BONE: a unifying framework for Bayesian online learning in non-stationary environments

Anonymous authors
Paper under double-blind review

Abstract

We propose a unifying framework for methods that perform Bayesian online learning in non-stationary environments. We call the framework BONE, which stands for (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 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. We show how this modularity allows us to write many different existing methods as instances of BONE; we also use this framework to propose a new method. We then experimentally compare existing methods with our proposed new method on several datasets; we provide insights into the situations that make one method more suitable than another for a given task.

1 Introduction

Our goal in this paper is to study adaptive probabilistic methods that learn to make accurate predictions about the next output $\boldsymbol{y}_{t+1}$ based on the next input $\boldsymbol{x}_{t+1}$ and a sequence of past inputs and outputs, $(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t})$, where t indexes time. This is often called prequential forecasting (Gama et al., 2008).

Many prediction methods assume that the data generating process (DGP) $p(\boldsymbol{y}_{t+1} \mid \boldsymbol{x}_{t+1}, \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t})$ is static through time. However, real-world data often comes from non-stationary distributions, where the underlying data distribution changes, either gradually (e.g., rising mean temperature) or abruptly (e.g., price shocks after a major news event). In this paper we propose a unified framework for tackling (one-step-ahead) forecasts in such (potentially) non-stationary environments. Our framework encompasses many existing lines of work, including online continual learning (Dohare et al., 2024), prequential forecasting (Liu, 2023), test-time adaptation (Schirmer et al., 2024), and neural contextual bandits (Riquelme et al., 2018).

The framework we propose in this paper, which we call BONE — which stands for (B)ayesian (O)nline learning in (N)on-stationary (E)nvironments — is based on a form of Bayesian inference in a hierarchical model,[1] and is composed of three modelling choices and two algorithmic choices. The modelling choices are: (M.1) a model for the measurements, (M.2) an auxiliary process to model non-stationarity, and (M.3) a prior over model parameters conditioned on the auxiliary process and the past data. The algorithmic choices

[1] It is worth clarifying what we mean by "a form of Bayesian inference". We focus on adaptive rules, which are defined as "accumulating experience about the properties of the environment and leveraging this experience to improve the model's performance" (Peterka, 1981). While the adaptive rules that we consider in this work align with Bayes' rule, this alignment does not imply that we adopt the full set of assumptions required to be strictly Bayesian. By strictly Bayesian, we mean a framework in which the properties of the data-generating process are fully known, well specified, and described within a formal mathematical model with uncertainty quantified about all unknowns (Faden & Rausser, 1976; Peterka, 1981; Knoblauch et al., 2022). Thus, our approach uses Bayes' rule only to provide a rational basis for sequential decision-making while preserving adaptability.

are: (A.1) an algorithm to compute an approximate posterior over the model parameters given the auxiliary variables, and (A.2) an algorithm to compute an approximate posterior —or more generally, a set of weights— over the auxiliary variables. We show how these different axes of variation span a wide variety of existing and new methods (see Table 2). We then perform an experimental comparison on a variety of tasks. Specifically, we consider prequential forecasting, classification, bandits, and unsupervised time-series segmentation. An easy-to-use library that implements these methods, written in Jax (Bradbury et al., 2018), is available at `[link-here-after-blind-review]`.

To summarise, our contributions are threefold: (1) we provide an extensive literature review on methods that tackle non-stationarity, and show that they can all be written as instances of our unified BONE framework; (2) we use the BONE framework to develop a new method; and (3) we perform an experimental comparison of many existing methods and the new method on environments with both abrupt changes and gradually-changing distributions.

2 The framework

2.1 The stationary case

Consider a sequence of target measurements $\boldsymbol{y}_{1:t} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$ with $\boldsymbol{y}_i \in \mathbb{R}^d$, and features $\boldsymbol{x}_{1:t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ with $\boldsymbol{x}_i \in \mathbb{R}^q$. Let $\mathcal{D}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ be a datapoint and $\mathcal{D}_{1:t} = (\mathcal{D}_1, \ldots, \mathcal{D}_t)$ the dataset at time t. We are interested in methods that efficiently compute $p(\boldsymbol{y}_{t+1} \,|\, \boldsymbol{x}_{t+1}, \mathcal{D}_{1:t})$ in a recursive, online way. 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 $\boldsymbol{x}_{1:t+1}$ and $\boldsymbol{y}_{1:t}$ at our disposal.[2]

Our estimate of the mean of $p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})$ is encoded in a parametric model $h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1})$ (e.g., a neural network), where $\boldsymbol{\theta}_t \in \mathbb{R}^m$ is the parameter vector. Since the parameters are unknown, we adopt a Bayesian approach and write the prediction function as

$$\mathbb{E}[h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \,|\, \mathcal{D}_{1:t}] = \int h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1})\, \lambda(\boldsymbol{\theta}_t;\, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t, \tag{1}$$

where $\lambda(\boldsymbol{\theta}_t;\, \mathcal{D}_{1:t})$ is a density over the model parameters $\boldsymbol{\theta}_t \in \mathbb{R}^m$. In what follows we take $\lambda(\boldsymbol{\theta}_t;\, \mathcal{D}_{1:t})$ to be the Bayesian posterior density (or an approximation)

$$\lambda(\boldsymbol{\theta}_t;\, \mathcal{D}_{1:t}) = p(\boldsymbol{\theta}_t \,|\, \mathcal{D}_{1:t}). \tag{2}$$

Next, we modify the above static framework to account for non-stationarity.

2.2 BONE: the non-stationary case

The class of methods described above works well when the data-generating process is well-specified. In practice, however, this is often not the case. Thus, to adapt to regime changes and other forms of non-stationarity, we introduce an auxiliary random variable $\psi_t \in \boldsymbol{\Psi}_t$, where $\boldsymbol{\Psi}_t$ is a set of possible values of the auxiliary variable ψ_t. The purpose of ψ_t is to encode information about the current regime so this can be used to predict the upcoming measurement. This can be achieved, for instance, by determining which past data points most closely align with the most recent measurements or by selecting a model —from a set of candidate models— that predicts the future observation. We describe the auxiliary variable in detail in Section 2.6.

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

$$\hat{\boldsymbol{y}}_{t+1}^{(\psi_t)} = \mathbb{E}_{\lambda_t}[h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \,|\, \psi_t] := \int h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1})\, \lambda(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t\,. \tag{3}$$

Here, we use the shorthand notation λ_t for $\lambda(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t})$, and we write $\mathbb{E}_{\lambda_t}[\cdot \,|\, \psi_t]$ and $\hat{\boldsymbol{y}}_{t+1}^{(\psi_t)}$ to draw attention to the dependence on ψ_t. Given that λ_t is the Bayesian posterior,

$$\lambda(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \propto \tau(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t), \tag{4}$$

[2]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.

where τ is the *conditional prior* density. This is a modelling choice that imposes an inductive bias on the model parameters based on the data $\mathcal{D}_{1:t-1}$ and the value ψ_t. The term $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)$ is the likelihood function, which is assumed to have the property that for any t

$$\mathbb{E}\left[\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t\right] = \int \boldsymbol{y}_t \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t) \, \mathrm{d}\boldsymbol{y}_t = h(\boldsymbol{\theta}_t; \boldsymbol{x}_t). \tag{5}$$

Finally, we introduce the function $\nu_t(\psi_t)$ that weights the importance of each $\psi_t \in \boldsymbol{\Psi}_t$ when making an estimate of $\boldsymbol{y}_{t+1}$. More precisely, we use $\nu_t(\psi_t)$ to make the estimation $\hat{\boldsymbol{y}}_{t+1}$ as follows

$$\begin{aligned} \hat{\boldsymbol{y}}_{t+1} &= \int \left(\int h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \, \lambda_t(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t}) \, \mathrm{d}\boldsymbol{\theta}_t \right) \nu_t(\psi_t) \, \mathrm{d}\psi_t \\ &=: \mathbb{E}_{\nu_t}\Big[\mathbb{E}_{\lambda_t}[h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \,|\, \psi_t] \Big]. \end{aligned} \tag{6}$$

In the Bayesian approach, we use $\nu_t(\psi_t) = p(\psi_t | \mathcal{D}_{1:t})$, but we also consider other ways of defining this measure weighting term in order to accommodate non-Bayesian approaches in the literature, as we discuss in Section 2.6.

