Signal Processing 238 (2026) 110166

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Differentiable interacting multiple model particle filtering

John-Joseph Brady [a,b,1,*], Yuhui Luo [b], Wenwu Wang [c], Víctor Elvira [d], Yunpeng Li [c,e]

[a] Computer Science Research Centre, University of Surrey, Guildford, Surrey, United Kingdom
[b] Data Science Department, National Physical Laboratory, Teddington, Greater London, United Kingdom
[c] Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, Surrey, United Kingdom
[d] School of Mathematics, University of Edinburgh, Edinburgh, United Kingdom
[e] Centre for Oral, Clinical & Translational Sciences, King's College London, London, United Kingdom

ARTICLE INFO

MSC:
62M20
62F12

Keywords:
Sequential Monte Carlo
Differentiable particle filtering
Regime switching

ABSTRACT

We propose a sequential Monte Carlo algorithm for parameter learning when the studied model exhibits random discontinuous jumps in behaviour. To facilitate the learning of high dimensional parameter sets, such as those associated to neural networks, we adopt the emerging framework of differentiable particle filtering, wherein parameters are trained by gradient descent. We design a new differentiable interacting multiple model particle filter to be capable of learning the individual behavioural regimes and the model which controls the jumping simultaneously. In contrast to previous approaches, our algorithm allows control of the computational effort assigned per regime whilst using the probability of being in a given regime to guide sampling. Furthermore, we develop a new gradient estimator that has a lower variance than established approaches and remains fast to compute, for which we prove consistency. We establish new theoretical results of the presented algorithms and demonstrate superior numerical performance compared to the previous state-of-the-art algorithms.

1. Introduction

There has been longstanding interest in Bayesian filtering for systems exhibiting discontinuous behavioural jumps, typically modelled by ascribing the system a finite number of distinct and indexed regimes. Two systems frequently modelled in this way include financial markets reacting swiftly to economic news [1,2], and tracked targets suddenly changing course or acceleration [3–6]. Much of this existing work is focused on Markov switching systems where the probability of jumping is allowed to depend only on the index of the current regime.

Particle filters [7,8] are a class of Monte-Carlo algorithms for estimating the posterior distribution of a Markov hidden signal given noisy observations of it. In the regime-switching setting, if the regime index is modelled as a Markov chain, one may treat it as a component of the hidden signal in a particle filter [3].

In [9] the authors developed the regime switching particle filter (RSPF), extending the approach in [3] to systems where the regime index can depend arbitrarily on its past. They achieved this by having every particle keep a memory of its entire regime history, similar to the fixed-lag smoother of [10]. The interacting multiple model particle filter (IMMPF), introduced in [5], assumes the regime index is a Markov chain, but allows it to depend on the latent state as well as the index at the previous time-step. We show that, under the reformulation of the non-Markov switching model that we develop in this paper, the former problem is a special case of the latter.

In many real world settings, the average number of time-steps a system spends in each regime can be large. This is typically modelled by taking switches to be rare events. Most particle filtering algorithms naturally focus computation on more likely regions of the state space. With a restricted particle count, overtime this can result in the number of particles in all regimes apart from the current one going to zero; so when jumps do occur they are not detected [6,11]. It has become common practice, therefore, in regime switching filters to set the number of particles assigned per regime at each time-step to be equal on average. This is achieved in [9] by proposing the regime index uniformly across all regime choices. However, this ignores the probability of each particle adopting the given regime.

The IMMPF takes a more principled approach, it combines the resampling and regime selection steps to improve sampling efficiency. However, the IMMPF is not strictly a particle filter in the sense studied in [12–14]. To the best of our knowledge, no proof of consistency for the IMMPF exists in the literature.

* Corresponding author at: Computer Science Research Centre, University of Surrey, Guildford, Surrey, United Kingdom.
E-mail address: john-joseph.brady@kcl.ac.uk (J.-J. Brady).
[1] J. Brady has transferred institutions to King's College London since this work was submitted for review.

https://doi.org/10.1016/j.sigpro.2025.110166
Received 1 October 2024; Received in revised form 17 December 2024; Accepted 30 April 2025
Available online 30 June 2025

Differentiable particle filters (DPFs) [15–17] are an emerging class of particle filters designed in such a way that the algorithm is end-to-end differentiable, so that one may obtain accurate gradient estimates for use in gradient based parameter inference. The motivating use case for DPFs is to learn model components as flexible neural networks, typically when the prior knowledge on the functional form of the underlying model is of poor quality. In this case, other parameter inference paradigms fail. For example, the EM algorithm [18] requires a specific functional form of the model for the maximisation step to be closed-form; and both derivative-free optimisation and particle Markov chain Monte Carlo [19] do not scale well to large dimensional parameter spaces.

The first effort to address switching models in a DPF framework is [20], with the regime switching differentiable bootstrap particle filter (RSDBPF). However, the RSDBPF is only capable of learning the individual regimes. The meta-model that controls the switching, henceforth the 'switching dynamic', is required to be known a priori. During inference, the RSDBPF runs a RSPF so does not sample particles as efficiently as the IMMPF. Furthermore, it has an asymptotically biased gradient update.

There are few approaches in the literature that operate under an unknown switching dynamic and, to the best of our knowledge, none in the more challenging parameter estimation framework. In [21], a related problem is studied: the system may belong to one of a set of candidate regimes but the regime may not change during a trajectory. Their strategy is to run a separate filter for each regime but assign computational effort, *i.e.* the number of particles, in proportion to the posterior probability that the system is in each regime. This strategy was generalised in [22], where the particles are permitted to occasionally exchange between regimes. However, this algorithm cannot provide a consistent estimator in the general case where the regime can switch at any time-step.

