Regime Learning for Differentiable Particle Filters

John-Joseph Brady
Computer Science Research Centre
University of Surrey
Guildford, United Kingdom
j.brady@surrey.ac.uk

Yuhui Luo
Data Science Department
National Physical Laboratory
Teddington, United Kingdom
yuhui.luo@npl.co.uk

Wenwu Wang
Centre for Vision, Speech, and Signal Processing
University of Surrey
Guildford, United Kingdom
w.wang@surrey.ac.uk

Víctor Elvira
School of Mathematics
University of Edinburgh
Edinburgh, United Kingdom
victor.elvira@ed.ac.uk

Yunpeng Li
Computer Science Research Centre
University of Surrey
Guildford, United Kingdom
yunpeng.li@surrey.ac.uk

***Abstract*—Differentiable particle filters are an emerging class of models that combine sequential Monte Carlo techniques with the flexibility of neural networks to perform state space inference. This paper concerns the case where the system may switch between a finite set of state-space models, *i.e.* regimes. No prior approaches effectively learn both the individual regimes and the switching process simultaneously. In this paper, we propose the neural network based regime learning differentiable particle filter (RLPF) to address this problem. We further design a training procedure for the RLPF and other related algorithms. We demonstrate competitive performance compared to the previous state-of-the-art algorithms on a pair of numerical experiments.**

***Index Terms*—Differentiable particle filtering, Regime-switching, Sequential Monte Carlo.**

I. Introduction

Particle filters, first introduced in [1], are a class of Monte Carlo sampling algorithms that sequentially update the posterior distribution of the unobserved state of a dynamical system upon receipt of noisy measurements. Systems of this structure are known as state-space models (SSMs). Particle filters have found applications in target tracking [2], robot localisation [3], [4], and financial product risk analysis [5]. Classical particle filtering algorithms require prior knowledge of the functional form of the SSM. Differentiable particle filters (DPFs) [3], [6] represent a recent effort to reduce the required prior knowledge by parameterising part of the model with flexible neural networks.

We consider the problem that the system of interest may vary dynamically between a discrete set of candidate state-space models or regimes. In [7], the authors introduced a particle filter where the regime choice is assumed to be a realisation of a Markov chain. In [8] the regime switching particle filter (RSPF) was introduced as a generalisation to settings where the regime choice can depend arbitrarily on its history. The regime switching differentiable bootstrap particle filter (RSDBPF) [9] learns a neural network parameterisation of the RSPF using the framework of differentiable particle filtering. However, the RSDBPF still requires that the probabilistic meta-model that determines regime choice, hereafter referred to as the "switching dynamic", is known a priori.

In the particle filtering literature, there have been past efforts to handle systems where the switching dynamic is unknown. One such example is the model averaging particle filter (MAPF) [10] where a separate particle filter is run for each regime. Computational effort is assigned per filter according to the posterior probability of each regime. Optionally, the filters are allowed to occasionally exchange particles; this strategy is poorly suited to settings where the model is allowed to change frequently.

In this paper, we introduce the regime learning particle filter, which uses neural networks to parameterise the switching dynamic. Our contributions are two-fold:

1) We develop a neural network parameterisation of regime switching systems, capable of learning the switching dynamic.
2) We propose a novel algorithm to train differentiable particle filters with a Markovian marginal component, that we demonstrate empirically to improve accuracy.

The structure of the paper is as follows: in Section II we formally set-up the problem the paper addresses; in Section III we review the required background knowledge. Section IV introduces the proposed regime learning particle filter algorithm, with Sections IV-A and IV-B introducing a novel parameterisation and Section IV-C devoted to a proposed training strategy. In Section V we demonstrate competitive performance, with respect to prior state-of-the-art approaches, on the test environments adopted in previous works. We conclude this paper in Section VI.

II. Problem Formulation

A state-space model describes a system of two components, an unobserved discrete time Markov process $\{\mathbf{x}_t\}$ and its noisy observations $\{\mathbf{y}_t\}$. In our problem we allow the system to randomly jump, at any time, between a number of different SSMs, indexed by $\{k_t\}$. We impose the assumption that the choice of model, $\{k_t\}$, is independent of $\{\mathbf{x}_t, \mathbf{y}_t\}$, but can

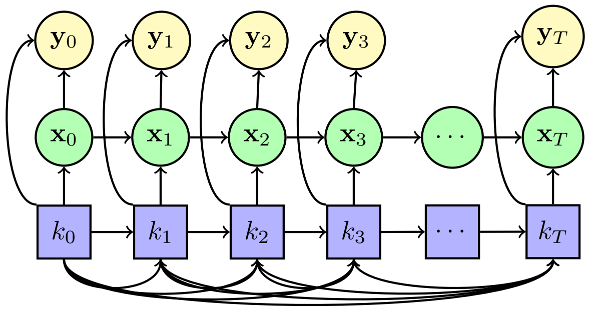

Fig. 1. Bayesian network representation of the general regime switching model.

