Adaptive, Robust, and Scalable Bayesian Filtering for Online Learning

Gerardo Duran-Martin

Supervised by Alexander Shestopaloff and Kevin Murphy

School of Mathematical Sciences

April 2025

Submitted in partial fulfillment of the requirements
of the Degree of Doctor of Philosophy

This work was supported by UKRI — Engineering and Physical Sciences Research Council

Statement of Originality

I, Gerardo Duran Martin, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by other, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that Queen Mary University of London has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Gerardo Duran-Martin
Date: April 2025.

Details of collaboration and publications:

1. The contents in Chapter 3 are based on the paper Duran-Martin et al. (2025).
2. The contents in Chapter 4 are based on the paper Duran-Martin et al. (2024).
3. The contents in Chapter 5 are based on the papers Duran-Martin et al. (2022); Cartea et al. (2023b); Chang et al. (2023).

Abstract

In this thesis, we introduce Bayesian filtering as a principled framework for tackling diverse sequential machine learning problems, including online (continual) learning, prequential (one-step-ahead) forecasting, and contextual bandits. To this end, this thesis addresses key challenges in applying Bayesian filtering to these problems: adaptivity to non-stationary environments, robustness to model misspecification and outliers, and scalability to the high-dimensional parameter space of deep neural networks. We develop novel tools within the Bayesian filtering framework to address each of these challenges, including: (i) a modular framework that enables the development adaptive approaches for online learning; (ii) a novel, provably robust filter with similar computational cost to standard filters, that employs Generalised Bayes; and (iii) a set of tools for sequentially updating model parameters using approximate second-order optimisation methods that exploit the overparametrisation of high-dimensional parametric models such as neural networks. Theoretical analysis and empirical results demonstrate the improved performance of our methods in dynamic, high-dimensional, and misspecified models.

"You know, allowing awareness for something that's unlikely is not a disease," she said. "If you're talking about a filter, you should understand how they work. Optimal filters will still block a few things that you actually wanted to go through—and will still allow for some things that you wanted to block to instead go through. That's for an optimal filter." [...] A species like ours, with survival so clearly based on intelligence and information, should not block the risk of blocking and throwing away potentially valuable ideas.

–Professor Karl Deisseroth, "Connections"

Acknowledgements

As a good friend of mine once wrote, this thesis would not have been possible without a combination of luck and hard work. After arriving in the UK to pursue my MSc studies during a global pandemic, I was fortunate to meet Alex Shestopaloff, who, despite my lack of research experience at the time, kindly welcomed me as one of his PhD students. Around the same time, I had the pleasure of working with Kevin Murphy during the Google Summer of Code programme in 2021. It was there that Kevin posed a question which has since become central to this thesis: "Did you know you can train a neural network with a Kalman filter?"

To Alex and Kevin, I owe my deepest gratitude for your trust, guidance, and patience throughout this process. Your mentorship has been invaluable, and this thesis would not exist without your support.

I thank Eftychia Solea and Arnaud Doucet for their insightful comments and constructive feedback, which have enhanced the quality and clarity of this thesis.

I extend my heartfelt thanks to Álvaro Cartea, the director of the Oxford-Man Institute of Quantitative Finance (OMI), for opening the institute's doors to me and for demonstrating the value of precision and care in academic research. I am equally grateful to members of the OMI, with whom I had the privilege of collaborating. In particular, I would like to acknowledge Alvaro Arroyo, Patrick Chang, Sergio Calvo-Ordoñez, and Fayçal Drissi for their friendship and for sharing their knowledge, from which I have greatly benefited.

To Leandro Sánchez-Betancourt, from whom I learnt to appreciate the writing process and to take care of the smallest mathematical details. It has been a true privilege to work alongside you and to call you a friend and a co-author.

I am also grateful to AHL Man Group for providing me with a summer home to learn and explore research ideas in the real-world. During my time there, I learnt the importance of prioritising mathematical simplicity. I would like to give special thanks to Anthony Ledford for the fascinating and thought-provoking discussions on mathematics, statistics, and their applications in finance.

The Google TPU Research Cloud's computational resources were crucial for the execution of the Jax code, and I am thankful for their support.

To my parents, I owe so much. From my late father, I learned the value of taking pride

in one's work and the joy of craftsmanship. From my mother, I learned the importance of being consistent.

Finally, to Andi Flores—none of this would have been possible without your unwavering love, support, and trust. You have reminded me to cherish life's simple pleasures and to embrace the adventures we have shared across the world. For that, and for so much more, I thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sequential problems are central to machine learning and arise in a variety of settings, including test-time adaptation (Schirmer et al., 2024), prequential forecasting (Gama et al., 2008), neural bandits (Riquelme et al., 2018), online continual learning (Dohare et al., 2024), and deep reinforcement learning (Asadi et al., 2024). These problems can be formulated as online learning tasks, where an agent observes a time-indexed sequence of data and predicts the next unobserved value in the sequence (Zhang, 2023, Chapter 1). Typically, predictions are informed by past observations and may depend on additional exogenous variables, referred to as *features*.

Filtering methods provide a principled framework for tackling such challenges in sequential decision-making. These methods focus on inferring the unknown state of a dynamic system from noisy observations, making them well-suited for parametric online learning tasks where the unknown state are taken to be the model parameters. Among filtering methods, the Kalman filter (KF) has been particularly influential, serving as a foundational algorithm that has inspired numerous extensions and applications (Leondes, 1970; Grewal and Andrews, 2010). Figure 1.1 presents an example of a two-dimensional dynamical system (top panel), which seeks to recover from a one-dimensional projection (bottom panel).

Arguably, the widespread appeal of filtering methods stems from three key properties: extensibility, a Bayesian foundation, and broad applicability. First, the extensibility of filtering algorithms has enabled researchers to develop numerous variants of the KF. For instance, extensions have made the KF (i) *robust* to outlier measurements (West, 1981), (ii) *adaptive* to non-stationary environments (Mehra, 1972), and (iii) *scalable* to high-dimensional state and observation spaces (Evensen, 1994). Variants of the KF remain an active area of research, with efforts to further improve robustness, adaptability, and scalability (see e.g., Tao and Yau, 2023; Zhu et al., 2022; Vilmarest and Wintenberger, 2024; Chen et al., 2022; Schmidt et al., 2023; Greenberg et al., 2023).

Second, the Bayesian interpretation of filtering methods, commonly referred to as Bayesian filters (Särkkä and Svensson, 2023, Chapter 1), provides a probabilistic frame-

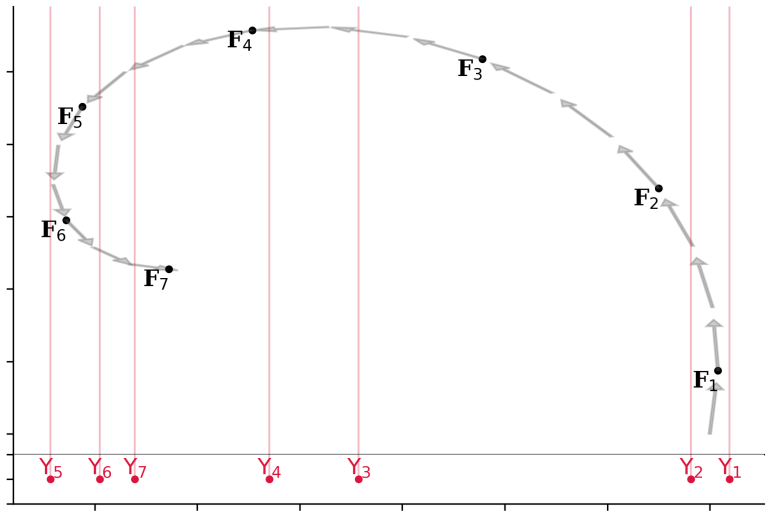

Figure 1.1: One-dimensional projection of a noisy-two dimensional dynamical system. In the top panel, the gray arrows represent the underlying evolution of the system and the black dots show the sampled (but unknown) locations of the system. In the bottom panel, the red dots show the observed measurements projected onto a one-dimensional plane. The red vertical lines denote the projection from latent space to observation space.

work for sequential decision-making. This perspective has inspired algorithms such as particle filters (Doucet et al., 2009) and variational-Bayes filters (Sarkka and Nummenmaa, 2009), which extend the applicability of Bayesian filters to non-linear and non-Gaussian systems.

Third, filtering methods are widely applicable beyond their traditional use in physical systems. Applications include financial modelling (Wells, 2013), signal processing (Basseville et al., 1993, Chapter 3), and other domains where dynamic systems must be monitored and predicted.

Motivated by these three properties, in this thesis, we advocate for a more prominent role of filtering methods in machine learning. Specifically, we leverage filtering techniques to design novel online learning algorithms that are (i) robust, (ii) adaptive, and (iii) scalable. Our approach treats the Bayesian framework as an algorithmic tool for "rationalising and formalising experience accumulation" (Peterka, 1981; Breiman, 2001). In contrast to the classical filtering perspective, we do not assume expert knowledge of the data-generating process. Instead, the resulting methods aim to maximise predictive performance by dynamically learning parametric (and potentially non-linear) models based solely on available observations and features.

The methods and frameworks developed in this thesis build upon the foundational principles discussed above, extending them to address challenges in robustness, adaptability, and scalability within online learning. The rest of this thesis is organised as follows:

Chapter 2 introduces basic concepts used throughout the thesis, including prequential forecasting and Bayes' rule as a mechanism for sequentially updating model parameters. This chapter revisits classical results in recursive statistical learning, extends these results

to non-linear observation models such as neural networks, and examines recent advances in recursive estimation.

Chapter 3 develops a framework for adaptive online learning, building on hierarchical Bayesian models and filtering methods, as outlined in Duran-Martin et al. (2025).

Chapter 4 focuses on the robustness of the methods introduced in Chapters 2 and 3 against outliers and misspecified observation models. After reviewing prior work in this area, we present a lightweight approach proposed in Duran-Martin et al. (2024), which leverages the generalised-Bayes principle by replacing the traditional log-likelihood in posterior computation with a more flexible loss function.

Chapter 5 addresses the scalability challenges of Bayesian filtering methods for online learning. Drawing from the work in Duran-Martin et al. (2022); Cartea et al. (2023b); Chang et al. (2023), we propose three novel strategies: training a linear subspace of model parameters, using low-rank posterior covariance matrices, and employing last-layer methods to separate feature transformation from observation approximation.

Finally, Chapter 6 summarises the key findings of this thesis and outlines directions for future research.

Chapter 2

Recursive Bayesian online learning

In this chapter, we approach the problem of parametric online learning from a Bayesian filtering perspective. Here, the focus is the recursive estimation of the posterior density over model parameters. We review how the Kalman filter can be viewed as a form of Bayesian online learning for linear models, and explore methods to incorporate non-linear measurement models, such as those used in neural networks.

The remainder of this chapter establishes the foundations for both sequential closed-form and fixed-point methods to compute or approximate the posterior density.

Section 2.1 introduces the problem of prequential inference and the recursive estimation of model parameters. Section 2.2 reviews the multivariate Gaussian and its properties, which we use throughout the thesis. Section 2.3 presents a recursive approach for updating model parameters assuming that both, the parameter dynamics model and the measurement model, are Gaussian and linear. This includes a recursive variant of the Ridge regression algorithm. Section 2.4 relaxes the assumptions of linear and Gaussian measurement models and presents various methods to obtain closed-form approximations of the posterior density under non-linear measurement models. Section 2.5 introduces state-space models in the context of sequential learning and demonstrates how filtering generalises probabilistic sequential learning of model parameters. Finally, Section 2.8 concludes the chapter with an outlook for the remainder of this thesis.

2.1 Recursive Bayesian inference

Consider a sequence of measurements $\boldsymbol{y}_{1:t} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$ with $\boldsymbol{y}_i \in \mathcal{Y} \subseteq \mathbb{R}^o$ and features $\boldsymbol{x}_{1:t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ with $\boldsymbol{x}_i \in \mathbb{R}^M$, where $o, M \in \mathbb{N}$. Let $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ be a datapoint and $\mathcal{D}_{1:t} = (\mathcal{D}_1, \ldots, \mathcal{D}_t)$ be the dataset at time t. We are interested in the one-step-ahead (prequential) forecast (Gama et al., 2008) for $\boldsymbol{y}_{t+1}$ conditioned on the feature $\boldsymbol{x}_{t+1}$ and

the data up to time t, i.e., $\mathcal{D}_{1:t}$. In our setting, one observes $\boldsymbol{x}_{t+1}$ just before observing $\boldsymbol{y}_{t+1}$; thus, to make a prediction about $\boldsymbol{y}_{t+1}$, we have both $\mathcal{D}_{1:t}$ and $\boldsymbol{x}_{t+1}$ at our disposal.[1]

To establish a link between the features $\boldsymbol{x}_{t+1}$ and the measurement $\boldsymbol{y}_{t+1}$, we consider a probability density of the measurement $p(\boldsymbol{y}_{t+1} \,|\, \boldsymbol{\theta}, \boldsymbol{x}_{t+1})$ such that

$$\int \boldsymbol{y}_t \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t) \mathrm{d}\boldsymbol{y}_t = h(\boldsymbol{\theta}, \boldsymbol{x}_t). \tag{2.1}$$

Here $h : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}^o$ is called the measurement function. Following a probabilistic approach, an estimate for $\boldsymbol{y}_{t+1}$, having data $\mathcal{D}_{1:t}$, features $\boldsymbol{x}_{t+1}$, and the measurement function h is given by the posterior predictive mean

$$\hat{\boldsymbol{y}}_{t+1} := \mathbb{E}_p[h(\boldsymbol{\theta}, \boldsymbol{x}_{t+1}) \,|\, \mathcal{D}_{1:t}] = \int h(\boldsymbol{\theta}, \boldsymbol{x}_{t+1}) \, p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}, \tag{2.2}$$

where $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$ is the posterior density over model parameters.

Throughout this work, the notation $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ represents the probability density of the measurement $\boldsymbol{y}_t$, given the latent (unknown) model parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ and the features $\boldsymbol{x}_t$. Similarly, $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$ represents the probability density of the model parameters $\boldsymbol{\theta}$ given the data up to time t—it reflects the belief over the model parameters after having seen t datapoints.

A natural approach to construct the posterior density $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$, whenever $\mathcal{D}_{1:T}$ arrives in a stream, i.e., one datapoint $\mathcal{D}_t$ at a time, is through Bayes' rule—suppose we have access to $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1})$, with $t < T$, and we are presented with $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$, which we model through the likelihood $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ Then,

$$\begin{aligned} p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t}) &\propto p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1}) \, p(\mathcal{D}_t \,|\, \boldsymbol{\theta}) \\ &= \underbrace{p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1})}_{\text{prior density}} \underbrace{p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)}_{\text{likelihood}}, \end{aligned} \tag{2.3}$$

where the second line in (2.3) follows from the assumption of an exogenous $\boldsymbol{x}_t$.

Given the initial condition $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:0})$, with $\mathcal{D}_{1:0} = \{\}$, recursive and closed-form estimation of (2.3) is obtained whenever $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1})$ and $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ are conjugate, i.e., the functional form of $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$ is that of $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1})$ (Robert et al., 2007, Section 3.3). As a consequence, the parameterisation over model parameters is characterised in such a way that it only depends on a set of parameters, which are recursively updated.

2.1.1 Example tasks

Here, we give some examples of machine learning tasks which can be tackled using recursive inference. We group these examples into unsupervised tasks and supervised tasks.

[1] The features $\boldsymbol{x}_{t+1}$ and measurements $\boldsymbol{y}_{t+1}$ can span different time-frames. For example, $\boldsymbol{x}_{t+1}$ can be the state of the stock market at a fixed date and $\boldsymbol{y}_{t+1}$ is the return on a stock some days into the future.

Unsupervised tasks

Unsupervised tasks involve estimating unobservable quantities of interest from the data $\mathcal{D}_{1:t}$. Below, we present three common tasks in this category.

`Segmentation` Segmentation involves partitioning the data stream into contiguous subsequences or "blocks" $\{\mathcal{D}_{1:t_1}, \mathcal{D}_{t_1+1:t_2}, \ldots\}$, where the DGP for each block is governed by a sequence of unknown functions (Barry and Hartigan, 1992). The goal is to determine the points in time when a new block begins, known as changepoints. This is useful in many applications, such as finance, where detecting changes in market trends is critical (see e.g., Arroyo et al. (2022)). In this setting, non-stationarity is assumed to be abrupt and occurring at unknown points in time. We study an example in Section 3.5.3. For a survey of segmentation methods, see e.g, Aminikhanghahi and Cook (2017); Gupta et al. (2024).

`Filtering using state-space models (SSM)` Filtering estimates an underlying latent state $\boldsymbol{\theta}_t$ that evolves over time (often representing a meaningful concept). The posterior estimate of $\boldsymbol{\theta}_t$ is computed by applying Bayesian inference to the corresponding state space model (SSM), which determines the choice of (M.1: likelihood), and how the state changes over time, through the choice of (M.3: prior). Examples include estimating the state of the atmosphere (Evensen, 1994), tracking the position of a moving object (Battin, 1982), or recovering a signal from a noisy system (Basseville et al., 1993). In this setting, non-stationarity is usually assumed to be continuous and occurring at possible time-varying rates. For a survey of filtering methods, see e.g., Chen et al. (2003).

`Segmentation using Switching state-space models (SSSM)` In this task, the modeller extends the standard SSM with a set of discrete latent variables $\psi_t \in \{1, \ldots, K\}$, which may change value at each time step according to a state transition matrix. The parameters of the rest of the DGP depend on the discrete state ψ_t. The objective is to infer the sequence of underlying discrete states that best "explains" the observed data (Ostendorf et al., 1996; Ghahramani and Hinton, 2000; Beal et al., 2001; Fox et al., 2007; Van Gael et al., 2008; Linderman et al., 2017). In this context, non-stationarity arises from the switching behaviour of the underlying discrete process.

Supervised tasks

Supervised tasks involve predicting a measurable outcome $\boldsymbol{y}_t$. Unlike unsupervised tasks, this allows the performance of the model to be assessed objectively, since we can compare the prediction to the actual observation. We present three common tasks in this category below.

`Prequential forecasting` Prequential (or one-step-ahead) forecasting (Gama et al., 2008) seeks to predict the value $\boldsymbol{y}_{t+1}$ given $\mathcal{D}_{1:t}$ and $\boldsymbol{x}_{t+1}$. This is distinct from time-series forecasting, which typically does not consider exogenous variables $\boldsymbol{x}_t$, and thus can forecast (or "roll out") many steps into the future. We study an example in Section 3.5.1. For a survey on prequential forecasting under non-stationarity, see e.g., Lu et al. (2018).

`Online continual learning (OCL)` OCL is a broad term used for learning regression or classification models online, typically with neural networks. These methods usually assume that the underlying data generating mechanism could shift. The objective of OCL methods is to train a model that performs consistently across both past and future data, rather than just focusing on future forecasting (Cai et al., 2021). The changepoints (corresponding to different "tasks") may or may not be known. This setting addresses the stability-plasticity dilemma, focusing on retaining previously learned knowledge while adapting to new tasks. We study an example of OCL for classification, when the task boundaries are not known, in Section 3.5.1. For a survey on recent methods for OCL, see e.g., Gunasekara et al. (2023).

`Contextual bandits` In contextual bandit problems, the agent is presented with features $\boldsymbol{x}_{t+1}$, and must choose an action (arm) that yields the highest expected reward (Li et al., 2010). We let $\boldsymbol{y}_{t+1} \in \mathbb{R}^A$ where $A > 2$ is the number of possible actions; this is a vector where the a-th entry contains the reward one would have obtained had one chosen arm a. Let $\boldsymbol{y}_t^{(a)}$ be the observed reward at time t after choosing arm a, i.e., the a-th entry of $\boldsymbol{y}_t$. A popular approach for choosing the optimal action (while tackling the exploration-exploitation tradeoff) at each step is Thomson sampling (TS) (Thompson, 1933), which in our setting works as follows: first, sample a parameter vector from the posterior, $\tilde{\boldsymbol{\theta}}_t$ from $p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t})$; then, greedily choose the best arm (the one with the highest expected payoff), $a_{t+1} = \arg\max_a \hat{\boldsymbol{y}}_{t+1}^{(a)}$, where $\hat{\boldsymbol{y}}_{t+1} = h(\tilde{\boldsymbol{\theta}}_t; \boldsymbol{x}_{t+1})$; and $\hat{\boldsymbol{y}}_{t+1}^{(a)}$ is the a-th entry of $\hat{\boldsymbol{y}}_{t+1}$; finally, receive a reward $\boldsymbol{y}_{t+1}^{(a_{t+1})}$. The goal is to select a sequence of arms $\{a_1, \ldots, a_T\}$ that maximises the cumulative reward $\sum_{t=1}^T \boldsymbol{y}_t^{(a_t)}$. TS for contextual bandits has been used in a number of papers, see e.g., Mellor and Shapiro (2013); Duran-Martin et al. (2022); Cartea et al. (2023a); Alami (2023); Liu et al. (2023).

2.2 The multivariate Gaussian

An important concept that we use throughout this thesis is that of the multivariate Gaussian density.

Definition 2.1. Let $\boldsymbol{x} \in \mathbb{R}^o$, $\boldsymbol{\mu} \in \mathbb{R}^o$, and $\boldsymbol{\Sigma}$ an $(o \times o)$ positive-definite matrix. The density function of a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is

$$\mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\,\pi)^{-o/2}\,|\boldsymbol{\Sigma}|^{-1/2}\,\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{2.4}$$

Gaussian densities are, in many cases, used for computational reasons. This is because it has mathematical properties that allows us to work with equations of the form (2.2) and retain a Gaussian structure. In particular, the following two propositions will be extensively used through this chapter to derive recursive updates.

Proposition 2.2. *Let $\boldsymbol{x} \in \mathbb{R}^M$ and $\boldsymbol{y} \in \mathbb{R}^o$ be two random vectors such that $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{m}, \mathbf{P})$ and $p(\boldsymbol{y} \,|\, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \,|\, \mathbf{H}\,\boldsymbol{x} + \boldsymbol{b}, \mathbf{S})$. The joint density for $(\boldsymbol{x}, \boldsymbol{y})$ is also multivariate Gaussian $p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left((\boldsymbol{x}, \boldsymbol{y}) \,|\, \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{y}}\right)$ with*

$$\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{m} & \mathbf{H}\,\boldsymbol{m} + \boldsymbol{b} \end{bmatrix}, \tag{2.5}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{y}} = \begin{bmatrix} \mathbf{P} & \mathbf{P}\,\mathbf{H}^\intercal \\ \mathbf{H}\,\mathbf{P} & \mathbf{H}\,\mathbf{P}\mathbf{H}^\intercal + \mathbf{S} \end{bmatrix}. \tag{2.6}$$

Proof. See Appendix A in Särkkä and Svensson (2023). □

Proposition 2.3. *Let $\boldsymbol{x} \in \mathbb{R}^M$ and $\boldsymbol{y} \in \mathbb{R}^o$ be two random vectors with joint density function*

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \,\middle|\, \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C}^\intercal \\ \mathbf{C} & \mathbf{B} \end{bmatrix}\right). \tag{2.7}$$

Then

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{a}, \mathbf{A}), \tag{2.8}$$

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{b}, \mathbf{B}), \tag{2.9}$$

$$p(\boldsymbol{x} \,|\, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{a} + \mathbf{C}\mathbf{B}^{-1}(\boldsymbol{y} - \boldsymbol{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^\intercal), \tag{2.10}$$

$$p(\boldsymbol{y} \,|\, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{b} + \mathbf{C}^\intercal\mathbf{A}^{-1}(\boldsymbol{x} - \boldsymbol{a}), \mathbf{B} - \mathbf{C}^\intercal\mathbf{A}\mathbf{C}). \tag{2.11}$$

Proof. See Appendix A in Särkkä and Svensson (2023). □

2.3 Linear and Gaussian measurement models

In this section, we present an algorithm to compute recursive updates whenever the prior density at time t is Gaussian, i.e., $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$, and the measurement model is linear and univariate Gaussian, i.e., $p(y_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(y_t \,|\, \boldsymbol{x}_t^\intercal\,\boldsymbol{\theta}, \beta^2)$ with known variance β^2. Here $h(\boldsymbol{\theta}, \boldsymbol{x}_t) = \boldsymbol{x}_t^\intercal\,\boldsymbol{\theta}$ and, as we will show, $\mathbb{E}_p[h(\boldsymbol{\theta}, \boldsymbol{x}_t) \,|\, \mathcal{D}_{1:t}] = h(\boldsymbol{\mu}_t, \boldsymbol{x}_t) = \boldsymbol{x}_t^\intercal\,\boldsymbol{\mu}_t$. The assumption of Gaussianity is convenient because it (i) provides a prior over each model parameter that spans the real line, (ii) preserves closed-form updates to the posterior $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$, and (iii) its first and second moments fully characterise the density.

A classical statistical model that can be interpreted having a Gaussian prior for the model parameters $p(\boldsymbol{\theta})$ and Gaussian measurement model $p(y_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ is Ridge regression. We recall the Ridge regression in the proposition below.

Proposition 2.4 (Ridge regression)**.** *Consider the dataset* $\mathcal{D}_{1:T}$ *such that* $\mathcal{D}_t = (\boldsymbol{x}_t \in \mathbb{R}^M, y_t \in \mathbb{R})$*, with measurement model* $p(y_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(y_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta}, \beta^2)$*, with* $\beta > 0$*, and and prior* $p(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta} \,|\, \mathbf{0},\, \alpha^2\, \mathbf{I}\right)$*, for* $\alpha > 0$*. Suppose* $\mathrm{Cov}\left(y_i, y_j \,|\, \boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{x}_j\right) = 0$ *for all* $i \neq j$*. Then,*

$$\hat{\boldsymbol{\mu}}_T := \mathbb{E}[\boldsymbol{\theta} \,|\, \mathcal{D}_{1:T}] = (\mathbf{X}^\intercal \, \mathbf{X} + \lambda \, \mathbf{I})^{-1} \mathbf{X}^\intercal \, \mathbf{Y} \tag{2.12}$$

with $\mathbf{X} = [\boldsymbol{x}_1^\intercal, \ldots, \boldsymbol{x}_T^\intercal]^\intercal$*,* $\mathbf{Y} = [y_1, \ldots, y_T]^\intercal$*, and* $\lambda = \beta^2 / \alpha^2$*.*

Proof. See Section 3.4.1 in Hastie et al. (2009). □

If the data $\mathcal{D}_{1:T}$ arrives in a stream, we can estimate the posterior density $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$ for $t = 1, \ldots, T$ using Bayes' rule (2.3).

Proposition 2.5 (Multivariate recursive Bayesian linear regression)**.** *Consider the dataset* $\mathcal{D}_{1:T}$ *with* $\mathcal{D}_t = (\boldsymbol{x}_t \in \mathbb{R}^M, \boldsymbol{y}_t \in \mathbb{R}^o)$*, and the measurement model* $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta}, \mathbf{R}_t)$ *with known covariance matrix* $\mathbf{R}_t$*. Suppose* $\mathrm{Cov}\left(\boldsymbol{y}_i, \boldsymbol{y}_j \,|\, \boldsymbol{\theta}\right) = 0$ *for* $i \neq q$*. Let* $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ *be the initial prior density over the model parameters. Then, the posterior density at time* t *takes the form* $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ *with*

$$\begin{aligned}
\mathbf{S}_t &= \boldsymbol{x}_t^\intercal \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t + \mathbf{R}_t, \\
\mathbf{K}_t &= \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t \mathbf{S}_t^{-1}, \\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} + \mathbf{K}_t \left(\boldsymbol{y}_t - \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1}\right), \\
\boldsymbol{\Sigma}_t &= \left(\mathbf{I} - \mathbf{K}_t \boldsymbol{x}_t\right) \boldsymbol{\Sigma}_{t-1}.
\end{aligned} \tag{2.13}$$

Proof. Following an induction argument, suppose $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t-1}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$. From proposition 2.2, the joint density for $(\boldsymbol{\theta}, \boldsymbol{y}_t)$, conditioned on $\mathcal{D}_{1:t-1}$ and $\boldsymbol{x}_t$ takes the form of a Gaussian density with mean

$$\begin{bmatrix} \boldsymbol{\mu}_{t-1}^\intercal & \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1} \end{bmatrix}^\intercal \tag{2.14}$$

and covariance matrix

$$\begin{bmatrix} \boldsymbol{\Sigma}_{t-1} & \boldsymbol{\Sigma}_{t-1} \, \boldsymbol{x}_t \\ \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} & \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t + \mathbf{R}_t \end{bmatrix}. \tag{2.15}$$

Then, by Proposition 2.3, the density for $\boldsymbol{\theta}$ conditioned on $\mathcal{D}_{1:t-1}$, $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ is Gaussian with mean and covariance matrix given by

$$\begin{aligned}
\mathbf{S}_t &= \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t + \mathbf{R}_t, \\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t \mathbf{S}_t^{-1} (\boldsymbol{y}_t - \boldsymbol{\mu}_{t-1}^\intercal \boldsymbol{x}_t), \\
\boldsymbol{\Sigma}_{t-1} &= \boldsymbol{\Sigma}_{t-1} - \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}^\intercal \boldsymbol{\Sigma}_t^{-1} \boldsymbol{x}^\intercal \boldsymbol{\Sigma}_{t-1}.
\end{aligned} \tag{2.16}$$

□

Proposition 2.6. *The estimate of the mean* $\boldsymbol{\mu}_T$ *in* (2.13) *is the exact posterior mean, i.e.,* $\boldsymbol{\mu}_T = \mathbb{E}[\boldsymbol{\theta} \,|\, \mathcal{D}_{1:T}]$*, and matches the Ridge regression estimate* (2.12) *whenever* $o = 1$*,*

$\boldsymbol{\mu}_0 = \mathbf{0}$, $\boldsymbol{\Sigma}_0 = \alpha^2\,\mathbf{I}$, *and* $\mathbf{R}_t = \beta^2$ *for all* $t \in \{1, \ldots, T\}$.

Proof. See Kelly (1990); Ismail and Principe (1996). □

Proposition 2.6 shows that, in the linear and Gaussian case, the final estimate of the recursive update of model parameters matches the *offline* batch estimate of model parameters. We summarise the recursive Bayesian linear regression method in Algorithm 1.

Algorithm 1 Pseudocode for the recursive Bayesian linear regression.

Require: $\mathcal{D}_{1:T}$, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$

1: **for** $t = 1, \ldots, T$ **do**
2: $\quad \mathbf{S}_t = \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t + \mathbf{R}_t$
3: $\quad \boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t \mathbf{S}_t^{-1} (\boldsymbol{y}_t - \boldsymbol{\mu}_{t-1}^\intercal \boldsymbol{x}_t)$
4: $\quad \boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_{t-1} - \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t^\intercal \boldsymbol{\Sigma}_t^{-1} \boldsymbol{x}_t^\intercal \boldsymbol{\Sigma}_{t-1}$
5: $\quad p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t}) = \mathcal{N}\left(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right)$
6: **end for**

The following example shows a numerical analysis of the results in Proposition 2.4 and Proposition 2.5

2.3.1 Example: recursive and batch Ridge regression

Consider the dataset $\mathcal{D}_{1:T}$ with $\mathcal{D}_t = (\boldsymbol{x}_t \in \mathbb{R}^2, y_t \in \mathbb{R})$ such that $y_t = \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta} + \boldsymbol{e}_t$, with $\boldsymbol{\theta} \in \mathbb{R}^2$ the true model parameters and $p(\boldsymbol{e}_t) = \mathcal{N}(\boldsymbol{e}_t \,|\, 0, 1)$. The left panel in Figure 2.1 shows the estimate for $\boldsymbol{\mu}_t \in \mathbb{R}^2$ along with the Ridge estimate $\hat{\boldsymbol{\mu}}_t$. The right panel in Figure 2.1 shows the root mean squared error (RMSE) evaluated over a *held-out* test set $\hat{\mathcal{D}}_{1:t}$ for the Ridge estimate and the recursive-Bayes estimate $\boldsymbol{\mu}_t$ as a function of the number of datapoints processed. We observe that the parameters of the recursive-Bayes

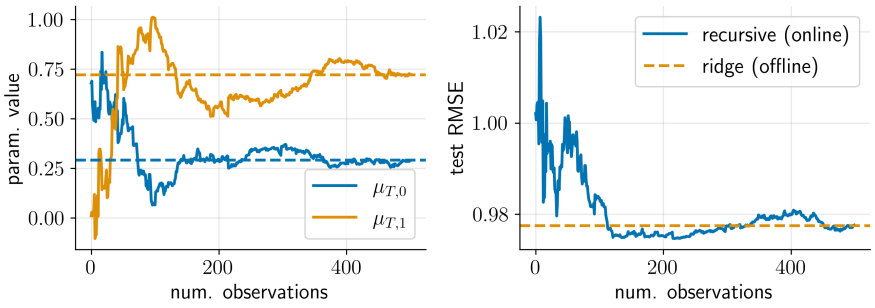

Figure 2.1: **(Left panel)** Mean estimate of the parameters. The solid lines correspond to the recursive-Bayes estimate of the mean. The dashed lines correspond to the offline estimate of the mean using Ridge regression. **(Right panel)** RMSE on a held-out test set.

estimate $\boldsymbol{\mu}_t$ tend to, and eventually match, the Ridge estimate of model parameters. As a consequence, the RMSE on the held-out set for the recursive-Bayes estimate match that of the Ridge estimate at T.

2.4 Non-Gaussian and non-linear measurement models

In some scenarios, the assumption of a linear Gaussian measurement model can be overly restrictive. For instance, when the measurement model h_t is parametrised by a neural network, h_t is a non-linear function of $\boldsymbol{\theta}$. Alternatively, in linear classification problems, the measurement model is best represented by a Bernoulli mass function, with mean $\sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_t)$, where $\sigma : \mathbb{R} \to [0, 1]$ is the sigmoid function.

In these scenarios— when the measurement model is either non-Gaussian or non-linear—the likelihood is no longer conjugate to a Gaussian prior over the model parameters. As a result, the posterior density, such as those that use (2.13), are no longer available. To address this challenge, we rely on approximations to the posterior density or Monte Carlo (MC) methods.

MC methods are commonly used to sample from (2.3) when an analytical form is either unknown or intractable. In these cases, (2.2) is approximated by the samples from the posterior. A wealth of literature exists on sample-based approaches for posterior inference. For detailed discussions, see Barbu et al. (2020) for Markov Chain Monte Carlo methods, Doucet et al. (2009) for online particle filtering, and Naesseth et al. (2019) for batch sequential Monte Carlo (SMC) techniques.

Functional approximations to the posterior, which is the primary focus in this thesis, include techniques such as variational Bayes or linearisation schemes. These computations are more tractable in many practical scenarios. We will nonetheless provide examples where sample-based methods are employed.

2.4.1 Variational Bayes

A popular approach to approximate the posterior density is through the use of variational Bayes (VB). In VB, the posterior density is typically written as an optimisation problem over a set of *candidate* densities living in a set $\mathcal{Q}$ that most closely resembles the posterior density $p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:t})$ up to a normalisation constant. In this text, our notion of closeness between the density $q \in \mathcal{Q}$ and the posterior density is based on the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) which we recall below.

Definition 2.7. (Kullback-Leibler divergence) Let p and q be probability densities defined over the same domain. Then, the KL divergence of q from p is defined as

$$\mathbf{D}_{\mathrm{KL}}\left(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})\right) := \mathbb{E}_q\left[\log\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right)\right] = \int q(\boldsymbol{\theta})\left[\log\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right)\right] \mathrm{d}\boldsymbol{\theta}. \tag{2.17}$$

See Soch et al. (2020).

A useful result that we will use throughout the thesis is that of the form of the KL divergence between two multivariate Gaussians.

Proposition 2.8. *Let $p_1(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \,|\, \boldsymbol{m}_1, \mathbf{S}_1\right)$ and $p_2(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \,|\, \boldsymbol{m}_2, \mathbf{S}_2\right)$ be two M-dimensional multivariate Gaussian densities. Then, the KL divergence between p_1 and p_2*

takes the form

$$\begin{aligned}
&\mathbf{D}_{\mathrm{KL}}\left(p_1(\boldsymbol{x}) \,||\, p_2(\boldsymbol{x})\right) \\
&= \mathbf{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x} \,|\, \boldsymbol{m}_1, \mathbf{S}_1\right) \,||\, \mathcal{N}\left(\boldsymbol{x} \,|\, \boldsymbol{m}_2, \mathbf{S}_2\right)\right) \\
&= \frac{1}{2}\left[\mathrm{Tr}\left(\mathbf{S}_2^{-1}\mathbf{S}_1\right) + (\boldsymbol{m}_2 - \boldsymbol{m}_1)^{\intercal}\mathbf{S}_2^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1) - M + \log\left(|\mathbf{S}_2|/|\mathbf{S}_1|\right)\right].
\end{aligned} \tag{2.18}$$

Proof. See Section 6.2.3 in Murphy (2022). □

Armed with a notion of closeness, we now define the objective of a VB method.

Definition 2.9. (variational approximation) Let p be a density with support $\mathrm{dom(p)}$ and $\mathcal{Q}$ a collection of candidate densities with $\mathrm{dom}(q) = \mathrm{dom(p)}$ for all $q \in \mathcal{Q}$. The variational approximation of p, under a KL divergence, selects $q_* \in \mathcal{Q}$ according to the criterion

$$q_*(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathbf{D}_{\mathrm{KL}}\left(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})\right). \tag{2.19}$$

Criterion 2.19 recovers exact Bayesian updating (2.3) if $\mathcal{Q}$ includes the Bayesian posterior within its family, i.e., $q_*(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ if $p \in \mathcal{Q}$ (Knoblauch et al., 2022).

Examples of VB for estimation of model parameters include the Bayes-by-backpropagation method (BBB) of Blundell et al. (2015), which assumes a diagonal posterior covariance (more expressive forms are also possible). Nguyen et al. (2017) extended BBB to non-stationary settings. More recent approaches involve recursive estimation, such as the recursive variational Gaussian approximation (R-VGA) method of Lambert et al. (2022) which uses a full rank Gaussian variational approximation (see Section 2.4.2); the limited-memory RVGA (L-RVGA) method of Lambert et al. (2023), which uses a diagonal plus low-rank (DLR) Gaussian variational approximation; the Bayesian online natural gradient (BONG) method of Jones et al. (2024), which combines the DLR approximation with EKF-style linearisation for additional speedups; and the natural gradient Gaussian approximation (NANO) method of Cao et al. (2024), which uses a diagonal Gaussian approximation similar to VD-EKF in Chang et al. (2022).

The following section outlines an important VB method for recursive estimation of a Gaussian density under non-linear measurement functions and non-Gaussian measurement models.

2.4.2 The recursive variational Gaussian approximation

The recursive variational Gaussian approximation (R-VGA), introduced in Lambert et al. (2022), constructs a sequence of variational Gaussian approximations. We outline this method below

Definition 2.10 (R-VGA)**.** Consider the dataset $\mathcal{D}_{1:T}$ with $\mathcal{D}_t = (\boldsymbol{x}_t \in \mathbb{R}^M, \boldsymbol{y}_t \in \mathcal{B} \subseteq \mathbb{R}^o)$. Let $q_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with prior mean $\boldsymbol{\mu}_0 \in \mathbb{R}^D$ and $(D \times D)$ covariance matrix

$\boldsymbol{\Sigma}_0$. The R-VGA method estimates a VB density for the $\boldsymbol{y}_t$ measurement according to

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathbf{D}_{\mathrm{KL}} \left(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta})\, q_{t-1}(\boldsymbol{\theta}) \,/\, Z_t \right), \tag{2.20}$$

where $\mathcal{Q}$ is the family of multivariate Gaussian densities and $Z_t = \int p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta})\, q_{t-1}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$. Because the Gaussian density is characterised by its mean and covariance matrix, the VB criteria (2.20) can be written as

$$\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathbf{D}_{\mathrm{KL}} \left(\mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,||\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta})\, \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}) \,/\, Z_t \right). \tag{2.21}$$

If the measurement model is absolutely continuous with respect to the model parameters $\boldsymbol{\theta}$ and the observations $\boldsymbol{y}_{1:T}$ are independent conditionally on $\boldsymbol{\theta}$, then the R-VGA is show to have updates for q_t (and hence $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$) as

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1}\, \mathbb{E}_{q_t}[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)], \\ \boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_t^{-1} - \mathbb{E}_{q_t}\left[\nabla^2_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)\right]. \end{aligned} \tag{2.22}$$

See Theorem 1 in Lambert et al. (2022) for a proof.

The right hand side of (2.22) requires the computation of the expected gradient and hessian of the log-likelihood according to the variational approximation at time t. In this sense, (2.22) correspond to an implicit update, i.e., the estimate of q_t is obtained after multiple inner iterations of (2.22). We summarise the R-VGA method in Algorithm 2. In Algorithm 2, the terms $\mathbb{E}_{q_t}[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)]$ and $\mathbb{E}_{q_t}[\nabla^2_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)]$ can be

Algorithm 2 Pseudocode for the R-VGA.

Require: dataset $\mathcal{D}_{1:T}$ with $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$
Require: initial state of model parameters $q_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
Require: number of iterations $I \geq 1$
for $t = 1, \ldots, T$ **do**
  // Initialise $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$
  $\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1}$
  $\boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_{t-1}$
  $q_t(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$
  for $i = 1, \ldots, I$ **do**
    $\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1}\, \mathbb{E}_{q_t}[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)]$
    $\boldsymbol{\Sigma}_t^{-1} \leftarrow \boldsymbol{\Sigma}_t^{-1} - \mathbb{E}_{q_t}[\nabla^2_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)]$
    $q_t(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$
  end for
end for

estimated through sampling—because $q_t(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta} \,|\, \mu_t,\, \boldsymbol{\Sigma}_t\right)$, then

$$\mathbb{E}_{q_t}[g(\boldsymbol{\theta})] \approx \frac{1}{S} \sum_{s=1}^{S} g\left(\boldsymbol{\theta}^{(s)}\right), \tag{2.23}$$

where $g : \mathbb{R}^D \to \mathbb{R}^K$, $K \geq 1$, $S \geq 1$ is the number of samples, and $\boldsymbol{\theta}^{(s)}$ is a sample from a multivariate Gaussian with mean $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$.

The terms in (2.22) resemble those in (2.16). In fact, the next proposition shows that R-VGA has the Bayesian recursive linear regression as a special case.

Proposition 2.11. *Consider a linear Gaussian model for $\boldsymbol{y}_t \in \mathbb{R}^o$, so that $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta}, \mathbf{R}_t)$. Then the R-VGA update (2.22) matches that of (2.13). As a consequence, it also matches the true posterior density $p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t})$.*

Proof. See Theorem 2 in Lambert et al. (2022). □

As we have seen, the R-VGA method provides a recursive approach to approximate the posterior density whenever the first- and second-order derivates of the log-measurement density are defined. In the following two experiments, we evaluate the R-VGA methods. Example 2.4.3 sequentially estimates the parameters of a logistic regression model and Example 2.4.4 sequentially estimates the parameters of a neural network to tackle a non-linear classification problem.

2.4.3 Experiment: R-VGA for logistic regression

In this experiment, we consider the problem of sequential estimation of model parameters for a binary classification problem. Assume that we observe a sequence of datapoints $\mathcal{D}_{1:T}$ with $\mathcal{D}_t = (\boldsymbol{x}_t \in \mathbb{R}^M, \boldsymbol{y}_t \in \{0, 1\})$. We model the measurements $\boldsymbol{y}_t$ as Bernoulli with mean $\sigma(h(\boldsymbol{\theta}, \boldsymbol{x}_t))$, i.e.,

$$p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = \mathrm{Bern}\left(\boldsymbol{y} \,|\, \sigma(h(\boldsymbol{\theta}, \boldsymbol{x})\right). \tag{2.24}$$

Here, $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function and $\mathrm{Bern}(y \,|\, p) = p^y \, (1 - p)^{1-y}$ is the Bernoulli probability mass function. For the logistic regression, we have $h(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{\theta}^\intercal \, \boldsymbol{x}$. As a consequence, the log-likelihood is given by

$$\log \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \boldsymbol{y}_t \, \log \sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_t) + (1 - \boldsymbol{y}_t) \, \log(1 - \sigma(\boldsymbol{\theta}^\intercal \, \boldsymbol{x}_t)). \tag{2.25}$$

To apply the R-VGA to the logistic regression model (2.24), we require to compute the first and second-order derivatives of (2.25), which can be readily done either explicitly or implicitly with the use of autodifferentiation computer libraries such as Jax (Bradbury et al., 2018), as well as sampling from the posterior density, as shown in (2.23). For this experiment, we take $p(\boldsymbol{\theta}_0) = \mathcal{N}\left(\boldsymbol{\theta} \,|\, \mathbf{0}, \mathbf{I}\right)$.

Figure 2.2 shows the decision boundaries for the logistic regression model as a function of the number of processed measurements. Next, Figure 2.3 shows the evolution of the mean estimate of the model parameters for the logistic regression and two standard deviations. We observe that the model parameters gradually convergence to the batch (offline) version of logistic regression. For details on the offline version, see Section 10.2.7 in Murphy (2022).

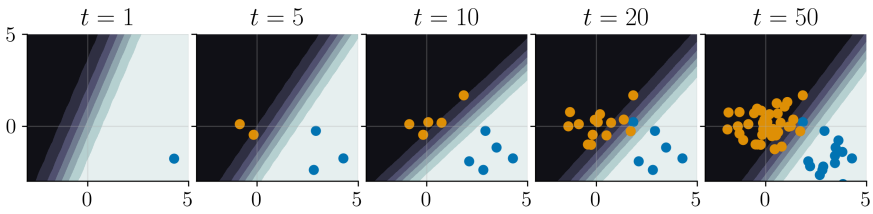

Figure 2.2: Decision boundaries for the logistic regression model as a function of the number of processed measurements.