In this paper we propose the differentiable multiple model particle filter (DIMMPF), the first DPF approach to filtering regime-switching models where neither the individual models nor the switching dynamic are known. The DIMMPF can be seen as an IMMPF that can return statistically consistent estimates of the gradient of its filtering mean with respect to the model parameters.

The main contributions of this work,[2] can be summarised as follows:

- We present the DIMMPF, a novel algorithm for learning to estimate the filtering mean of a general regime-switching model.
- We develop a neural network architecture to parameterise a general unknown switching dynamic.
- We prove that the DIMMPF generates consistent estimators of filtering means and their gradients. Entailing a derivation of, to the best of our knowledge, the first proof that filtering estimates of the IMMPF are consistent.
- We evaluate the DIMMPF on a set of simulated data experiments and demonstrate state-of-the-art performance.

The remainder of this article is structured as follows. In Section 2 we introduce the problem statement. Section 3 reviews the relevant background for the paper and explains how this paper builds on previous work. Section 4 develops our algorithmic contribution, the DIMMPF. Section 5 describes the experiments and presents the results. We conclude in Section 6.

[2] A limited version of this work was presented by the authors in the conference paper [23] which presents a simpler version of our methodology that has a biased gradient update. The conference paper contains limited discussion, no theoretical insight, and a more basic set of experiments.

2. Problem statement

We define a state-space model (SSM) to describe a discrete time system of two parallel processes: a latent Markov process, $\{\hat{x}_t\}$; and their associated observations $\{\hat{y}_t\}$, where t is the discrete time index. Every observation $\hat{y}_t$ is conditionally independent of all other variables at previous time steps given $\hat{x}_t$. Algebraically, an SSM is defined as:

$$\begin{aligned} \hat{x}_0 &\sim \hat{M}_0\left(\hat{x}_0\right), \\ \hat{x}_{t\geq 1} &\sim \hat{M}\left(\hat{x}_t \mid \hat{x}_{t-1}\right), \\ \hat{y}_t &\sim \hat{G}\left(\hat{y}_t \mid \hat{x}_t\right), \end{aligned} \tag{1}$$

for the set of states $\hat{x}_t \in \mathcal{X}$, the set of observations $\hat{y}_t \in \mathcal{Y}$, the random measure $\hat{M}_0$, and the probability kernels $\hat{M}$ and $\hat{G}$.

We consider an SSM where at each time-step the latent and observation processes may, together, adopt one of a set of N_{reg} regimes. To model this system, we introduce two additional latent variables: the regime index, $k_t \in \mathcal{K} := \{1, 2, \ldots, N_{\text{reg}}\}$; and a cache that acts as a memory of previous regimes, $r_t \in \mathcal{R} \subseteq \mathbb{R}^{d_r}$, where d_r is the chosen dimension of the regime cache. We illustrate this system graphically in Fig. 1 and define it algebraically as:

$$\begin{aligned} r_0 &= R_0^\theta\left(k_0\right), \\ r_{t\geq 1} &= R^\theta\left(k_t, r_{t-1}\right), \\ k_0 &\sim K_0^\theta\left(k_0\right), \\ k_{t\geq 1} &\sim K^\theta\left(k_t \mid r_{t-1}\right), \\ x_0 &\sim M_0^\theta\left(x_0 \mid k_0\right), \\ x_{t\geq 1} &\sim M^\theta\left(x_t \mid x_{t-1}, k_t\right), \\ y_t &\sim G^\theta\left(y_t \mid x_t, k_t\right), \end{aligned} \tag{2}$$

where we have made explicit any dependence on the model parameters θ. R_0^θ and R^θ are deterministic functions, K_0^θ and K^θ are categorical distributions. To avoid confusion with the generic SSM Eq. (1), SSM model components and variables are denoted by a circumflex (ˆ) whereas components of the studied regime switching model, Eq. (2) are not. For notational simplicity we do not make explicit any time dependence of the model components; by treating the time as a series of constants, time dependence can be introduced without change to the theoretical analysis.

This paper addresses the problem of accurately estimating filtering means, $\mathbb{E}^\theta\left[x_t \mid y_{0:t}\right]$. Unlike previous work [9,20,22], our formulation allows all of the dynamic model, M_0^θ, M^θ; the observation model, G^θ; and the switching dynamic, R_0^θ, R^θ, K_0^θ, K^θ, to depend simultaneously on the learned parameters θ. However, we require that the number of regimes N_{reg} be given; efficiently determining N_{reg} is left for future work.

Problem analysis

The advantage of our formulation is that the joint hidden process $\{x_t, k_t, r_t\}$ is explicitly Markov. Identifying $\{x_t, k_t, r_t\}$ with $\{\hat{x}_t\}$ and $\{y_t\}$ with $\{\hat{y}_t\}$ it is clear that our model (2) is a special case of an SSM (1). Existing results and algorithms concerning SSMs, *i.e.*particle filters, can be applied directly in our setting.

It is instructive to demonstrate a pair of special cases of our model. Taking

$$R_0^\theta\left(k_0\right) = k_0,\ R^\theta\left(k_t, r_{t-1}\right) = k_t, \tag{3}$$

one recovers the popular Markov switching model [1,4,24]. Alternatively taking

$$R_0^\theta\left(k_0\right) = \{k_0\},\ R^\theta\left(k_t, r_{t-1}\right) = \left(\{k_t\} \cup r_{t-1}\right) \setminus \{k_{t-\tau}\}, \tag{4}$$

for some fixed τ, one obtains the model with perfect memory of its regimes for a fixed lag, similar to the strategy of [25]. Taking τ to be larger than the trajectory length gives the model with perfect regime memory, as considered by [9,20].

