r)$ is an $mT \times 1$ vector and $\Sigma_\phi(\mathbf{x}_r)$ is an $mT \times mT$ covariance matrix. In the above setup,

$$q_\phi(\mathbf{z}_{r1}) \sim \mathcal{N}(\tilde{\mu_1}, \tilde{Q}_1), \tag{3.7}$$

$$q_\phi(\mathbf{z}_{rt}|\mathbf{z}_{r(t-1)}) \sim \mathcal{N}(\tilde{A}\mathbf{z}_{r(t-1}, \tilde{Q}), \tag{3.8}$$

$$q_\phi(\mathbf{z}_{rt}|\mathbf{x}_{rt}) \sim \mathcal{N}(m_{\tilde{\psi}}(\mathbf{x}_{rt}), c_{\tilde{\psi}}(\mathbf{x}_{rt}))\,, \tag{3.9}$$

where the matrices $\tilde{A}, \tilde{Q}$ and $\tilde{Q}_1$ are $m \times m$ trainable parameters with $m_{\tilde{\psi}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ and $c_{\tilde{\psi}}(\cdot) : \mathbb{R}^m \to \mathbb{R}^{m \times m}$ defined as nonlinear functions of the observations. Specifically, the covariance matrix $c_{\tilde{\psi}}(\mathbf{x}_{rt}) = \left(r_{\tilde{\psi}}(\mathbf{x}_{rt}) c_{\tilde{\psi}}(\mathbf{x}_{rt})^T\right)^{-1}$ is defined as the product as two matrix valued functions. All factors in Eq. (3.7), Eq. (3.8) and Eq. (3.9) are Gaussian so that $q_\phi(\mathbf{z}_{r(1:T)}|\mathbf{x}_{r(1:T)})$ retains Gaussian functional form. Deducing all terms needed for Eq. (3.3) via normalization yields:

$$\Sigma_\phi(\mathbf{x}_r) = \left(\mathbf{D}^{-1} + \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\right)^{-1} \tag{3.10}$$

$$\mu_\phi(\mathbf{x}_r) = \left(\mathbf{D}^{-1} + \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\right)^{-1} \mathbf{C}_\phi^{-1}(\mathbf{x}_r)\mathbf{M}_\phi(\mathbf{x}_r)\,. \tag{3.11}$$

In the above, $\mathbf{D} = (\mathbb{I} - \mathbf{A})^{-T}\mathbf{Q}(\mathbb{I} - \mathbf{A})^{-1}$ with

$$\mathbf{Q} = \mathbb{I}_{T\times T} \otimes Q = \begin{bmatrix} \tilde{Q}_1 & & & \\ & \tilde{Q} & & \\ & & \ddots & \\ & & & \tilde{Q} \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 0 & & & \\ \tilde{A} & 0 & & \\ & \ddots & & \\ & & \tilde{A} & 0 \end{bmatrix}, \tag{3.12}$$

$$\mathbf{C}_{\tilde{\psi}}(\mathbf{x}_r) = \begin{bmatrix} c_{\tilde{\psi}}(\mathbf{x}_{r1}) & & & \\ & c_{\tilde{\psi}}(\mathbf{x}_{r2}) & & \\ & & \ddots & \\ & & & c_{\tilde{\psi}}(\mathbf{x}_{rT}) \end{bmatrix}, \qquad \mathbf{M}_{\tilde{\psi}}(\mathbf{x}) = \begin{bmatrix} c_{\tilde{\psi}}(\mathbf{x}_{r1}) \\ \vdots \\ c_{\tilde{\psi}}(\mathbf{x}_{rT}) \end{bmatrix} \in \mathbb{R}^{mT} \tag{3.13}$$

Here Σ is a dense matrix whose inverse Σ^{-1} is parameterized as block tri-diagonal. Matrix inversion and sampling is accomplished via Cholesky decomposition. The computation of the lower-triangular factor r is linear in the length of the time series T [116].

3.3 Nonlinear Latent Dynamics with Nonlinear Observations

An extension of fLDS involves a joint density $p(\mathbf{X}, \mathbf{Z})$ which factorizes as follows:

$$p(\mathbf{X}, \mathbf{Z}) \equiv p_{\phi,\theta}(\mathbf{X}, \mathbf{Z}) = c_{\phi,\theta} \cdot H_\phi(\mathbf{Z}) \prod_{t=0}^{T} g_\theta(\mathbf{x}_t|\mathbf{z}_t)\,, \tag{3.14}$$

with distribution parameters ϕ, θ denoted explicitly and unnormalized observation model g_θ. As in fLDS, g_θ can be either Gaussian, $\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}\big(m_\theta(\mathbf{z}_t), \mathbf{\Sigma}\big)$, or Poisson, $\mathbf{x}_t|\mathbf{z}_t \sim \text{Poisson}\big(\lambda_\theta(\mathbf{z}_t)\big)$ with mean $m_\theta(\mathbf{z}_t)$ and rate $\lambda_\theta(\mathbf{z}_t)$ parameterized as nonlinear functions of latent state $\mathbf{z}_t$ represented as neural networks. $c_{\phi,\theta}$ is a normalization constant, $\mathbf{\Sigma}$ is a $\mathbf{z}_t$ independent covariance matrix and H_ϕ is a Markovian latent evolution term [48, 61, 62, 3, 27]:

$$H_\phi(\mathbf{Z}) = h_0(\mathbf{z}_0) \prod_{t=1}^{T} h_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}) \,, \tag{3.15}$$

$$\mathbf{z}_0 \sim \mathcal{N}\big(a_0, \Gamma_0\big) \,, \tag{3.16}$$

$$\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}\big(a_\phi(\mathbf{z}_{t-1}),\, \Gamma\big) \,. \tag{3.17}$$

We wish to represent $a_\phi(\mathbf{z})$ as a nonlinear function parameterized by a neural network with Γ as a trainable parameter. Combining Eq. (3.14) and the posterior distribution of the Generative Model (GM) can be factorized as

$$p_{\phi,\theta}(\mathbf{Z}|\mathbf{X}) = \frac{c_{\phi,\theta} \prod g_\theta(\mathbf{x}_t|\mathbf{z}_t) \cdot H_\phi(\mathbf{Z})}{p_{\phi,\theta}(\mathbf{X})} \,. \tag{3.18}$$

Marginalizing (3.18) to compute the evidence is intractable due to the nonlinearity in $H_\phi(\mathbf{Z})$.

Variational Inference. VI is a technique for approximating the posterior $p(\mathbf{Z}|\mathbf{X})$ when marginalization of latent variables is not analytically feasible. The idea is to introduce a tractable distribution q and to form a lower bound to the log-likelihood:

$$\log p(\mathbf{X}) \geq \mathcal{L}_{ELBO}(\mathbf{X}) = \mathop{\mathbb{E}}_{q}[\log p(\mathbf{X}, \mathbf{Z})] - \mathop{\mathbb{E}}_{q}[\log q(\mathbf{Z})] \,. \tag{3.19}$$

Autoencoding Variational Bayes (AEVB) simultaneously trains q and p. The expectation in Eq. (3.19) is approximated by averaging Monte Carlo samples from q which are reparameterized by evaluating a deterministic function of a ϕ-independent random variable.

3.4 Variational Inference for Nonlinear Dynamics

Approximate Posterior. In designing a variational approximation, there exists a trade-off between tractabiliy and expressiveness of the approximate posterior. An expressive variational approximation would represent the nonlinear evolution in latent space by including the prior term H_ϕ in Eq. (3.18). Consider a recognition model that shares the nonlinear evolution term with the generative model:

$$Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) = \kappa_{\phi,\varphi}(\mathbf{X})\, G_\varphi(\mathbf{X},\mathbf{Z}) H_\phi(\mathbf{Z})\,, \tag{3.20}$$

where $\kappa_{\phi,\varphi}$ is a normalization constant and where G_φ factorizes as follows,

$$G_\varphi(\mathbf{X},\mathbf{Z}) = \prod_{t=0}^{T} g_\varphi(\mathbf{z}_t|\mathbf{x}_t)\,, \quad \mathbf{z}_t|\mathbf{x}_t \sim \mathcal{N}(\mu_\varphi(\mathbf{x}_t), \sigma_\varphi(\mathbf{x}_t))\,, \tag{3.21}$$

with $\mu_\varphi(\mathbf{x})$ and $\sigma_\varphi(\mathbf{x})$ defined by nonlinear functions. In this setup however, regardless of the choice of the encoding function G_φ, it is not possible to compute the normalization constant $\kappa_{\phi,\varphi}$ in closed form. After integration with respect to $\mathbf{z}_T$, the non-Gaussian term $h(\mathbf{z}_T|\mathbf{z}_{T-1})$ produces an intractable $\mathbf{z}_{T-1}$-dependent factor, see App. A. Therefore, due to the H_ϕ term, (3.20) cannot be used to directly define a variational approximation.

Parent-Child Approximations. The recognition model of AEVB is responsible for two tasks which we delineate as *inference* and *learning*. The inference task is to evaluate the expectation in Eq. (3.19) by marginalizing with respect to $\mathbf{Z}_{1:T}$. This is accomplished by sampling latent states $\mathbf{Z}_{1:T} \sim q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ and evaluating the ratio of p to q. The learning task is to train the parameters $\Theta \coloneqq (\theta, \phi, \varphi)$ for $q_{\phi,\varphi}$ and $p_{\phi,\theta}$ by differentiating the variational objective $\mathcal{L}_{ELBO}$. VIND offers a solution to the problem of marginilization when using the H_ϕ term to define the recognition model. This effectively allows for an intractable,

unnormalized $Q_{\phi,\varphi}$, which we refer to as the *parent* distribution, to be used as the recognition model for VI. The trick is to use two related approximations for inference and learning.

Consider a Gaussian approximation $q_{\phi,\varphi}$ to the parent $Q_{\phi,\varphi}$, which we refer to as the *child* distribution. Define $q_{\phi,\varphi}$ to be a Laplace approximation to $Q_{\phi,\varphi}$,

$$q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) = \mathcal{N}\big(\mathbf{P}_{\phi,\varphi}(\mathbf{X}), \mathbf{C}^{-1}_{\phi,\varphi}(\mathbf{X})\big)\,. \tag{3.22}$$

By definition, the mean $\mathbf{P}_{\phi,\varphi}$ in Eq. (3.22) is the solution to the following equation in $\mathbf{Z}$,

$$\frac{\partial}{\partial \mathbf{Z}} \log Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) = \mathbf{0}\,, \tag{3.23}$$

and the precision is defined by

$$[\mathbf{C}_{\phi,\varphi}(\mathbf{X})]_{mn} = \frac{\partial^2}{\partial \mathbf{Z}_m \partial \mathbf{Z}_n} \log Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})\bigg|_{\mathbf{Z}=\mathbf{P}_{\phi,\varphi}(\mathbf{X})} \equiv \Big[s_{\phi,\varphi}\big(\mathbf{P}_{\phi,\varphi}(\mathbf{X}), \mathbf{X}\big)\Big]_{mn}\,, \tag{3.24}$$

where Eq. (3.24) defines $s_{\phi,\varphi}$. The samples used to compute Eq. (3.19) can then be taken with respect to $q_{\phi,\varphi}$ in Eq. (3.22). The inference problem becomes tractable since the child distribution $q_{\phi,\varphi}$ is normal, however, by design, the variational parameters in $q_{\phi,\varphi}$ to be optimized are inherited from the parent distribution, $Q_{\phi,\varphi}$ and are used within the objective $\mathcal{L}_{ELBO}$. After training, the H_ϕ term can be replaced back into $Q_{\phi,\varphi}$. This setup defines an expressive variational family parameterizing nonlinear dynamics $a_\phi(\mathbf{z})$ in the latent space.

Fixed-point iteration. In general, Eq. (3.22) does not admit a closed form solution. For any distribution $Q_{\phi,\varphi}$ such that $\log Q_{\phi,\varphi}$ includes terms quadratic in $\mathbf{Z}$, it is always possible to rewrite Eq. (3.23) in the form

$$\mathbf{Z} = r_{\phi,\varphi}(\mathbf{Z}, \mathbf{X})\,, \tag{3.25}$$

where $r_{\phi,\varphi}$ is a nonlinear function that depends on the trainable parameters in Q, the observation sequence $\mathbf{X}_{1:T}$ and the choice of the nonlinearities in H. Eq. (3.25) can now

be solved numerically by applying FPI method. That is, a root for Eq. (3.25) is found by choosing an initial point $\mathbf{P}^{(0)}$ and iterating

$$\mathbf{P}^{(n)} = r_{\phi,\varphi}(\mathbf{P}^{(n-1)}, \mathbf{X}) \,, \tag{3.26}$$

The FPI convergence can be guaranteed via the Picard-Banach-Cacciopoli theorem by ensuring that the eigenvalues of the Jacobian of the map in Eq.(3.26) are bounded. For a discussion, see B.

Spatially Dependent Locally Linear Dynamics. In order to define the recognition model, functions for the G and H terms in $Q_{\phi,\varphi}$ must be specified. The mean μ_φ and the standard deviation σ_φ in Eq. (3.21) are parameterized using deep neural networks:

$$\mu_\varphi = \mathrm{NN}_{\varphi_\mu}(\mathbf{x}_t) \,, \quad \sigma_\varphi = \mathrm{NN}_{\varphi_\sigma}(\mathbf{x}_t) \,. \tag{3.27}$$

The nonlinear dynamics are specified as $a_\phi(\mathbf{z}) = A_\phi(\mathbf{z})\mathbf{z}$, where $A_\phi(\mathbf{z})$ is a state-space dependent $d_Z \times d_Z$ matrix. The latent evolution law H_ϕ is then specified as follows

$$h_\varphi(\mathbf{z}_{t+1}|\mathbf{z}_t) = exp\left\{-\frac{1}{2}\big(\mathbf{z}_{t+1} - A_\varphi(\mathbf{z}_t)\mathbf{z}_t\big)^T \Gamma \big(\mathbf{z}_{t+1} - A_\varphi(\mathbf{z}_t)\mathbf{z}_t\big)\right\} \,, \tag{3.28}$$

where Γ is a constant precision matrix, and $A_\phi(\mathbf{z}_t)$ is defined as follows

$$A_\phi(\mathbf{z}_t) = \mathbb{A} + \alpha \cdot B_\phi(\mathbf{z}_t) \,. \tag{3.29}$$

$\mathbb{A}$ is initialized to the identity, $B_\phi(\mathbf{z}_t) = \mathrm{NN}_{\phi_B}(\mathbf{z}_t)$, and α is a tunable hyperparameter of the model. We refer to this choice of H as a Locally Linear Dynamical System (LLDS). When $\alpha = 0$, both the statistical model and the algorithm of LLDS/VIND reduces to GfLDS/PfLDS, [3, 27]. The parameter α controls the degree of relaxation of the FPI cost from the quadratic form in the variables $\mathbf{Z}$. This plays an important role in convergence analysis.

With terms from Eqs. (3.15) and (3.21), we can now write the log-likelihood of the parent explicitly

$$\log Q_{\phi,\varphi} = \log C_{\phi,\varphi} - \frac{1}{2}\left[(\mathbf{Z}-\mathbf{M}_\varphi)^T \mathbf{\Lambda}_\varphi (\mathbf{Z}-\mathbf{M}_\varphi) + \mathbf{Z}^T \mathbf{S}_\phi(\mathbf{Z})\mathbf{Z}\right] \tag{3.30}$$

where the parameter $\mathbf{M}_\varphi = \{\boldsymbol{\mu}_\varphi(\mathbf{x}_1), \ldots, \boldsymbol{\mu}_\varphi(\mathbf{x}_T)\}$, $\mathbf{\Lambda}_\varphi$ is a block-diagonal precision matrix,

$$\mathbf{\Lambda}_\varphi = \mathrm{diag}\{\sigma(\mathbf{x}_1), \ldots, \sigma_\varphi(\mathbf{x}_T)\}\,, \tag{3.31}$$

and $\mathbf{S}_\phi(\mathbf{Z})$ is a state-space-dependent, block-tridiagonal covariance whose $d_Z \times d_Z$ blocks are given by:

$$\left[S_\phi(\mathbf{Z})\right]_{t,\tau} = \begin{cases} A_t^T \Gamma A_t & \text{for } \tau = t \\ -\Gamma A_t & \text{for } \tau = t+1 \\ -A_t^T \Gamma & \text{for } \tau = t-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

Here $A_t \equiv A_\phi(\mathbf{z}_t)$. Writing the block-tridiagonal form explicitly:

$$= \begin{bmatrix} A_\phi(\mathbf{z}_t)^T \Gamma A_\phi(\mathbf{z}_t) & -\Gamma A_\phi(\mathbf{z}_t) & & \\ -A_\phi(\mathbf{z}_t)^T \Gamma & A_\phi(\mathbf{z}_t)^T \Gamma A_\phi(\mathbf{z}_t) & -\Gamma A_\phi(\mathbf{z}_t) & \\ & & \ddots & \\ & & -A_\phi(\mathbf{z}_t)^T \Gamma & A_\phi(\mathbf{z}_t)^T \Gamma A_\phi(\mathbf{z}_t) \end{bmatrix}. \tag{3.33}$$

We can now derive the FPI equation for the posterior mean, Eq. (3.25), by differentiating Eq. (3.23),

$$r_{\phi,\varphi}(\mathbf{Z}, \mathbf{X}) = \left[\mathbf{\Lambda}_\varphi + \mathbf{S}_\phi(\mathbf{Z})\right]^{-1} \cdot \mathbf{Y}(\mathbf{Z}) \tag{3.34}$$

$$\mathbf{Y}(\mathbf{Z}) = \mathbf{\Lambda}_\varphi \mathbf{M}_\varphi - \frac{1}{2}\mathbf{Z}^T \frac{\partial \mathbf{S}_\phi(\mathbf{Z})}{\partial \mathbf{Z}} \mathbf{Z}\,. \tag{3.35}$$

In the above, the normalization constant $C_{\phi,\varphi}$ is not required for the FPI step nor for the gradient descent step, thus intractability is evaded. As in fLDS, the time complexity of VIND is $O(T)$. In particular the matrix $\mathbf{\Lambda}_\varphi + \mathbf{S}_\phi(\mathbf{Z})$ is block-tridiagonal and can be inverted in linear time [116].

3.5 VIND Algorithm

The VIND algorithm involves two computations within each epoch to perform both inference and learning. We summarize the procedure in Algorithm 1. The inference step is involves freezing the current values of the parameters ϕ, φ and computing a FPI to obtain the mean and variance of a Laplace approximation to the parent. The learning step is an ADAM gradient descent update [55] of $\mathcal{L}_{ELBO}$ with respect to ϕ, φ, θ. Gradients are estimated via the "reparameterization trick", [53, 47]. Samples are taken from child distribution via:

$$\mathbf{Z}_i = \mathbf{P}_{\phi,\varphi}(\mathbf{X}_i) + [\mathbf{C}_{\phi,\varphi}(\mathbf{X}_i)]^{-1/2}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbb{I})\,. \tag{3.36}$$

Note that the FPI produces a closed form expression for $\mathbf{P}^{(n)}_{\phi,\varphi}$ so that derivatives can be taken with respect to the variational parameters. The normalization constant for the child distribution $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ involves the determinant of the precision matrix which can be computed in closed form given that the precision is block-tridiagonal [117]. In practice, we find that $n = 2$ FPI iterations is enough for good convergence results.

Smoothing Dynamics. One desirable property of Algorithm 1 is that the FPI update (Step 9) mixes all the components of the mean $\mathbf{P}_{\phi,\varphi}$. Note that the t-th component of $\mathbf{P}^{(n)}_{\phi,\varphi}$ depends on all the time steps $\mathbf{Z}_{1:T}$, both past and future, in $\mathbf{P}^{(n-1)}_{\phi,\varphi}$ via the inverse covariance in Eq. (C.2). The VIND algorithm is thus a smoother. After training, the parameters ϕ, φ, θ that maximize $\mathcal{L}_{ELBO}$ can be used to obtain $a_\phi(\mathbf{z})$, the dynamical law that propagates the latent trajectories inferred from the data.

Algorithm 2: Variational Inference for Nonlinear Dynamics

Data: At every epoch $\mathbf{P}_i^{(ep)}$ is the numerical estimate of the hidden path for observation i while $\mathbf{P}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i)$ is the ϕ, φ-dependent posterior mean.