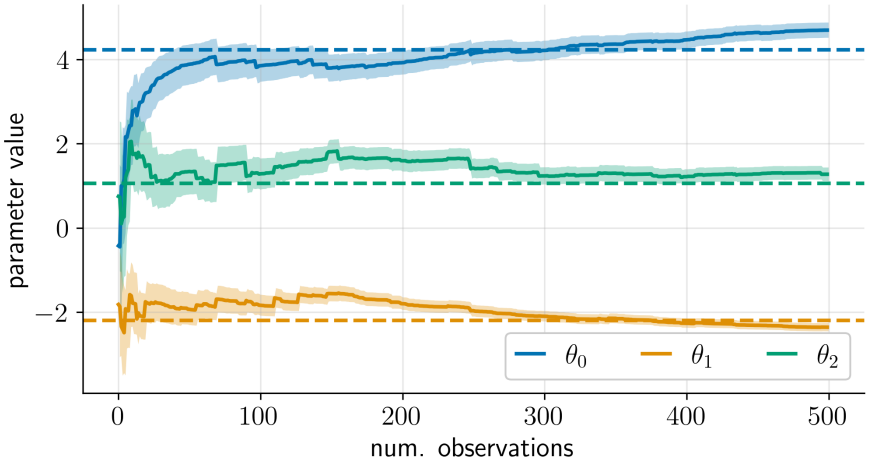

Figure 2.3: The solid lines show the posterior mean estimate of model parameters and two standard deviations. The dashed lines show the batch estimate of the logistic regression parameters.

2.4.4 Experiment: recursive learning of neural networks

Because the R-VGA update equations (2.22) require only the computation of the Jacobian and the hessian of the log-likelihood, we can apply the R-VGA methodology to any probabilistic model whose Jacobian and hessian of the log-likelihood are defined.

In this example, we consider the problem of non-linear binary classification problem. We consider a Bernoulli measurement model, whose log-likelihood is given by

$$\log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{y}_t \, \log \sigma(h(\boldsymbol{\theta}; \boldsymbol{x})) + (1 - \boldsymbol{y}_t) \, \log \left(1 - \sigma(h(\boldsymbol{\theta}; \boldsymbol{x}))\right), \tag{2.26}$$

with $h : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}$ a neural network. In particular, we consider a three-hidden-layer multi-layered perceptron with 5 units per layer and $\mathrm{leakyReLU}$ activation function. We run the R-VGA algorithm with $I = 4$ inner iterations and $S = 1,000$ samples following (2.23). To make a prediction at time t, we take the previous mean, i.e., $\boldsymbol{\mu}_{t-1}$ and use this value to make a forecast with the features $\boldsymbol{x}_t$, i.e., $\hat{\boldsymbol{y}}_t = h(\boldsymbol{\mu}_{t-1}, \boldsymbol{x}_t)$.

Figure 2.4 shows the predictions made by the decision boundary $\sigma(h(\boldsymbol{\mu}_t, \boldsymbol{x}))$ for $\boldsymbol{x} \in [-2, 2]^2$ as a function of t. The orange and blue dots represent the past datapoints and the cyan-coloured datapoints represents the datapoint observed at t.

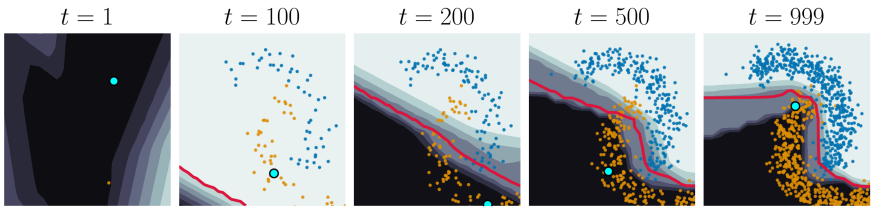

Figure 2.4: Decision boundaries for the classification problem using a neural network trained using the R-VGA. The crimson line shows the decision boundary at 0.5. The dot coloured in cyan shows the observation seen at time t.

Next, Figure 2.5 shows the expanding prequential accuracy as a function of the number of processed observations. We define the prequential accuracy as 1.0 if $\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t$ and 0.0 otherwise. For this experiment, we observe that accuracy of the R-VGA rapidly increases,

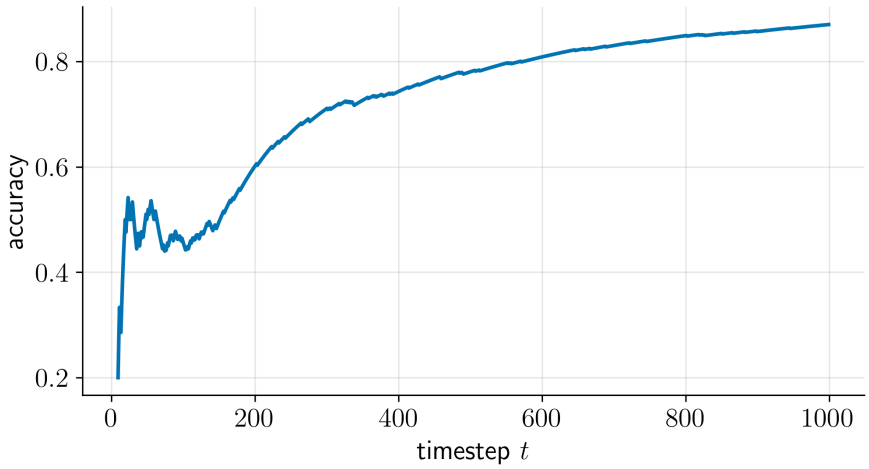

Figure 2.5: Cumulative prequential accuracy for the non-linear classification problem trained using the R-VGA.

and then plateaus for 100 steps before its predictive power starts to increase.

2.5 State-space-models for sequential supervised learning

The methods introduced in Sections 2.3 and 2.4 assume that the measurement model accurately represents the true data-generating process. This assumption allows us, given access to $\boldsymbol{x}$ and $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$, to sample random variables $\hat{\boldsymbol{y}}$ whose density is $p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x})$. As a consequence, and under certain conditions, the Bayesian posterior density can be shown to converge to a point estimate (Van der Vaart, 2000, Sec. 10.2).

However, the assumption of a well-specified likelihood may not always hold in practice. For instance, in neural network training (as discussed in Example 2.4.4), the choice of measurement model h is often driven by practical considerations rather than precise

knowledge of the true data-generating process. In these scenarios, we must adapt to the changing environment or adjust our misspecified model to make accurate predictions.

One such adaptation scheme involves the assumption of time-varying parameters. Instead of assuming a random vector $\boldsymbol{\theta}$, whose posterior density we wish to estimate, we now assume that the model parameters follow a stochastic process that evolve over time according to a *state-transition* function $f : \mathbb{R}^D \to \mathbb{R}^D$, perturbed by zero-mean dynamic noise $\boldsymbol{u}_t \in \mathbb{R}^D$ with $\mathrm{Var}(\boldsymbol{u}_t) = \mathbf{Q}_t$. This is commonly referred to as the state-space model assumption. We define a state-space model below.

Definition 2.12 (state-space model)**.** A state-space model (SSM) is a signal plus noise model of the form

$$\begin{aligned} \boldsymbol{\theta}_t &= f(\boldsymbol{\theta}_{t-1}) + \boldsymbol{u}_t, \\ \boldsymbol{y}_t &= h(\boldsymbol{\theta}_t, \boldsymbol{x}_t) + \boldsymbol{e}_t, \end{aligned} \tag{2.27}$$

with $f : \mathbb{R}^D \to \mathbb{R}^D$ the state-transition function, $h : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}^o$ the measurement function, $\mathrm{Var}(\boldsymbol{u}_t) = \mathbf{Q}_t$ a $D \times D$ positive semidefinite matrix, $\mathrm{Var}(\boldsymbol{e}_t) = \mathbf{R}_t$ a $o \times o$ positive definite matrix, and $\boldsymbol{x}_t \in \mathbb{R}^M$ exogenous features.

If one assumes zero-mean Gaussian priors with known covariance matrices for $\boldsymbol{u}_t$ and $\boldsymbol{e}_t$, the terms (2.27) can be represented as

$$\begin{aligned} p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}) &= \mathcal{N}(\boldsymbol{\theta}_t \,|\, f(\boldsymbol{\theta}_{t-1}), \mathbf{Q}_t), \\ p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t) &= \mathcal{N}(\boldsymbol{y}_t \,|\, h(\boldsymbol{\theta}_t, \boldsymbol{x}_t), \mathbf{R}_t). \end{aligned} \tag{2.28}$$

2.5.1 Linear SSMs and the Kalman filter

A well-known choice of SSM is that of linear SSMs with known dynamics covariance $\mathbf{Q}_t$, measurement covariance $\mathbf{R}_t$. In this scenario, the SSM takes the form

$$\begin{aligned} \boldsymbol{\theta}_t &= \mathbf{F}_t \, \boldsymbol{\theta}_{t-1} + \boldsymbol{u}_t, \\ \boldsymbol{y}_t &= \mathbf{H}_t \, \boldsymbol{\theta}_t + \boldsymbol{e}_t, \end{aligned} \tag{2.29}$$

where $\mathbf{F}_t$ is the $(D \times D)$ transition matrix and $\mathbf{H}_t$ is the $(o \times D)$ projection matrix.

The next proposition shows that estimation of the posterior density $p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t})$ assuming (2.29) can be done recursively following the so-called Kalman filter (KF) equations.

Proposition 2.13. *Assume an initial Gaussian prior density for model parameters* $p(\boldsymbol{\theta}_0) = \mathcal{N}(\boldsymbol{\theta}_0 \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ *and known* $\mathbf{Q}_t$, $\mathbf{R}_t$ *for all* t*. Then, the posterior predictive density for model parameters conditioned on* $\mathcal{D}_{1:t-1}$ *is Gaussian of the form*

$$p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t-1}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}\right), \tag{2.30}$$

with predictive mean and predictive covariance

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1} &= \mathbf{F}_t \, \boldsymbol{\mu}_{t-1}, \\ \boldsymbol{\Sigma}_{t|t-1} &= \mathbf{F}_t \, \boldsymbol{\Sigma}_{t-1} \, \mathbf{F}_t^\intercal + \mathbf{Q}_t. \end{aligned} \tag{2.31}$$

Furthermore, the density for model parameters, conditioned on $\mathcal{D}_{1:t}$ is Gaussian of the form

$$p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t\right), \tag{2.32}$$

with

$$\begin{aligned} \mathbf{S}_t &= \mathbf{H}_t\, \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal + \mathbf{R}_t, \\ \mathbf{K}_t &= \boldsymbol{\Sigma}_{t|t-1} \mathbf{H}_t^\intercal\, \mathbf{S}_t^{-1}, \\ \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\, (\boldsymbol{y}_t - \mathbf{H}_t\, \boldsymbol{\mu}_{t|t-1}), \\ \boldsymbol{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t^\intercal)\, \boldsymbol{\Sigma}_{t|t-1}. \end{aligned} \tag{2.33}$$

Here, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are the posterior mean and covariance matrix respectively and $\mathbf{K}_t$ is the gain matrix.

Proof. The result follows as a direct consequence of Proposition 2.2 and Proposition 2.3. For details, see Theorem 6.6 in Särkkä and Svensson (2023). □

Remark 2.14. Proposition 2.13 with $\mathbf{H}_t = \boldsymbol{x}_t^\intercal$ and $\mathbf{Q}_t = 0\,\mathbf{I}$ recovers the recursive Bayesian linear regression shown in Algorithm 1.

Algorithm 3 Predict and update steps for the Kalman filter for $t \geq 1$ and given prior mean $\boldsymbol{\mu}_{t-1}$ and covariance $\boldsymbol{\Sigma}_{t-1}$.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // datapoint
Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ // previous mean and covariance
Require: $\mathbf{H}_t$ // projection matrix
Require: $\mathbf{Q}_t, \mathbf{R}_t$ // dynamics covariance and measurement-noise covariance
// predict step
$\boldsymbol{\mu}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\mu}_{t-1}$
$\boldsymbol{\Sigma}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\Sigma}_{t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t$
//update step
$\mathbf{S}_t = \mathbf{H}_t\, \boldsymbol{\Sigma}_{t-1} \mathbf{H}_t^\intercal + \mathbf{R}_t$
$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1} \mathbf{H}_t^\intercal\, \mathbf{S}_t^{-1}$
$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \mathbf{K}_t (\boldsymbol{y}_t - \mathbf{H}_t \boldsymbol{\mu}_{t|t-1})$
$\boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t^\intercal \boldsymbol{\Sigma}_{t|t-1}$
return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$

An alternative formulation of the KF update equations is expressed through a rule that updates the posterior prediction matrix $\boldsymbol{\Sigma}_t^{-1}$. We formalise this result below.

Proposition 2.15 (Kalman filter with precision matrix updates)**.** *Assume an initial Gaussian prior density for model parameters $p(\boldsymbol{\theta}_0) = \mathcal{N}\left(\boldsymbol{\theta}_0 \,|\, \boldsymbol{\mu}_0,\, \boldsymbol{\Sigma}_0\right)$ and known $\mathbf{Q}_t$, $\mathbf{R}_t$ for all t. Then, the density for model parameters, conditioned on $\mathcal{D}_{1:t}$ is Gaussian of the form*

$$p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t\right), \tag{2.34}$$

with

$$\begin{aligned}\boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t, \\ \mathbf{K}_t &= \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}, \\ \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\, (\boldsymbol{y}_t - \mathbf{H}_t\, \boldsymbol{\mu}_{t|t-1}),\end{aligned} \tag{2.35}$$

Proof. Let $\boldsymbol{\Sigma}_{t|t-1}^{-1}$ be a precision matrix, $\mathbf{H}_t \in \mathbb{R}^{o\times D}$, and $\mathbf{R}_t$ the covariance of the measurement process. First, we show that $\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t$.

By algebraic manipulation of the of the posterior covariance (2.33), the Woodbury identity, and the definition of $\mathbf{S}_t$, it follows that

$$\begin{aligned}\boldsymbol{\Sigma}_t^{-1} &= \left((\mathbf{I}_D - \mathbf{K}_t\, \mathbf{H}_t^\intercal)\, \boldsymbol{\Sigma}_{t|t-1}\right)^{-1} \\ &= \boldsymbol{\Sigma}_{t|t-1}^{-1} \left(\mathbf{I}_D - \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal \mathbf{S}_t^{-1}\, \mathbf{H}_t\right)^{-1} \\ &= \boldsymbol{\Sigma}_{t|t-1}^{-1} \left[\mathbf{I}_D + \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal (\mathbf{S}_t - \mathbf{H}_t\, \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal)^{-1} \mathbf{H}_t\right] \\ &= \boldsymbol{\Sigma}_{t|t-1}^{-1} \left[\mathbf{I}_D + \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal (\mathbf{H}_t\, \boldsymbol{\Sigma}_{t|t-1}^{-1}\, \mathbf{H}_t^\intercal + \mathbf{R}_t - \mathbf{H}_t\, \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal)^{-1} \mathbf{H}_t\right] \\ &= \boldsymbol{\Sigma}_{t|t-1}^{-1} \left[\mathbf{I}_D + \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1} \mathbf{H}_t\right] \\ &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^\intercal \mathbf{R}_t^{-1} \mathbf{H}_t.\end{aligned}$$

Next, let $\boldsymbol{\Sigma}_t^{-1}$ be the posterior precision matrix and $\mathbf{R}_t^{-1}$ be the precision matrix of the observation at time t. The precision matrix of the innovations take the form

$$\mathbf{S}_t^{-1} = \mathbf{R}_t^{-1} - \mathbf{R}_t^{-1}\, \mathbf{H}_t\, \boldsymbol{\Sigma}_t^{-1}\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}. \tag{2.36}$$

To see this, observe that $\boldsymbol{\Sigma}_{t|t-1}^{-1} = \boldsymbol{\Sigma}_t^{-1} - \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t$, and thus

$$\begin{aligned}\mathbf{S}_t^{-1} &= \left(\mathbf{H}_t\, \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal + \mathbf{R}_t\right)^{-1} \\ &= \mathbf{R}_t^{-1} - \mathbf{R}_t^{-1} \mathbf{H}_t \left(\boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t\right)^{-1} \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} \\ &= \mathbf{R}_t^{-1} - \mathbf{R}_t^{-1} \mathbf{H}_t \left(\boldsymbol{\Sigma}_t^{-1} - \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t + \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t\right)^{-1} \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} \\ &= \mathbf{R}_t^{-1} - \mathbf{R}_t^{-1} \mathbf{H}_t \boldsymbol{\Sigma}_t^{-1}\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}.\end{aligned}$$

Finally, the Kalman gain matrix takes the form

$$\mathbf{K}_t = \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}. \tag{2.37}$$

This is because

$$\begin{aligned}\boldsymbol{\Sigma}_{t|t-1} &= \left(\boldsymbol{\Sigma}_t^{-1} - \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t\right)^{-1} \\ &= \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \left(\mathbf{R}_t - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\right)^{-1} \mathbf{H}_t \boldsymbol{\Sigma}_t,\end{aligned} \tag{2.38}$$

and, following Proposition 2.13, the Kalman gain matrix takes the form

$$
\begin{aligned}
&\mathbf{K}_t \\
&= \boldsymbol{\Sigma}_{t|t-1}\, \mathbf{H}_t^\intercal\, \mathbf{S}_t^{-1} \\
&= \left(\boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, (\mathbf{R}_t - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal)^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t\right)\, \mathbf{H}_t^\intercal\, \mathbf{S}_t^{-1} \\
&= \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + \boldsymbol{\Sigma}_t\, \mathbf{H}_t^{-1}\, \mathbf{R}_t^{-1}\left[-\mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal \mathbf{S}_t^{-1}\right].
\end{aligned}
\tag{2.39}
$$

We need to show that the second term is a zero-valued matrix. This amounts to showing that $-\mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t \mathbf{H}_t^\intercal \mathbf{S}_t^{-1} = \mathbf{0}$. To show this, note that

$$
\begin{aligned}
&-\mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{S}_t^{-1} \\
&= -\mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \left(\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t^{-1}\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\right) \\
&= -\mathbf{I}_o + (\mathbf{I}_o - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}) + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \left(\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t^{-1}\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\right) \\
&= -\mathbf{I}_o + \left[\mathbf{I}_o + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \mathbf{H}_t \boldsymbol{\Sigma}_t \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\right]\, (\mathbf{I}_o - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}) \\
&= -\mathbf{I}_o + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, \left[\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} + \mathbf{H}_t \boldsymbol{\Sigma}_t \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\right]\, (\mathbf{I}_o - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}) \\
&= -\mathbf{I}_o + (\mathbf{I}_o - \mathbf{H}_t \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1})^{-1}\, (\mathbf{I}_o - \mathbf{H}_t\, \boldsymbol{\Sigma}_t\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}) \\
&= \mathbf{0},
\end{aligned}
\tag{2.40}
$$

which concludes the proof. □

Algorithm 4 predict and update steps at time t for the Kalman filter under precision updates.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // datapoint
Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}^{-1})$ // previous mean and precision matrix
Require: $\mathbf{H}_t$ // projection matrix
Require: $\mathbf{Q}_t, \mathbf{R}_t$ // dynamics covariance and measurement-noise covariance
// predict step
$\boldsymbol{\mu}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\mu}_{t-1}$
$\boldsymbol{\Sigma}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\Sigma}_{t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t$
//update step
$\boldsymbol{\Sigma}_t^{-1} \leftarrow \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t$
$\mathbf{K}_t = \boldsymbol{\Sigma}_t \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}$
$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \mathbf{H}_t \boldsymbol{\mu}_{t|t-1})$
return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t^{-1})$

2.5.2 Experiment: the Kalman filter for non-stationary linear regression

In the classical filtering literature, the term $\mathbf{Q}_t$ is typically used to model system dynamics. However, it has long been known that inflating $\mathbf{Q}_t$ can compensate for unmodelled errors

(Kelly, 1990; Kuhl, 1990). We study this result in the context of online learning in the following experiment.

We evaluate the performance of the Kalman Filter (KF) in a linear regression problem with varying levels of $\mathbf{Q}_t = q\,\mathbf{I}$, where $q \geq 0$ is the dynamics covariance *inflation* factor. This allows us to explore how inflating $\mathbf{Q}_t$ can handle unmodelled errors in non-stationary data streams. Recall that $\mathbf{Q}_t = 0\,\mathbf{I}$ corresponds to the static online regression problem discussed in Section 2.3, while increasing $\mathbf{Q}_t$ introduces dynamics in the model parameters.

We consider a piece-wise linear regression model with standard-Gaussian errors, i.e., $p(\boldsymbol{e}_t) = \mathcal{N}(\boldsymbol{e}_t \,|\, 0, 1)$. The features are sampled according to $\boldsymbol{x}_t \sim \mathcal{U}[-2, 2]$, and the measurements are sampled according to $\boldsymbol{y}_t \sim \mathcal{N}\big(\phi(\boldsymbol{x}_t)^\intercal \boldsymbol{\theta}_t, 1\big)$ with $\phi(x) = (1,\, x,\, x^2)$. At every timestep, the parameters take the value

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_{t-1} & \text{w.p. } 1 - p_\epsilon, \\ \mathcal{U}[-3, 3]^3 & \text{w.p. } p_\epsilon, \end{cases} \tag{2.41}$$

with $p_\epsilon = 0.001$, and $\boldsymbol{\theta}_0 \sim \mathcal{U}[-3, 3]^3$. Figure 2.6 shows a sample run of this process.

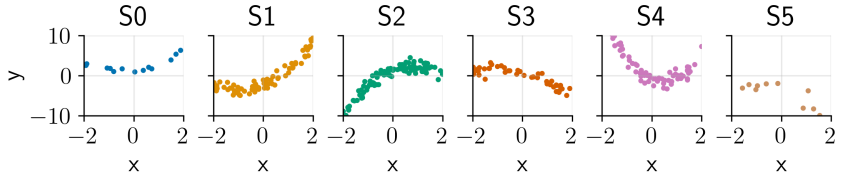

Figure 2.6: Sample run of the piecewise regression process. Each box titled Si represents the samples that belong to the i-th regime.

Given a sample run of the process $\mathcal{D}_{1:T}$, with $T = 300$, we make use of Algorithm 3 with varying levels of a fixed $\mathbf{Q}_t = q\,\mathbf{I}$ with $q \geq 0$. Figure 2.7 shows the rolling prequential RMSE and the total prequential RMSE. We observe that different values of $\mathbf{Q}_t$ lead to

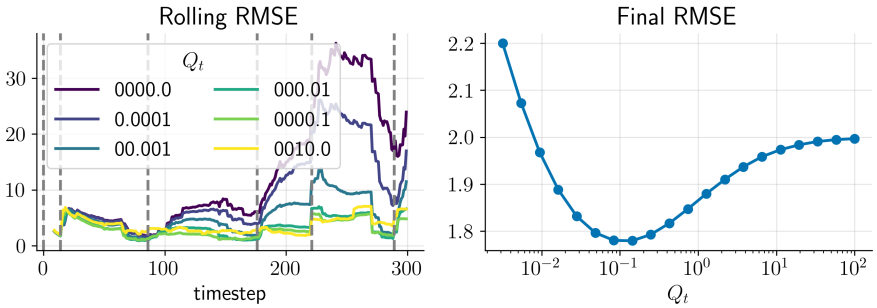

Figure 2.7: **(Left panel)** Rolling prequential RMSE of the linear model trained using various levels of $\mathbf{Q}_t$. **(Right panel)** Total prequential RMSE of the linear model

varying levels of prediction error. When $\mathbf{Q}_t$ is close to zero, the model exhibits limited adaptability, retaining more of the previous parameter estimates. Conversely, when $\mathbf{Q}_t$ is

significantly larger, the model quickly forgets past information, leading to high adaptability but potentially overreacting to noise. An ideal adaptive method would be able to adjust $\mathbf{Q}_t$ dynamically in an online fashion. We revisit this idea in Chapter 3.

2.6 Extensions to the Kalman filter

In this section, we introduce two variants of the KF that we use throughout the text; namely, the extended Kalman filter (EKF) and the ensemble Kalman filter (EnKF).

2.6.1 The extended Kalman filter

The extended Kalman filter (EKF) is a modification of the KF whenever the measurement and state functions are non-linear but differentiable w.r.t. the model parameters $\boldsymbol{\theta}$. Broadly speaking, the EKF linearises the measurement function h and the state-transition function f via a first-order Taylor approximation, which yields KF-like update equations.

The EKF has its origins in the Apollo guidance computer program to estimate the space trajectories of a spacecraft based on a "sequence of measurements of angles between selected pairs of celestial bodies, together with the measurement of the angular diameter of a nearby plane" (Battin, 1982; Grewal and Andrews, 2010). For details; see Chapter 3 in Leondes (1970). Because of the mild requirements of the EKF, it has been succesfully applied in multiple settings. Of particular interest to this thesis is the use of the EKF to train neural networks, which dates back to Singhal and Wu (1988), where it was proposed as an alternative to the backpropagation algorithm of Rumelhart et al. (1986) used to train multilayered perceptrons (MLPs).

The next proposition introduces the EKF algorithm.

Proposition 2.16. *Consider the SSM in definition 2.12 with transition function f_t and measurement function h_t, both differentiable w.r.t. $\boldsymbol{\theta}$. Define the linearised SSM*

$$\begin{aligned} \boldsymbol{\theta}_t &= \bar{f}_t(\boldsymbol{\theta}_{t-1}) + u_t, \\ \boldsymbol{y}_t &= \bar{h}_t(\boldsymbol{\theta}_t) + e_t, \end{aligned} \tag{2.42}$$

with $\bar{f}(\boldsymbol{\theta}) = \mathbf{F}_t(\boldsymbol{\theta} - \boldsymbol{\mu}_{t-1}) + f_t(\boldsymbol{\mu}_{t-1})$, $\bar{h}_t(\boldsymbol{\theta}_t) = \mathbf{H}_t(\boldsymbol{\theta}_t - \boldsymbol{\mu}_{t|t-1}) + h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t)$, $\mathbf{F}_t = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\mu}_{t-1})$, and $\mathbf{H}_t = \nabla_{\boldsymbol{\theta}} h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t)$. Consider the priors $p(\boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{u}_t \,|\, \mathbf{0}, \mathbf{Q}_t)$, $p(\boldsymbol{e}_t) = \mathcal{N}(\boldsymbol{e}_t \,|\, \mathbf{0}, \mathbf{R}_t)$.

Then, the predict and update steps are Gaussian with form

$$\begin{aligned} p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t-1}) &= \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}\right), \\ p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) &= \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right). \end{aligned} \tag{2.43}$$

With predictive mean and variance given by (2.31) and posterior mean and variance given by (2.33).

Proof. See Section 7.2 in Särkkä and Svensson (2023). □

Algorithm 5 predict and update steps for the extended Kalman filter for $t \geq 1$ and given prior mean and covariance $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // datapoint
Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ // previous mean and covariance
// predict step
$\mathbf{F}_t \leftarrow \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\mu}_{t-1})$
$\boldsymbol{\mu}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\mu}_{t-1}$
$\boldsymbol{\Sigma}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\Sigma}_{t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t$
//update step
$\mathbf{H}_t \leftarrow \nabla_{\boldsymbol{\theta}} h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t)$
$\mathbf{S}_t = \mathbf{H}_t \boldsymbol{\Sigma}_{t-1} \mathbf{H}_t^\intercal + \mathbf{R}_t$
$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1} \mathbf{H}_t^\intercal \mathbf{S}_t^{-1}$
$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \mathbf{K}_t (\boldsymbol{y}_t - \mathbf{H}_t \boldsymbol{\mu}_{t|t-1})$
$\boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t^\intercal \boldsymbol{\Sigma}_{t|t-1}$
return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$

2.6.2 The ensemble Kalman filter

The ensemble Kalman filter (EnKF) is a popular alternative to the EKF, whenever the state-space is high-dimensional. The EnKF was originally introduced as a sample-based approximation to the equations that define the Kalman filter (Evensen, 1994).

Recall that the Kalman filter update estimates the posterior mean $\boldsymbol{\mu}_t$ according to

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t (\boldsymbol{y} - \hat{\boldsymbol{y}}_t), \tag{2.44}$$

with $\mathbf{K}_t$ the Kalman gain matrix. It can be shown that under the linear SSM assumptions (2.29), the Kalman gain matrix $\mathbf{K}_t$ takes the form

$$\begin{aligned} \mathbf{K}_t &= \operatorname{Cov}(\boldsymbol{\theta}_t, \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t) \operatorname{Var}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^{-1} \\ &= \operatorname{Cov}(\boldsymbol{\theta}_t, \boldsymbol{y}_t) \operatorname{Var}(\boldsymbol{y}_t)^{-1}. \end{aligned} \tag{2.45}$$

See Ch.4 in Eubank (2005).

The EnKF propagates a bank of candidate parameters $\left\{\boldsymbol{\theta}_t^{(s)}\right\}_{s=1}^{S}$ following (2.27) and then updates each value in the ensemble according to a sample-based gain matrix. The *predict* terms are obtained by propagating

$$\begin{aligned} \boldsymbol{\theta}_{t|t-1}^{(s)} &= f\left(\boldsymbol{\theta}_{t-1}^{(s)}\right) + \boldsymbol{u}_t^{(s)}, \\ \boldsymbol{y}_{t|t-1}^{(s)} &= h\left(\boldsymbol{\theta}_{t|t-1}^{(s)}\right) + \boldsymbol{e}_t^{(s)}. \end{aligned} \tag{2.46}$$

The EnKF then updates each member in the ensemble according to

$$\boldsymbol{\theta}_t^{(s)} = \boldsymbol{\theta}_{t|t-1}^{(s)} + \hat{\mathbf{K}}_t \left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}^{(s)}\right), \tag{2.47}$$

with $\hat{\mathbf{K}}_t$ a sample-based estimate of the Kalman gain matrix that takes the form

$$\hat{\mathbf{K}}_t = \mathbf{C}_{t|t-1}\, \mathbf{V}_{t|t-1}^{-1}, \tag{2.48}$$

with

$$\mathrm{Cov}(\boldsymbol{\theta}_t, \boldsymbol{y}_t) \approx \mathbf{C}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{\theta}_{t|t-1}^{(s)} - \bar{\boldsymbol{\theta}}_{t|t-1}\right)\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)^{\intercal}, \tag{2.49}$$

and

$$\mathrm{Var}(\boldsymbol{y}_t) \approx \mathbf{V}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)^{\intercal} \tag{2.50}$$

Here, $\bar{\boldsymbol{y}}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{y}_{t|t-1}^{(s)}$ and $\bar{\boldsymbol{\theta}}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\theta}_{t|t-1}^{(s)}$. Algorithm 6 shows the predict and update steps for the EnKF.

Algorithm 6 Predict and update steps for the ensemble Kalman filter

Require: $\mathcal{D}_{1:T}$ with $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$
Require: $\{\boldsymbol{\theta}_0^{(s)}\}_{s=1}^{S}$ with $\boldsymbol{\theta}_0^{(s)} \sim \mathcal{N}(\cdot \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
for $t = 1, \ldots, T$ **do**
  // predict step
  for $s = 1, \ldots, S$ **do**
    $\boldsymbol{\theta}_{t|t-1}^{(s)} \leftarrow f\left(\boldsymbol{\theta}_{t-1}^{(s)}\right) + \boldsymbol{u}_t^{(s)}$
    $\boldsymbol{y}_{t|t-1}^{(s)} \leftarrow h\left(\boldsymbol{\theta}_{t|t-1}^{(s)}, \boldsymbol{x}_t\right) + \boldsymbol{e}_t^{(s)}$
  end for
  // build sample gain matrix
  $\mathbf{C}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{\theta}_{t|t-1}^{(s)} - \bar{\boldsymbol{\theta}}_{t|t-1}\right)\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)^{\intercal}$
  $\mathbf{V}_{t|t-1} = \frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)\left(\boldsymbol{y}_{t|t-1}^{(s)} - \bar{\boldsymbol{y}}_{t|t-1}\right)^{\intercal}$
  $\bar{\mathbf{K}}_t = \mathbf{C}_{t|t-1}\mathbf{V}_{t|t-1}^{-1}$
  // update step
  for $t = 1, \ldots, T$ **do**
    $\boldsymbol{\theta}_t^{(s)} \leftarrow \boldsymbol{\theta}_{t|t-1}^{(s)} + \bar{\mathbf{K}}_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}^{(s)})$
  end for
end for

It can be shown that under a linear SSM, the EnKF matches the KF whenever $S \to \infty$ (Evensen, 2003).

2.6.3 Exponential-family EKF

Here, we consider the modified EKF method introduced in Ollivier (2019). This method modifies the update equations in Algorithm 5 to make use of any member of the exponential family. This generalisation is helpful in scenarios where the target variables $\boldsymbol{y}_t$ cannot be reasonably modelled as Gaussians. For example when dealing with binary classification problems, in which the observations are modelled as Bernoulli, or in multi-class

classification problems, where the observations are modelled as Multinomial.

Given the measurement model $p(\boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{x}_t)$ parametrised as an exponential family with mean $h(\boldsymbol{\theta}, \boldsymbol{x}_t)$, the exponential-family extended Kalman filter (ExpfamEKF) replaces the likelihood model at time t with a Gaussian whose mean and covariance are found by matching the first two moments of the linearised log-likelihood with respect to the previous mean $\boldsymbol{\mu}_{t-1}$.

More precisely, the mean at time t is approximated by a first-order approximation around the predicted mean $\boldsymbol{\mu}_{t|t-1}$. This takes the form

$$\bar{h}_t(\boldsymbol{\theta}) := h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t) + \mathbf{H}_t \left(\boldsymbol{\theta}_t - \boldsymbol{\mu}_{t|t-1}\right). \tag{2.51}$$

Next, the measurement variance is taken as the covariance of the linearised model. For example, if $\boldsymbol{y}_t$ is modelled as a Bernoulli with mean $\sigma(h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}))$, then $\bar{\mathbf{R}}_t = \sigma(h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x})(1 - \sigma(h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}))$. Conversely, if $\boldsymbol{y}_t$ is modelled as a multivariate Gaussian with known observation variance, then $\bar{\mathbf{R}}_t = \mathbf{R}_t$.

2.6.4 Example: Recursive Learning of Neural Networks II

In this experiment we compare the performance of the ExpfamEKF method on the moons dataset presented in Section 2.4.4. We evaluate two configurations of the ExpfamEKF over 100 initialisations. The first configuration assumes static dynamics ($Q_t = 0\,\mathbf{I}$), which corresponds to the R-VGA under a linearised measurement model. The second configuration uses $Q_t = 10^{-5}\,\mathbf{I}$, accounting for unmodelled errors.

Figure 2.8 illustrates the median rolling prequential accuracy and interquartile range using a window of 50 steps and 100 different initial states. comparing the predicted and actual class labels over time.

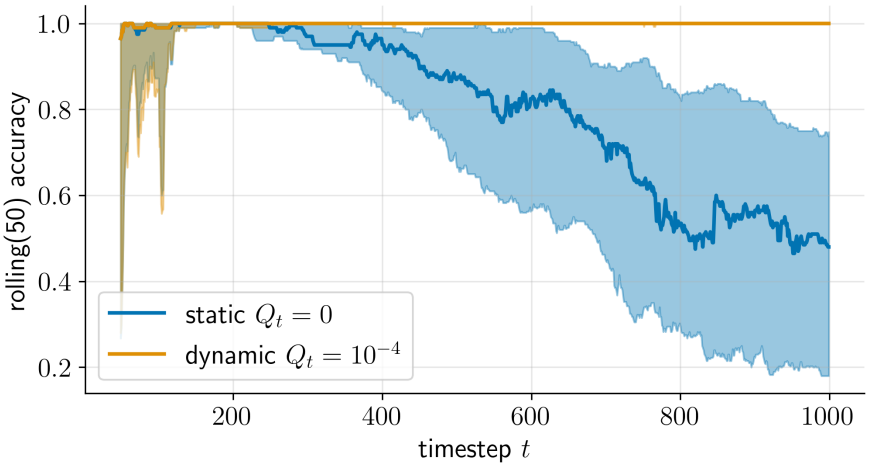

Figure 2.8: Rolling prequential accuracy for the non-linear classification problem, trained using R-VGA and ExpfamEKF.

We observe that the dynamic ExpfamEKF commences with high noise and then stabilises at a median accuracy of approximately 100%. Similarly, the static configuration exhibits high variability initially and stabilises its accuracy around 100% for 200 steps before its predictive performance begins to decline and its variability increases. This behaviour is attributed to numerical instability. This experiment highlights the importance of small random-walk noise in parameter space for maintaining numerical stability when deploying these methods online.

2.7 Alternative update methods

Alternative approaches for handling nonlinear or nonconjugate measurements have been proposed.

For instance sequential Monte Carlo (SMC) methods have been used to train neural networks (de Freitas et al., 2000). These sample-based methods are particularly advantageous when the state-transition function f is highly non-linear or when a more exact posterior approximation is required. Next, Generalised Bayesian methods, such as Mishkin et al. (2018); Knoblauch et al. (2022), generalise the VB target (2.20) to allow the likelihood to be a loss function; see Chapter 4 for further details. Alternatively, online gradient descent methods like Bencomo et al. (2023) emulate state-space modelling via gradient-based optimisation, and gradient-free methods like Goulet et al. (2021) estimate the weights of the neural network assuming a diagonal posterior covariance matrix.

2.8 Conclusion

In this chapter, we introduced the problem of Bayesian sequential estimation of model parameters from a stream of data. We laid the groundwork for the remainder of this thesis by establishing the necessary foundational concepts, and presented an array of methods that address this problem by computing or approximating the posterior density over model parameters recursively. For computational efficiency, we focused on methods that maintain a multivariate Gaussian posterior density.

We demonstrated that the problem of sequential Bayesian online linear regression can be viewed as an *online* version of the Ridge regression, as a special case of the R-VGA algorithm, and as a specific instance of the Kalman filter, where there is an absence of noise in the system dynamics.

Furthermore, we discussed how the linear assumption underlying both the Kalman filter and online linear regression can be extended to accommodate nonlinear measurement functions. Specifically, we presented three different approaches: the Extended Kalman Filter (EKF), the Ensemble Kalman Filter (EnKF), and the Recursive Variational Gaussian Approximation (R-VGA).

Chapter 3

Adaptivity

In this chapter, we propose a unifying framework for methods that perform probabilistic online learning in non-stationary environments. We call the framework BONE, which stands for generalised (B)ayesian (O)nline learning in (N)on-stationary (E)nvironments. BONE provides a common structure to tackle a variety of problems, including online continual learning, prequential forecasting, and contextual bandits.

The motivation for BONE arises from a key challenge in sequential online learning: non-stationarity in the data-generating process. While a number of methods have been proposed to address this issue, the literature remains relatively sparse. Different communities often tackle similar problems using overlapping ideas, but without a shared methodological foundation. As a result insights from one area are not easily transferable across domains.

In this chapter, we show that many of these methods, despite their apparent differences, can all be understood within a common framework that can be seen as a form of generalised Bayes posterior predictive. This insight enables a fair and principled comparison across method and lays the groundwork for developing new approaches. The framework requires specifying three modelling choices: (i) a model for measurements (e.g., a neural network), (ii) an auxiliary process to model non-stationarity (e.g., the time since the last changepoint), and (iii) a conditional prior over model parameters (e.g., a multivariate Gaussian). The framework also requires two algorithmic choices, which we use to carry out approximate inference under this framework: (i) an algorithm to estimate beliefs (posterior distribution) about the model parameters given the auxiliary variable, and (ii) an algorithm to estimate beliefs about the auxiliary variable.

A key insight provided by this framework is that most existing methods assume either abrupt changes or gradual drift in the underlying process. However, real-world scenarios often involve a combination of both. To address this, we introduce a novel method that accounts for both types of non-stationarity. Abrupt changes are modelled via the time since the last parameter reset, while slow drift is captured through an Ornstein–Uhlenbeck process over the model parameters. We evaluate this method in a range of experiments

and demonstrate its performance across diverse settings

3.1 The framework

In Chapter 2 we introduced the notion of the Bayesian posterior density to estimate, or approximate, the model parameters of the measurement function h recursively. The methods presented work well when the data-generating process is well-specified, the variational approximation family is big enough to accommodate the Bayesian posterior density, or in the case of the Kalman filter, a *good* choice of dynamics covariance $\mathbf{Q}_t$ is determined.

In practice, however, this is not often the case. Thus, to adapt to regime changes and other forms of non-stationarity, we introduce an auxiliary random variable $\psi_t \in \mathbf{\Psi}_t$, that evolves following the dynamics $p(\psi_t \mid \psi_{t-1})$ and encodes information about the non-stationarity of the sequence at time t. Here, $\mathbf{\Psi}_t$ is the set of possible values of the auxiliary variable ψ_t. The purpose of this variable is, for instance, to determine which past datapoints $\boldsymbol{y}_{1:t-1}$ most closely align with the most recent measurement $\boldsymbol{y}_t$. We describe the auxiliary variable in detail in Section 3.1.3. Finally, the model parameters $\boldsymbol{\theta}_t$ evolve following the dynamics $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \psi_t)$. This represents how much parameters change, given the state of the auxiliary variable.

Figure 3.1 shows the probabilistic graphical model that motivates our formulation; this resembles the one in Doucet et al. (2000) with an additional optional dependence between the auxiliary variable and the measurements; in what follows we omit this dependence for brevity.

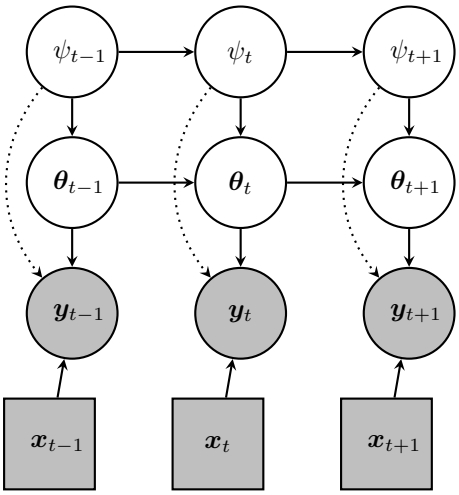

Figure 3.1: Two-levelled hierarchical state-space model (SSM) with known dynamics, motivating our BONE framework Solid arrows indicate required dependencies, while dashed arrows represent optional dependencies. Rectangles denote exogenous variables, and circles represent random variables. Observed elements are shaded in gray. The left shift in $\boldsymbol{x}_t$ represents that features are observed before observing $\boldsymbol{y}_t$.

For an experiment of length $T \in \mathbb{N}$, the joint conditional density over the model

parameters, induced by the graphical model shown in Figure 3.1, is given by

$$p(\boldsymbol{y}_{1:T}, \boldsymbol{\theta}_{0:T}, \psi_{0:T} \,|\, \boldsymbol{x}_{1:T}) = p(\boldsymbol{\theta}_0)\, p(\psi_0) \prod_{t=1}^{T} p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}, \psi_t)\, p(\psi_t \,|\, \psi_{t-1}). \tag{3.1}$$

In this chapter, we are interested in methods that efficiently compute the so-called *expected* posterior predictive $\hat{\boldsymbol{y}}_{t+1} := \mathbb{E}_{p(\boldsymbol{\theta}_t, \psi_t \,|\, \mathcal{D}_{1:t})}[h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1})]$ in an online and recursive manner. In our setting, one observes $\boldsymbol{x}_{t+1}$ just before observing $\boldsymbol{y}_{t+1}$; thus, to make a prediction about $\boldsymbol{y}_{t+1}$, we have $\boldsymbol{x}_{t+1}$ and $\mathcal{D}_{1:t}$ at our disposal.[1] For the case of a discrete auxiliary variable $\psi_t \in \Psi_t$, the form of the expected posterior predictive for $\boldsymbol{y}_{t+1}$, induced by (3.1), is

$$\begin{aligned} \hat{\boldsymbol{y}}_{t+1} &= \mathbb{E}_{p(\boldsymbol{\theta}_t, \psi_t \,|\, \mathcal{D}_{1:t})}[h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1})] \\ &= \sum_{\psi_t \in \Psi_t} \int h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1})\, p(\boldsymbol{\theta}_t, \psi_t \,|\, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t \\ &= \sum_{\psi_t \in \Psi_t} p(\psi_t \,|\, \mathcal{D}_{1:t}) \int h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1}) p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t, \end{aligned} \tag{3.2}$$

where

$$p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t}) \propto p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1}), \tag{3.3}$$

$$p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1}) = \int p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}, \psi_t)\, p(\boldsymbol{\theta}_{t-1} \,|\, \mathcal{D}_{1:t-1})\, \mathrm{d}\boldsymbol{\theta}_{t-1}, \tag{3.4}$$

$$p(\psi_t \,|\, \mathcal{D}_{1:t}) = p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \psi_t, \mathcal{D}_{1:t-1}) \sum_{\psi_{t-1} \in \boldsymbol{\Psi}_{t-1}} p(\psi_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(\psi_t \,|\, \psi_{t-1}, \mathcal{D}_{1:t-1}). \tag{3.5}$$

From (3.2), (3.3), (3.4), and (3.5) we argue that there are three key modelling choices and two algorithmic choices. Specifically, the three key modelling choices are: (M.1) the conditional mean $h(\boldsymbol{\theta}, \boldsymbol{x})$ together with the likelihood $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x})$; (M.2) the auxiliary variable ψ_t; and (M.3) the conditional prior $p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1})$. Additionally, the two algorithmic choices are: (A.1) the algorithm to compute (or approximate) the conditional posterior over model parameters $p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t})$, and (A.2) the algorithm that computes (or approximates) the posterior over weights $p(\psi_t, \,|\, \mathcal{D}_{1:t})$.

The BONE framework generalises these choices, allowing for greater flexibility while maintaining the motivating probabilistic structure. Instead of the likelihood model $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)$ with conditional mean $h(\boldsymbol{\theta}_t, \boldsymbol{x}_t)$, we consider a general function $\exp(-\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t))$, where $\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t)$ could be either a loss function or a log-likelihood. Next, instead of the conditional prior $p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1})$, we introduce a more general modelling function,

[1] The input features $\boldsymbol{x}_{t+1}$ and output measurements $\boldsymbol{y}_{t+1}$ can correspond to different time steps. For example, $\boldsymbol{x}_{t+1}$ can be the state of the stock market at a fixed date and $\boldsymbol{y}_{t+1}$ is the return on a stock some days into the future.

$\pi(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t-1})$ that governs the prior over model parameters.[2] Similarly, instead of the posterior density $p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t})$, we employ the function $q(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t})$; e.g., an approximation of the posterior, or a generalised posterior (Bissiri et al., 2016). Finally, instead of the posterior over weights $p(\psi_t \,|\, \mathcal{D}_{1:t})$, we consider a weighting function $\nu(\psi_t; \mathcal{D}_{1:t})$, which can be the Bayesian posterior or an ad-hoc time-dependent weighting function.