It is well known that, as a consequence of the resampling step, particle filters suffer from path degeneracy [26], wherein late-time

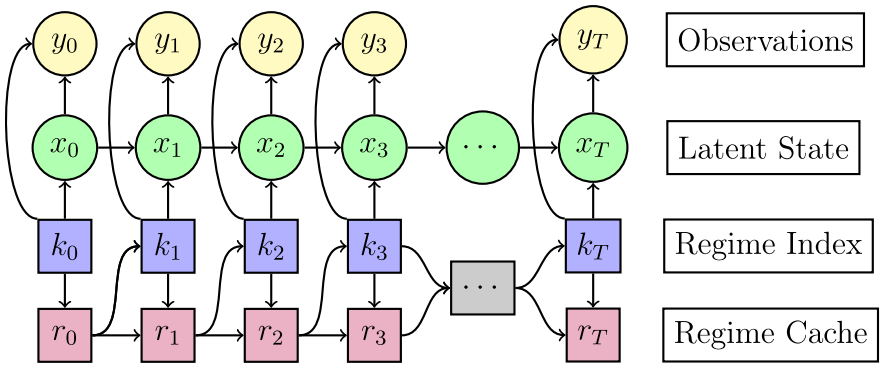

Fig. 1. Bayesian network representation of the considered regime switching model.

particles are descended from only a small subset of the early time particles. In the perfect memory formulation this means that late-time particles will form a poor quality sample of the early time regime indices, so keeping full trajectories is not useful. R^θ can be thought of a caching function that keeps the useful information in the regime history. Moreover, myriad practical challenges arise in trying to store and utilise a linearly growing amount of information.

Since the regime may take only one of N_{reg} values for each time-step, at time t, $|\mathcal{R}| \leq N_{\text{reg}}^{t+1}$, with $|\mathcal{R}|$ being the cardinality of the set $\mathcal{R}$. However, an $\mathbb{R}^{d_r}$ representation is more convenient for a neural network implementation.

3. Background

3.1. The interacting multiple model particle filter

Our algorithm may be seen as differentiable variant of the IMMPF [5], indeed, we show in Theorem 2 that the two are exactly equivalent on the forward pass. In [5] the IMMPF is formulated for regime switching systems where the system state and regime-index, $\{x_t, k_t\}$, jointly form a Markov process. Endowing the state with the auxiliary variable r_t it is clear that our problem, Eq. (2), is a special case of the problem solved by the IMMPF. For clarity, we present the IMMPF as applied to Eq. (2).

In filters for regime switching systems, to avoid the sample collapsing to a single regime, it is standard to propose the regime index uniformly [6,11]. The obvious strategy [9] to filter our system, Eq. (2), would therefore be to run a particle filter with hidden state $\{x_t, k_t, r_t\}$ and observations y_t. Where k_t is proposed uniformly. However, doing so is agnostic to each particles probability of being in its sampled state.

The IMMPF [5] approach is to modify the resampling step as so: for every particle at time t, to first choose a regime index, k_t, then resample a particle at time $t-1$ with weight equal to an estimate of their posterior probability conditional on being propagated to regime k_t at time t. The other components of the hidden state $\{x_t, r_t\}$ can be propagated in the usual way. Under this approach, the resampling probabilities depend on the value of the latent state at the next time-step, as such it is not possible to express this algorithm as a sequential importance sampling–resampling particle filter [8,13]. In practice we deterministically choose there to be an equal number of particles in each regime, but for the sake of exposition we focus on the case that the regime indices are uniformly sampled. We justify this simplification by the observation that the deterministic case can be seen as a deterministic sample of the mixture model resulting from the uniform case, and so is unbiased given the population at the previous time-step [27].

In Algorithm 1 we introduce a very general sequential Monte-Carlo algorithm that includes both the usual particle filter and the IMMPF as special cases. Consider repeatedly importance sampling a series of target, $\mu(\hat{x}_t)$, and proposal, $\lambda_t(\hat{x}_t)$, distributions that may depend on the sample at the previous time-step. We refer to this algorithm as multiple sequential importance sampling (SMIS), since $\mu_{t\geq 1}$ and $\lambda_{t\geq 1}$ will be mixtures in all cases of interest.

Algorithm 1 Sequential Multiple Importance Sampling. All operations indexed by n should be repeated for all $n \in \{1, \ldots, N\}$. c is a, typically unknown, constant.

Input: proposal mixtures λ_0, λ_t time extent T particle count N

1: $\hat{x}_0^n \sim \lambda_0(\hat{x}_0^n)$;
2: $w_0^n \leftarrow \frac{\mu_0(\hat{x}_0^n)}{\lambda_0(\hat{x}_0^n)}$; $\mu_0(\hat{x}_0)$ is defined in Eq. (5).
3: $\bar{w}_0^n \leftarrow \frac{w_0^n}{\sum_{i=1}^N w_0^i}$;
4: **for** $t = 1$ **to** T **do**
5: $\quad \hat{x}_t^n \sim \lambda_t(\hat{x}_t^n)$
6: $\quad w_t^n \leftarrow c\frac{\mu_t(\hat{x}_t^n)}{\lambda_t(\hat{x}_t^n)}$; $\mu_{t\geq 1}(\hat{x}_t)$ is defined in Eq. (6).
7: $\quad \bar{w}_t^n \leftarrow \frac{w_t^n}{\sum_{i=1}^N w_t^i}$;
8: **end for**
9: **return** $\hat{x}_{0:T}^{1:N}$, $w_{0:T}^{1:N}$, $\bar{w}_{0,T}^{1:N}$.