depend arbitrarily on its history. We illustrate this system graphically in Fig. 1, and represent it algebraically as:

$$\begin{aligned} k_0 &\sim K_0^\theta\left(k_0\right) , \\ \mathbf{x}_0 &\sim M_0^\theta\left(\mathbf{x}_0|k_0\right) , \\ k_{t\geq 1} &\sim K^\theta\left(k_t|k_{0:t-1}\right) , \\ \mathbf{x}_{t\geq 1} &\sim M^\theta\left(\mathbf{x}_t|\mathbf{x}_{t-1}, k_t\right) , \\ \mathbf{y}_t &\sim G^\theta\left(\mathbf{y}_t|\mathbf{x}_t, k_t\right) . \end{aligned} \tag{1}$$

Throughout this paper we will refer to t as the "time-index"; θ as the "model parameters"; k_t, which may only take integers in the set $\mathcal{K} = \{0, \ldots, K\}$, as the "model index"; $\mathbf{x}_t$ as the "latent state"; and $\mathbf{y}_t$ as the "observations". The model components shall be referred to as follows: K_0^θ, K^θ as the "switching dynamic"; M_0^θ, M^θ as the "dynamic models"; and G^θ as the "observation models". For the sake of simplicity and to maintain clarity in our discussion, we allow the overloading of notations and use $M_0^\theta(\mathbf{x}_0)$, $M^\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $G^\theta(\mathbf{y}_t|\mathbf{x}_t)$ to represent the prior, dynamic model and measurement model, respectively, when there is a single regime, *i.e.* the system is a vanilla SSM. We assume that during training we have access to the ground truth state, $\{\mathbf{x}_t\}$, but not the model indices, $\{k_t\}$.

We desire to find an accurate estimator of $\mathbf{x}_t$ given $\mathbf{y}_{0:t}$, we propose to do this via a particle filtering estimate of $\mathbb{E}^\theta\left[\mathbf{x}_t|\mathbf{y}_{0:t}\right]$, denoted by $\hat{\mathbf{x}}_t$.

III. Preliminaries

A. Regime switching particle filtering

We first introduce a generic particle filtering framework. Classical particle filtering algorithms provide inference on SSMs, *i.e.* our problem (Eq. (1)) if there were only one regime.

Given an SSM, a particle filter is a procedure to sequentially obtain an importance sample from its filtering distribution, $P\left(\mathbf{x}_t|\mathbf{y}_{0:t}\right)$. A tutorial on particle filtering can be found in [11]. The full algorithm is illustrated in pseudo-code in Algorithm 1.

Algorithm 1 Generic Particle Filter. All operations indexed by i should be repeated for all $i \in \{1, \ldots, N\}$.

```
Input:  prior $M_0$                          dynamic model $M$
        proposal prior $Q_0$                 proposal $Q$
        observation model $G$                time length $T$
        particle count $N$                   observations $\mathbf{y}_{0:T}$
Output: particle locations $\tilde{\mathbf{x}}_{0:T}^{0:N}$    particle weights $w_{0:T}^{0:N}$
        normalised weights $\bar{w}_{0:T}^{0:N}$
1:  Sample $\tilde{\mathbf{x}}_0^i \sim Q_0\left(\tilde{\mathbf{x}}_0^i\right)$;
2:  $w_0^i \leftarrow \frac{M_0\left(\tilde{\mathbf{x}}_0^i\right)}{Q_0\left(\tilde{\mathbf{x}}_0^i\right)} G\left(\mathbf{y}_0|\tilde{\mathbf{x}}_0^i\right)$;
3:  $\bar{w}_0^i \leftarrow \frac{w_0^i}{\sum_n^N w_0^n}$;
4:  for $t = 1$ to $T$ do
5:      Set the resampled indices and weights, $A_t^i$, $\tilde{w}_t^i$, according to the chosen resampling scheme;
6:      Sample $\tilde{\mathbf{x}}_t^i \sim Q\left(\tilde{\mathbf{x}}_t^i|\tilde{\mathbf{x}}_{t-1}^{A_t^i}\right)$;
7:      $w_t^i \leftarrow \tilde{w}_t^i \frac{M\left(\tilde{\mathbf{x}}_t^i|\tilde{\mathbf{x}}_{t-1}^{A_t^i}\right)}{Q\left(\tilde{\mathbf{x}}_t^i|\tilde{\mathbf{x}}_{t-1}^{A_t^i}\right)} G\left(\mathbf{y}_t|\tilde{\mathbf{x}}_t^i\right)$;
8:      $\bar{w}_t^i \leftarrow \frac{w_t^i}{\sum_n^N w_t^n}$;
9:  end for
10: return $\tilde{\mathbf{x}}_{0:T}^{1:N}$, $w_{0:T}^{1:N}$, $\bar{w}_{0,T}^{1:N}$.
```

Several quantities can be estimated from the particle approximation of the filtering distribution; of interest to us are the approximation of the filtering mean, $\mathbb{E}\left[\mathbf{x}_t|\mathbf{y}_{0:t}\right]$:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^N \tilde{\mathbf{x}}_t^i \bar{w}_t^i, \tag{2}$$

where N is the total number of particles; and the observation-likelihood:

$$\hat{p}\left(\mathbf{y}_{0:t}\right) = \prod_{s=0}^t \frac{1}{N} \sum_{i=1}^N w_t^i. \tag{3}$$

The mean, $\hat{\mathbf{x}}_t$, relies on auto-normalised importance weights so is biased in general [12], but $\hat{p}\left(\mathbf{y}_{0:t}\right)$ is unbiased [13]. The derivation of both equations can be found in [12].