This generalisation is important because it unifies a wide range of existing methods under a common framework. Many well-known approaches in the literature can be written as elements of BONE by appropriately selecting the model for measurements, the conditional prior, and the posterior approximations. BONE highlights connections between different methods, it also enables systematic comparisons under a common umbrella, and it allows us to develop novel algorithms. Table 3.1 explicitly contrasts the choices in BONE with those in the classical Bayesian formalism.

component	BONE	Bayes
(M.1: likelihood)	$h(\boldsymbol{\theta}_t, \boldsymbol{x}_t)$ & $\exp(-\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t))$	$h(\boldsymbol{\theta}_t, \boldsymbol{x}_t)$ & $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)$
(M.2: auxvar)	ψ_t	ψ_t
(M.3: prior)	$\pi_t(\boldsymbol{\theta}_t;\, \psi_t) := \pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1})$	$p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1})$
(A.1: posterior)	$q_t(\boldsymbol{\theta}_t; \psi_t) := q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t})$	$p(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t})$
(A.2: weighting)	$\nu_t(\psi_t) := \nu(\psi_t;\, \mathcal{D}_{1:t})$	$p(\psi_t \,|\, \mathcal{D}_{1:t})$

Table 3.1: Components of the BONE framework.

With these modifications, the expected posterior predictive under BONE is

$$\hat{\boldsymbol{y}}_t := \sum_{\psi_t \in \Psi_t} \underbrace{\nu(\psi_t \,|\, \mathcal{D}_{1:t})}_{\text{(A.2: weighting)}} \int \underbrace{h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1})}_{\text{(M.1: likelihood)}} \underbrace{q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t})}_{\text{(A.1: posterior)}} \mathrm{d}\boldsymbol{\theta}_t, \tag{3.6}$$

where

$$q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \propto \underbrace{\pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1})}_{\text{(M.3: prior)}} \underbrace{\exp(-\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t))}_{\text{(M.1: likelihood)}} \tag{3.7}$$

takes the form of a generalised posterior (Bissiri et al., 2016). In classical Bayesian setting, the loss function takes the form of the negative log-likelihood, i.e.,

$$\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t) = -\log p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t). \tag{3.8}$$

Unless stated otherwise, we work with the negative log-likelihood in (3.8) found in the classical Bayesian setting.

A prediction for $\boldsymbol{y}_{t+1}$ given $\mathcal{D}_{1:t}$, $\boldsymbol{x}_{t+1}$, and ψ_t is

$$\hat{\boldsymbol{y}}_{t+1}^{(\psi_t)} = \mathbb{E}_{q_t}[h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \,|\, \psi_t] := \int h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1})\, q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t\,. \tag{3.9}$$

Here we use the shorthand notation $q_t = q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t})$ and $\mathbb{E}_{q_t}[\cdot \,|\, \psi_t]$ to highlight de-

[2] This function adopts an ad hoc approach to parameter evolution instead of explicitly solving the integration step (3.4).

pendence on ψ_t.

Algorithm 7 provides pseudocode for the prediction and update steps in the BONE framework. Notably, these components can be broadly divided into two categories: modelling and algorithmic. The modelling components determine the inductive biases in the model, and correspond to h, ℓ, ψ_t, and π. The algorithmic components dictate how operations are carried out to produce a final prediction — this corresponds to q_t and ν_t.

Algorithm 7 Generic predict and update step of BONE with discrete ψ_t at time t.

Require: $\mathcal{D}_{1:t}$ // past data
Require: $\boldsymbol{x}_{t+1}$ // optional inputs
Require: $h(\boldsymbol{\theta}, \boldsymbol{x}_t)$ // Choice of (M.1: likelihood)
Require: $\boldsymbol{\Psi}_t$ // Choice of (M.2: auxvar)
for $\psi_t \in \boldsymbol{\Psi}_t$ **do**
  $\pi_t(\boldsymbol{\theta}_t;\, \psi_t) \leftarrow \pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1})$ // choice of (M.3: prior)
  $q_t(\boldsymbol{\theta}_t;\, \psi_t) \leftarrow q(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \propto \pi_t(\boldsymbol{\theta}_t\ \psi_t)\, \exp(-\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t))$// choice of (A.1: posterior)
  $\nu_t(\psi_t) \leftarrow \nu(\psi_t;\, \mathcal{D}_{1:t})$ // choice of (A.2: weighting)
  $\hat{\boldsymbol{y}}_{t+1}^{(\psi_t)} \leftarrow \mathbb{E}_{q_t}[h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1});\, \psi_t]$ // conditional prequential prediction
end for
$\hat{\boldsymbol{y}}_{t+1} \leftarrow \sum_{\psi_t} \nu_t(\psi_t)\, \hat{\boldsymbol{y}}_{t+1}^{(\psi_t)}$ // weighted prequential prediction

3.1.1 Details of BONE

In the following subsections, we describe each component of the BONE framework in detail, provide illustrative examples, and reference relevant literature for further reading.

3.1.2 The measurement model (M.1)

Recall that $h(\boldsymbol{\theta}, \boldsymbol{x})$ is a parametric model that encodes the conditional mean for $\boldsymbol{y}$, given $\boldsymbol{\theta}$ and $\boldsymbol{x}$. For linear measurement models, $h(\boldsymbol{\theta}, \boldsymbol{x})$ is given by:

$$h(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{cases} \boldsymbol{\theta}^\intercal \boldsymbol{x} & \text{(regression)}, y \in \mathbb{R} \\ \sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}) & \text{(binary classification)}, y \in \{0, 1\} \\ \mathrm{Softmax}(\boldsymbol{\theta}^\intercal \boldsymbol{x}) & \text{(multi-class classification)}, \boldsymbol{y} \in \{0, 1\}^C \end{cases} \tag{3.10}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function, $C \in \mathbb{N}$ is the number of classes, $\mathrm{Softmax}(\boldsymbol{z})_k = \exp(\boldsymbol{z}_k)/\sum_i \exp(\boldsymbol{z}_i)$ represents the softmax function with $\boldsymbol{z} \in \mathbb{R}^o$ and $\boldsymbol{z}_i$ the i-th element of $\boldsymbol{z}$. In the machine learning literature, the vector $\boldsymbol{z}$ is called the logits of the classifier. For non-linear measurement models, such as neural networks, $h(\boldsymbol{\theta}, \boldsymbol{x})$ represents the output of the network parameterised by $\boldsymbol{\theta}$. The best choice of h will depend on the nature of the data, as well as the nature of the task, in particular, whether it is supervised or unsupervised. We give some examples in Section 3.5.

3.1.3 The auxiliary variable (M.2)

The choice of auxiliary variable ψ_t is crucial to identify changes in the data-generating process, allowing our framework to track non-stationarity. Below, we give a list of the common auxiliary variables used in the literature.

RL (runlength): $\psi_t = r_t \in \{0, \ldots, t\}$ is a scalar representing a *lookback window*, defined as the number of steps since the last regime change. The value $r_t = 0$ indicates the start of a new regime at time t, while $r_t \geq 1$ denotes the continuation of a regime with a lookback window of length r_t. This choice of auxiliary variable is common in the changepoint detection literature. See e.g., Adams and MacKay (2007); Knoblauch et al. (2018); Alami et al. (2020); Agudelo-España et al. (2020); Altamirano et al. (2023c); Alami (2023). This auxiliary variable is useful for non-stationary data with non-repeating temporal segments, provided we know the intensity with which new segments appear.

RLCC (runlength and changepoint count): $\psi_t = (r_t, c_t) \in \{0, \ldots, t\} \times \{0, \ldots, t\}$ is a vector that represents both the runlength and the total number of changepoints, as proposed in Wilson et al. (2010). When $r_t = t$, this implies $c_t = 0$, meaning no changepoints have occurred. Conversely, $r_t = 0$ indicates the start of a new regime and implies $c_t \in \{1, \ldots, t\}$, accounting for at least one changepoint. For a given $r_t \geq 0$, the changepoint count c_t belongs to the range $\{1, \ldots, t - r_t\}$. As with RL, this auxiliary variable assumes consecutive time blocks, but additionally allows us to estimate the likelihood of entering a new regime by tracking the number of changepoints seen so far. This auxiliary variable is useful for non-stationary data with non-repeating temporal segments when the intensity with which new segments appear is unknown.

CPT (changepoint timestep): $\psi_t = \boldsymbol{\zeta}_t$, with $\boldsymbol{\zeta}_t = \{\zeta_{1,t}, \ldots, \zeta_{\ell,t}\}$, is a set of size $\ell \in \{0, \ldots t\}$ containing the ℓ times at which there was a changepoint, with the convention that $0 \leq \zeta_{1,t} < \zeta_{2,t} < \ldots < \zeta_{\ell,t} \leq t$. This choice of auxiliary variable was introduced in Fearnhead and Liu (2007) and has been studied in Fearnhead and Liu (2011); Fearnhead and Rigaill (2019). Under mild assumptions, it can be shown that CPT is equivalent to RL, see e.g., Knoblauch and Damoulas (2018). This auxiliary variable is useful for non-stationary data with non-repeating temporal segments when the probability of a new segment appearing is unknown and knowledge of the changepoint location is required.

CPL (changepoint location): $\psi_t = s_{1:t} \in \{0, 1\}^t$ is a binary vector. In one interpretation, $s_i = 1$ indicates the occurrence of a changepoint at time i, as in Li et al. (2021), while in another, it means that $\mathcal{D}_t$ belongs to the current regime, as in Nassar et al. (2022). This auxiliary variable is useful for non-stationary data with repeating temporal segments. It is useful when the segments are formed of non-consecutive datapoints.

CPV (changepoint probability vector): $\psi_t = v_{1:t} \in (0, 1)^t$ is a t-dimensional random vector representing the probability of each element in the history belonging to the current regime. This generalises CPL and was introduced in Nassar et al. (2022) for online continual learning, allowing for a more fine-grained representation of changepoints over time. This auxiliary variable is useful for non-stationary data with repeating temporal segments. Unlike CPL, it takes a vector of weights in $(0, 1)$ which allows for higher flexibilty when

compared to CPL.

CPP (changepoint probability): $\psi_t = v_t \in (0, 1)$ represents the probability of a changepoint. This is a special case of CPV that tracks only the most recent changepoint probability; this choice was used in Titsias et al. (2024) for online continual learning.

ME (mixture of experts): $\psi_t = \alpha_t \in \{1, \ldots, K\}$ represents one of K experts. Each expert corresponds to either a choice of model or one of K possible hyperparameters. This approach has been applied to filtering (Chaer et al., 1997) and prequential forecasting (Liu, 2023; Abélès et al., 2024). This auxiliary variable facilitates the weighting of predictions made by models when one has a fixed number of competing models.

C: $\psi_t = c$ represents a constant auxiliary variable, where c is just a placeholder or dummy value. This is equivalent to not having an auxiliary variable, or alternatively, to having a single expert that encodes all available information.

Space-time complexity There is a tradeoff between the complexity that ψ is able to encode and the computation power needed to perform updates. Loosely speaking, this can be seen in the cardinality of the set of possible values of ψ through time. Let $\mathbf{\Psi}_t$ be the space of possible values for ψ_t. Depending on the choice of ψ_t, the cardinality of $\mathbf{\Psi}_t$ either stay constant or increase over time, i.e., $\mathbf{\Psi}_{t-1} \subseteq \mathbf{\Psi}_t$ for all $t = 1, \ldots, T$. For instance, the possible values for RL increase by one at each timestep; the possible values of CPL double at each timestep; finally, the possible values for ME do not increase. Table 3.2 shows the space of values and cardinality that $\mathbf{\Psi}_t$ takes as a function of the choice of auxiliary variable.

name	C	CPT	CPP	CPL	CPV	ME	RL	RLCC
values	$\{c\}$	$2^{\{0,1,\ldots,t\}}$	$[0, 1]$	$\{0, 1\}^t$	$(0, 1)^t$	$\{1, \ldots, K\}$	$\{0, 1, \ldots, t\}$	$\{\{0, t\}, \ldots, \{t, 0\}\}$
cardinality	1	2^t	∞	2^t	inf	K	t	$2 + t(t + 1)/2$

Table 3.2: Design space for the auxiliary random variables ψ_t. Here, T denotes the total number of timesteps and K denotes a fixed number of candidates.

3.1.4 Conditional prior (M.3)

This component defines the prior predictive distribution over model parameters conditioned on the choice of (M.2: auxvar) ψ_t and the dataset $\mathcal{D}_{1:t-1}$. In some cases, explicit access to past data is not needed.

For example, a common assumption is to have a Gaussian conditional prior over model parameters. In this case, we assume that, given data $\mathcal{D}_{1:t-1}$ and the auxiliary variable ψ_t, the conditional prior takes the form

$$\pi(\boldsymbol{\theta}_t;\, \psi_t,\, \mathcal{D}_{1:t-1}) = \mathcal{N}\big(\boldsymbol{\theta}_t \,|\, g_t(\psi_t, \mathcal{D}_{1:t-1}), G_t(\psi_t, \mathcal{D}_{1:t-1})\big), \tag{3.11}$$

with $g_t : \mathbf{\Psi}_t \times \mathbb{R}^{(D+o)(t-1)} \to \mathbb{R}^D$ a function that returns the mean vector of model parameters, $\mathbb{E}[\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1}]$, and $G_t : \mathbf{\Psi}_t \times \mathbb{R}^{(D+o)(t-1)} \to \mathbb{R}^{D \times D}$ a function that returns a D-dimensional covariance matrix, $\mathrm{Cov}[\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1}]$. In what follows, we let

$(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ be the pre-defined initial prior mean and covariance. Furthermore, we denote $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ be the posterior mean and covariance found at time $t-1$, which is used as a prior at time t.

Below, we provide a non-exhaustive list of possible combinations of choices for (M.2: auxvar) and (M.3: prior) of the form (3.11) that can be found in the literature, and we also introduce a new combination.

C-LSSM (constant linear with affine state-space model). We assume the parameter dynamics can be modeled by a linear-Gaussian state space model (LSSM), i.e., $\mathbb{E}[\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}] = \mathbf{F}_t\, \boldsymbol{\theta}_{t-1} + \boldsymbol{b}_t$ and $\mathrm{Cov}[\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}] = \mathbf{Q}_t$, for given $(D \times D)$ dynamics matrix $\mathbf{F}_t$, $(D \times 1)$ bias vector $\boldsymbol{b}_t$, and $(D \times D)$ positive semi-definite matrix $\mathbf{Q}_t$. We also assume $\psi_t = c$ is a fixed (dummy) constant, which is equivalent to not having an auxiliary variable. The characterisation of the conditional prior takes the form

$$\begin{aligned} g_t(c, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t\, \boldsymbol{\mu}_{t-1} + \boldsymbol{b}_t, \\ G_t(c, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t\, \boldsymbol{\Sigma}_{t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t\,, \end{aligned} \tag{3.12}$$

This is a common baseline model that we will specialise below.

C-OU (constant with Ornstein-Uhlenbeck process). This is a special case of the C-LSSM model where $\mathbf{F}_t = \gamma \mathbf{I}$, $\boldsymbol{b}_t = (1-\gamma)\boldsymbol{\mu}_0$, $\mathbf{Q}_t = (1-\gamma^2)\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_0 = \sigma_0^2 \mathbf{I}$, $\gamma \in [0, 1]$ is the fixed rate, and $\sigma_0 \geq 0$. The conditional prior mean and covariance are a convex combination of the form

$$\begin{aligned} g(c, \mathcal{D}_{1:t-1}) &= \gamma \boldsymbol{\mu}_{t-1} + (1-\gamma)\boldsymbol{\mu}_0, \\ G(c, \mathcal{D}_{1:t-1}) &= \gamma^2 \boldsymbol{\Sigma}_{t-1} + (1-\gamma^2)\boldsymbol{\Sigma}_0. \end{aligned} \tag{3.13}$$

This combination is used in Kurle et al. (2019). Smaller values of the rate parameter γ correspond to a faster resetting, i.e., the distribution of model parameters revert more quickly to the prior belief $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, which means the past data will be forgotten.

CPP-OU (changepoint probability with Ornstein-Uhlenbeck process). Here $\psi_t = \upsilon_t \in [0, 1]$ is the changepoint probability that we use as the rate of an Ornstein-Uhlenbeck (OU) process, as proposed in Titsias et al. (2024); Galashov et al. (2024). The characterisation of the conditional prior takes the form

$$\begin{aligned} g(\upsilon_t, \mathcal{D}_{1:t-1}) &= \upsilon_t \boldsymbol{\mu}_{t-1} + (1-\upsilon_t)\boldsymbol{\mu}_0\,, \\ G(\upsilon_t, \mathcal{D}_{1:t-1}) &= \upsilon_t^2 \boldsymbol{\Sigma}_{t-1} + (1-\upsilon_t^2)\boldsymbol{\Sigma}_0\,. \end{aligned} \tag{3.14}$$

An example on how to compute υ_t using an empirical Bayes procedure is given in (3.45).

C-ACI (constant with additive covariance inflation). This corresponds to a special case of C-LSSM in which $\mathbf{F} = \mathbf{I}$, $\boldsymbol{b} = \mathbf{0}$, and $\mathbf{Q} = \alpha \mathbf{I}$ for $\alpha > 0$ is the amount of noise added at each step. This combination is used in Kuhl (1990); Duran-Martin et al. (2022); Chang et al. (2022, 2023) . The characterisation of the conditional prior takes the form

$$\begin{aligned} g(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{t-1}, \\ G(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{t-1} + \mathbf{Q}_t. \end{aligned} \tag{3.15}$$

This is similar to C-OU with $\gamma = 1$, however, here we inject new noise at each step. Another variant of this scheme, known as *shrink-and-perturb* (Ash and Adams, 2020), takes $g(c, \mathcal{D}_{1:t-1}) = q\,\boldsymbol{\mu}_{t-1}$ and $G(c, \mathcal{D}_{1:t-1}) = \boldsymbol{\Sigma}_{t-1} + \mathbf{Q}_t$, where $0 < q < 1$ is the shrinkage parameters, and $\mathbf{Q}_t = \sigma_0^2\,\mathbf{I}$.

C-Static (constant with static parameters). Here $\psi_t = c$ (with c a dummy variable). This is a special case of the C-ACI configuration in which $\alpha = 0$. The conditional prior is characterised by

$$\begin{aligned} g_t(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{t-1}, \\ G_t(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{t-1}. \end{aligned} \tag{3.16}$$

ME-LSSM (mixture of experts with LSSM). Here $\psi_t = \alpha_t \in \{1, \ldots, K\}$, and we have a bank of K independent LSSM models; the auxiliary variable specifies which model to use at each step. The characterisation of the conditional prior takes the form

$$\begin{aligned} g_t(\alpha_t, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t^{(\alpha_t)}\,\boldsymbol{\mu}_{t-1}^{(\alpha_t)} + \boldsymbol{b}_t^{(\alpha_t)}\,, \\ G_t(\alpha_t, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t^{(\alpha_t)}\,\boldsymbol{\Sigma}_{t-1}^{(\alpha_t)}\mathbf{F}_t^{\intercal} + \mathbf{Q}_t^{(\alpha_t)}\,. \end{aligned} \tag{3.17}$$

The superscript (α_t) denotes the conditional prior for the k-th expert. More precisely, $\boldsymbol{\mu}_{t-1}^{(\alpha_t)}, \boldsymbol{\Sigma}_{t-1}^{(\alpha_t)}$ are the posterior at time $t-1$ using $\mathbf{F}_{t-1}^{(\alpha_t)}$ and $\mathbf{Q}_{t-1}^{(\alpha_t)}$ from the k-th expert. This combination was introduced in Chaer et al. (1997).

RL-PR (runlength with prior reset): for $\psi_t = r_t$, this choice of auxiliary variable constructs a new mean and covariance considering the past $t - r_t$ observations. We have

$$\begin{aligned} g_t(r_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_0\,\mathbb{1}(r_t = 0) + \boldsymbol{\mu}_{(r_{t-1})}\mathbb{1}(r_t > 0), \\ G_t(r_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_0\,\mathbb{1}(r_t = 0) + \boldsymbol{\Sigma}_{(r_{t-1})}\mathbb{1}(r_t > 0), \end{aligned} \tag{3.18}$$

where $\boldsymbol{\mu}_{(r_{t-1})}, \boldsymbol{\Sigma}_{(r_{t-1})}$ denotes the posterior belief computed using observations from indices $t - r_t$ to $t - 1$. The case $r_t = 0$ corresponds to choosing the initial pre-defined prior mean and covariance $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. This combination assumes that data from a single regime arrives in sequential *blocks* of time of length r_t. This choice of (M.3: prior) was first studied in Adams and MacKay (2007).

RL[1]-OUPR* (greedy runlength with OU and prior reset): This is a new combination we consider in this paper, which is designed to accommodate both gradual changes and sudden changes. More precisely, we assume $\psi_t = r_t$, and we choose the conditional prior as either a hard reset to the prior, if $\nu_t(r_t) > \varepsilon$, or a convex combination of the prior and the previous belief state (using an OU process), if $\nu_t(r_t) \leq \varepsilon$. That is, we define the conditional prior as

$$g_t(r_t, \mathcal{D}_{1:t-1}) = \begin{cases} \boldsymbol{\mu}_0\,(1 - \nu_t(r_t)) + \boldsymbol{\mu}_{(r_t)}\,\nu_t(r_t) & \nu_t(r_t) > \varepsilon, \\ \boldsymbol{\mu}_0 & \nu_t(r_t) \leq \varepsilon, \end{cases} \tag{3.19}$$

$$G_t(r_t, \mathcal{D}_{1:t-1}) = \begin{cases} \boldsymbol{\Sigma}_0 \left(1 - \nu_t(r_t)^2\right) + \boldsymbol{\Sigma}_{(r_t)} \, \nu_t(r_t)^2 & \nu_t(r_t) > \varepsilon, \\ \boldsymbol{\Sigma}_0 & \nu_t(r_t) \leq \varepsilon. \end{cases} \tag{3.20}$$

Here $\nu_t(r_t) = p(r_t \,|\, \mathcal{D}_{1:t})$, with $r_t = r_{t-1} + 1$, is the probability we are continuing a segment, and $\nu_t(r_t)$ with $r_t = 0$ is the probability of a changepoint. For details on how to compute $\nu_t(r_t)$, see (3.40). The value of the threshold parameter ε controls whether an abrupt change or a gradual change should take place. In the limit when $\varepsilon = 1$, this new combination does not learn, since it is always doing a hard reset to the initial beliefs at time $t = 0$. Conversely, when $\varepsilon = 0$, we obtain an OU-type update weighted by ν_t. When $\varepsilon = 0.5$, we revert back to prior beliefs when the most likely hypothesis is that a changepoint has just occurred. Finally, we remark that the above combination allows us to make use of non-Markovian choices for (M.1: likelihood), as we see in Section 3.5.3. This is, to the best of our knowledge, a new combination that has not been proposed in the previous literature; for further details see Appendix 3.3.3.

CPL-Sub (changepoint location with subset of data): for $\psi_t = s_{1:t}$, this conditional prior constructs the mean and covariance as

$$\begin{aligned} g_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{(s_{1:t-1})}, \\ G_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{(s_{1:t-1})}, \end{aligned} \tag{3.21}$$

where $\boldsymbol{\mu}_{(s_{1:t-1})}, \boldsymbol{\Sigma}_{(s_{1:t-1})}$ denotes the posterior belief computed using the observations for entries where $s_{1:t-1}$ have value of 1. This combination assumes that data from the current regime is scattered from the past history. That is, it assumes that data from a past regime could become relevant again at a later date. This combination has been studied in Nguyen et al. (2017).

CPL-MCI (changepoint location with multiplicative covariance matrix): for $\psi_t = s_{1:t}$, this choice of conditional prior maintains the prior mean, but increases the norm of the prior covariance by a constant term $\beta \in (0, 1)$. More precisely, we have that

$$\begin{aligned} g_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{(s_{1:t-1})}, \\ G_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \begin{cases} \beta^{-1} \boldsymbol{\Sigma}_{(s_{1:t-1})} & s_t = 1\,, \\ \boldsymbol{\Sigma}_{(s_{1:t-1})} & s_t = 0\,. \end{cases} \end{aligned} \tag{3.22}$$

This combination was first proposed in Li et al. (2021).

CPT-MMPR (changepoint timestep using moment-matched prior reset): for $\psi_t = \boldsymbol{\zeta}_t$, with $\boldsymbol{\zeta}_t = \{\zeta_{1,t}, \ldots, \zeta_{\ell,t}\}$, and $\zeta_{\ell,t}$ the position of the last changepoint, the work of Fearnhead and Liu (2011) assumes a dependence structure between changepoints. That is, to build the conditional prior mean and covariance, they consider the past $\mathcal{D}_{\zeta_{\ell,t}:t-1}$ datapoints whenever $\zeta_{\ell,t} \leq t - 1$ and a moment-matched approximation to the mixture density over all possible subset densities since the last changepoint whenever $\zeta_{\ell,t} = t$. Mathematically,

if $\zeta_{\ell,t} \leq t-1$, the prior mean and the prior covariance take the form

$$\begin{aligned} g(\boldsymbol{\zeta}_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{(\zeta_{\ell,t}:t-1)}, \\ G(\boldsymbol{\zeta}_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{(\zeta_{\ell,t}:t-1)}. \end{aligned} \tag{3.23}$$

If $\zeta_{\ell,t} = t$, the conditional prior mean and conditional prior covariance are built using a moment-matching approach. For the Gaussian case, moment-matching is equivalent to minimising a KL divergence (Minka, 2013). This takes the form

$$\begin{aligned} g(\boldsymbol{\zeta}_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_t \,, \\ G(\boldsymbol{\zeta}_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_t, \end{aligned} \tag{3.24}$$

where

$$\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t = \underset{\boldsymbol{\mu}, \boldsymbol{\Sigma}}{\arg\min}\, \mathbf{D}_{\mathrm{KL}} \left(\sum_{\zeta_{\ell,t-1}=1}^{t-1} \tilde{q}(\boldsymbol{\theta}_t \,|\, \zeta_{\ell,t-1}, \zeta_{\ell,t}, \mathcal{D}_{1:t-1}) \,||\, \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right). \tag{3.25}$$

Here, $\mathbf{D}_{\mathrm{KL}}$ denotes the KL divergence and $\tilde{q}(\boldsymbol{\theta}_t \,|\, \zeta_{\ell,t-1}, \zeta_{\ell,t}, \mathcal{D}_{1:t-1})$ is the unnormalised density

$$\begin{aligned} &\tilde{q}(\boldsymbol{\theta}_t \,|\, \zeta_{\ell,t-1}, \zeta_{\ell,t}, \mathcal{D}_{1:t-1}) \\ &= p(\zeta_{\ell,t-1} \,|\, \mathcal{D}_{1:t-1})\, p(\zeta_{\ell,t} \,|\, \zeta_{\ell,t-1}, \mathcal{D}_{1:t-1})\, \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{(\zeta_{\ell,t-1}:t-1)}, \boldsymbol{\Sigma}_{(\zeta_{\ell,t-1}:t-1)}). \end{aligned}$$

Choosing `MMPR` couples the choice of (M.3: prior) with the algorithmic choice (A.2: weighting) through $p(\zeta_{\ell,t-1} \,|\, \mathcal{D}_{1:t-1})$ and the choice $p(\zeta_{\ell,t} \,|\, \zeta_{\ell,t-1}, \mathcal{D}_{1:t-1})$. For an example of `MMPR` with `RL` choice of (M.2: auxvar), see Appendix 3.3.2.

3.1.5 Algorithm to compute the posterior over model parameters (A.1)

This section presents algorithms for estimating the density $q(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t})$; we focus on methods that yield Gaussian posterior densities. Specifically, we are interested in practical approaches for approximating the conditional Bayesian posterior, as given in (3.7).

There is a vast body of literature on methods for estimating the posterior over model parameters, many of which were introduced in Chapter 2. Here, we focus on three common approaches for computing the Gaussian posterior: conjugate updates (`Cj`), linear-Gaussian approximations (`LG`), and variational Bayes (`VB`). For an overview of choices of (A.1: posterior) and a comparison in terms of their computational complexity, see Table 3 in Jones et al. (2024).

Conjugate updates (`Cj`)

Conjugate updates (`Cj`) provide a classical approach for computing the posterior by leveraging conjugate prior distributions. Conjugate updates occur when the functional form of the conditional prior $\pi(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t-1})$ matches that of the measurement model $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ (Robert et al., 2007, Section 3.3). This property allows the posterior to remain within the same family as the prior, which leads to analytically tractable updates and facilitates efficient recursive estimation.

A common example is the conjugate Gaussian model, where the measurement model is Gaussian with known variance and the prior is a multivariate Gaussian. This results in closed-form updates for both the mean and covariance. Another example is the Beta-Bernoulli pair, where the measurement model follows a Bernoulli distribution with an unknown probability, and the prior is a Beta distribution. See e.g., Bernardo and Smith (1994); West and Harrison (1997) for details.

The recursive nature of conjugate updates makes them particularly useful for real-time or sequential learning scenarios, where fast and efficient updates are crucial.

Linear-Gaussian approximation (`LG`)

The linear-Gaussian (`LG`) method builds on the conjugate updates (`Cj`) above. More precisely, the prior is Gaussian and the measurement model is approximated by a linear Gaussian model, which simplifies computations.

The prior over model parameters is taken as:

$$\pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\right), \tag{3.26}$$

where $\boldsymbol{\mu}_{t-1}^{(\psi_t)}$ and $\boldsymbol{\Sigma}_{t-1}^{(\psi_t)}$ are the mean and covariance, respectively. We use the measurement function h to define a first-order approximation $\bar{h}_t$ around the prior mean which is given by

$$\bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t) = h\left(\boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{x}_t\right) + \mathbf{H}_t\left(\boldsymbol{\theta}_t - \boldsymbol{\mu}_{t-1}^{(\psi_t)}\right). \tag{3.27}$$

Here, $\mathbf{H}_t$ is the Jacobian of $h(\boldsymbol{\theta}, \boldsymbol{x}_t)$ with respect to $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\mu}_{t-1}^{(\psi_t)}$. The approximate posterior measure is given by

$$\begin{aligned} q(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t}) &\propto \mathcal{N}(\boldsymbol{y}_t \,|\, \bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t), \mathbf{R}_t)\, \pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) \\ &= \mathcal{N}(\boldsymbol{y}_t \,|\, \bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t), \mathbf{R}_t)\, \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\right) \\ &\propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t^{(\psi_t)}, \boldsymbol{\Sigma}_t^{(\psi_t)}), \end{aligned} \tag{3.28}$$

where $\mathbf{R}_t$ is a known noise covariance matrix of the measurement $\boldsymbol{y}_t$. Under the `LG`

algorithmic choice, the updated equations are

$$\begin{aligned}
\hat{\boldsymbol{y}}_t^{(\psi_t)} &= h\left(\boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{x}_t\right), \\
\mathbf{S}_t^{(\psi_t)} &= \mathbf{H}_t \, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)} \, \mathbf{H}_t^\intercal + \mathbf{R}_t, \\
\mathbf{K}_t^{(\psi_t)} &= \boldsymbol{\Sigma}_{t-1}^{(\psi_t)} \, \mathbf{H}_t^\intercal \left(\mathbf{S}_t^{(\psi_t)}\right)^{-1}, \\
\boldsymbol{\mu}_t^{(\psi_t)} &= \boldsymbol{\mu}_{t-1}^{(\psi_t)} + \mathbf{K}_t^{(\psi_t)} \left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t^{(\psi_t)}\right), \\
\boldsymbol{\Sigma}_t^{(\psi_t)} &= \boldsymbol{\Sigma}_{t-1}^{(\psi_t)} - \left(\mathbf{K}_t^{(\psi_t)}\right) \left(\mathbf{S}_t^{(\psi_t)}\right) \left(\mathbf{K}_t^{(\psi_t)}\right)^\intercal.
\end{aligned} \tag{3.29}$$

Variational Bayes (`VB`)

Here, we consider an extension of the variational Bayes method introduced in Section 2.4.1, where we condition on the auxiliary variable ψ_t. We have the following optimisation problem for the posterior variational parameters:

$$(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \underset{\boldsymbol{\mu}, \boldsymbol{\Sigma}}{\arg\min} \, \mathbf{D}_{\mathrm{KL}} \left(\mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,\|\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t) \, \pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1})\right), \tag{3.30}$$

where $\pi_t(\boldsymbol{\theta}_t;\, \psi_t)$ is the chosen prior distribution (M.3: prior).

Alternative methods

Alternative approaches for handling nonlinear or nonconjugate measurements have been proposed, such as sequential Monte Carlo (SMC) methods (de Freitas et al., 2000), and ensemble Kalman filters (EnKF) (Roth et al., 2017). These sample-based methods are particularly advantageous when the dimensionality of $\boldsymbol{\theta}$ is large, or when a more exact posterior approximation is required, providing greater flexibility in non-linear and non-Gaussian environments.

Generalised Bayesian methods, such as Mishkin et al. (2018); Knoblauch et al. (2022), generalise the VB update of (3.30) by allowing the right-hand side to be a loss function. Alternatively, online gradient descent methods like Bencomo et al. (2023) emulate state-space modelling via gradient-based optimisation.

3.1.6 Weighting function for auxiliary variable (A.2)

The term $\nu_t(\psi_t)$ defines the weights over possible values of the auxiliary variable (M.2: auxvar). We compute it as the marginal posterior distribution $\nu_t(\psi_t) = p(\psi_t \,|\, \mathcal{D}_{1:t})$ (see e.g., Adams and MacKay (2007); Fearnhead and Liu (2007, 2011); Li et al. (2021)) or with *ad-hoc* rules (see e.g., Nassar et al. (2022); Abélès et al. (2024); Titsias et al. (2024)).

In the former case, the weighting function takes the form

$$\begin{aligned}\nu_t(\psi_t) &= p(\psi_t \,|\, \mathcal{D}_{1:t}) \\ &= p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \psi_t, \mathcal{D}_{1:t-1}) \int_{\psi_{t-1} \in \boldsymbol{\Psi}_{t-1}} p(\psi_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(\psi_t \,|\, \psi_{t-1}, \mathcal{D}_{1:t-1}) \mathrm{d}\psi_{t-1},\end{aligned} \tag{3.31}$$

where one assumes that $\boldsymbol{y}_t$ is conditionally independent of ψ_{t-1}, given ψ_t, and $\boldsymbol{x}_t$ is an exogenous vector. The first term on the right hand side of (3.31) is known as the conditional posterior predictive, and is given by

$$p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \psi_t, \mathcal{D}_{1:t-1}) = \int p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, \pi(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_t. \tag{3.32}$$

This integral over $\boldsymbol{\theta}_t$ may require approximations, as we discussed in Section 3.1.5. Furthermore, the integral over ψ_{t-1} in (3.31) may also require approximations, depending on the nature of the auxiliary variable ψ_t, and the modelling assumptions for $p(\psi_t \,|\, \psi_{t-1}, \mathcal{D}_{1:t-1})$. We provide some examples below.

Discrete auxiliary variable (DA)

Here we assume the auxiliary variable takes values in a discrete space $\psi_t \in \boldsymbol{\Psi}_t$. The weights for the discrete auxiliary variable (`DA`) can be computed with a fixed number of hypotheses $K \geq 1$ or with a growing number of hypotheses if the cardinality of $\boldsymbol{\Psi}_t$ increases through time; we denote these cases by `DA[K]` and `DA[inf]` respectively. Below, we provide three examples that estimate the weights under `DA[inf]` recursively.

`RL` (runlength with Markovian assumption): for $\psi_t = r_t$, the work of Adams and MacKay (2007) takes

$$p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} 1 - H(r_{t-1}) & \text{if } r_t = r_{t-1} + 1, \\ H(r_{t-1}) & \text{if } r_t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.33}$$

where $H : \mathbb{N}_0 \to (0, 1)$ is the hazard function. A popular choice is to take $H(r) = \kappa \in (0, 1)$ to be a fixed constant hyperparameter known as the hazard rate. The choice `RL[inf]-PR` is known as the Bayesian online changepoint detection model (BOCD).

`CPL` (changepoint location): for $\psi_t = s_{1:t}$, the work of Li et al. (2021) takes

$$p(\tilde{s}_{1:t} \,|\, s_{1:t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} \kappa & \text{if } ([\tilde{s}_{1:t} \setminus \tilde{s}_t] = s_{1:t-1}) \text{ and } \tilde{s}_t = 1, \\ 1 - \kappa & \text{if } ([\tilde{s}_{1:t} \setminus \tilde{s}_t] = s_{1:t-1}) \text{ and } \tilde{s}_t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.34}$$

i.e., the sequence of changepoints at time t correspond to the sequence of changepoints up to time $t-1$, plus a newly sampled value for t. See Appendix 3.3.4 for details on how

to compute $\nu_t(s_{1:t})$.

CPT (changepoint timestep with Markovian assumption): for $\psi_t = \boldsymbol{\zeta}_t$, the work of Fearnhead and Liu (2007) takes

$$p(\boldsymbol{\zeta}_t \,|\, \boldsymbol{\zeta}_{t-1}, \mathcal{D}_{1:t-1}) = p(\zeta_{\ell,t} \,|\, \zeta_{\ell,t-1}) = J(\zeta_{\ell,t} - \zeta_{\ell,t-1}), \tag{3.35}$$

with $J : \mathbb{N}_0 \to (0, 1)$ a probability mass function. Note that $\zeta_{\ell,t} - \zeta_{\ell,t-1}$ is the distance between two changepoints, i.e., a runlength. In this sense, $\zeta_{\ell,t} - \zeta_{\ell,t-1} = r_t$, which relates CPT to RL. See their paper for details on how to compute $\nu_t(\boldsymbol{\zeta}_t)$.

Low-memory variants — from `DA[inf]` to `DA[K]` In the examples above, the number of computations to obtain $\sum_{\psi_t} \nu_t(\psi_t)$ grows in time. To fix the computational cost, one can restrict the sum to be over a subset $\mathcal{A}_t$ of the space of ψ_t with cardinality $|\mathcal{A}_t| = K \geq 1$. Each element in the set $\mathcal{A}_t$ is called a hypothesis and given $K \geq 1$, we keep the K most likely elements —according to $\nu_t(\psi_t)$— in $\mathcal{A}_t$. We then define the normalised weighting function

$$\hat{\nu}_t(\psi_t) = \frac{\nu_t(\psi_t)}{\sum_{\psi'_t \in \mathcal{A}_t} \nu_t(\psi'_t)}, \tag{3.36}$$

which we use instead of $\nu_t(\psi_t)$. For example, in RL above, $\mathcal{A}_{t-1} = \{r_{t-1}^{(k)} : k = 1, \ldots, K\}$ are the unique K most likely runlengths where the superscript represents the ranking according to $\nu_{t-1}(\cdot)$. Then, at time t, the augmented set $\bar{\mathcal{A}}_t$ becomes $(\mathcal{A}_{t-1} + 1) \cup \{0\}$, where the sum is element-wise, and we then compute the K most likely elements of $\bar{\mathcal{A}}_t$ to define $\mathcal{A}_t$. In CPL, $\mathcal{A}_{t-1}$ contains the K most likely sequences of changepoints, $\bar{\mathcal{A}}_t$ is defined as the collection of the $2\,K$ sequences where each sequence of $\mathcal{A}_{t-1}$ has a zero or one concatenated at the end. Finally, the K most likely elements in $\bar{\mathcal{A}}_t$ define $\mathcal{A}_t$. This style of pruning is common in segmentation methods; see, e.g., Saatçi et al. (2010), but other pruning are also possible, such as those proposed by Li et al. (2021), or sampling-based approaches; see e.g. Doucet et al. (2000).

Other choices for `DA[K]` Finally, some choices of weighting functions are derived using ad-hoc rules, meaning that explicit or approximate solutions to the Bayesian posterior are not needed. One of the most popular choices of ad-hoc weighting functions are mixture of experts, which weight different models according to a given criterion.

ME (mixture of experts with algorithmic weighting): Consider $\psi_t = \boldsymbol{\alpha}_t$. Let $\boldsymbol{\alpha}_{t,k} = k$ denote the k-th *configuration* over (M.3: prior). Next, denote by $\boldsymbol{w}_t = \{\boldsymbol{w}_{t,1}, \ldots, \boldsymbol{w}_{t,K}\}$ a set of weights, where $\boldsymbol{w}_{t,k}$ corresponds to the weight for the k-th expert at time t. The work of Chaer et al. (1997) considers the weighting function

$$\nu_t(\boldsymbol{w}_t)_k = \frac{\exp(\boldsymbol{w}_{t,k}^\intercal \, \boldsymbol{y}_t)}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_{t,j}^\intercal \, \boldsymbol{y}_t)}, \tag{3.37}$$

for $k = 1, \ldots, K$. The set of weights $\boldsymbol{w}_t$ are determined by maximising the surrogate gain function

$$\mathcal{G}_t(\boldsymbol{w}_t) = p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) = \sum_{k=1}^{K} p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \boldsymbol{\alpha}_{t,k}, \mathcal{D}_{1:t-1})\, \nu_t(\boldsymbol{w}_t)_k, \tag{3.38}$$

with respect to $\boldsymbol{w}_{t,k}$ for all $k = 1, \ldots, K$ at every timestep t.

We write `DA[K]`, where `K` is the number hypothesis, for methods that use `K` hypotheses at most. On the other hand, we write `DA[inf]` when we do not impose a bound on the number of hypotheses used. Note that even when the choice of (A.2: weighting) is built using `DA[inf]`, one can modify it to make it `DA[K]`.

Discrete auxiliary variable with greedy hypothesis selection (`DA[1]`) A special case of the above is `DA[1]`, where we employ a single hypothesis. In these scenarios, we set $\nu(\psi_t) = 1$ where ψ_t is the most likely hypothesis.

RL (Greedy runlength): For $\psi_t = r_t$ and `DA[1]`, we take

$$p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} 1-\kappa & \text{if } r_t = r_{t-1} + 1, \\ \kappa & \text{if } r_t = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.39}$$

Our choice of (A.2: weighting) is based on the marginal predictive likelihood ratio, which is derived from the computation of $p(r_t \,|\, \mathcal{D}_{1:t})$ under either either an increase in the runlength ($r_t^{(1)} = r_{t-1} + 1$) or a reset of the runlength ($r_t^{(0)} = 0$). Under these assumptions, the form of $\nu_t(r_t^{(1)})$ is

$$\nu_t(r_t^{(1)}) = \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\kappa)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,\kappa + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\kappa)}. \tag{3.40}$$

For details on the computation of (A.2: weighting), see Appendix 3.3.3. For a detailed implementation of (M.2: auxvar) `RL`, (M.3: prior) `OUPR`, (A.2: weighting) `DA[1]`, and (A.1: posterior) `LG`, see Algorithm 10 in the Appendix.

For example, `RL[1]` is a runlength r_t with a single hypothesis. We provide another example next.

CPL (changepoint location with retrospective membership): for $\psi_t = s_{1:t}$, the work of Nassar et al. (2022) evaluates the probability of past datapoints belonging in the current regime. In this scenario,

$$p(s_{1:t} \,|\, s_{1:t-1}, \mathcal{D}_{1:t-1}) = p(s_{1:t} \,|\, \mathcal{D}_{1:t-1}), \tag{3.41}$$

so that

$$p(s_{1:t} \,|\, \mathcal{D}_{1:t}) \propto p(s_{1:t} \,|\, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, s_{1:t}, \mathcal{D}_{1:t-1}). \tag{3.42}$$

This method allows for exact computation by summing over all possible 2^t elements. However, to reduce the computational cost, they propose a discrete optimisation over possible values $\{\nu_t(s_{1:t}) : s_{1:t} \in \{0,1\}^t\}$, where $\nu_t(s_{1:t}) = p(s_{1:t} \mid \mathcal{D}_{1:t})$. Then, the hypothesis with highest probability is stored and gets assigned a weight of one.

Continuous auxiliary variable (CA)

Here, we briefly discuss continuous auxiliary variables (CA). For some choices of ψ_t and transition densities $p(\psi_t \mid \psi_{t-1}, \mathcal{D}_{1:t-1})$, computation of (3.31) becomes infeasible. In these scenarios, we use simpler approximations. We give an example below.
CPP (Changepoint probability with empirical Bayes estimate): for $\psi_t = \upsilon_t$, consider

$$p(\upsilon_t \mid \upsilon_{t-1}, \mathcal{D}_{1:t-1}) = p(\upsilon_t), \tag{3.43}$$

so that

$$p(\upsilon_t \mid \mathcal{D}_{1:t}) \propto p(\upsilon_t)\, p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \upsilon_t). \tag{3.44}$$

The work of Titsias et al. (2024) takes $\nu_t(\upsilon_t) = \delta(\upsilon_t - \upsilon_t^*)$, where δ is the Dirac delta function and υ_t^* is a point estimate centred at the maximum of the marginal posterior predictive likelihood:

$$\upsilon_t^* = \underset{\upsilon \in [0,1]}{\arg\max}\, p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \upsilon, \mathcal{D}_{1:t-1}). \tag{3.45}$$

In practice, (3.45) is approximated by taking gradient steps towards the minimum. This is a form of empirical Bayes approximation, since we compute the most likely value of the prior after marginalizing out $\boldsymbol{\theta}_t$. The work of Galashov et al. (2024) considers a modified configuration with choice of (M.2: auxvar) $\boldsymbol{\upsilon}_t \in (0,1)^D$.

3.2 Unified view of examples in the literature

Table 3.3 shows that many existing methods can be written as instances of BONE. Rather than specifying the choice of (M.1), we instead write the task for which it was designed, as discussed in Section 2.1.1. We will experimentally compare a subset of these methods in Section 3.5.

The methods presented in Table 3.3 can be directly applied to tackle any of the problems mentioned in Section 2.1.1. However, as choice of (M.1: likelihood), we specify the task under which the configuration was introduced.[3]

3.3 Worked examples

In this section, we provide a detailed calculation of $\nu_t(\psi_t)$ for some choices of ψ_t. We consider a choice of (M.1: likelihood) to be linear Gaussian with known observation

[3] In general, the components of BONE can be thought as the building blocks for new methods. Some of these combinations would not be useful, but they can be employed nonetheless.