1. **for** $i = 1$ **to** N:

$$\mathbf{P}_i^{(ep)} \leftarrow \mathbf{P}_i^{(0)}$$

2. $ep \leftarrow 1, n \leftarrow 0$
3. $\mathbf{P}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i) \leftarrow \mathbf{P}_i^{(ep-1)}$
4. $\mathbf{C}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i) \leftarrow s_{\phi,\varphi}(P^{(ep-1)}, \mathbf{X}_i)$
5. **while not converged**:
   6. Sample from Child $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ via reparameterization:

$$\epsilon \sim \mathcal{N}(0, \mathbb{I}) \qquad \mathbf{Z}_i \coloneqq \mathbf{P}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i) + \left[\mathbf{C}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i)\right]^{-1/2} \epsilon$$

   7. Form Elbo $\mathcal{L}_{VIND}$ and Adam update θ, ϕ, φ

$$\mathcal{L}_{VIND} \coloneqq \frac{1}{M} \sum_{i=1}^{M} \left(\log p_{\theta,\phi}(\mathbf{Z}_i, \mathbf{X}_i) - \log q_{\phi,\varphi}(\mathbf{Z}_i|\mathbf{X}_i)\right)$$

$$\mathcal{L}_{VIND} \leftarrow \nabla_{\theta,\phi,\varphi}\, \mathcal{L}_{VIND}$$

   8. Update $\mathbf{P}$ and carry the Fpi:

$$\mathbf{P}_i^{(ep)} \leftarrow \mathbf{P}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i)\Big|_{\phi,\varphi}$$

   9. **while** $n \leq$ Nfpis:
      10. $\mathbf{P}_i^{(ep)} \leftarrow r_{\phi,\varphi}(\mathbf{P}_i^{(ep)}, \mathbf{X})$
      11. $n \leftarrow n - 1$
   12. Initialize next epoch $ep \leftarrow ep + 1$; $n \leftarrow 0$
   13. $\mathbf{P}_i^{(ep)} \leftarrow \mathbf{P}_i^{(ep-1)}$
   14. $\mathbf{C}_{\phi,\varphi}^{(ep)}(\mathbf{X}_i) \leftarrow s_{\phi,\varphi}(P^{(ep-1)}, \mathbf{X}_i)$
15. **return** latent trajectories $\mathbf{Z}_{1:T}$ and parameters θ, ϕ, φ

3.6 Evaluation Metric

In order to quantify the performance of the trained dynamics, we compute the k-step mean squared error (MSE) and its normalized version, the R_k^2. To do so, the trained transition function is applied to the latent state without any input data over a rolling window of k steps into the future. The emission function is then used to obtain a prediction $\hat{\mathbf{x}}_{t+k}$ which we compare with the observation $\mathbf{x}_{t+k}$.

$$\text{MSE}_k = \sum_{t=1}^{T-k} \left(\mathbf{x}_{t+k} - \hat{\mathbf{x}}_{t+k}\right)^2 , \qquad R_k^2 = 1 - \frac{\text{MSE}_k}{\sum_{t=1}^{T-k} \left(\mathbf{x}_{t+k} - \bar{\mathbf{x}}_k\right)^2} , \tag{3.37}$$

where $\bar{\mathbf{x}}_k$ is the average of $\mathbf{x}_{k+1:T}$. We note that $\mathcal{L}_{ELBO}$ is not a performance statistic that generalizes across models. In contrast, the R_k^2 provides a metric to quantify the inferred dynamics. When examining latent trajectories, there is no interpretability constraint imposed on how the model chooses to represent the latent trajectories. Thus, in principle, any smooth 1-to-1 mappings of a physical trajectory results in an equivalent representation. This would be analogous to replacing variables by some smooth functions of them, which would satisfy a different set of equations and lose interpretability; while still describing the same physics. For this reason, we argue that the focus should be on topological or qualitative features of the inferred trajectories (such as whether there are limit cycles or fixed points) rather than their exact numerical values.

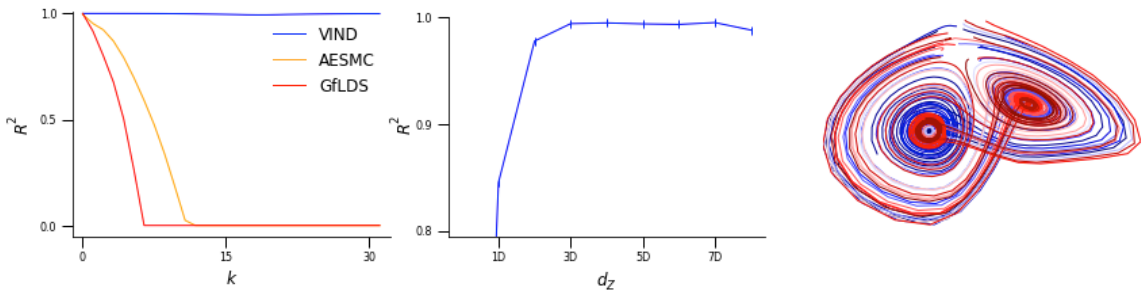


Figure 3.1: Comparison of results for the Lorenz dataset ($d_z = 3$) between GfLDS and VIND: (left) R^2_k comparison; (center) R^2_{10} as a function of dimension of the latent space; (right) VIND's inferred validation trajectories for this dataset.

3.7 Simulations and Real Data Analysis

Before integrating stochastic differential equations, we will utilize the Lorenz system to study VIND's capabilities for inferring chaotic nonlinear dynamics with no input noise in the latent space.

3.7.1 Lorenz Attractor

The Lorenz attractor is a classical system of three coupled nonlinear differential equations:

$$\begin{aligned} \dot{z}_1 &= \sigma(z_2 - z_1)\,, \\ \dot{z}_2 &= z_1(\rho - z_3) - z_2\,, \\ \dot{z}_3 &= z_1 z_2 - \beta z_3\,. \end{aligned} \tag{3.38}$$

Numerical solutions of the Lorenz system with $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ were produced by integrating over 250 time steps from randomly generated initial conditions without noise. A $\mathbf{z}$-dependent neural network was used to map the latent state onto 10D Gaussian observations. The complete synthetic dataset consisted of 100 trials, each comprising 250

time-steps, of which 66% was used for training and the remaining were evenly split for test and validation.

Fig. 3.1 provides the results of the Lorenz experiment. The left panel provide the R_k^2 comparison for VIND and GfLDS fits, with $d_Z = 3$. Remarkably VIND's performance is near perfect over a 30-step forward interpolation. The left panel compares VIND with our implementation of the GfLDS and AESMC algorithms. The center panel illustrates VIND's capability to infer properties of the underlying dynamics: VIND hits peak performance at $d_Z = 3$, the true dimensionality of this system. In the right panel we show the complete set of inferred latent trajectories illustrating the two cycles.

We note that while the latent trajectories are topologically similar to the Lorenz attractor, they do not reproduce it exactly. VIND's decoder can, in principle, learn to undo any smooth transformation applied to the true Lorenz trajectories. As a result, the same set of observations can be described by different sets of latent paths connected by smooth transformations.

3.7.2 Real Data Analysis: Single Cell Voltage Traces

We use VIND to analyze electrophysiology data recorded from single cells. Under this setup, the system is partially observable. The aim is to recover the latent phase space and latent variable trajectories from a single variable. We note that dimensionality expansion is more challenging than dimensionality reduction due to a loss of information. We begin by forming a benchmark dataset from the Allen Brain Atlas [49]

Intracellular voltage recordings from cells from the Primary Visual Cortex of the mouse, area layer 4 were selected. Trials with no spikes were removed, resulting in 44 trials from 7 different cells. The input for each of the remaining trials consists of a step-function with

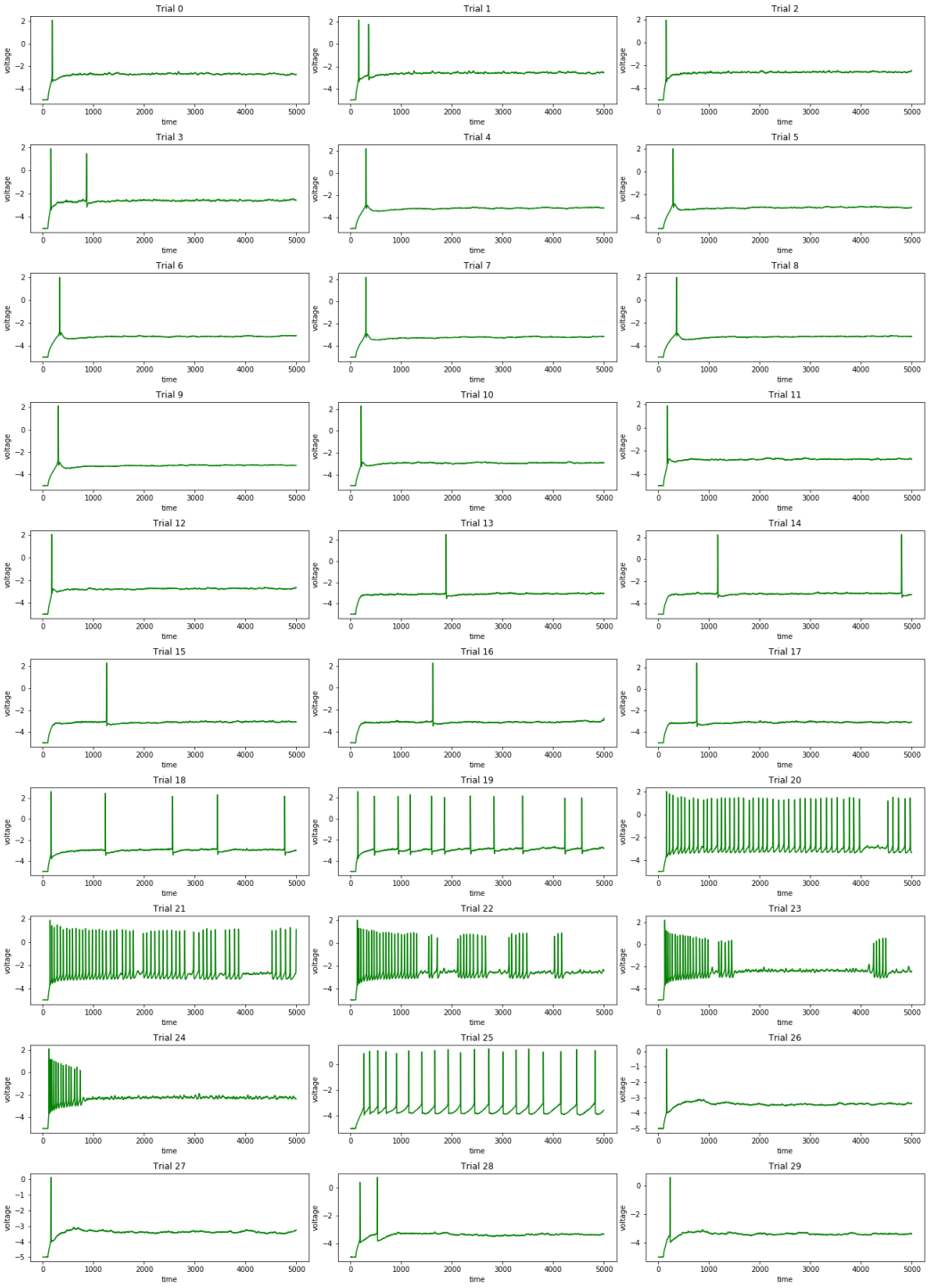


Figure 3.2: The complete set of 30 trials collected from the Allen brain atlas. Neurons respond to an input current. The dataset exhibits a large amount of variability in spiking dynamics.

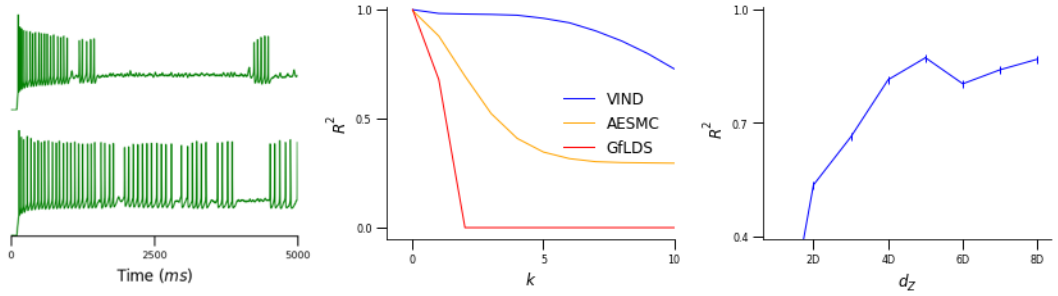


Figure 3.3: Summary of the LLDS/VIND fit to the Allen dataset: (left) The dataset, neurons respond to an input current; (center) VIND vs GfLDS comparison for the best 5D fits; (right) R^2_{10} for different dimensions. The performance increases up to $d_Z = 5$ possibly indicating the hidden dimensionality of the system.

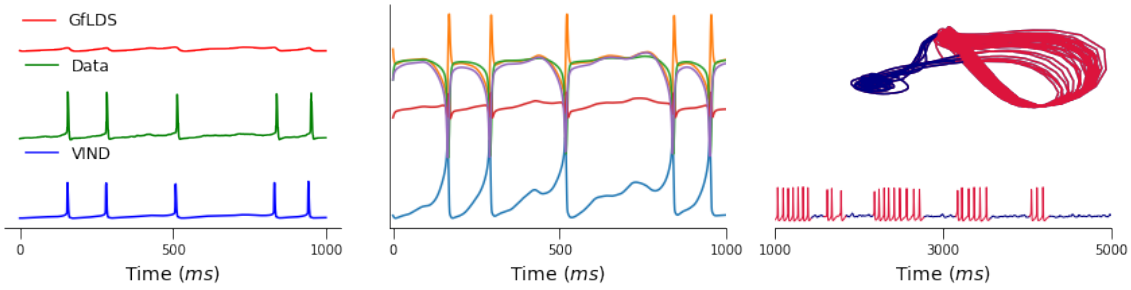


Figure 3.4: Inferred sample paths: (left) Original data (green) versus the 10-step (2ms) forward interpolation given by VIND and by GfLDS; (center) Latent trajectories for a 5D VIND fit of this data, showing behavior similar to the Hodgkin-Huxley gating variables; (right) A 3D cross-section of the latent space showing the representation of the spikes as big cycles (red) and the transient periods (blue).

an amplitude between 80 and 151pA. Observations were split into training (30 trials) and validation sets (14 trials). The data was then down-sampled from $50,000$ time bins (sample rate of 50 kHz) to $5,000$ in equal-time intervals, and subsequently normalized by dividing each trial by its maximal value. Figure 3.2 displays all 30 trials from the training set after preprocessing. The dataset exhibits rich spiking dynamics, some trials spike one time whereas other trials spike between 47 and 52 times.

LLDS/VIND was fit to this data for $d_Z = 2, \ldots, 8$, repeated across 10 runs for a total of 70 full experiments. The top three fits were averaged and the results are summarized in Fig. 3.3. The center panel displays the R^2_{10} values for each choice of latent dimensionality. The fits consistently improve up to $d_Z = 5$, after which there are diminishing returns. We note that single cell voltage data has traditionally been modeled using variants of the classical Hodgkin-Huxley neuron model ([41]), a set of nonlinear differential equations in 4 independent variables, plus an optional independent input current. It is noteworthy that 5 is exactly the minimal number of latent dimensions that provide a good VIND fit for this data. The right panel displays R^2_k with $d_Z = 5$ for VIND, AESMC and for GfLDS. v outperforms GfLDS by an order of magnitude.

The forward-interpolated observations and sample paths for selected runs of VIND and GfLDS are shown in Fig. 3.4. The left panel represents the observations over a rolling window, $k = 10$ time-points in advance for both VIND and GfLDS. The dynamics inferred by GfLDS is unable to capture the nonlinear behavior in both the hyperpolarization and depolarization epochs, a task at which VIND succeeds. The VIND latent trajectories are plotted in the center panel, with the latent dimensions exhibiting similar behavior to that of Hodgkin-Huxley gating variables. In state-space, spikes are represented by big cycles (red), while interspiking fluctuations correspond to separate regions of phase space (blue).

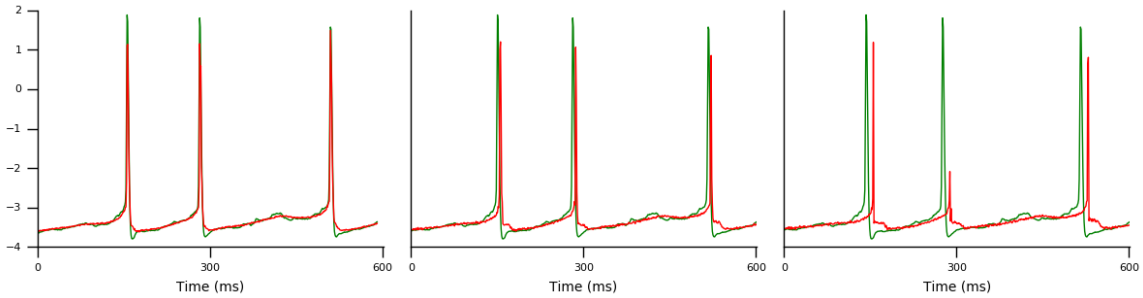


Figure 3.5: Data (green) versus simulation of the observations (red) from the smoothed path: 10 steps ahead (left), 20 steps ahead (center), and 30 steps ahead (right). Some signs of deterioration of the prediction start to appear for the latter (failed spikes, late spiking times).

This is shown in the right panel.

Fig. 3.5 shows simulated paths (forward interpolation with noise) versus the corresponding real data. The expected, progressive deterioration of the VIND prediction as k increases is of note. Fig. 3.6 shows several views of the same two latent paths corresponding to two different input currents showing VIND's different placement of the paths for two different input currents.

Figure 3.6: Different views of a 3D cross section of 5D latent paths for two different trials, showing how the paths occupy different regions of state-space depending on the value of the constant input current.

3.8 Discussion

A body of work has addressed the problem of inference for sequential data. Deep Kalman Filters (DKF) [61] also describes latent nonlinear evolution with nonlinear observations. One key difference between DKF and VIND is that DKF is a *filter* whereas VIND is a *smoother*. That is, DKF uses information only up to the current time point $\mathbf{X}_{1:t}$ to estimate the current latent state $\mathbf{z}_t$, whereas VIND uses information from the complete time ordered observation sequence $\mathbf{X}_{1:T}$ to infer the latent state $\mathbf{z}_t$. The approximate posterior proposed in DKF analogous to Eq. (3.20) is plugged directly into the ELBO, forcing their recognition model to be Gaussian conditioned on observations. VIND is able to perform inference on factorizations of the parent distribution that are unnormalizable. The trick is to compute a second approximation which makes inference tractable because the child distribution that is strictly normal. An extension of DKF was proposed in which parameters between the generative and recognition model are shared. VIND also shares the evolution parameters between the generative model and the inference network. AESMC [66], FIVO [77] and VSMC [89] are methods for model inference and learning where maximize a lower bound to the marginal log likelihood, where Sequential Monte Carlo is used as the likelihood estimator. These three methods also can be implemented to share terms between proposal and target distribution, however, like DKF, they are filters. We propose a novel approach to smooth these methods in Chapter 4.

We have presented a structured approximate posterior describing spatially-dependent linear dynamics to handle intractable distributions, as well as an algorithm that relies on the fixed-point iteration method to achieve convergence. We have introduced a benchmark dataset of single-cell voltage data and demonstrated variational inference in partially observ-

able nonlinear dynamical systems. VIND's fits to electrophysiology data behave qualitatively like Hodgkin Huxley variables and outperform. VIND is implemented for the specific case of Locally Linear Dynamical Systems, which allows for efficient inference with complexity linear in the length of the time series.

Chapter 4

Particle Smoothing Variational Objectives

Sequential Monte Carlo (SMC) and Variational Inference (VI) are two families of approximate inference algorithms for Bayesian latent variable models. A body of recent work uses SMC to construct a filtered estimate of the log marginal likelihood which is used to specify a variational objective by forming a lower bound. We present a novel backward simulation technique and a variational objective constructed from a smoothed approximate posterior. Our method sub-samples auxiliary random variables to enhance the support of the proposal and increase particle diversity. Recent literature argues that increasing the number of samples K to obtain tighter variational bounds may hurt the proposal learning, due to a signal-to-noise ratio (SNR) of gradient estimators decreasing at the rate $\mathcal{O}(\sqrt{1/K})$. As a second contribution, we develop theoretical and empirical analysis of the SNR in filtering SMC, which motivates our choice of biased gradient estimators. We prove that introducing bias by dropping CATEGORICAL terms from the gradient estimate or using Gumbel-Softmax mitigates the adverse effect on the SNR. We demonstrate our approach on three benchmark latent nonlinear dynamical systems tasks consistently outperforming filtered

objectives when given fewer Monte Carlo samples.

This work, which is published as [83, 85, 86] was done jointly with Zizhao Wang, Luhuan Wu, Iddo Drori and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/PSVO`.

4.1 Introduction and Motivation

Latent variable models for time series are often formalized as a set of ordered, discrete-time measurements taken on a hidden dynamical system. A collection of recent work is concerned with inferring both the latent trajectories and latent dynamics of these systems when transition and emission functions are nonlinear [3, 13, 37, 61, 83, 85, 95]. Variational Inference (VI) and Sequential Monte Carlo (SMC) are two families of approximate inference algorithms for non-linear or non-conjugate Bayesian models. Recently, connections have been established between VI and SMC by using the latter to define a flexible variational family for hidden Markov models [66, 77, 89].