As we show in this paper, many methods in the literature can be written as instances of (6). In particular, these methods depend on the modelling choices of: (M.1: likelihood) the model for the measurements in the form of the density $p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)$ which in turns determines h through (5); (M.2: auxvar) an auxiliary variable ψ_t that modulates the predictions when non-stationarity is present; (M.3: prior) a conditional prior density τ over model parameters conditioned on past information and the auxiliary variable ψ. Having specified (M.1 – M.3) inference of the model is given by (A.1: posterior) a method to compute or approximate λ_t in (4), and (A.2: weighting) a method to compute or approximate ν_t which weights possible values of ψ_t.

Algorithm 1 presents pseudocode for the prediction and update steps of a method within the BONE framework. Notably, these components can be broadly divided into two categories: modeling and algorithmic. The modelling components determine the inductive biases in the model, and correspond to h, ψ_t, and τ. The algorithmic components dictate how operations are carried out to produce a final prediction — this corresponds to λ_t and ν_t.

Algorithm 1 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: $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t)$ // Choice of (M.1: likelihood)
Require: $\boldsymbol{\Psi}_t$ // Choice of (M.2: auxvar)
for $\psi_t \in \boldsymbol{\Psi}_t$ **do**
  $\tau_t(\boldsymbol{\theta}_t;\, \psi_t) \leftarrow \tau(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t-1})$ // choice of (M.3: prior)
  $\lambda_t(\boldsymbol{\theta}_t;\, \psi_t) \leftarrow \lambda(\boldsymbol{\theta}_t \,|\, \psi_t, \mathcal{D}_{1:t}) \propto \tau_t(\boldsymbol{\theta}_t \; \psi_t) \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)$// choice of (A.1: posterior)
  $\nu_t(\psi_t) \leftarrow \nu(\psi_t \,|\, \mathcal{D}_{1:t})$ // choice of (A.2: weighting)
  $\hat{\boldsymbol{y}}_{t+1}^{(\psi_t)} \leftarrow \mathbb{E}_{\lambda_t}[h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) \,|\, \psi_t]$ // 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

2.3 Example tasks that can be solved with BONE

Before going into more detail about BONE, we give some concrete examples of tasks in which the BONE framework can be applied. We group these examples into unsupervised tasks and supervised tasks.

2.3.1 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 & 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. For further reference, see e.g, Aminikhanghahi & Cook (2017); Gupta et al. (2024). In this setting, non-stationarity is assumed to be abrupt and occurring at unknown points in time. We study an example in Section 4.3.1.

Filtering using state-space models (SSM) Filtering aims to estimate an underlying latent state $\boldsymbol{\theta}_t$ which evolves over time and often represents 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.

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 & 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.

2.3.2 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 in an objective manner, since we can compare the prediction to the actual observation. Supervised tasks have been the main focus of the machine learning community. 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 4.1.

Online continual learning (OCL) OCL is similar to prequential learning, in that the goal is to learn a supervised model for regression or classification online. However, the objective 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 4.1.2.

(Non-stationary) 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 $\lambda(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t})$; 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 $\sum_{t=1}^{T} \boldsymbol{y}_t^{(a_t)}$. When the mapping function

h is a neural network, this model is called a neural bandit. TS for neural bandits has been studied in many papers, see e.g., Duran-Martin et al. (2022) and references therein. Non-stationary bandits have been studied in Mellor & Shapiro (2013); Cartea et al. (2023a); Alami (2023); Liu et al. (2023). We study an example in Section 4.2.

2.4 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.

2.5 The measurement (likelihood) model (M.1)

Recall that $h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1})$ is defined by

$$h(\boldsymbol{\theta}_t;\, \boldsymbol{x}_{t+1}) = \mathbb{E}[\boldsymbol{y}_{t+1} \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_{t+1}]. \tag{7}$$

For example, for linear measurement models, the expected value is given by:

$$h(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{cases} \boldsymbol{\theta}^\intercal \boldsymbol{x} & \text{(regression)}, y \in \mathbb{R} \\ \sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}) & \text{(binary classification)}, y \in \{0, 1\} \\ \text{Softmax}(\boldsymbol{\theta}^\intercal \boldsymbol{x}) & \text{(multi-class classification)}, \boldsymbol{y} \in \{0, 1\}^C \end{cases} \tag{8}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function, $C \in \mathbb{N}$ is the number of classes, $\text{Softmax}(\boldsymbol{z})_k = \exp(\boldsymbol{z}_k)/\sum_i \exp(\boldsymbol{z}_i)$ represents the softmax function with $\boldsymbol{z} \in \mathbb{R}^d$ 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 4.

2.6 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.

<u>RL</u> (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 & MacKay (2007); Knoblauch et al. (2018); Alami et al. (2020); Agudelo-España et al. (2020); Altamirano et al. (2023); Alami (2023).

<u>RLCC</u> (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.

<u>CPT</u> (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 & Liu (2007) and has been studied in Fearnhead & Liu (2011); Fearnhead & Rigaill (2019). Under mild assumptions, it can be shown that `CPT` is equivalent to `RL`, see e.g., Knoblauch & Damoulas (2018).

<u>CPL</u> (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).

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.

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): $\psi_t = \alpha_t \in \{1, \dots, K\}$ 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).

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; this choice recovers the stationary model in (1).

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 $t = 1, \dots, T$. 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 1 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	$\{0\}$	$2^{\{0,1,\dots,t\}}$	$[0,1]$	$\{0,1\}^t$	$(0,1)^t$	$\{1,\dots,K\}$	$\{0,1,\dots,t\}$	$\{\{0,t\},\dots,\{t,0\}\}$
cardinality	1	2^t	∞	2^t	inf	K	t	$2 + t(t+1)/2$

Table 1: 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.

2.7 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

$$\tau(\boldsymbol{\theta}_t \mid \psi_t, \mathcal{D}_{1:t-1}) = \mathcal{N}\big(\boldsymbol{\theta}_t \mid g_t(\psi_t, \mathcal{D}_{1:t-1}), G_t(\psi_t, \mathcal{D}_{1:t-1})\big), \tag{9}$$

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

Below, we provide a non-exhaustive list of possible combinations of choices for (M.2: auxvar) and (M.3: prior) of the form (9) 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., $\mathbb{E}[\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}] = \mathbf{F}_t \, \boldsymbol{\theta}_{t-1} + \boldsymbol{b}_t$ and $\mathrm{Cov}[\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}] = \mathbf{Q}_t$, for given $(m \times m)$ dynamics matrix $\mathbf{F}_t$, $(m \times 1)$ bias vector $\boldsymbol{b}_t$, and $(m \times m)$ 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

$$\begin{aligned} g_t(c, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t \, \boldsymbol{\mu}_{t-1} + \boldsymbol{b}_t, \\ G_t(c, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t \, \mathbf{\Sigma}_{t-1} \mathbf{F}_t^\intercal + \mathbf{Q}_t \,. \end{aligned} \tag{10}$$

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

$$\begin{aligned} g(c, \mathcal{D}_{1:t-1}) &= \gamma\boldsymbol{\mu}_{t-1} + (1-\gamma)\boldsymbol{\mu}_0, \\ G(c, \mathcal{D}_{1:t-1}) &= \gamma^2\boldsymbol{\Sigma}_{t-1} + (1-\gamma^2)\boldsymbol{\Sigma}_0. \end{aligned} \tag{11}$$

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

$$\begin{aligned} g(\upsilon_t, \mathcal{D}_{1:t-1}) &= \upsilon_t\boldsymbol{\mu}_{t-1} + (1-\upsilon_t)\boldsymbol{\mu}_0\,, \\ G(\upsilon_t, \mathcal{D}_{1:t-1}) &= \upsilon_t^2\boldsymbol{\Sigma}_{t-1} + (1-\upsilon_t^2)\boldsymbol{\Sigma}_0\,. \end{aligned} \tag{12}$$

An example on how to compute υ_t using an empirical Bayes procedure is given in (40).