Reference	Task	M.2: auxvar	M.3: prior	A.1: posterior	A.2: weight
Kalman (1960)	filtering	C	LSSM	LG	DA[1]
Magill (1965)	filtering	ME	LSSM	LG	DA[K]
Chang and Athans (1978)	filtering	ME	LSSM	LG	CA
Chaer et al. (1997)	filtering	ME	LSSM	LG	DA[K]
Ghahramani and Hinton (2000)	SSSM	ME	Static	VB	CA
Adams and MacKay (2007)	seg.	RL	PR	Cj	DA[inf]
Fearnhead and Liu (2007)	seg. & preq.	CPT/ME	PR	Any	DA[inf]
Wilson et al. (2010)	seg.	RLCC	PR	Cj	DA[inf]
Fearnhead and Liu (2011)	seg.	CPT/ME	MMPR	Any	DA[inf]
Mellor and Shapiro (2013)	bandits	RL	PR	Cj	DA[inf]
Nguyen et al. (2017)	OCL	CPL	Sub	VB	DA[1]
Knoblauch and Damoulas (2018)	seg.	RL/ME	PR	Cj	DA[inf]
Kurle et al. (2019)	OCL	CPV	Sub	VB	DA[1]
Li et al. (2021)	OCL	CPL	MCI	VB	DA[inf]
Nassar et al. (2022)	bandits & OCL	CPV	Sub	LG	DA[1]
Liu (2023)	preq.	ME	C,LSSM	Any	DA[K]
Chang et al. (2023)	OCL	C	ACI	LG	DA[1]
Titsias et al. (2024)	OCL	CPP	OU	LG	CA
Galashov et al. (2024)	CL	CPP	OU	VB	CA
Abélès et al. (2024)	preq.	ME	LSSM	LG	DA[K]
RL[1]-OUPR* (ours)	any	RL	SPR	Any	DA[1]

Table 3.3: List of methods ordered by publication date. The tasks are discussed in Section 2.1.1. We use the following abbreviations: SSSM means switching state space model; (O)CL means (online) continual learning; seg. means segmentation; preq. means prequential. Methods that consider two choices of (M.2: auxvar) are denoted by 'X/Y'. This corresponds to a double expectation in (3.6)—one for each choice of auxiliary variable.

variance $\mathbf{R}_t$, i.e.,

$$p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta}_t, \mathbf{R}_t). \tag{3.46}$$

3.3.1 Runlength with prior reset (RL-PR)

Unbounded number of hypotheses RL[inf]-PR

The work in Adams and MacKay (2007) takes $\psi_t = r_t$ to be the runlength, with $r_t \in \{0, 1, \ldots, t\}$, that that counts the number of steps since the last changepoint. Assume the runlength follows the dynamics (3.33). We consider $\nu_t(r_t) = p(r_t|\mathcal{D}_{1:t})$ such that

$$p(r_t \,|\, \mathcal{D}_{1:t}) = \frac{p(r_t, \mathcal{D}_{1:t})}{\sum_{\hat{r}_t=0}^{t} p(\hat{r}_t, \mathcal{D}_{1:t})}, \tag{3.47}$$

for $r_t \in \{0, \ldots, t\}$. The RL-PR method estimates $p(r_t, \mathcal{D}_{1:t})$ for all $r_t \in \{0, \ldots, t\}$ at every timestep. To estimate this value recursively, we sum over all possible previous runlengths as follows

$$\begin{aligned}
&p(r_t, \mathcal{D}_{1:t}) \\
&= \sum_{r_{t-1}=0}^{t-1} p(r_t, r_{t-1}, \mathcal{D}_{1:t-1}, \mathcal{D}_t) \\
&= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, r_t, r_{t-1}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \\
&= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1}) p(r_t \,|\, r_{t-1}).
\end{aligned} \tag{3.48}$$

In the last equality, there are two implicit assumptions, (i) the runlength at time t is conditionally independent of the data $\mathcal{D}_{1:t-1}$ given the runlength at time $t-1$, and (ii) the model is Markovian in the runlength, that is, conditioned on r_t, the value of r_{t-1} bears no information. Mathematically, this means that (i) $p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1}) = p(r_t \,|\, r_{t-1})$ and (ii) $p(\boldsymbol{y}_t \,|\, r_t, r_{t-1}, \mathcal{D}_{1:t-1}) = p(\boldsymbol{y}_t \,|\, r_t, \mathcal{D}_{1:t-1})$. From (3.48), we observe there are only two possible scenarios for the value of r_t. Either $r_t = 0$ or $r_t = r_{t-1} + 1$ with $r_{t-1} \in \{0, \ldots, t-1\}$. Thus, $p(r_t, \mathcal{D}_{1:t})$ becomes

$$\begin{aligned} p(r_t, \mathcal{D}_{1:t}) &= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) && \text{if } r_t \geq 1 \\ p(r_t, \mathcal{D}_{1:t}) &= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) && \text{if } r_t = 0\,. \end{aligned} \tag{3.49}$$

The joint density (3.49) considers two possible scenarios: either we stay in a regime considering the past $r_t \geq 1$ observations, or we are in a new regime, in which $r_t = 0$. Finally, note that (3.49) depends on three terms: (i) the transition probability $p(r_t \,|\, r_{t-1})$, which it is assumed to be known, (ii) the previous log-joint $p(r_{t-1}, \mathcal{D}_{1:t-1})$, with $r_{t-1} \in \{0, 1, \ldots, t-1\}$, which is estimated at the previous timestep, and (iii) the prior predictive density

$$p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) = \int p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, p(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_t. \tag{3.50}$$

For a choice of (M.1: likelihood) given by (3.46) and a choice of (M.3: prior) given by (3.18), the posterior predictive (3.50) takes the form.

$$\begin{aligned} p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) &= \int \mathcal{N}\left(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\theta}_t, \mathbf{R}_t\right) \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(r_t)}, \boldsymbol{\Sigma}_{t-1}^{(r_t)}\right) \mathrm{d}\boldsymbol{\theta}_t \\ &= \mathcal{N}\left(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \boldsymbol{\mu}_{t-1}^{(r_t)},\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\Sigma}_{t-1}^{(r_t)}\, \boldsymbol{x}_t + \mathbf{R}_t\right), \end{aligned} \tag{3.51}$$

with $r_t \in \{0, \ldots, t-1\}$. Here, $\left(\boldsymbol{\mu}_{t-1}^{(r_t)}, \boldsymbol{\Sigma}_{t-1}^{(r_t)}\right)$ are the posterior mean and covariance at time $t-1$ built using the last $r_t \geq 1$ observations. If $r_t = 0$, then $(\boldsymbol{\mu}_{t-1}^{(r_t)}, \boldsymbol{\Sigma}_{t-1}^{(r_t)}) = (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.

Bounded number of hypotheses `RL[K]-PR`

If we maintain a set of K possible hypotheses, then $\boldsymbol{\Psi}_t = \{r_{t-1}^{(1)}, \ldots, r_{t-1}^{(K)}\} \in \{0, \ldots, t-1\}^K$ is a collection of K unique runlengths obtained at time $t-1$. Next, (3.48) takes the

form

$$p(r_t, \mathcal{D}_{1:t}) = p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \qquad \text{if } r_t \geq 1, \tag{3.52}$$

$$p(r_t, \mathcal{D}_{1:t}) = p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1} \in \boldsymbol{\Psi}_{t-1}} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \qquad \text{if } r_t = 0\,. \tag{3.53}$$

Here, we have that either $r_t = r_{t-1} + 1$ when $r_{t-1} \in \boldsymbol{\Psi}_{t-1}$ or $r_t = 0$. After computing $p(r_t, \mathcal{D}_{1:t})$ for all $K+1$ possibles values of r_t, a choice is made to keep K hypotheses. For timesteps $t \leq K$, we evaluate all possible hypotheses until $t > K$.

Algorithm 8 shows an update step under this process when we maintain a set of K possible hypotheses.

3.3.2 Runlength with moment-matched prior reset (RL-MMPR)

Here, we consider a modified version of the method introduced in Fearnhead and Liu (2011). We consider the choice of RL and adjust the choice of (M.3: prior) for RL-PR introduced in Appendix 3.3.1 whenever $r_t = 0$. In this combination, for $r_t = 0$, we take $\tau(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1}) = p(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1})$. Next

$$\begin{aligned} p(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1}) &= \sum_{r_{t-1}=0}^{t-1} p(\boldsymbol{\theta}_t, r_{t-1} \,|\, r_t, \mathcal{D}_{1:t-1}) \\ &= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, p(\boldsymbol{\theta}_t \,|\, r_t, r_{t-1}, \boldsymbol{y}_{1:t-1}) \\ &= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})}, \boldsymbol{\Sigma}_{t-1}^{(r_{t-1})}). \end{aligned} \tag{3.54}$$

Because (3.54) is a mixture model, we choose a conditional prior to be Gaussian that approximates the first two moments. We obtain

$$\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}] = \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})} \tag{3.55}$$

for the first moment, and

$$\mathbb{E}[\boldsymbol{\theta}_t\, \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \left(\boldsymbol{\Sigma}_{t-1}^{(r_{t-1})} + \boldsymbol{\mu}_{t-1}^{(r_{t-1})}\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})\intercal} \right) \tag{3.56}$$

for the second moment. The conditional prior mean and prior covariance under $r_t = 0$ take the form

$$\begin{aligned}\boldsymbol{\mu}_t^{(0)} &= \mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}],\\ \boldsymbol{\Sigma}_t^{(0)} &= \mathbb{E}[\boldsymbol{\theta}_t \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] - (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])\,(\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])^\intercal\,.\end{aligned} \tag{3.57}$$

Algorithm 9 shows an update step under this process when we maintain a set of K possible hypotheses.

3.3.3 Runlength with OU dynamics and prior reset (RL[1]-OUPR*)

In this section, we provide pseudocode for the new hybrid method we propose. Specifically, our choices in BONE are: RL[1]-OUPR* for (M.2: auxvar) and (M.3: prior), LG for (A.1: posterior), and DA[1] for (A.2: weighting). Because of our choice of (A.2: weighting), RL[1]-OUPR* considers a single hypothesis (or runlength) which, at every timestep, is either increased by one or set back to zero, according to the probability of a changepoint and a threshold $\epsilon \in (0, 1)$.

In essence, RL[1]-OUPR* follows the logic behind RL[1]-PR introduced in Section 3.3.1 with $K = 1$ hypothesis and different choice of (M.3: prior). To derive the algorithm for RL[1]-OUPR* at time $t > 1$, suppose r_{t-1} is available (the only hypothesis we track). Denote by $r_t^{(1)}$ the hypothesis of a runlength increase, i.e., $r_t = r_{t-1} + 1$ and denote by $r_t^{(0)}$ the hypothesis of a runlenght reset, i.e., $r_t = 0$. The probability of a runlength increase under a single hypothesis takes the form:

$$\begin{aligned}\nu_t(r_t^{(1)}) &= p(r_t^{(1)} \,|\, \mathcal{D}_{1:t})\\ &= \frac{p(r_t^{(1)}, \mathcal{D}_{1:t})}{p(r_t^{(1)}, \mathcal{D}_{1:t}) + p(r_t^{(0)}, \mathcal{D}_{1:t})}\\ &= \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\,(1-\kappa)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\,\kappa + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\,(1-\kappa)}\\ &= \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\kappa)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,\kappa + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\kappa)}.\end{aligned} \tag{3.58}$$

where $\kappa = p(r_t \,|\, r_{t-1})$ with $r_t = 0$ is the prior probability of a changepoint and and $1 - \kappa = p(r_t \,|\, r_{t-1}$ with $r_t = r_{t-1} + 1$ is the probability of continuation of the current segment.

Next, we use $\nu_t(r_t)$ to decide whether to update our parameters or reset them according to a prior belief according to some threshold ϵ. This implements our choice of (M.3: prior) given in (3.19) and (3.20). Because we maintain a single hypothesis, the weight at the end of the update step is set to 1. Algorithm 10 shows an update step for RL[1]-OUPR* under the choice of (M.1: likelihood) given by (3.46).

3.3.4 Changepoint location with multiplicative covariance inflation `CPL-MCI`

The work in Li et al. (2021) takes $\psi_t = s_{1:t}$ to be a t-dimensional vector where the i-th element is a binary vector that determines a changepoint at time t. Then, the sum of the entries of $s_{1:t}$ represents the total number of changepoints up to, and including, time t.

We take $\nu_t(s_{1:t}) = p(s_{1:t}|\mathcal{D}_{1:t})$, which is recursively expressed as

$$\begin{aligned} p(s_{1:t} \,|\, \mathcal{D}_{1:t}) &= p(s_t, s_{1:t-1} \,|\, \boldsymbol{y}_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \\ &= p(s_{1:t-1} \,|\, \mathcal{D}_{1:t-1}) p(s_t \,|\, s_{1:t-1}, \boldsymbol{x}_t, \boldsymbol{y}_t, \mathcal{D}_{1:t-1}). \end{aligned} \tag{3.59}$$

Here, $p(s_{1:t-1} \,|\, \mathcal{D}_{1:t-1})$ is inferred at the previous timestep $t-1$. The estimate of a changepoint conditioned on the past changes and the measurements is

$$\begin{aligned} &p(s_t = 1 \,|\, s_{1:t-1}, \mathcal{D}_{1:t}) \\ &\quad = \frac{p(s_t = 1) p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, s_{1:t-1}, s_t = 1, \mathcal{D}_{1:t-1})}{p(s_t = 1) p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \boldsymbol{y}_{1:t-1}) + p(s_t = 0) p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})} \\ &\quad = \left(1 + \exp\left(-\log\left(\frac{p(s_t = 1) p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \boldsymbol{y}_{1:t-1})}{p(s_t = 0) p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}\right)\right)\right)^{-1} = \sigma(m_t), \end{aligned} \tag{3.60}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ and

$$m_t = \log\left(\frac{p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}{p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}\right) + \log\left(\frac{p(s_t = 1)}{p(s_t = 0)}\right), \tag{3.61}$$

and similarly,

$$p(s_t = 0 \,|\, s_{1:t-1}, \mathcal{D}_{1:t}) = 1 - \sigma(m_t). \tag{3.62}$$

Finally, the transition between states is given by $p(s_{1:t} \,|\, s_{1:t-1}) = p(s_t)$.

3.4 Algorithms

Algorithm 8 Implementation of `RL[K]-PR`. We consider an update at time t and one-step ahead forecasting at time $t+1$ under a Gaussian linear model with known observation variance.

Require: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ // default prior beliefs

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // current observation

Require: $\{r_{t-1}^{(k)}\}_{k=1}^K \in \{0, \ldots, t-1\}^K$ // bank of runlengths at time $t-1$

Require: $\{p(r_{t-1}^{(k)}, \mathcal{D}_{1:t})\}_{k=1}^K$ // joint from past hypotheses

Require: $\left\{(\boldsymbol{\mu}_{t-1}^{(k)}, \boldsymbol{\Sigma}_{t-1}^{(k)})\right\}_{k=1}^K$ // beliefs from past hypotheses

Require: $\boldsymbol{x}_{t+1}$ // next-step observation

Require: $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}, \mathbf{R}_t)$ // Choice of (M.1: likelihood)

// Evaluate hypotheses if there is no changepoint

for $k = 1, \ldots, K$ **do**

  $r_t^{(k)} \leftarrow r_{t-1}^{(k)} + 1$

  $p(\boldsymbol{y}_t \,|\, r_t^{(k)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1}^{(k)}, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1}^{(k)} \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive for k-th hypothesis

  $p(r_t^{(k)}, \, \mathcal{D}_{1:t}) \leftarrow p(\boldsymbol{y}_t \,|\, r_t^{(k)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \, p(r_{t-1}^{(k)}, \mathcal{D}_{1:t-1}) \, p(r_t^{(k)} \,|\, r_{t-1}^{(k)})$ // update joint density

  $(\bar{\boldsymbol{\mu}}_t^{(k)}, \bar{\boldsymbol{\Sigma}}_t^{(k)}) \leftarrow (\boldsymbol{\mu}_{t-1}^{(k)}, \boldsymbol{\Sigma}_{t-1}^{(k)})$

  $\tau_t(\boldsymbol{\theta}_t; r_t^{(k)}) \leftarrow \mathcal{N}(\boldsymbol{\theta}_t \,|\, \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ // choice of (M.3: prior)

  $q_t(\boldsymbol{\theta}_t; \, r_t^{(k)}) \propto \tau_t(\boldsymbol{\theta}_t; r_t^{(k)}) \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})$ // following (3.29)

end for

// Evaluate hypothesis under a changepoint

$r_t^{(k+1)} \leftarrow 0$

$p(\boldsymbol{y}_t \,|\, r_t^{(k+1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_0, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_0 \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive for k-th hypothesis

$p(r_t^{(k+1)}, \mathcal{D}_{1:t}) \leftarrow p(\boldsymbol{y}_t \,|\, r_t^{(k+1)}, \, \boldsymbol{x}_t, \, \mathcal{D}_{1:t-1}) \sum_{k=1}^K p(r_t^{(k)}, \mathcal{D}_{1:t}) \, p(r_t^{(t+1)} \,|\, r_t^{(k)} - 1)$

// Extend number of hypotheses to $K+1$ and keep top K hypotheses

$I_{1:k} = \text{top.k}(\{p(r_t^{(1)}, \, \mathcal{D}_{1:t}), \ldots, p(r_t^{(k+1)}, \mathcal{D}_{1:t})\}, \, K)$

$\{p(r_t^{(k)}, \mathcal{D}_{1:t})\}_{k=1}^K \leftarrow \text{slice.at}(\{p(r_t^{(k)}, \, \mathcal{D}_{1:t})\}_{k=1}^{K+1}, \, I_{1:K})$

$\{(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^K \leftarrow \text{slice.at}(\{(\boldsymbol{\mu}_t^{(k)}, \, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^{K+1}, \, I_{1:K})$

// build weight and make prequential prediction

$\nu_t(r_t^{(k)}) \leftarrow \frac{p(r_t^{(k)}, \mathcal{D}_{1:t})}{\sum_{j=1}^K p(r_t^{(j)}, \mathcal{D}_{1:t})}$ for $k = 1, \ldots, K$

$\hat{\boldsymbol{y}}_{t+1} \leftarrow \boldsymbol{x}_{t+1}^\intercal \left(\sum_{k=1}^K \nu_t(r_t^{(k)}) \boldsymbol{\mu}_t^{(k)} \right)$ // prequential prediction under a linear-Gaussian model

return $\{(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)}, r_t^{(k)})\}_{k=1}^K, \, \hat{\boldsymbol{y}}_{t+1}$

In Algorithm 8, the function $\text{top.k}(A, K)$ returns the indices of the top $K \geq 1$ elements of A with highest value. The function $\text{slice.at}(A, B)$ returns the elements in A according to the list of indices B. If $|A| \leq |B|$, we return all elements in A.

Algorithm 9 Implementation of `RL[K]-MMPR`. We consider an update at time t and one-step ahead forecasting at time $t+1$ under a Gaussian linear model with known observation variance.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // current observation

Require: $\{r_{t-1}^{(k)}\}_{k=1}^K \in \{0, \ldots, t-1\}^K$ // bank of runlengths at time $t-1$

Require: $\{p(r_{t-1}^{(k)}, \mathcal{D}_{1:t})\}_{k=1}^K$ // joint from past hypotheses

Require: $\left\{(\boldsymbol{\mu}_{t-1}^{(k)}, \boldsymbol{\Sigma}_{t-1}^{(k)})\right\}_{k=1}^K$ // beliefs from past hypotheses

Require: $\boldsymbol{x}_{t+1}$ // next-step observation

Require: $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}, \mathbf{R}_t)$ // Choice of (M.1: likelihood)

// Evaluate hypotheses if there is no changepoint

for $k = 1, \ldots, K$ **do**

  $r_t^{(k)} \leftarrow r_{t-1}^{(k)} + 1$

  $p(\boldsymbol{y}_t \,|\, r_t^{(k)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1}^{(k)}, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1}^{(k)} \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive for k-th hypothesis

  $p(r_t^{(k)}, \, \mathcal{D}_{1:t}) \leftarrow p(\boldsymbol{y}_t \,|\, r_t^{(k)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \, p(r_{t-1}^{(k)}, \mathcal{D}_{1:t-1}) \, p(r_t^{(k)} \,|\, r_{t-1}^{(k)})$ // update joint density

  $(\bar{\boldsymbol{\mu}}_t^{(k)}, \bar{\boldsymbol{\Sigma}}_t^{(k)}) \leftarrow (\boldsymbol{\mu}_{t-1}^{(k)}, \boldsymbol{\Sigma}_{t-1}^{(k)})$

  $\tau_t(\boldsymbol{\theta}_t; r_t^{(k)}) \leftarrow \mathcal{N}(\boldsymbol{\theta}_t \,|\, \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ // choice of (M.3: prior)

  $q_t(\boldsymbol{\theta}_t; \, r_t^{(k)}) \propto \tau_t(\boldsymbol{\theta}_t; r_t^{(k)}) \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})$ // following (3.29)

end for

// Evaluate hypothesis under a changepoint

$r_t^{(k+1)} \leftarrow 0$

$\boldsymbol{\mu}_0 \leftarrow \mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}]$ // following (3.55)

$\boldsymbol{\Sigma}_0 \leftarrow \mathbb{E}[\boldsymbol{\theta}_t \, \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] - (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}]) \, (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])^\intercal$ // following (3.55) and (3.56)

$p(\boldsymbol{y}_t \,|\, r_t^{(k+1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_0, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_0 \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive for k-th hypothesis

$p(r_t^{(k+1)}, \mathcal{D}_{1:t}) \leftarrow p(\boldsymbol{y}_t \,|\, r_t^{(k+1)}, \, \boldsymbol{x}_t, \, \mathcal{D}_{1:t-1}) \sum_{k=1}^K p(r_t^{(k)}, \mathcal{D}_{1:t}) \, p(r_t^{(t+1)} \,|\, r_t^{(k)} - 1)$

// Extend number of hypotheses to $K+1$ and keep top K hypotheses

$I_{1:k} = \text{top.k}(\{p(r_t^{(1)}, \, \mathcal{D}_{1:t}), \ldots, p(r_t^{(k+1)}, \mathcal{D}_{1:t})\}, \, K)$

$\{p(r_t^{(k)}, \mathcal{D}_{1:t})\}_{k=1}^K \leftarrow \text{slice.at}(\{p(r_t^{(k)}, \, \mathcal{D}_{1:t})\}_{k=1}^{K+1}, \, I_{1:K})$

$\{(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^K \leftarrow \text{slice.at}(\{(\boldsymbol{\mu}_t^{(k)}, \, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^{K+1}, \, I_{1:K})$

// build weight and make prequential prediction

$\nu_t(r_t^{(k)}) \leftarrow \frac{p(r_t^{(k)}, \mathcal{D}_{1:t})}{\sum_{j=1}^K p(r_t^{(j)}, \mathcal{D}_{1:t})}$ for $k = 1, \ldots, K$

$\hat{\boldsymbol{y}}_{t+1} \leftarrow \boldsymbol{x}_{t+1}^\intercal \left(\sum_{k=1}^K \nu_t(r_t^{(k)}) \boldsymbol{\mu}_t^{(k)} \right)$ // prequential prediction under a linear-Gaussian model

return $\{(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)}, r_t^{(k)})\}_{k=1}^K, \, \hat{\boldsymbol{y}}_{t+1}$

Algorithm 10 Implementation of `RL[1]-OUPR*`, with update at time t and for one-step ahead forecasting at time $t+1$, under a Gaussian linear model with known observation variance.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // current observation

Require: $\boldsymbol{x}_{t+1}$ // next-step observation

Require: $\epsilon \in (0,1)$ // restart threshold

Require: $r_{t-1} \in \{0, \dots, t-1\}$ // runlength at time $t-1$

Require: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ // default prior beliefs

Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ // beliefs from prior step

Require: $p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\theta}^\intercal \boldsymbol{x}, \mathbf{R}_t)$ // Choice of (M.1: likelihood)

$(r_t^{(0)}, r_t^{(1)}) \leftarrow (0, r_{t-1}+1)$ // choice of (M.2: auxvar)

$p(\boldsymbol{y}_t \mid r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \mid \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_0, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_0 \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive at changepoint

$p(\boldsymbol{y}_t \mid r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \mid \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1}, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive if no changepoint

$\nu_t(r^{(1)}) \leftarrow \frac{p(\boldsymbol{y}_t \mid r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})(1-\pi)}{p(\boldsymbol{y}_t \mid r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\pi) + p(\boldsymbol{y}_t \mid r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,\pi}$ // probability of no-changepoint at timestep t

if $\nu(r_t^{(1)}) > \epsilon$ **then**

  $r_t \leftarrow r_t^{(1)}$

  $\bar{\boldsymbol{\mu}}_t^{(r_t)} \leftarrow \boldsymbol{\mu}_{t-1}^{(r_{t-1})} \, \nu(r_t^{(1)}) + \boldsymbol{\mu}_0 \, \left(1 - \nu(r_t^{(1)})\right)$

  $\bar{\boldsymbol{\Sigma}}_t^{(r_t)} \leftarrow \boldsymbol{\Sigma}_{t-1}^{(r_{t-1})} \, \nu(r_t^{(1)})^2 + \boldsymbol{\Sigma}_0 \, \left(1 - \nu(r_t^{(1)})^2\right)$

else if $\nu(r_t^{(1)}) \leq \epsilon$ **then**

  $r_t \leftarrow r_t^{(0)}$

  $\bar{\boldsymbol{\mu}}_t^{(r_t)} \leftarrow \boldsymbol{\mu}_0$

  $\bar{\boldsymbol{\Sigma}}_t^{(r_t)} \leftarrow \boldsymbol{\Sigma}_0$

end if

$\tau_t(\boldsymbol{\theta}_t; r_t) \leftarrow \mathcal{N}(\boldsymbol{\theta}_t \mid \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ // choice of (M.3: prior)

$q_t(\boldsymbol{\theta}_t;\, r_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \mid \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)\, p(\boldsymbol{y}_t \mid \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ // choice of (A.1: posterior)—via (3.29)

$\hat{\boldsymbol{y}}_{t+1} \leftarrow \boldsymbol{x}_{t+1}^\intercal \, \boldsymbol{\mu}_t$ // prequential prediction (given linear-Gaussian model)

return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, r_t)$, $\hat{\boldsymbol{y}}_{t+1}$

3.5 Experiments

In this section we experimentally evaluate different algorithms within the BONE framework on a number of tasks.

Each experiment consists of a *warmup* period where the hyperparameters are chosen, and a *deploy* period where sequential predictions and updates are performed. In each experiment, we fix the choice of measurement model h (M.1: likelihood) and posterior inference method (A.1: posterior), and then compare different methods with respect to their choice of (M.2: auxvar), (M.3: prior), and (A.2: weighting). For `DA` methods, we append the number of hypotheses in brackets to determine how many hypotheses are

being considered. For example, `RL[1]-PR` denotes one hypothesis, `RL[K]-PR` denotes K hypotheses, and `RL[inf]-PR` denotes all possible hypotheses. In all experiments, unless otherwise stated, we consider a single hypothesis for choices of `DA`. See Table 3.4 for the methods we compare.

M.2-M.3	Eq.	A.2	Description	Sections
			static	
`C-Static`	(3.16)	-	This corresponds to the static case with a classical Bayesian update. This method does not assume changes in the environment.	3.5.3, 4.7.4
			abrupt changes	
`RL-PR`	(3.18)	`DA[inf]`	This approach, commonly referred to as Bayesian online changepoint detection (**BOCD**), assumes that non-stationarity arises from independent blocks of time, each with stationary data. Estimates are made using data from the current block. See Appendix 3.3.1 for more details.	3.5.1, 3.5.1, 3.5.2, 3.5.3, 4.7.4,
`WoLF+RL-PR*`	(3.18)	`DA[inf]`	Special case of `RL-PR` with explicit choice of (M.1: likelihood) which makes it robust to outliers.	4.7.4
			gradual changes	
`CPP-OU`	(3.14)	`CA`	Updates are done using a discounted mean and covariance according to the probability estimate that a change has occurred.	3.5.1, 3.5.1, 3.5.2
`C-ACI`	(3.15)	-	At each timestep, this method assumes that the parameters evolve according to a linear map $\mathbf{F}_t$, at a rate given by a known positive semidefinite covariance matrix $\mathbf{Q}_t$.	3.5.1, 3.5.1, 3.5.2,
			abrupt & gradual changes	
`RL-MMPR`	(3.57)	`DA[inf]`	Modification of `CPT-MMPR` that assumes dependence between any two consecutive blocks of time and with choice of `RL`. This combination employs a moment-matching approach when evaluating the prior mean and covariance under a changepoint. See Appendix 3.3.2 for more details.	3.5.3
`RL-OUPR`	(3.19)	`DA[1]`	Depending on the threshold parameter, updates involve either (i) a convex combination of the prior belief with the previous mean and covariance based on the estimated probability of a change (given the run length), or (ii) a hard reset of the mean and covariance, reverting them to prior beliefs. See Appendix 3.3.3 for more details.	3.5.1, 3.5.1, 3.5.2, 3.5.3

Table 3.4: List of methods we compare in our experiments. The first column, **M.2–M.3**, is defined by the choices of (M.2: auxvar) and (M.3: prior). The second column, **Eq.**, references the equation that define M.2–M.3. The third column, **A.2**, determines the choice of (A.2: weighting). The fourth column, **Description**, provides a brief summary of the method. The fifth column, **Sections**, shows the sections where the method is evaluated. The choice of (M.1: likelihood) and (A.1: posterior) are defined on a per-experiment basis. (The only exception being `WolF+RL-PR`). For (M.2: auxvar) the acronyms are as follows: `RL` means runlength, `CPP` means changepoint probability, `C` means constant, and `CPT` means changepoint timestep. For (M.3: prior) the acronyms are as follows: `PR` means prior reset, `OU` means Ornstein–Uhlenbeck, `LSSM` means linear state-space model, `Static` means full Bayesian update, `MMR` means moment-matched prior reset, and `OUPR` means Ornstein–Uhlenbeck and prior reset. We use the convention in Hušková (1999) for the terminology abrupt/gradual changes.

3.5.1 Prequential prediction

In this section, we give several examples of non-stationary prequential prediction problems.

Online regression for hour-ahead electricity forecasting

In this experiment, we consider the task of predicting the hour-ahead electricity load before and after the Covid pandemic. We use the dataset presented in Farrokhabadi et al. (2022), which has 31,912 observations; each observation contains 7 features $\boldsymbol{x}_t$ and a single target variable $\boldsymbol{y}_t$. The 7 features correspond to pressure (kPa), cloud cover (%), humidity (%), temperature (C) , wind direction (deg), and wind speed (KmH). The target variable is

the hour-ahead electricity load (kW). To preprocess the data, we normalise the target variable $\boldsymbol{y}_t$ by subtracting an exponentially weighted moving average (EWMA) mean with a half-life of 20 hours, then dividing the resulting series by an EWMA standard deviation with the same half-life. To normalise the features $\boldsymbol{x}_t$, we divide each by a 20-hour half-life EWMA. The features are lagged by one hour.

Our choice of measurement model h is a two-hidden layer multilayered perceptron (MLP) with four units per layer and a ReLU activation function.

For this experiment, we consider `RL[1]-OUPR*` (our proposed method), `RL[1]-PR` (a classical method), `C-ACI` (a simple benchmark), and `CPP-OU` (a modern method). For computational convenience, we plug in a point-estimate (MAP estimate) of the neural network parameters when making predictions using h. More precisely, given ψ_t, we use $h(\boldsymbol{\theta}_t^*, \boldsymbol{x}_{t+1})$ to make a (conditional) prediction, where $\boldsymbol{\theta}_t^* = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}; \psi_t, \mathcal{D}_{1:t})$. For a fully Bayesian treatment of neural network predictions, see Immer et al. (2021); we leave the implementation of these approaches for future work.

The hyperparameters of each method are found using the first 300 observations (around 13 days) and deployed on the remainder of the dataset. Specifically, during the warmup period we tune the value of the probability of a changepoint for `RL[1]-OUPR*` and `RL[1]-PR`. For `C-ACI` we tune $\mathbf{Q}_t$, and for `CPP-OU` we tune the learning rate. See the open-source notebooks for more details.

In the top panel of Figure 3.2 we show the evolution of the target variable $\boldsymbol{y}_t$ between March 3 2020 and March 10 2020. The bottom panel of Figure 3.2 shows the 12-hour rolling mean absolute error (MAE) of predictions made by the methods. We see that there is a changepoint around March 7 2020 as pointed out in Farrokhabadi et al. (2022). This is likely due to the introduction of Covid lockdown rules. Among the methods considered, `C-ACI` and `RL[1]-OUPR*` adapt the quickest after the changepoint and maintain a low rolling MAE compared to `RL[1]-PR` and `CPP-OU`.

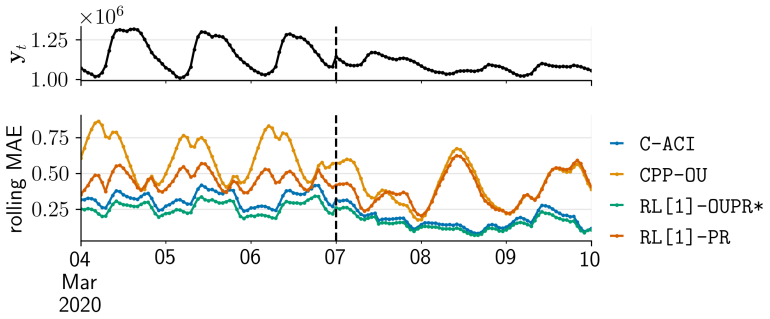

Figure 3.2: The **top panel** shows the target variable (electricity consumption) from March 1 2020 to March 12 2020. The **bottom panel** shows the twelve-hour rolling relative absolute error of predictions for the same time window. The dotted black line corresponds to March 7 2020, when Covid lockdown began.

Next, Figure 3.3, shows the forecasts made by each method between March 4 2020 and March March 8 2020. We observe a clear cyclical pattern before March 7 2020 but

less so afterwards, indicating a change in daily electricity usage from diurnal to constant.

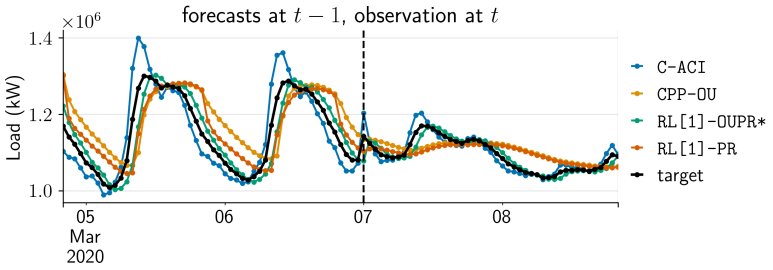

Figure 3.3: One day ahead electricity forecasting results for Figure 3.2. The dotted black line corresponds to March 7 2020.

We also observe that RL[1]-PR and CPP-OU slow-down their rate of adaptation. One possible explanation of this behaviour is that the changes are not abrupt enough to be captured by the algorithms. To provide evidence for this hypothesis, Figure 3.4 shows, on the left y-axis, the predictions for RL[1]-PR and the target variable $\boldsymbol{y}_t$. On the right y-axis, we show the estimated runlength.

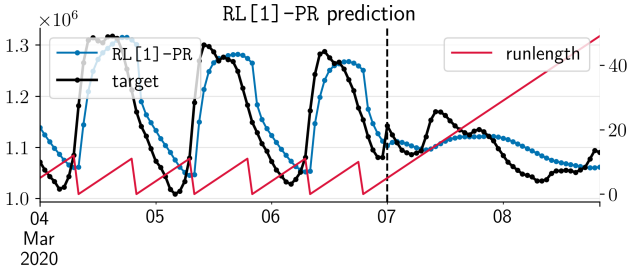

Figure 3.4: One day ahead electricity forecasting results for RL[1]-PR together with the target variable on the left y-axis, and the value for runlength (RL) on the right y-axis. We see that after the 7 March changepoint, the runlength monotonically increases, indicating a stationary regime.

We see that RL[1]-PR resets approximately twice every day until the time of the changepoint. After that, there is no evidence of a changepoint (as provided by the hyperparameters and the modelling choices), so RL[1]-PR does not reset which translates to less adaptation for the period to the right of the changepoint.

Finally, we compare the error of predictions made by the competing methods. This is quantified in Figure 3.5, which shows a box-plot of the five-day MAE for each of the competing methods over the whole dataset, from March 2017 to November 2020. Our new RL[1]-OUPR* method has the lowest MAE.

Online classification with periodic drift

In this section we study the performance of C-ACI, CPP-OU, RL[1]-PR, and RL[1]-OUPR* for the classification experiment of Section 6.2 in Kurle et al. (2019). More precisely, in this experiment $x_{t,i} \sim \text{Unif}[-3,3]$ for $i \in \{1,2\}$, $\boldsymbol{x}_t = (x_{t,1}, x_{t,2}) \in \mathbb{R}^2$, $y_t \sim$

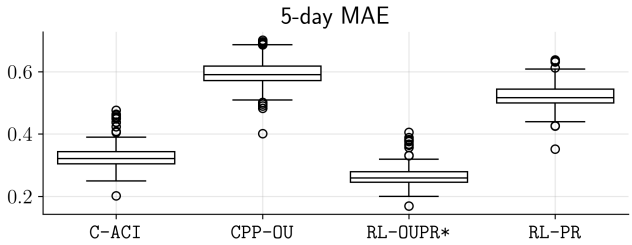

Figure 3.5: Distribution of the 5-day mean absolute error (MAE) for each of the competing methods on electricity forecasting over the entire period. For this calculation we split the dataset into consecutive buckets containing five days of data each, and for a given bucket we compute the average absolute error of the predictions and observations that fall within the bucket.

$\text{Bernoulli}(\sigma(\boldsymbol{\theta}_t^\intercal \boldsymbol{x}_t))$ with $\boldsymbol{\theta}_t^{(1)} = 10\ \sin(5^\circ\, t)$ and $\boldsymbol{\theta}_t^{(2)} = 10\ \cos(5^\circ\, t)$. Thus the unknown values of model parameters are slowly drifting deterministically according to sine and cosine functions. The timesteps go from 0 to 720.

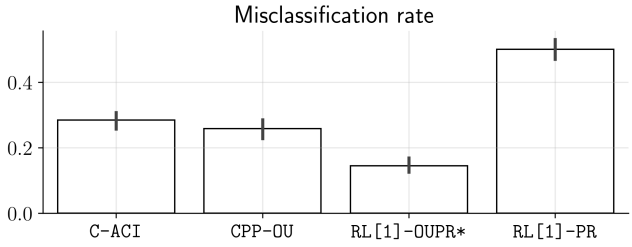

Figure 3.6: Misclassification rate of various methods on the online classification with periodic drift task.

Figure 3.6 summarises the results of the experiment where we show the misclassification rate (which is one minus the accuracy) for the competing methods. Our `RL[1]-OUPR*` method works the best, and signifcantly outperforms `RL[1]-PR`, since we use an OU drift process with a soft prior reset rather than assuming constant parameter with a hard prior rset.

We can improve the performance of `RL[K]-PR` if the number of hypotheses K increases, and if we vary the changepoint probability threshold κ, as shown in Figure 3.7. However, even then the performance of this method still does not match our method.

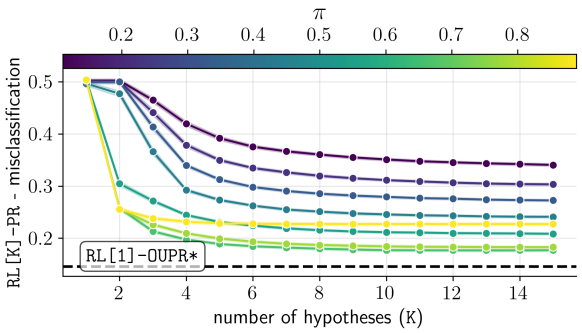

Figure 3.7: Accuracy of predictions for RL[1]-PR as a function of the number of hypothesis and the prior probability of a changepoint κ. The black dotted line is the performance of RL[1]-OUPR* reported in Figure 3.6.

Online classification with drift and jumps

In this section we study the performance of C-ACI, CPP-OU, RL[1]-PR, and RL[1]-OUPR* for an experiment with drift and sudden changes. More precisely, we assume that the parameters of a logistic regression problem evolve according to

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_{t-1} + \boldsymbol{\epsilon}_t & \text{w.p. } 1 - p_\epsilon, \\ \mathcal{U}[-2, 2]^2 & \text{w.p. } p_\epsilon, \end{cases} \tag{3.63}$$

with $p_\epsilon = 0.01$, $\boldsymbol{\theta}_0 \sim \mathcal{U}[-2, 2]^2$, and $\boldsymbol{\epsilon}_t$ is a zero-mean distributed random vector with isotropic covariance matrix $(0.01)^2\, \mathbf{I}_2$ (where $\mathbf{I}_2$ is a 2×2 identity matrix). Intuitively, this experiment has model parameters that drift slowly with occasional abrupt changes (at a rate of 0.01).

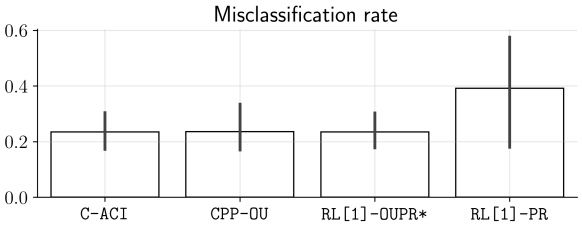

Figure 3.8: Misclassification rate of various methods on the online classification with drift and jumps task.

Figure 3.8 shows the misclassification rate among the competing methods. We observe that C-ACI, CPP-OU, and RL[1]-OUPR* have comparable performance, whereas RL[1]-PR is the method with highest misclassification rate.

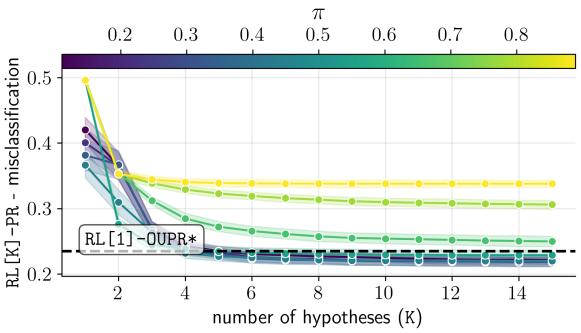

Figure 3.9: Accuracy of predictions for RL[K]−PR as a function of the number of hypotheses (K) and the probability of a changepoint κ. The black dotted line is the performance of RL[1]−OUPR* reported in Figure 3.8.

To explain this behaviour, Figure 3.9 shows the performance of RL[K]−PR as a function of number of hypotheses and prior probability of a changepoint κ. We observe that up to three hypotheses, the lowest misclassification error of RL[K]−PR is higher than that of RL[1]−OUPR*, which only considers one hypothesis. However, as we increase the number of hypotheses, the best performance for RL[K]−PR obtains a lower misclassification rate than RL[1]−OUPR*. This is in contrast to the results in Figure 5. Here, we see that with more hypotheses RL[K]−PR outperforms our new method at the expense of being more memory intensive.

3.5.2 Contextual bandits

In this section, we study the performance of C−ACI, CPP−OU, RL[1]−PR, and RL[1]−OUPR* for the simple Bernoulli bandit from Section 7.3 of Mellor and Shapiro (2013). More precisely, we consider a multi-armed bandit problem with 10 arms, 10,000 steps per simulation, and 100 simulations. The payoff of a given arm is the outcome of a Bernoulli random variable with unknown probability $\boldsymbol{\theta}_t = \min\{\max\{\boldsymbol{\theta}_{t-1} + 0.03\, Z_t, 0\}, 1\}$ for $\{Z_t\}_{t\in\{1,2,\ldots,10{,}000\}}$ independent and identically distributed standard normal random variables. We take $\boldsymbol{\theta}_0 \sim \text{Unif}[0,1]$ and use the same formulation for all ten arms with independence across arms. The observations are the rewards and there are no features (non-contextual).

The idea of using RL[1]−PR in multi-armed bandits problems was introduced in Mellor and Shapiro (2013). With this experiment, we extend the concept to other members of the BONE framework. We use Thompson sampling for each of the competing methods. Figure 3.10 shows the regret of using C−ACI, CPP−OU, RL[1]−PR, and RL[1]−OUPR* for the above multi-armed bandits problem. The results we obtain are similar to those of Section 3.5.1. This is because both problems have a similar drift structure.

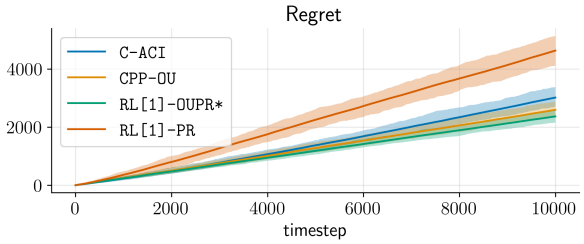

Figure 3.10: Regret of competing methods on the contextual bandits task. Confidence bands are computed with one hundred simulations.

3.5.3 Segmentation and prediction

In this section, we evaluate methods both in terms of their ability to "correctly" segment the observed output signal, and to do one-step-ahead predictions. Note that by "correct segmentation", we mean one that matches the ground truth data generating process. This metric can only be applied to synthetic data.

Autoregression with dependence across the segments

In this experiment, we consider the synthetic autoregressive dataset introduced in Section 2 of Fearnhead and Liu (2011), consisting of a set of one dimensional polynomial curves that are constrained to match up at segmentation boundaries, as shown in the top left of Figure 3.11.

We compare the performance of the three methods in the previous subsection. For this experiment, we employ a probability of a changepoint $\kappa = 0.01$. Since this dataset has dependence of the parameters across segments, we allow for the choice of (M.1: likelihood) to be influenced by the choice of (M.2: auxvar), i.e., our choice of model is given by $h(\boldsymbol{\theta}_t; \psi_t, \boldsymbol{x}_t)$. For this experiment, we take (M.2: auxvar) to be `RL` and our choice of (M.1: likelihood) becomes

$$h(\boldsymbol{\theta}_t; r_t, \boldsymbol{x}_{1:t}) = \boldsymbol{\theta}_t^{\mathsf{T}} \boldsymbol{h}(\boldsymbol{x}_{1:t}, r_t), \tag{3.64}$$

with $\boldsymbol{h}(\boldsymbol{x}_{1:t}, r_t) = [1, \Delta, \Delta^2]$, $\Delta = (x_t - x_{r_t})$, and $x_{r_t} \geq x_t$. Intuitively this represents a quadratic curve fit to the beginning x_{r_t} and end points x_t of the current segment. Given the form of (M.1: likelihood) in (3.64), here we do not consider `C-ACI` nor `CPP-OU`. Instead, we use runlength with moment-matching prior reset, i.e., `RL-MMPR` (see Table 3.4) which was designed for segmentation with dependence.