For filtering, ideally the target is the posterior.

$$\mu_t(\hat{x}_t) = p(\hat{x}_t \mid \hat{y}_{0:t}) . \tag{5}$$

However, at $t > 0$ the true posterior is not typically available so we replace it by an empirical approximation:

$$\mu_{t\geq 1}(\hat{x}_t) \overset{\text{def}}{=} \frac{\hat{G}(\hat{y}_t \mid \hat{x}_t) \sum_{n=1}^N \bar{w}_{t-1}^n \hat{M}(\hat{x}_t \mid \hat{x}_{t-1}^n)}{p(\hat{y}_t \mid \hat{y}_{0:t-1})} . \tag{6}$$

We only consider SMIS algorithms with the target distribution given in Eqs. (5) and (6).

For example, we recover the generic bootstrap particle filter with

$$(\lambda_{\text{Boot}})_{t\geq 1}(\hat{x}_t) \overset{\text{def}}{=} \sum_{n=1}^N \bar{w}_{t-1}^n \hat{M}(\hat{x}_t \mid \hat{x}_{t-1}^n) , \tag{7}$$

and the IMMPF with uniform index sampling, for model (2), with

$$\begin{aligned}(\lambda_{\text{IMMPF}})_{t\geq 1}(\hat{x}_t) &= (\lambda_{\text{IMMPF}})_{t\geq 1}(x_t, k_t, r_t) \\ &\overset{\text{def}}{=} \frac{\sum_{n=1}^N \bar{w}_{t-1}^n K^\theta(k_t \mid r_{t-1}^n) M^\theta(x_t \mid x_{t-1}^n, k_t) \delta_{R^\theta(r_{t-1}^n, k_t)}(r_t)}{N_{\text{reg}} \sum_{l=1}^N \bar{w}_{t-1}^l K^\theta(k_t \mid r_{t-1}^l)} ,\end{aligned} \tag{8}$$

where $\delta_z(\cdot)$ is density of the Dirac measure at z. Note that Eq. (8) cannot be reduced to a mixture density over the particles, motivating the use of the more general SMIS framework.

Theorem 1. *Defining* $\mathcal{F}_t(\psi) \overset{\text{def}}{=} \sum_{n=1}^N \bar{w}_t^n \psi(\hat{x}_t^n)$, *and* $\mathbb{P}_t(\psi)$ *to be true posterior mean of some test function* $\psi : \mathcal{X} \to \mathbb{R}$, *respectively. Then under*

the sufficient set of assumptions that $|\psi|$ is upper bounded, μ_t is absolutely continuous with respect to λ_t, and the Radon–Nikodym derivative $\frac{\mu_t}{\lambda_t}$ is finitely upper bounded:

$$\mathcal{F}_t(\psi) \xrightarrow{L^2} \mathbb{P}_t(\psi)\,, \tag{9}$$

as $N \to \infty$, implying that $\mathcal{F}_t(\psi)$ is a weakly consistent estimator of $\mathbb{P}_t(\psi)$.

Proof. See Appendix B for a proof. □

We are not the first to formalise particle filtering as a SMIS procedure [28,29], see also [30] for a tutorial that introduces the well-known auxiliary particle filter [31] as an SMIS procedure. We extend Theorem 1 to an equivalent result for the deterministic regime sampling case in Corollary B.1.

3.2. Differentiable particle filtering

A diverse taxonomy of strategies exist to estimate the parameters of SSMs. We refer the readers to [18] for an overview of SMC approaches. Our problem differs from the classical cases in two respects: using neural networks, our parameter space is very high-dimensional; and with the flexibility we allow for, the latent system is not identifiable from the observations alone. With modern automatic differentiation, the ubiquitous-in-machine-learning, gradient-based schemes are an attractive choice. They perform well when the parameter space is high-dimensional and generalise readily to specialised loss functions. However, standard SMC algorithms are not differentiable.

Differentiable particle filter (DPF) refers, in the literature, to any SMC filtering algorithm that is designed to return estimates of the gradients of its outputs. The first DPF [15] used the well known reparameterisation trick to differentiate sampling from the proposal. But it did not pass gradients through resampling, setting them to zero, so that gradients are not propagated over time-steps.

In [32], the first fully smoothly differentiable particle filter is proposed. Their strategy is to find a differentiable transport map from the particle approximation of $p(x_t \mid y_{0:t-1})$ to $p(x_t \mid y_{0:t})$. This results in consistent gradient estimates. However, it is computationally costly and can suffer from numerical issues if not carefully tuned. We refer to this method as *optimal transport resampling* as the map chosen is an approximation of the entropy regularised Wasserstein-2 optimal map.

Several works [33–35] have explored applying REINFORCE [36] to differentiate through the discrete resampling steps. However all these papers opted to ignore the gradient due to REINFORCE due to high variance. In [37] it is proposed to apply REINFORCE separately to each resampled particle; however this is less generally applicable than optimal transport resampling. Its gradient estimates have been shown to be consistent [37,38] only for a restricted class of filters and functions of the filtering outputs.

The only prior work in differentiable filtering specifically concerned with regime-switching models is the RSDBPF [20]. The RSDBPF is a direct application of the biased DPF of [15] to the RSPF [9]. However, it does not attempt to learn the switching dynamic R^θ, K^θ and assumes these components are known a priori.

4. The differentiable interacting multiple model particle filter

The core algorithmic contribution of this work is to develop the IMMPF into a differentiable variant, the DIMMPF, so that it can be included into an end-to-end machine learning framework.