Standard variational SMC methods construct a *filtered* estimate of the log marginal likelihood which is used to specify a variational objective by forming a lower bound to the evidence [66, 77, 89, 92, 130]. This enables model learning and inference at the same time. In this approach, however, both the state-sequence and the objective are estimated using information only up to the current time point. This results in degraded posterior estimations when there exists significant observation noise or the system is partially observable. In contrast, particle smoothing methods generate a state-sequence conditioned on future observations [1, 7, 31, 59, 98]. This leads to improved inferred trajectories when the hidden dynamical system is described by a highly nonlinear or chaotic differential equation [37, 95]. For example, neurobiologists measuring a single-dimensional voltage trace are often inter-

ested in recovering nonlinear latent dynamics and trajectories that can be characterized using systems of coupled differential equations such as the Hodgkin Huxley [40]. However, two limitations of the existing particle smoothing literature are as follows:

i) Learning the model parameters that define the transition and emission functions is a distinct task typically handled using an EM algorithm.

ii) The majority of particle smoothing methods do not directly provide an unbiased estimate of the marginal likelihood [7, 59], thus making the construction of a smoothing-based variational objective a challenge.

We highlight the contributions of this section as follows:

- **Particle Smoothing Variational Objective**: We propose Smoothing Variational Objectives (SVO), a framework for performing VI on nonlinear hidden Markov models. SVO jointly estimates the model parameters and the marginal likelihood from the smoothed state-sequence, analogous to the approach of the variational auto-encoder. SVO is a novel recursive backward-sampling algorithm and approximate smoothing posterior defined through a subsampling process. This augments the support of the proposal and boosts particle diversity.

- **SNR Guarantees**: Recent literature argues that increasing the number of samples K to obtain tighter variational bounds may hurt the proposal learning, due to a signal-to-noise ratio (SNR) of gradient estimators decreasing at the rate $\mathcal{O}(\sqrt{1/K})$ [101]. In [66] it was speculated that a result similar to [101] holds for filtering SMC, motivating the design of distinct variational bounds for generative and proposal networks. SMCs resampling step introduces challenges for standard reparameterization

due to the CATEGORICAL distribution. As a second contribution, we analyze the SNR for filtering SMC. We prove that SNR degradation does not apply to the inference network of filtering SMC due to the resampling step. We present theoretical and empirical evidence pointing to an increasing SNR dependent on the choice of the gradient estimator.

- **Unbiased Likelihood Estimator**: We prove that SVO generates an unbiased estimate of the marginal likelihood from the backward state-sequence. We explore the ability of SVO to recover nonlinear embeddings, transition and emission functions from only the observations. To quantify the learned dynamics, we repeatedly apply the trained transition function in the target to propagate the system forwards without input data and then use the emission function to make observation predictions. We show that our smoothed objective generates an improved estimate of the latent state as measured by the ability of the target to more accurately predict observations using the dynamics learned.

- **Applications**: We demonstrate our approach on to three benchmark latent nonlinear dynamical systems tasks, including single cell voltage trace data. SVO outperforms filtered objectives when given fewer Monte Carlo samples on all three tasks.

4.2 Preliminaries

Inference in State Space Models Let $\mathbf{X} \equiv \{\mathbf{x}_1, \dots \mathbf{x}_T\}$ denote a sequence of T observations of a $\mathbb{R}^{d_{\mathbf{x}}}$-dependent random variable. State space models (SSMs) posit a generating process for $\mathbf{X}$ through a sequence $\mathbf{Z} \equiv \{\mathbf{z}_1, \dots \mathbf{z}_T\}$, $\mathbf{z}_t \in \mathbb{R}^{d_{\mathbf{z}}}$ of unobserved latent variables,

that transitions according to a stochastic evolution law. The joint density then factorizes:

$$p_\theta(\mathbf{X}, \mathbf{Z}) = F_\theta(\mathbf{Z}) \cdot \prod_{t=1}^{T} g_\theta(\mathbf{x}_t|\mathbf{z}_t)\,, \tag{4.1}$$

where $g_\theta(\mathbf{x}|\mathbf{z})$ is an observation model, and $F_\theta(\mathbf{Z})$ is a prior representing the evolution in the latent space. In this work, we focus on the case of Markov evolution with Gaussian conditionals:

$$F_\theta(\mathbf{Z}) = f_1(\mathbf{z}_1) \prod_{t=2}^{T} f_\theta(\mathbf{z}_t|\mathbf{z}_{t-1})\,,$$
$$f_1 = \mathcal{N}\big(\psi_1, \mathbf{Q}_1\big)\,, \qquad \mathbf{z}_t \sim \mathcal{N}\big(\psi_\theta(\mathbf{z}_{t-1}),\, \mathbf{Q}\big)\,. \tag{4.2}$$

Inference in SSMs requires marginalizing the joint distribution with respect to the hidden variables $\mathbf{Z}$,

$$\log p_\theta(\mathbf{X}) = \int \log p_\theta(\mathbf{X}, \mathbf{Z})\, d\mathbf{Z}. \tag{4.3}$$

This procedure is intractable when $\psi_\theta(\mathbf{z}_t)$ is a nonlinear function or when $g_\theta(\mathbf{x}_t|\mathbf{z}_t)$ is non-Gaussian.

Variational Inference VI describes a family of techniques for approximating $\log p_\theta(\mathbf{X})$ when marginalization is analytically impossible. The idea is to define a tractable distribution $q_\phi(\mathbf{Z}|\mathbf{X})$ and then optimize a lower bound to the log-likelihood:

$$\log p_\theta(\mathbf{X}) \geq \mathcal{L}_{\text{ELBO}}(\theta, \phi, \mathbf{X}) = \mathop{\mathbb{E}}_{q}\left[\log \frac{p_\theta(\mathbf{X}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X})}\right]. \tag{4.4}$$

Tractability and expressiveness of the variational approximation $q_\phi(\mathbf{Z}|\mathbf{X})$ are contrasting goals. Auto Encoding Variational Bayes [57] (AEVB) is a method to simultaneously train $q_\phi(\mathbf{Z}|\mathbf{X})$ and $p_\theta(\mathbf{X}, \mathbf{Z})$. The expectation value in Eq. (5.7) is approximated by summing over samples from the recognition distribution; which in turn are drawn by evaluating a deterministic function of a ϕ-independent random variable (the reparameterization trick).

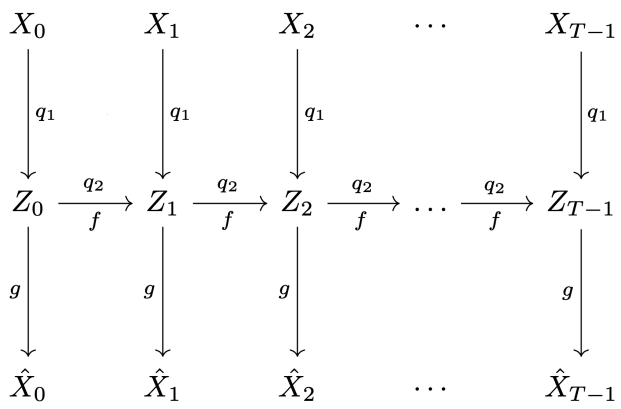


Figure 4.1: SMC terms for the HMM with transition $f(\cdot)$ and emission $g(\cdot)$ functions denoted. Closed-form inference is not possible when f and g are non-conjugate or nonlinear. Parameter estimation is performed via AEVB with nonlinear proposal terms for encoding q_1 and transition q_2 denoted.

Building upon this, the Importance Weighted Auto Encoder [11, 21] (IWAE) constructs tighter bounds than the AEVB through mode averaging as opposed to mode matching. The idea to achieve a better estimate of the log-likelihood is to draw K samples from the proposal and to average probability ratios.

Filtering SMC SMC is a family of techniques for inference in SSMs with an intractable joint. Given a proposal distribution $q_\phi(\mathbf{Z}|\mathbf{X})$, these methods operate sequentially, approximating $p_\theta(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ (the *target*) for each t by performing inference on a sequence of increasing probability spaces. K samples (*particles*) are drawn from a proposal distribution and used to compute importance weights:

$$\mathbf{z}_t^k \sim q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k, \mathbf{x}_t) \ , \quad w_t^k \coloneqq \frac{f_\theta(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k) g_\theta(\mathbf{x}_t|\mathbf{z}_t^k)}{q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^k, \mathbf{x}_t)}. \tag{4.5}$$

A resampling strategy ensures that particles remain on regions of high probability mass. SMC accomplishes this goal by resampling the particle indices (*ancestors*) according to

their weights at the previous time step:

$$a_{t-1}^k \sim \textsc{Categorical}(\cdot|\bar{w}_{t-1}^1, \cdots, \bar{w}_{t-1}^K) \ , \qquad w_t^k \coloneqq \frac{f_\theta(\mathbf{z}_t^k|\mathbf{z}_{t-1}^{a_{t-1}^k}) g_\theta(\mathbf{x}_t|\mathbf{z}_t^{a_{t-1}^k})}{q_\phi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^{a_{t-1}^k}, \mathbf{x}_t)}. \tag{4.6}$$

The posterior can be evaluated at the final time step. The functional integral is approximated below where $\delta_{\mathbf{z}_{1:T}^k}(\mathbf{z}_{1:T})$ is the Dirac measure:

$$\sum_{k=1}^K \bar{w}_T^k \delta_{\mathbf{z}_{1:T}^k}(\mathbf{z}_{1:T}) \quad \text{where} \quad \bar{w}_T^k = w_T^k / \sum_{j=1}^K w_T^j. \tag{4.7}$$

The SMC algorithm is deterministic conditioning on $(\mathbf{z}_{1:T}^{1:K}, a_{1:T-1}^{1:K})$ [77, 66]. This implies that the proposal density can be reparameterized to act as a variational distribution that can be encoded:

$$Q_{\text{SMC}}(\mathbf{Z}_{1:T}^{1:K}, \mathbf{A}_{1:T-1}^{1:K}) \coloneqq \left(\prod_{k=1}^K q_{1,\phi}(\mathbf{z}_1^k) \right) \times \prod_{t=2}^T \prod_{k=1}^K q_{t,\phi}(\mathbf{z}_t^k|\mathbf{z}_{1:t-1}^{a_{t-1}^k}) \cdot \textsc{Categorical}(a_{t-1}^k|\bar{w}_{t-1}^{1:K}). \tag{4.8}$$

An unbiased estimate for the marginal likelihood and the corresponding objective are defined below:

$$\hat{\mathcal{Z}}_{\text{SMC}} \coloneqq \prod_{t=1}^T \Big[\frac{1}{K} \sum_{k=1}^K w_t^k \Big], \quad \mathcal{L}_{\text{SMC}} \coloneqq \mathop{\mathbb{E}}_{Q_{\text{SMC}}} \Big[\log \hat{Z}_{\text{SMC}} \Big] . \tag{4.9}$$

Particle Smoothing with Backward Simulation Forward Filtering Backward Simulation (FFBSi) [31] is an approach to approximate the smoothing posterior which admits the following factorization

$$p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = p(\mathbf{z}_T|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} p(\mathbf{z}_t|\mathbf{z}_{t+1:T}, \mathbf{x}_{1:T}) \ , \tag{4.10}$$

where, by Markovian assumptions, the conditional backward kernel can be written as:

$$p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \propto p(\mathbf{z}_t|\mathbf{x}_{1:t}) f(\mathbf{z}_{t+1}|\mathbf{z}_t). \tag{4.11}$$

FFBSi begins with filtering to obtain $\{\mathbf{z}_{1:T}^{1:K}, w_{1:T}^{1:K}\}$ which provides a particulate approximation to the backward kernel:

$$p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \approx \sum_{i=1}^{K} w_{t|t+1}^{k} \delta_{\mathbf{z}_t^i}(\mathbf{z}_t), \tag{4.12}$$

$$\text{where} \qquad w_{t|t+1}^{i} = \frac{w_t^i f(\mathbf{z}_{t+1}|\mathbf{z}_t^i)}{\sum_{j=1}^{K} w_t^j f(\mathbf{z}_{t+1}|\mathbf{z}_t^j)}.$$

Backward simulation generates states in the reverse-time direction conditioning on future states by choosing $\tilde{\mathbf{z}}_t = \mathbf{z}_t^i$ with probability $w_{t|T}^i$. This corresponds to a *discrete* resampling step in the backward pass. As a result the backward kernel is approximated from particles that are drawn from the proposal $q(\mathbf{z}_t|\mathbf{z}_{t-1})$ in the forward pass. The FFBSi can only generate trajectories suppported by the forward filtering particles, thus limiting the expressiveness of the variational distribution.

4.3 Particle Smoothing Variational Objectives

We will utilize the smoothing posterior in Eq. (4.10) to define a backward proposal distribution and sample trajectories to construct a variational objective. We propose a novel approximate posterior to overcome the limitation of the FFBSi by augmenting the support of the backward kernel through the subsampling of auxiliary random variables.

Overview We provide an overview of Particle Smoothing Variational Objectives (SVO) before presenting a detailed derivation and description in Algorithm 3 (we have annotated the overview with steps from the algorithm). Smoothing is based on filtering SMC which provides the forward weights and particles $\{\mathbf{z}_{1:T}^{1:K}, w_{1:T}^{1:K}\}$ (*step 1*). With outputs from filtering SMC, SVO proceeds to generate backward trajectories. This is done by approximating a sequence of backward posteriors through a process of self-normalized importance sampling. At time T, for each trajectory we will draw M *subparticles* from a continuous-domain conditional kernel (*step 3*). While the final time step requires some care, these subparticles will be used to initialize *subweights* relative to the conditional kernel (*step 4*). The subweights in turn, are used to update the corresponding particle by drawing a backward index from a resampling process (*step 5*). The trajectory is initialized with the selected particle and extended sequentially (*step 6*). SVO iterates by drawing M subparticles from a continuous-domain backward proposal for each of the K trajectories at the current time step (*step 9*). SVO then computes subweights for each subparticle (*step 10*) in order to select a single backward particle from the set of M candidates (*step 11*). Finally the backward kernel is evaluated using the chosen resampled particle (*step 13*). The output of this procedure is a collection of particle trajectories from the smoothing posterior that are used to define a variational objective.

4.4 Approximate Posterior

We introduce a *continuous* reverse-dynamics proposal $q(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T})$ that is used to sample M subparticles for each $k \in \{1, \cdots, K\}$, $\tilde{\mathbf{z}}_t^{k,1:M} \sim q(\mathbf{z}_t|\tilde{\mathbf{z}}_{t+1}^k, \mathbf{x}_{1:T})$. These samples are used

Algorithm 3: Particle Smoothing Variational Objectives

1. Perform forward filtering to obtain $\{\mathbf{z}_{1:T}^{1:K}, w_{1:T}^{1:K}\}$
2. Initialization. For $k = 1, \cdots, K$:
3. Sample M subparticles: $\{\tilde{\mathbf{z}}_T^{k,m}\}_{m=1}^M \sim q(\cdot|\mathbf{x}_{1:T})$
4. Initialize subweight for each subparticle:

$$\omega_{T|T}^{k,m} \propto \Big[\sum_j \bar{w}_{T-1}^j f(\tilde{\mathbf{z}}_T^{k,m}|\mathbf{z}_{T-1}^j)\Big] \frac{g(\mathbf{x}_T|\tilde{\mathbf{z}}_T^{m,k})}{q(\tilde{\mathbf{z}}_T^{k,m}|\mathbf{x}_{1:T})}$$

5. Sample index: $b_T^k \sim \textsc{Categorical}(\cdot|\omega_{T|T}^{k,1}, \cdots, \omega_{T|T}^{k,M})$
6. Set backward particle: $\tilde{\mathbf{z}}_T^k \leftarrow \tilde{\mathbf{z}}_T^{k,b_t^m}$, $\omega_{T|T}^k \leftarrow \omega_{T|T}^{k,b_T^k}$
7. Evaluate the backward proposal: $\Omega_T^k \coloneqq M \cdot \omega_{T|T}^k \cdot q(\tilde{\mathbf{z}}_T^k|\mathbf{x}_{1:T})$,
8. Backward Simulation.

   For $t = T-1, \cdots, 1$ and $k = 1, \cdots, K$:
9. Sample M subparticles from reverse-dynamics proposal:

   $\{\tilde{\mathbf{z}}_t^{k,m}\}_{m=1}^M \sim q(\cdot|\tilde{\mathbf{z}}_{t+1}^k, \mathbf{x}_{1:T})$
10. Compute subweights:

$$\omega_{t|T}^{k,m} \propto \sum_j \bar{w}_{t-1}^j f(\tilde{\mathbf{z}}_t^{k,m}|\mathbf{z}_{t-1}^j) \times \frac{f(\tilde{\mathbf{z}}_{t+1}^k|\tilde{\mathbf{z}}_t^{k,m}) g(\mathbf{x}_t|\tilde{\mathbf{z}}_t^{k,m})}{q(\tilde{\mathbf{z}}_t^{k,m}|\tilde{\mathbf{z}}_{t+1}^k, \mathbf{x}_{1:T})}$$

11. Sample index. $b_t^k \sim \textsc{Categorical}(\cdot|\omega_{t|T}^{k,1}, \cdots, \omega_{t|T}^{k,M})$
12. Set backward particle: $\tilde{\mathbf{z}}_t^k \leftarrow \tilde{\mathbf{z}}_t^{k,b_t^m}$, $\omega_{t|T}^k \leftarrow \omega_{t|T}^{k,b_t^k}$
13. Evaluate the backward proposal: $\Omega_t^k = M \cdot \omega_{t|T}^k \cdot q(\tilde{\mathbf{z}}_t^k|\tilde{\mathbf{z}}_{t+1}^k, \mathbf{x}_{1:T})$
14. **return**

$$\tilde{\mathbf{z}}_{1:T}^{1:K}, \hat{\mathcal{L}}_{SVO}(\mathbf{x}_{1:T}) \coloneqq \log\left(\frac{1}{K}\sum_{k=1}^K \frac{p(\tilde{\mathbf{z}}_{1:T}^k, \mathbf{x}_{1:T})}{\prod_{t=1}^T \Omega_t^k}\right)$$

to define subweights through a process of self normalized importance sampling

$$\begin{aligned}
&p(\tilde{\mathbf{z}}_t^{k,m}|\tilde{\mathbf{z}}_{t+1}^{k,m}, \mathbf{x}_{1:T})/q(\tilde{\mathbf{z}}_t^{k,m}|\tilde{\mathbf{z}}_{t+1}^{k}, \mathbf{x}_{1:T}) \\
&\propto \int p(\mathbf{z}_{t-1}, \tilde{\mathbf{z}}_t^{k,m}|\mathbf{x}_{1:t-1}) d\mathbf{z}_{t-1} \frac{f(\tilde{\mathbf{z}}_{t+1}^{k}|\tilde{\mathbf{z}}_t^{k,m}) g(\mathbf{x}_t|\tilde{\mathbf{z}}_t^{k,m})}{q(\tilde{\mathbf{z}}_t^{k,m}|\tilde{\mathbf{z}}_{t+1}^{k}, \mathbf{x}_{1:T})} \\
&\approx \left[\sum_{j=1}^{K} \bar{w}_{t-1}^{j} f(\tilde{\mathbf{z}}_t^{k,m}|\mathbf{z}_{t-1}^{j})\right] \frac{f(\tilde{\mathbf{z}}_{t+1}^{k}|\tilde{\mathbf{z}}_t^{k,m}) g(\mathbf{x}_t|\tilde{\mathbf{z}}_t^{k,m})}{q(\tilde{\mathbf{z}}_t^{k,m}|\tilde{\mathbf{z}}_{t+1}^{k}, \mathbf{x}_{1:T})} \\
&\coloneqq \omega_{t|T}^{k,m}.
\end{aligned} \tag{4.13}$$

A single particle is selected by sampling an index with probability proportional to the subweight $\omega_{t|T}$: $b_t^k \sim \textsc{Categorical}(b_t^k|\omega_{t|T}^{k,1}, \cdots, \omega_{t|T}^{k,M})$, $\tilde{\mathbf{z}}_t^k \leftarrow \tilde{\mathbf{z}}_t^{k,b_t^k}$. This modified particulate distribution now generates hidden states from a *continuous* domain given the future state and all observations. Repeating this process sequentially in the reverse-time direction produces K i.i.d. sample trajectories, $\{\tilde{\mathbf{z}}_{1:T}^{1:K}\}$ (see Algorithm 3).