If one is able to express a problem as an SSM then they can apply particle filtering (Alg. 1). For example, the regime switching problem (Eq. (1)) can be reformulated as an SSM by taking the latent state to be $\{\mathbf{x}_t, k_{0:t}\}$, and the observations as $\{\mathbf{y}_t\}$. This is the strategy the RSPF [8] uses to approximate the joint posterior $P\left(\mathbf{x}_t, k_t|\mathbf{y}_{0:t}\right)$.

B. Differentiable particle filtering

Differentiable particle filters (DPFs) [3], [6] propose to learn an optimal parameter set via stochastic gradient descent (SGD) on some target loss function, $\mathcal{L}^\theta$. The adoption of SGD, rather than the traditionally preferred expectation maximisation (EM) algorithm [14], allows the flexibility in model required to use a neural network parameterisation.

In DPFs the gradient of the loss is calculated by back-propagating through the filtering process. This raises an issue for the sampling steps; sampling a continuous distribution in a differentiable way is straightforward: choose the sampling output to be a differentiable, deterministic function of its input and some random variable that is independent of any values we want to pass gradient through – this is known as "the reparameterisation trick". Unfortunately, in the case that the target distribution is discrete, as it is during resampling, no such differentiable function exists in general. An alternative approach is to sample from some prior distribution over the classes and perform importance sampling.

In practice, although the true prior on the resampling indices is uniform, a uniform proposal can be too far from the target distribution to produce effective importance samples. So, at the cost of biased gradients, we adopt a proposal that is a mixture of the target distribution, with some probability α; and a uniform distribution, with probability $1-\alpha$. This is known as "soft-resampling" [4]. More recent unbiased resampling schemes, such as optimal transport-based resampling [15], do not apply as they require the latent state to be continuous. Further detail on DPFs can be found in [16].

In supervised settings, defined as having access to both the ground truth latent state $\mathbf{x}_{0:T}$ as well as the observations $\mathbf{y}_{0:T}$ in training, it is common to directly use the target objective as the training loss. In the non-regime-switching analog to our problem, the appropriate loss function is the mean squared error between the estimated states and the ground truth, as previously used in [3], [4]:

$$\mathcal{L}_{\text{MSE}}\left(\hat{\mathbf{x}}_{0:T}, \mathbf{x}_{0:T}\right) = \frac{1}{T+1}\sum_{t=0}^{T}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 \,. \tag{4}$$

One can also consider the unsupervised problem, where only the observations are available during both training and testing. In this case it is common to use the usual variational inference approach of minimising an evidence lower bound [17]–[19]:

$$\mathcal{L}_{\text{ELBO}}\left(\mathbf{y}_{0:T}\right) = -\log\left(\hat{p}\left(\mathbf{y}_{0:T}\right)\right) \,. \tag{5}$$

Note that, whilst the estimator of the likelihood $\hat{p}\left(\mathbf{y}_{0:T}\right)$ is unbiased, its back-propagated gradient is not. $\hat{\mathbf{x}}_t$ and $\hat{p}\left(\mathbf{y}_{0:T}\right)$ are defined in Eqs. (2) and (3) respectively.

The regime switching differentiable particle filter (RS-DBPF) was proposed in [9] to address the regime-switching filtering problem (1) where the switching dynamic is assumed to be known but the other model components are not. It can be thought of a RSPF where the parameters are learned by gradient descent. They choose not to employ soft-resampling, and instead zero out the contribution to the gradient of the particles due their ancestors at every time-step, improving gradient variance at the cost of bias. This is found empirically to improve performance in some cases [20].

IV. The regime-learning particle filter

In this section we propose the regime learning particle filter (RLPF). We first, in Section IV-A, introduce an equivalent redefinition of the problem, Eq. (1), that naturally leads to a structure which we paramaterise in Section IV-B. Finally, we introduce a novel training strategy in Section IV-C.