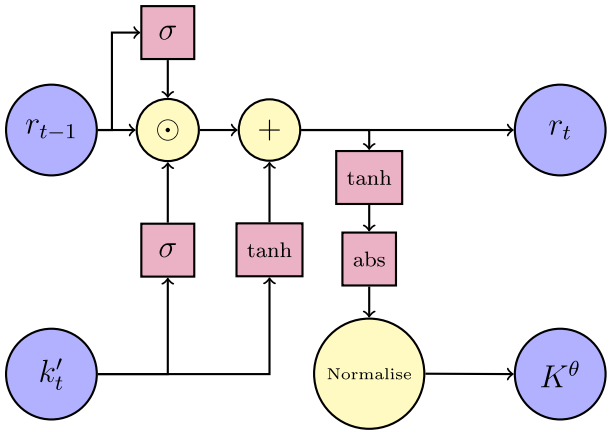

Fig. 2. Graphical representation of our proposed switching dynamic. Blue nodes are input/outputs. Purple nodes are fully connected network layers with the specified activation. Yellow nodes are non-learned functions. The switching probability mass, $K^\theta(k_{t+1} \mid r_t)$, is the value at the k_{t+1}th index of the model output K^θ.

4.1. Parameterising the model

For the dynamic and observation models, the choice is problem dependent, so we do not make any specific recommendations; the architectures used in our experiments can be found in Section 5.2. We take inspiration from long short term memory networks (LSTMs) [39] to design the parameterisation of R^θ. It is well known that SSMs, for which the latent process does not forget its past in some sense result in poorly performing SMC algorithms. Existing bounds on the stability of filtering algorithms typically rely on either conditions that lead to strong mixing [14], or a related drift condition [28]. Furthermore, there are SSMs that do not obey these assumptions for which it can be shown that their associated particle filter diverges exponentially (or worse) in mean squared error with T [13].

For this reason, we include forget gates in our parameterisation of the switching dynamic; *i.e.* we set $r_t = r_{t-1} \odot a + b$, where every element of vector a is between 0 and 1, and b is a function only of k_t. We desire that this has the effect of decreasing the weight of information from past states at each time-step before introducing information from the noisy k_t. Algebraically, the model can be expressed as,

$$\begin{aligned} r_t &= R^\theta(k_t, r_{t-1}) \\ &= \sigma(\Theta_1 r_{t-1}) \odot \sigma(\Theta_2 k'_t) \odot r_{t-1} + \tanh(\Theta_3 k'_t)\,, \end{aligned} \tag{10a}$$

$$r_0 = R_0^\theta(k_0) = R^\theta(k_0, \vec{0})\,, \tag{10b}$$

$$K'^\theta(k'_t \mid r_{t-1}) = |\Theta_4 \tanh(\Theta_5 r_{t-1})| \cdot k'_t\,, \tag{10c}$$

$$K^\theta(k_t \mid r_{t-1}) = \frac{K'^\theta_t(k'_t \mid r_{t-1})}{\sum_{c \in \mathcal{K}} K'^\theta(c' \mid r_{t-1})}\,, \tag{10d}$$

where k'_t is the one hot encoding of k_t, $\odot$ is the Hadamard product, and $\Theta_{1:5}$ are learned matrices, $\vec{0}$ is the zero vector, and $\sigma(\cdot)$ is the sigmoid function. K_0^θ is represented by a learned vector of regime probabilities (see Fig. 2).

4.2. Estimating the gradient of the DIMMPF

The large majority of DPFs proposed [15,16,32,40,41] take derivatives with respect to the proposal distribution's parameters by the low variance reparameterisation trick. However, for our case where the state space has a discrete component no explicit reparameterisation function exists. Implicit reparameterisations for mixture models [42] have been shown to perform poorly in particle filtering [41], so we avoid their usage in favour of REINFORCE. Furthermore, the model

index, k_t, is categorical rather than atomic so continuous relaxations such as the one used in [43] do not apply.

Under the IMMPF proposal it is natural to combine the model selection steps and the resampling steps. We apply REINFORCE separately to each resampled weight as done in [37,38] but use a reparameterisation to sample $M^\theta\left(x_t \mid x_{t-1}, k_t\right)$,

$$\tilde{w}_t^n = \sum_{m=1}^{N} \tilde{w}_{t-1}^m K^\theta\left(k_t^n \mid r_{t-1}^m\right) \perp\left[M^\theta\left(x_t^n \mid x_{t-1}^m, k_t^n\right)\right], \tag{11a}$$

$$w_t^n = \frac{\tilde{w}_t^n G^\theta\left(y_t \mid x_t^n, k_t^n\right) \perp\left[\sum_{l=1}^{N} \tilde{w}_{t-1}^l K^\theta\left(k_t^n \mid r_{t-1}^l\right)\right]}{N_{\text{reg}} \perp\left[\tilde{w}_t^n\right]}, \tag{11b}$$

where we use $\perp[\cdot]$ to denote the stop gradient operator, which is defined to be the operation that is the identity on the forward pass, but sets the gradient of the enclosed quantity to zero. In modern auto-differentiation libraries, this is easy to implement and computationally cheap by detaching the operand from the computation graph. We refer the reader to [44] for an expansive overview of Monte Carlo gradient estimation.