The approximate posterior and variational objective are defined below via Algorithm 3. Note again that the following expectations are also conditioned on the forward filtering system.

$$\mathcal{L}_{SVO} \coloneqq \mathop{\mathbb{E}}_{q}\left[\log \hat{\mathcal{Z}}_{SVO}\right], \quad \hat{\mathcal{Z}}_{SVO} \coloneqq \frac{1}{K}\sum_{k=1}^{K} \frac{p(\tilde{\mathbf{z}}_{1:T}^{k}, \mathbf{x}_{1:T})}{q(\tilde{\mathbf{z}}_{1:T}^{k}|\mathbf{x}_{1:T})}, \tag{4.14}$$

where $q(\tilde{\mathbf{z}}_{1:T}^{k}|\mathbf{x}_{1:T})$ is defined below,

$$q(\tilde{\mathbf{z}}_{1:T}^{k}|\mathbf{x}_{1:T}) \coloneqq M^T \cdot \omega_{T|T}^{k} \cdot q(\tilde{\mathbf{z}}_T^{k}|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1}\left[\omega_{t|T}^{k} \cdot q(\tilde{\mathbf{z}}_t^{k}|\tilde{\mathbf{z}}_{t+1}^{k}, \mathbf{x}_{1:T})\right]. \tag{4.15}$$

We note that while the sequence of target distributions is filtered, our objective is constructed using samples from a smoothing posterior. This heuristic facilitates smoothing the target when performing VI to simultaneously train $p(\mathbf{Z}|\mathbf{X})$ and $q(\mathbf{Z}|\mathbf{X})$ by pulling $p(\mathbf{Z}|\mathbf{X}) \rightarrow q(\mathbf{Z}|\mathbf{X})$. This functional dependence motivates sharing the transition function between proposal and target.

4.5 Analysis of Unbiasedness

Theorem 2. *$\hat{Z}_{SVO}$ is an unbiased estimate of $p(\mathbf{x}_{1:T})$.*

$$\mathbb{E}_{Q(\hat{\mathbf{z}}_{1:T}^{1:K,1:M})}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{p(\hat{\mathbf{z}}_{1:T}^{k},\mathbf{x}_{1:T})}{\prod_{t=1}^{T}\Omega_t^K}\right] = p(\mathbf{x}_{1:T}),$$

where $Q(\hat{\mathbf{z}}_{1:T}^{1:K,1:M})$ denotes the sampling distribution of $\hat{z}_{1:T}^{1:M}$ according to Algorithm 1.

Proof. We will define auxiliary variables λ and distributions $q(\lambda|x)$, $q(z|\lambda,x)$, and $r(\lambda|z,x)$ such that

$$\hat{\mathcal{Z}}_{SVO} \equiv \hat{p}(x) = \frac{p(x,z)r(\lambda|z,x)}{q(z,\lambda|x)} = \frac{p(x,z)r(\lambda|z,x)}{q(z|\lambda,x)q(\lambda|x)},$$

where $z,\lambda \sim q(z,\lambda|x)$. For a treatment of auxiliary random variables see [21, 64]. Here the auxiliary latent variables are the unselected subparticles,

$$\lambda = \{\tilde{\mathbf{z}}_{1:T}^{\neg b_{1:T}^{1:K}}\}.$$

For convenience, we omit the conditioning on the forward system. To further simplify notation, we will rearrange particles to omit the backward ancestor indices by defining $\hat{\mathbf{z}}_t^{k,1} \leftarrow \tilde{\mathbf{z}}_t^{k,b_t^k}$, $\hat{\omega}_{t|T}^{k} \leftarrow \omega_{t|T}^{k,b_t^k}$ and $\hat{\mathbf{z}}_t^{k,2:M} \leftarrow \tilde{\mathbf{z}}_t^{k,\neg b_t^k}, \hat{\omega}_{t|T}^{k,2:M} \leftarrow \omega_{t|T}^{k,\neg b_t^k}$. By the linearity of expectation, it suffices to show the case of $K=1$ (as a result, for clarity, we will omit k, in the superscripts):

$$\mathbb{E}_{\hat{\mathbf{z}}_{1:T}^{1:M}}\left[\frac{p(\hat{\mathbf{z}}_{1:T}^{1},\mathbf{x}_{1:T})}{\prod_{t=1}^{T}\Omega_t}\right] = p(\mathbf{x}_{1:T})$$

We begin by expressing the generative distribution of the sampling process for the rearranged particles $\hat{\mathbf{z}}_{1:T}^{1:M}$ as factorizing:

$$Q(\hat{\mathbf{z}}_{1:T}^{1:M}|\mathbf{x}_{1:T}) = Q(\hat{\mathbf{z}}_{T}^{1:M}|\mathbf{x}_{1:T})\prod_{t=1}^{T-1}Q(\hat{\mathbf{z}}_{t}^{1:M}|\hat{\mathbf{z}}_{t+1}^{1},\mathbf{x}_{1:T}).$$

Consider the sampling process last time step,

1. Sample $\{\tilde{\mathbf{z}}_T^m\}_{m=1}^M \sim q(\cdot|\mathbf{x}_{1:T})$, and compute the associated weights $\tilde{\omega}_{T|T}^{1:M}$ as outlined in Algorithm 1

2. Sample $b_T \sim \text{CATEGORICAL}(\cdot|\tilde{\omega}_{T|T}^1, \cdots, \tilde{\omega}_{T|T}^M)$

3. Set $\hat{\mathbf{z}}_T^1 \leftarrow \tilde{\mathbf{z}}_T^{b_T}, \hat{\mathbf{z}}_T^{2:M} \leftarrow \tilde{\mathbf{z}}_T^{\neg b_T}$, and $\hat{\omega}_{T|T}^1 \leftarrow \tilde{\omega}_{T|T}^{b_T}, \hat{\omega}_{T|T}^{2:M} \leftarrow \tilde{\omega}_{T|T}^{\neg b_T}$

The marginal distribution of $\hat{\mathbf{z}}_T^{1:M}$ is obtained as follows:

$$
\begin{aligned}
&Q(\hat{\mathbf{z}}_T^{1:M}|\mathbf{x}_{1:T}) \\
&= \int \left(\prod_{m=1}^M \underbrace{q(\tilde{\mathbf{z}}_T^m|\mathbf{x}_{1:T})}_{\text{Step 1}} \right) \cdot \left(\sum_{b_T=1}^M \underbrace{p(b_T|\tilde{\mathbf{z}}_T^{1:M})}_{\text{Step 2}} \cdot \underbrace{p(\hat{\mathbf{z}}_T^{1:M}|\tilde{\mathbf{z}}_T^{1:M}, b_T)}_{\text{Step 3}} \right) d\tilde{\mathbf{z}}_T^{1:M} \\
&= \sum_{b_T=1}^M \int \left(\prod_{m=1}^M q(\tilde{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \right) \cdot \frac{\tilde{\omega}_{T|T}^{b_T}}{\tilde{\omega}_{T|T}^{b_T} + \sum_{i \in \neg b_T} \tilde{\omega}^i}_{T|T} \delta(\hat{\mathbf{z}}_T^1 - \tilde{\mathbf{z}}_T^{b_T}) \delta(\hat{\mathbf{z}}_T^{2:M} - \tilde{\mathbf{z}}_T^{\neg b_T}) d\tilde{\mathbf{z}}_T^{1:M} \\
&= M \int \left(\prod_{m=1}^M q(\tilde{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \right) \cdot \frac{\tilde{\omega}_{T|T}^1}{\tilde{\omega}_{T|T}^1 + \sum_{i=2:M} \tilde{\omega}_{T|T}^i} \delta(\hat{\mathbf{z}}_T^1 - \tilde{\mathbf{z}}_T^1) \delta(\hat{\mathbf{z}}_T^{2:M} - \tilde{\mathbf{z}}_T^{2:M}) d\tilde{\mathbf{z}}_T^{1:M} \\
&= M \left(\prod_{m=1}^M q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \right) \frac{\hat{\omega}_{T|T}^1}{\sum_{i=1}^M \hat{\omega}_{T|T}^i},
\end{aligned}
$$

where the third equality follows from collapsing all possible cases to $b_T = 1$ by symmetry, and the last equality follows from integrating over the dirac measures.

Similarly, we have the following for $t = 1, \ldots, T-1$,

$$
Q(\hat{\mathbf{z}}_t^{1:M}|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T}) = \left(\prod_{m=1}^M q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T}) \right) \cdot M \cdot \frac{\hat{\omega}_{t|T}^1}{\sum_{m=1}^M \hat{\omega}_{t|T}^m}.
$$

Therefore,

$$
Q(\hat{\mathbf{z}}_{1:T}^{1:M}|\mathbf{x}_{1:T}) = \underbrace{\left(\prod_{t=1}^T \Omega_t \right)}_{q(z|\lambda,x)} \cdot \underbrace{\prod_{m=2}^M \left(q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T}) \right)}_{q(\lambda|x)}.
$$

Now, define the target distribution to be:

$$
P(\hat{\mathbf{z}}_{1:T}^{1:M}, \mathbf{x}_{1:T}) = p(\hat{\mathbf{z}}_{1:T}^1, \mathbf{x}_{1:T}) r(\lambda|\mathbf{x}_{1:T}, \mathbf{z}_{1:T})
$$

where

$$r(\lambda|\mathbf{x}_{1:T}, \mathbf{z}_{1:T}^{1:M}) = q(\lambda|\mathbf{x}_{1:T}) \\ = \prod_{m=2}^{M} \left[q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T}) \right].$$

Overall,

$$\mathbb{E}_{Q(\hat{\mathbf{z}}_{1:T}^{1:M})}[\hat{Z}_{SVO}] = \mathbb{E}_{Q(\hat{\mathbf{z}}_{1:T}^{1:M})} \left[\frac{p(\hat{\mathbf{z}}_{1:T}^1, \mathbf{x}_{1:T})}{\prod_{t=1}^{T} \Omega_t} \right]$$

Writing the augmented target and proposal explicitly:

$$= \mathop{\mathbb{E}}_{Q} \left[\frac{p(\hat{\mathbf{z}}_{1:T}^1, \mathbf{x}_{1:T}) \times \prod_{m=2}^{M} q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T})}{\prod_{t=1}^{T} \Omega_t \times \prod_{m=2}^{M} q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T})} \right]$$

Combining the selected and unselected subparticles:

$$= \mathop{\mathbb{E}}_{Q} \left[\frac{P(\hat{\mathbf{z}}_{1:T}^{1:M}, \mathbf{x}_{1:T})}{Q(\hat{\mathbf{z}}_{1:T}^{1:M})} \right] \\ = \int P(\hat{\mathbf{z}}_{1:T}^{1:M}, \mathbf{x}_{1:T}) d\hat{\mathbf{z}}_{1:T}^{1:M}$$

Split the integral over selected and unselected subparticles to evaluate:

$$= \int p(\hat{\mathbf{z}}_{1:T}^1, \mathbf{x}_{1:T}) \times \left(\int \prod_{m=2}^{M} q(\hat{\mathbf{z}}_T^m|\mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q(\hat{\mathbf{z}}_t^m|\hat{\mathbf{z}}_{t+1}^1, \mathbf{x}_{1:T}) d\hat{\mathbf{z}}_{1:T}^{2:M} \right) d\hat{\mathbf{z}}_{1:T}^1 \\ = \int p(\hat{\mathbf{z}}_{1:T}^1, \mathbf{x}_{1:T}) d\hat{\mathbf{z}}_{1:T}^1 \\ = p(\mathbf{x}_{1:T}).$$

□

4.6 Implemention Details

In the forward filtering pass, we define the proposal distribution as follows:

$$q_{\phi,\varphi}(\mathbf{z}_{1:T}^k|\mathbf{x}_{1:T}) \propto \underbrace{f_\varphi(\mathbf{z}_1^k)}_{initial\ state} \prod_{t=1}^{T} \underbrace{h_\phi(\mathbf{z}_t^k|\mathbf{x}_t)}_{encoding} \prod_{t=2}^{T} \underbrace{\textsc{Categorical}(a_{t-1}^k|\bar{w}_{t-1}^{1:K})}_{resampling} \underbrace{f_\varphi(\mathbf{z}_t^k|\mathbf{z}_{t-1}^{a_{t-1}^k})}_{transition},$$

Proposition 4.5.1. *Assume that the first four moments of w_t^1 and ∇w_t^1 are all finite and their variances are non-zero for $t \in 1:T$, then the signal-to-noise ratio converges at the following rate:*

$$SNR_K(\theta,\varphi,\phi) = \left| \frac{\nabla \log Z + \sum_{t=2}^{T}\sum_{t'\geq t+1}^{T} \mathbb{E}\left[\nabla \frac{w_{t-1}^1}{Z_{t-1}} \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right] + \mathcal{O}(1/K)}{\sqrt{1/K \left\{ \sum_{t=1}^{T} \mathbb{E}\left[(\nabla \frac{w_t^1}{Z_t})^2\right] + \sum_{t'\neq t,t'=1}^{T}\sum_{t=1}^{T} \sqrt{\mathrm{Var}\left[\nabla \frac{w_t^1}{Z_t}\right] \mathrm{Var}\left[\nabla \frac{w_{t'}^1}{Z_{t'}}\right]} \right\} + \mathcal{O}(T^2/K^2)}} \right| \tag{4.16}$$

where $Z = p_\theta(\mathbf{x}_{1:T})$ and $Z_t = p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ for $t \in \{1,\cdots,T\}$. Further assuming the resampling bias $\sum_{t=2}^{T}\sum_{t'\geq t+1}^{T} \mathbb{E}\left[\nabla \frac{w_{t-1}^1}{Z_{t-1}} \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right] = \mathcal{O}(1)$ leads to $\mathrm{SNR}_K(\theta,\phi,\varphi) = \mathcal{O}(\sqrt{K})$.

Proof. See Section 4.9. □

where the proposal density factorizes into evolution and encoding functions,

$$f_\varphi(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\psi(\mathbf{z}_{t-1}),\Sigma), \quad h_\phi(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\gamma(\mathbf{x}_t),\Lambda). \tag{4.17}$$

We define $\psi : \mathbb{R}^{d_\mathbf{z}} \to \mathbb{R}^{d_\mathbf{z}}$ and $\gamma : \mathbb{R}^{d_\mathbf{x}} \to \mathbb{R}^{d_\mathbf{z}}$ as nonlinear time invariant functions represented with deep neural networks. The covariances Σ and Λ are taken as time invariant trainable parameters or nonlinear functions of the latent space. This proposal choice allows the transition term of the inference network $f_\varphi(\mathbf{z}_t|\mathbf{z}_{t-1})$ to share the parameters φ defining $\{\psi,\Sigma\}$ with the transition term $f_\varphi(\mathbf{z}_t|\mathbf{z}_{t-1})$ of the target defined in Eq. (4.1) [66, 77, 89]. The evolution term of the variational posterior is exact, retaining both tractability and expressiveness.

The transition and emission densities are specified as follows:

$$f_\varphi(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\psi(\mathbf{z}_{t-1}),\Sigma), \quad g_\theta(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\upsilon(\mathbf{z}_t),\Gamma)\ . \tag{4.18}$$

The decoding term is defined using a deterministic nonlinear rate function $v : \mathbb{R}^{d_{\mathbf{z}}} \to \mathbb{R}^{d_{\mathbf{x}}}$ represented with a deep network and a noise model that need not be conjugate. Without loss of generality we consider a Gaussian emission density. The backward proposal defining the smoothing distribution below

$$q(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \propto r(\mathbf{z}_t|\zeta(\mathbf{z}_{t+1}))e(\mathbf{z}_t|\chi(\mathbf{x}_{1:T})), \tag{4.19}$$

is specified using nonlinear time invariant functions $\zeta : \mathbb{R}^{d_{\mathbf{z}}} \to \mathbb{R}^{d_{\mathbf{z}}}$ and $\chi : \mathbb{R}^{d_{\mathbf{x}}} \to \mathbb{R}^{d_{\mathbf{z}}}$ which we take as deep networks.

4.7 SNR of Gradient Estimators in Filtering SMC

$\mathcal{L}_{\text{SMC}}$ is a consistent estimator of the log marginal likelihood under some mild conditions [77]. Intuition suggests increasing the number of particles K provides a better surrogate objective. However, [101] points out that the SNR of the inference network gradient estimator decreases to 0 as K increases in the IWAE setting. [66] extends the result to the filtering SMC without providing theoretical evidence. Here, we argue that the result does not generalize to SMC due to the resampling step. Formally, for a gradient estimator of $\mathcal{L}_{\text{SMC}}$ (denoted Δ_K), constructed by K particles, the SNR is defined as:

$$\text{SNR}_K = \left| \frac{\mathbb{E}[\Delta_K]}{\sqrt{\text{Var}[\Delta_K]}} \right|. \tag{4.20}$$

For the SNR of $\nabla\mathcal{L}_K$, we have the following Proposition 1. We add empirical evidence to this result in Section 6. We consider three types of stochastic gradient estimators. A full definition is given in the Appendix.

1. The biased estimator without resampling gradient, $\nabla\mathcal{L}_K$.

2. The unbiased estimator, $\nabla\mathcal{L}_K$ + CATEGORICAL.

3. The relaxed estimator, $\nabla \mathcal{L}_K + \textsc{Concrete}(\lambda)$ [46, 78].

4.8 Experimental Results

In order to quantify the performance of the trained dynamics, we compute the k-step mean squared error (MSE) and its normalized version, the R_k^2. To do so, the trained transition function is applied to the latent state without any input data over a rolling window of k steps into the future. The emission function is then used to obtain a prediction $\hat{\mathbf{x}}_{t+k}$ which we compare with the observation $\mathbf{x}_{t+k}$.

$$\text{MSE}_k = \sum_{t=1}^{T-k} \left(\mathbf{x}_{t+k} - \hat{\mathbf{x}}_{t+k}\right)^2 , \qquad R_k^2 = 1 - \frac{\text{MSE}_k}{\sum_{t=1}^{T-k} \left(\mathbf{x}_{t+k} - \bar{\mathbf{x}}_k\right)^2} , \tag{4.21}$$

where $\bar{\mathbf{x}}_k$ is the average of $\mathbf{x}_{k+1:T}$. We note that the ELBO is not a performance statistic that generalizes across models. In contrast, the R_k^2 provides a metric to quantify the inferred dynamics. This procedure is defined in [37]. In all experiments, SVO is only given access to the observation sequence, and not the equations that govern the nonlinear systems. The latent trajectories and dynamics, transition, emission and encoding functions are all inferred.

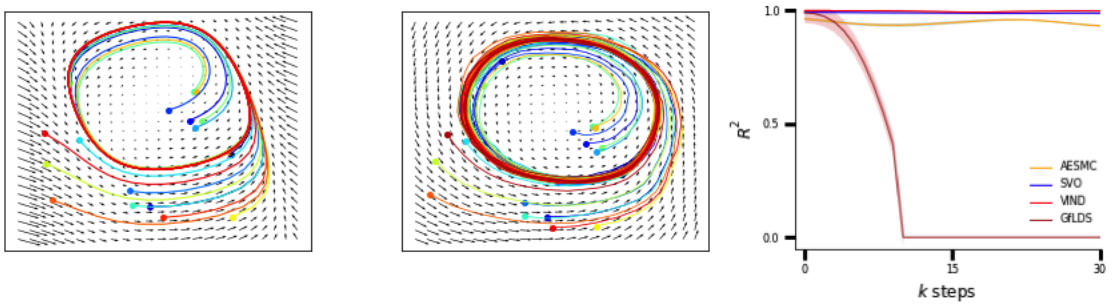


Figure 4.2: Summary of the Fitzhugh-Nagumo results: the observation is one-dimensonal while the phase space and latent variables are two-dimensional; (left) ground truth dynamics and trajectories for the original system; (center) latent dynamics and trajectories inferred by SVO; Initial points (denoted by markers) located both inside and outside the limit cycle are topologically invariant in the SVO reconstruction; (right) R_k^2 for various models on the dimensionality expansion task. Results are averaged over 3 random seeds.

4.8.1 Fitzhugh-Nagumo

The Fitzhugh-Nagumo (FN) system is a two dimensional simplification of the Hodgkin-Huxley model. The FN provides a geometric interpretation of the dynamics of spiking neurons and is described by two independent variables V_t and W_t with cubic and linear functions,

$$\begin{aligned} \dot{V} &= V - V^3/3 - W + I_{ext} \\ \dot{W} &= a(bV - cW). \end{aligned} \tag{4.22}$$

Eq. (4.22) was integrated over 200 time points with $I_{ext} = 1$ held constant and $a = 0.7, b = 0.8, c = 0.08$. The initial state was sampled uniformly over $[-3, 3]^2$ to generate 100 trials using 66 for training, 17 for validation and 17 for testing. We emphasize that dimensionality expansion is intrinsically harder than dimensionality reduction due to a loss of information. A one-dimensional Gaussian observation is defined on V_t with $\mathbf{x}_t = \mathcal{N}(V_t, 0.01)$. SVO is

used to recover the two dimensional phase space and latent trajectories $\mathbf{z}_t = (V_t, W_t)$ of the original system. This task requires using information from future observations to correctly infer the initial state. Fig. 4.2 shows the results of the FN experiment. The left panel displays the original system. The center panel displays the learned dynamics and inferred trajectories on the test set using SVO to perform dimensionality expansion. Initial points (denoted with markers) located both inside and outside of the limit cycle in the original system are topologically invariant in the reconstruction. The right panel shows the R_k^2 comparison across models. AESMC with $K = 1024$ gives an $R_{30}^2 = 0.954$ in contrast to SVO with $K = 32, M = 32$ which gives an $R_{30}^2 = 0.993$. SVO outperforms AESMC and GFLDS.