Figure 3.11 shows the results. On the right, we observe that `RL[1]-OUPR*` has the lowest RMSE. On the left, we plot the predictions of each method, so we can visualise the nature of their errors. For `RL[1]-PR`, the spikes occur because the method has many false positive beliefs in a changepoint occurring, and this causes breaks in the predictions due the explicit dependence of h on r_t and the hard parameter reset upon changepoints. For `RL-MMPR`, the slow adaptation (especially when $x_t \in [1, 5]$) is because the method

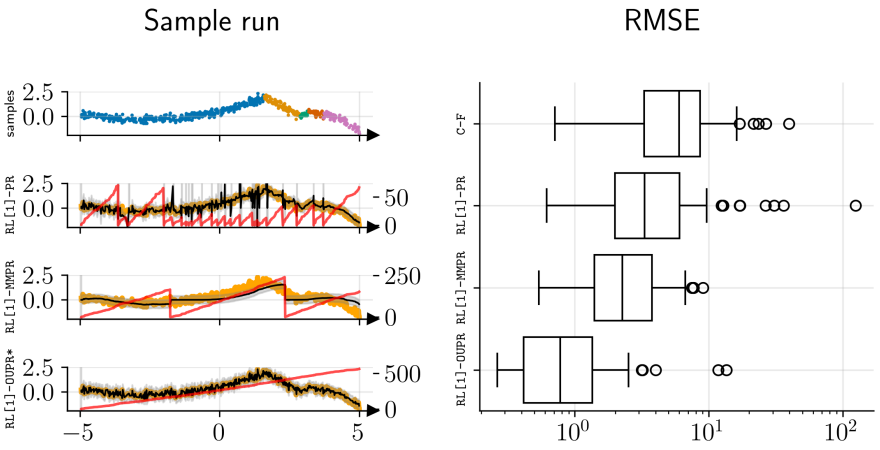

Figure 3.11: The **left panel** shows a sample run of the piecewise polynomial regression with dependence across segments. The x-axis is for the features, the (left) y-axis is for measurements together with the estimations made by `RL[1]-PR`, `RL-MMPR`, and `RL[1]-OUPR*`, the (right) y-axis is for the value of r_t under each model. The orange line denotes the true data-generating process and the red line denotes the value of the hypothesis `RL`. The **right panel** shows the RMSE of predictions over 100 trials.

does not adjust beliefs as quickly as it should. Our `RL[1]-OUPR*` method strikes a good compromise.

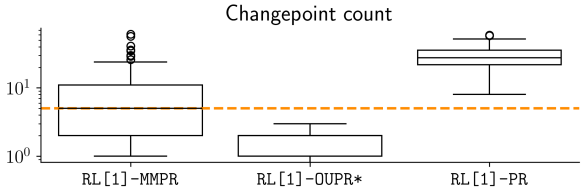

Figure 3.12: Count of changepoints over an experiment for 100 trials. The orange line shows the true number of changepoints for all trials.

Figure 3.12 shows the distribution (over 100 simulations) of the number of detected changepoints, i.e., instances where $\nu_t(r_t)$ with $r_t = 0$ is the highest. We observe that superior predictive performance in Figure 3.11 does not necessarily translate to a better segmentation capability. For example, the distribution produced by `RL-MMPR` sits around the actual number of changepoints (better at segmenting) whereas `RL[1]-OUPR*`, which is detecting far fewer changepoints, is the best performing prediction method. This reflects the discrepancy between the objectives of segmentation and prediction. For a more thorough analysis and evaluation of changepoint detection methods on time-series data, see Van den Burg and Williams (2020).

3.6 Conclusion

We introduced a unified Bayesian framework to perform online predictions in non-stationary environments, and showed how it covers many prior works. We also used our framework

to design a new method, RL[1]-OUPR*, which is suited to tackle prediction problems when the observations exhibit both abrupt and gradual changes. We hope to explore other novel variants and applications in future work.

Chapter 4

Robustness

In this chapter, we derive a novel, provably robust, and closed-form Bayesian update rule for online learning under a state-space model in the presence of outliers and misspecified measurement models. Our method combines generalised Bayesian (GB) inference with filtering methods such as the extended and ensemble Kalman filter. We use the former to show robustness and the latter to ensure computational efficiency in the case of nonlinear models.

Robustness to outliers is a critical requirement in sequential online learning, where noisy or corrupted observations can easily degrade performance. Recent (closed-form) methods addressing this challenge often rely on variational Bayes (VB), which, while effective, tend to be more computationally expensive than standard Kalman filter (KF) approaches due to the overhead of inner-loop computations. (see e.g., Wang et al. (2018); Tao and Yau (2023)). Although more classical (low-cost) alternatives exist, they are typically restricted to linear systems and the results may not extend to nonlinear or high-dimensional cases.

The method we propose is simple to implement, computationally efficient, and robust to both outliers and model misspecification. It offers performance comparable to, or better than, existing robust filtering approaches, including VB-based methods, but at a much lower computational cost. We demonstrate the effectiveness of our approach across a range of online learning tasks with outlier-contaminated observations.

4.1 The method

Our method is based on the GB approach where one modifies the update equation in (2.3) to use a loss function $\ell_t : \mathbb{R}^D \to \mathbb{R}$ in place of the likelihood of the measurement process. This gives the choice of (A.1: posterior) of the form

$$q(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) \propto \exp(-\ell_t(\boldsymbol{\theta}_t))\, q(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t-1}), \tag{4.1}$$

where $\psi_t = \{\}$. We propose to gain robustness to outliers in observation space by taking the loss function to be the model's negative log-likelihood scaled by a data-dependent weighting term

$$\ell_t(\boldsymbol{\theta}_t) = -W^2(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) \log p(\boldsymbol{y}_t|\boldsymbol{\theta}_t), \tag{4.2}$$

with $W : \mathbb{R}^o \times \mathbb{R}^o \to \mathbb{R}_{++}$ the weighting function—see Section 3.1.6 for examples—and $p(\boldsymbol{y}_t|\boldsymbol{\theta}_t)$ the choice of likelihood. We call our method the *weighted observation likelihood filter (WoLF)*. To specify an instance of our method, ones needs to define the likelihood $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t)$ and the weighting function W. In the next subsections, we show the flexibility of WoLF and derive weighted-likelihood-based KF and EKF algorithms. Setting $W(\boldsymbol{y}_t, \bar{\boldsymbol{y}}_t) = 1$ trivially recovers existing methods, but we will instead use non-constant weighting functions inspired by the work of Barp et al. (2019); Matsubara et al. (2022); Altamirano et al. (2023a,b).

4.2 Linear weighted observation likelihood filter

In this section, we present the WoLF method under a linear SSM (2.29). In particular, the following proposition provides a closed-form solution for the update step of WoLF under a linear measurement function and a Gaussian likelihood.

Proposition 4.1. *Consider the linear-Gaussian SSM* (2.29) *with weighting function* $W : \mathbb{R}^o \times \mathbb{R}^o \to \mathbb{R}$. *Then, the update step of WoLF with loss function* (4.2) *is given by Proposition 2.15 with* $\mathbf{R}_t^{-1}$ *replaced by* $\bar{\mathbf{R}}_t^{-1} = W^2(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)\, \mathbf{R}_t^{-1}$.

Proof. Let $w_t^2 := W^2(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)$. The loss function takes the form

$$\begin{aligned}
\ell_t(\boldsymbol{\theta}_t) &= -w_t^2 \log \mathcal{N}\left(\boldsymbol{y}_t \,|\, \mathbf{H}_t\boldsymbol{\theta}_t,\, \mathbf{R}_t\right) \\
&= \frac{1}{2}\left(\boldsymbol{y}_t - \mathbf{H}_t\boldsymbol{\theta}_t\right)^\intercal \left(\mathbf{R}_t/w_t^2\right)^{-1}\left(\boldsymbol{y}_t - \mathbf{H}_t\boldsymbol{\theta}_t\right) - \frac{w_t^2\, o}{2}\log \pi - \frac{w_t^2}{2}\log|\mathbf{R}_t| \\
&= \frac{1}{2}\left(\boldsymbol{y}_t - \mathbf{H}_t\boldsymbol{\theta}_t\right)^\intercal \bar{\mathbf{R}}_t^{-1}\left(\boldsymbol{y}_t - \mathbf{H}_t\boldsymbol{\theta}_t\right) + C,
\end{aligned} \tag{4.3}$$

with $\bar{\mathbf{R}}_t = \mathbf{R}_t/w_t^2$, and where $C = -\frac{w_t^2\, o}{2}\log \pi - \frac{w_t^2}{2}\log|\mathbf{R}_t|$ is a term that does not depend on $\boldsymbol{\theta}_t$. The remaining follows from the standard KF derivation. Note that the loss function does not correspond to the log-likelihood for a homoskedastic Gaussian model since $\bar{\mathbf{R}}_t$ may depend on all data, including $\boldsymbol{y}_t$. □

Figure 4.1 shows the weighted log-likelihood (4.3) for a univariate $\mathcal{N}(0, 1)$ Gaussian density as a function of the weighting term $w_t^2 \in (0, 1]$. We observe that a weighted log-likelihood resembles a heavy-tailed likelihood for $w_t < 1$.

The resulting predict and update steps for WoLF under linear dynamics and zero-mean Gaussians for the state and measurement process are shown in Algorithm 11.

The computational complexity of WoLF under linear dynamics matches that of the KF, i.e., $O(D^3)$. Alternative robust filtering algorithms require multiple iterations per

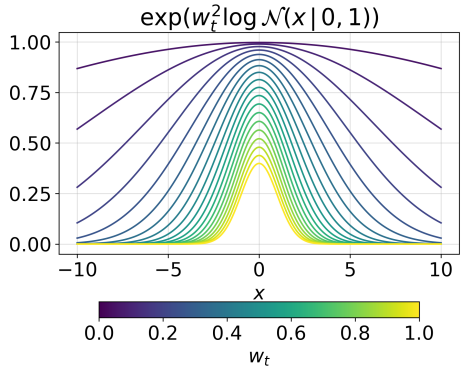

Figure 4.1: Weighted likelihood (unnormalised) for a standard Gaussian.

Algorithm 11 WoLF predict and update step

Require: $\mathbf{F}_t$, $\mathbf{Q}_t$ // predict step

$\boldsymbol{\mu}_{t|t-1} \leftarrow \mathbf{F}_t \, \boldsymbol{\mu}_{t-1}$

$\boldsymbol{\Sigma}_{t|t-1} \leftarrow \mathbf{F}_t \, \boldsymbol{\Sigma}_{t-1} \, \mathbf{F}_t^\intercal + \mathbf{Q}_t$

Require: $\boldsymbol{y}_t$, $\mathbf{H}_t$, $\mathbf{R}_t$ // update step

$\hat{\boldsymbol{y}}_t \leftarrow \mathbf{H}_t \, \boldsymbol{\mu}_{t|t-1}$

$w_t \leftarrow W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)$

$\boldsymbol{\Sigma}_t^{-1} \leftarrow \boldsymbol{\Sigma}_{t|t-1}^{-1} + w_t^2 \, \mathbf{H}_t^\intercal \, \mathbf{R}_t^{-1} \, \mathbf{H}_t$

$\mathbf{K}_t \leftarrow w_t^2 \, \boldsymbol{\Sigma}_t \, \mathbf{H}_t^\intercal \, \mathbf{R}_t^{-1}$

$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t \, (\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)$

measurement to achieve robustness and stability, making them significantly slower; see Table 4.1 for the computational complexity for the methods we consider, and Figure 4.5 for empirical comparisons.

4.3 Nonlinear weighted observation likelihood filter

The WoLF method readily extends to other learning algorithms. For example, a WoLF version of the EKF, which is obtained by introducing a weighting function to (2.42) yielding the approximate choice of (M.1: likelihood):

$$\log p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) = W^2(\boldsymbol{y}_t, \bar{\boldsymbol{y}}_t) \, \log \mathcal{N}\left(\boldsymbol{y}_t \,|\, \bar{\boldsymbol{y}}_t, \, \mathbf{R}_t\right). \tag{4.4}$$

We can also derive a novel outlier-robust exponentially-weighted moving average algorithm (see Section 4.7.5).

4.4 The choice of weighting function

Weighted likelihoods have a well-established history in Bayesian inference and have demonstrated their efficacy in improving robustness (Grünwald, 2012; Holmes and Walker, 2017;

Grünwald and van Ommen, 2017; Miller and Dunson, 2018; Bhattacharya et al., 2019; Alquier and Ridgway, 2020; Dewaskar et al., 2023). In this context, the corresponding posteriors are often referred to as fractional, tempered, or power posteriors. In most existing work, the determination of weights relies on heuristics and the assigned weights remain constant across all data points so that $W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = w \in \mathbb{R}$ for all t. In contrast, we dynamically incorporate information from the most recent observations without incurring additional computational costs by defining the weight as a function of the current observation $\boldsymbol{y}_t$ and its prediction $\hat{\boldsymbol{y}}_t = h_t(\boldsymbol{\mu}_{t|t-1})$, which is based on all of the past observations.

To define the weighting function, we take inspiration from previous work for dealing with outliers. In particular, Wang et al. (2018) proposed classifying robust filtering algorithms into two main types: *compensation-based* algorithms, which incorporate information from tail events into the model in a robust way (see, e.g., Huang et al., 2016; Agamennoni et al., 2012), and *detect-and-reject* algorithms, which assume that outlier observations bear no useful information and thus are ignored (see, e.g., Wang et al., 2018; Mu and Yuen, 2015). Below we show how both of these strategies can be implemented using our WoLF method by merely changing the weighting function.

Inverse multi-quadratic weighting function: As an example of a compensation-based method, we follow Altamirano et al. (2023b) and use the Inverse Multi-Quadratic (IMQ) weighting, which in our SSM setting is

$$W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = \left(1 + \frac{||\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t||_2^2}{c^2}\right)^{-1/2}, \tag{4.5}$$

where $c > 0$ is the soft threshold and $\|\cdot\|_2$ denotes the l_2 norm. We call WoLF with IMQ weighting "WoLF-IMQ".

Mahalanobis-based weighting function: The l_2 norm in the IMQ can be modified to account for the covariance structure of the measurement process by replacing it with the Mahalanobis distance between $\boldsymbol{y}_t$ and $\hat{\boldsymbol{y}}_t$:

$$W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = \left(1 + \frac{\|\mathbf{R}_t^{-1/2}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\|_2^2}{c^2}\right)^{-1/2}. \tag{4.6}$$

We call WoLF with this weighting function the WoLF-MD method. This type of weighted IMQ function has been used extensively in the kernel literature (see e.g. Chen et al., 2019; Detommaso et al., 2018; Riabiz et al., 2022).

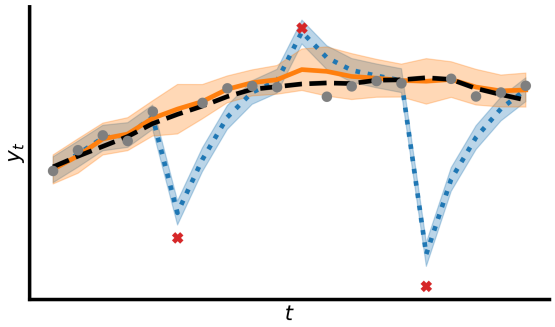

Figure 4.2: First state component of the SSM (4.42). The grey dots are measurements sampled from (4.42) and the red crosses are measurements sampled from an outlier measurement process. The dotted blue line shows the KF posterior mean estimate and the solid orange line shows our proposed WoLF posterior mean estimate. The regions around the posterior mean cover two standard deviations. For comparison, the dashed black line shows the true sampled state process.

Threshold Mahalanobis-based weighting function: As an example of a detect-and-reject method, we consider

$$W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = \begin{cases} 1 & \text{if } \|\mathbf{R}_t^{-1/2}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\|_2^2 \leq c, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

with $c > 0$ the fixed threshold. The weighting function (4.7) corresponds to ignoring information from estimated measurements whose Mahalanobis distance to the true measurement is larger than some predefined threshold c. In the linear setting, this weighting function is related to the benchmark method employed in Ting et al. (2007). We refer to WoLF with this weighting function as "WoLF-TMD".

The proposed weighting functions — the IMQ, the MD, and the TMD — are defined such that $W : \mathbb{R}^o \times \mathbb{R}^o \to [0, 1]$ and therefore can only down-weight observations. This means that our updates are always conservative, i.e., our posteriors will be wider in the presence of outliers (see Figure 4.2 for an example).

4.5 Robustness properties

In this section, we prove the outlier-robustness for WoLF-type methods. We use the classical framework of Huber (1981). Consider measurements $\boldsymbol{y}_{1:t}$. We measure the influence of a contamination $\boldsymbol{y}_t^c$ by examining the divergence between the posterior with the original observation $\boldsymbol{y}_t$ and the posterior with the contamination $\boldsymbol{y}_t^c$, which is allowed to be arbitrarily large. As a function of $\boldsymbol{y}_t^c$, this divergence is called the *posterior influence function* (PIF) and was studied in Matsubara et al. (2022); Altamirano et al. (2023a,b). Following Altamirano et al. (2023b), we consider the Kullback-Leibler (KL) divergence.

Definition 4.2 (posterior influence function)**.** Let $\mathcal{D}_{1:t} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_t, \boldsymbol{y}_t)\}$ and $\mathcal{D}_{1:t}^c = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_t, \boldsymbol{y}_t^c)\}$ be two datasets. Consider the choice of (A.1: posterior)

q. The posterior influence function (PIF) of the measurement $\boldsymbol{y}_t^c$ under q, given the dataset $\mathcal{D}_{1:t}$, is defined by

$$\mathrm{PIF}_q(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) = \mathbf{D}_{\mathrm{KL}}\left(q(\boldsymbol{\theta}_t \mid \mathcal{D}_{1:t}^c) \,||\, q(\boldsymbol{\theta}_t \mid \mathcal{D}_{1:t})\right). \tag{4.8}$$

Definition 4.3 (outlier robust posterior)**.** We denote the choice of (A.1: posterior) q to be outlier robust to measurements (or simply outlier robust) if, for any given $\boldsymbol{x}_t \in \mathbb{R}^M$, the effect of the contamination $\boldsymbol{y}_t^c$ is bounded, i.e.,

$$\sup_{\boldsymbol{y}_t^c \in \mathbb{R}^o} \mathrm{PIF}_q(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) < \infty. \tag{4.9}$$

Theorem 4.4. *Let q_{LG} be the linear Gaussian update function shown in Section 2.6.1, then, q_{LG} has an unbounded PIF and is not outlier robust.*

Theorem 4.5. *Let $q_{\mathrm{W-LG}}$ be the generalised posterior (4.1) with loss function (4.3) and weight W such that $\sup_{\boldsymbol{y}_t \in \mathbb{R}^d} W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) < \infty$ and $\sup_{\boldsymbol{y}_t \in \mathbb{R}^d} W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)^k \, \|\boldsymbol{y}_t\|_2 < \infty$ for $k \geq 2$. Then, $q_{\mathrm{W-LG}}$ has a bounded PIF and is, therefore, outlier robust.*

In particular, the conditions are satisfied when W is (4.5), (4.6), or (4.7), which are the focus of this work.

4.6 Proof of robustness

In this section, we prove Theorems 4.4 and 4.5 stated above. We begin by stating some remarks that we use to prove the theorems.

Remark 4.6. Let $\mathbf{A}$ be a $D \times D$ Hermitian matrix. By the spectral theorem, the eigenvalues of $\mathbf{A}$ are real (Tao, 2010). We denote these eigenvalues as $\sigma_{\max}(\mathbf{A}) = \sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_D(\mathbf{A}) = \sigma_{\min}(\mathbf{A})$, where $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ represent the largest and smallest eigenvalues, respectively.

Remark 4.7. Let $\mathbf{A}$ be a $D \times D$ positive semidefinite matrix. Then, for every $\boldsymbol{x} \in \mathbb{R}^D$,

$$\sqrt{\boldsymbol{x}^\intercal \, \mathbf{A} \, \boldsymbol{x}} = \|\mathbf{A}^{1/2} \, \boldsymbol{x}\|_2 \leq \|\mathbf{A}^{1/2}\|_2 \, \|\boldsymbol{x}\|_2 = \sqrt{\sigma_{\max}(\mathbf{A})} \, \|\boldsymbol{x}\|_2. \tag{4.10}$$

Here, $\|\mathbf{A}\|_2 = \sqrt{\sigma_{\max}(\mathbf{A}\,\mathbf{A})}$ is the spectral norm. See Coppersmith et al. (1997) for details.

Remark 4.8. Let $\mathbf{A}$ be a $D \times D$ positive semidefinite matrix. Then,

$$\|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}\,\mathbf{A}) \leq \mathrm{Tr}(\mathbf{A})^2. \tag{4.11}$$

See Lemma 2.3 in Yang and Feng (2002) for details.

Remark 4.9. The Frobenius norm is a submutiplicative matrix norm and compatible with the L2 vector norm, i.e., for any two $D \times D$ Hermitian matrices $\mathbf{A}$ and $\mathbf{B}$, and a vector $\boldsymbol{z} \in \mathbb{R}^D$, $\|\mathbf{A}\,\mathbf{B}\|_F \leq \|\mathbf{A}\|_F \, \|\mathbf{B}\|_F$ and $\|\mathbf{A}\boldsymbol{x}\|_2 \leq \|\mathbf{A}\|_F \, \|\boldsymbol{x}\|_2$.

See Section 5.6 in Horn and Johnson (2012) for a proof on submultiplicativity. Next, compatibility follows from Remark 4.7 and properties of the trace. In particular: $\|\mathbf{A}\,\boldsymbol{x}\|_2 \leq \|\mathbf{A}\|_2\,\|\boldsymbol{x}\|_2 = \sqrt{\sigma_{\max}(\mathbf{A}\,\mathbf{A})}\,\|\boldsymbol{x}\|_2 \leq \sqrt{\sum_{d=1}^{D}\sigma_d(\mathbf{A}\mathbf{A})} = \|\mathbf{A}\|_F\,\|\boldsymbol{x}\|_2$.

Remark 4.10. Let $\mathbf{A}$ and $\mathbf{B}$ be $D \times D$ Hermitian matrices. From Weyl's inequality, it follows that

$$\sigma_{\min}(\mathbf{A}+\mathbf{B})^{-1} \leq (\sigma_{\min}(\mathbf{A}) + \sigma_{\min}(\mathbf{B}))^{-1}\,. \tag{4.12}$$

See Tao (2010) for details.

Remark 4.11. For any two positive semi-definite matrices $\mathbf{A}$ and $\mathbf{B}$,

$$|\mathbf{A}| + |\mathbf{B}| \leq |\mathbf{A}+\mathbf{B}|. \tag{4.13}$$

The result follows from $|\mathbf{A}+\mathbf{B}| = |\mathbf{A}|\,|\mathbf{I}_D + \mathbf{A}^{-1}\mathbf{B}| \geq |\mathbf{A}|\,(1+|\mathbf{A}^{-1}\mathbf{B}|) = |\mathbf{A}| + |\mathbf{B}|$, where we used the fact that, for a matrix $\mathbf{C}$, $\sigma_d(\mathbf{C}+c) = c + \sigma_d(\mathbf{C})$ for $d \in \{1,\ldots,D\}$. So that $|\mathbf{A}^{-1}\,\mathbf{B}+1| = \prod_{d=1}^{D}(1+\sigma_d(\mathbf{A}^{-1}\,\mathbf{B})) \geq 1 + \prod_{d=1}^{D}\sigma_d(\mathbf{A}^{-1}\,\mathbf{B}) = 1 + |\mathbf{A}^{-1}\,\mathbf{B}|$.

4.6.1 Proof of Theorem 4.4 — KF is not outlier-robust

Let $q_{\mathrm{LG}}(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) = \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and $q_{\mathrm{LG}}(\boldsymbol{\theta}_t \,|\, \mathcal{D}^c_{1:t}) = \mathcal{N}(\boldsymbol{\theta}_t \,|\, \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ be the posterior densities under q_{LG}, conditioned on the uncorrupted dataset $\mathcal{D}_{1:t}$ and the corrupted dataset $\mathcal{D}^c_{1:t}$, respectively. Furthermore, to avoid cases where the PIF is bounded through the choice of $\mathbf{H}_t$, suppose that $\|\mathbf{K}\|_F \neq 0 \iff \mathbf{H}_t \neq \mathbf{0}$.

We show that q_{LG} is not outlier robust, i.e.,

$$\sup_{\boldsymbol{y}_t^c} \mathrm{PIF}_{q_{\mathrm{LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) = \infty. \tag{4.14}$$

To establish this, we first derive the explicit form of the PIF under the choice of q_{LG}. The result is presented in the lemma below.

Lemma 4.12. *The PIF between the posterior density q_{LG} under the corrupted and uncorrupted dataset takes the form*

$$\mathrm{PIF}_{q_{\mathrm{LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) = \tfrac{1}{2}\left[\mathbf{K}_t\,(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right]^{\intercal}\,\boldsymbol{\Sigma}_t^{-1}\left[\mathbf{K}_t\,(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right]. \tag{4.15}$$

where $\mathbf{K}_t = \boldsymbol{\Sigma}_t\,\mathbf{H}_t^{\intercal}\,\mathbf{R}_t^{-1}$

Proof. Following proposition 2.15, the update equations for the contaminated and uncontaminated densities take the form

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\,(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\,, \quad & \bar{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\,, \\ \boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^{\intercal}\mathbf{R}_t^{-1}\mathbf{H}_t, \quad & \bar{\boldsymbol{\Sigma}}_t^{\,-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t^{\intercal}\,\mathbf{R}_t^{-1}\,\mathbf{H}_t, \end{aligned} \tag{4.16}$$

where we observe that $\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t$.

Next, following Proposition 2.8, we write the PIF as

$$\begin{aligned}\mathrm{PIF}_{q_{\mathrm{LG}}}(\boldsymbol{y}_t^c, \boldsymbol{y}_{1:t}) &= \frac{1}{2}\left(\mathrm{Tr}\left(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_t\right) - D + (\boldsymbol{\mu}_t - \boldsymbol{\mu}_t^c)^\intercal \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_t^c) + \ln\left(\frac{\det \boldsymbol{\Sigma}_t}{\det \boldsymbol{\Sigma}_t}\right)\right) \\ &= \mathrm{Tr}\left(\mathbf{I}_D\right) - D + \tfrac{1}{2}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_t^c)^\intercal \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_t^c) \\ &= \tfrac{1}{2}(\boldsymbol{y}_t - \boldsymbol{y}_t^c)^\intercal \mathbf{K}_t^\intercal \boldsymbol{\Sigma}_t^{-1}\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c) \\ &= \tfrac{1}{2}\left[\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right]^\intercal \boldsymbol{\Sigma}_t^{-1}\left[\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right].\end{aligned} \tag{4.17}$$

□

Proof of Theorem 4.4

Proof. First write an upper bound for the PIF following Lemma 4.12, Remark 4.9, and Remark 4.9:

$$\begin{aligned}\mathrm{PIF}_{q_{\mathrm{LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) &= \tfrac{1}{2}\left[\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right]^\intercal \boldsymbol{\Sigma}_t^{-1}\left[\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\right] \\ &\leq \tfrac{1}{2}\sigma_{\max}(\boldsymbol{\Sigma}_t^{-1})\|\mathbf{K}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^c)\|_2^2 \\ &\leq \tfrac{1}{2}\sigma_{\max}(\boldsymbol{\Sigma}_t^{-1})\|\mathbf{K}_t\|_F^2\,\|\boldsymbol{y}_t - \boldsymbol{y}_t^c\|_2^2 \\ &= C_1\,\|\boldsymbol{y}_t - \boldsymbol{y}_t^c\|_2^2.\end{aligned} \tag{4.18}$$

Then, since $\|\mathbf{K}_t\|_F^2 \neq 0$ by assumption,

$$\sup_{\boldsymbol{y}_t^c} \mathrm{PIF}_{q_{\mathrm{LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) \leq \sup_{\boldsymbol{y}_t^c} C_1\,\|\boldsymbol{y}_t - \boldsymbol{y}_t^c\|_2^2 = \infty. \tag{4.19}$$

□

4.6.2 Proof of Theorem 4.5 — WoLF is outlier-robust

Let W be a weighting function such that $\sup_{\boldsymbol{y}_t \in \mathbb{R}^o} W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) < \infty$ and $\sup_{\boldsymbol{y}_t \in \mathbb{R}^o} W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)^k\,\|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\|_2 < \infty$ for $k \geq 2$. For simplicity, denote $w_t := W(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)$ and $\bar{w}_t := W(\boldsymbol{y}_t^c, \hat{\boldsymbol{y}}_t)$. Let $q_{\mathrm{W-LG}}(\boldsymbol{\theta}_t \mid \mathcal{D}_{1:t}) = \mathcal{N}(\boldsymbol{\theta}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and $q_{\mathrm{W-LG}}(\boldsymbol{\theta}_t \mid \mathcal{D}_{1:t}^c) = \mathcal{N}(\boldsymbol{\theta}_t \mid \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ be the posterior densities under $q_{\mathrm{W-LG}}$, conditioned on the uncorrupted dataset $\mathcal{D}_{1:t}$ and the corrupted dataset $\mathcal{D}_{1:t}^c$, respectively, using the weighting function W.

Our goal is to show that q_{LG} is outlier robust, i.e.,

$$\sup_{\boldsymbol{y}_t^c} \mathrm{PIF}_{q_{\mathrm{W-LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) < \infty. \tag{4.20}$$

To establish this, we first derive the explicit form of the PIF under the choice of $q_{\mathrm{W-LG}}$. The result is presented in the lemma below.

Lemma 4.13. *The PIF between the posterior density $q_{\mathrm{W-LG}}$ under the corrupted and*

uncorrupted dataset takes the form

$$\mathrm{PIF}_{q_{\mathrm{W-LG}}}(\boldsymbol{y}_t^c, \mathcal{D}_{1:t}) = \frac{1}{2}\Bigg(\underbrace{\mathrm{Tr}\left(\bar{\boldsymbol{\Sigma}}_t{}^{-1}\boldsymbol{\Sigma}_t\right)}_{(\mathrm{T}.1)} - D + \underbrace{(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t)^\intercal\, \bar{\boldsymbol{\Sigma}}_t^{-1}\,(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t)}_{(\mathrm{T}.2)} + \underbrace{\log\left(\frac{\det \bar{\boldsymbol{\Sigma}}_t}{\det \boldsymbol{\Sigma}_t}\right)}_{(\mathrm{T}.3)}\Bigg), \tag{4.21}$$

where

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + w_t^2\,\mathbf{K}_t\,(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\,, \quad & \bar{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_{t|t-1} + \bar{w}_t{}^2\,\bar{\mathbf{K}}_t\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\,, \\ \boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + w_t^2\mathbf{H}_t^\intercal\mathbf{R}_t^{-1}\mathbf{H}_t, & \bar{\boldsymbol{\Sigma}}_t{}^{-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \bar{w}_t{}^2\,\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t, \end{aligned} \tag{4.22}$$

Proof. The proof follows directly from Algorithm 11 and Proposition 2.8 on the KL divergence between two multivariate Gaussian densities. □

Lemma 4.13 establishes that if the PIF under $q_{\mathrm{W-LG}}$ is outlier-robust, then each of the terms $(\mathrm{T}.1)$, $(\mathrm{T}.2)$, and $(\mathrm{T}.3)$ must remain finite (bounded) under the supremum over all possible values of $\boldsymbol{y}_t^c$. Before demonstrating that this is indeed the case, we first present two auxiliary lemmas that will assist in proving the boundedness of these terms.

Lemma 4.14. *The Frobenius norm of the corrupted posterior covariance $\bar{\boldsymbol{\Sigma}}_t$ shown in (4.22) is bounded above by*

$$\|\bar{\boldsymbol{\Sigma}}_t\|_{\mathrm{F}} \leq \frac{D}{\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + w_t^2\,\sigma_{\min}(\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t))}. \tag{4.23}$$

Proof. Let $\bar{\boldsymbol{\Sigma}}_t$ be the posterior covariance under the corrupted dataset $\mathcal{D}_{1:t}^c$. Following Remark 4.6, Remark 4.8, and the properties of the trace, we obtain

$$\|\bar{\boldsymbol{\Sigma}}_t\|_F \leq \mathrm{Tr}\left(\bar{\boldsymbol{\Sigma}}_t\right) = \sum_{d=1}^{D}\sigma_d(\bar{\boldsymbol{\Sigma}}_t) \leq D\,\sigma_{\max}(\bar{\boldsymbol{\Sigma}}_t). \tag{4.24}$$

Next, an upper bound for $\sigma_{\max}(\bar{\boldsymbol{\Sigma}}_t)$ is given by

$$\begin{aligned} \sigma_{\max}(\bar{\boldsymbol{\Sigma}}_t) &= \sigma_{\max}\left(\left(\boldsymbol{\Sigma}_{t|t-1}^{-1} + \bar{w}_t^2\,\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t\right)^{-1}\right) \\ &= \left(\sigma_{\min}\left(\boldsymbol{\Sigma}_{t|t-1}^{-1} + \bar{w}_t^2\,\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t\right)\right)^{-1} \\ &\leq \left(\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + \sigma_{\min}\left(\bar{w}_t^2\,\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t\right)\right)^{-1} \\ &= \left(\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + \bar{w}_t^2\,\sigma_{\min}\left(\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t\right)\right)^{-1}. \end{aligned} \tag{4.25}$$

In (4.25), the second equality is a consequence of the positive definiteness (symmetry) of $\bar{\boldsymbol{\Sigma}}_t$, and the upper bound follows from Remark 4.10.

The desired result follows as a consequence of (4.24) and (4.25). □

Lemma 4.15. *The L2 norm of the difference between the corrupted and uncorrupted posterior mean is bounded above by*

$$\|\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\|_2^2 \leq C_3 \left(w_t^2 \|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|\right)^2 + C_1, \tag{4.26}$$

where C_1 and C_3 are real-valued elements that do not depend on $\boldsymbol{y}_t^c$.

Proof. We begin by expanding (4.26):

$$\begin{aligned}
\|\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\|_2^2 &= \|\mathbf{K}_t\,(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t) - \bar{\mathbf{K}}_t\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2 \\
&\leq \|\mathbf{K}_t\,(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\|_2^2 + \|\bar{\mathbf{K}}_t\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2 \\
&= \|\bar{w}_t^2\,\bar{\boldsymbol{\Sigma}}_{t|t-1}\,\mathbf{H}_t\,\mathbf{R}_t^{-1}\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2 + C_1 \\
&= \bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\,\mathbf{H}_t\,\mathbf{R}_t^{-1}\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2 + C_1,
\end{aligned} \tag{4.27}$$

where $C_1 = \|\mathbf{K}_t\,(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)\|_2^2$ does not depend on $\boldsymbol{y}_t^c$.

Next, following Remark 4.9, we bound $\bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\,\mathbf{H}_t\,\mathbf{R}_t^{-1}\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2$ as follows

$$\begin{aligned}
&\bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\,\mathbf{H}_t\,\mathbf{R}_t^{-1}\,(\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t)\|_2^2 \\
&\leq \bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\|_F^2\,\|\mathbf{H}_t\,\mathbf{R}_t^{-1}\|_2^2\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t{}^2\|_2^2 \\
&= C_2\,\bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\|_F^2\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t{}^2\|_2^2,
\end{aligned} \tag{4.28}$$

where $C_2 = \|\mathbf{H}_t\,\mathbf{R}_t^{-1}\|_2^2$.

Finally, using Lemma 4.14, an upper bound for (4.28) is given by

$$\begin{aligned}
&C_2\,\bar{w}_t^4\,\|\bar{\boldsymbol{\Sigma}}_{t|t-1}\|_F^2\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t{}^2\|_2^2 \\
&\leq C_2\,\bar{w}_t^4 \left(\frac{D}{\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + w_t^2\,\sigma_{\min}(\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t))}\right)^2 \|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2^2 \\
&= C_2\,D^2\,\frac{\bar{w}_t^4}{\left(\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + w_t^2\,\sigma_{\min}(\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t))\right)^2}\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2^2 \\
&= C_2\,D^2\,\frac{1}{\left(w_t^{-2}\,\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) + \sigma_{\min}(\mathbf{H}_t^\intercal\,\mathbf{R}_t^{-1}\,\mathbf{H}_t))\right)^2}\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2^2 \\
&\leq \frac{C_2\,D^2}{\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1})}\,\bar{w}_t^4\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2^2 + C_1 \\
&= C_3\,\left(\bar{w}_t^2\,\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2\right)^2,
\end{aligned} \tag{4.29}$$

where $C_3 = \frac{C_2\,D^2}{\sigma_{\min}(\boldsymbol{\Sigma}_{t|t-1}^{-1})}$. □

With the auxiliary lemmas established and following the remarks above, we now proceed to demonstrate that each of the terms (T.1), (T.2), and (T.3) are indeed bounded, as required for outlier robustness. Since the key components of the following propositions were already defined in Proposition 4.13, we proceed directly by stating each term and

showing that it is bounded over all possible values of $\boldsymbol{y}_t^c$.

Proposition 4.16 (Bound for T.1)**.**

$$\sup_{\boldsymbol{y}_t^c} \operatorname{Tr}\left(\bar{\boldsymbol{\Sigma}}_t^{-1} \boldsymbol{\Sigma}_t\right) < \infty. \tag{4.30}$$

Proof. Using (4.22), (T.1) in (4.21) can be written as

$$\begin{aligned} \operatorname{Tr}\left(\bar{\boldsymbol{\Sigma}}_t^{-1} \boldsymbol{\Sigma}_t\right) &= \operatorname{Tr}\left(\left[\boldsymbol{\Sigma}_{t|t-1}^{-1} + \bar{w}_t^2\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t\right]\, \boldsymbol{\Sigma}_t\right) \\ &= \operatorname{Tr}\left(\boldsymbol{\Sigma}_{t|t-1}^{-1}\, \boldsymbol{\Sigma}_t\right) + \bar{w}_t^2\, \operatorname{Tr}\left(\mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1}\, \mathbf{H}_t\, \boldsymbol{\Sigma}_t\right) \end{aligned} \tag{4.31}$$

Since $\sup_{\boldsymbol{y}_t^c} \bar{w}_t$ is bounded, it follows that

$$\sup_{\boldsymbol{y}_t^c} \operatorname{Tr}\left(\bar{\boldsymbol{\Sigma}}_t^{-1} \boldsymbol{\Sigma}_t\right) < \infty. \tag{4.32}$$

□

Proposition 4.17 (Bound for T.2)**.**

$$\sup_{\boldsymbol{y}_t^c} \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \bar{\boldsymbol{\Sigma}}_t^{-1} \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right) < \infty \tag{4.33}$$

Proof. Begin by expanding the left hand side of (4.33):

$$\begin{aligned} &\left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \bar{\boldsymbol{\Sigma}}_t^{-1} \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right) \\ &= \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \left(\boldsymbol{\Sigma}_{t|t-1} + \bar{w}_t^2\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} \mathbf{H}_t\right) \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right) \\ &= \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \boldsymbol{\Sigma}_{t|t-1}^{-1} \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right) + \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \left(\bar{w}_t^2\, \mathbf{H}_t^\intercal\, \mathbf{R}_t^{-1} \mathbf{H}_t\right) \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right). \end{aligned} \tag{4.34}$$

Next, we bound each of the terms in (4.34).

For the first term, we obtain

$$\begin{aligned} &\left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right)^\intercal \boldsymbol{\Sigma}_{t|t-1}^{-1} \left(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\right) \\ &\leq \sigma_{\max}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) \|\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t\| \\ &\leq \sigma_{\max}(\boldsymbol{\Sigma}_{t|t-1}^{-1}) \left(C_3 \left(\bar{w}_t^2 \left\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\right\|_2\right)^2 + C_1\right) \\ &= C_5 \left(\bar{w}_t^2 \left\|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\right\|_2\right)^2 + C_6, \end{aligned} \tag{4.35}$$

where we make use of Remark 4.7 and Lemma 4.15.

By assumption, $\sup_{\boldsymbol{y}_t^c} \bar{w}_t^2 \|\boldsymbol{y}_t^c - \boldsymbol{y}_t\| < \infty$, which ensures that the supremum over $\boldsymbol{y}_t^c$ for (4.35) is bounded.

Similarly, for the second term, suppose $\sigma_{\max}(\mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\mathbf{H}_t) > 0$, then

$$\begin{aligned}
&(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t)^\intercal \left(\bar{w}_t^2\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\mathbf{H}_t\right)(\boldsymbol{\mu}_t - \bar{\boldsymbol{\mu}}_t) \\
&\leq \bar{w}_t^2\, \sigma_{\max}(\mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\mathbf{H}_t)\left(C_3\, \left(\bar{w}_t^2\, \|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2\right)^2 + C_1\right) \\
&\leq C_7\, \left(\bar{w}_t^3\, \|\boldsymbol{y}_t^c - \hat{\boldsymbol{y}}_t\|_2\right)^2 + C_8.
\end{aligned} \tag{4.36}$$

Again, by assumption, $\sup_{\boldsymbol{y}_t^c} \bar{w}_t^3 \|\boldsymbol{y}_t^c - \boldsymbol{y}_t\| < \infty$, which ensures that the supremum over $\boldsymbol{y}_t^c$ for (4.36) is bounded. □

Proposition 4.18 (Bound for T.3)**.**

$$\sup_{\boldsymbol{y}_t^c} \log\left(\frac{|\bar{\boldsymbol{\Sigma}}_t|}{|\boldsymbol{\Sigma}_t|}\right) < \infty \tag{4.37}$$

Proof. We first expand (T.3) to obtain

$$\log\left(\frac{|\bar{\boldsymbol{\Sigma}}_t|}{|\boldsymbol{\Sigma}_t|}\right) = \log|\bar{\boldsymbol{\Sigma}}_t| - \log|\boldsymbol{\Sigma}_t|, \tag{4.38}$$

which shows that we only to bound $\log|\bar{\boldsymbol{\Sigma}}_t|$. For this, consider

$$\begin{aligned}
\log|\bar{\boldsymbol{\Sigma}}_t| &= \log|(\boldsymbol{\Sigma}_{t|t-1} + \bar{w}_t^2\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t)^{-1}| \\
&= -\log|\boldsymbol{\Sigma}_{t|t-1}^{-1} + \bar{w}_t^2\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t| \\
&\leq -\log\left(|\boldsymbol{\Sigma}_{t|t-1}^{-1}| + |\bar{w}_t^2\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|\right) \\
&= -\log\left(|\boldsymbol{\Sigma}_{t|t-1}^{-1}| + \bar{w}_t^{2D}|\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|\right)
\end{aligned} \tag{4.39}$$

Since $\boldsymbol{\Sigma}_{t|t-1}^{-1}$ is positive definite and $\mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t$ is positive semidefinite, we upper bound (4.39) further by taking the minimum of the two terms, i.e.,

$$\log|\bar{\boldsymbol{\Sigma}}_t| \leq -\log\left(\min\{|\boldsymbol{\Sigma}_{t|t-1}^{-1}|,\, \bar{w}_t^{2D}|\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|\}\right). \tag{4.40}$$

If the smallest determinant is given by the posterior predictive covariance $\boldsymbol{\Sigma}_{t|t-1}$, then (4.39) is bounded. Conversely, if the smallest determinant is $\bar{w}_t^{2D}|\, \mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|$, we obtain

$$\begin{aligned}
\log|\bar{\boldsymbol{\Sigma}}_t| &\leq -\log\left(\bar{w}_t^{2D}|\mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|\right) \\
&= -2D\log(\bar{w}_t) - \log|\mathbf{H}_t^\intercal \mathbf{R}_t^{-1}\, \mathbf{H}_t|.
\end{aligned} \tag{4.41}$$

Finally, note that $\sup_{\boldsymbol{y}_t^c} \bar{w}_t < \infty$ implies $\sup_{\boldsymbol{y}_t^c} \log \bar{w}_t < \infty$. So that (4.41) is bounded. □

Proof of Theorem 4.5

Proof. The proof follows from Propositions 4.16, 4.17, and 4.18. □

4.7 Experiments

In this section, we study the performance of the WoLF algorithm in multiple experiments.

For our robust baselines, we make use of three methods that are representative of recent state-of-the-art approaches to robust filtering: the Bernoulli KF of Wang et al. (2018) (`KF-B`), which is an example of a detect-and-reject strategy; and the inverse-Wishart filter of Agamennoni et al. (2012) (`KF-IW`), which is an example of a compensation-based strategy. The `KF-B` and `KF-IW` are deterministic and optimise a VB objective to compute a Gaussian approximation to the state posterior. For the neural network fitting problem, we also consider a variant of online gradient descent (`OGD`) based on Adam (Kingma and Ba, 2015), which uses multiple inner iterations per step (measurement). This method does scale to high-dimensional state spaces, but only gives a maximum a posteriori (MAP) estimate and is not as sample efficient as a robust Bayesian filter.

For experiments where KF or EKF is used as the baseline, we consider the following WoLF variants: (i) the WoLF version with inverse multi-quadratic weighting function (`WoLF-IMQ`), (ii) the thresholded WoLF with Mahalanobis-based weighting function (`WoLF-TMD`).

Method	Cost	#HP	Ref
`KF`	$O(D^3)$	0	Kalman (1960)
`KF-B`	$O(I\,D^3)$	3	Wang et al. (2018)
`KF-IW`	$O(I\,D^3)$	2	Agamennoni et al. (2012)
`OGD`	$O(I\,D^2)$	2	Bencomo et al. (2023)
`WoLF-IMQ`	$O(D^3)$	1	(Ours)
`WoLF-TMD`	$O(D^3)$	1	(Ours)

Table 4.1: Computational complexity of the update step, assuming $d \leq D$ and assuming linear dynamics. Here, I is the number of inner iterations, #HP refers to the number of hyperparameters we tune, and "Cost" refers to the computational complexity.