Theorem 2. *The weighting function Eqs. (11a) and (11b), reduces to the same value as the weights obtained using the λ_{IMMPF}, Eq. (8), in algorithm 1, on the forward pass.*

Proof. In the forward pass, Eq. (11b) simplifies to:

$$N_{\text{reg}} w_t^n = G^\theta\left(y_t^n \mid x_t^n, k_t^n\right) \sum_{l=1}^{N} \tilde{w}_{t-1}^l K^\theta\left(k_t^n \mid r_{t-1}^l\right). \tag{12}$$

Evaluating Eq. (8) for a given latent state gives:

$$\left(\lambda_{\text{IMMPF}}\right)\left(x_t^n, k_t^n, r_t^n\right) = \frac{\sum_{m=1}^{N} \tilde{w}_{t-1}^m K^\theta\left(k_t^n \mid r_{t-1}^m\right) M^\theta\left(x_t^n \mid x_{t-1}^m, k_t^n\right)}{N_{\text{reg}} \sum_{l=1}^{N} \tilde{w}_{t-1}^l K^\theta\left(k_t^n \mid r_{t-1}^l\right)}. \tag{13}$$

Then calculating w_n^t as in line 6 of Algorithm 1 from Eq. (13), gives Eq. (12). □

Theorem 3. *Under the weights given in Eq. (11); assuming the model components are such that the proposal dominates the target; and the magnitudes of w_t^n, $\nabla_\theta w_t^n$, ψ and $\nabla_\theta \psi$ are upper-bounded for all t then,*

$$\nabla_\theta \mathcal{F}_t(\psi) \xrightarrow{L^2} \nabla_\theta \mathbb{P}_t(\psi) \tag{14}$$

as $N \to \infty$, implying that $\nabla_\theta \mathcal{F}_t(\psi)$ is a weakly consistent estimator of $\nabla_\theta \mathbb{P}_t(\psi)$.

Proof. See Appendix C for a proof. □

Our weight function, Eq. (11), is not the unique estimator that applies REINFORCE separately to each resampled particle. We improve on the naïve choice,

$$\left(w_{\text{Naïve}}\right)_t^n = G^\theta\left(y_t \mid x_t^n, k_t^n\right) \frac{\tilde{w}_{t-1}^{a_t^n}}{\perp\left[\tilde{w}_{t-1}^{a_t^n}\right]} \perp\left[\sum_{l=1}^{N} \tilde{w}_{t-1}^l K^\theta\left(k_t^n \mid r_{t-1}^l\right)\right], \tag{15}$$

where a_t^n is the index of the particle at time $t-1$ that particle n at time t has been propagated from, by, at each time-step, partially Rao-Blackwellising over the previous time-step. Only the REINFORCE gradient terms benefit from this Rao-Blackwellisation, reparameterised gradients track only through the ancestral path. We find experimentally, in Table 1, that the Rao-Blackwellised estimator performs significantly better than its naïve counterpart. The naïve estimator's variance is such that it performs worse than the biased estimator that zeros out the REINFORCE terms.

The Rao-Blackwellisation comes at a computational cost; the density $M^\theta\left(x_t^n | x_{t-1}^m, k_t^n\right)$ must be computed for every pair $\{n, m\}$ at an operation cost of $\mathcal{O}\left(N^2\right)$ in the forward pass compared to the usual particle filter $\mathcal{O}(N)$ complexity. In practice these operations can be computed in parallel so the Rao-Blackwellised estimator, Eq. (11), is only slightly slower to compute than the naïve estimator, Eq. (15). Furthermore, by Theorem 2, the Rao-Blackwellisation only affects the gradient estimators, so we revert to the $\mathcal{O}(N)$ algorithm during inference. See Table 2 for our results.

We remark that the proof of Theorem 3, and its corollaries rely on the fact that, using the DIMMPF weights Eq. (11), gradients are only taken with respect to the model components and not the proposal. For this reason, one cannot expect our estimator to provide a useful learning signal for the proposal process. In [37] the authors investigate a closely related estimator and are unable to find clear theoretical or experimental evidence that it does. Under our set-up the proposal has no extra parameters in addition to the model. Developing a practical gradient estimator that can learn an efficient proposal process remains an open problem.

4.3. Training the DIMMPF

Consider two approaches to train a particle filter when there is access to the ground truth latent state during training. The first is to estimate the latent state and minimise some distance between the estimator and the ground truth. The second is to maximise the joint likelihood of the observations and their associated ground truth latent state. We find the best results are obtained when optimising a loss that combined the two strategies. A related strategy is recommended in [15] where the optimisation objective is a combination of the MSE of filtering estimates and a loss on the measurement and observation models individually.

In our case, the MSE of filtering estimates is obtained by Algorithm 2. To estimate the joint likelihood, we re-partition the model so that during training the available quantities $\{x_t, y_t\}$ are taken as observations and $\{r_t, k_t\}$ are the latent state estimated by filtering. Since the observations now depend on each other, this formulation does not strictly satisfy the usual definition of an SSM, described in Eq. (1). However, in particle filtering, the observations are treated as a sequence of non-random constants so the algorithm generalises freely to situations where the observations have arbitrary backwards-in-time interdependence. To account for this interdependence, we replace the conditional likelihood in Eq. (1), with

$$\hat{y}_t \sim \hat{G}\left(\hat{y}_t \mid \hat{x}_t, \hat{y}_{0:t-1}\right). \tag{16}$$

We provide precise details on how we calculate these losses in Appendix A.

Needing to run two filters incurs an extra computational cost. However, for the second filter the conditional likelihood only depends on the state through the discrete model index, so one can precompute all the conditional likelihoods for each choice of model index using GPU parallelism. Then the only neural network we need to evaluate per-particle during filtering is $R^\theta\left(k_t, r_{t-1}\right)$.