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

$$\begin{aligned} g(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{t-1}, \\ G(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{t-1} + \mathbf{Q}_t. \end{aligned} \tag{13}$$

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 & Adams, 2020), takes $g(c, \mathcal{D}_{1:t-1}) = \lambda\,\boldsymbol{\mu}_{t-1}$ and $G(c, \mathcal{D}_{1:t-1}) = \boldsymbol{\Sigma}_{t-1} + \mathbf{Q}_t$, where $0 < \lambda < 1$ 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

$$\begin{aligned} g_t(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{t-1}, \\ G_t(c, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{t-1}. \end{aligned} \tag{14}$$

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

$$\begin{aligned} g_t(\alpha_t, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t^{(\alpha_t)}\,\boldsymbol{\mu}_{t-1}^{(\alpha_t)} + \boldsymbol{b}_t^{(\alpha_t)}\,, \\ G_t(\alpha_t, \mathcal{D}_{1:t-1}) &= \mathbf{F}_t^{(\alpha_t)}\,\boldsymbol{\Sigma}_{t-1}^{(\alpha_t)}\mathbf{F}_t^{\intercal} + \mathbf{Q}_t^{(\alpha_t)}\,. \end{aligned} \tag{15}$$

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

$$\begin{aligned} g_t(r_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_0\,\mathbb{1}(r_t = 0) + \boldsymbol{\mu}_{(r_{t-1})}\mathbb{1}(r_t > 0), \\ G_t(r_t, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_0\,\mathbb{1}(r_t = 0) + \boldsymbol{\Sigma}_{(r_{t-1})}\mathbb{1}(r_t > 0), \end{aligned} \tag{16}$$

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 & 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

$$g_t(r_t, \mathcal{D}_{1:t-1}) = \begin{cases} \boldsymbol{\mu}_0 \left(1 - \nu_t(r_t)\right) + \boldsymbol{\mu}_{(r_t)} \, \nu_t(r_t) & \nu_t(r_t) > \varepsilon, \\ \boldsymbol{\mu}_0 & \nu_t(r_t) \leq \varepsilon, \end{cases} \tag{17}$$

$$G_t(r_t, \mathcal{D}_{1:t-1}) = \begin{cases} \boldsymbol{\Sigma}_0 \left(1 - \nu_t(r_t)^2\right) + \boldsymbol{\Sigma}_{(r_t)} \, \nu_t(r_t)^2 & \nu_t(r_t) > \varepsilon, \\ \boldsymbol{\Sigma}_0 & \nu_t(r_t) \leq \varepsilon. \end{cases} \tag{18}$$

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 (35). 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 4.3.1. 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 A.3.

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

$$\begin{aligned} g_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{(s_{1:t-1})}, \\ G_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\Sigma}_{(s_{1:t-1})}, \end{aligned} \tag{19}$$

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

$$\begin{aligned} g_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \boldsymbol{\mu}_{(s_{1:t-1})}, \\ G_t(s_{1:t}, \mathcal{D}_{1:t-1}) &= \begin{cases} \beta^{-1} \boldsymbol{\Sigma}_{(s_{1:t-1})} & s_t = 1\,, \\ \boldsymbol{\Sigma}_{(s_{1:t-1})} & s_t = 0\,. \end{cases} \end{aligned} \tag{20}$$

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 & 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 $\zeta_{\ell,t} = t$. For an example of MMPR with RL choice of (M.2: auxvar), see Appendix A.2.

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

This section presents algorithms for estimating the density $\lambda(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t})$; 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 (4).

There is a vast body of literature on methods for estimating the posterior over model parameters. Here, we focus on three common approaches for computing the Gaussian posterior: conjugate updates (`Cj`), linear-Gaussian approximations (`LG`), and variational Bayes (`VB`).

2.8.1 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 $\tau(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t-1})$ 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 & Smith (1994); West & 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.

2.8.2 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 assumed to be:

$$\tau_t(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) = \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\right), \tag{21}$$

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

$$\bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t) = h\left(\boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{x}_t\right) + \mathbf{H}_t\,\left(\boldsymbol{\theta}_t - \boldsymbol{\mu}_{t-1}^{(\psi_t)}\right). \tag{22}$$

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

$$\begin{aligned}
\lambda(\boldsymbol{\theta}_t; \psi_t, \mathcal{D}_{1:t}) &\propto \mathcal{N}(\boldsymbol{y}_t \,|\, \bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t), \mathbf{R}_t)\, \tau_t(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) \\
&= \mathcal{N}(\boldsymbol{y}_t \,|\, \bar{h}_t(\boldsymbol{\theta}_t, \boldsymbol{x}_t), \mathbf{R}_t)\, \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\right) \\
&\propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t}^{(\psi_t)}, \boldsymbol{\Sigma}_{t}^{(\psi_t)}),
\end{aligned} \tag{23}$$

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

$$\begin{aligned}
\hat{\boldsymbol{y}}_t^{(\psi_t)} &= h\left(\boldsymbol{\mu}_{t-1}^{(\psi_t)}, \boldsymbol{x}_t\right), \\
\mathbf{S}_t^{(\psi_t)} &= \mathbf{H}_t\, \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\, \mathbf{H}_t^{\intercal} + \mathbf{R}_t, \\
\mathbf{K}_t^{(\psi_t)} &= \boldsymbol{\Sigma}_{t-1}^{(\psi_t)}\, \mathbf{H}_t^{\intercal}\, \left(\mathbf{S}_t^{(\psi_t)}\right)^{-1}, \\
\boldsymbol{\mu}_t^{(\psi_t)} &= \boldsymbol{\mu}_{t-1}^{(\psi_t)} + \mathbf{K}_t^{(\psi_t)}\left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t^{(\psi_t)}\right), \\
\boldsymbol{\Sigma}_t^{(\psi_t)} &= \boldsymbol{\Sigma}_{t-1}^{(\psi_t)} - \left(\mathbf{K}_t^{(\psi_t)}\right)\left(\mathbf{S}_t^{(\psi_t)}\right)\left(\mathbf{K}_t^{(\psi_t)}\right)^{\intercal}.
\end{aligned} \tag{24}$$

This linear approximation enables efficient computation of the posterior in a Gaussian form. Examples include the extended Kalman filter (EKF) (Haykin, 2004), which applies local linearisation to non-linear systems,

the exponential family EKF (Ollivier, 2018), which approximates the measurement model as Gaussian by matching the first two moments, and the low-rank Kalman filter (LoFi) method (Chang et al., 2023), which assumes a diagonal-plus-low-rank (DLR) posterior precision matrix. See Särkkä & Svensson (2023) for more details on such Gaussian filtering methods.

2.8.3 Variational Bayes (VB)

Variational Bayes (VB) is a popular method for approximating a posterior distribution of model parameters by choosing a parametric family (such as Gaussians) that is computationally tractable. The primary objective of VB is to minimise the Kullback-Leibler (KL) divergence between a candidate Gaussian distribution and the density λ_t. It can be shown that we can safely ignore the normalisation constant for λ_t, which is often computationally expensive, so we can replace λ_t with its unnormalised form. We have the following optimisation problem for the posterior variational parameters:

$$(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathbf{D}_{\mathrm{KL}} \left(\mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,\|\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, \tau_t(\boldsymbol{\theta}_t;\, \psi_t) \right), \tag{25}$$

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

An example of VB for neural network models is 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; the low-rank RVGA (L-RVGA) method of Lambert et al. (2023), which uses a diagonal plus low-rank (DLR) Gaussian variational approximation; the Bayesian online natural gradien (BONG) method of Jones et al. (2024), which combines the DLR approximation with EKF-style linearisation for additional speedups; 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); and the projection-based unification of last-layer and subspace estimation (PULSE) method of Cartea et al. (2023b), which targets different posterior densities for a subspace of the hidden layers and a full-rank covariance over the final layer of a neural network.

2.8.4 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 (25) 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.

2.9 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 & MacKay (2007); Fearnhead & 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

$$\begin{aligned} \nu_t(\psi_t) &= p(\psi_t \,|\, \mathcal{D}_{1:t}) \\ &= p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \psi_t, \mathcal{D}_{1:t-1}) \int_{\psi_{t-1} \in \boldsymbol{\Psi}_{t-1}} p(\psi_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(\psi_t \,|\, \psi_{t-1}, \mathcal{D}_{1:t-1}) \mathrm{d}\psi_{t-1}, \end{aligned} \tag{26}$$

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 (26) is known as the conditional posterior predictive, and is given by

$$p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \psi_t, \mathcal{D}_{1:t-1}) = \int p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, \tau(\boldsymbol{\theta}_t;\, \psi_t, \mathcal{D}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_t. \tag{27}$$

This integral over $\boldsymbol{\theta}_t$ may require approximations, as we discussed in Section 2.8. Furthermore, the integral over ψ_{t-1} in (26) 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.