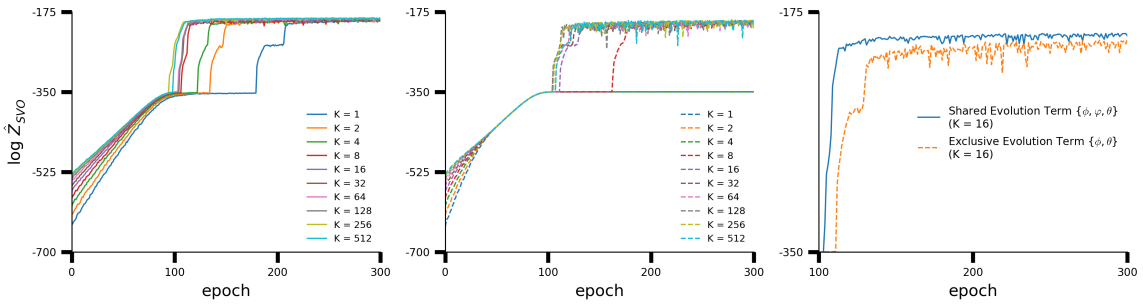


Figure 4.3: ELBO convergence across epochs for SVO using exclusive parameters θ, ϕ and shared parameters θ, φ, ϕ; (left) log $\hat{\mathcal{Z}}_{SVO}$ across epochs as K increases using shared evolution network; (center) log $\hat{\mathcal{Z}}_{SVO}$ across epochs as K increases using independent evolution networks; (right) log $\hat{\mathcal{Z}}_{SVO}$ convergence for shared vs independent evolution networks with $K = 16$ highlighting faster convergence to a higher ELBO.

4.8.2 Sharing Transition Terms

We study the effect of sharing the transition function between the proposal and target distribution. Fig. 4.3 illustrates the ELBO convergence as the number of particles K is increased. The left panel plots ELBO for SVO with network parameters shared between proposal and target. Increasing K produces a faster convergence and lower stochastic gradient noise. The center panel illustrates separate evolution networks for the proposal and the target. In contrast to sharing the transition function, separate evolution networks require a larger number of epochs for corresponding value of K. The ELBO obtains a lower value with larger stochastic gradient noise. The right panel juxtaposes shared and separate transition functions for $K = 16$ particles.

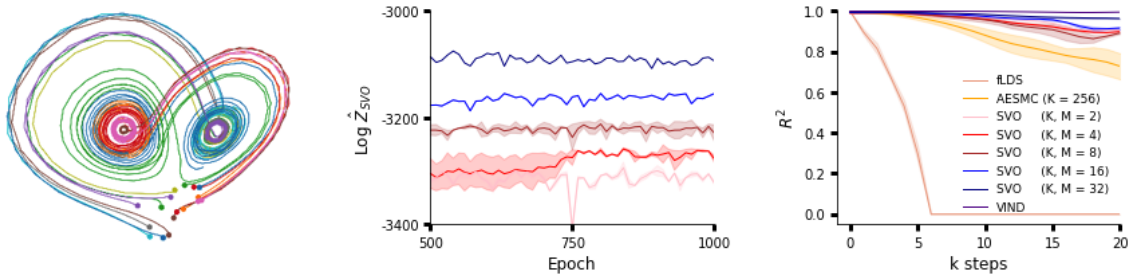


Figure 4.4: Summary of the Lorenz results: (left) latent trajectories inferred from nonlinear 10D observations; (center) $\log \hat{Z}_{SVO}$ as K, M increase (legend on the right). Larger K, M produce higher ELBO values; (right) R_k^2 on the dimensionality reduction task illustrating near-perfect reconstruction at 20 steps ahead on the validation set. Results averaged over 3 random seeds.

4.8.3 Lorenz Attractor

The Lorenz attractor is a chaotic nonlinear dynamical system defined by 3 independent variables,

$$\begin{aligned} \dot{z}_1 &= \sigma(z_2 - z_1)\,, \\ \dot{z}_2 &= z_1(\rho - z_3) - z_2\,, \\ \dot{z}_3 &= z_1 z_2 - \beta z_3\,. \end{aligned} \tag{4.23}$$

Eq. (4.23) is integrated over 250 time points with $\sigma = 10, \rho = 28, \beta = 8/3$ by generating randomized initial states in $[-10, 10]^3$. A $\mathbf{z}$-dependent neural network is used to produce ten dimensional nonlinear Gaussian observations with 100 trials, 66 for training, 17 for validation and 17 for testing. Fig. 4.4 provides the results of the Lorenz experiment. The left panel provides the inferred latent paths illustrating the attractor. The center plot provides $\log \hat{Z}_{SVO}$ as K, M increase (legend on the right). Larger K, M produce higher ELBO values. The right panel displays the R_k^2 comparison with $d_{\mathbf{z}} = 3$. Results are averaged

over 3 random seeds. Increasing K, M produces R_k^2 improvements. SVO with $K, M = 2$ gives a higher R_k^2 than both GFLDS and AESMC using $K = 256$.

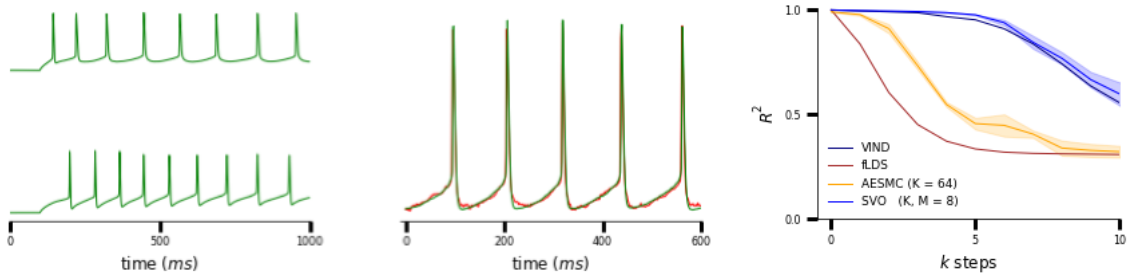


Figure 4.5: Summary of the Allen results: (left) two trials from the dataset; (center) the data against the predicted observation value using the dynamics learned over a rolling window ten steps ahead on the validation set. Hyperpolarization and depolarization nonlinearities are predicted by the inferred dynamics; (right) R_k^2 with $K, M = 8$ particles. SVO outperforms GfLDS and AESMC with $K = 64$. Results are averaged across 3 random seeds.

4.8.4 Electrophysiology Data

Neuronal electrophysiology data was downloaded from the Allen Brain Atlas [49]. Intracellular voltage recordings from primary Visual Cortex of mouse, area layer 4 were collected. A step-function input current with an amplitude between 80 and 151pA was applied to each cell. A total of 40 trials from 5 different cells were split into 30 trials for training and 10 for validation. Each trial was divided into five parts and down-sampled from 10,000 time bins to 1,000 time bins in equal intervals. Each trial was normalized by its maximal value. Fig. 4.5 summarizes the Allen experiment. The left panel provides two trials of the 1D observations from the training set. The center panel illustrates the predicted observation using the dynamics learned over a rolling window ten steps ahead on the validation set. SVO captures hyperpolarization and depolarization nonlinearities when appying the inferred dynamics. The right panel displays the R_k^2 comparison with $d_{\mathbf{z}} = 3$. SVO outperforms AESMC and GfLDS.

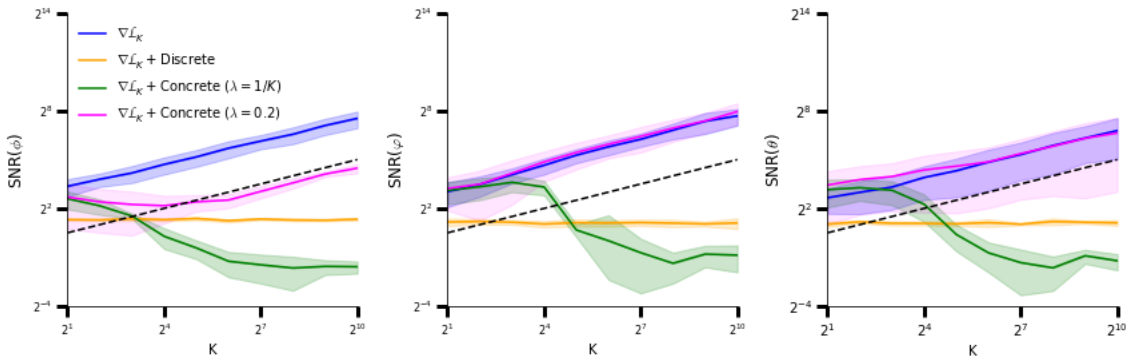


Figure 4.6: Convergence of SNRs of gradient estimators in the encoder network (left), transition network (center) and decoder network (right) with increasing K. Distinct solid lines correspond to empirical SNRs of the four gradient estimators, averaging over 6 random seeds. The black dashed line with slope 1 illustrates a signal-to-noise-ratio of convergence rate $\mathcal{O}(\sqrt{K})$.

4.8.5 SNR Gradient Estimators

We report the l_2 norm of empirical SNRs for the encoder network (ϕ), evolution network (φ) and decoder network (θ), where the gradient is taken with respect to ϕ, φ and θ correspondingly. Fig. 4.6 presents four gradient estimators where the expectation and variance are calculated using $N = 100$ gradient samples collected in the middle training stage of running filtering SMC on Fitzhugh-Nagumo data. The gradient estimator that ignores the resampling step possesses an SNR of convergence rate $\mathcal{O}(\sqrt{K})$, which aligns with the theoretical result. Similarly this holds for the relaxed Concrete gradient estimator with a constant temperature ($\lambda = 0.2$). The unbiased Categorical resampling gradient and the relaxed Concrete gradient with decreasing temperature ($\lambda = K^{-1}$) suffer from large variance, leading to a relatively low and even vanishing SNR for increasing K. Moreover, the level of relaxation λ in the Concrete gradient estimator leads to different behaviors of SNR. These observations imply that introducing bias reduces the variance and mitigates

the degradation of the SNR with increasing K.

4.9 Discussion

AESMC [66], FIVO [77] and VSMC [89] are three closely related methods that form a lower bound to the log marginal likelihood which is estimated using filtering SMC, however without conditioning the latent state on future observations they may fail to capture long-term dependencies. VSMC draws a single sample at the final time step to produce a trajectory from the corresponding ancestral path. While this heuristic produces *one* sample conditioned on all observations, the resulting path is not used to construct the surrogate ELBO which is filtered.

Particle smoothing methods include the previously discussed FFBSI [31] and the Two Filter Smoother (TFS) [98]. The FFBSI defines a posterior over an entire trajectory and gives a way to sample the trajectory backward in time. In contrast, TFS defines a posterior only at a single time step. Additionally they differ in their methods. For TFS, the backward filtering is independent of the forward filtering. However, our backward simulation is conditional on forward filtering, where the subweight depends on the forward system. Unlike standard particle smoothing methods, SVO is a framework for performing VI on state-space models, jointly for the states and the model itself, analogous to the approach of the variational auto-encoder [57]. The proposal and the target distribution are trained from the observation sequence.

Particle smoothing methods incur a cost that is quadratic in the number of particles due to the pairwise interactions in the summation of Eq (13). For SVO, smoothing incurs a cost of $\mathcal{O}(TK^2Md_z)$ operations in contrast to $\mathcal{O}(TGd_z)$ in AESMC (where G denotes the number of particles). For a fair comparison we give AESMC the corresponding extra

particles. Empirically, SVO with small K and M (4 or 8) can provide a more accurate posterior approximation than AESMC with a much larger value of G. For the FHN task, SVO with $K = M = 32$ outperforms AESMC with $G = 1024$ (see Fig 1); For the Lorenz task SVO with $K = M = 2$ also outperforms AESMC with $G = 256$ (see Fig 3). SVO also works with larger T, K, and M (with $T = 1000$ on the Allen data). All the experiments were run on 16 core CPU machines. Despite the summation in Eq. 4.13 being $\mathcal{O}(TK^2M)$, the main cost is evaluating the neural network $\psi(\cdot)$ and its gradients for $f(\cdot|\mathbf{z}_{t-1}^j) = \mathcal{N}(\cdot|\psi(\mathbf{z}_{t-1}^j), \Sigma)$. The computation is $\mathcal{O}(TK)$ here and $\mathcal{O}(TKM)$ in the emission term. The $\mathcal{O}(TK^2M)$ is fast relative to the evaluation of the neural network.

Two variational smoothing methods for inference in non-conjugate SSMs are GFLDS [27, 3] and VIND [37]. These methods simultaneously train recognition and generative models using AEVB analogous to proposal and target distributions in SVO. GFLDS is a generative model and approximation for linear latent dynamics together with nonlinear emission densities. Building upon this, VIND is governed by nonlinear latent dynamics and emissions. GFLDS and VIND both require inverting a block-tridiagonal matrix which mixes components of state space through the inverse covariance. This incurs a complexity of $\mathcal{O}(Td_z^3)$ where T is the length of the time series and d_z is the state dimension. An alternative approach is to directly modify the target distribution in SMC to achieve smoothing [36]. TVSMC [65] and SMC-Twist [72] augment the intermediate target distribution with a twisting function, which in turn is approximated with deterministic algorithms such as temporal difference learning and Laplace approximation. When applied to nonlinear time series it was reported that TVSMC underperforms relative to filtering using VSMC [65].

We have introduced SVO, a framework for performing VI on state-space models jointly for hidden state inference and model parameter learning. SVO produces an unbiased esti-

mate of the marginal likelihood constructed using a sample from the smoothed, and not only filtered, state sequence. We have defined a novel backward simulation algorithm and approximate posterior obtained by sub-samping auxiliary random variables. This augments the support of the proposal and boosts particle diversity. Unlike standard particle smoothing methods, SVO simultaneously trains both the proposal and the target distribution from the observation sequence. SVO provides an estimate of nonlinear transition and emission functions in addition to latent states. Highlights include the ability to produce accurate long-range forecasts given smooth initial conditions from noisy, nonlinear differential equations using the trained latent dynamics. SVO consistently outperforms filtered objectives on all three experiments given fewer Monte Carlo samples. SVO is written in TensorFlow. An implementation is publicly available online.

Proof of Proposition 4.5.1

Proof. It suffices to show the convergence rate of expectation and variance of gradient estimate with respect to K. Throughout the analysis, we will extensively apply the result from [101], and exploit the factorization of the filtering SMC objective: $\hat{Z} \coloneqq \hat{Z}_{\text{SMC}} = \prod_{t=1}^{T} \hat{Z}_t$ where $\hat{Z}_t = \frac{1}{K} \sum_{k=1}^{K} w_t^k$. Assume that $\mathbf{z}_{1:T}^{1:K}$ are obtained by passing the Guassian noise $\epsilon_{1:T}^{1:K}$ through the reparameterization function.

1. Expectation.

$$\mathbb{E}\left[\nabla \log \hat{Z}\right] = \nabla \mathbb{E}\left[\log \hat{Z}\right] - \mathbb{E}\Big[\nabla \log \prod_{t=2}^{T} \prod_{k=1}^{K} \textsc{Categorical}(a_{t-1}^k | w_{t-1}^{1:K}) \cdot \log \hat{Z}\Big] \tag{4.24}$$

The expectation decomposes into two terms, where the convergence rate for the first

directly follows the result from [101]:

$$\nabla \mathbb{E}\left[\log \hat{Z}\right] = \nabla \sum_{t=1}^{T} \mathbb{E}\left[\log \hat{Z}_t\right] \tag{4.25}$$

$$= \nabla \log Z - \frac{1}{2K}\Big[\sum_{t=1}^{T} \nabla\left(\frac{\mathrm{Var}[w_t^1]}{Z_t^2}\right)\Big] + \mathcal{O}\left(\frac{T}{K^2}\right) \tag{4.26}$$

For the remaining term that includes the resampling gradient, we apply a thorough analysis as follows.

$$\mathbb{E}\Big[\nabla \log \prod_{t=2}^{T}\prod_{k=1}^{K} \textsc{Categorical}(a_{t-1}^k | w_{t-1}^{1:K}) \cdot \log \hat{Z}\Big]$$

$$= \sum_{t=2}^{T}\sum_{k=1}^{K} \mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^k | w_{t-1}^{1:K}) \log \hat{Z}\right] \tag{4.27}$$

$$= K \sum_{t=2}^{T}\sum_{t'=1}^{T} \mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \log \hat{Z}_{t'}\right] \tag{4.28}$$

Taylor expand $\log\ \hat{Z}_{t'}$ about $Z_{t'}$:

$$= K \sum_{t=2}^{T}\sum_{t'=2}^{T} \mathbb{E}\Bigg\{\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K})$$

$$\cdot \Big(\log Z_{t'} + \frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}} - \frac{(\hat{Z}_{t'} - Z_{t'})^2}{2Z_{t'}^2} + R_3(\hat{Z}_{t'})\Big)\Bigg\} \tag{4.29}$$

where $R_3(\hat{Z}_{t'})$ denotes the remainder in the Taylor expansion of $\log \hat{Z}_{t'}$ about $Z_{t'}$.

For $t' \leq t-1$, we have:

$$\mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(\hat{Z}_{t'} - Z_{t'})}{Z_{t'}}\right]$$

$$= \mathbb{E}_{\epsilon_{1:t-1}^{1:K}, a_{1:t-2}^{1:K}}\left\{\frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}} \times \mathbb{E}_{a_{t-1}^1}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K})\right]\right\} \tag{4.30}$$

$$= \mathbb{E}_{\epsilon_{1:t-1}^{1:K}, a_{1:t-2}^{1:K}}\left[\frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}} \cdot 0\right] = 0.$$

For $t' \geq t$, we have:

$$\begin{aligned}
&\mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(\hat{Z}_{t'} - Z_{t'})}{Z_{t'}}\right] \\
&= \mathbb{E}_{\epsilon_{1:t-1}^{1:K}, a_{1:t-1}^{1:K}}\left\{\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \times \mathbb{E}_{\epsilon_{t:t'}^{1:K}, a_{t:t'-1}^{1:K}}\left[\frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}}\right]\right\} \\
&= \mathbb{E}_{\epsilon_{1:t-1}^{1:K}, a_{1:t-1}^{1:K}}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot 0\right] \\
&= 0
\end{aligned} \tag{4.31}$$

Hence, it suffices to compute the convergence rate of the following:

$$K \sum_{t=2}^{T} \sum_{t'=2}^{T} \mathbb{E}\left\{\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(\hat{Z}_{t'} - Z_{t'})^2}{2Z_{t'}^2}\right\}$$

Note that when $t' \leq t-1$, we obtain similar results as Eq. (4.30). Thus, we turn to the case when $t' \geq t$. For $t' \geq t+1$, each $w_{t'}^k$ has dependence on a_{t-1}^1, hence:

$$\begin{aligned}
&K \cdot \mathbb{E}\left[\nabla \log \textsc{Categorial}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(\hat{Z}_{t'} - Z_{t'})^2}{2Z_{t'}^2}\right] \\
&= K \cdot \mathbb{E}\left\{\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \times \frac{\left(1/K \sum_{k=1}^{K} (w_{t'}^k - Z_{t'})\right)^2}{2Z_{t'}^2}\right\} \\
&= \mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2}\right]
\end{aligned} \tag{4.32}$$

Applying the score function derivative trick to the distribution of a_{t-1}^1:

$$\begin{aligned}
&= \sum_{i=1}^{K} \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla \frac{w_{t-1}^1}{K \hat{Z}_{t-1}} \cdot \frac{(w_t^1 - Z_t)^2}{2Z_t^2} \middle| \left[a_{t-1}^1 = i\right]\right]\right\} \\
&= K \cdot \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla \frac{w_{t-1}^1}{K \hat{Z}_{t-1}} \cdot \frac{(w_t^1 - Z_t)^2}{2Z_t^2} \middle| \left[a_{t-1}^1 = 1\right]\right]\right\}
\end{aligned}$$

Applying the Taylor expansion of $\frac{1}{\hat{Z}_{t-1}}$ around Z_{t-1}: $\frac{1}{\hat{Z}_{t-1}} = \frac{1}{Z_{t-1}} + R_2(\hat{Z}_{t-1})$:

$$\begin{aligned}
&= \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla \frac{w_{t-1}^1}{Z_{t-1}} \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right]\right\} \\
&+ \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla (w_{t-1}^1 R_2(\hat{Z}_{t-1})) \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right]\right\}
\end{aligned} \tag{4.33}$$

For $t' = t$, only w_t^1 depends on a_{t-1}^1, only one term that conditions on $a_{t-1}^1 = 1$ in (4.33) is not zero. Consequently we have:

$$\begin{aligned}
&K \cdot \mathbb{E}\left[\nabla \log \textsc{Categorical}(a_{t-1}^1 | w_{t-1}^{1:K}) \cdot \frac{(\hat{Z}_{t'} - Z_{t'})^2}{2Z_{t'}^2}\right] \\
&= \frac{1}{K} \cdot \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla \frac{w_{t-1}^1}{Z_{t-1}} \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right]\right\} \\
&\quad + \frac{1}{K} \cdot \mathbb{E}_{\epsilon_{1:t-1}^{1:K} a_{1:t-2}^{1:K}}\left\{\mathbb{E}_{\epsilon_t^1}\left[\nabla (w_{t-1}^1 R_2(\hat{Z}_{t-1})) \cdot \frac{(w_{t'}^1 - Z_{t'})^2}{2Z_{t'}^2} \middle| \left[a_{t-1}^1 = 1\right]\right]\right\} \qquad (4.34)
\end{aligned}$$

2. Variance.

$$\begin{aligned}
&\mathrm{Var}\left[\nabla \log \hat{Z}\right] = \mathrm{Var}\Big[\sum_{t=1}^{T} \nabla \log \hat{Z}_t\Big] \\
&= \sum_{t=1}^{T} \mathrm{Var}\left[\nabla \log \hat{Z}_t\right] + 2\sum_{t=1}^{T} \sum_{t' \neq t, t'=1}^{T} \mathrm{Cov}\left(\nabla \log \hat{Z}_t, \nabla \log \hat{Z}_{t'}\right) \qquad (4.35)
\end{aligned}$$

Decomposing the variance into the sum of variance at each time points, and the pairwise covariance across different time point, we will show that both terms are $\mathcal{O}(1/K)$.