4.7.1 Robust KF for tracking a 2D object

We consider the classical problem of estimating the position of an object moving in 2D with constant velocity, which is commonly used to benchmark tracking problems (see e.g., Example 8.2.1.1 in Murphy (2023) or Example 4.5 in Särkkä and Svensson (2023)). The SSM takes the form

$$\begin{aligned} p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) &= \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \mathbf{F}_t\boldsymbol{\theta}_{t-1},\, \mathbf{Q}_t\right), \\ p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) &= \mathcal{N}\left(\boldsymbol{y}_t \,|\, \mathbf{H}_t\boldsymbol{\theta}_t,\, \mathbf{R}_t\right), \end{aligned} \tag{4.42}$$

where $\mathbf{Q}_t = q\,\mathbf{I}_4$, $\mathbf{R}_t = r\,\mathbf{I}_2$, $(\boldsymbol{\theta}_{0,t}, \boldsymbol{\theta}_{1,t})$ is the position, $(\boldsymbol{\theta}_{2,t}, \boldsymbol{\theta}_{3,t})$ is the velocity,

$$\mathbf{F}_t = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

$\Delta = 0.1$ is the sampling rate, $q = 0.10$ is the system noise, $r = 10$ is the measurement noise, and $\mathbf{I}_K$ is a $K \times K$ identity matrix. We simulate 500 trials, each with 1,000 steps. For each method, we compute the scaled RMSE metric $J_{T,i} = \sqrt{\sum_{t=1}^{T} (\boldsymbol{\theta}_{t,i} - \boldsymbol{\mu}_{t,i})^2}$ for $i \in \{0, 1, 2, 3\}$ as well as the total running time (relative to the `KF`).

In our experiments, the true data generating process is one of two variants of (4.42). The first variant (which we call **Student observations**) corresponds to a system whose measurement process comes from the Student-t likelihood:

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) &= \mathrm{St}(\boldsymbol{y}_t \,|\, \mathbf{H}_t\boldsymbol{\theta}_t,\, \mathbf{R}_t, \nu_t) \\ &= \int_0^\infty \mathcal{N}\left(\boldsymbol{y}_t \,|\, \mathbf{H}_t\boldsymbol{\theta}_t, \frac{\mathbf{R}_t}{\tau}\right) \mathrm{Gam}\left(\tau|\frac{\nu_t}{2}, \frac{\nu_t}{2}\right) d\tau, \end{aligned} \tag{4.43}$$

with $\mathrm{Gam}(\cdot|a, b)$ the Gamma density function with shape a and rate b, and $\nu_t = 2.01$. The second variant (which we call **mixture observations**) corresponds to a system where the mean of the observations changes sporadically. Instances of this variant can occur as a form of human error or a software bug in a data-entry program. To emulate this scenario, we modify (4.42) by using the following mixture model for the observation process:

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) &= \mathcal{N}\left(\boldsymbol{y}_t \,|\, \boldsymbol{m}_t,\, \mathbf{R}_t\right), \\ \boldsymbol{m}_t &= \begin{cases} \mathbf{H}_t\,\boldsymbol{\theta}_t & \text{w.p. } 1 - p_\epsilon, \\ 2\,\mathbf{H}_t\,\boldsymbol{\theta}_t & \text{w.p. } p_\epsilon, \end{cases} \end{aligned} \tag{4.44}$$

where $p_\epsilon = 0.05$.

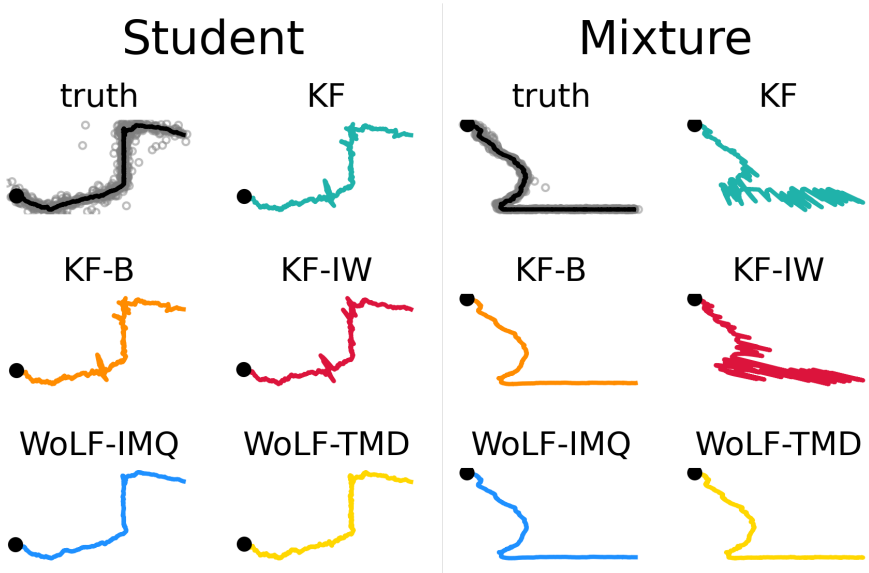

Figure 4.3: The left panel shows a sample path using the Student variant and the right panel shows a sample path using the mixture variant. The top left figure on each panel shows the true underlying state in black, and the measurements as grey dots.

Results Figure 4.3 shows a sample of each variant along with the filtered state for each method. For the Student variant (left panel), the `WoLF-IMQ` and the `WoLF-TMD` estimate the true state more closely than the competing methods. Both the `KF-IW` and the `KF-B` look comparable to the `KF`, which are not robust to outliers. For the mixture variant (right panel). the `WoLF-IMQ`, the `WoLF-TMD`, and the `KF-B` filter the true state correctly. In contrast, the `KF-IW` and the `KF` are not robust to outliers.

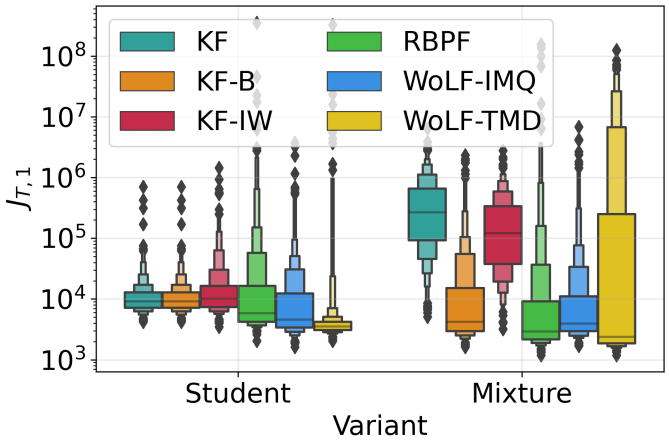

Figure 4.4: Distribution (across 500 2d tracking trials) of RMSE for first component of the state vector, $J_{T,0}$. Left panel: Student observation model. Right panel: Mixture observation model.

The results in Figure 4.3 hold for multiple trials as shown in Figure 4.4, which plots the distribution of the errors in the first component of the state vector. As a benchmark, we include the particle-filter-based method of Boustati et al. (2020), which we denote `RBPF`. The `RBPF` performs comparably to our proposed method; however, it has much higher computational cost and does not have a closed-form solution.

Method	Student	Mixture
`KF-B`	2.0x	3.7x
`KF-IW`	1.2x	5.3x
`WoLF-IMQ` (ours)	1.0x	1.0x
`WoLF-TMD` (ours)	1.0x	1.0x

Table 4.2: Mean slowdown rate over `KF`.

Table 4.2 shows the median slowdown (in running time) to process the measurements relative to the `KF`. The slowdown for method `X` is obtained dividing the running time of method `X` over the running time of the `KF`. Under the Student variant, the `WoLF-IMQ`, the `WoLF-TMD`, and the `KF-IW` have similar running time to the `KF`. In contrast, the `KF-B` takes twice the amount of time. Under the mixture variant, the `KF-B` and the `KF-IW` are almost four times and five times slower than the `KF` respectively. The changes in slowdown rate

are due the number of inner iterations that were chosen during the first trial.

4.7.2 Online learning of a neural network in the presence of outliers

In this section, we benchmark the methods using a corrupted version of the tabular UCI regression datasets.[1] Here, we consider a single-hidden-layer multi-layered perceptron (MLP) with twenty hidden units and a real-valued output unit. In this experiment, the state dimension (number of parameters in the MLP) is $D = (n_{\text{in}} \times 20 + 20) + (20 \times 1 + 1)$, where n_{in} is the dimension of the feature $\boldsymbol{x}_t$. In Table 4.3, we show the the values that n_{in} takes for each dataset.

Dataset	#Examples T	#Features o	#Parameters D
Boston	506	14	321
Concrete	1,030	9	221
Energy	768	9	221
Kin8nm	8,192	9	221
Naval	11,934	18	401
Power	9,568	5	141
Protein	45,730	10	241
Wine	1,599	12	281
Yacht	308	7	181

Table 4.3: Description of UCI datasets. Number of parameters refers to the size of the one-layer MLP.

Below, we take the static case $\mathbf{Q}_t = 0\,\mathbf{I}_D$, so that the prior predictive mean is $\boldsymbol{\mu}_{t|t-1} = \boldsymbol{\mu}_{t-1}$.

Each trial is carried out as follows: first, we randomly shuffle the rows in the dataset; second, we divide the dataset into a warmup dataset (10% of rows) and a corrupted dataset (remaining 90% of rows); third, we normalise the corrupted dataset using min-max normalisation from the warmup dataset; fourth, with probability $p_\epsilon = 0.1$, we replace a measurement $\boldsymbol{y}_t \in \mathbb{R}$ with a corrupted data point $\boldsymbol{u}_t \sim \mathcal{U}[-50, 50]$; and fifth, we run each method on the corrupted dataset.

For each dataset and for each method, we evaluate the prior predictive RMedSE

$$\text{RMedSE} = \sqrt{\text{median}\{(\boldsymbol{y}_t - h_t(\boldsymbol{\mu}_{t|t-1}))^2\}_{t=1}^{T}} \tag{4.45}$$

which is the squared root of the median squared error between the measurement $\boldsymbol{y}_t$ and the prior predictive $h_t(\boldsymbol{\mu}_{t|t-1}) = h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t)$.[2] Here, h is the MLP. We also evaluate the average time step of each method, i.e., we run each method and divide the total running time by the number of samples in the corrupted dataset.

Figure 4.5 shows the percentage change of the RMedSE and the percentage change of running time with respect to those of the `OGD` for all corrupted UCI datsets. Given the computational complexity of the remaining methods, ideally, a robust Bayesian alternative to the `OGD` should be as much to the left as possible on the x-axis (rel. time step) and as low as possible on the y-axis (rel. RMedSE). We observe that the `WoLF-IMQ` and the

[1] The dataset is available at https://github.com/yaringal/DropoutUncertaintyExps.

[2] We use median instead of mean because we have outliers in measurement space.

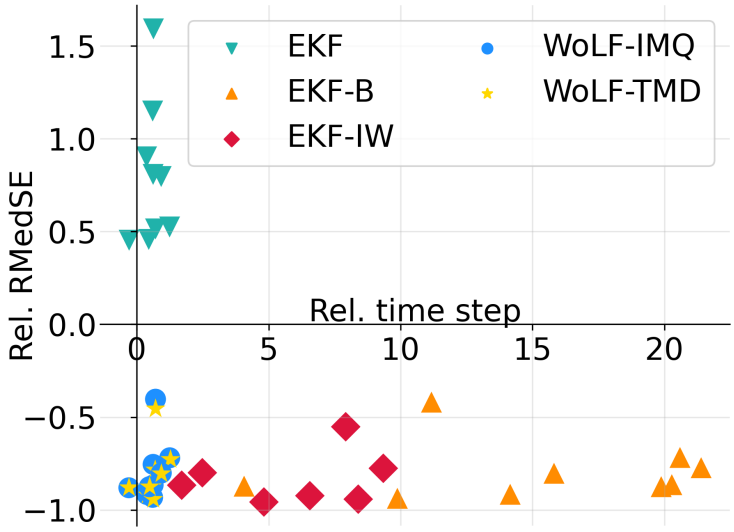

Figure 4.5: RMedSE versus time per step (relative to the `OGD` minus 1) across the corrupted UCI datasets.

`WoLF-TMD` have both of these traits. In particular, we observe that the only two points in the third quadrant are those of the `WoLF-IMQ` and the `WoLF-TMD`. Note that the `EKF-IW` and the `EKF-B` have much higher relative running time and the `EKF` has much higher relative RMedSE.

4.7.3 Robust EKF for online MLP regression (1d)

In this section, we consider an online nonlinear 1d regression, with the training data coming either from an i.i.d. source, or a correlated source. The latter corresponds to a non-stationary problem.

We present a stream of observations $\mathcal{D}^{\text{filter}} = (y_1, x_1), \ldots (y_T, x_T)$ with $y_t \in \mathbb{R}$ the measurements, $x_t \in \mathbb{R}$ the exogenous variables, and $T = 1500$. The measurements and exogenous variables are sequentially sampled from the processes

$$y_t = \begin{cases} \boldsymbol{\theta}_1^* x_t - \boldsymbol{\theta}_2^* \cos(\boldsymbol{\theta}_3^* x_t \, \pi) + \boldsymbol{\theta}_4^* x_t^3 + V_t & \text{w.p. } 1 - p_\epsilon, \\ U_t & \text{w.p. } p_\epsilon, \end{cases} \tag{4.46}$$

where the parameters of the observation model are $\boldsymbol{\theta}^* = (0.2, -10, 1.0, 1.0)$, the inputs are $x_t \sim \mathcal{U}[-3, 3]$, and the noise is $V_t \sim \mathcal{N}(0, 3)$, $U_t \sim \mathcal{U}[-40, 40]$, and $p_\epsilon = 0.05$.

We consider four configurations of this experiment. In each experiment the data is either sorted by x_t value (i.e, the exogenous variable satisfies $x_i < x_j$ for all $i < j$, representing a correlated source) or is unsorted (representing an i.i.d. source), and the measurement function is either a clean version of the true data generating process (i.e., (4.46) with $p_\epsilon = 0$ and unknown coefficients $\boldsymbol{\theta}$), or a neural network with unknown

parameters $\boldsymbol{\theta}$. Specifically, we use a multi-layered perceptron (MLP) with two hidden layers and 10 units per layer:

$$h(\boldsymbol{\theta}_t, x_t) = \boldsymbol{w}_t^{(3)} \phi\left(\boldsymbol{w}_t^{(2)} \phi\left(\boldsymbol{w}_t^{(1)} x_t + \boldsymbol{b}_t^{(1)}\right) + \boldsymbol{b}_t^{(2)}\right) + \boldsymbol{b}_t^{(3)}, \tag{4.47}$$

with activation function $\phi(u) = \max\{0, u\}$ applied elementwise. Thus the state vector encodes the parameters:

$$\boldsymbol{\theta}_t = (\boldsymbol{w}_t^{(1)} \in \mathbb{R}^{10\times 1}, \boldsymbol{w}_t^{(2)} \in \mathbb{R}^{10\times 10}, \boldsymbol{w}_t^{(3)} \in \mathbb{R}^{1\times 10}, \boldsymbol{b}_t^{(1)} \in \mathbb{R}^{10}, \boldsymbol{b}_t^{(2)} \in \mathbb{R}^{10}, \boldsymbol{b}_t^{(3)} \in \mathbb{R})$$

and has size so that $\boldsymbol{\theta} \in \mathbb{R}^{141}$. Note that in this experiment $h_t(\boldsymbol{\theta}) = h(\boldsymbol{\theta}, x_t)$. We set $Q_t = 10^{-4}\mathbf{I}$, which allows the parameters to slowly drift over time and provides some regularisation.

For each method, we evaluate the RMedSE. The `EKF-IW` and the `EKF-B` methods are taken with two inner iterations, which implies that their computational complexity is twice that of the WoLF methods.

MLP measurement model Figure 4.6 shows results when the data are presented in sorted order of x_t. We show the performance on 100 trials. The left panel shows the mean prior-predictive $h(\boldsymbol{\mu}_{t|t-1}, x_t)$ of each method, and the underlying true state process, for a single trial. The right panel shows the RMedSE after multiple trials. We observe on

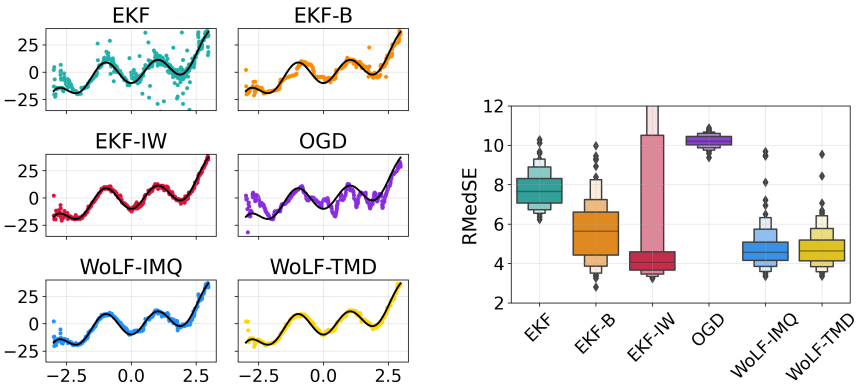

Figure 4.6: Results with sorted data. Left panel shows a run of each filter on the 1d regression, with the true underlying data-generating function in solid black line and the next-step predicted observation as dots. Right panel shows the RMedSE distribution over multiple trials.

the right panel that the `WoLF-IMQ` and the `EKF-IW` have the lowest mean error and lowest standard deviation among the competing methods. However, the `EKF-IW` takes twice as long to run the experiment. For all methods, the performance worse on the left-most side of the plot on the left panel, which is a region with not enough data to determine whether a measurement is an inlier or an outlier.

Figure 4.7 shows the results when data are presented in random order of $\boldsymbol{x}_t$. We show

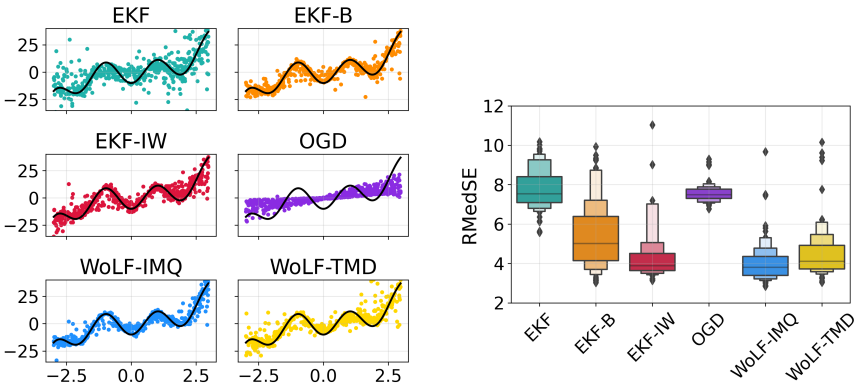

Figure 4.7: Results with unsorted inputs. The left panel shows a run of each filter with the underlying data-generating function in solid black line and the next-step predicted observation as dots. The right panel shows the distribution of G_T for multiple runs. We remove all values of G_T that have a value larger than 800.

results for a single run on the left panel and the the RMedSE after multiple trials on the right panel. Similar to the sorted configuration, we observe that the `EKF-IW` and the `WoLF-IMQ` are the methods with lowest RMedSE. However, the `EKF-IW` has longer tails than the `WoLF-IMQ`.

True measurement model We modify the experiment above by taking the measurement function to be $h_t(\boldsymbol{\theta}_t) = h(\boldsymbol{\theta}_t, x_t) = \boldsymbol{\theta}_{t,1} x_t - \boldsymbol{\theta}_{t,2} \cos(\boldsymbol{\theta}_{t,3} x_t \, \pi) + \boldsymbol{\theta}_{t,4} x_t^3$, with state $\boldsymbol{\theta}_t \in \mathbb{R}^4$ and $\boldsymbol{\theta}_{t,i}$ the i-th entry of the state vector $\boldsymbol{\theta}_t$. Figure 4.8 shows a single run of the filtering process when the data is presented unsorted (left panel) and sorted (right panel). We observe that the behaviour of the `WoLF-IMQ`, the `WoLF-TMD`, and the `EKF-IW`

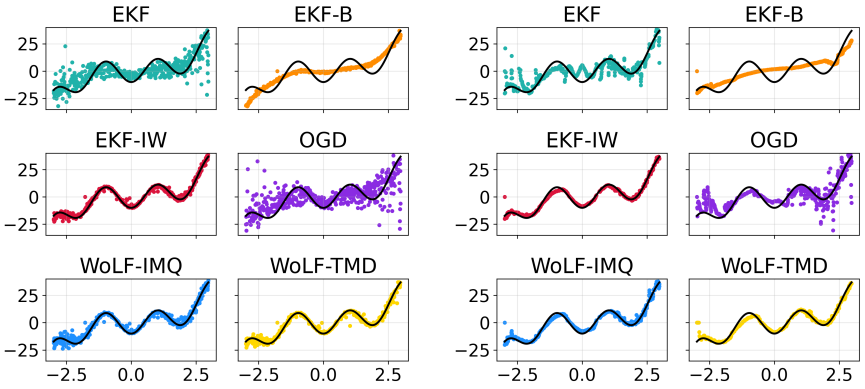

Figure 4.8: The figure shows a run of each filter with the underlying data-generating function in solid black line and the evaluation of $h(\boldsymbol{\mu}_{t|t-1}, x_t)$ in points. The left panel shows the configuration with unsorted x_t values and the right panel shows the configuration with sorted x_t values.

have similar performance. However, the `EKF-IW` takes twice the amount of time to run. The `OGD` and the `EKF` are not able to correctly filter out outlier measurement at the

tails. Finally, the EKF-B over-penalises inliers and does not capture the curvature of the measurement process.

4.7.4 Non-stationary heavy-tailed regression

In this experiment, we make use of the BONE framework developed in Chapter 3 and the WoLF filter discussed in this chapter to develop an outlier-robust method for linear regression with heavy-tailed noise and changing model parameters.

It is well-known that the combination RL-PR is sensitive to outliers if the choice of (M.1: likelihood) is misspecified, since an observation that is "unusual" may trigger a changepoint unnecessarily. As a consequence, various works have proposed outlier-robust variants to the RL[inf]-PR for segmentation (Knoblauch et al., 2018; Fearnhead and Rigaill, 2019; Altamirano et al., 2023c; Sellier and Dellaportas, 2023) and for filtering (Reimann, 2024). In what follows, we show how we can easily accommodate robust methods into the BONE framework by changing the way we compute the likelihood and/or posterior. In particular, we consider the WoLF-IMQ method of Duran-Martin et al. (2024).[3] We use WoLF-IMQ because it is a provably robust algorithm and it is a straightforward modification of the linear Gaussian posterior update equations. We denote RL[inf]-PR with (A.1: posterior) taken to be LG as LG+RL[inf]-PR and RL[inf]-PR with (A.1: posterior) taken to be WoLF-IMQ as WoLF+RL[inf]-PR*.

To demonstrate the utility of a robust method, we consider a piecewise linear regression model with Student-t errors, where the measurement are sampled according to $\boldsymbol{x}_t \sim \mathcal{U}[-2, 2]$, $\boldsymbol{y}_t \sim \mathrm{St}\big(\phi(\boldsymbol{x}_t)^\intercal \boldsymbol{\theta}_t, 1,\ 2.01\big)$ a Student-t distribution with location $\phi(\boldsymbol{x}_t)^\intercal \boldsymbol{\theta}_t$, scale 1, degrees of freedom 2.01, and $\phi(\boldsymbol{x}_t) = (1,\, x,\, x^2)$. At every timestep, the parameters take the value

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_{t-1} & \text{w.p. } 1 - p_\epsilon, \\ \mathcal{U}[-3, 3]^3 & \text{w.p. } p_\epsilon, \end{cases} \tag{4.48}$$

with $p_\epsilon = 0.01$, and $\boldsymbol{\theta}_0 \sim \mathcal{U}[-3, 3]^3$. Intuitively, at each timestep, there is probability p_ϵ of a changepoint, and conditional on a changepoint occurring, the each of the entries of the new parameters $\boldsymbol{\theta}_t$ are sampled from a uniform in $[-3, 3]$. Figure 4.9 shows a sample data generated by this process.

To process this data, our choice of (M.1: likelihood) is $h(\boldsymbol{\theta}_t, \boldsymbol{x}_t) = \boldsymbol{\theta}_t^\intercal\, \phi(\boldsymbol{x}_t)$ with

$$\ell(\boldsymbol{y}_t; \boldsymbol{\theta}_t, \boldsymbol{x}_t) = -W^2\big(\boldsymbol{y}_t, h(\boldsymbol{\theta}_t, \boldsymbol{x}_t)\big)\, \log \mathcal{N}(\boldsymbol{y}_t \,|\, h(\boldsymbol{\theta}_t, \boldsymbol{x}_t), 1.0), \tag{4.49}$$

a weighted Gaussian log-likelihood and $W(u, z) = (1 + \frac{(u-z)^2}{c^2})^{-1/2}$ the inverse multi-quadratic (IMQ) function with soft threshold value $c = 4$, representing four standard deviations of tolerance to outliers. Here $u, z \in \mathbb{R}$.

[3] We set the soft threshold value to 4, representing four standard deviations of tolerance before declaring an outlier.

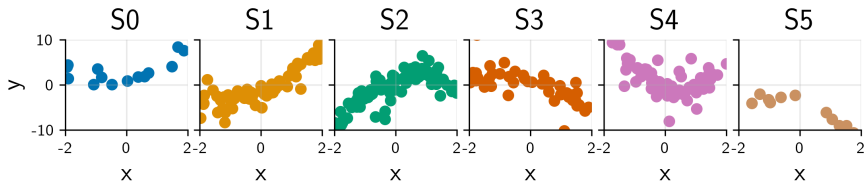

Figure 4.9: Sample run of the heavy-tailed-regression process. Each box corresponds to the samples within a segment.

The left panel in Figure 4.10 shows the rolling mean (with a window of size 10) of the RMSE for `LG+RL[inf]-PR`, `WoLF+RL[inf]-PR*`, and `LG+C-Static`. The right panel in Figure 4.10 shows the distribution of the RMSE for all methods after 30 trials.

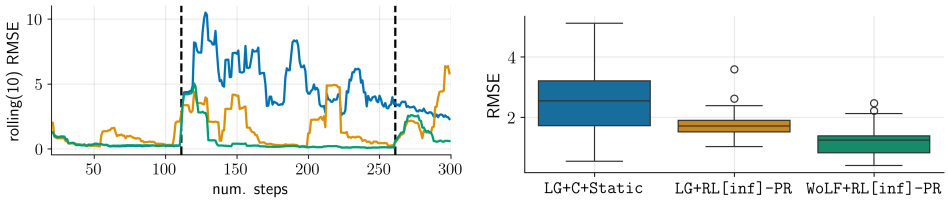

Figure 4.10: The **left panel** shows the rolling RMSE using a window of the 10 previous observations. The **right panel** shows the distribution of final RMSE over 30 runs. The vertical dotted line denotes a change in the true model parameters.

The left panel of Figure 4.10 shows that `LG+C-Static` has a lower rolling RMSE error than `LG+RL[inf]-PR` up to first changepoint (around 100 steps). The performance of `LG+C-Static` significantly deteriorates afterwards. Next, `LG+RL[inf]-PR` wrongly detects changepoints and resets its parameters frequently. This results in periods of increased rolling RMSE. Finally, `WoLF+RL[inf]-PR*` has the lowest error among the methods. After the regime change, its error increases at a similar rate to the other methods, however, it correctly adapts to the regime and its error decreases soon after the changepoint.

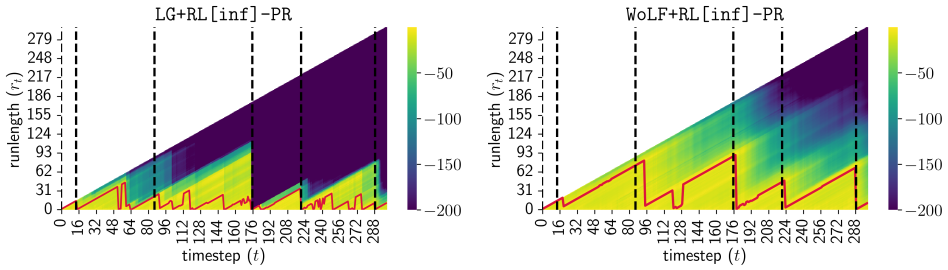

Figure 4.11: Segmentation of the non-stationary linear regression problem. The left panel shows the segmentation done by `LG+RL[inf]-PR`. The right panel shows the segmentation done by `WoLF+RL[inf]-PR*`. The x-axis is the timestep t, the y-axis is the runlength r_t (note that it is always the case that $r_t \leq t$), and the color bar shows the value $\log p(r_t \mid \boldsymbol{y}_{1:t})$. The red line in either plot is the trajectory of the mode, i.e., the set $r^*_{1:t} = \{\arg\max_{r_1} p(r_1 \mid \mathcal{D}_1), \ldots, \arg\max_{r_t} p(r_t \mid \mathcal{D}_{1:t})\}$. Note that the non-robust method (left) oversegments the signal.

Figure 4.11 shows the posterior belief of the value of the runlength using `LG+RL[inf]-PR` and `WoLF+RL[inf]-PR*`. The constant reaction to outliers in the case of `LG+RL[inf]-PR` means that the parameters keep reseting back to the initial prior belief. As a consequence, the RMSE of `LG+RL[inf]-PR` deteriorates. On the other hand, `WoLF+RL[inf]-PR*` resets less often, and accurately adjusts to the regime changes when they do happen. This results in the lowest RMSE among the three methods.

4.7.5 Outlier-robust exponentially-weighted moving average

In this experiment, we present an outlier-robust exponentially weighted moving average (EWMA) as an instance of our WoLF method.[4]

This experiment is organised as follows: first, we recap the EWMA. Then, we introduce the unidimensional SSM with unit (unobserved) signal and observation coefficients and show that the EWMA is a special case of the KF in this setting. Next, we derive the WoLF method for an EWMA. Finally, we show a numerical experiment that illustrates the robustness of the WoLF method in corrupted one-dimensional financial data.

The exponentially weighted moving average (EWMA) Given a sequence of observations (or measurements) $y_{1:t} = (y_1, \ldots, y_t)$, the EWMA of the observations at time t is given by

$$m_t = \beta\, y_t + (1 - \beta)\, m_{t-1}, \tag{4.50}$$

where $\beta \in (0, 1]$ is the smoothing factor (or learning rate). Higher levels of β give more weight to recent observations.

The Kalman filter in one dimension Consider the following one-dimensional SSM:

$$\begin{aligned} z_t &= z_{t-1} + w_t, \\ y_t &= z_t + e_t, \end{aligned} \tag{4.51}$$

where z_t is the (unobserved) signal, w_t is the process noise, and e_t is the observation noise. We assume that $\mathrm{var}(w_t) = q_t$ and $\mathrm{var}(e_t) = r_t$. Put simply, the SSM model (2) assumes that the observations y_t are generated by a (unobserved) signal z_t plus noise e_t. The (unobserved) signal z_t evolves over time according to a random walk with noise w_t and the observations y_t are generated by the (unobserved) signal z_t plus noise e_t.

Proposition 4.19. *Under the initial density $p(z_0) = \mathcal{N}(z_0 \,|\, m_0, s_0)$ and measurement model $p(y_t \,|\, z_t) = \mathcal{N}(y_t \,|\, z_t, r_t^2)$. The posterior density $p(z_t \,|\, y_{1:t})$ is given by*

$$\begin{aligned} p(z_t \,|\, y_{1:t}) &\propto p(y_t \,|\, z_t)\, p(z_t \,|\, y_{1:t-1}) \\ &= \mathcal{N}(z_t \,|\, m_t, s_t^2), \end{aligned} \tag{4.52}$$

[4] This experiment is based on the notes https://gerdm.github.io/posts/wolf-ewma/.

with

$$
\begin{aligned}
k_t &= \frac{s_{t-1}^2 + q_t^2}{s_{t-1}^2 + q_t^2 + r_t^2}, \\
s_t^2 &= k_t\, r_t^2, \\
m_t &= k_t\, y_t + (1 - k_t)\, m_{t-1}.
\end{aligned}
\tag{4.53}
$$

Proof. Suppose $p(z_0) = \mathcal{N}(z_0 \,|\, m_0, s_0)$. Let $p(z_{t-1} \,|\, y_{1:t-1}) = \mathcal{N}(z_{t-1} \,|\, m_{t-1}, s_{t-1})$ and $p(y_t \,|\, z_t) = \mathcal{N}(y_t \,|\, z_t, r_t)$. Then, the prediction step is given by

$$
\begin{aligned}
p(z_t \,|\, y_{1:t-1}) &= \int p(z_t \,|\, z_{t-1})\, p(z_{t-1} \,|\, y_{1:t-1})\, dz_{t-1} \\
&= \int \mathcal{N}(z_t \,|\, z_{t-1}, q_t)\, \mathcal{N}(z_{t-1} \,|\, m_{t-1}, s_{t-1})\, dz_{t-1} \\
&= \mathcal{N}(z_t \,|\, m_{t-1}, s_{t-1} + q_t) \\
&= \mathcal{N}(z_t \,|\, m_{t-1}, s_{t|t-1}),
\end{aligned}
$$

with $s_{t|t-1} = s_{t-1} + q_t$. Next, the update step is given by

$$
\begin{aligned}
p(z_t \,|\, y_{1:t}) &\propto p(y_t \,|\, z_t)\, p(z_t \,|\, y_{1:t-1}) \\
&= \mathcal{N}(y_t \,|\, z_t, r_t^2)\, \mathcal{N}(z_t \,|\, m_{t-1}, s_{t-1} + q_t^2).
\end{aligned}
$$

To compute the posterior, consider the log-posterior density

$$
\begin{aligned}
\log p(z_t \,|\, y_{1:t}) &= -\frac{1}{s_{t|t-1}^2}(z_t - m_{t-1})^2 - \frac{1}{r_t^2}(y_t - z_t)^2 + \text{const.} \\
&= -\frac{1}{s_{t|t-1}}(z_t^2 - 2 z_t m_{t-1} + m_{t-1}^2) - \frac{1}{r_t^2}(y_t^2 - 2 y_t z_t + z_t^2) + \text{const.} \\
&= -\frac{1}{s_{t|t-1}}(z_t^2 - 2 z_t m_{t-1}) - \frac{1}{r_t^2}(z_t^2 - 2 z_t y_t) + \text{const.} \\
&= -\left((s_{t|t-1}^{-2} + r_t^{-2}) z_t^2 - 2 z_t \left(\frac{m_{t-1}}{s_{t|t-1}^2} + \frac{y_t}{r_t^2} \right) \right) + \text{const.} \\
&= -\left(s_{t|t-1}^{-2} + r_t^{-2} \right) \left[z_t^2 - 2 z_t \left(s_{t|t-1} + r_t^{-2} \right)^{-1} \left(\frac{m_{t-1}}{s_{t|t-1}^2} + \frac{y_t}{r_t^2} \right) \right] + \text{const} \\
&= -\left(s_{t|t-1}^{-2} + r_t^{-2} \right) \left[z_t - \left(s_{t|t-1} + r_t^{-2} \right)^{-1} \left(\frac{m_{t-1}}{s_{t|t-1}^2} + \frac{y_t}{r_t^2} \right) \right]^2 + \text{const}
\end{aligned}
$$

where const denotes a constant that does not depend on z_t. From the above, we see that the posterior $p(z_t \,|\, y_{1:t})$ is a Gaussian density with mean m_t and variance s_t^2, where

$$
\begin{aligned}
m_t &= \left(s_{t|t-1} + r_t^{-2} \right)^{-1} \left(\frac{m_{t-1}}{s_{t|t-1}^2} + \frac{y_t}{r_t^2} \right), \\
s_t^2 &= \left(s_{t|t-1}^{-2} + r_t^{-2} \right)^{-1}.
\end{aligned}
$$

Next, we simplify the above expressions to obtain the Kalman filter equations (4.53). For

the posterior mean m_t, we have

$$\begin{aligned} m_t &= \left(s_{t|t-1}^2 + r_t^{-2}\right)^{-1} \left(\frac{m_{t-1}}{s_{t|t-1}^2} + \frac{y_t}{r_t^2}\right), \\ &= \frac{s_{t|t-1}^2 r_t^2}{s_{t|t-1} + r_t^2} \left(\frac{r_t^2 m_{t-1} + s_{t|t-1}^2 y_t}{s_{t|t-1}^2 r_t^2}\right), \\ &= \frac{r_t^2}{s_{t|t-1}^2 + r_t^2} m_{t-1} + \frac{s_{t|t-1}^2}{s_{t|t-1}^2 + r_t^2} y_t, \\ &= \left(1 - \frac{s_{t|t-1}^2}{s_{t|t-1}^2 + r_t^2}\right) m_{t-1} + \frac{s_{t|t-1}^2}{s_{t|t-1}^2 + r_t^2} y_t, \\ &= (1 - k_t) m_{t-1} + k_t y_t. \end{aligned}$$

with $k_t = s_{t|t-1}^2/(s_{t|t-1}^2 + r_t^2)$. Finally, compute the posterior variance s_t^2:

$$s_t^2 = \frac{1}{s_{t|t-1}^{-2} + r_t^{-2}} = \frac{s_{t|t-1}^2 r_t^2}{s_{t|t-1}^2 + r_t^2} = \left(\frac{s_{t|t-1}^2}{s_{t|t-1}^2 + r_t^2}\right) r_t^2 = k_t r_t^2.$$

□

Proposition 4.19 shows that the Kalman filter applied to the SSM (4.51) is equivalent to the EWMA with β replaced by k_t, i.e., the KF is an EWMA with a time-varying smoothing factor.

The WoLF method for the EWMA To create a 1D version of WoLF, recall that WoLF replaces the r_t in the KF equations (4.53) for $r_t^2 = r^2/w_t^2$ with $w_t : \mathbb{R} \to \mathbb{R}$ a weight function. Intuitively, the weight function w_t determines degree of certainty that y_t is an outlier. Here, we consider the IMQ weight function

$$w_t = \left(1 + \frac{(y_t - m_{t-1})^2}{c^2}\right)^{-1/2}, \tag{4.54}$$

where $c > 0$ is the soft threshold.

Next, consider the SSM (4.51) with $q_t^2 = q^2$ and $r_t^2 = r^2/w_t^2$. Here $q \geq 0$ and $r > 0$ are fixed hyperparameters. With these assumptions, the rate k_t in (4.53) for WoLF becomes

$$k_t = \frac{s_{t-1}^2 + q^2}{s_{t-1}^2 + q^2 + r^2/w_t^2}. \tag{4.55}$$

As a consequence, we obtain that, as $y_t \to \infty$, the rate k_t converges to 0 faster than y_t tends to ∞. We obtain

$$(m_t \to m_{t-1} \text{ and } s_t^2 \to s_{t-1}^2) \text{ as } y_t \to \infty. \tag{4.56}$$

In other words, with 1D Wolf, large and unexpected errors get discarded. The larger the

error, the less information it provides to the estimate m_t.

The WoLF EWMA is computed using

$$\begin{aligned} k_t &= \frac{s_{t-1}^2 + q^2}{s_{t-1}^2 + q^2 + r^2/w_t^2}, \\ s_t^2 &= k_t\, r_t^2, \\ m_t &= k_t\, y_t + (1 - k_t)\, m_{t-1}. \end{aligned} \tag{4.57}$$

A robust EWMA for log-returns To test the 1d-WoLF, consider data from the Dow Jones Industrial Average (DJI) from 2019 to the end of 2024. We want to estimate the EWMA of log-returns for DJI. Suppose that the data is corrupted with outliers and we do know in advance the level of corruption or their occurrence. Figure 4.12 shows the log-returns DJI from 2020 to 2024.

Figure 4.12: Log-returns of DJI from 2019 to 2024. The outliers at the beginning of 2021 and at the beginning of 2023 correspond to erroneous datapoints.

Next, Figure 4.13 shows the EWMA estimate and the WoLF-EWMA estimate with hyperparameters $\beta = 0.095$ for the EWMA and $m_0 = 0$, $s_0 = 1$, $q = 0.01$, $r = 1.0$, and $c = 0.05$ for WoLF-EWMA. We observe that under the standard EWMA, an outlier event significantly biases the posterior estimate of the signal z_t. However, WoLF-EWMA ignores the outliers and provides a more robust estimate of the DJI signal. Furthermore, the WoLF-EWMA estimate closely resembles that of EWMA outside *outlier events*. Finally, Figure 4.14 shows the smoothing factors β_t for the EWMA and the k_t. We observe that the smoothing factor for WoIF-EWMA k_t is fairly stable and closely follows the fixed value β for EWMA. However, it decreases when the observation y_t is is unusually large. This occurs at the outlier events.

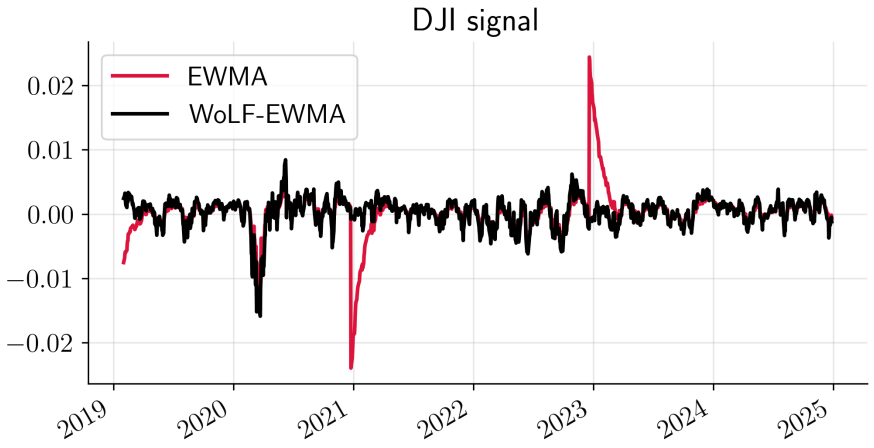

Figure 4.13: EWMA and WoIF-EWMA estimates over DJI log-returns from 2019 to 2024.

Figure 4.14: Smoothing factors for EWMA and WoIF-EWMA for the DJI log-returns from 2019 to 2024.

4.8 Conclusion

We introduced a provably robust filtering algorithm based on generalised Bayes which we call the weighted observation likelihood filter or WoLF. Our algorithm is as fast as the KF, has closed-form update equations, and is straightforward to apply to various filtering methods. The superior performance of the WoLF is shown on a wide range of filtering problems. In contrast, alternative robust methods either have higher computational complexity than the WoLF, or similar computational complexity but not higher performance.

Chapter 5

Scalability

Throughout this thesis, we have assumed that the posterior density over model parameters is Gaussian with a full-rank covariance matrix. This assumption, while simple and powerful, results in significant computational and memory demands. Specifically, most of the methods introduced in Chapter 2 scale at rates of at least $O(D^2)$ in memory and $O(D^3)$ in computation time. This limitation makes such methods impractical for online learning in high-dimensional parameter spaces, as is common with deep neural networks — the $O(D^2)$ memory requirement alone makes them infeasible for training even moderately sized neural networks.

For example, consider a multi-layer perceptron (MLP) with 28^2 input units, three hidden layers, and 100 units in each hidden layer. Such a network contains approximately 99,000 parameters. Representing these parameters as 32-bit floats would require around 40GB of memory for a single step of the Extended Kalman Filter (EKF) algorithm (see Section 2.6.1). This makes it infeasible to run on standard GPU devices and impractical for real-time applications.

To address these limitations, this chapter focuses on the challenge of recursively training deep neural networks using filtering techniques. We propose three scalable algorithms that build on the methods introduced in Chapter 2. Each algorithm leverages different structural assumptions about the state space and applies approximations to reduce both the computational and memory requirements.

First, in Section 5.1, we present the method introduced in Duran-Martin et al. (2022), which projects the weights of the neural network onto a lower-dimensional affine subspace and performs EKF-like updates on the reduced space. Next, Section 5.2 introduces the method proposed in Cartea et al. (2023b), which builds upon the subspace approach and leverages ideas from the "last-layer" literature in Bayesian neural networks. Specifically, this method projects the hidden layers of a neural network onto a lower-dimensional affine space and works with a full-covariance matrix for the last layer. Then, Section 5.3 describes the low-rank Kalman filter (LoFi) method introduced in Chang et al. (2023). This method imposes a diagonal-plus-low-rank structure on the posterior precision matrix.

This results in EKF-like updates. In contrast to the previous two methods, LoFi updates all the parameters of the neural network but reduces the computational cost by maintaining a diagonal plus low-rank posterior precision matrix.

Finally, Section 5.4 provides an overview of the methods presented in this chapter and Section 5.5 presents an empirical study of the performance of the proposed algorithms on an online classification task using the Fashion MNIST dataset (Xiao et al., 2017).

5.1 Subspace parameters

There is a growing literature showing that the number of parameters (or degrees of freedom) required to fit a neural network is much smaller than the total number of parameters contained in the neural network architecture. These *active* parameters of a neural network can either be found as a subset of nodes (Frankle, 2023), or contained within a lower-dimensional linear subspace (Li et al., 2018; Larsen et al., 2022). This observation is called the lottery-ticket hypothesis.

Research on the lottery-ticket hypothesis exploits the over-parametrisation of neural networks in the sense that "a randomly-initialised dense neural network contains subnetworks (or linear subspaces) that, when trained in isolation, reach test accuracy comparable to that of the original network" (Frankle, 2023). The subnetworks or linear subspaces that satisfy the lottery ticket hypothesis are called *winning tickets*.

The work in Duran-Martin et al. (2022) exploits the lottery-ticket hypothesis by projecting the weights of the neural network to a lower-dimensional affine subspace. Then, they make use of the EKF to perform sequential updates over the lower-dimensional space. We provide details below.

5.1.1 The SSM for the subspace EKF and update step

Consider a *projection* matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$, with $d \ll D$. Write the parameters of the neural network as a linear mapping between the projection matrix $\mathbf{A}$ and a lower-dimensional vector $\boldsymbol{z} \in \mathbb{R}^d$, plus an *offset* vector $\boldsymbol{\theta}_* \in \mathbb{R}^D$:

$$\boldsymbol{\theta}(\boldsymbol{z}) = \mathbf{A}\,\boldsymbol{z} + \boldsymbol{\theta}_*. \tag{5.1}$$

The lower-dimensional vector $\boldsymbol{z}$ is assumed to be time-dependent and evolving according to the equation $\boldsymbol{z}_t = f(\boldsymbol{z}_{t-1}) + \boldsymbol{u}_t$, with $\boldsymbol{u}_t$ a zero-mean random variable with known covariance matrix.[1] Under assumption (5.1), the SSM over model parameters and obser-

[1] The term (5.1) is different from techniques such as LoRA (Hu et al., 2021) that reduce the parameters in each layer using a linear projection. Here, we project all of the parameters of the network onto a single lower-dimensional space.

vations becomes

$$\begin{aligned} \boldsymbol{z}_t &= f(\boldsymbol{z}_{t-1}) + \boldsymbol{u}_t, \\ \boldsymbol{\theta}_t &:= \boldsymbol{\theta}(\boldsymbol{z}_t) = \mathbf{A}\,\boldsymbol{z}_t + \boldsymbol{\theta}_*, \\ \boldsymbol{y}_t &= h(\boldsymbol{\theta}_t, \boldsymbol{x}_t) + \boldsymbol{e}_t. \end{aligned} \tag{5.2}$$

The SSM (5.2) corresponds to that presented in Section 2.6.1 with measurement function $h(\mathbf{A}\,\boldsymbol{z}_t + \boldsymbol{\theta}_*, \boldsymbol{x}_t)$. Assuming Gaussian prior for the subspace $\mathcal{N}(\boldsymbol{z}_0 \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, application of the EKF is straightforward. Algorithm 12 shows a step of the EKF under the subspace assumption. For simplicity, we assume that $f(\boldsymbol{z}) = \boldsymbol{z}$.