2.9.1 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 & MacKay (2007) takes

$$p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} 1 - H(r_{t-1}) & \text{if } r_t = r_{t-1} + 1, \\ H(r_{t-1}) & \text{if } r_t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

where $H : \mathbb{N}_0 \to (0, 1)$ is the hazard function. A popular choice is to take $H(r) = \pi \in (0, 1)$ 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

$$p(\tilde{s}_{1:t} \,|\, s_{1:t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} \pi & \text{if } ([\tilde{s}_{1:t} \setminus \tilde{s}_t] = s_{1:t-1}) \text{ and } \tilde{s}_t = 1, \\ 1 - \pi & \text{if } ([\tilde{s}_{1:t} \setminus \tilde{s}_t] = s_{1:t-1}) \text{ and } \tilde{s}_t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{29}$$

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 A.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 & Liu (2007) takes

$$p(\boldsymbol{\zeta}_t \,|\, \boldsymbol{\zeta}_{t-1}, \mathcal{D}_{1:t-1}) = p(\zeta_{\ell,t} \,|\, \zeta_{\ell,t-1}) = J(\zeta_{\ell,t} - \zeta_{\ell,t-1}), \tag{30}$$

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

$$\hat{\nu}_t(\psi_t) = \frac{\nu_t(\psi_t)}{\sum_{\psi'_t \in \mathcal{A}_t} \nu_t(\psi'_t)}, \tag{31}$$

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). However, other styles of pruning are also possible; see e.g., Li et al. (2021).

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

$$\nu_t(\boldsymbol{w}_t)_k = \frac{\exp(\boldsymbol{w}_{t,k}^\intercal \boldsymbol{y}_t)}{\sum_{j=1}^K \exp(\boldsymbol{w}_{t,j}^\intercal \boldsymbol{y}_t)}, \tag{32}$$

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

$$\mathcal{G}_t(\boldsymbol{w}_t) = p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) = \sum_{k=1}^K p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \boldsymbol{\alpha}_{t,k}, \mathcal{D}_{1:t-1})\, \nu_t(\boldsymbol{w}_t)_k, \tag{33}$$

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

$$p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1}) = \begin{cases} 1-\pi & \text{if } r_t = r_{t-1}+1, \\ \pi & \text{if } r_t = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

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

$$\nu_t(r_t^{(1)}) = \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\pi)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,\pi + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\pi)}. \tag{35}$$

For details on the computation of (A.2: weighting), see Appendix A.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 4 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,

$$p(s_{1:t} \,|\, s_{1:t-1}, \mathcal{D}_{1:t-1}) = p(s_{1:t} \,|\, \mathcal{D}_{1:t-1}), \tag{36}$$

so that

$$p(s_{1:t} \,|\, \mathcal{D}_{1:t}) \propto p(s_{1:t} \,|\, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, s_{1:t}, \mathcal{D}_{1:t-1}). \tag{37}$$

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 $\nu_t(s_{1:t}) = p(s_{1:t} \,|\, \mathcal{D}_{1:t})$. Then, the hypothesis with highest probability is stored and gets assigned a weight of one.

2.9.2 Continuous auxiliary variable (CA)

Here, we briefly discuss continuous auxiliary variables (CA). For some choices of ψ_t and transition densities $p(\psi_t \,|\, \psi_{t-1}, \mathcal{D}_{1:t-1})$, computation of (26) 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

$$p(\upsilon_t \,|\, \upsilon_{t-1}, \mathcal{D}_{1:t-1}) = p(\upsilon_t), \tag{38}$$

so that

$$p(\upsilon_t \,|\, \mathcal{D}_{1:t}) \propto p(\upsilon_t)\, p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \upsilon_t). \tag{39}$$

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:

$$\upsilon_t^* = \arg\max_{\upsilon \in [0,1]} p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \upsilon, \mathcal{D}_{1:t-1}). \tag{40}$$

In practice, (40) 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) $\boldsymbol{\upsilon}_t \in (0,1)^m$.

3 Unified view of examples in the literature

Table 2 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 2.3. We will experimentally compare a subset of these methods in Section 4.

The methods presented in Table 2 can be directly applied to tackle any of the the problems mention in Section 2.3. However, as choice of (M.1: likelihood), we specify the task under which the configuration was introduced.

4 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 for the methods we compare.

4.1 Prequential prediction

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

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 & Athans (1978)	filtering	ME	LSSM	LG	CA
Chaer et al. (1997)	filtering	ME	LSSM	LG	DA[K]
Ghahramani & Hinton (2000)	SSSM	ME	Static	VB	CA
Adams & MacKay (2007)	seg.	RL	PR	Cj	DA[inf]
Fearnhead & Liu (2007)	seg. & preq.	CPT/ME	PR	Any	DA[inf]
Wilson et al. (2010)	seg.	RLCC	PR	Cj	DA[inf]
Fearnhead & Liu (2011)	seg.	CPT/ME	MMPR	Any	DA[inf]
Mellor & Shapiro (2013)	bandits	RL	PR	Cj	DA[inf]
Nguyen et al. (2017)	OCL	CPL	Sub	VB	DA[1]
Knoblauch & 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]
Titsias et al. (2024)	OCL	CPP	OU	LG	CA
Abélès et al. (2024)	preq.	ME	LSSM	LG	DA[K]
RL[1]-OUPR* (ours)	any	RL	SPR	Any	DA[1]

Table 2: List of methods ordered by publication date. The tasks are discussed in Section 2.3. We use the following abbreviations: SSSM means switching state space model; OCL 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 (6) —one for each choice of auxiliary variable.

4.1.1 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.

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

M.2-M.3	Eq.	A.2	Description	Sections
			static	
`C-Static`	(14)	-	This corresponds to the static case with a classical Bayesian update. This method does not assume changes in the environment.	4.3.1, 4.3.2
			abrupt changes	
`RL-PR`	(16)	`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 A.1 for more details.	4.1.1, 4.1.2, 4.2, 4.3.1, 4.3.2,
`WoLF+RL-PR*`	(16)	`DA[inf]`	Special case of `RL-PR` with explicit choice of (A.1: posterior) which makes it robust to outliers.	4.3.2
			gradual changes	
`CPP-OU`	(12)	`CA`	Updates are done using a discounted mean and covariance according to the probability estimate that a change has occurred.	4.1.1, 4.1.2, 4.2
`C-ACI`	(13)	-	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 $\mathbf{Q}_t$.	4.1.1, 4.1.2, 4.2,
			abrupt & gradual changes	
`RL-MMPR`	(55)	`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 A.2 for more details.	4.3.1
`RL-OUPR`	(17)	`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 A.3 for more details.	4.1.1, 4.1.2, 4.2, 4.3.1

Table 3: 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.

For this experiment, we consider `RL[1]-OUPR*`, `RL[1]-PR`, `C-ACI`, and `CPP-OU`. For computational convenience, we plug in a point-estimate (MAP estimate) of the neural network parameters when making predictions in (6). 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 1 we show the evolution of the target variable $\boldsymbol{y}_t$ between March 3 2020 and March 10 2020. The bottom panel of Figure 1 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`.

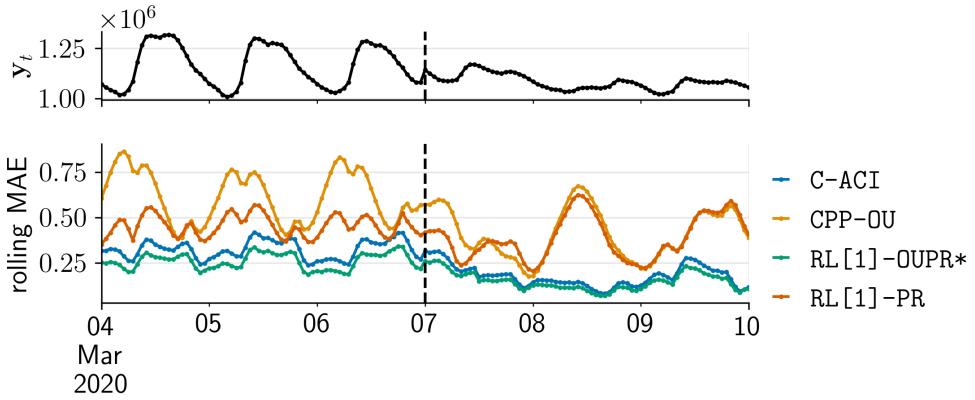

Figure 1: The **top panel** shows the target variable (electricity consumption) from March 1 2020 to March 12 2020. The **bottom panel** shows the twelve-hour rolling relative absolute error of predictions for the same time window. The dotted black line corresponds to March 7 2020, when Covid lockdown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

Next, Figure 2, 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.