(a) Variance at each time step. $\forall t = 1 : T$,

$$\begin{aligned}
\mathrm{Var}\left[\nabla \log \hat{Z}_t\right] &= \frac{1}{K} \cdot \mathbb{E}\left[\left(\frac{Z_t \nabla w_t^1 - w_t^1 \nabla Z_t}{Z_t^2}\right)^2\right] + \mathcal{O}\left(\frac{1}{K^2}\right) \qquad (4.36) \\
&= \frac{1}{K} \cdot \mathbb{E}\left[\left(\frac{\nabla w_t^1}{Z_t}\right)^2\right] + \mathcal{O}\left(\frac{1}{K^2}\right) \qquad (4.37)
\end{aligned}$$

(b) Covariance between different time steps.

For $t \neq t' \in 1 : T$, we first apply Taylor theorem to $\log \hat{Z}_t$ around Z_t, and then exploit the fact that $\hat{Z}_t$ is an unbiased estimation of Z_t, and exploit the definition

of covariance to expand and collapse terms, as follows:

$$
\begin{aligned}
&\operatorname{Cov}\left(\nabla \log \hat{Z}_t, \nabla \log \hat{Z}_{t'}\right) \\
&\quad= \operatorname{Cov}\left(\nabla\left(\log Z_t + \frac{\hat{Z}_t - Z_t}{Z_t} + R_1(\hat{Z}_t)\right), \nabla\left(\log Z_{t'} + \frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}} + R_1(\hat{Z}_{t'})\right)\right) \\
&\quad= \operatorname{Cov}\left(\nabla\left(\frac{\hat{Z}_t - Z_t}{Z_t} + R_1(\hat{Z}_t)\right), \nabla\left(\frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}} + R_1(\hat{Z}_{t'})\right)\right) \qquad (4.38) \\
&\quad= \mathbb{E}\left[\nabla\left(\frac{\hat{Z}_t}{Z_t}\right) \cdot \nabla\left(\frac{\hat{Z}_{t'}}{Z_{t'}}\right)\right] + \mathbb{E}\left[\nabla\left(\frac{\hat{Z}_{t'}}{Z_{t'}}\right) \nabla R_1(Z_t)\right] \\
&\qquad+ \mathbb{E}\left[\nabla\left(\frac{\hat{Z}_t}{Z_t}\right) \cdot \nabla R_1(Z_{t'})\right] + \operatorname{Cov}\left(\nabla R_1(\hat{Z}_t), \nabla R_1(\hat{Z}_{t'})\right) \qquad (4.39)
\end{aligned}
$$

i. For the first term in Eq. (4.39), since $\mathbf{z}_t^k$ are i.i.d. for fixed t, we have:

$$
\begin{aligned}
\mathbb{E}\left[\nabla\left(\frac{\hat{Z}_t}{Z_t}\right) \cdot \nabla\left(\frac{\hat{Z}_{t'}}{Z_{t'}}\right)\right] &= \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \nabla\left(\frac{w_t^k}{Z_t}\right) \cdot \frac{1}{K}\sum_{k'=1}^{K} \nabla\left(\frac{w_{t'}^{k'}}{Z_{t'}}\right)\right] \qquad (4.40) \\
&= \frac{1}{K^2} \cdot \sum_{k=1}^{K}\sum_{k'=1}^{K} \mathbb{E}\left[\nabla\left(\frac{w_t^k}{Z_t}\right) \cdot \nabla\left(\frac{w_{t'}^{k'}}{Z_{t'}}\right)\right] \\
&= \mathbb{E}\left[\nabla \frac{w_t^1}{Z_t} \cdot \nabla \frac{w_{t'}^1}{Z_{t'}}\right] \qquad (4.41) \\
&= \operatorname{Cov}\left(\nabla \frac{w_t^1}{Z_t}, \nabla \frac{w_{t'}^1}{Z_{t'}}\right) \qquad (4.42)
\end{aligned}
$$

Without loss of generality, we assume $t' > t$. First, when $t' = t + 1$,

$$
\Pr\left(\mathbf{z}_{t+1}^1 \text{ depends on } \mathbf{z}_t^1\right) = \mathbb{E}\left[\frac{w_t^1}{\sum_{k=1}^{K} w_t^k}\right] = \frac{1}{K} \qquad (4.43)
$$

When $t' > t + 1$, using chain rule and by induction we also have,

$$
\Pr(\mathbf{z}_{t'}^1 \text{ depends on } \mathbf{z}_t^1) = \frac{1}{K} \qquad (4.44)
$$

Hence,

$$\begin{aligned}
&\mathrm{Cov}\left(\nabla\frac{w_t^1}{Z_t}, \nabla\frac{w_{t'}^1}{Z_{t'}}\right) && (4.45)\\
&= \frac{1}{K}\cdot\mathrm{Cov}\left(\nabla\frac{w_t^1}{Z_t}, \nabla\frac{w_{t'}^1}{Z_{t'}}\middle|\left[z_{t'}^1 \text{ depends on } z_t^1\right]\right)\\
&\leq \frac{1}{K}\sqrt{\mathrm{Var}\left[\nabla\frac{w_t^1}{Z_t}\right]\mathrm{Var}\left[\nabla\frac{w_{t'}^1}{Z_{t'}}\right]} && (4.46)
\end{aligned}$$

ii. For the second and third term in Eq. (4.39), without loss of generality, we analyze the second term $\mathbb{E}\left[\nabla\left(\hat{Z}_{t'}/Z_{t'}\right)\cdot\nabla R_1(Z_t)\right]$, and assume $t' > t$.

Using the i.i.d. property of particles at fixed time step, we have:

$$\begin{aligned}
&\mathbb{E}\left[\nabla\left(\frac{\hat{Z}_{t'}}{Z_{t'}}\right)\cdot\nabla R_1(Z_t)\right] && (4.47)\\
&= \frac{1}{K^3}\cdot\mathbb{E}\left[\sum_{k=1}^{K}\nabla\frac{w_{t'}^k}{Z_{t'}}\mathcal{O}\left(\sum_{k=1}^{K}(w_t^k - Z_t)^2\right)\right]\\
&= \frac{1}{K}\ \cdot\ \mathbb{E}\left[\nabla\frac{w_{t'}^1}{Z_{t'}}\ \mathcal{O}\left((w_t^1 - Z_t)^2\right)\right] && (4.48)
\end{aligned}$$

Similar to the previous analysis on covariance, we can show that

$$\mathbb{E}\left[\nabla\frac{w_{t'}^1}{Z_{t'}}\cdot\mathcal{O}\left((w_t^1 - Z_t)^2\right)\right] = \mathcal{O}\left(\frac{1}{K}\right) \tag{4.49}$$

Hence,

$$\mathbb{E}\left[\nabla\left(\frac{\hat{Z}_{t'} - Z_{t'}}{Z_{t'}}\right)\cdot\nabla R_1(Z_t)\right] = \mathcal{O}\left(\frac{1}{K^2}\right) \tag{4.50}$$

iii. For the last term in Eq. (4.39), note that $|\mathrm{Cov}(A,B)| \leq \sqrt{\mathrm{Var}(A)\mathrm{Var}(B)}$, and $\mathrm{Var}[\nabla R_1(\hat{Z}_t)] = \mathcal{O}\left(1/K^2\right)$, hence we obtain:

$$\mathrm{Cov}\left(\nabla R_1(\hat{Z}_t), \nabla R_1(\hat{Z}_{t'})\right) = \mathcal{O}\left(\frac{1}{K^2}\right) \tag{4.51}$$

Substituting Eq. (4.37), Eq. (4.39) and Eq. (4.42) into Eq. (4.35), we arrive at the final expression for the variance of gradient estimate:

$$\begin{aligned}&\mathrm{Var}\left[\nabla \log \hat{Z}\right] \\ &= \frac{1}{K}\left\{\sum_{t=1}^{T}\mathbb{E}\left[\left(\nabla \frac{w_t^1}{Z_t}\right)^2\right] + \sum_{t=1}^{T}\sum_{t'\neq t,t'=1}^{T}\sqrt{\mathrm{Var}\left[\nabla \frac{w_t^1}{Z_t}\right]\mathrm{Var}\left[\nabla \frac{w_{t'}^1}{Z_{t'}}\right]}\right\} + \mathcal{O}\left(\frac{T^2}{K^2}\right)\end{aligned} \tag{4.52}$$

□

Chapter 5

Variational Combinatorial Sequential Monte Carlo for Bayesian Phylogenetic Inference

Bayesian phylogenetic inference is often conducted via local or sequential search algorithms such as random-walk Markov chain Monte Carlo or Combinatorial Sequential Monte Carlo. These methods perform inference by sampling tree topologies and branch lengths, however when used to perform optimization or evolutionary parameter learning, MCMC often requires long runs with inefficient state space exploration. Here we introduce Variational Combinatorial Sequential Monte Carlo (VCSMC), a novel Variational Inference method that simultaneously performs both parameter inference and model learning. VCSMC uses sequential search to construct a variational objective defined on the composite space of phylogenetic trees. We show that VCSMC is computationally efficient and explores higher probability spaces when compared with state-of-the-art Hamiltonian Monte Carlo methods.

This work, which was published as [84] was done jointly with Liyi Zhang and Itsik Pe'er. An implementation can be found online at `https://github.com/amoretti86/phylo`.

5.1 Introduction

Bayesian phylogenetic inference plays a central role in molecular evolutionary biology due to its ability to represent evolutionary uncertainty and incorporate prior information. Inference often involves three distinct tasks: (i) sampling from a discrete distribution to approximate an intractable summation over tree topologies, (ii) for each tree, integrating over the continuous parameters and branch lengths that govern the evolutionary model of interest, and (iii) performing parameter estimation or model learning. The sampling of tree topologies and branch lengths is typically accomplished via local search algorithms such as random-walk Markov chain Monte Carlo [42] or sequential search algorithms such as Combinatorial Sequential Monte Carlo [6]. Sophisticated proposal methods based on Hamiltonian Monte Carlo or particle MCMC have been suggested to sample from composite spaces and infer evolutionary parameters [20, 23, 118], however these methods are often difficult to implement, slow to converge and heavily dependent upon heuristics.

Variational Inference (VI) is a computationally efficient alternative to MCMC that simultaneously performs both inference and model learning. VI posits an approximate distribution and then recovers parameters of both the model and approximation by maximizing a lower bound to the log marginal likelihood. One approach to learning variational distributions on phylogenetic trees is to parameterize a tree as a sequence of *subsplits*, or ordered partitions on clades [128] and to recast the problem as a Bayesian network. One drawback of this setup is that the support of the conditional probability tables scales exponentially with the number of taxa [127]. A body of recent work has established connections between VI and sequential search by defining a variational family of distributions on hidden Markov models, where Sequential Monte Carlo is used as the marginal likelihood estima-

tor [66, 89, 85, 86]. Here we introduce Variational Combinatorial Sequential Monte Carlo (VCSMC), a novel variational objective and structured approximate posterior defined on the composite space of phylogenetic trees. Unlike standard variational SMC methods, our objective is constructed from *partial* states where the likelihood is not directly available and where states are formed by sampling from a large combinatorial set. VCSMC provides suitable estimates of the posterior when applied to a benchmark dataset of primate mitochondrial DNA and performs favorably when compared with the state of the art HMC methods.

5.2 Background

Phylogenetic Trees We wish to infer a latent bifurcating tree that describes the evolutionary relationships among a set of observed molecular sequences. A phylogeny is defined by a tree topology τ and a set of branch lengths $\mathcal{B}$. A *tree topology* is defined as a connected acyclic graph (V, E) where V is a set of vertices and E is a set of edges. *Leaf nodes* denote vertices of degree 1 and correspond to observed taxa. *Internal nodes* designate vertices of degree 3 (one parent and two children) and represent unobserved taxa (e.g. DNA bases of ancestral species). A special vertex called the *root node* of degree 2 (two children) represents the common evolutionary ancestor of all taxa.

For each edge $e \in E$, we associate a *branch length*, denoted $b(e) \in \mathbb{R}_{>0}$, $b(e) \in \mathcal{B}$. The branch length captures the intensity of the evolutionary changes between two vertices. An *ultrametric tree* is one with constant evolutionary rate along all paths from v to its descendants. More formally we define an ultrametric tree as one which satisfies the following: for all $v \in V$ and descendants of v denoted x, x', we have that $b(v, x) = b(v, x')$. *Nonclock trees* are general trees that do not require ultrametric assumptions. In this work we focus on

phylogenetic inference methods for nonclock trees as these are most pertinent to biologists.

Bayesian Phylogenetic Inference Let $\mathbf{Y} = \{Y_1, \cdots, Y_M\} \in \Omega^{NxM}$ denote the observed molecular sequences with characters in Ω of length M over N species. Bayesian inference requires specifying the prior density and likelihood function over tree topology τ, branch length set $\mathcal{B}$ and generative model parameters θ to write the joint posterior,

$$P(\mathcal{B}, \tau, \theta|\mathbf{Y}) = \frac{P(\mathbf{Y}|\tau, \mathcal{B}, \theta)P(\tau, \mathcal{B}|\theta)P(\theta)}{P(\mathbf{Y})}. \tag{5.1}$$

The prior is uniform over topologies and a product of independent exponential distributions over branch lengths with rate λ_{bl}. The evolution of each site is modeled independently using a continuous time Markov chain with rate matrix $\mathbf{Q}$. Let $\zeta_{v,m}$ denote the state of genome for species v at site m and define the evolutionary model along branch $b(v \to v')$:

$$P(\zeta_{v',s} = j|\zeta_{v,s} = i) = \exp\left(b(e)\mathbf{Q}_{i,j}\right). \tag{5.2}$$

The likelihood of a given phylogeny $P(\mathbf{Y}|\tau, \mathcal{B}, \theta) = \prod_{i=1}^{M} P(Y_i|\tau, \mathcal{B}, \theta)$ can be evaluated in linear time using the sum-product or Felsenstein's pruning algorithm [25] via the formula:

$$P(\mathbf{Y}|\tau, \mathcal{B}, \theta) \coloneqq \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_\rho) \prod_{(u,v)\in E(\tau)} \exp\left(-b_{u,v}\mathbf{Q}_{a^i_u, a^i_v}\right), \tag{5.3}$$

where ρ is the root node, a^i_u is the assigned character of node u, $E(\tau)$ represents the set of edges in τ and η is the prior or stationary distribution of the Markov chain. The normalization constant $P(\mathbf{Y})$ requires marginalizing the $(2N-3)!!$ distinct topologies [112] which is intractable.

Combinatorial Sequential Monte Carlo CSMC is a method to sample from a probability measure $\bar{\pi}$ by performing inference on a sequence of increasing probability spaces [23]. The target measure $\bar{\pi}$ and its normalization constant $\|\pi\|$ corresponding to the numerator

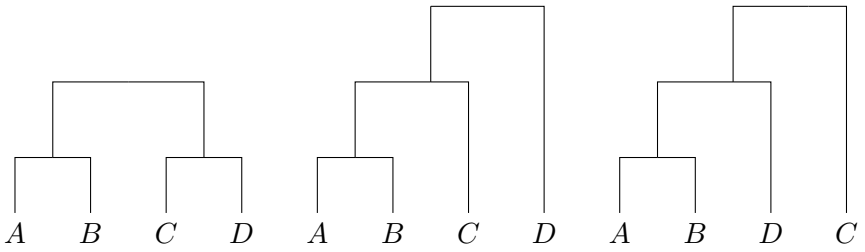


Figure 5.1: An example of the partial state $s = \{A, B\}$ for four taxa $\{A, B, C, D\}$ illustrated using its dual representation $\mathcal{D}(s)$. The dual state $\mathcal{D}(s) \subseteq \mathcal{T}$ corresponds to the three complete tree topologies. (left): $\{\{A, B\}, \{C, D\}\}$ (center): $\{\{A, B\}, \{A, B, C\}\}$ and (right): $\{\{A, B\}, \{A, B, D\}\}$.

and denominator in Eq. (5.1) are approximated by sequential importance resampling in R steps. Unlike standard SMC methods, the target is defined on a combinatorial set (the space of tree topologies $\mathcal{T}$). K sampled *partial states* (or *particles*) $\{s_{r,k}\}_{k=1}^{K} \in \mathcal{S}_r$ are drawn at each rank r and used to form a discrete positive measure,

$$\pi_{r,k} = \|\pi_{r-1,k}\| \frac{1}{K} \sum_{k=1}^{K} w_{r,k} \delta_{s,k}(s) \qquad \forall s \in \mathcal{S}, \tag{5.4}$$

where δ_s is the Kronecker delta and $w_{r,k}$ are the importance weights. Resampling ensures that particles remain on areas of high probability mass. Each resampled state $\tilde{s}_{r-1,k}$ of rank $r-1$ is then extended to a state of rank r by drawing from a proposal distribution $s_{r,l} \sim \nu^{+}_{s_{r,k}} : \mathcal{S} \to [0, 1]$. The importance weights are computed as follows:

$$w_{r,k} = w_(\tilde{s}_{r-1,k}, s_{r,k}) = \frac{\pi(s_{r,k})}{\pi(\tilde{s}_{r-1,k})} \cdot \frac{\nu^{-}_{s_{r,k}}(\tilde{s}_{r-1,k})}{\nu^{+}_{\tilde{s}_{r,k}}(s_{r,k})}, \tag{5.5}$$

where $\nu^{-}_{s_{r,k}}$ is a probability density over $\mathcal{S}$ correcting an over-counting problem [23]. The procedure is summarized in Algorithm 1. An unbiased estimate for the marginal likelihood can be constructed from the weights which converges in L^2 norm,

$$\hat{\mathcal{Z}}_{CSMC} \coloneqq \|\pi_{R,K}\| = \prod_{r=1}^{R} \left(\frac{1}{K} \sum_{k=1}^{K} w_{r,k} \right) \to \|\pi\|. \tag{5.6}$$

Algorithm 1 Combinatorial Sequential Monte Carlo

0. Initalization. $\forall\ k,\ s_{0,k} \leftarrow \perp,\ w_{0,k} \leftarrow 1/K$;

1. **for** *r = 0 to* $|X| - 1$ **do**

   2. **for** *k=1 to K* **do**

      a. Resample partial states

$$\tilde{s}_{r-1,1}, \cdots, \tilde{s}_{r-1,k} \sim \bar{\pi}_{r-1,k}$$

      b. Extend partial states

$$s_{r,k} \sim \nu^{+}_{\tilde{s}_{r-1,k}}$$

      c. Compute weights for new particles

$$w_{r,k} = w_(\tilde{s}_{r-1,k}, s_{r,k}) = \frac{\pi(s_{r,k})}{\pi(\tilde{s}_{r-1,k})} \cdot \frac{\nu^{-}_{s_{r,k}}(\tilde{s}_{r-1,k})}{\nu^{+}_{\tilde{s}_{r,k}}(s_{r,k})}$$

   end

end

Variational Inference VI is a technique for approximating the posterior $\log P_\theta(\mathcal{B}, \tau|\mathbf{Y})$ when marginalization of latent variables is not analytically feasible. By introducing a tractable distribution $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ it is possible to form a lower bound to the log-likelihood:

$$\log P_\theta(\mathbf{Y}) \geq \mathcal{L}_{\text{ELBO}}(\theta, \phi, \mathbf{Y}) \coloneqq \mathop{\mathbb{E}}_{Q}\left[\log \frac{P_\theta(\mathbf{Y}, \mathcal{B}, \tau)}{Q_\phi(\mathcal{B}, \tau|\mathbf{Y})}\right]. \tag{5.7}$$

Auto Encoding Variational Bayes [58] (AEVB) simultaneously trains $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ and $P_\theta(\mathbf{Y}, \mathbf{Z})$. The expectation in Eq. (5.7) is approximated by averaging Monte Carlo samples from $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ which are reparameterized by evaluating a deterministic function of a ϕ-independent random variable. When the ratio $P_\theta(\mathbf{Y}, \mathcal{B}, \tau)/Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ is concentrated around its mean, Jensen's inequality produces a tighter bound. The Importance Weighted Auto Encoder [10] (IWAE) leverages this observation by using estimators with the same mean that are more concentrated. K samples are drawn from the proposal and averaged over probability ratios

to form multi-sample objectives.