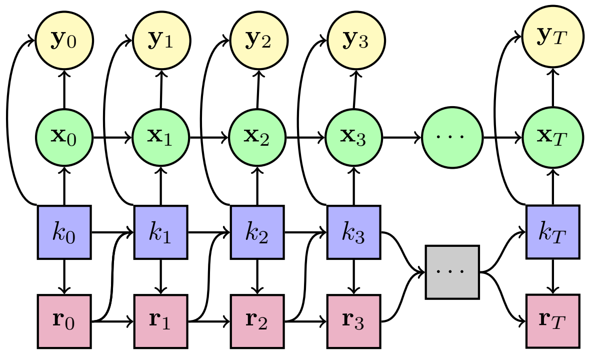

Fig. 2. Bayesian network representation of the proposed modified regime switching model.

A. Redefining the model

We find it instructive, both from an implementation and an understanding perspective, to propose the following reformulation of the regime-switching dynamic in (1):

$$\begin{aligned} k_0 &\sim K_0^{\theta}\left(k_0\right) \,, \\ k_{t\geq 1} &\sim K^{\theta}\left(k_t | \mathbf{r}_{t-1}\right) \,, \\ \mathbf{r}_{t\geq 0} &= R^{\theta}\left(k_t, \mathbf{r}_{t-1}\right) \,. \end{aligned} \tag{6}$$

One can easily see that this formulation is equivalent to Eq. (1) by taking $R^{\theta}\left(k_{t-1}, \mathbf{r}_{t-1}\right) = k_{t-1} \bigoplus \mathbf{r}_{t-1}$, where $\bigoplus$ denotes concatenation, but doing so requires an unbounded amount of memory. In practice, and due to our desire to parameterise the switching dynamic by neural networks, we apply the practical constraint $\mathbf{r}_t \subseteq \mathbb{R}^{d_r}$ for a constant dimensionality d_r. The problem of learning the switching dynamic then reduces to finding embedding functions R^{θ}, and regime probability masses, K^{θ}. K_0^{θ} can be represented as a learnable vector, and we define $\mathbf{r}_{-1}$ to be a vector of zeros, but alternatively it can be learned. We note that $\{\mathbf{x}_t, k_t, \mathbf{r}_t\}$ is a Markov process; it is clear that (6) is an SSM. So particle filtering (Alg. 1) can be use to estimate the joint posterior $P\left(\mathbf{x}_t, k_t, \mathbf{r}_t | \mathbf{y}_{0:t}\right)$. We illustrate this formulation as a Bayesian network diagram in Fig. 2.

B. Parameterising the switching dynamic

We propose a neural network parameterisation of the switching dynamic. It is well known that particle filters for which the late-time state depends strongly on the early-time state do not preform well, in fact, permitting strong dependence, one can construct filters that diverge in variance at any desired rate [12]. Late-time particles are genealogically non-diverse at early time and so form poor samples of the early-time state; a phenomenon often referred to as "path-degeneracy". For this reason, we take inspiration from the architecture of the long-short term memory (LSTM) unit [21] and build forget gates into our model as:

$$\begin{aligned} \mathbf{r}_t =& R^{\theta}\left(\mathbf{k'}_t, \mathbf{r}_{t-1}\right) \\ =& \sigma\left(\Theta_1 \mathbf{r}_{t-1}\right) \odot \sigma\left(\Theta_2 \mathbf{k'}_t\right) \odot \mathbf{r}_{t-1} \\ &+ \tanh\left(\Theta_3 \mathbf{k'}_t\right) \odot \sigma\left(\Theta_4 \mathbf{k'}_t\right) \,, \end{aligned} \tag{7}$$

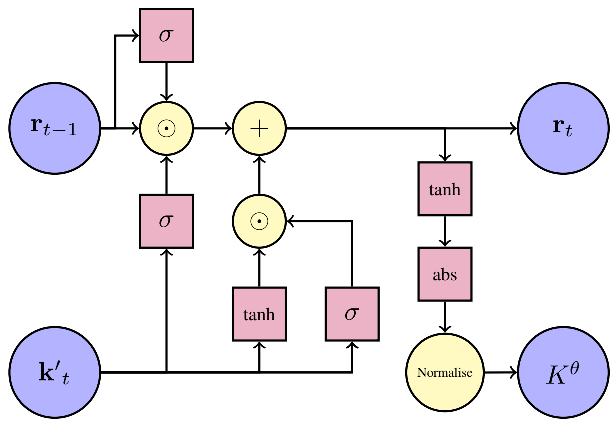

Fig. 3. Graphical representation of our proposed switching dynamic. Blue nodes are input/outputs, purple nodes are fully connected network layers with the specified activation, and yellow nodes are non-learned functions. The switching probability mass, $K^\theta(k_{t+1}|\mathbf{r}_t)$, is the value at the k_{t+1}^{th} index of the model output K^θ.

where $\mathbf{k}'_t$ is the one-hot enconding of the model index; $\{\Theta_{1:4}\}$ are matrices of weights to be learned; $\odot$ denotes the Hadamard product.

$$K'^\theta(\mathbf{k}'_t|\mathbf{r}_{t-1}) = |\Theta_5 \tanh(\Theta_6 \mathbf{r}_{t-1})| \cdot \mathbf{k}'_t,$$
$$K^\theta(\mathbf{k}'_t|\mathbf{r}_{t-1}) = \frac{K_t'^\theta(\mathbf{k}'_t|\mathbf{r}_{t-1})}{\sum_{c\in\mathcal{K}} K'^\theta(\mathbf{c}'|\mathbf{r}_{t-1})}, \tag{8}$$

where $\cdot$ denotes a dot product, and Θ_5, Θ_6 are additional learnable weight matrices. We represent the model graphically in Fig. 3.