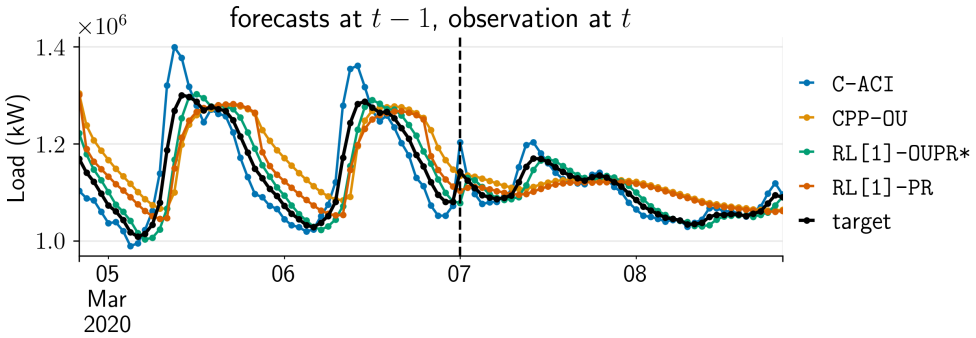

Figure 2: One day ahead electricity forecasting results for Figure 1. The dotted black line corresponds to March 7 2020.

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 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.

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 4, 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.

4.1.2 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 $i \in \{1, 2\}$, $\boldsymbol{x}_t = (x_{t,1}, x_{t,2}) \in \mathbb{R}^2$, $y_t \sim \text{Bernoulli}(\sigma(\boldsymbol{\theta}_t^\intercal \boldsymbol{x}_t))$ with $\boldsymbol{\theta}_t^{(1)} = 10 \sin(5^\circ\, t)$ and $\boldsymbol{\theta}_t^{(2)} = 10 \cos(5^\circ\, t)$.

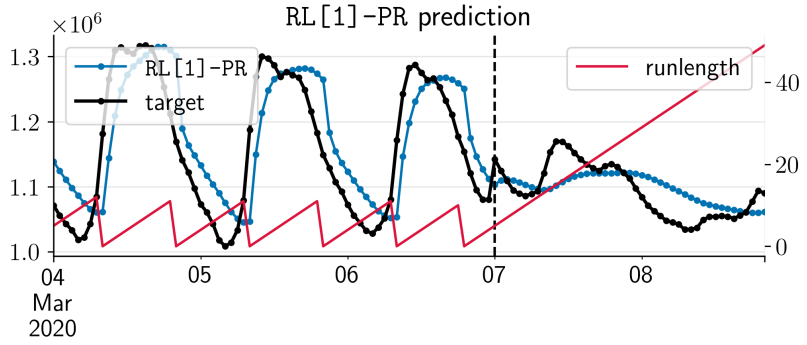

Figure 3: One day ahead electricity forecasting results for `RL[1]-PR` together with the target variable on the left y-axis, and the value for runlength (`RL`) on the right y-axis. We see that after the 7 March changepoint, the runlength monotonically increases, indicating a stationary regime.

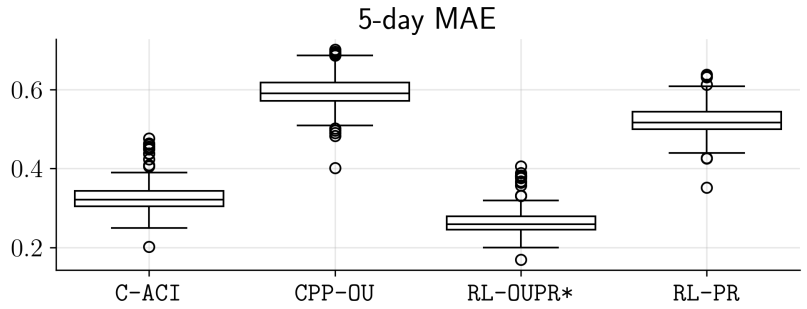

Figure 4: Distribution of the 5-day mean absolute error (MAE) for each of the competing methods on electricity forecasting over the entire period. For this calculation we split the dataset into consecutive buckets containing five days of data each, and for a given bucket we compute the average absolute error of the predictions and observations that fall within the bucket.

Thus the unknown values of model parameters are slowly drifting deterministically according to sine and cosine functions. The timesteps go from 0 to 720.

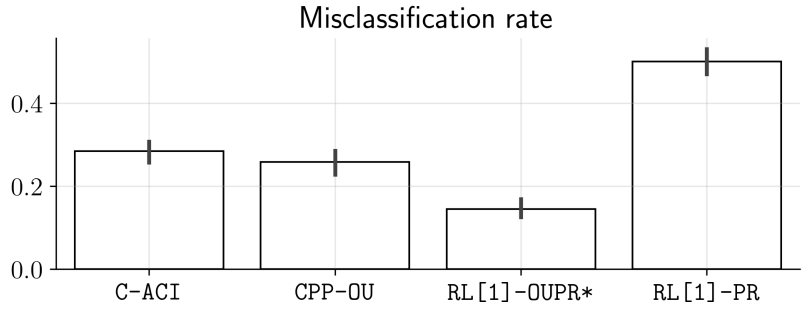

Figure 5: Misclassification rate of various methods on the online classification with periodic drift task.

Figure 5 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 π, as shown in Figure 6. However, even then the performance of this method still does not match our method.

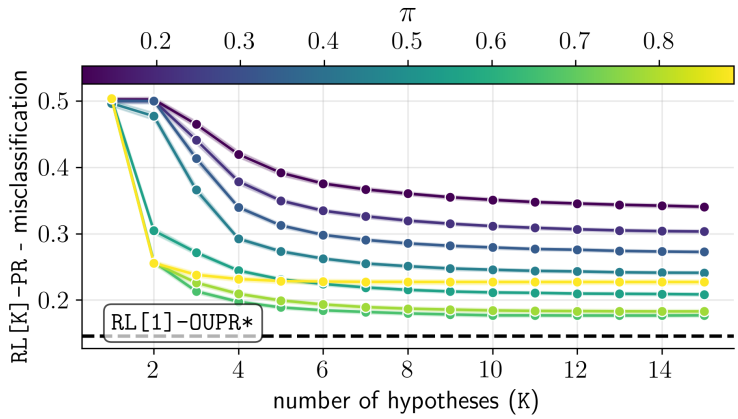

Figure 6: Accuracy of predictions for `RL[1]-PR` as a function of the number of hypothesis and the prior probability of a changepoint π. The black dotted line is the performance of `RL[1]-OUPR*` reported in Figure 5.

4.1.3 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

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_{t-1} + \boldsymbol{\epsilon}_t & \text{w.p. } 1 - p_\epsilon, \\ \mathcal{U}[-2, 2]^2 & \text{w.p. } p_\epsilon, \end{cases} \tag{41}$$

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).

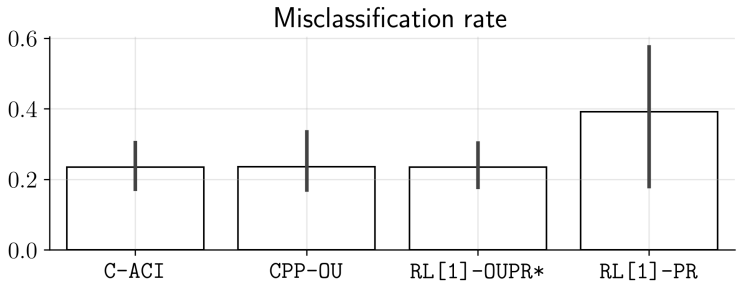

Figure 7: Misclassification rate of various methods on the online classification with drift and jumps task.

Figure 7 shows the missclasification 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.

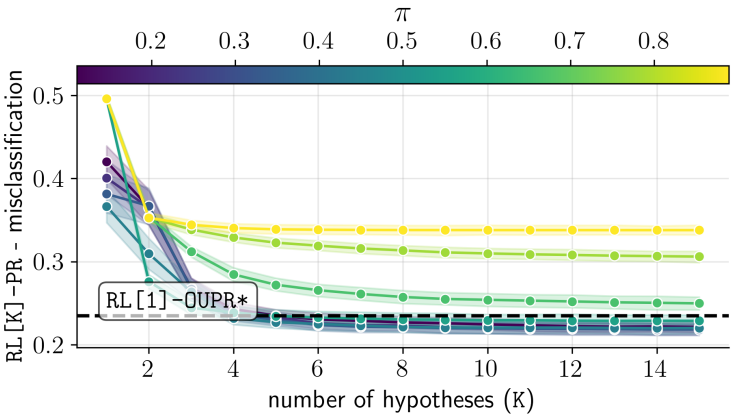

Figure 8: Accuracy of predictions for `RL[K]-PR` as a function of the number of hypotheses (`K`) and the probability of a changepoint π. The black dotted line is the performance of `RL[1]-OUPR*` reported in Figure 7.