Formally, Theorem 3 does not prove the consistency of the gradients of either the losses we adopt, however the appropriate results follow as corollaries. See Corollaries C.1 and C.2 for formal statements and proofs of the consistency of the gradients of the MSE and log-likelihood respectively. When we calculate the log-likelihood of $\{x_t, y_t\}$ using the re-partitioned model, the state dynamics have no reparameterised component. In [37] it is shown, for this case, that the resultant gradient estimator is unbiased. It is closely related to the Rao-Blackwellised recursion obtained in [45].

4.4. The full algorithm and practical considerations

We present the full DIMMPF algorithm[3] in pseudo-code in Algorithm 2. It can be checked that the filtering loop of Algorithm 2 is

[3] Our implementation, including the code to run the experiments in this paper can be found at https://github.com/John-JoB/DIMMPF.

equivalent, in the forward pass, to running Algorithm 1 with target (6) and proposal (8), with the model indices deterministically chosen rather than sampled uniformly. When computing the data-likelihood, Algorithm 2 is modified such that the distribution of x_t is accounted for in the weights instead of being sampled from, see Appendix A for detail.

In practice, it is common to use a variance-reduced scheme to resample the particle indices in lieu of multinomial sampling. One such scheme is the systematic resampling of [46] and its closely related variants, which are found to work well empirically. The increased stability offered by systematic resampling is of increased importance in the context of DPFs, where it stabilises gradient updates as well as the forward pass [47]. However, under systematic resampling the particles are no longer sampled i.i.d., so the central limit theorem used to prove Theorem 1 no longer holds. We make the recommendation that practitioners use systematic resampling based on our empirical results.

Algorithm 2 Differentiable Interacting Multiple Model Particle Filter. All operations indexed by n should be repeated for all $n \in \{1, \dots, N\}$ and those by q for $q \in \{1, \dots, N_{\text{reg}}\}$.

Input: priors M_0^θ dynamic models M^θ
regime prior K_0^θ switching dynamic K^θ
observation models G^θ encoding functions R^θ
time length T particle count N
loss coefficient λ observations $y_{0:T}$
ground truth $\tilde{x}_{0:T}$ number of regimes N_{reg}
Output: unormalised weights $w_{0:T}^{1:N}$ particle locations $\{x_{0:T}^{1:N}, k_{0:T}^{1:N}, r_{0:T}^{1:N}\}$

$k_0^n \leftarrow \lfloor \frac{nN_{\text{reg}}}{N} \rfloor$
Sample $x_0^n \sim M_0^\theta\left(x_0^n \mid k_0^n\right)$
$r_0^n = R^\theta\left(k_0^n, \vec{0}\right)$
$w_0^n = G^\theta\left(y_0 \mid x_0^n, k_0^n\right)$
$\bar{w}_0^n = \frac{w_0^n}{\sum_{m=1}^N w_0^m}$
for $t = 1$ to T **do**
  $k_t^n \leftarrow \lfloor \frac{nN_{\text{reg}}}{N} \rfloor$
  $\hat{w}_t^{n,q} \leftarrow \frac{\bar{w}_{t-1}^n K^\theta\left(q \mid r_{t-1}^n\right)}{\sum_{m=1}^N \bar{w}_{t-1}^m K^\theta\left(q \mid r_{t-1}^m\right)}$
  Sample ancestor indices $a_t^n = m$ with probability equal to $\hat{w}_t^{m,k_t^n}$
  Sample $x_t^n \sim M^\theta\left(x_t^n \mid x_{t-1}^{a_t^n}, k_t^n\right)$
  $r_t^n \leftarrow R^\theta\left(k_t^n, r_{t-1}^{a_t^n}\right)$
  Calculate w_t^n from Eq. (11)
  $\bar{w}_t^n = \frac{w_t^n}{\sum_{m=1}^N w_t^m}$
end for
return $w_{0:T}^{1:N}, x_{0:T}^{1:N}, k_{0:T}^{1:N}, r_{0:T}^{1:N}$

5. Numerical experiments

In this section, we present the results from a set of numerical experiments.

5.1. Simulated environments

We repeat the test environment of [9,20], in which the dynamic and observation models of each regime are uni-variate and Gaussian.

$$M_0\left(x_0\right) = \mathcal{U}\left(-0.5, 0.5\right), \tag{17a}$$

$$M\left(x_t \mid x_{t-1}, k_t\right) = \mathcal{N}\left(a_{k_t} x_{t-1} + b_{k_t}, \sigma^2\right), \tag{17b}$$

$$G\left(y_t \mid x_t, k_t\right) = \mathcal{N}\left(a_{k_t} \sqrt{|x_t|} + b_{k_t}, \sigma^2\right), \tag{17c}$$

$$\left[a_1, \dots, a_8\right] = [-0.1, -0.3, -0.5, -0.9, 0.1, 0.3, 0.5, 0.9], \tag{17d}$$

$$\left[b_1, \dots, b_8\right] = [0, -2, 2, -4, 0, 2, -2, 4], \tag{17e}$$

$$\sigma^2 = 0.1. \tag{17f}$$

This model poses some challenges to state estimation. Because the observation location depends on the state only through its absolute value, it is impossible to estimate the state using the observations alone. Furthermore, the coefficients a_i, b_i are chosen so that it is hard to identify the current regime over short sequences, for example, regimes 1 and 5 have identical data-likelihoods when each is run in isolation. It is therefore required that all of the observation model, the dynamic model and the switching dynamic are well-learned.