As with resampling, $K^\theta(k_t|\mathbf{r}_{t-1})$ is a discrete distribution; in order to back-propagate gradients through the network we use importance sampling to select the model indices.

C. Training the RLPF

We adopt a novel training strategy. Instead of treating the problem as a supervised regression on $\hat{\mathbf{x}}_{0:T}$ with the MSE-loss $\mathcal{L}_{\text{MSE}}(\hat{\mathbf{x}}_{0:T}, \mathbf{x}_{0:T})$ (Eq. (4)) as in previous work [9], we introduce an additional $\mathcal{L}_{\text{ELBO}}(\{\mathbf{x}_{0:T}, \mathbf{y}_{0:T}\})$ (Eq. (5)) unsupervised loss term in the training objective.

$$\mathcal{L}^\theta_{\text{RLPF}}(\hat{\mathbf{x}}_{0:T}, \mathbf{x}_{0:T}, \mathbf{y}_{0:T}) = \mathcal{L}_{\text{ELBO}}(\{\mathbf{x}_{0:T}, \mathbf{y}_{0:T}\}) + \lambda\mathcal{L}_{\text{MSE}}(\hat{\mathbf{x}}_{0:T}, \mathbf{x}_{0:T}), \tag{9}$$

where $\lambda \in \mathbb{R}^+$ is a hyper-parameter.

For the supervised term, we run a particle filter on an SSM defined as in Section IV-A, with the state as $\{\mathbf{x}_t, k_t, \mathbf{r}_t\}$ and the observation as $\mathbf{y}_t$. For the unsupervised term, we treat all the information we have access to in training as observed, *i.e.* take $\{\mathbf{x}_t, \mathbf{y}_t\}$ to be the observations. However, in doing so the observations depend on their history not only through the state, so is not, by our definition, an SSM. However, treating the past observations as constant, it is easy to show that the usual particle filtering algorithm remains correct. We refer to this strategy as a "marginal filter" as it runs a particle filter only on the marginal process that is the switching dynamic. The intuition behind this strategy is that training on the ELBO loss maximises the data likelihood to bring the learned model closer to the true model; and the MSE loss serves to guide the training towards a lower validation objective.

The validation objective is the MSE calculated by running a filter with the θ learned by optimising the combined loss. Furthermore, there is no need for the algorithm to be differentiable during evaluation, so there we use the non-differentiable systematic resampling [22] and the target distribution, K^θ, to propose the model indices.

We present the full RLPF procedure in Algorithm 2.[1]

Algorithm 2 Regime Learning Particle Filter. The $\bullet$ operator denotes multiplication of probability densities, and β is the learning rate according to optimiser choice.

```
Input: priors $M_0^\theta$                    dynamic models $M^\theta$
       regime prior $K_0^\theta$              switching dynamic $K^\theta$
       observation models $G^\theta$          encoding functions $R^\theta$
       time length $T$                        particle count $N$
       loss coefficient $\lambda$             observations $\mathbf{y}_{0:T}$
       ground truth $\mathbf{x}_{0:T}$
Output: model parameters $\theta$
 1: while $\theta$ not converged do
 2:     Run a particle filter (Algorithm 1) to obtain $\tilde{\mathbf{x}}_{0:T}^{1:N}, \bar{w}_{0:T}^{1:N}$
        with inputs:
        $M_0 \leftarrow K_0^\theta \bullet M_0^\theta \bullet R^\theta$   $Q_0 \leftarrow \text{Uniform}(k\in\mathcal{K}) \bullet M_0^\theta \bullet R^\theta$
        $M \leftarrow K^\theta \bullet M^\theta \bullet R^\theta$   $Q \leftarrow \text{Uniform}(k\in\mathcal{K}) \bullet M^\theta \bullet R^\theta$
        $G \leftarrow G^\theta$   $T \leftarrow T$
        $N \leftarrow N$   $\mathbf{y}_{0:T} \leftarrow \mathbf{y}_{0:T}$
 3:     $\hat{\mathbf{x}}_{0:T} \leftarrow \sum_{n=1}^N \tilde{\mathbf{x}}_{0:T}^n \bar{w}_{0:T}^n$, Eq. (2)
 4:     $\mathcal{L}^\theta_{\text{MSE}} \leftarrow \frac{1}{T+1}\sum_{t=0}^T \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|$, Eq. (4)
 5:     Run a particle filter (Algorithm 1) to obtain $w_{0:T}^{1:N}$ with
        inputs:
        $M_0 \leftarrow K_0^\theta \bullet R^\theta$,   $Q_0 \leftarrow \text{Uniform}(k\in\mathcal{K}) \bullet R^\theta$
        $M \leftarrow K^\theta \bullet R^\theta$   $Q \leftarrow \text{Uniform}(k\in\mathcal{K}) \bullet R^\theta$
        $G_0 \leftarrow G^\theta \bullet M_0^\theta$   $G_{t\geq 1} \leftarrow G^\theta \bullet M^\theta(\cdot|\mathbf{x}_{t-1})$
        $T \leftarrow T$   $N \leftarrow N$
        $\mathbf{y}_{0:T} \leftarrow \{\mathbf{y}_{0:T}, \mathbf{x}_{0:T}\}$
 6:     $\mathcal{L}^\theta_{\text{ELBO}} \leftarrow -\sum_{t=0}^T \log\left(\frac{1}{N}\sum_{n=1}^N w_t^n\right)$, Eqs. (3), (5)
 7:     $\mathcal{L}^\theta_{\text{RLPF}} \leftarrow \mathcal{L}^\theta_{\text{ELBO}} + \lambda\mathcal{L}^\theta_{\text{MSE}}$, Eq. (9)
 8:     $\theta \leftarrow \theta + \beta\nabla_\theta\mathcal{L}^\theta_{\text{RLPF}}$
 9: end while
10: return $\theta$
```