5.3 Variational Combinatorial Sequential Monte Carlo

Variational Objective. The idea of VCSMC is to simultaneously train the target and proposal distribution by maximizing a lower bound to the data log-likelihood, while using CSMC as the marginal likelihood estimator. We begin by defining a structured approximate posterior which factorizes over rank events. To do so, we will change notation from CSMC writing the resampled state $\tilde{s}_{r-1,k}$ as $s_{r-1}^{a_{r-1}^k}$ to make explicit the dependency of $\tilde{s}_{r-1}$ on its resampled index a_{r-1}^k. Let $q_\phi(s_{r,k}|s_{r-1}^{a_{r-1}^k})$ denote conditional the probability of state $s_{r,k}$ given the resampled state at the previous rank $s_{r-1}^{a_{r-1}^k}$. Subscripts ϕ and ψ denote discrete and continuous proposal parameters respectively:

$$Q_{\phi,\psi}\left(\mathcal{S}_{1:R}^{1:K}\right) \coloneqq \left(\prod_{k=1}^{K} q_\phi(s_{1,k}) \cdot q_\psi(\mathcal{B}_{1,k})\right) \tag{5.8}$$
$$\times \left(\prod_{k=1}^{K}\prod_{r=1}^{N-1} q_\phi\left(s_{r,k}|s_{r-1}^{a_{r-1}^k}\right) \cdot q_\psi\left(\mathcal{B}_{r,k}|\mathcal{B}_{r-1}^{a_{r-1}^k}\right) \cdot \textsc{Categorical}\left(a_{r-1}^k|\bar{w}_{r-1}^{1:K}\right)\right).$$

At the final rank event, an unbiased approximation to the likelihood is formed by averaging over importance weights, which, in turn represent the sample phylogenies that are constructed iterativly. A multi-sample variational objective formed is via the lower bound:

$$\mathcal{L}_{VCSMC} \coloneqq \mathop{\mathbb{E}}_{Q}\left[\log \hat{\mathcal{Z}}_{VCSMC}\right], \qquad \hat{\mathcal{Z}}_{VCSMC} \coloneqq \|\pi_{R,K}\| = \prod_{r=1}^{R}\left(\frac{1}{K}\sum_{k=1}^{K} w_{r,k}\right) \tag{5.9}$$

The presence of the DISCRETE densities over partial states presents a challenge for variational reparameterization. Unlike standard variational SMC methods, states are formed by sampling from a large combinatorial set. We take two approaches, the first is to drop discrete terms from the gradient estimates. The second is to reparameterize these terms as Gumbel-Softmax random variables forming a differentiable approximation through a con-

vex relaxation over the simplex. Continuous proposal terms are drawn by evaluating a deterministic function of a ψ-independent random variable.

Implementation Details. Constructing the objective $\mathcal{L}_{VCSMC}$ is done iteratively in three steps. The ExtendPartialState procedure requires selecting two partial states to coalesce by sampling without replacement. This is accomplished by defining Gumbel-Softmax random variables. The uniform log-probability for each index is perturbed by adding independent Gumbel distributed noise, after which the largest two elements are returned. For example let $U \sim \text{Uniform}(0, 1)$, we then form $G = \gamma - \log(-\log U)$ so that G can be reparameterized as $G' = G + \gamma$. The Resample procedure can also be reparameterized similarly by defining Gumbel-Softmax random variables.

The ComputeWeights step requires some care. In order to compute importance weights, the likelihood of a partial state must be computed using the sum-product algorithm, however the probability measure π is only defined on the target space of trees $\mathcal{T}$, and not the larger sample space of partial states $\mathcal{S} \coloneqq \cup_r \mathcal{S}_r$. Intuitively, the sum-product or pruning algorithm yields a maximum likelihood estimate for an evolutionary tree, but partial states contain disjoint subtrees or disjoint leaf nodes. To illustrate this, consider the jump chain for the partial state $\{A, B\}$ defined on the four taxa $\{A, B, C, D\}$ written as $s_1 = \{\{A, B\}, \{C\}, \{D\}\}$. This partial state admits three possible evolutionary trees (depicted in Fig 5.1 of the Appendix). The likelihood for each of these phylogenies contains a factor corresponding to the message passed from $\{A, B\}$ to the parent node $\text{Pa}(A, B)$. At the root node, in order to form the likelihood from a distribution over discrete characters, the pruning algorithm evaluates the inner product of Pa and the prior η (the stationary state of $\mathbf{Q}$). One extension of the target measure π into a measure on $\mathcal{S}$ suggested by [23]

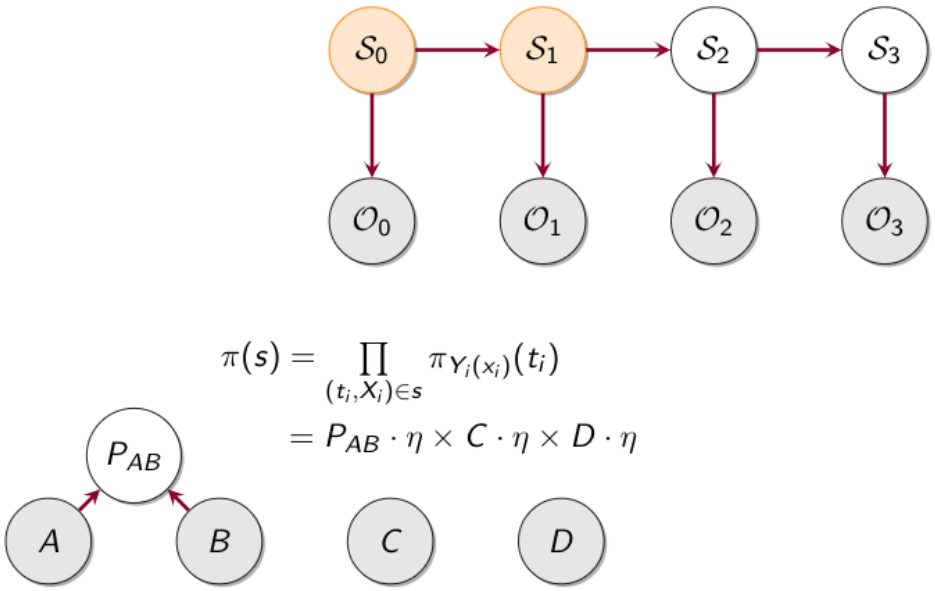
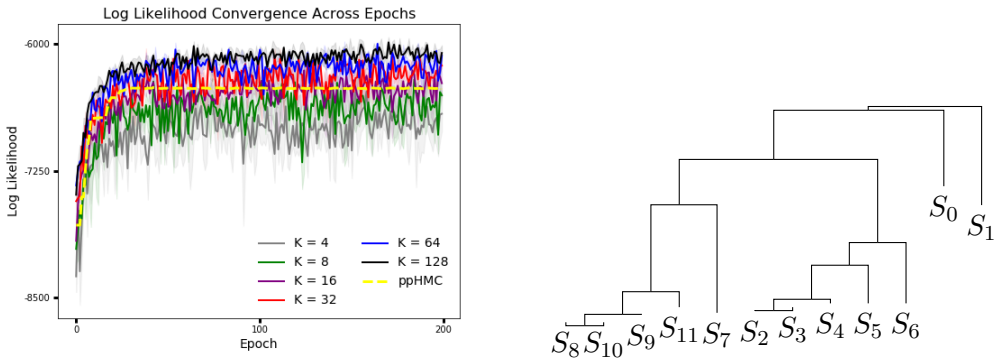


Figure 5.2: Illustration of the CSMC procedure to sample topologies. (Top): The graphical model showing dependencies between the observed taxa (DNA bases) $\mathcal{O} := \{A\}, \{B\}, \{C\}, \{D\}$ and the hidden state (DNA bases of the ancestral species) $\mathcal{S}_r | \mathcal{S}_{r-1}$. (Bottom): Illustration of the topology sampled for a single particle. At each rank event, two posets are selected uniformly to coalesce. The sum-product algorithm is then applied to marginalize over ancestral nodes. A probability is assigned to each disjoint set of clades by multiplying the distribution over characters with η. The probability of he sampled state is the product of all of the connected components in the forest.

is to treat all elements of the jump chain as trees (in this case, the subtree consisting of $\{A, B\}$ or PA(A, B) and non-coalescing singletons $\{C\}$ and $\{D\}$). The contribution of each of the elements in the jump chain to the likelihood is multiplied by taking the inner product of each distribution over characters with η. This extension has the advantage of passing information from the non-coalescing elements to the local weight update. We explore other extensions in future work.



(a) Log likelihood across epochs

(b) Phylogeny sampled from the posterior

Figure 5.3: (Left): Log likelihood values for K $= \{4, 8, 16, 32, 64, 128\}$ samples of VCSMC on the primates data averaged across 3 random seeds. Higher values of K produce tighter ELBO / larger log likelihood values with lower stochastic gadient noise. VCSMC with $K \geq 16$ outperforms probabilistic path Hamiltonain Monte Carlo (ppHMC) which is shown (yellow) for comparison. (Right): A single nonclock phylogeny sampled from the posterior with probability proportional to the importance weights at the final step. From left to right: M Mulatta, M Sylvanus, M Fascicularis, Saimiri Sciureus, Macaca Fuscata, Homo Sapiens, Pan, Gorilla, Pongo, Hylobates, Tarsius Syrichta, Lemur Catta. The leftmost clade partitions monkeys whereas the central and right clades partition hominids and prosimians respectively.

5.4 Results

Primate Mitochondrial DNA. We evaluate VCSMC on a benchmark dataset of nucleotide sequences of homologous fragments of primate mitochondrial DNA [35]. The dataset consists of 12 taxa $\{S_0, \cdots, S_{11}\}$ over 898 sites admitting 13,749,310,575 distinct tree topologies. The set of taxa includes five species of homonoids, four species of old world monkeys, one species of new world monkey and two species of prosimians. VCSMC is run with $K = \{4, 8, 16, 32, 64, 128\}$ particles, averaged over 3 random seeds. Fig 5.3 (left) shows higher values of K produce larger log likelihood values (tighter ELBO values) with lower

stochastic gradient noise. VCSMC with $K \geq 16$ outperforms probabilistic path Hamiltonain Monte Carlo (ppHMC) shown (yellow trace) for comparison. Fig 5.3 (right) illustrates a single phylogeny sampled from the posterior with probability proportional to the importance weights at the final step. From left to right: M Mulatta, M Sylvanus, M Fascicularis, Saimiri Sciureus, Macaca Fuscata, Homo Sapiens, Pan, Gorilla, Pongo, Hylobates, Tarsius Syrichta, Lemur Catta. The leftmost clade partitions monkeys whereas the central and right clades partition hominids and prosimians respectively.

5.5 Conclusion

We have sketched VCSMC, a method for model inference and parameter learning in Bayesian phylogenetics. To our knowledge, VCSMC is the first method to define a variational objective on the composite space of phylogenetic trees using Sequential Monte Carlo. VCSMC is written in Tensorflow. An implementation is available online at `https://github.com/amoretti86/phylo`.

Chapter 6

Summary and Future Work

This thesis has developed four statistical models and algorithms for approximate inference of spatial statistics and nonlinear dynamics. We summarize the main contributions below and discuss open questions as well as opportunities for future work.

- **Autoencoding Topographic Factors.**
  - We have extended Topographic Factor Analysis by proposing AETF, an amortized variational inference method that separates a set of overlapping signals into spatially localized source functions without knowledge of the original signals or the mixing process. We show that under this setup, model parameters scale independently of dataset size. AETF produces significant improvements over TFA in reconstruction error.
- **Nonlinear Evolution via Spatially Dependent Linear Dynamics.**
  - We have developed VIND, a variational inference framework that extends fLDS by modeling nonlinear evolution in the latent space. VIND uses a structured approximate posterior describing spatially-dependent linear dynamics and leverages

the fixed-point iteration method to speed up convergence.

- We have demonstrated VIND on single cell voltage data with state-of-the-art results in reconstruction error and explored the geometry of nonlinear spiking dynamics. We quantified the performance of the latent dynamics VIND by predicting future neural activity, substantially outperforming current methods.

- **Particle Smoothing Variational Objectives.**

  - We have presented SVO, the first variational inference method based on particle smoothing. In doing so, we have designed a novel backward simulation technique and a variational objective constructed from a smoothed approximate posterior. SVO sub-samples auxiliary random variables to enhance the support of the proposal and increase particle diversity.

  - We have developed a theoretical and empirical analysis of the signal to noise ratio (SNR) in filtering SMC, which motivates our choice of biased gradient estimators. We prove that introducing bias by dropping CATEGORICAL terms from the gradient estimate or using Gumbel-Softmax mitigates the adverse effect on the SNR.

  - We demonstrated our approach on three benchmark latent nonlinear dynamical systems tasks using a quantitative metric, rigorously showing that our algorithm consistently outperforms filtered objectives when given fewer Monte Carlo samples.

- **Variational Combinatorial Sequential Monte Carlo.**

  - We have sketched VCSMC, a method for simultaneous model inference and pa-

rameter learning in Bayesian phylogenetics. We established connections between discrete and continuous variational sequential search. To our knowledge, VCSMC is the first method to define a variational objective on the composite space of phylogenetic trees using Sequential Monte Carlo.

– We have shown that VCSMC provides suitable estimates of the posterior when applied to a benchmark dataset of primate mitochondrial DNA and performs favorably when compared with the state of the art HMC methods.

An alternative interpretation of SVO involves *twisting*, or changing the sequence of intermediate target distributions to maximize the accuracy of the estimate $\hat{\mathrm{Z}}_{SVO}$. For a review of twisting, see [92]. The SVO algorithm can be thought of as performing twisting by using information from future observations to change the target density [91]. One direction for future work is to develop *auxiliary* or *twisted* backwards variational SMC methods without costly subsampling.

Twisting can also be used in developing extensions to VCSMC. The CSMC algorithm samples vertices to coalesce uniformly from the proposal defined in Eq. (5.8). Another direction for future work thus involves twisting the target density in VCSMC by using information from future iterations to guide sampling from the proposal distribution. It would also be useful to develop alternative extensions of the probability measure π defined in Eq. (5.4) from the space of partial states to the space of complete phylogenies. We find that one limitation of the natural forest extension introduced by [23] is that the likelihood estimate increases across rank events. Twisting may also play a role in designing novel methods of assigning probabilities to partial states that yield unbiased likelihood estimates.

As a variational autoencoder, VCSMC has the ability to accomodate expressive genera-

tive models of evolution. Another direction for future work involves pararameterizing the components of the transition rate matrix with a deep generative model or with the output of a neural network. In this setup, the evolution of each site is modeled as a nonlinear function of spatial position on the genome and learning can be sped up via stochastic gradient descent with minibatch iteration. There also exist exciting opportunities for applications to betacoronavirus and spike glycoprotein data.

Bibliography

[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[2] Lang Annika and Potthoff Jürgen. Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3):195–214, 2011.

[3] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv: 1511.07367*, 2015.

[4] Jean Bérard, Pierre Del Moral, Arnaud Doucet, et al. A lognormal central limit theorem for particle approximations of normalizing constants. *Electronic Journal of Probability*, 19, 2014.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[6] Alexandre Bouchard-Côté, Sriram Sankararaman, and Michael Jordan. Phylogenetic inference via sequential monte carlo. *Systematic biology*, 61:579–93, 01 2012.

[7] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61, Jun 2009.

[8] J. Rodney Brister, Danso Ako-adjei, Yiming Bao, and Olga Blinkova. NCBI Viral Genomes Resource. *Nucleic Acids Research*, 43(D1):D571–D577, 11 2014.

[9] K. E. Buchanan, J. Friedrich, Ian Kinsella, Patrick Stinson, Pengcheng Zhou, Felipe Gerhard, John Ferrante, Graham Dempsey, and Liam Paninski. Constrained matrix factorization methods for denoising and demixing voltage imaging data. In *Cosyne*, 2018.

[10] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders, 2015.

[11] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.

[12] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[13] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018.

[14] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *CoRR*, abs/1506.02216, 2015.

[15] C. Cremer, Q. Morris, and D. Duvenaud. Reinterpreting Importance-Weighted Autoencoders. *Workshop at the International Conference on Learning Representations*, 2017.

[16] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17:1500 EP –, 08 2014.

[17] Persi Diaconis. The markov chain monte carlo revolution. *Bulletin of the American Mathematical Society*, 46:179–205, 2009.

[18] Daniel Hernandez Diaz, Antonio Khalil Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. A novel variational family for hidden nonlinear markov models. *arXiv preprint arXiv:1611.00712*, 2018.

[19] Daniel Hernandez Diaz, Antonio Khalil Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. A novel variational family for hidden nonlinear markov models. 2019.

[20] Vu Dinh, Arman Bilge, Cheng Zhang, and Frederick A. Matsen, IV. Probabilistic path Hamiltonian Monte Carlo. volume 70 of *Proceedings of Machine Learning Research*, pages 1009–1018, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[21] Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4470–4479. Curran Associates, Inc., 2018.

[22] A Doucet, SJ Godsill, and C Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197 – 208, 2000.

[23] Arnaud Doucet, Liangliang Wang, and Alexandre Bouchard-Côté. Bayesian phylogenetic inference using a combinatorial sequential monte carlo method. *Journal of the American Statistical Association*, 01 2015.

[24] Eric W. Eisstein. Gershgorin circle theorem. from mathworld–a wolfram web resource.

[25] J Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

[26] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *arXiv:1710.05741*, 2017.

[27] Yuanjun Gao, Evan Archer, Liam Paninski, and John P. Cunningham. Linear dynamical neural population models through nonlinear embedding. *NIPS 2016*, 2016.

[28] Yuanjun Gao, Lars Busing, Krishna V Shenoy, and John P Cunningham. High-dimensional neural spike train analysis with generalized count linear dynamical systems. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2044–2052. Curran Associates, Inc., 2015.

[29] Samuel Gershman, David M. Blei, Kenneth A. Norman, and Per B. Sederberg. Decomposing spatiotemporal brain patterns into topographic latent sources. *NeuroImage*, 98:91–102, 2014.

[30] Samuel Gershman, David M. Blei, Francisco Pereira, and Kenneth A. Norman. A topographic latent source model for fmri data. *NeuroImage*, 57(1):89–100, 2011.

[31] Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, 2004.

[32] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113, 1993.

[33] Pieralberto Guarniero, Adam Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 08 2016.

[34] Zengcai V. Guo, Nuo Li, Daniel Huber, Eran Ophir, Diego Gutnisky, Jonathan T. Ting, Guoping Feng, and Karel Svoboda. Flow of cortical activity underlying a tactile decision in mice. *Neuron*, 81(1):179–194, 2018/05/16 2014.

[35] K Hayasaka, T Gojobori, and S Horai. Molecular phylogeny and evolution of primate mitochondrial DNA. *Molecular Biology and Evolution*, 5(6):626–644, 11 1988.

[36] Jeremy Heng, Adrian N. Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo, 2017.

[37] Daniel Hernandez, Antonio Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. A novel variational family for hidden nonlinear markov models. *CoRR*, abs/1811.02459, 2018.

[38] Daniel Hernandez, Antonio Khalil Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. Nonlinear evolution via spatially-dependent linear dynamics for electrophysiology and calcium data. *Neurons, Behavior, Data analysis and Theory*, 2018.

[39] Daniel Hernandez, Liam Paninski, and John Cunningham. Variational inference for nonlinear dynamics. *TSW, NIPS 2017*, 2017.

[40] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Bulletin of Mathematical Biology*, 52(1):25–71, Jan 1990.

[41] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[42] John P. Huelsenbeck and Fredrik Ronquist. MRBAYES: Bayesian inference of phylogenetic trees . *Bioinformatics*, 17(8):754–755, 08 2001.

[43] Sebastian Höhna and Alexei Drummond. Guided tree topology proposals for bayesian phylogenetic inference. *Systematic biology*, 61:1–11, 01 2012.

[44] E Izhikevich. Dynamical systems in neuroscience. *MIT Press*, page 111, July 2007.

[45] et al J. Friedrich. Fast constrained non-negative matrix factorization for whole-brain calcium imaging data. In *NIPS workshop on Statistical Methods for Understanding Neural Systems*, 2015.

[46] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[47] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ICML2014*, January 2014.

[48] Matthew J. Johnson, David Duvenaud, Alexander B. Wiltschko, Sandeep R. Datta, and Ryan P. Adams. Composing graphical models with neural networks for structured representations and fast inference. *arXiv: 1603.06277*, 2016.

[49] Allan R. Jones, Caroline C. Overly, and Susan M. Sunkin. The allen brain atlas: 5 years and beyond. *Nature Reviews Neuroscience*, 10:821 EP –, 10 2009.