To explain this behaviour, Figure 8 shows the performance of `RL[K]-PR` as a function of number of hypotheses and prior probability of a changepoint π. 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.

4.2 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 & 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 & 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 9 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 4.1.2. This is because both problems have a similar drift structure.

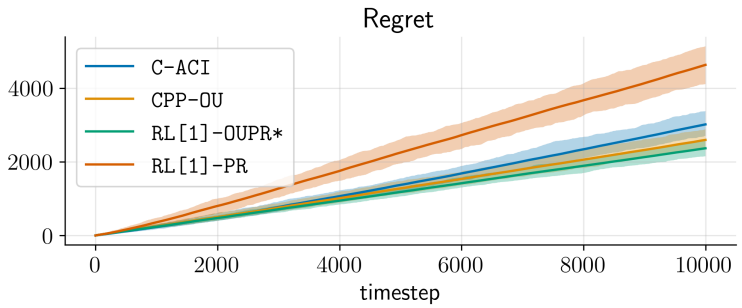

Figure 9: Regret of competing methods on the contextual bandits task. Confidence bands are computed with one hundred simulations.

4.3 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.

4.3.1 Autoregression with dependence across the segments

In this experiment, we consider the synthetic autoregressive dataset introduced in Section 2 of Fearnhead & 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 10.

We compare the performance of the three methods in the previous subsection. For this experiment, we employ a probability of a changepoint $\pi = 0.01$. 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., the conditional expectation (5) can be written as $h(\boldsymbol{\theta}_t; \psi_t, \boldsymbol{x}_t)$. For this experiment, we take (M.2: auxvar) to be `RL` and our choice of (M.1: likelihood) becomes

$$h(\boldsymbol{\theta}_t; r_t, \boldsymbol{x}_{1:t}) = \boldsymbol{\theta}_t^\intercal \, \boldsymbol{h}(\boldsymbol{x}_{1:t}, r_t), \tag{42}$$

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 (42), 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) which was designed for segmentation with dependence.

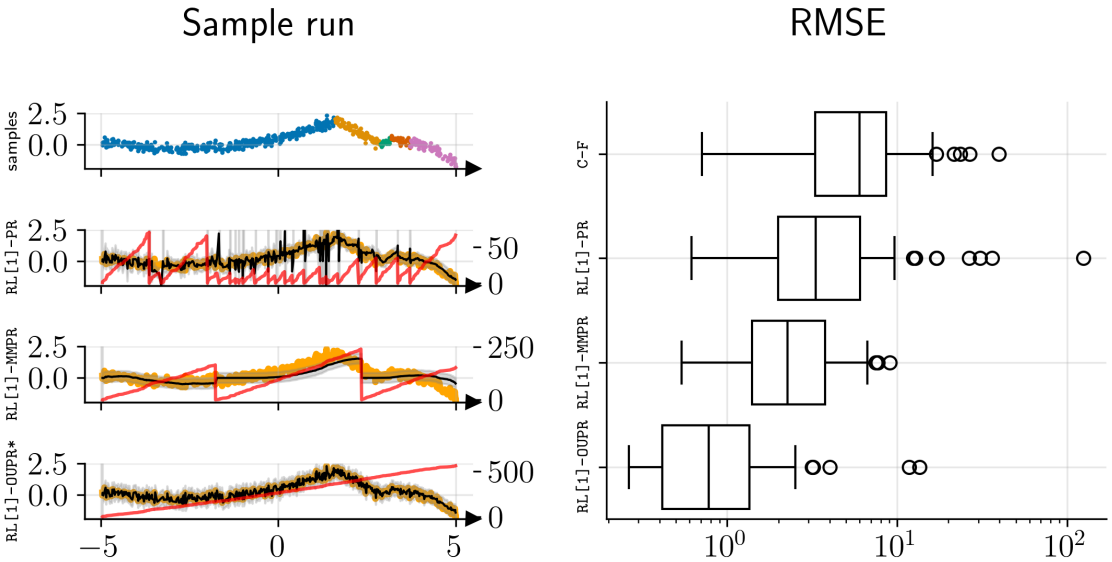

Figure 10: The **left panel** shows a sample run of the piecewise polynomial regression with dependence across segments. The x-axis is for the features, the (left) y-axis is for measurements together with the estimations made by `RL[1]-PR`, `RL-MMPR`, and `RL[1]-OUPR*`, the (right) y-axis is for the 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.

Figure 10 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 does not adjust beliefs as quickly as it should. Our `RL[1]-OUPR*` method strikes a good compromise.

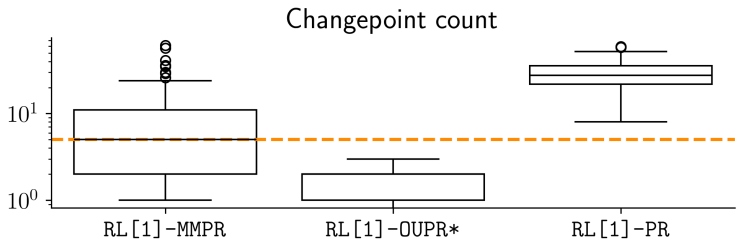

Figure 11: Count of changepoints over an experiment for 100 trials. The orange line shows the true number of changepoints for all trials.

Figure 11 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 10 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 & Williams (2020).

4.3.2 Non-stationary heavy-tailed regression with `DA[inf]`

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 & Rigaill, 2019; Altamirano et al., 2023; Sellier & Dellaportas, 2023) and for filtering (Reimann, 2024). In what follows, we show how we can easily accomodate 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

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_{t-1} & \text{w.p. } 1 - p_\epsilon, \\ \mathcal{U}[-3, 3]^3 & \text{w.p. } p_\epsilon, \end{cases} \tag{43}$$

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 12 shows a sample data generated by this process.

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

The left panel in Figure 13 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 13 shows the distribution of the RMSE for all methods after 30 trials.

The left panel of Figure 13 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

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

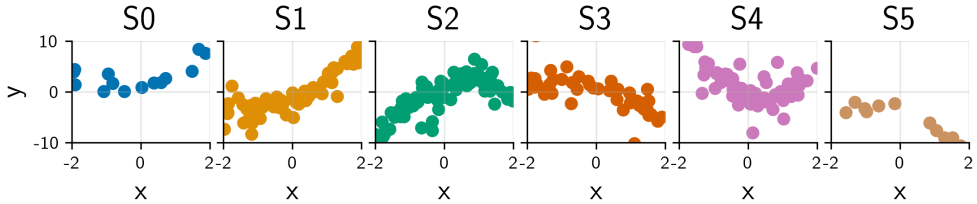

Figure 12: Sample run of the heavy-tailed-regression process. Each box corresponds to the samples within a segment.

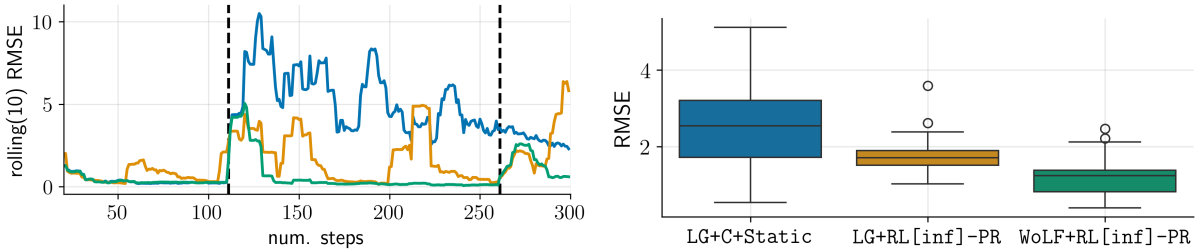

Figure 13: The **left panel** shows the rolling RMSE using a window of the 10 previous observations. The **right panel** shows the distribution of final RMSE over 30 runs. The vertical dotted line denotes a change in the true model parameters.

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.