V. Experiments

We repeat the experiment set-up adopted in [8] and [9], where $x_t, y_t \in \mathbb{R}$ and, for each of the eight regimes, the dynamic model and observation model are linear Gaussian and non-linear Gaussian respectively.

[1]Our implementation, that was used to generate all results reported in this paper, can be found at: https://github.com/John-JoB/Regime_Switching

A. Dynamic and observation models

$$\begin{aligned}
M_0(x_0) &= \mathcal{U}(-0.5, 0.5) \ , \\
M(x_t|x_{t-1}, k_t) &= \mathcal{N}\left(\mu_M(x_{t-1}, k_t), \sigma^2\right) \ , \\
G(y_t|x_t, k_t) &= \mathcal{N}\left(\mu_G(x_t, k_t), \sigma^2\right) \ , \\
\mu_M(x_{t-1}) &= a_{k_t} x_{t-1} + b_{k_t} \ , \\
\mu_G(x_\tau) &= a_{k_t}\sqrt{|x_t|} + b_{k_t} \ , \\
[a_1, \dots, a_8] &= [-0.1, -0.3, -0.5, -0.9, 0.1, 0.3, 0.5, 0.9] \ , \\
[b_1, \dots, b_8] &= [0, -2, 2, -4, 0, 2, -2, 4] \ , \\
\sigma^2 &= 0.1 \ ,
\end{aligned} \tag{10}$$

where $M_0(x_0), M(x_t|x_{t-1}, k_t), G(y_t|x_t, kt)$ are the true models, and we have defined eight discrete regimes. Interestingly, $G(y_t|x_t, k_t)$ is bimodal, so accurate estimation of the state at time t requires using information over time-steps. For example, Regimes 1 and 5 lead to identical distributions on the observations despite their disparate posteriors. So, knowledge of the switching dynamic is crucial in accurately estimating the full posterior of the switching system.

B. Switching dynamic

We test our algorithm on two switching dynamics: a Markov switching system and a Pólya-Urn distribution. For both dynamics, $K_0^\theta(k_0)$ is uniform over $\mathcal{K}$. The Markov switching dynamic at non-zero time can be expressed as:

$$\begin{gathered}
K(k_t|k_{0:t-1}) = (\mathbf{k}'_t)^T B\mathbf{k}'_{t-1} \ , \\
B = \begin{pmatrix}
0.8 & 0.15 & \rho & \dots & \rho \\
\rho & 0.8 & 0.15 & \dots & \rho \\
\vdots & & \ddots & & \vdots \\
\rho & \rho & \dots & 0.8 & 0.15 \\
0.15 & \rho & \dots & \rho & 0.8
\end{pmatrix} \ , \\
\rho = \frac{1}{120} \ ,
\end{gathered} \tag{11}$$

where, as before, $\mathbf{k}'_t$ is the one-hot encoding of k_t. And the Pólya-Urn distribution:

$$K(k_t|k_{0:t-1}) = \frac{1 + \sum_{s=0}^{t} \mathbb{1}(k_s = k_t)}{8 + \sum_{c=1}^{8}\sum_{s=0}^{t} \mathbb{1}(k_s = c)} \ . \tag{12}$$

C. Baseline Models

The point estimation of states can be considered as a sequence-to-sequence supervised learning task; we desire to learn sequence $x_{0:T}$ given the sequence $y_{0:T}$. As such, we can apply any state-of-the-art sequence-to-sequence learning algorithms. We compare with the baseline models of a unidirectional LSTM [21], and a decoder-only transformer model [23]. The recurrent structure of an LSTM is more alike the underlying SSMs; however, as transformer models have been receiving growing attention, we include it for completeness.