Algorithm 12 predict and update steps for the subspace extended Kalman filter with subspace for $t \geq 1$.

Require: $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ // datapoint
Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ // previous d-dimensional mean and covariance
Require: $\mathbf{A} \in \mathbb{R}^{D \times d}$ // projection matrix
Require: $\boldsymbol{\theta}_* \in \mathbb{R}^D$ // offset vector
// predict step
$\mathbf{F}_t \leftarrow \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\mu}_{t-1})$
$\boldsymbol{\mu}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\mu}_{t-1}$
$\boldsymbol{\Sigma}_{t|t-1} \leftarrow \mathbf{F}_t \boldsymbol{\Sigma}_{t|t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t$
//update step
$\mathbf{H}_t \leftarrow \nabla_{\boldsymbol{\theta}} h(\mathbf{A}\,\boldsymbol{\mu}_{t|t-1} + \boldsymbol{\theta}_*, \boldsymbol{x}_t)$
$\mathbf{S}_t = \mathbf{H}_t\,\mathbf{A}\,\boldsymbol{\Sigma}_{t-1}\,\mathbf{A}^\intercal\,\mathbf{H}_t^\intercal + \mathbf{R}_t$
$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}\,\mathbf{A}^\intercal\,\mathbf{H}_t^\intercal\,\mathbf{S}_t^{-1}$
$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \mathbf{H}_t\,\mathbf{A}\,\boldsymbol{\mu}_{t|t-1})$
$\boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t\,\mathbf{A}^\intercal\,\mathbf{H}_t^\intercal\,\boldsymbol{\Sigma}_{t|t-1}$
return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$

5.1.2 Warmup phase: estimating the projection matrix and the offset term

Two important components of the method are the projection matrix $\mathbf{A}$ and the offset vector $\boldsymbol{\theta}_*$. The offset can either be initialised from a random process, as in Li et al. (2018), or determined during a *warmup* period, as described in Larsen et al. (2022). In this section, we focus on the approach proposed in Larsen et al. (2022), where the projection matrix $\mathbf{A}$ is constructed by performing a singular value decomposition (SVD) on iterates of batch stochastic gradient descent (SGD), and the offset $\boldsymbol{\theta}_*$ is taken as the final step of the warmup procedure.

Suppose we are given a warm-up dataset $\mathcal{D}_{\text{warmup}}$ with $N_{\text{warmup}} \geq 1$ datapoints, which we divide into into B non-intersecting random batches $\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(B)}$ such that

$$\bigcup_{b=1}^{B} \mathcal{D}_{(b)} = \mathcal{D}_{\text{warmup}}.$$

Consider the negative log-likelihood

$$-\log p(\mathcal{D}_{\text{warmup}} \,|\, \boldsymbol{\theta}) = -\sum_{n=1}^{N_{\text{warmup}}} \log p(\boldsymbol{y}_n \,|\, \boldsymbol{\theta}, \boldsymbol{x}_n). \tag{5.3}$$

And train for E epochs using randomised batch SGD. At the end of the E epochs, we obtain $\boldsymbol{\theta}^{(E)}$. Then, the offset vector $\boldsymbol{\theta}_*$ is given by

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}^{(E)}.$$

Next, to estimate the projection matrix $\mathbf{A}$, we take the iterates found during the SGD optimisation procedure. To avoid redundancy, we skip the first n iterations and store the iterates every k steps. Let

$$\mathcal{E} = \begin{bmatrix} — & \boldsymbol{\theta}^{(n)} & — \\ — & \boldsymbol{\theta}^{(n+k)} & — \\ — & \boldsymbol{\theta}^{(n+2k)} & — \\ & \vdots & \\ — & \boldsymbol{\theta}^{(E)} & — \end{bmatrix} \in \mathbb{R}^{\hat{E} \times D},$$

where $\hat{E} = \lfloor (E-n)/k \rfloor + 1$. With the SVD decomposition $\mathcal{E} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}$ and the first d columns of the matrix $\mathbf{V}$, the projection matrix is

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{V}_{:,1} & \mathbf{V}_{:,d} & \cdots & \mathbf{V}_{:,d} \\ | & | & & | \end{bmatrix},$$

where $\mathbf{V}_{:,\,k}$ denotes the k-th column of $\mathbf{V}$. Algorithm 13 shows the warmup procedure to find $\mathbf{A}$ and $\boldsymbol{\theta}_*$ using a warmup dataset $\mathcal{D}_{\text{warmup}}$. In Algorithm 13, the function $\boldsymbol{\kappa} : \mathbb{R}^M \to \mathbb{R}^M$ is the per-step transformation of the Adam algorithm; see Kingma and Ba (2015).

Algorithm 13 Initialisation of the offset vector and projection matrix via batch SGD.

Require: $\boldsymbol{\theta}^{(0)}$ // initial set of parameters
Require: $E \geq 1$ // number of epochs
Require: $\mathcal{D}_{\text{warmup}}$ // warmup dataset
Require: $\mathcal{E} = []$ // empty matrix
Require: $n \geq 0$, $k \geq 1$ // skip and stride numbers
for epoch $e = 1, \ldots, E$ **do**
  for batch $m = 1, \ldots, M$ **do**
    $G_e \leftarrow -\nabla_{\boldsymbol{\theta}} \log p(\mathcal{D}_{(m)} \,|\, \boldsymbol{\theta})$
    $\boldsymbol{\theta}^{(e)} \leftarrow \boldsymbol{\theta}^{(e-1)} - \boldsymbol{\kappa}(G_T)$
  end for
  if $(e \geq n)$ **and** $(e \mod k = 0)$ **then**
    $\mathcal{E} \leftarrow \left[\mathcal{E}^\intercal, \left(\boldsymbol{\theta}^{(e)}\right)^\intercal\right]^\intercal$ // stack vertically
  end if
end for
$\mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V} \leftarrow \text{SVD}(\mathcal{E})$ // SVD decomposition of stacked SGD iterates
$\mathbf{A} \leftarrow \begin{bmatrix} \mathbf{V}_{:,1} & \ldots & \mathbf{V}_{:,d} \end{bmatrix}$ // define projection matrix
$\boldsymbol{\theta}_* \leftarrow \boldsymbol{\theta}^{(E)}$ // define offset parameters

5.2 Subspace and last-layer parameters

In the previous section, we leveraged the lottery ticket hypothesis to make online estimation of the posterior density of high-dimensional model parameters computationally tractable. However, the performance of this method heavily depends on the choice of the projection matrix $\mathbf{A}$, which projects the parameters of the hidden and output layers onto a single linear subspace. This reliance on $\mathbf{A}$ can limit the method's flexibility and effectiveness in practice.

An alternative approach involves treating the last layer and the hidden layers of a neural network as separate components. Commonly referred to as Bayesian last-layer methods (Harrison et al., 2024) or neural-linear methods, these approaches improve the predictive power of neural networks by optimising the weights of the hidden layers using standard techniques, such as Adam, and then placing a posterior density over the parameters of the last layer (Murphy, 2023, Section 17.3.5). This separation is particularly useful in applications such as Bayesian neural contextual bandit problems (Riquelme et al., 2018).

Because the hidden layers are typically much more computationally expensive to train than the output layers, these methods often update the hidden layers intermittently, while the output layer is updated more frequently. This division helps to reduce the computational cost of training while maintaining predictive accuracy.

Building on these ideas, the work in Cartea et al. (2023b) proposes a fully online Bayesian version of last-layer methods, called PULSE (projection-based unification of

last-layer and subspace estimation). PULSE combines the strengths of the lottery ticket hypothesis with the power of neural-linear approaches by applying subspace projections only to the hidden layers, while maintaining a full-covariance Gaussian posterior over the parameters of the last layer.[2] This approach allows for a fully-online treatment of online learning of high-dimensional neural networks, while maintaining a reduced computational cost.

Specifically, PULSE provides a Bayesian framework for sequentially updating the parameters of a neural network. It uses subspace projections to represent the hidden-layer parameters compactly while maintaining a detailed posterior density for the last layer. Unlike previous approaches, which either update all parameters or focus solely on the last layer during online learning, PULSE strikes a balance by combining both strategies: projecting the hidden-layer parameters and fully updating the last-layer parameters. Whereas the original PULSE method decouples the update for the hidden layer and the subspace entirely, here, we introduce a modification that solves a coupled system of equations.

5.2.1 The SSM for PULSE

We decompose the model parameters for a neural network $\boldsymbol{\theta} \in \mathbb{R}^D$ between the hidden-layer parameters $\boldsymbol{\psi} \in \mathbb{R}^{D_{\text{hidden}}}$ and the last-layer parameters $\mathbf{w} \in \mathbb{R}^{d_{\text{last}}}$. That is, $\boldsymbol{\theta} = (\boldsymbol{\psi}, \mathbf{w})$. Here, $D = D_{\text{hidden}} + d_{\text{last}}$. Next, write the hidden layer parameters $\boldsymbol{\psi}$ as an affine projection of the form

$$\boldsymbol{\psi} = \mathbf{A}\,\boldsymbol{z} + \boldsymbol{\psi}_*. \tag{5.4}$$

Similar to (5.1), $\mathbf{A}$ is a $(D_{\text{hidden}} \times d_{\text{hidden}})$ fixed projection matrix and $\boldsymbol{z}_t \in \mathbb{R}^{d_{\text{hidden}}}$ are the projected (subspace) parameters such that $d_{\text{hidden}} \ll D_{\text{hidden}}$, and $\boldsymbol{\psi}_* \in \mathbb{R}^{D_{\text{hidden}}}$ is the offset term. The term $\boldsymbol{\psi}_*$ is initialised as in Section 5.1.2 but considering only the last layer parameters.

Similar to Section 5.1, the lower-dimensional vector $\boldsymbol{z}$ is assumed to be time-dependent and evolving according to the equation $\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \boldsymbol{u}_t^{\text{hidden}}$, with $\boldsymbol{u}_t^{\text{hidden}}$ a zero-mean random vector with known covariance matrix. Next, the parameters of the last layer are assumed to be time-dependent and evolve according to $\mathbf{w}_t = \mathbf{w}_{t-1} + \boldsymbol{u}_t^{\text{last}}$ with $\boldsymbol{u}_t^{\text{last}}$ a zero-mean random vector with known covariance matrix. Here, we take $\text{Cov}(\boldsymbol{u}_s^{\text{hidden}}, \boldsymbol{u}_t^{\text{rlast}}) = \mathbf{0}$ for all s, t. Under these assumptions, the SSM over model

[2] In Cartea et al. (2023b), PULSE is employed to detect the so-called *toxicity* of trades being sent to a broker. This is different from other models (Cartea and Sánchez-Betancourt, 2022; Bergault and Sánchez-Betancourt, 2024; Cartea et al., 2024b; Aqsha et al., 2024) because we do not assume that traders are informed or uninformed, instead, we think of each trade separately and we predict whether the trade is informed or uninformed.

parameters and observations take the form

$$\begin{aligned}
\boldsymbol{z}_t &= \boldsymbol{z}_{t-1} + \boldsymbol{u}_t^{\text{hidden}}, \\
\mathbf{w}_t &= \mathbf{w}_{t-1} + \boldsymbol{u}_t^{\text{last}}, \\
\boldsymbol{\theta}_t &= (\mathbf{A}\,\boldsymbol{z}_t + \boldsymbol{\psi}_*, \mathbf{w}_t), \\
\boldsymbol{y}_t &= h(\boldsymbol{\theta}_t, \boldsymbol{x}_t) + \boldsymbol{e}_t,
\end{aligned} \tag{5.5}$$

where, as before, $\boldsymbol{e}_t$ is a zero mean random vector with known covariance matrix $\mathbf{R}_t$, $\boldsymbol{y}_t \in \mathbb{R}^o$ are the measurements and $h : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}^o$ is a differentiable function w.r.t. the first entry, e.g., a neural network.

5.2.2 Objective

We introduce Gaussian priors for both $\mathbf{w}$ and $\boldsymbol{z}$ at the beginning of the deploy stage. Let $t = 1$ be the first timestamp of the deploy dataset. Denote the prior densities for $\mathbf{w}$ and $\boldsymbol{z}$ by

$$\begin{aligned}
&\mathcal{N}\left(\mathbf{w} \,|\, \mathbf{w}^{(E)},\, \sigma_{\mathbf{w}}^2\, \mathbf{I}_{d_{\text{last}}}\right), \\
&\mathcal{N}\left(\boldsymbol{z} \,|\, \mathbf{0},\, \sigma_{\boldsymbol{z}}^2\, \mathbf{I}_{d_{\text{hidden}}}\right),
\end{aligned}$$

where $\mathbf{w}^{(E)}$ is the last iterates from the warmup stage, and $\sigma_{\mathbf{w}}^2$, $\sigma_{\boldsymbol{z}}^2$ are the coefficients of the prior covariance matrix.

At each timestep during the deploy stage, we approximate the posterior densities of the last-layer parameters $\mathbf{w}$, and the subspace hidden-layer parameters $\boldsymbol{z}$, as disjoint multivariate Gaussians.

To compute the posterior mean and covariance of the subspace hidden-layer parameters $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, as well as the posterior mean and covariance of the last-layer parameters $(\boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)$, we solve a recursive variational inference (VI) optimisation problem. This approach resembles the R-VGA method introduced in Section 2.4.2, so that $\text{Cov}(\boldsymbol{u}_t^{\text{hidden}}) = \mathbf{0}$ and $\text{Cov}(\boldsymbol{u}_t^{\text{last}}) = \mathbf{0}$. Specifically, at each timestep, we optimise the block mean-field objective

$$\boldsymbol{\Theta}_t = \underset{\boldsymbol{\mu},\boldsymbol{\nu},\boldsymbol{\Sigma},\boldsymbol{\Gamma}}{\arg\min}\, \mathbf{D}_{\text{KL}}\left(\mathcal{N}\left(\mathbf{w} \,|\, \boldsymbol{\nu},\, \boldsymbol{\Gamma}\right) \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu},\, \boldsymbol{\Sigma}\right) \,||\, p(\mathbf{w}, \boldsymbol{z} \,|\, \mathcal{D}_{1:T})\right), \tag{5.6}$$

where $\boldsymbol{\Theta}_t = (\boldsymbol{\mu}_t, \boldsymbol{\nu}_t, \boldsymbol{\Sigma}_t, \boldsymbol{\Gamma}_t)$ represents the set of optimised posterior parameters, and $p(\mathbf{w}, \boldsymbol{z} \,|\, \mathcal{D}_{1:t})$ denotes the reference posterior density at time t, given by

$$\begin{aligned}
p(\mathbf{w}, \boldsymbol{z} \,|\, \mathcal{D}_{1:t}) &\propto q_{t-1}(\mathbf{w}, \boldsymbol{z} \,|\, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) \\
&= \mathcal{N}\left(\mathbf{w} \,|\, \boldsymbol{\nu}_{t-1},\, \boldsymbol{\Gamma}_{t-1}\right) \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu}_{t-1},\, \boldsymbol{\Sigma}_{t-1}\right)\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t).
\end{aligned} \tag{5.7}$$

Here, $\boldsymbol{\theta} = (\mathbf{A}\,\boldsymbol{z} + \boldsymbol{\psi}_*, \mathbf{w})$ and $q_{t-1}(\mathbf{w}, \boldsymbol{z} \,|\, \mathcal{D}_{1:t-1})$ is the posterior density from the previous timestep, which we take to be the prior when computing the posterior at time t. In

general, the density $p(\boldsymbol{y}_t \,|\, (\boldsymbol{z}, \mathbf{w}), \boldsymbol{x}_t)$ might not be linear-Gaussian (h not linear in $\boldsymbol{\theta}$). In such cases, we employ a linearised moment-matched Gaussian likelihood $\hat{p}(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$—we return to this point in the next subsection.

Then, the approximated posterior at time t is

$$q_t(\boldsymbol{z}, \mathbf{w}) = \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right) \mathcal{N}\left(\mathbf{w} \,|\, \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t\right). \tag{5.8}$$

5.2.3 Update step

To obtain closed-form updates, we consider a modified likelihood for the measurement $\boldsymbol{y}_t$, which we take as the one introduced in the exponential-family EKF of Section 2.6.3. Let $h(\boldsymbol{\theta}, \boldsymbol{x})$ be the mean of the measurement model $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x})$. A first-order approximation of $h(\boldsymbol{\theta}, \boldsymbol{x}_t)$ around $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\nu}_{t-1})$ yields

$$\bar{h}_t(\boldsymbol{z}, \mathbf{w}) = \kappa_t + \bar{\mathbf{Z}}_t \left(\boldsymbol{z} - \boldsymbol{\mu}_{t-1}\right) + \bar{\mathbf{W}}_t \left(\mathbf{w} - \boldsymbol{\nu}_{t-1}\right) \tag{5.9}$$

with

$$\begin{aligned} \kappa_t &= h\big((\boldsymbol{\mu}_{t-1}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big), \\ \bar{\mathbf{Z}}_t &= \nabla_{\boldsymbol{z}} h\big((\boldsymbol{z}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big)|_{\boldsymbol{z}=\boldsymbol{\mu}_{t-1}}, \\ \bar{\mathbf{W}}_t &= \nabla_{\mathbf{w}} h\big((\boldsymbol{\mu}_{t-1}, \mathbf{w}), \boldsymbol{x}_t\big)|_{\mathbf{w}=\boldsymbol{\nu}_{t-1}}. \end{aligned} \tag{5.10}$$

The density for the measurements is taken to be

$$\hat{p}(\boldsymbol{y}_t \,|\, (\boldsymbol{z}, \mathbf{w}), \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \,|\, \bar{h}_t(\boldsymbol{z}, \mathbf{w}), \bar{\mathbf{R}}_t) \propto \exp\left(-\tfrac{1}{2} \| \bar{\mathbf{R}}_t^{-1/2} \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{z}, \mathbf{w})\right) \|_2^2\right), \tag{5.11}$$

where $\bar{\mathbf{R}}_t$ is the moment-matched observation variance. Next, our reference posterior is modified as

$$\begin{aligned} p(\boldsymbol{z}, \mathbf{w} \,|\, \mathcal{D}_{1:t}) &\propto q_{t-1}(\boldsymbol{z}, \mathbf{w} \,|\, \mathcal{D}_{1:t-1})\, \hat{p}(\boldsymbol{y}_t \,|\, (\boldsymbol{z}, \mathbf{w}), \boldsymbol{x}_t) \\ &= \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}\right) \mathcal{N}\left(\mathbf{w} \,|\, \boldsymbol{\nu}_{t-1}, \boldsymbol{\Gamma}_{t-1}\right) \hat{p}(\boldsymbol{y}_t \,|\, (\boldsymbol{z}, \mathbf{w}), \boldsymbol{x}_t). \end{aligned} \tag{5.12}$$

Proposition 5.1. *Optimising the objective* (5.6) *under the linearised measurement model* (5.11) *yields the fixed-point equations*

$$\begin{aligned} \boldsymbol{\nu}_t &= \boldsymbol{\nu}_{t-1} - \boldsymbol{\Gamma}_{t-1} \nabla_{\boldsymbol{\nu}_t} \mathcal{E}_t, \\ \boldsymbol{\Gamma}_t^{-1} &= \boldsymbol{\Gamma}_{t-1}^{-1} + 2\, \nabla_{\boldsymbol{\Gamma}_t} \mathcal{E}_t, \\ \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} - \boldsymbol{\Sigma}_{t-1} \nabla_{\boldsymbol{\mu}_t} \mathcal{E}_t, \\ \boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t-1}^{-1} + 2\, \nabla_{\boldsymbol{\Sigma}_t} \mathcal{E}_t, \end{aligned} \tag{5.13}$$

where $\mathcal{E}_t := \mathbb{E}_{\mathcal{N}(\boldsymbol{z} \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \,|\, \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}[\log \hat{p}(\boldsymbol{y}_t \,|\, \boldsymbol{z}, \mathbf{w}, \boldsymbol{x}_t)]$.

Proof. We begin by rewriting the objective function (5.6) using the linearised measurement model (5.11). To simplify notation, let $\hat{p}(\boldsymbol{y}_t) := p(\boldsymbol{y}_t \,|\, \boldsymbol{z}, \mathbf{w}; \boldsymbol{x}_t)$, $\varphi_t(\boldsymbol{z}) :=$

$\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, and $\phi_t(\mathbf{w}) := \mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)$.

$$
\begin{aligned}
\mathcal{K}_t &= \mathbf{D}_{\mathrm{KL}}\left(\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \,||\, \phi_{t-1}(\mathbf{w})\,\varphi_{t-1}(\boldsymbol{z})\,\hat{p}(\boldsymbol{y}_t)\right) \\
&= \iint \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t) \log\left(\frac{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}{\varphi_{t-1}(\boldsymbol{z})\,\phi_{t-1}(\mathbf{w})\,p(\boldsymbol{y}_t)}\right) \mathrm{d}\boldsymbol{z}\,\mathrm{d}\mathbf{w} \\
&= \iint \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t) \left[\log\left(\frac{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)}{\varphi_{t-1}(\boldsymbol{z})}\right) + \log\left(\frac{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)}{\phi_{t-1}(\mathbf{w})}\right) - \log\hat{p}(\boldsymbol{y}_t)\right] \mathrm{d}\boldsymbol{z}\,\mathrm{d}\mathbf{w} \\
&= \int \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\,\log\left(\frac{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)}{\varphi_{t-1}(\boldsymbol{z})}\right) \mathrm{d}\boldsymbol{z} + \int \mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t) \log\left(\frac{\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}{\phi_{t-1}(\mathbf{w})}\right) \mathrm{d}\mathbf{w} \\
&\quad + \iint \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)\,\log\hat{p}(\boldsymbol{y}_t)\,\mathrm{d}\mathbf{w}\,\mathrm{d}\boldsymbol{z} \\
&= \mathbb{E}_{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)}\left[\log\left(\frac{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)}{\varphi_{t-1}(\boldsymbol{z})}\right)\right] + \mathbb{E}_{\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}\left[\log\left(\frac{\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}{\phi_{t-1}(\mathbf{w})}\right)\right] \\
&\quad + \mathbb{E}_{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}[\log\hat{p}(\boldsymbol{y}_t)]\,.
\end{aligned}
\tag{5.14}
$$

Finally, we obtain

$$
\mathcal{K}_t = \mathbf{D}_{\mathrm{KL}}\left(\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t) \,||\, \phi_{t-1}(\mathbf{w})\right) + \mathbf{D}_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \,||\, \varphi_{t-1}(\boldsymbol{z})\right) + \mathcal{E}_t \tag{5.15}
$$

where $\mathcal{E}_t := \mathbb{E}_{\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t)}[\log\hat{p}(\boldsymbol{y}_t)]$.

The first and second terms in (5.15) correspond to a Kullback–Leibler divergence between two multivariate Gaussians. The last term corresponds to the posterior-predictive marginal log-likelihood for the t-th observation. To minimise (5.15) with respect to $\boldsymbol{\Theta}_t$, we use the Kullback-Leibler divergence between two multivariate Gaussian derived in Proposition 2.8.

The derivative of $\mathcal{K}_t$ with respect to $\boldsymbol{\nu}_t$ is

$$
\begin{aligned}
\nabla_{\boldsymbol{\nu}_t}\mathcal{K}_t &= \nabla_{\boldsymbol{\nu}_t}\left(\mathbf{D}_{\mathrm{KL}}\left(\mathcal{N}(\mathbf{w} \mid \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t) \,||\, \phi_{t-1}(\mathbf{w})\right) + \mathcal{E}_t\right) \\
&= \nabla_{\boldsymbol{\nu}_t}\left(\frac{1}{2}\boldsymbol{\nu}_t^{\intercal}\boldsymbol{\Gamma}_{t-1}^{-1}\boldsymbol{\nu}_t - \boldsymbol{\nu}_t^{\intercal}\boldsymbol{\Gamma}_{t-1}\boldsymbol{\nu}_{t-1} + \nabla_{\boldsymbol{\nu}_t}\mathcal{E}_t\right) \\
&= \boldsymbol{\Gamma}_{t-1}^{-1}\boldsymbol{\nu}_t - \boldsymbol{\Gamma}_{t-1}^{-1}\boldsymbol{\nu}_{t-1} + \nabla_{\boldsymbol{\nu}_t}\mathcal{E}_t \\
&= \boldsymbol{\Gamma}_{t-1}^{-1}\left(\boldsymbol{\nu}_t - \boldsymbol{\nu}_{t-1} - \boldsymbol{\Gamma}_{t-1}\nabla_{\boldsymbol{\nu}_t}\mathcal{E}_t\right).
\end{aligned}
\tag{5.16}
$$

Set (5.16) to zero and solve for

$$
\boldsymbol{\nu}_t = \boldsymbol{\nu}_{t-1} - \boldsymbol{\Gamma}_{t-1}\nabla_{\boldsymbol{\nu}_t}\mathcal{E}_t.
$$

Next, we estimate the condition for $\boldsymbol{\Gamma}_t$. Use (5.15) to obtain

$$
\begin{aligned}
\nabla_{\boldsymbol{\Gamma}_t}\mathcal{K}_t &= \nabla_{\boldsymbol{\Gamma}_t}\left(-\frac{1}{2}\log|\boldsymbol{\Gamma}_t| + \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Gamma}_t\,\boldsymbol{\Gamma}_{t-1}^{-1}\right) + \mathcal{E}_t\right) \\
&= -\frac{1}{2}\,{\boldsymbol{\Gamma}_t}^{-1} + \frac{1}{2}\,\boldsymbol{\Gamma}_{t-1}^{-1} + \nabla_{\boldsymbol{\Gamma}}\mathcal{E}_t.
\end{aligned}
\tag{5.17}
$$

The fixed-point solution for (5.17) satisfies

$$\boldsymbol{\Gamma}_t{}^{-1} = \boldsymbol{\Gamma}_{t-1}^{-1} + 2\,\nabla_{\boldsymbol{\Gamma}_t}\mathcal{E}_t.$$

The fixed-point conditions for $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are derived similarly. □

Proposition 5.1 yields the fixed-point equations that the terms in $\boldsymbol{\Theta}_t$ must satisfy. Furthermore, these equations are expressed in terms of the gradients of the expected linearised likelihood model (5.11). In the following proposition, we provide explicit expressions for the gradients of the linearised likelihood model with respect to each term in $\boldsymbol{\Theta}$.

Proposition 5.2. *Let* $\mathcal{E}_t := \mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t)]$ *be the expected linearised log-likelihood. The derivative of* $\mathcal{E}_t$ *w.r.t.* $\boldsymbol{\nu}_t$ *and* $\boldsymbol{\Gamma}_t$ *take the form*

$$\nabla_{\boldsymbol{\nu}_t}\,\mathcal{E}_t = -\bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t,\boldsymbol{\nu}_t)\right), \tag{5.18}$$

$$\nabla_{\boldsymbol{\Gamma}_t}\,\mathcal{E}_t = \tfrac{1}{2}\bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\bar{\mathbf{W}}_t. \tag{5.19}$$

Similarly, the derivative of $\mathcal{E}_t$ *w.r.t.* $\boldsymbol{\mu}_t$ *and* $\boldsymbol{\Sigma}_t$ *take the form*

$$\nabla_{\boldsymbol{\mu}_t}\,\mathcal{E}_t = -\bar{\mathbf{Z}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t,\boldsymbol{\nu}_t)\right), \tag{5.20}$$

$$\nabla_{\boldsymbol{\Sigma}_t}\,\mathcal{E}_t = \tfrac{1}{2}\bar{\mathbf{Z}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\bar{\mathbf{Z}}_t. \tag{5.21}$$

Proof. From Bonnet's Theorem and Price's Theorem (See Theorem 3 and Theorem 4 in Lin et al. (2019)), we obtain

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}}\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t)] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\nabla_{\mathbf{w}}\log \hat{p}(\boldsymbol{y}_t)],\\
\nabla_{\boldsymbol{\Gamma}}\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t)] &= \tfrac{1}{2}\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\nabla^2_{\mathbf{w}}\log \hat{p}(\boldsymbol{y}_t)].
\end{aligned} \tag{5.22}$$

The Jacobian and the Hessian and the log-likelihood w.r.t. $\mathbf{w}$ are

$$\begin{aligned}
\nabla_{\mathbf{w}}\,\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t) &= -\bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{z},\mathbf{w})\right),\\
\nabla^2_{\mathbf{w}}\,\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t) &= \bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\bar{\mathbf{W}}_t.
\end{aligned} \tag{5.23}$$

Hence,

$$\begin{aligned}
\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\nabla_{\mathbf{w}}\,\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t)] &= -\bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\left(\boldsymbol{y}_t - \bar{h}_t\right),\\
\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{\mu}_t,\,\boldsymbol{\Sigma}_t)\mathcal{N}(\mathbf{w}\,|\,\boldsymbol{\nu}_t,\,\boldsymbol{\Gamma}_t)}[\nabla^2_{\mathbf{w}}\,\log \hat{p}(\boldsymbol{y}_t\,|\,(\boldsymbol{z},\mathbf{w}),\boldsymbol{x}_t)] &= \bar{\mathbf{W}}_t{}^{\intercal}\,\bar{\mathbf{R}}_t^{-1}\,\bar{\mathbf{W}}_t.
\end{aligned} \tag{5.24}$$

The result is derived similarly for the parameters of hidden subspace. □

Plugging the result of Proposition 5.2 into the fixed-point equations given in Proposition 5.1 yields a pair of linear equations whose only unknowns are the terms in $\boldsymbol{\Theta}$. The

next theorem derives the update equations for PULSE, which explicitly solve for each term in $\boldsymbol{\Theta}$, yielding $\boldsymbol{\Theta}_t$.

Theorem 5.3 (PULSE)**.** *The approximated posterior that solves objective* (5.6) *under the linearised measurement model* (5.11) *at time* t *is*

$$q_t(\boldsymbol{z}, \mathbf{w}) = \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t\right) \mathcal{N}\left(\mathbf{w} \,|\, \boldsymbol{\nu}_t,\, \boldsymbol{\Gamma}_t\right), \tag{5.25}$$

where

$$\begin{aligned}
\boldsymbol{\nu}_t &= \boldsymbol{\nu}_{t-1} + \hat{\mathbf{K}}_{\mathbf{w},t}\left(\boldsymbol{y}_t - h\big((\boldsymbol{\mu}_{t-1}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big)\right), \\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} + \hat{\mathbf{K}}_{\boldsymbol{z},t}\left(\boldsymbol{y}_t - h\big((\boldsymbol{\mu}_{t-1}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big)\right), \\
\boldsymbol{\Gamma}_t^{-1} &= \boldsymbol{\Gamma}_{t-1}^{-1} + \bar{\mathbf{W}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \bar{\mathbf{W}}_t, \\
\boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t-1}^{-1} + \bar{\mathbf{Z}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \bar{\mathbf{Z}}_t,
\end{aligned} \tag{5.26}$$

and

$$\begin{aligned}
\mathbf{K}_{\boldsymbol{z},t} &= \boldsymbol{\Sigma}_t\, \bar{\mathbf{Z}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}, \\
\mathbf{K}_{\mathbf{w},t} &= \boldsymbol{\Gamma}_t\, \bar{\mathbf{W}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}, \\
\hat{\mathbf{K}}_{\boldsymbol{z},t} &= \left(\mathbf{I}_{d_{\text{hidden}}} - \mathbf{K}_{\boldsymbol{z},t}\, \bar{\mathbf{W}}_t\, \mathbf{K}_{\mathbf{w},t}\, \bar{\mathbf{Z}}_t\right)^{-1} \mathbf{K}_{\boldsymbol{z}}\left(\mathbf{I}_o - \bar{\mathbf{W}}_t\, \mathbf{K}_{\mathbf{w},t}\right), \\
\hat{\mathbf{K}}_{\mathbf{w},t} &= \left(\mathbf{I}_{d_{\text{last}}} - \mathbf{K}_{\mathbf{w},t}\, \bar{\mathbf{Z}}_t\, \mathbf{K}_{\boldsymbol{z},t}\, \bar{\mathbf{W}}_t\right)^{-1} \mathbf{K}_{\mathbf{w}}\left(\mathbf{I}_o - \bar{\mathbf{Z}}_t\, \mathbf{K}_{\boldsymbol{z},t}\right),
\end{aligned} \tag{5.27}$$

where $\mathbf{I}_m$ *is an* $m \times m$ *identity matrix.*

Proof. For the precision matrix of the last-layer parameters, we obtain

$$\begin{aligned}
\boldsymbol{\Gamma}_t^{-1} &= \boldsymbol{\Gamma}_{t-1}^{-1} + 2\, \nabla_{\boldsymbol{\Gamma}_t} \mathcal{E}_t \\
&= \boldsymbol{\Gamma}_{t-1}^{-1} + \bar{\mathbf{W}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \bar{\mathbf{W}}_t.
\end{aligned} \tag{5.28}$$

Next, the precision matrix for the subspace hidden layer parameters take the form

$$\begin{aligned}
{\boldsymbol{\Sigma}_t}^{-1} &= \boldsymbol{\Sigma}_{t-1}^{-1} + 2\, \nabla_{\boldsymbol{\Sigma}_t} \mathcal{E}_t \\
&= \boldsymbol{\Sigma}_{t-1}^{-1} + \bar{\mathbf{Z}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \bar{\mathbf{Z}}_t.
\end{aligned} \tag{5.29}$$

The posterior mean of the last layer of model parameters take the form

$$\begin{aligned}
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} - \boldsymbol{\Sigma}_{t-1} \nabla_{\boldsymbol{\mu}_t} \mathcal{E}_t \\
&= \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1} \bar{\mathbf{Z}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t, \boldsymbol{\nu}_t)\right).
\end{aligned} \tag{5.30}$$

Similarly, the posterior mean of the subspace model parameters take the form

$$\begin{aligned}
\boldsymbol{\nu}_t &= \boldsymbol{\nu}_{t-1} - \boldsymbol{\Gamma}_{t-1} \nabla_{\boldsymbol{\nu}_t} \mathcal{E}_t \\
&= \boldsymbol{\nu}_{t-1} + \boldsymbol{\Gamma}_{t-1} \bar{\mathbf{W}}_t{}^{\intercal}\, \bar{\mathbf{R}}_t^{-1}\, \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t, \boldsymbol{\nu}_t)\right).
\end{aligned} \tag{5.31}$$

Hence, we need to solve the system of equations

$$\boldsymbol{\nu}_t = \boldsymbol{\nu}_{t-1} + \boldsymbol{\Gamma}_{t-1}\bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t, \boldsymbol{\nu}_t)\right), \tag{5.32}$$

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \boldsymbol{\Sigma}_{t-1}\bar{\mathbf{Z}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t, \boldsymbol{\nu}_t)\right). \tag{5.33}$$

In the rest of the proof, we show the solution for $\boldsymbol{\nu}_t$. Similar steps yield the solution for $\boldsymbol{\mu}_t$. We begin by expanding the right hand side of (5.32). We obtain

$$\begin{aligned}\boldsymbol{\nu}_t &= \boldsymbol{\nu}_{t-1} + \boldsymbol{\Gamma}_{t-1}\bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - \bar{h}_t(\boldsymbol{\mu}_t, \boldsymbol{\nu}_t)\right) \\ &= \boldsymbol{\nu}_{t-1} + \boldsymbol{\Gamma}_{t-1}\bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - [\kappa_t + \bar{\mathbf{Z}}_t\left(\boldsymbol{z} - \boldsymbol{\mu}_{t-1}\right) + \bar{\mathbf{W}}_t\left(\mathbf{w}_t - \boldsymbol{\nu}_{t-1}\right)]\right).\end{aligned} \tag{5.34}$$

Expanding terms and grouping for $\boldsymbol{\nu}_t$ and $\boldsymbol{\nu}_{t-1}$, we obtain

$$\begin{aligned}\boldsymbol{\nu}_t &= \boldsymbol{\nu}_{t-1} + \left[\boldsymbol{\Gamma}_{t-1}^{-1} + \bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \bar{\mathbf{W}}_t\right]^{-1} \bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{Z}}_t\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\right)\right) \\ &= \boldsymbol{\nu}_{t-1} + \boldsymbol{\Gamma}_t\bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{Z}}_t\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\right)\right) \\ &= \boldsymbol{\nu}_{t-1} + \mathbf{K}_{\mathbf{w}_t,t} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{Z}}_t\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\right)\right),\end{aligned} \tag{5.35}$$

where we defined $\mathbf{K}_{\mathbf{w}_t,t} = \boldsymbol{\Gamma}_t\bar{\mathbf{W}}_t^\intercal \bar{\mathbf{R}}_t^{-1}$.

Similarly, for the subspace hidden-layer parameters, we obtain

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \mathbf{K}_{\boldsymbol{z},t} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{W}}\left(\boldsymbol{\nu}_t - \boldsymbol{\nu}_{t-1}\right)\right) \tag{5.36}$$

with $\mathbf{K}_{\boldsymbol{z},t} = \boldsymbol{\Sigma}_t \bar{\mathbf{Z}}_t^\intercal \bar{\mathbf{R}}_t^{-1}$.

The terms in (5.35) and (5.36) resemble a Kalman filter update, as described in Proposition 2.15. However, these terms are interdependent, as each relies on the unknown variables $\boldsymbol{\mu}_t$ and $\boldsymbol{\nu}_t$. In what follows, we derive a solution to this system of equations.

From (5.36), note that

$$\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1} = \mathbf{K}_{\boldsymbol{z},t} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{W}}\left(\boldsymbol{\nu}_t - \boldsymbol{\nu}_{t-1}\right)\right). \tag{5.37}$$

Then, (5.35) takes the form

$$\boldsymbol{\nu}_t = \boldsymbol{\nu}_{t-1} + \mathbf{K}_{\mathbf{w}_t,t} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{Z}}_t \left(\mathbf{K}_{\boldsymbol{z},t} \left(\boldsymbol{y}_t - \kappa_t - \bar{\mathbf{W}}\left(\boldsymbol{\nu}_t - \boldsymbol{\nu}_{t-1}\right)\right)\right)\right). \tag{5.38}$$

Expanding and grouping the terms $\boldsymbol{\nu}_t$ and $\boldsymbol{\nu}_{t-1}$, we obtain

$$\boldsymbol{\nu}_t = \boldsymbol{\nu}_{t-1} + \left(\mathbf{I}_{d_{\text{last}}} - \mathbf{K}_{\mathbf{w}_t,t}\bar{\mathbf{Z}}_t\mathbf{K}_{\boldsymbol{z}}\bar{\mathbf{W}}_t\right)^{-1} \mathbf{K}_{\mathbf{w}_t,t}\left(\mathbf{I}_o - \bar{\mathbf{W}}_t\mathbf{K}_{\boldsymbol{z},t}\right)\left(\boldsymbol{y}_t - \kappa_t\right). \tag{5.39}$$

A similar derivation yields $\boldsymbol{\mu}_t$. □

Theorem 5.3 demonstrates that the update equations for PULSE share a structural resemblance with those of the extended Kalman filter (EKF) and the exponential family EKF (expfamEKF), as introduced in Sections 2.6.1 and 2.6.3. However, a key distinction lies

in the modification of the gain matrix, which is adjusted to account for the dependencies between the last-layer parameters and the subspace hidden-layer parameters.

Algorithm 14 provides pseudocode for the update step for PULSE.

Algorithm 14 Update step for PULSE

Require: $\boldsymbol{y}_t$ // measurement at time t

Require: $\mathbf{R}_t$ // observation variance at time t

Require: $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ // previous posterior mean and covariance for subspace parameters

Require: $(\boldsymbol{\nu}_{t-1}, \boldsymbol{\Gamma}_{t-1})$ // previous posterior mean and covariance for the last-layer parameters

$\hat{\boldsymbol{y}}_t \leftarrow h\big((\boldsymbol{\mu}_{t-1}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big)$

$\bar{\mathbf{Z}}_t \leftarrow \nabla_{\boldsymbol{z}} h\big((\boldsymbol{z}, \boldsymbol{\nu}_{t-1}), \boldsymbol{x}_t\big)|_{\boldsymbol{z}=\boldsymbol{\mu}_{t-1}}$

$\bar{\mathbf{W}}_t \leftarrow \nabla_{\mathbf{w}} h\big((\boldsymbol{\mu}_{t-1}, \mathbf{w}), \boldsymbol{x}_t\big)|_{\mathbf{w}=\boldsymbol{\nu}_{t-1}}$

$\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}_{t-1}^{-1} + \bar{\mathbf{Z}}_t^{\mathsf{T}} \mathbf{R}_t^{-1} \bar{\mathbf{Z}}_t$

$\boldsymbol{\Gamma}_t^{-1} = \boldsymbol{\Gamma}_{t-1}^{-1} + \bar{\mathbf{W}}_t^{\mathsf{T}} \mathbf{R}_t^{-1} \bar{\mathbf{W}}_t$

// posterior mean updates

$\mathbf{K}_{\boldsymbol{z},t} = \boldsymbol{\Sigma}_t \bar{\mathbf{Z}}_t^{\mathsf{T}} \mathbf{R}_t^{-1}$

$\mathbf{K}_{\mathbf{w},t} = \boldsymbol{\Gamma}_t \bar{\mathbf{W}}_t^{\mathsf{T}} \mathbf{R}_t^{-1}$

$\hat{\mathbf{K}}_{\boldsymbol{z},t} = \left(\mathbf{I}_{d_{\text{hidden}}} - \mathbf{K}_{\boldsymbol{z},t} \bar{\mathbf{W}}_t \mathbf{K}_{\mathbf{w},t} \bar{\mathbf{Z}}_t\right)^{-1} \mathbf{K}_{\boldsymbol{z}} \left(\mathbf{I}_o - \bar{\mathbf{W}}_t \mathbf{K}_{\mathbf{w},t}\right)$

$\hat{\mathbf{K}}_{\mathbf{w},t} = \left(\mathbf{I}_{d_{\text{last}}} - \mathbf{K}_{\mathbf{w},t} \bar{\mathbf{Z}}_t \mathbf{K}_{\boldsymbol{z},t} \bar{\mathbf{W}}_t\right)^{-1} \mathbf{K}_{\mathbf{w}} \left(\mathbf{I}_o - \bar{\mathbf{Z}}_t \mathbf{K}_{\boldsymbol{z},t}\right)$

$\boldsymbol{\nu}_t \leftarrow \boldsymbol{\nu}_{t-1} + \hat{\mathbf{K}}_{\mathbf{w},t} \left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\right)$

$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t-1} + \hat{\mathbf{K}}_{\boldsymbol{z},t} \left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\right)$

5.3 The low-rank extended Kalman filter

The low-rank Kalman filter (LoFi) introduced in Chang et al. (2023) approximates the the distribution over model parameters as Gaussian, $q(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right)$. Here, the posterior precision is diagonal plus low rank (DLR), i.e., it has the form $\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Upsilon}_t + \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}}$, where $\boldsymbol{\Upsilon}_t$ is diagonal and $\mathbf{W}_t$ is a $D \times d$ matrix. Here, we seek to work with $\boldsymbol{\Upsilon}_t$ and $\mathbf{W}_t$. In this sense, we seek "predict and update equations" that depend on these terms, rather than the covariance matrix itself.

Below, we show an efficient recursive form to estimate the terms that comprise the DLR posterior precision matrix, as well as the posterior mean. This has two main steps—a predict step and an update step. The predict step takes $O(D\,d^2 + d^3)$ time, And the update step takes $O(D(d+o)^2)$ time. We define these steps below:

$$\begin{aligned} p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}) &= \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1}, \hat{q}\,\mathbf{I}), \\ p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_{t-1}) &= \mathcal{N}(\boldsymbol{y}_t \,|\, \mathbf{H}_t \boldsymbol{\theta}_t, \mathbf{R}_t), \end{aligned} \tag{5.40}$$

with $\hat{q} \geq 0$.

5.3.1 Predict step

Here, we derive a predict step that makes use of the DLR structure of the covariance matrix.

Proposition 5.4 (Posterior predictive covariance matrix)**.** *Suppose* $\boldsymbol{\Sigma}_{t-1}^{-1} = \boldsymbol{\Upsilon}_{t-1} + \mathbf{W}_{t-1}\mathbf{W}_{t-1}^{\intercal}$. *Then, under* (5.40)*, the predict step* $p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t-1})$ *is of Gaussian form with mean* $\boldsymbol{\mu}_{t-1}$ *and covariance*

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1}\mathbf{B}_{t|t-1}\mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}, \tag{5.41}$$

where

$$\boldsymbol{\Upsilon}_{t|t-1}^{-1} = \boldsymbol{\Upsilon}_{t-1}^{-1} + \hat{q}\,\mathbf{I}_D, \tag{5.42}$$

$$\mathbf{B}_{t|t-1} = \left(\mathbf{I}_d + \mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1}\right)^{-1}. \tag{5.43}$$

Proof. Following Proposition (2.13), we obtain $\boldsymbol{\mu}_{t|t-1} = \boldsymbol{\mu}_{t-1}$. Next,

$$\begin{aligned}
\boldsymbol{\Sigma}_{t|t-1} &= \boldsymbol{\Sigma}_{t-1} + \hat{q}\,\mathbf{I}_D \\
&= \left(\boldsymbol{\Upsilon}_{t-1} + \mathbf{W}_{t-1}\mathbf{W}_{t-1}^{\intercal}\right)^{-1} + \hat{q}\,\mathbf{I}_D \\
&= \left(\boldsymbol{\Upsilon}_{t-1}^{-1} + \hat{q}\,\mathbf{I}_D\right) - \boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1}\left(\mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1} + \mathbf{I}_d\right)^{-1}\mathbf{W}_{t-1}\boldsymbol{\Upsilon}_{t-1}^{-1} \\
&= \boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1}\mathbf{B}_{t|t-1}\mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1},
\end{aligned} \tag{5.44}$$

where $\boldsymbol{\Upsilon}_{t|t-1}$ and $\mathbf{B}_{t|t-1}$ are defined in (5.42) and (5.43) respectively. □

Next, we provide predict-step equations that depend on $\boldsymbol{\Upsilon}_{t-1}$ and $\mathbf{W}_{t-1}$ only. To do this, we work with the precision matrix.