[50] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, Nov 1999.

[51] Leo P. Kadanoff. More is the same; phase transitions and mean field theories. *Journal of Statistical Physics*, 137(5-6):777–797, Sep 2009.

[52] R. Kalantari, J. Ghosh, and M. Zhou. Nonparametric Bayesian Sparse Graph Linear Dynamical Systems. *ArXiv: 1802.07434*, February 2018.

[53] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv: 1312.6114*, December 2013.

[54] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[55] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[56] Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational autoencoders with inverse autoregressive flow. In *NIPS*, pages 4736–4744, 2016.

[57] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[59] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[60] Mike Klaas, Mark Briers, Nando De Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang. Fast particle smoothing: If i had a million particles. In *Proceedings of the 23rd international conference on Machine learning*, pages 481–488. ACM, 2006.

[61] Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep kalman filters, 2015.

[62] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models, 2016.

[63] Clemens Lakner, Paul van der Mark, John P. Huelsenbeck, Bret Larget, and Fredrik Ronquist. Efficiency of Markov Chain Monte Carlo Tree Proposals in Bayesian Phylogenetics. *Systematic Biology*, 57(1):86–103, 02 2008.

[64] Dietrich Lawson, George Tucker, Dai Bo, and Ragesh Raganath. Revisiting auxiliary latent variables in generative models. *ICLR Workshops*, 2019.

[65] Dietrich Lawson, George Tucker, Christian Naesseth, Chris Maddison, Ryan Adams, and Yeh Teh. Twisted variational sequential monte carlo. *Bayesian Deep Learning Workshop, NIPS*, 2016.

[66] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.

[67] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[68] Nuo Li, Tsai-Wen Chen, Zengcai V. Guo, Charles R. Gerfen, and Karel Svoboda. A motor cortex circuit for motor planning and movement. *Nature*, 519:51 EP –, 02 2015.

[69] Dawen Liang, Rahul Krishnan, Matthew Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of The Web Conference (WWW), 2018*, 2018.

[70] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922, 2017.

[71] Fredrik Lindsten, Jouni Helske, and Matti Vihola. Graphical model inference: Sequential monte carlo meets deterministic approximations. In *Advances in Neural Information Processing Systems 31*. 2018.

[72] Fredrik Lindsten, Jouni Helske, and Matti Vihola. Graphical model inference: Sequential monte carlo meets deterministic approximations. In *Advances in Neural Information Processing Systems*, pages 8190–8200, 2018.

[73] Fredrik Lindsten and Thomas B. Schön. Backward simulation methods for monte carlo statistical inference. *Found. Trends Mach. Learn.*, 6(1):1–143, August 2013.

[74] Hedibert Freitas Lopes, Esther Salazar, and Dani Gamerman. Spatial dynamic factor analysis. *Bayesian Anal.*, 3(4):759–792, 12 2008.

[75] Ying Ma, Mohammed A Shaik, Sharon H Kim, Mariel G Kozberg, David N Thibodeaux, Hanzhi T Zhao, Hang Yu, and Elizabeth MC Hillman. Wide-field optical mapping of neural activity and brain haemodynamics: considerations and novel approaches. *Phil. Trans. R. Soc. B*, 371(1705):20150360, 2016.

[76] Ying Ma, Mohammed A Shaik, Mariel G Kozberg, Sharon H Kim, Jacob P Portes, Dmitriy Timerman, and Elizabeth MC Hillman. Resting-state hemodynamics are spatiotemporally coupled to synchronized and symmetric neural activity in excitatory neurons. *Proceedings of the National Academy of Sciences*, 113(52):E8463–E8471, 2016.

[77] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6573–6583. Curran Associates, Inc., 2017.

[78] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[79] Jeremy R. Manning, Rajesh Ranganath, Waitsang Keung, Nicholas B. Turk-Browne, Jonathan D. Cohen, Kenneth A. Norman, and David M. Blei. Hierarchical topographic factor analysis. In *International Workshop on Pattern Recognition in Neuroimaging, PRNI 2014, Tübingen, Germany, June 4-6, 2014*, pages 1–4, 2014.

[80] Jeremy R. Manning, Rajesh Ranganath, Kenneth A. Norman, and David M. Blei. Topographic Factor Analysis: A Bayesian Model for Inferring Brain Networks from Neural Data. *PLoS ONE*, 9(5):e94914, may 2014.

[81] Antonio Moretti, Andrew Stirn, Gabriel Marks, and Itsik Pe'er. Autoencoding topographic factors. *Journal of Computational Biology*, 26(6):546–560, 2019. PMID: 30526005.

[82] Antonio Moretti, Andrew Stirn, Gabriel Marks, and Itsik Pe'er. Autoencoding topographic factors. *Journal of Computational Biology*, 26(6):546–560, 2019.

[83] Antonio Moretti, Zizhao Wang, Luhuan Wu, and Itsik Pe'er. Smoothing nonlinear variational objectives with sequential monte carlo. *ICLR Workshops*, 2019.

[84] Antonio Moretti, Liyi Zhang, and Itsik Pe'er. Variational combinatorial sequential monte carlo for bayesian phylogenetic inference. *Machine Learning in Computational Biology*, 2020.

[85] Antonio Khalil Moretti, Zizhao Wang, Luhuan Wu, Iddo Drori, and Itsik Pe'er. Particle smoothing variational objectives. *CoRR*, abs/1909.09734, 2019.

[86] Antonio Khalil Moretti, Zizhao Wang, Luhuan Wu, Iddo Drori, and Itsik Pe‘er. Variational objectives for markovian dynamics with backward simulation. *European Conference on Artificial Intelligence*, 2020.

[87] D.A. Morrison. Multiple sequence alignment for phylogenetic purposes. *Aust. Syst. Bot.*, 19:476–539, 01 2006.

[88] Eran A Mukamel, Axel Nimmerjahn, and Mark J. Schnitzer. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63:747–760, 2009.

[89] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 968–977, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[90] Christian A. Naesseth, Scott W. Linderman, Rajesh Ranganath, and David M. Blei. Variational sequential monte carlo. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pages 968–977. PMLR, 2018.

[91] Christian A. Naesseth, Fredrik Lindsten, and David Blei. Markovian score climbing: Variational inference with kl(p——q), 2020.

[92] Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Elements of sequential monte carlo, 2019.

[93] Manfred Opper and David Saad, editors. *Advanced mean field methods: theory and practice*. Neural Information Processing. MIT, February 2001.

[94] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search, 2012.

[95] Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018.

[96] Liam Paninski and John Cunningham. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *bioRxiv*, 2017.

[97] Liam Paninski and John Cunningham. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *bioRxiv*, page 196949, 2017.

[98] Adam Persing and Ajay Jasra. Likelihood computation for hidden Markov models via generalized two-filter smoothing. *Statistics & Probability Letters*, 83(5):1433–1442, 2013.

[99] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[100] Eftychios A. Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A. Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, Misha Ahrens, Randy Bruno, Thomas M. Jessell, Darcy S. Peterka, Rafael Yuste, and Liam Paninski. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285 – 299, 2016.

[101] Tom Rainforth, Adam R Kosiorek, Tuan Anh Le, Chris J Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537*, 2018.

[102] Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference, 2013.

[103] Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pages 814–822, 2014.

[104] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1530–1538, 2015.

[105] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

[106] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

[107] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[108] Fredrik Ronquist, Maxim Teslenko, Paul Mark, Daniel Ayres, Aaron Darling, Sebastian Höhna, Bret Larget, Liang Liu, Marc Suchard, and John Huelsenbeck. Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, 61:539–42, 03 2012.

[109] Tim Salimans, Diederik P. Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap, 2014.

[110] Tim Salimans and David A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, Dec 2013.

[111] Laurent Saloff-Coste. *Lectures on finite Markov chains*, pages 301–413. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[112] Charles Semple and Mike Steel. Phylogenetics. 2003.

[113] Zarrar Shehzad, A. M. Clare Kelly, Philip T. Reiss, Dylan G. Gee, Kristin Gotimer, Lucina Q. Uddin, Sang Han Lee, Daniel S. Margulies, Amy Krain Roy, Bharat B. Biswal, Eva Petkova, F. Xavier Castellanos, and Michael P. Milham. The resting brain: Unconstrained yet reliable. *Cerebral Cortex*, 19(10):2209–2229, 2009.

[114] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. Lfads - latent factor analysis via dynamical systems, 2016.

[115] Valentine Svensson, Sarah A. Teichmann, and Oliver Stegle. SpatialDE - Identification Of Spatially Variable Genes. *bioRxiv*, pages 143321+, May 2017.

[116] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

[117] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. pub-SIAM, pub-SIAM:adr, 1997.

[118] Shijia Wang and Liangliang Wang. Particle gibbs sampling for bayesian phylogenetic inference, 2020.

[119] Ziqiang Wei, Hidehiko Inagaki, Nuo Li, Karel Svoboda, and Shaul Druckmann. An orderly single-trial organization of population dynamics in premotor cortex predicts behavioral variability. *Nature Communications*, 10(1):216, 2019.

[120] Nick Whiteley and Adam M. Johansen. *Auxiliary particle filtering: recent developments*, page 52–81. Cambridge University Press, 2011.

[121] K. J. Worsley, S. Marrett, P. Neelin, and A.C. Evans. A unified statistical approach for determining significant signals in location and scale space images of cerebral activation, 1996.

[122] Anqi Wu, S Pashkovski, R.S. Datta, and J.W. Pillow. Learning a latent manifold of odor representations from neural responses in piriform cortex. *NIPS 2018*, 2018.

[123] Anqi Wu, NG Roy, S Keeley, and JW Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *NIPS 2017*, 2017.

[124] Anqi Wu, Nicholas A. Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3496–3505. Curran Associates, Inc., 2017.

[125] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1881–1888. Curran Associates, Inc., 2009.

[126] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614–635, 07 2009.

[127] Cheng Zhang and Frederick A. Matsen IV. Variational bayesian phylogenetic inference. In *International Conference on Learning Representations*, 2019.

[128] Cheng Zhang and Frederick A Matsen IV. Generalizing tree probability estimation via bayesian networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1444–1453. Curran Associates, Inc., 2018.

[129] Y. Zhao and I. Memming Park. Variational Joint Filtering. *arXiv: 1707.09049*, July 2017.

[130] Yuan Zhao, Josue Nassar, Ian Jordan, Mónica Bugallo, and Il Memming Park. Streaming variational monte carlo, 2019.

Appendix A

Intractability of VIND

Consider a parent distribution that factorizes across two time steps. Using Eq. (3.20), $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ can be written as:

$$Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) = \kappa_{\phi,\varphi}(\mathbf{X})\tilde{Q}_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})\,, \tag{A.1}$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$ and

$$\tilde{Q}(\mathbf{Z}|\mathbf{X}) = g(\mathbf{z}_0|\mathbf{x}_0)g(\mathbf{z}_1|\mathbf{x}_1) \cdot h_0(\mathbf{z}_0)h(\mathbf{z}_1|\mathbf{z}_0)\,, \tag{A.2}$$

with the normalization constant:

$$\kappa^{-1}_{\phi,\varphi}(\mathbf{X}) = \int \tilde{Q}(\mathbf{Z}|\mathbf{X})\, d\mathbf{Z}\,. \tag{A.3}$$

We illustrate that direct integration of $\tilde{Q}$, as in Eq. (A.3), is intractable. For simplicity, set the variance parameters to the identity:

$$\Gamma_0 = \Gamma = \sigma_\varphi = \mathbb{I}_{d_Z}\,. \tag{A.4}$$

Then, marginalizing first with respect to $\mathbf{z}_1$:

$$\int \tilde{Q}\, d\mathbf{z}_1 = h(\mathbf{z}_0)g(\mathbf{z}_0|\mathbf{x}_0) \cdot I(\mathbf{z}_0|\mathbf{x}_1) \tag{A.5}$$

where $I(\mathbf{z}_0|\mathbf{x}_1)$ is given by

$$I(\mathbf{z}_0|\mathbf{x}_1) = \int \exp\left\{-\frac{1}{2}\Delta(\mathbf{z}_1|\mathbf{z}_0)^T\Delta(\mathbf{z}_1|\mathbf{z}_0) - \frac{1}{2}\Delta(\mathbf{z}_1|\mathbf{x}_1)^T\Delta(\mathbf{z}_1|\mathbf{x}_1)\right\} d\mathbf{z}_1\,, \tag{A.6}$$

with

$$\Delta(\mathbf{z}_1|\mathbf{z}_0) = \mathbf{z}_1 - a_\phi(\mathbf{z}_0)\,, \tag{A.7}$$

$$\Delta(\mathbf{z}_1|\mathbf{x}_1) = \mathbf{z}_1 - \mu_\varphi(\mathbf{x}_1)\,. \tag{A.8}$$

Evaluating the integral,

$$I(\mathbf{z}_0|\mathbf{x}_1) = \frac{1}{(2\pi)^{d_Z}} \exp\left\{-\frac{1}{4}\big(a_\phi(\mathbf{z}_0) - \mu_\varphi(\mathbf{x}_1)\big)^2\right\}. \tag{A.9}$$

The desired normalizing constant is then

$$\kappa^{-1} = \int h(\mathbf{z}_0) g(\mathbf{z}_0|\mathbf{x}_0) I(\mathbf{z}_0|\mathbf{x}_1)\, d\mathbf{z}_0\,. \tag{A.10}$$

The exponential in the integrand includes terms in $a_\phi(\mathbf{z}_0)$ and $a_\phi(\mathbf{z}_0)^2$ which are non-quadratic in $\mathbf{z}_0$. These terms ensure that marginalization is intractable. However, these are required by VIND's factorization of the approximate posterior inherited from the Generative Model.

Appendix B

FPI Convergence

The Fixed-Point Iteration method. The Fixed-Point Iteration (FPI) is a general method for numerically approximating solution of k nonlinear equations in k independent variables:

$$F_i(x) = 0\,. \quad i = 1, \ldots, k \tag{B.1}$$

where $x \in \mathbb{R}^k$. Rewriting the equation in the form

$$x = T(x) \tag{B.2}$$

is always possible for some $T : \mathbb{R}^k \to \mathbb{R}^k$. An initial estimate $\mathbf{x}_0$ is chosen and the FPI algorithm produces the sequence x_n by repeatedly applying T:

$$x_n = T(x_{n-1})\,. \tag{B.3}$$

If this sequence converges, then it is Cauchy and its limit is the solution of Eq. (B.2).

Theorem 3. (PBC) Let T be Lipschitz-continuous in $U \subset X$. That is

$$d_X\big(T(x), T(y)\big) \leq K \cdot d_X(x, y)\,, \qquad \textit{for } x, y \in U \tag{B.4}$$

for some real number K. If $K \in [0, 1)$ then T has a unique fixed point $x^* \in U$ and the Picard sequence $\{x_n\}$ for $n = 0, \ldots, \infty$ where

$$x_n = T(x_{n-1}) = T^n(x_0) \tag{B.5}$$

converges to x^* for any initial guess $x_0 \in U$.

It can be further shown that the rate of convergence is exponential in the iteration number

$$d_X(x_n, x^*) \leq K^n \cdot d_X(x_0, x^*)\,. \tag{B.6}$$

When the PBC theorem holds, we say the map T is a K-contraction.

Let $J_{ij}(x)$ be the Jacobian of the map T, $i, j = 1, \ldots, k$. Let $\{\lambda_i(x_0)\}$ be the eigenvalues of J_{ij} evaluated at x_0. A common way to show that a mapping $T : \mathbb{R}^k \to \mathbb{R}^k$ is a contraction under the Euclidean distance in a neighborhood of $x_0 \in \mathbb{R}^k$, is to show that $\max \lambda_i < 1$. In turn this can be proven using the Gershgorin Circle Theorem that gives a bound to the spectrum of a square matrix A:

Theorem 4. *(Gershgorin) Let A be an $n \times n$ matrix with entries in $\mathbb{C}$. For each i, let D_i be the disc,*

$$D_i = \left\{ z \in \mathbb{C} : |z - A_{ii}| \leq \sum_{j \neq i} |A_{ij}| \right\}\,, \tag{B.7}$$

then the eigenvalues of A lie in $D_1 \cup D_2 \cup \cdots \cup D_n$.

It follows that an upper bound on the maximum absolute value for the eigenvalues of A is given by:

$$\max_i \lambda_i \leq \max_i \sum_j |a_{ij}|\,. \tag{B.8}$$

The FPI iteration convergence is satisfied if the following holds:

$$\max_i \sum_j |J_{ij}| = \max_i \sum_j \left| \frac{\partial T_i}{\partial x_j} \right| < 1\,. \tag{B.9}$$

Appendix C

Implementation Details for VIND's FPI Convergence

The results of Chapter 3 were obtained by setting $\alpha = 10^{-2}$. We find that for LLDS/VIND to converge, the FPI map $r_{\phi,\varphi}$:

$$\mathbf{P} = r_{\phi,\varphi}(\mathbf{P}, \mathbf{X}) \tag{C.1}$$

$$r_{\phi,\varphi}(\mathbf{P}, \mathbf{X}) = \tilde{\mathbf{\Lambda}}^{-1} \cdot \mathbf{Y}(\mathbf{P}) \tag{C.2}$$

$$\tilde{\mathbf{\Lambda}} = \mathbf{\Lambda} + \mathbf{S}(\mathbf{Z}) \tag{C.3}$$

$$\mathbf{Y}(\mathbf{P}) = \mathbf{\Lambda}_\varphi \mathbf{M}_\varphi - \frac{1}{2}\mathbf{P}^T \frac{\partial \mathbf{S}_\phi(\mathbf{P})}{\partial \mathbf{P}} \mathbf{P}\,. \tag{C.4}$$

must be in the contractive regime within a domain D, $D \subset \mathbb{R}^{T \times d_Z}$. As discussed in App. B, a necessary condition for this to occur is that the Jacobian J of the map $r_{\phi,\varphi}$:

$$J_{ij}(\mathbf{Z}) = \frac{\partial r_i}{\partial Z_j}\,, \quad \text{for } i, j \in 1, \ldots, T \times d_Z\,. \tag{C.5}$$

satisfies Eq. (B.9).

We note that when $\alpha = 0$, $\log Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ is a quadratic form in $\mathbf{Z}$. In this case, VIND reduces to fLDS and the FPI is a convex optimization problem. Eq.(3.23) is linear with a closed form solution. As a result, deviations from convergence and convexity are always $O(\alpha)$.

To guarantee that the FPI is in a contractive regime, the entries J_{ij} should be suppressed both by the small hyperparameter α and by the gradients of the deep neural network $B_\phi(\mathbf{z}_t)$,

Eq. (3.29). Dropping the subleading terms in Eq. (C.4) proportional to the gradient of $\mathbf{S}(\mathbf{Z})$:

$$\frac{\partial r_i}{\partial Z_j} \simeq \tilde{\mathbf{\Lambda}}^{-1} \frac{\partial \tilde{\mathbf{\Lambda}}}{\partial Z_j} \tilde{\mathbf{\Lambda}}^{-1} \cdot \mathbf{\Lambda} \mathbf{M}_\varphi \simeq \tilde{\mathbf{\Lambda}}^{-1}_{ik} \frac{\partial \tilde{\mathbf{\Lambda}}_{kl}}{\partial Z_j} \cdot r_l \,. \tag{C.6}$$

Let L be the linear dimension of a bounding box in the phase space subsuming the latent paths,

$$r \sim L \,. \tag{C.7}$$

Let σ^2 be the typical scale of the entries of the diagonal recognition covariance matrix $\mathbf{\Lambda}$, and let $\sigma^2_{\text{ev}} = \Gamma^{-1}$ represent the typical scale of the evolution covariance. We consider the case in which $\mathbf{\Lambda} \gtrsim \mathbf{S}(\mathbf{Z})$ for simplicty, so that in magnitude,

$$\tilde{\mathbf{\Lambda}}^{-1} \sim \sigma^2 \cdot \mathbb{I} \tag{C.8}$$

Let Δ be the typical rate of variation of the entries of the matrix $B(\mathbf{z}_t)$. Then

$$\frac{\partial \tilde{\mathbf{\Lambda}}_{kl}}{\partial Z_j} \sim \frac{\alpha \Delta}{\sigma^2_{\text{ev}}} V_{klj} \tag{C.9}$$

where V_{klj} is a sparse tensor and only the (j, j), $(j, j+1)$ and $(j+1, j)$ blocks in $\tilde{\mathbf{\Lambda}}_{kl}$ can depend on Z_j. Substituting terms into Eq. (B.9) produces a simple rule that suggests when the FPI is in the contractive regime

$$\max_i \sum_j \left| \frac{\partial r_i}{\partial Z_j} \right| \sim c \frac{\sigma^2}{\sigma^2_{\text{ev}}} \, \alpha \, \Delta \, L \,. \tag{C.10}$$

where c is an $O(1)$ constant.

In the experiments, the hyperparameters and architecture of the evolution network are chosen so that

$$\alpha \Delta \ll \frac{\sigma^2_{\text{ev}}}{L \sigma^2} \tag{C.11}$$

at initialization with good results.