The LSTM and Transformer baselines cannot provide an interpretable statistical output, as the proposed model does. So, we introduce two particle filtering baselines. The first is to run a DPF where the latent state is taken to be x_t only. This strategy implicitly assumes that there is no regime switching so we expect poor performance in the examined settings. The second is a modification of the model averaging particle filter (MAPF) [10] to make it differentiable, which we name "the model averaging differentiable particle filter" (MADPF). The MAPF was proposed to solve the inverse to the problem addressed by the RSDBPF; it requires knowledge of the individual models but not the switching dynamic. The MADPF is a MAPF trained by gradient descent, analogous to how the RSDBPF is derived from the RSPF.

We include two oracle approaches: the RSDBPF which simulates from the true switching dynamic but learns the individual regimes; and the RSPF which corresponds to a particle filter run on the true model.

The novel training strategy proposed in Section IV-C and the parameterisation of the switching dynamic, in Sections IV-A and IV-B, can be applied independently. So we include two ablation models, an RLPF trained on solely the MSE loss, and an RSDBPF trained on our proposed loss, equation (9).

D. Experiment settings

We generate 2,000 trajectories from the model (Eqs. (10) with (11) or (12)) and use 1,000 for training, 500 for validation, and 500 for testing. All algorithms are initialised and trained from scratch for 20 repeats on a different generation of the dataset. During training and validation, particle filters are run with 200 particles; this is increased to 2,000 for testing. The parameter set that obtained the best validation loss is passed to testing. We use the Adam optimiser with weight decay [24].

Throughout the experiments, we parameterise each μ_G and μ_M with a two-layer fully connected neural network with hyperbolic tangent activation on the hidden layer. We set the dimensionality of $\mathbf{r}_t$ to the number of models. All other hyperparameters are chosen by grid search, on a per-experiment basis. Soft resampling is only used for the marginal filter. When using $\mathcal{L}^\theta_{\text{RLPF}}$, we impose a prior that the dynamic and observation model variances are less than 1; or else the ELBO is lowered rapidly in the first few iterations by raising these variances, instead of approximating the true model.

It was shown in [8] that, for these experiments, the RSPF performs equally well whether the model indices are proposed uniformly or from the target distribution ($K^\theta(k_t)$). So, we use a uniform proposal during training. However, in a case where there are a large number of very unlikely regimes, a uniform proposal would lead to highly inefficient sampling. In such a case, introducing bias-inducing soft-sampling, as we did in section III-B for resampling, might improve performance.

E. Results

We present the results for both experiments in Table I including baselines and ablation results, and a comparison of the computation costs under the Pólya-Urn experiment in Table II for the non-oracle methods. The Markov times were similar so are excluded for brevity. For the RLPF and the RSDBPF the qualification -λ or -MSE denotes whether the experiment used the proposed training algorithm (Section IV-C), or the classical MSE only approach. All timing experiments were performed using an NVIDA RTX 4090 GPU.

For both experiments, the proposed RLPF-λ leads to the smallest mean squared errors among non-oracle algorithms. Moreover, being a particle filter, it has a statistical interpretation; with it one can evaluate properties such as uncertainties, the data likelihood, and make predictions about future state or data. The LSTM is the next best in the reported error metrics, but it does not offer the statistical properties seen in particle filters. We also note that for both the RSDBPF and the RLPF, the variants trained using the proposed training objective achieved lower MSEs and variances than the respective pure-MSE approaches.

Aside from the Transformer, all algorithms considered scale linearly in time with trajectory length. However, the relatively simple architecture of the LSTM and better ability to take advantage of GPU parallelism give it a significant speed advantage, as shown in Table II. Achieving better parallelism for the RLPF to bring its time cost closer to that of the DPF is left for future work.

TABLE I
FILTERING ACCURACY FOR THE DISCUSSED ALGORITHMS. REPORTED VALUES ARE THE ACHIEVED MEAN SQUARED FILTERING ERROR, AVERAGED ACROSS 20 INDEPENDENT TRAINING RUNS.

Algorithm	Markov MSE	Pólya MSE
Transformer (baseline)	1.579 ± 0.169	1.508 ± 0.112
LSTM (baseline)	0.713 ± 0.114	0.655 ± 0.027
DBPF (baseline)	4.689 ± 4.689	2.395 ± 0.723
MADPF (baseline)	2.688 ± 0.607	1.987 ± 0.171
RLPF-MSE (baseline)	0.970 ± 0.187	0.693 ± 0.080
RLPF-λ (proposed)	$\mathbf{0.698 \pm 0.164}$	$\mathbf{0.613 \pm 0.072}$
RSDBPF-λ (partial oracle)	0.802 ± 0.128	0.590 ± 0.064
RSDBPF-MSE (partial oracle)	1.153 ± 0.217	0.709 ± 0.122
RSPF (oracle)	0.274 ± 0.019	0.413 ± 0.012

TABLE II
AVERAGE COMPUTATION TIMES PER TRAINING EPOCH (10 BATCHES OF 100 PARALLEL FILTERS OF 200 PARTICLES EACH), AND TESTING RUN (1 BATCH OF 500 PARALLEL FILTERS OF 2000 PARTICLES EACH) ON THE PÓLYA EXPERIMENT.