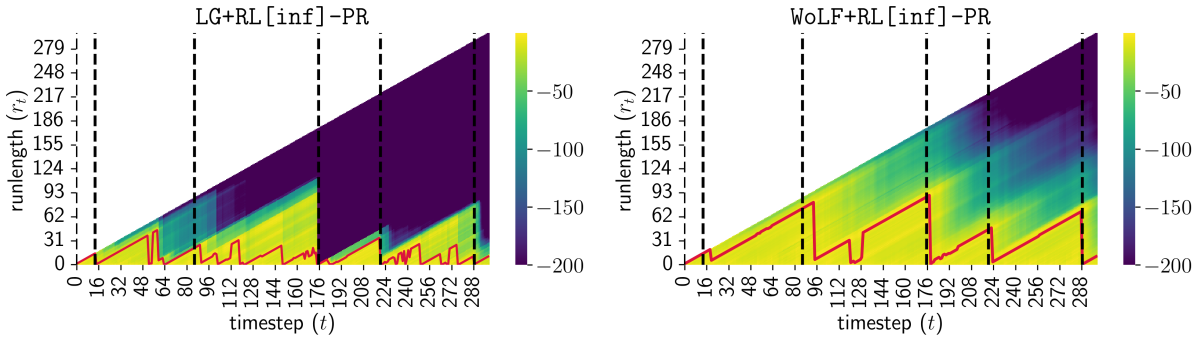

Figure 14: Segmentation of the non-stationary linear regression problem. The left panel shows the segmentation done by `LG+RL[inf]-PR`. The right panel shows the segmentation done by `WoLF+RL[inf]-PR*`. The x-axis is the timestep t, the y-axis is the runlength r_t (note that it is always the case that $r_t \leq t$), and the color bar shows the value $\log p(r_t \,|\, \boldsymbol{y}_{1:t})$. The red line in either plot is the trajectory of the mode, i.e., the set $r^*_{1:t} = \{\arg\max_{r_1} p(r_1 \,|\, \mathcal{D}_1), \ldots, \arg\max_{r_t} p(r_t \,|\, \mathcal{D}_{1:t})\}$. Note that the non-robust method (left) oversegments the signal.

Figure 14 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.

5 Conclusions

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.

References

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.

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*, pp. 320–329. PMLR, 2020.

Reda Alami. Bayesian change-point detection for bandit feedback in non-stationary environments. In *Asian Conference on Machine Learning*, pp. 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*, pp. 211–221. PMLR, 2020.

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and scalable bayesian online changepoint detection, 2023.

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.

Jordan T Ash and Ryan P Adams. On warm-starting neural network training. In *NIPS*, 2020. URL http://arxiv.org/abs/1910.08475.

Daniel Barry and John A Hartigan. Product partition models for change point problems. *The Annals of Statistics*, pp. 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, 2023.

J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley, 1994.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

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/google/jax.

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*, pp. 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, Fayçal Drissi, and Pierre Osselin. Bandits for algorithmic trading with signals. *Available at SSRN 4484004*, 2023a.

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*, pp. 1025–1071. PMLR, 2023.

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.

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.

Gerardo Duran-Martin, Aleyna Kara, and Kevin Murphy. Efficient online bayesian inference for neural bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 6002–6021. PMLR, 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.

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.

Arnold M Faden and Gordon C Rausser. Econometric policy model construction: the post-bayesian approach. In *Annals of Economic and Social Measurement, Volume 5, number 3*, pp. 349–363. NBER, 1976.

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.

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.

Muktesh Gupta, Rajesh Wadhvani, and Akhtar Rasool. Comprehensive analysis of change-point dynamics detection in time series data: A review. *Expert Systems with Applications*, pp. 123342, 2024.

Simon Haykin. *Kalman filtering and neural networks*. John Wiley & Sons, 2004.

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*, pp. 703–711. PMLR, 2021.

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.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal bayesian on-line changepoint detection with model selection. In *International Conference on Machine Learning*, pp. 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 (eds.), *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.

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.

Aodong Li, Alex Boyd, Padhraic Smyth, and Stephan Mandt. Detecting and adapting to irregular distribution shifts in bayesian online learning, 2021.

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*, pp. 661–670, 2010.

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*, pp. 914–922. PMLR, 2017.

Bin Liu. Robust sequential online prediction with dynamic ensemble of multiple models: A review. *Neurocomputing*, pp. 126553, 2023.

Yueyang Liu, Benjamin Van Roy, and Kuang Xu. Nonstationary bandit learning via predictive sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 6215–6244. PMLR, 2023.

David Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, 1965.

Joseph Mellor and Jonathan Shapiro. Thompson sampling in switching environments with bayesian online change point detection. *arXiv preprint arXiv:1302.3721*, 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.

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. Online natural gradient as a Kalman filter. *Electronic Journal of Statistics*, 12(2):2930 – 2961, 2018. doi: 10.1214/18-EJS1468. URL https://doi.org/10.1214/18-EJS1468.

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*, pp. 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.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling, 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.

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)*, pp. 927–934, 2010.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Test-time adaptation with state-space models. *arXiv preprint arXiv:2407.12492*, 2024.

Jeremy Sellier and Petros Dellaportas. Bayesian online change point detection with hilbert space approximate student-t process. In *International Conference on Machine Learning*, pp. 30553–30569. PMLR, 2023.

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.

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.

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*, pp. 1088–1095, 2008.

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.

A Worked examples for BONE methods

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

$$p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\theta}_t, \mathbf{R}_t). \tag{44}$$

A.1 Runlength with prior reset (`RL-PR`)

A.1.1 Unbounded number of hypotheses `RL[inf]-PR`

The work in Adams & 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 (28). We consider $\nu_t(r_t) = p(r_t|\mathcal{D}_{1:t})$ such that

$$p(r_t \,|\, \mathcal{D}_{1:t}) = \frac{p(r_t, \mathcal{D}_{1:t})}{\sum_{\hat{r}_t=0}^{t} p(\hat{r}_t, \mathcal{D}_{1:t})}, \tag{45}$$

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

$$\begin{aligned}
&p(r_t, \mathcal{D}_{1:t}) \\
&= \sum_{r_{t-1}=0}^{t-1} p(r_t, r_{t-1}, \mathcal{D}_{1:t-1}, \mathcal{D}_t) \\
&= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}, \mathcal{D}_{1:t-1})\, p(\boldsymbol{y}_t \,|\, r_t, r_{t-1}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \\
&= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1}) p(r_t \,|\, r_{t-1}).
\end{aligned} \tag{46}$$

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 (46), 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

$$\begin{aligned} p(r_t \,|\, \mathcal{D}_{1:t}) &= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) && \text{if } r_t \geq 1 \\ p(r_t \,|\, \mathcal{D}_{1:t}) &= p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1}=0}^{t-1} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) && \text{if } r_t = 0\,. \end{aligned} \tag{47}$$

The joint density (47) 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 (47) 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

$$p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) = \int p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}_t, \boldsymbol{x}_t)\, p(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_t. \tag{48}$$

For a choice of (M.1: likelihood) given by (44) and a choice of (M.3: prior) given by (16), the posterior predictive (48) takes the form.

$$\begin{aligned} p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) &= \int \mathcal{N}\left(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\theta}_t, \mathbf{R}_t\right) \mathcal{N}\left(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(r_t)}, \boldsymbol{\Sigma}_{t-1}^{(r_t)}\right) \mathrm{d}\boldsymbol{\theta}_t \\ &= \mathcal{N}\left(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \boldsymbol{\mu}_{t-1}^{(r_t)},\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\Sigma}_{t-1}^{(r_t)}\, \boldsymbol{x}_t + \mathbf{R}_t\right), \end{aligned} \tag{49}$$

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 $(\boldsymbol{\mu}_{t-1}^{(r_t)}, \boldsymbol{\Sigma}_{t-1}^{(r_t)}) = (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.