Lemma 5.5. *For* $\mathbf{B}_{t|t-1}$ *defined in* (5.43)*,* $\boldsymbol{\Upsilon}_{t-1}$ *and* $\boldsymbol{\Upsilon}_{t|t-1}$ D*-dimensional diagonal covariance matrices, and* $\mathbf{W}_{t-1}$ *a* $D \times d$ *matrix. The following identity holds*

$$\begin{aligned}
\mathbf{C}_t^{-1} &:= \mathbf{B}_{t|t-1}^{-1} - \mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1} \\
&= \mathbf{I}_d + \mathbf{W}_{t-1}^{\intercal}\left(\boldsymbol{\Upsilon}_{t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1}\boldsymbol{\Upsilon}_{t-1}^{-1}\right)\mathbf{W}_{t-1}.
\end{aligned} \tag{5.45}$$

Proof. The proof follows through algebraic manipulation:

$$\begin{aligned}
\mathbf{C}_t^{-1} &= \mathbf{B}_{t|t-1}^{-1} - \mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1} \\
&= \left(\mathbf{I}_d + \mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1}\right) - \mathbf{W}_{t-1}^{\intercal}\boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1}\boldsymbol{\Upsilon}_{t-1}^{-1}\mathbf{W}_{t-1} \\
&= \mathbf{I}_d + \mathbf{W}_{t-1}^{\intercal}\left(\boldsymbol{\Upsilon}_{t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1}\boldsymbol{\Upsilon}_{t-1}^{-1}\right)\mathbf{W}_{t-1}.
\end{aligned} \tag{5.46}$$

□

Proposition 5.6 (Posterior predictive precision matrix)**.** *The predicted precision matrix derived from Proposition 5.4 takes the form*

$$\boldsymbol{\Sigma}_{t|t-1}^{-1} = \boldsymbol{\Upsilon}_{t|t-1} + \mathbf{W}_{t|t-1}\,\mathbf{W}_{t|t-1}^{\intercal}, \tag{5.47}$$

where

$$\boldsymbol{\Upsilon}_{t|t-1} = \left(\boldsymbol{\Upsilon}_{t-1}^{-1} + \hat{q}\,\mathbf{I}_D\right)^{-1}, \tag{5.48}$$

$$\mathbf{W}_{t|t-1} = \boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\,\mathbf{C}_t^{1/2}. \tag{5.49}$$

Here $\mathbf{A}^{1/2}$ refers to the Cholesky decomposition of a positive definite matrix $\mathbf{A}$.

Proof. Consider the posterior predictive covariance matrix derived in Proposition 5.4. Then, the posterior predictive precision matrix takes the form

$$\begin{aligned}
&\boldsymbol{\Sigma}_{t|t-1}^{-1} \\
&= \left(\boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\mathbf{B}_{t|t-1}\,\mathbf{W}_{t-1}^{\intercal}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\right)^{-1} \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\left(\mathbf{B}_{t|t-1}^{-1} - \mathbf{W}_{t-1}^{\intercal}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\right)^{-1}\mathbf{W}_{t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1} \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\,\mathbf{C}_t^{-1}\,\mathbf{W}_{t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\boldsymbol{\Upsilon}_{t|t-1} \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \mathbf{W}_{t|t-1}\,\mathbf{W}_{t|t-1}^{\intercal}.
\end{aligned} \tag{5.50}$$

Here, $\mathbf{C}_t$ is derived in Lemma 5.5 and $\mathbf{W}_{t|t-1}$ is defined in (5.49). □

Proposition 5.6 derives predict steps in terms of $\boldsymbol{\Upsilon}_{t-1}$ and $\mathbf{W}_{t-1}$, so that dependence in $\boldsymbol{\Sigma}_{t|t-1}$ is implicit. The computational cost of the predict step is $O(D\,d+d^3)$. Algorithm 15 provides pseudocode for the predict step.

Algorithm 15 Predict step for LoFi

Require: $\boldsymbol{\mu}_{t-1}$ // previous mean

Require: $(\boldsymbol{\Upsilon}_{t-1}, \mathbf{W}_{t-1})$ // previous diagonal and low-rank parts

Require: q_t // scalar dynamics covariance

$\boldsymbol{\mu}_{t|t-1} \leftarrow \boldsymbol{\mu}_{t-1}$

$\boldsymbol{\Upsilon}_{t|t-1} \leftarrow \left(\boldsymbol{\Upsilon}_{t-1}^{-1} + q_t\,\mathbf{I}_D\right)^{-1}$

$\mathbf{C}_t^{-1} \leftarrow \mathbf{I}_d + \mathbf{W}_{t-1}^{\intercal}\left(\boldsymbol{\Upsilon}_{t-1}^{-1} - \boldsymbol{\Upsilon}_{t-1}^{-1}\,\boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\right)\mathbf{W}_{t-1}$

$\mathbf{W}_{t|t-1} \leftarrow \boldsymbol{\Upsilon}_{t|t-1}\,\boldsymbol{\Upsilon}_{t-1}^{-1}\,\mathbf{W}_{t-1}\,\mathbf{C}_t^{1/2}$

$\hat{\boldsymbol{y}}_t \leftarrow h(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{x}_t)$ // one-step-ahead forecast

5.3.2 Update step

The LoFi update step makes use of the DLR form of the precision matrix.

Proposition 5.7. *The form of the posterior mean and posterior covariance after an update step takes the form*

$$\begin{aligned}
\boldsymbol{e}_t &= \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\,, \\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \left[\boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t|t-1}^{-1}\,\tilde{\mathbf{W}}_t\left(\mathbf{I}_{d+o} + \tilde{\mathbf{W}}_t^{\intercal}\,\boldsymbol{\Upsilon}_{t|t-1}^{-1}\,\tilde{\mathbf{W}}_t\right)^{-1}\tilde{\mathbf{W}}_t^{\intercal}\,\boldsymbol{\Upsilon}_{t|t-1}^{-1}\right]\mathbf{H}_t^{\intercal}\,\mathbf{R}_t^{-1}\,\boldsymbol{e}_t, \\
\boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Upsilon}_{t|t-1} + \tilde{\mathbf{W}}_t\,\tilde{\mathbf{W}}_t^{\intercal},
\end{aligned} \tag{5.51}$$

where

$$\tilde{\mathbf{W}}_{t|t-1} = \begin{bmatrix}\mathbf{W}_{t|t-1} & \mathbf{H}_t\,\mathbf{R}_t^{1/2}\end{bmatrix}, \tag{5.52}$$

$\mathbf{W}_{t|t-1}$ *is given by* (5.49) *and* $\boldsymbol{\Upsilon}_{t|t-1}$ *is given by* (5.48).

Proof. Following the Kalman filter update for the precision matrix shown in Proposition 2.15, we obtain

$$\begin{aligned}
\boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t|t-1}^{-1} + \mathbf{H}_t\,\mathbf{R}_t^{-1}\,\mathbf{H}_t \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \mathbf{W}_{t|t-1}\,\mathbf{W}_{t|t-1}^{\intercal} + \mathbf{H}_t^{\intercal}\,\mathbf{R}_t^{-\intercal/2}\,\mathbf{R}_t^{-1/2}\,\mathbf{H}_t \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \begin{bmatrix}\mathbf{W}_{t|t-1} & \mathbf{H}_t\,\mathbf{R}_{t-1}^{\intercal}\end{bmatrix}\begin{bmatrix}\mathbf{W}_{t|t-1}^{\intercal} \\ \mathbf{R}_t^{-1/2}\,\mathbf{H}_t\end{bmatrix} \\
&= \boldsymbol{\Upsilon}_{t|t-1} + \tilde{\mathbf{W}}_t\,\tilde{\mathbf{W}}_t^{\intercal},
\end{aligned} \tag{5.53}$$

where $\tilde{\mathbf{W}}_t$ is given by (5.52).

Next, using the Woodbury identity matrix, $\boldsymbol{\Sigma}_t$ takes the form

$$\begin{aligned}
\boldsymbol{\Sigma}_t &= \left(\boldsymbol{\Upsilon}_{t|t-1} + \tilde{\mathbf{W}}_t\,\tilde{\mathbf{W}}_t^{\intercal}\right)^{-1} \\
&= \boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t|t-1}^{-1}\,\tilde{\mathbf{W}}_t\left(\mathbf{I}_{d+o} + \tilde{\mathbf{W}}_t^{\intercal}\boldsymbol{\Upsilon}_{t|t-1}^{-1}\tilde{\mathbf{W}}_t\right)^{-1}\tilde{\mathbf{W}}_t{}^{\intercal}\,\boldsymbol{\Upsilon}_{t|t-1}^{-1}.
\end{aligned} \tag{5.54}$$

The proof concludes from the update step of the Kalman filter shown in Proposition 2.15 with covariance given by (5.54). □

After the update step of Proposition (5.7), the low-rank component $\tilde{\mathbf{W}}$ is a $D\times(d+o)$ matrix. To maintain a $D\times d$ low-rank matrix, LoFi performs a singular value decomposition (SVD) over the $D \times (d + o)$ low rank matrix $\tilde{\mathbf{W}}_t$ and maintains the top d singular components.

Algorithm 16 provides pseudocode for the update step and subsequent low-rank projection of the matrix $\tilde{\mathbf{W}}_t$.

Algorithm 16 Update step for LoFi

Require: $\boldsymbol{y}_t$ // measurement at time t

Require: $\boldsymbol{\mu}_{t|t-1}$ // predicted mean

Require: $(\boldsymbol{\Upsilon}_{t|t-1}, \mathbf{W}_{t|t-1})$ // predicted diagonal and low-rank components

1: $\tilde{\mathbf{W}}_{t|t-1} \leftarrow \begin{bmatrix} \mathbf{W}_{t|t-1} & \mathbf{H}_t \, \mathbf{R}_t^{1/2} \end{bmatrix}$

2: $\tilde{\mathbf{P}}_t \leftarrow \boldsymbol{\Upsilon}_{t|t-1}^{-1} - \boldsymbol{\Upsilon}_{t|t-1}^{-1} \, \tilde{\mathbf{W}}_t \left(\mathbf{I}_{d+o} + \tilde{\mathbf{W}}_t^\intercal \, \boldsymbol{\Upsilon}_{t|t-1}^{-1} \, \tilde{\mathbf{W}}_t \right)^{-1} \tilde{\mathbf{W}}_t^\intercal \, \boldsymbol{\Upsilon}_{t|t-1}^{-1}$

3: $\tilde{\mathbf{K}}_t \leftarrow \tilde{\mathbf{P}}_t \, \mathbf{H}_t^\intercal \, \mathbf{R}^{-1}$

4: $\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_{t|t-1} + \tilde{\mathbf{K}}_t \, (\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)$

5: // build low-rank component (using reduced-rank SVD)

6: $\tilde{\mathbf{W}}_t^\intercal \, \tilde{\mathbf{W}}_t \leftarrow \tilde{\mathbf{V}}_t \, \tilde{\mathbf{S}}_t^2 \, \tilde{\mathbf{V}}_t^\intercal$ // right-singular vectors

7: $\tilde{\mathbf{U}}_t \leftarrow \tilde{\mathbf{W}} \, \tilde{\mathbf{V}} \, \tilde{\mathbf{S}}_t^{-1}$ // left singular vectors

8: $\mathbf{W}_t \leftarrow \tilde{\mathbf{U}}_{:,:d} \, (\tilde{\mathbf{S}}_t)_{:d,:}$ // low-rank construction

9: $(\mathbf{D}_t)_i \leftarrow \sum_{j=d}^{d+o} (\tilde{\mathbf{U}})_{i,j} \, (\tilde{\mathbf{S}}_t)_j \, (\tilde{\mathbf{U}})_{i,j} \, (\tilde{\mathbf{S}}_t)_j$ // dropped variance

10: $\boldsymbol{\Upsilon}_t \leftarrow \boldsymbol{\Upsilon}_{t|t-1} + \mathbf{D}_t$

In Algorithm 16, Line 7 is computed for all $i = 1, \ldots, D$ and $j = 1, \ldots, d$; and Line 8 is computed for all $i = 1, \ldots, D$. Finally $\mathbf{A}_{:,d:} = \begin{bmatrix} \mathbf{A}_{:,1} & \ldots & \mathbf{A}_{:,d} \end{bmatrix}$ and similarly for $\mathbf{A}_{d:,:}$.

5.4 Summary of methods

In this section, we summarise the methods introduced in this chapter, namely, the subspace method, the PULSE method, and the LoFi method.

Method	Assumption	Time complexity	Memory complexity
subspace	$\boldsymbol{\theta}_t = \mathbf{A} \, \boldsymbol{z}_t + \boldsymbol{\psi}_*$	$O(d \, D + d^3)$	$O(d \, D + d^2)$
LoFi	$\boldsymbol{\Sigma}_t^{-1} = \mathbf{W}_t \, \mathbf{W}_t^\intercal + \boldsymbol{\Upsilon}_t$	$O(D \, (d+o)^2 + (d+o)^3)$	$O(d + d \, D)$
PULSE	$\boldsymbol{\theta}_t = (\mathbf{A} \, \boldsymbol{z}_t + \boldsymbol{\psi}_*, \mathbf{w}_t)$	$O(d_{\text{hidden}}^3 + d_{\text{last}}^3)$	$O(D \, d_{\text{hidden}} + d_{\text{hidden}}^2 + d_{\text{last}}^2)$

Table 5.1: Time and memory complexity of the update step for various methods. The row *Assumption* denotes the change from the assumptions in used in the EKF algorithm.

5.5 Experiments

In this section, we present empirical results in which we evaluate the performance and speed (time to run) of the methods presented in this chapter. We also study the effects of various hyper-parameters of our algorithm, such as how we choose the subspace.

5.5.1 Online classification with a convolutional neural network

In this experiment, we consider consider the problem of one-step-ahead classification of a stream of images coming from the FashionMNIST dataset (Xiao et al., 2017). We test the

Subspace method, the PULSE method, and the LoFi method to train a modified LeNet5 architecture with ReLU activation; we add an additional 20-unit dense layer (LeCun et al., 1998).

For each of the methods presented in this chapter, we consider dimensions of sizes $d \in \{1, 5, 10, 25, 50, 70, 100\}$, which corresponds to the subspace parameters for the Subspace method (Section 5.1), the hidden-layer subspace for PULSE (Section 5.2), and the low-rank component for LoFi (Section 5.3).

For the subspace and PULSE methods, we estimate the projection matrix $\mathbf{A}$ using the procedure outlined in Section 5.1.2 using a warmup dataset of 2000 samples separate from the training data.

In the following figures we show the one-step-ahead classification accuracy for each of the methods as a function of d, using an exponentially-weighted moving average (EWMA) with a span value of 100 observations.

Subspace: Figure 5.1 shows the result of the Fashion MNIST task for the subspace method. We consider the additional dimensions $\{150, 200, 250\}$ to compare to the total number of parameters in PULSE (see below). We observe that for $d \leq 10$, the method

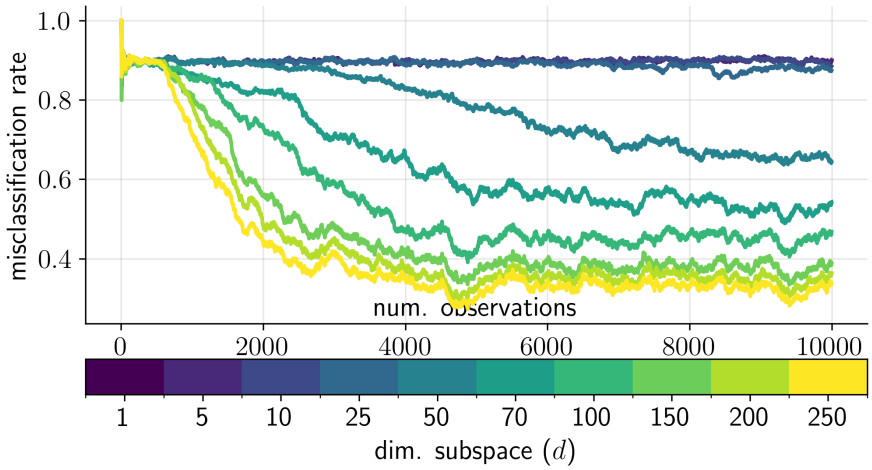

Figure 5.1: Comparison of the prequential accuracy on the Fashion MNIST dataset for the subspace method. The y-axis shows the EWMA accuracy using a span of 100 observations.

performs no better than a random classifier, which corresponds to a misclassification rate of 0.9. Then, for $d > 10$, the method improves its performance as d increases. This is because we allow the algorithm to consider higher degrees of freedom.

PULSE: Next, Figure 5.2 shows the results for the PULSE method. Here d is the dimension of the subspace of the parameters in the hidden-layers. Because the output is 10-dimensional and the second-to-last layer has 20 units, the total number of parameters that get updated is $200 + d$. As a consequence, the performance of the model places strong emphasis on the last layer parameters. We observe that $d = 1$ provides a much

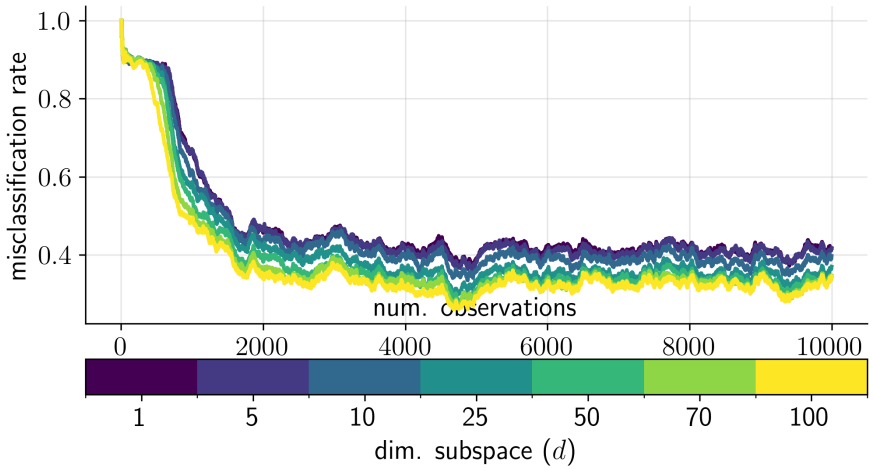

Figure 5.2: Comparison of the prequential accuracy on the Fashion MNIST dataset for the PULSE method. The y-axis shows the EWMA accuracy using a span of 100 observations.

better result than the subspace counterpart. However, the performance of the method as we increase d is not as stark as in the subspace case shown in Figure 5.1.

LoFi: Finally, Figure 5.3 the results for the LoFi method. Here d is the rank of the DLR matrix that characterises the posterior precision matrix. For LoFi, all D model parameters get updated, however, the total number of meta-parameters required to perform the update step is $(D + D\,d)$: D diagonal terms and $D\,d$ low-rank terms. We observe that

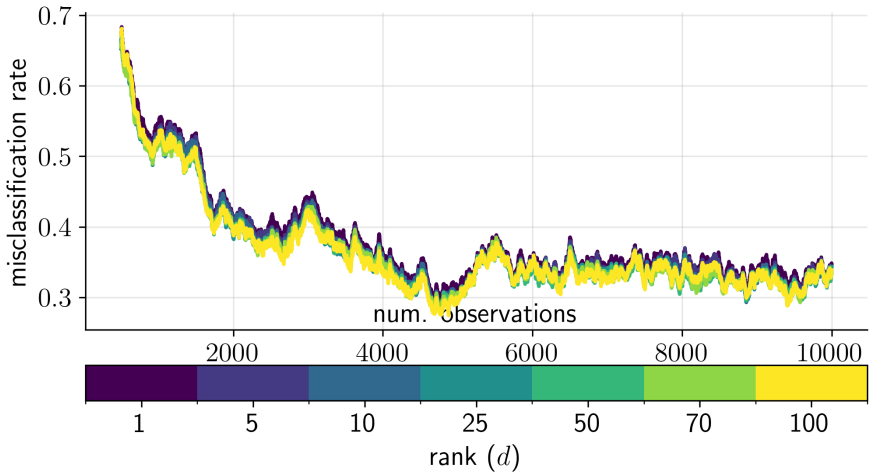

Figure 5.3: Comparison of the prequential accuracy on the Fashion MNIST dataset. The y-axis shows the EWMA accuracy using a span of 100 observations.

for this dataset, the total rank does not have a strong influence on the performance of the method as the rank increases, relative to the Subspace and the PULSE method.

Showdown: Figure 5.4 compares the subspace, PULSE, and LoFi method on the Fashion MNIST task. The x-axis shows the running time in seconds and the y-axis shows the final one-step-ahead misclassification rate. Each marker corresponds to the final misclassification rate. The lines correspond to the median performance across choices of d. We observe that LoFi is the method that maintains a consistent misclassification rate

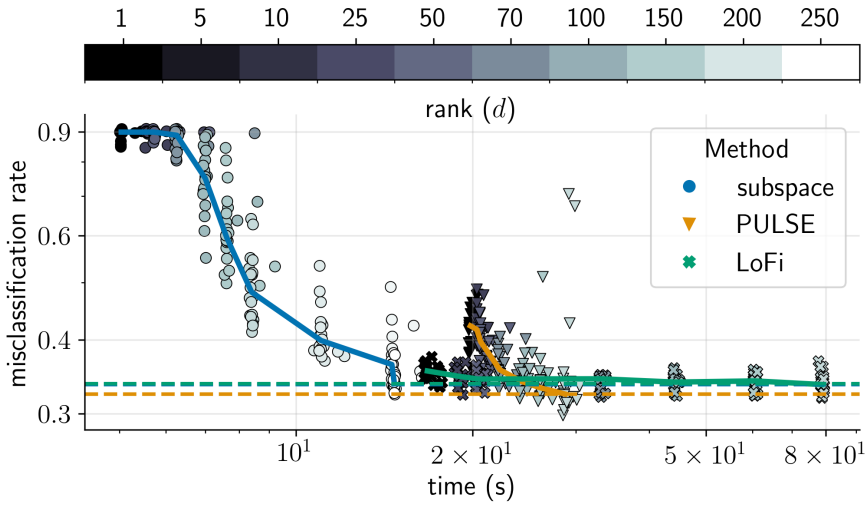

Figure 5.4: Comparison of the prequential misclassification rate on the Fashion MNIST dataset on the last 8000 observations. The dashed lines denote the best-performing configuration for each method.

as a function of d. However, the running time increases significantly as we increase d. Next, PULSE starts a higher misclassification rate than LoFi. However, at $d = 70$ PULSE matches the performance of LoFi at around half the running time of Lofi. Finally, the subspace method has the highest variability among the competing methods. However, its performance significantly improves for $d \geq 100$ and matches the performance of LoFi with less running time.

5.6 Conclusion

In this chapter, we presented three methods for scalable filtering in high-dimensional parametric models, such as those found in neural networks. Specifically, we introduced: (i) the subspace method, which projects the weights of a neural network into a lower-dimensional subspace and performs filtering within this reduced space; (ii) the PULSE method, which extends the subspace method by defining a density function over the subspace and the last-layer parameters found using variational optimisation; and (iii) the LoFi method, which tracks model parameters using a low-rank plus diagonal representation of the posterior precision matrix.

To evaluate these methods, we conducted experiments on an online classification problem using the Fashion MNIST dataset. The results demonstrate the applicability of

these approaches to neural networks, highlighting their potential for scalable and efficient Bayesian filtering in high-dimensional settings.

Chapter 6

Final remarks

This thesis has proposed Bayesian filtering as a principled framework for adaptive, robust, and scalable online learning in the presence of non-stationary environments, misspecified models, and high-dimensional parameters.

The primary contribution of this thesis has been the development of novel filtering methods, demonstrating their effectiveness in addressing sequential problems in machine learning. Specifically, we have introduced: (i) a unified framework for adapting to non-stationary environments, (ii) a novel, lightweight filtering method that is provably robust and a straightforward extension of the Kalman filter, and (iii) a suite of methods designed to reduce the time and memory complexity of classical filters when applied to high-dimensional parametric models.

Despite these contributions, several limitations remain. In particular, in future work, we would like to evaluate the proposed methods on novel architectures such as the Transformer architecture (Vaswani et al., 2017; Moreno-Pino et al., 2024) or Graph Neural Networks (Scarselli et al., 2008; Arroyo et al., 2025) in temporal settings. Furthermore, future work will address (i) reinforcement learning problems, where agents can experience non-stationarity even in static environments (Zhang et al., 2022; Waldon et al., 2024) and (ii) sequential decision making problems in finance (Cartea et al., 2024a; Drissi, 2022; Scalzo et al., 2021; Arroyo et al., 2024).

These limitations present avenues for future research, including fully-online reinforcement learning and temporal problems involving non-stationarity on graph-structured data.

In conclusion, this thesis demonstrates the versatility and potential of Bayesian filtering as a framework for parametric online learning and lays the foundation for further advancements in adaptive, robust, and scalable algorithms. By bridging the gap between traditional Bayesian approaches and the demands of modern machine learning, this work contributes to the development of tools that will become increasingly essential in real-world applications characterised by uncertainty, non-stationarity, and high-dimensionality.

Bibliography

Baptiste Abélès, Joseph de Vilmarest, and Olivier Wintemberger. Adaptive time series forecasting with markovian variance switching, 2024.

Ryan Prescott Adams and David J. C. MacKay. Bayesian online changepoint detection, 2007.

G Agamennoni, J I Nieto, and E M Nebot. Approximate inference in state-space models with heavy-tailed noise. *IEEE Transactions on Signal Processing*, 2012.

Diego Agudelo-España, Sebastian Gomez-Gonzalez, Stefan Bauer, Bernhard Schölkopf, and Jan Peters. Bayesian online prediction of change points. In *Conference on Uncertainty in Artificial Intelligence*, pages 320–329. PMLR, 2020.

Reda Alami. Bayesian change-point detection for bandit feedback in non-stationary environments. In *Asian Conference on Machine Learning*, pages 17–31. PMLR, 2023.

Réda Alami, Odalric Maillard, and Raphael Féraud. Restarted bayesian online change-point detector achieves optimal detection delay. In *International conference on machine learning*, pages 211–221. PMLR, 2020.

Pierre Alquier and James Ridgway. Concentration of tempered posteriors and of their variational approximations. *The Annals of Statistics*, 48(3):1475 – 1497, 2020. doi: 10.1214/19-AOS1855. URL https://doi.org/10.1214/19-AOS1855.

Matias Altamirano, Francois-Xavier Briol, and Jeremias Knoblauch. Robust and scalable Bayesian online changepoint detection. In *International Conference on Machine Learning*, 2023a.

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and conjugate Gaussian process regression. *arXiv:2311.00463*, 2023b.

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and scalable bayesian online changepoint detection, 2023c.

Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

Alif Aqsha, Fayçal Drissi, and Leandro Sánchez-Betancourt. Strategic learning and trading in broker-mediated markets, 2024. URL https://arxiv.org/abs/2412.20847.

Alvaro Arroyo, Bruno Scalzo, Ljubiša Stanković, and Danilo P Mandic. Dynamic portfolio cuts: A spectral approach to graph-theoretic diversification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5468–5472. IEEE, 2022.

Alvaro Arroyo, Alvaro Cartea, Fernando Moreno-Pino, and Stefan Zohren. Deep attentive survival analysis in limit order books: Estimating fill probabilities with convolutional-transformers. *Quantitative Finance*, 24(1):35–57, 2024.

Álvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael Bronstein, and Pierre Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in gnns: Bridging recurrent and graph learning. *arXiv preprint arXiv:2502.10818*, 2025.

Kavosh Asadi, Shoham Sabach, Yao Liu, Omer Gottesman, and Rasool Fakoor. Td convergence: An optimization perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Jordan T Ash and Ryan P Adams. On warm-starting neural network training. In *NIPS*, 2020. URL http://arxiv.org/abs/1910.08475.

Adrian Barbu, Song-Chun Zhu, et al. *Monte carlo methods*, volume 35. Springer, 2020.

A. Barp, F.-X. Briol, A. B. Duncan, M. Girolami, and L. Mackey. Minimum Stein discrepancy estimators. In *Neural Information Processing Systems*, pages 12964–12976, 2019.

Daniel Barry and John A Hartigan. Product partition models for change point problems. *The Annals of Statistics*, pages 260–279, 1992.

Michele Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs, 1993.

Richard H Battin. Space guidance evolution-a personal narrative. *Journal of Guidance, Control, and Dynamics*, 5(2):97–110, 1982.

Matthew Beal, Zoubin Ghahramani, and Carl Rasmussen. The infinite hidden markov model. *Advances in neural information processing systems*, 14, 2001.

Gianluca M Bencomo, Jake C Snell, and Thomas L Griffiths. Implicit maximum a posteriori filtering via adaptive optimization. *arXiv:2311.10580*, 2023.

Philippe Bergault and Leandro Sánchez-Betancourt. A mean field game between informed traders and a broker. *arXiv preprint arXiv:2401.05257*, 2024.

J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley, 1994.

Anirban Bhattacharya, Debdeep Pati, and Yun Yang. Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39 – 66, 2019. doi: 10.1214/18-AOS1712. URL https://doi.org/10.1214/18-AOS1712.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

Ayman Boustati, Omer Deniz Akyildiz, Theodoros Damoulas, and Adam Johansen. Generalised Bayesian filtering via sequential monte carlo. *Advances in neural information processing systems*, 33:418–429, 2020.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8281–8290, 2021.

Wenhan Cao, Tianyi Zhang, Zeju Sun, Chang Liu, Stephen S-T Yau, and Shengbo Eben Li. Nonlinear bayesian filtering with natural gradient gaussian approximation. *arXiv [eess.SY]*, October 2024. URL http://arxiv.org/abs/2410.15832.

Álvaro Cartea and Leandro Sánchez-Betancourt. Brokers and informed traders: dealing with toxic flow and extracting trading signals. *Available at SSRN 4265814*, 2022.

Álvaro Cartea, Fayçal Drissi, and Pierre Osselin. Bandits for algorithmic trading with signals. *Available at SSRN 4484004*, 2023a.

Álvaro Cartea, Fayçal Drissi, and Marcello Monga. Decentralized finance and automated market making: Predictable loss and optimal liquidity provision. *SIAM Journal on Financial Mathematics*, 15(3):931–959, 2024a.

Álvaro Cartea, Sebastian Jaimungal, and Leandro Sánchez-Betancourt. Nash equilibrium between brokers and traders. *arXiv preprint arXiv:2407.10561*, 2024b.

Álvaro Cartea, Gerardo Duran-Martin, and Leandro Sánchez-Betancourt. Detecting toxic flow, 2023b.

Wassim S Chaer, Robert H Bishop, and Joydeep Ghosh. A mixture-of-experts framework for adaptive kalman filtering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(3):452–464, 1997.

Chaw-Bing Chang and Michael Athans. State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(3): 418–425, 1978.

Peter G Chang, Kevin Patrick Murphy, and Matt Jones. On diagonal approximations to the extended kalman filter for online training of bayesian neural networks. In *Continual Lifelong Learning Workshop at ACML 2022*, 2022.

Peter G Chang, Gerardo Duran-Martin, Alex Shestopaloff, Matt Jones, and Kevin Patrick Murphy. Low-rank extended kalman filtering for online learning of neural networks from streaming data. In *Conference on Lifelong Learning Agents*, pages 1025–1071. PMLR, 2023.

Wilson Ye Chen, Alessandro Barp, François-Xavier Briol, Jackson Gorham, Mark Girolami, Lester Mackey, and Chris Oates. Stein point markov chain Monte Carlo. In *International Conference on Machine Learning*, pages 1011–1021, 2019.

Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. Autodifferentiable ensemble kalman filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, 2022. doi: 10.1137/21M1434477. URL https://doi.org/10.1137/21M1434477.

Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

Don Coppersmith, Alan J Hoffman, and Uriel G Rothblum. Inequalities of rayleigh quotients and bounds on the spectral radius of nonnegative symmetric matrices. *Linear algebra and its applications*, 263:201–220, 1997.

Joao FG de Freitas, Mahesan Niranjan, Andrew H. Gee, and Arnaud Doucet. Sequential monte carlo methods to train neural network models. *Neural computation*, 12(4): 955–993, 2000.

Gianluca Detommaso, Tiangang Cui, Youssef Marzouk, Alessio Spantini, and Robert Scheichl. A Stein variational newton method. *Advances in Neural Information Processing Systems*, 2018.

Miheer Dewaskar, Christopher Tosh, Jeremias Knoblauch, and David B Dunson. Robustifying likelihoods by optimistically re-weighting data. *arXiv:2303.10525*, 2023.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *UAI*, 2000. URL http://arxiv.org/abs/1301.3853.

Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

Fayçal Drissi. Solvability of differential riccati equations and applications to algorithmic trading with signals. *Applied Mathematical Finance*, 29(6):457–493, 2022.

Gerardo Duran-Martin, Aleyna Kara, and Kevin Murphy. Efficient online bayesian inference for neural bandits. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 6002–6021. PMLR, 28–30 Mar 2022.

Gerardo Duran-Martin, Matias Altamirano, Alexander Y. Shestpaloff, Leandro Sánchez-Betancourt, Jeremias Knoblauch, Matt Jones, Briol François-Xavier, and Kevin P. Murphy. Outlier-robust kalman filtering through generalised bayes. In *International Conference on Machine Learning*. PMLR, 2024.

Gerardo Duran-Martin, Leandro Sánchez-Betancourt, Alex Shestopaloff, and Kevin Patrick Murphy. A unifying framework for generalised bayesian online learning in non-stationary environments. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=osesw2V10u.

Randall L Eubank. *A Kalman filter primer*. Chapman and Hall/CRC, 2005.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53:343–367, 2003.

Mostafa Farrokhabadi, Jethro Browell, Yi Wang, Stephen Makonin, Wencong Su, and Hamidreza Zareipour. Day-ahead electricity demand forecasting competition: Post-covid paradigm. *IEEE Open Access Journal of Power and Energy*, 9:185–191, 2022. doi: 10.1109/OAJPE.2022.3161101.

Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(4):589–605, 2007.

Paul Fearnhead and Zhen Liu. Efficient bayesian analysis of multiple changepoint models with dependence across segments. *Statistics and Computing*, 21:217–229, 2011.

Paul Fearnhead and Guillem Rigaill. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 114(525):169–183, 2019.

Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. The sticky hdp-hmm: Bayesian nonparametric hidden markov models with persistent states. *Arxiv preprint*, 2, 2007.

Jonathan Frankle. *The Lottery Ticket Hypothesis: On Sparse, Trainable Neural Networks*. PhD thesis, Massachusetts Institute of Technology, 2023.

Alexandre Galashov, Michalis K Titsias, András György, Clare Lyle, Razvan Pascanu, Teh Yee Whye, and Maneesh Sahani. Non-stationary learning of neural networks with automatic soft parameter reset. In *NIPS*, November 2024. URL https://arxiv.org/abs/2411.04034.

Joao Gama, Jesus Aguilar-Ruiz, and Ralf Klinkenberg. Knowledge discovery from data streams. *Intelligent Data Analysis*, 12(3):251–252, 2008.

Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

James-A Goulet, Luong Ha Nguyen, and Saeid Amiri. Tractable approximate gaussian inference for bayesian neural networks. *Journal of Machine Learning Research*, 22(251): 1–23, 2021.

Ido Greenberg, Netanel Yannay, and Shie Mannor. Optimization or architecture: How to hack kalman filtering. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50482–50505. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9dfcc83c01e94d02c751c47517855c9f-Paper-Conference.pdf.

Mohinder S Grewal and Angus P Andrews. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3): 69–78, 2010.

Peter Grünwald. The safe Bayesian. In *International Conference on Algorithmic Learning Theory*, pages 169–183, 2012.

Peter Grünwald and Thijs van Ommen. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4):1069 – 1103, 2017.

Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, and Albert Bifet. Survey on online streaming continual learning. In *IJCAI*, pages 6628–6637, 2023.

Muktesh Gupta, Rajesh Wadhvani, and Akhtar Rasool. Comprehensive analysis of change-point dynamics detection in time series data: A review. *Expert Systems with Applications*, page 123342, 2024.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers, 2024. URL https://arxiv.org/abs/2404.11599.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Chris C Holmes and Stephen G Walker. Assigning a value to a power likelihood in a general Bayesian model. *Biometrika*, 104(2):497–503, 2017.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yulong Huang, Yonggang Zhang, Ning Li, and Jonathon Chambers. A robust Gaussian approximate filter for nonlinear systems with heavy tailed measurement noises. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4209–4213, 2016.

Peter J Huber. Robust statistics. *Wiley Series in Probability and Mathematical Statistics*, 1981.

M Hušková. Gradual changes versus abrupt changes. *Journal of Statistical Planning and Inference*, 76(1-2):109–125, 1999.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In *International conference on artificial intelligence and statistics*, pages 703–711. PMLR, 2021.

MY Ismail and JC Principe. Equivalence between rls algorithms and the ridge regression technique. In *Conference Record of The Thirtieth Asilomar Conference on Signals, Systems and Computers*, pages 1083–1087. IEEE, 1996.

Matt Jones, Peter Chang, and Kevin Murphy. Bayesian online natural gradient (BONG). In *Advances in Neural Information Processing Systems*, May 2024. URL http://arxiv.org/abs/2405.19681.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

RJ Kelly. Reducing geometric dilution of precision using ridge regression. *IEEE transactions on aerospace and electronic systems*, 26(1):154–168, 1990.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal bayesian on-line changepoint detection with model selection. In *International Conference on Machine Learning*, pages 2718–2727. PMLR, 2018.

Jeremias Knoblauch, Jack E Jewson, and Theodoros Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a3f390d88e4c41f2747bfa2f1b5f87db-Paper.pdf.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An optimization-centric view on bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022. URL http://jmlr.org/papers/v23/19-1047.html.

Mark R Kuhl. Ridge regression signal processing. *NASA, Langley Research Center, Joint University Program for Air Transportation Research, 1989-1990*, 1990.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick Van Der Smagt, and Stephan Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2019.

Marc Lambert, Silvère Bonnabel, and Francis Bach. The recursive variational gaussian approximation (r-vga). *Statistics and Computing*, 32(1):10, 2022.

Marc Lambert, Silvère Bonnabel, and Francis Bach. The limited-memory recursive variational gaussian approximation (l-rvga). *Statistics and Computing*, 33(3):70, 2023.

Brett W. Larsen, Stanislav Fort, Nic Becker, and Surya Ganguli. How many degrees of freedom do we need to train deep networks: a loss landscape perspective. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ChMLTGRjFcU.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

C T Leondes. Theory and applications of kalman filtering. Technical Report AGARDOGRAPH-139, NASA, February 1970.

Aodong Li, Alex Boyd, Padhraic Smyth, and Stephan Mandt. Detecting and adapting to irregular distribution shifts in bayesian online learning, 2021.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Stein's lemma for the reparameterization trick with exponential family mixtures. *arXiv preprint arXiv:1910.13398*, 2019.

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. PMLR, 2017.

Bin Liu. Robust sequential online prediction with dynamic ensemble of multiple models: A review. *Neurocomputing*, page 126553, 2023.

Yueyang Liu, Benjamin Van Roy, and Kuang Xu. Nonstationary bandit learning via predictive sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 6215–6244. PMLR, 2023.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

David Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, 1965.

Takuo Matsubara, Jeremias Knoblauch, François-Xavier Briol, and Chris J Oates. Robust generalised Bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society: Series B*, 84(3):997–1022, 2022.

Raman Mehra. Approaches to adaptive filtering. *IEEE Transactions on automatic control*, 17(5):693–698, 1972.

Joseph Mellor and Jonathan Shapiro. Thompson sampling in switching environments with bayesian online change point detection. *arXiv preprint arXiv:1302.3721*, 2013.

Jeffrey W Miller and David B Dunson. Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 2018.

Thomas P Minka. Expectation propagation for approximate bayesian inference. *arXiv preprint arXiv:1301.2294*, 2013.

Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. *Advances in neural information processing systems*, 31, 2018.

Fernando Moreno-Pino, Álvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough transformers: Lightweight and continuous time series modelling through signature patching. *Advances in Neural Information Processing Systems*, 37:106264–106294, 2024.

He-Qing Mu and Ka-Veng Yuen. Novel outlier-resistant extended Kalman filter for robust online structural identification. *Journal of Engineering Mechanics*, 141(1):04014100, 2015.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL http://probml.github.io/book1.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.

Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

Josue Nassar, Jennifer Brennan, Ben Evans, and Kendall Lowrey. Bam: Bayes with adaptive memory. *arXiv preprint arXiv:2202.02405*, 2022.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Yann Ollivier. The extended kalman filter is a natural gradient descent in trajectory space. *arXiv preprint arXiv:1901.00696*, 2019.

Mari Ostendorf, Vassilios V Digalakis, and Owen A Kimball. From hmm's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on speech and audio processing*, 4(5):360–378, 1996.

Václav Peterka. Bayesian approach to system identification. In *Trends and Progress in System identification*, pages 239–304. Elsevier, 1981.

Hans Reimann. Towards robust inference for bayesian filtering of linear gaussian dynamical systems subject to additive change. masterthesis, Universität Potsdam, 2024.

Marina Riabiz, Wilson Chen, Jon Cockayne, Pawel Swietach, Steven A. Niederer, Lester Mackey, and Chris. J. Oates. Optimal thinning of MCMC output. *arXiv:2005.03952*, 2022.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown. In *International conference on learning representations*, volume 9, 2018.

Christian P Robert et al. *The Bayesian choice: from decision-theoretic foundations to computational implementation*, volume 2. Springer, 2007.

Michael Roth, Gustaf Hendeby, Carsten Fritsche, and Fredrik Gustafsson. The ensemble kalman filter: a signal processing perspective. *EURASIP J. Adv. Signal Processing*, 2017(1):56, 2017. URL https://doi.org/10.1186/s13634-017-0492-x.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6, 1986.

Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934, 2010.

Simo Sarkka and Aapo Nummenmaa. Recursive noise adaptive kalman filtering by variational bayesian approximations. *IEEE Transactions on Automatic control*, 54(3):596–600, 2009.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

Bruno Scalzo, Alvaro Arroyo, Ljubiša Stanković, and Danilo P Mandic. Nonstationary portfolios: Diversification in the spectral domain. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5155–5159. IEEE, 2021.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Test-time adaptation with state-space models. *arXiv preprint arXiv:2407.12492*, 2024.

Jonathan Schmidt, Philipp Hennig, Jörg Nick, and Filip Tronarp. The rank-reduced kalman filter: Approximate dynamical-low-rank filtering in high dimensions. *Advances in Neural Information Processing Systems*, 36:61364–61376, 2023.

Jeremy Sellier and Petros Dellaportas. Bayesian online change point detection with hilbert space approximate student-t process. In *International Conference on Machine Learning*, pages 30553–30569. PMLR, 2023.

Sharad Singhal and Lance Wu. Training multilayer perceptrons with the extended kalman algorithm. *Advances in neural information processing systems*, 1, 1988.

Joram Soch, Thomas J Faulkenberry, Kenneth Petrykowski, and Carsten Allefeld. The book of statistical proofs. *Open, Zenodo*, 10, 2020.

Terence Tao. 254a, notes 3a: Eigenvalues and sums of hermitian matrices. https://terrytao.wordpress.com/2010/01/12/254a-notes-3a-eigenvalues-and-sums-of-hermitian-matrices/, 2010. Accessed: 2024-12-18.

Yangtianze Tao and Stephen Shing-Toung Yau. Outlier-robust iterative extended kalman filtering. *IEEE Signal Processing Letters*, 30:743–747, 2023. doi: 10.1109/LSP.2023.3285118.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Jo-Anne Ting, Evangelos Theodorou, and Stefan Schaal. Learning an outlier-robust kalman filter. In *European Conference on Machine Learning*, pages 748–756. Springer, 2007.

Michalis K Titsias, Alexandre Galashov, Amal Rannen-Triki, Razvan Pascanu, Yee Whye Teh, and Jorg Bornschein. Kalman filter for online classification of non-stationary data. In *ICLR*, 2024.

Gerrit JJ Van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Joseph de Vilmarest and Olivier Wintenberger. Viking: variational bayesian variance tracking. *Statistical Inference for Stochastic Processes*, pages 1–22, 2024.

Harrison Waldon, Fayçal Drissi, Yannick Limmer, Uljad Berdica, Jakob Nicolaus Foerster, and Alvaro Cartea. Dare: The deep adaptive regulator for control of uncertain continuous-time systems. In *ICML 2024 Workshop: Foundations of Reinforcement Learning and Control–Connections and Perspectives*, 2024.

Hongwei Wang, Hongbin Li, Jun Fang, and Heping Wang. Robust Gaussian Kalman filter with outlier detection. *IEEE Signal Processing Letters*, 25(8):1236–1240, 2018.

Curt Wells. *The Kalman filter in finance*, volume 32. Springer Science & Business Media, 2013.

Mike West. Robust sequential approximate bayesian estimation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 43(2):157–166, 1981.

Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models*. Springer, 1997.

Robert C Wilson, Matthew R Nassar, and Joshua I Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural computation*, 22(9):2452–2476, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Zhong Peng Yang and Xiao Xia Feng. A note on the trace inequality for products of hermitian matrix power. *Journal of Inequalities in Pure and Applied Mathematics*, 3 (5), 2002.

Tiantian Zhang, Xueqian Wang, Bin Liang, and Bo Yuan. Catastrophic interference in reinforcement learning: A solution based on context division and knowledge distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9925–9939, 2022.

Tong Zhang. *Mathematical Analysis of Machine Learning Algorithms*. Cambridge University Press, 2023. doi: 10.1017/9781009093057.

Fengchi Zhu, Yulong Huang, Chao Xue, Lyudmila Mihaylova, and Jonathon Chambers. A sliding window variational outlier-robust kalman filter based on student's t-noise modeling. *IEEE Transactions on Aerospace and Electronic Systems*, 58(5):4835–4849, 2022.