Algorithm	Av. train epoch time (s)	Av. test time (s)
Transformer (baseline)	0.182	0.00310
LSTM (baseline)	**0.0145**	**0.000792**
DBPF (baseline)	0.553	0.243
MADPF (baseline)	18.6	3.73
RLPF-MSE (baseline)	4.42	0.620
RLPF-λ (proposed)	6.66	0.612

VI. CONCLUSIONS

In this paper we have proposed a novel particle filtering approach suited to situations where the model may switch, according to some unknown dynamic, between a set of candidate models whose forms are also unknown. We designed a learning strategy and demonstrated its superiority experimentally. Future directions include the incorporation of more advanced DPFs, more interpretable regime identification procedure, and more complex test scenarios including real world data.

REFERENCES

[1] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proc. F, Radar and Sig. Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[2] R. Jinan and T. Raveendran, "Particle filters for multiple target tracking," *Procedia Technology*, vol. 24, pp. 980–987, 2016.

[3] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," in *Proc. Robot.: Sci. and Syst.*, Pittsburgh, PA, USA, July 2018.

[4] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," in *Proc. Conf. on Robot Learning*, Zurich, CH, Oct 2018, pp. 169–178.

[5] E. Vankov, M. Guindani, and K. Ensor, "Filtering and estimation for a class of stochastic volatility models with intractable likelihoods," *Bayesian Analysis*, vol. 14, Mar 2018.

[6] X. Chen and Y. Li, "Normalising flow-based differentiable particle filters," 2024, *arXiv:2403.01499*.

[7] S. Mcginnity and G. Irwin, "A multiple model extension to the bootstrap filter for manoeuvring targets," in *IFAC Proc. Workshop Alg. & Arch. for Real Time Contr. (AARTC)*, Cancun, Mexico, April 1998, pp. 23–28.

[8] Y. El-Laham, L. Yang, P. M. Djurić, and M. F. Bugallo, "Particle filtering under general regime switching," in *Proc. Euro. Sig. Process. Conf. (EUSIPCO)*, Amsterdam, NL, Jan 2021, pp. 2378–2382.

[9] W. Li, X. Chen, W. Wang, V. Elvira, and Y. Li, "Differentiable bootstrap particle filters for regime-switching models," in *Proc. IEEE Stat. Sig. Process. Workshop (SSP)*, Hanoi, Vietnam, 2023, pp. 200–204.

[10] L. Martino, J. Read, V. Elvira, and F. Louzada, "Cooperative parallel particle filters for online model selection and applications to urban mobility," *Digit. Sig. Process.*, vol. 60, pp. 172–185, 2017.

[11] A. Doucet and A. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later*, 2009.

[12] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*. New York, USA: Springer International Publishing, 2020.

[13] P. Del Moral and A. Doucet, "Particle methods: An introduction with applications," in *ESAIM: Proc.*, vol. 44, Jan 2014.

[14] N. Kantas, A. Doucet, S. Singh, and J. Maciejowski, "An overview of sequential monte carlo methods for parameter estimation in general state-space models," in *IFAC Proc. Volumes*, vol. 42, no. 10, Saint-Malo, FR, 2009, pp. 774–785.

[15] A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet, "Differentiable particle filtering via entropy-regularized optimal transport," in *Proc. Int. Conf. on Machine Learn. (ICML)*, Online, 18–24 Jul 2021, pp. 2100–2111.

[16] X. Chen and Y. Li, "An overview of differentiable particle filters for data-adaptive sequential bayesian inference," *Founds. of Data Sci.*, Oct 2023.

[17] C. Naesseth, S. Linderman, R. Ranganath, and D. Blei, "Variational sequential monte carlo," in *Int. Conf. Art. Int. and Stat. (AISTATS)*. Lanzarote, Canary Islands: PMLR, April 2018, pp. 968–977.

[18] T. Le, M. Igl, T. Rainforth, T. Jin, and F. Wood, "Auto-encoding sequential monte carlo," in *Int. Conf. Learn Represent. (ICLR)*, Vancouver, Canada, May 2018.

[19] C. J. Maddison, J. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. Teh, "Filtering variational objectives," *Proc. Adv. in Neural Info. Process. Syst. (NeurIPS)*, December 2017.

[20] M. Zhu, K. P. Murphy, and R. Jonschkowski, "Towards differentiable resampling," 2020, *arXiv:2004.11938*.

[21] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, *arXiv:1402.1128*.

[22] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," in *IEE Proc. Radar, Sonar and Navi.*, vol. 146, February 1999, pp. 2–7(5).

[23] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. in Neural Info. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, December 2017.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.