A.1.2 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, (46) takes the form

$$p(r_t, \mathcal{D}_{1:t}) = p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \qquad \text{if } r_t \geq 1, \tag{50}$$

$$p(r_t, \mathcal{D}_{1:t}) = p(\boldsymbol{y}_t \,|\, r_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \sum_{r_{t-1} \in \boldsymbol{\Psi}_{t-1}} p(r_{t-1}, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \qquad \text{if } r_t = 0\,. \tag{51}$$

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

A.2 Runlength with moment-matched prior reset (`RL-MMPR`)

Here, we consider a modified version of the method introduced in Fearnhead & Liu (2011). We consider the choice of `RL` and adjust the choice of (M.3: prior) for `RL-PR` introduced in Appendix A.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

$$\begin{aligned} p(\boldsymbol{\theta}_t \,|\, r_t, \mathcal{D}_{1:t-1}) &= \sum_{r_{t-1}=0}^{t-1} p(\boldsymbol{\theta}_t, r_{t-1} \,|\, r_t, \mathcal{D}_{1:t-1}) \\ &= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, p(\boldsymbol{\theta}_t \,|\, r_t, r_{t-1}, \boldsymbol{y}_{1:t-1}) \\ &= \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})}, \boldsymbol{\Sigma}_{t-1}^{(r_{t-1})}). \end{aligned} \tag{52}$$

Because (52) is a mixture model, we choose a conditional prior to be Gaussian that approximates the first two moments. We obtain

$$\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}] = \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1})\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})} \tag{53}$$

for the first moment, and

$$\mathbb{E}[\boldsymbol{\theta}_t\, \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] \sum_{r_{t-1}=0}^{t-1} p(r_{t-1} \,|\, \mathcal{D}_{1:t-1})\, p(r_t \,|\, r_{t-1}) \left(\boldsymbol{\Sigma}_{t-1}^{(r_{t-1})} + \boldsymbol{\mu}_{t-1}^{(r_{t-1})}\, \boldsymbol{\mu}_{t-1}^{(r_{t-1})\intercal} \right) \tag{54}$$

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

$$\begin{aligned} \boldsymbol{\mu}_t^{(0)} &= \mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}], \\ \boldsymbol{\Sigma}_t^{(0)} &= \mathbb{E}[\boldsymbol{\theta}_t\, \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] - (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])\, (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])^\intercal\,. \end{aligned} \tag{55}$$

Algorithm 3 shows an update step under this process when we maintain a set of K possible hypotheses.

A.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 A.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:

$$\begin{aligned} \nu_t(r_t^{(1)}) &= p(r_t^{(1)} \,|\, \mathcal{D}_{1:t}) \\ &= \frac{p(r_t^{(1)}, \mathcal{D}_{1:t})}{p(r_t^{(1)}, \mathcal{D}_{1:t}) + p(r_t^{(0)}, \mathcal{D}_{1:t})} \\ &= \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, (1-\pi)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, \pi + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, p(r_{t-1}, \mathcal{D}_{1:t-1})\, (1-\pi)} \\ &= \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, (1-\pi)}{p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, \pi + p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\, (1-\pi)}. \end{aligned} \tag{56}$$

where $\pi = p(r_t \,|\, r_{t-1})$ with $r_t = 0$ is the prior probability of a changepoint and and $1 - \pi = p(r_t \,|\, r_{t-1}$ 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 (17) and (18). Because we maintain a single hypothesis, the weight at the end of the update step is set to 1. Algorithm 4 shows an update step for `RL[1]-OUPR*` under the choice of (M.1: likelihood) given by (44).

A.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

$$\begin{aligned} p(s_{1:t} \,|\, \mathcal{D}_{1:t}) &= p(s_t, s_{1:t-1} \,|\, \boldsymbol{y}_t, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \\ &= p(s_{1:t-1} \,|\, \mathcal{D}_{1:t-1}) p(s_t \,|\, s_{1:t-1}, \boldsymbol{x}_t, \boldsymbol{y}_t, \mathcal{D}_{1:t-1}). \end{aligned} \tag{57}$$

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

$$\begin{aligned} &p(s_t = 1 \,|\, s_{1:t-1}, \mathcal{D}_{1:t}) \\ &\quad = \frac{p(s_t = 1) p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, s_{1:t-1}, s_t = 1, \mathcal{D}_{1:t-1})}{p(s_t = 1) p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \boldsymbol{y}_{1:t-1}) + p(s_t = 0) p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})} \\ &\quad = \left(1 + \exp\left(-\log\left(\frac{p(s_t = 1) p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \boldsymbol{y}_{1:t-1})}{p(s_t = 0) p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}\right)\right)\right)^{-1} = \sigma(m_t), \end{aligned} \tag{58}$$

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

$$m_t = \log\left(\frac{p(\boldsymbol{y}_t \,|\, s_t = 1, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}{p(\boldsymbol{y}_t \,|\, s_t = 0, \boldsymbol{x}_t, s_{1:t-1}, \mathcal{D}_{1:t-1})}\right) + \log\left(\frac{p(s_t = 1)}{p(s_t = 0)}\right), \tag{59}$$

and similarly,

$$p(s_t = 0 \,|\, s_{1:t-1}, \mathcal{D}_{1:t}) = 1 - \sigma(m_t). \tag{60}$$

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

B Algorithms

Algorithm 2 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: $\{r_{t-1}^{(k)}\}_{k=1}^{K} \in \{0, \dots, t-1\}^K$ // 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 $k = 1, \dots, K$ **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)
  $\lambda_t(\boldsymbol{\theta}_t;\, r_t^{(k)}) \propto \tau_t(\boldsymbol{\theta}_t; r_t^{(k)})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})$ // following (24)
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
$I_{1:k} = \text{top.k}(\{p(r_t^{(1)},\, \mathcal{D}_{1:t}), \dots, p(r_t^{(k+1)}, \mathcal{D}_{1:t})\},\, K)$
$\{p(r_t^{(k)}, \mathcal{D}_{1:t})\}_{k=1}^{K} \leftarrow \text{slice.at}(\{p(r_t^{(k)},\, \mathcal{D}_{1:t})\}_{k=1}^{K+1},\, I_{1:K})$
$\{(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^{K} \leftarrow \text{slice.at}(\{(\boldsymbol{\mu}_t^{(k)},\, \boldsymbol{\Sigma}_t^{(k)})\}_{k=1}^{K+1},\, I_{1:K})$
// 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 $k = 1, \dots, K$
$\hat{\boldsymbol{y}}_{t+1} \leftarrow \boldsymbol{x}_{t+1}^\intercal \left(\sum_{k=1}^{K} \nu_t(r_t^{(k)}) \boldsymbol{\mu}_t^{(k)}\right)$ // 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 2, the function top.k(A, K) returns the indices of the top $K \geq 1$ elements of A with highest value. The function 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 3 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)
  $\lambda_t(\boldsymbol{\theta}_t;\, r_t^{(k)}) \propto \tau_t(\boldsymbol{\theta}_t; r_t^{(k)})\, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})$ // following (24)
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 (53)
$\boldsymbol{\Sigma}_0 \leftarrow \mathbb{E}[\boldsymbol{\theta}_t\, \boldsymbol{\theta}_t^\intercal \,|\, r_t, \boldsymbol{y}_{1:t-1}] - (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])\, (\mathbb{E}[\boldsymbol{\theta}_t \,|\, r_t, \boldsymbol{y}_{1:t-1}])^\intercal$ // following (53) and (54)
$p(\boldsymbol{y}_t \,|\, r_t^{(k+1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\mu}_0,\, \boldsymbol{x}_t^\intercal\, \boldsymbol{\Sigma}_0\, \boldsymbol{x}_t + \mathbf{R}_t)$ // 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 4 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: $p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}, \mathbf{R}_t)$ // Choice of (M.1: likelihood)

$(r_t^{(0)}, r_t^{(1)}) \leftarrow (0, r_{t-1}+1)$ // choice of (M.2: auxvar)
$p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_0, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_0 \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive at changepoint
$p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1}) \leftarrow \mathcal{N}(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^\intercal \, \boldsymbol{\mu}_{t-1}, \, \boldsymbol{x}_t^\intercal \, \boldsymbol{\Sigma}_{t-1} \, \boldsymbol{x}_t + \mathbf{R}_t)$ // posterior predictive if no changepoint
$\nu_t(r^{(1)}) \leftarrow \frac{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})(1-\pi)}{p(\boldsymbol{y}_t \,|\, r_t^{(1)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,(1-\pi) + p(\boldsymbol{y}_t \,|\, r_t^{(0)}, \boldsymbol{x}_t, \mathcal{D}_{1:t-1})\,\pi}$ // 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
$\tau_t(\boldsymbol{\theta}_t; r_t) \leftarrow \mathcal{N}(\boldsymbol{\theta}_t \,|\, \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$ // choice of (M.3: prior)
$\lambda_t(\boldsymbol{\theta}_t; \, r_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t) \, p(\boldsymbol{y}_t \,|\, \boldsymbol{\theta}^\intercal \boldsymbol{x}_t, \mathbf{R}_t) \propto \mathcal{N}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ // choice of (A.1: posterior)— via (24)
$\hat{\boldsymbol{y}}_{t+1} \leftarrow \boldsymbol{x}_{t+1}^\intercal \, \boldsymbol{\mu}_t$ // prequential prediction (given linear-Gaussian model)
return $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, r_t)$, $\hat{\boldsymbol{y}}_{t+1}$