We include three different switching dynamics. The first is a Markov switching system where the probability of remaining in the same regime is 0.8; switching to the next regime, with regimes 9 and 1 identified, is 0.15; and all other regimes have probability $\frac{1}{120}$. Algebraically:

$$K\left(k_t \mid k_{0:t-1}\right) = \left(\mathbf{k}'_{t-1}\right)^T B \mathbf{k}'_t, \tag{18a}$$

$$B = \begin{pmatrix} 0.8 & 0.15 & \rho & \dots & \rho \\ \rho & 0.8 & 0.15 & \dots & \rho \\ \vdots & & \ddots & & \vdots \\ \rho & \rho & \dots & 0.8 & 0.15 \\ 0.15 & \rho & \dots & \rho & 0.8 \end{pmatrix}, \tag{18b}$$

$$\rho = \frac{1}{120}, \tag{18c}$$

where k'_t are the one-hot encodings of the regime index.

In the second setting, the regimes follow a Pólya-urn distribution where the sampled regimes are more likely to appear at later time-steps:

$$K\left(k_t \mid k_{0:t-1}\right) = \frac{1 + \sum_{s=0}^{t-1} \mathbb{1}\left(k_s = k_t\right)}{8 + t}. \tag{19}$$

The Pólya-urn setting has frequent switching and often the distribution of model indices looks close to uniform, making it simpler to approximate but harder to perform inference on than the Markov setting. In both cases we set $K_0\left(k_0\right) = \frac{1}{8}$.

To demonstrate the versatility of our algorithm, we introduce a more challenging switching dynamic than has been used in previous work. For the third setting, the time between regime switches is approximately Erlang distributed. The order of the Erlang distribution is equal to the number of periods for which the system has been in the current regime. Once the Erlang distributed period is finished, the system jumps randomly to one of the two adjacent regimes. However, we include a small probability that at any time-step the system jumps to any regime. This system is most simply expressed algebraically as

$$m_t \sim \text{Bernoulli}\left(0.01\right), \tag{20a}$$

$$n_t \sim \text{Bernoulli}\left(0.2\right), \tag{20b}$$

$$c\left(k_{0:t}, k\right) = \sum_{s=0}^{t-1} \mathbb{1}\left(\left(k_s = k\right) \wedge \left(\neg\left(k_{s+1} = k\right) \vee \left(m_s = 1\right)\right)\right), \tag{20c}$$

$$l_t = \begin{cases} l_{t-1}, & n_t = 0, \\ l_{t-1} - 1, & \neg\left(l_{t-1} = 0\right) \wedge \left(n_t = 1\right), \\ c\left(k_{0:t}, k_t\right), & \left(l_{t-1} = 0\right) \wedge \left(n_t = 1\right), \end{cases} \tag{20d}$$

$$\alpha_t = \left(n_t = 1\right) \wedge \left(l_{t-1} = 0\right), \tag{20e}$$

$$K\left(k_t \mid k_{0:t-1}\right) = \begin{cases} \frac{1}{N_{\text{reg}}}, & m_t = 1, \\ \mathbb{1}\left(k_t = k_{t-1}\right), & \left(m_t = 0\right) \wedge \neg\alpha_t, \\ \begin{pmatrix} 0.6\mathbb{1}\left(k_t = k_{t-1}+1 \left(\text{mod } N_{\text{reg}}\right)\right) \\ +0.4\mathbb{1}\left(k_t = k_{t-1}-1 \left(\text{mod } N_{\text{reg}}\right)\right) \end{pmatrix}, & \left(m_t = 0\right) \wedge \alpha_t. \end{cases} \tag{20f}$$

We choose this dynamic because of its complexity to learn. There are both strong dependence between the index at successive time-steps, like the Markov setting; and long term dependencies, like the Pólya setting.

Table 1
Filtering accuracy for the discussed algorithms. Reported values are the achieved mean squared filtering error and averaged across 20 independent training runs.

Algorithm	Markov MSE	Pólya MSE	Erlang
Transformer (baseline)	1.579 ± 0.169	1.508 ± 0.112	1.614 ± 0.160
LSTM (baseline)	0.732 ± 0.083	0.667 ± 0.053	0.978 ± 0.103
RLPF (baseline)	0.536 ± 0.143	0.509 ± 0.071	0.771 ± 0.110
DIMMPF-OT (baseline)	0.891 ± 0.128	0.866 ± 0.134	0.873 ± 0.122
DIMMPF-N (baseline)	0.751 ± 0.0694	0.741 ± 0.071	0.742 ± 0.072
DIMMPF (ours)	**0.500 ± 0.100**	**0.490 ± 0.052**	**0.712 ± 0.115**
IMMPF (oracle)	0.274 ± 0.019	0.408 ± 0.014	0.473 ± 0.025

Table 2
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**
RLPF (baseline)	5.98	0.814
DIMMPF-OT (baseline)	425	Out of memory
DIMMPF-N (baseline)	8.56	0.773
DIMMPF (Ours)	10.5	0.759

5.2. Experiment details

In addition to the our DIMMPF, we present a number of baseline approaches. The problem of sequential state estimation can be described as learning to predict a sequence of latent states from a sequence of observations, so any available sequence-to-sequence techniques can apply. We choose to use a transformer [48] and an LSTM [39] as baseline approaches as they represent the state-of-the-art in sequence-to-sequence prediction. The transformer is encoder only and the LSTM is unidirectional so that only past information is used. We also compare to the regime learning particle filter (RLPF), a preliminary version of our methodology that we presented in the conference paper [23]. Finally we include two variants on the DIMMPF: the DIMMPF-OT that uses a transport map based resampler [32] to be differentiable, instead of the gradient estimator developed in Section 4.2; and the DIMMPF-N that uses the $\mathcal{O}(N)$ naïve gradient estimator, Eq. (15).

All filtering based models parameterise both the measurement and dynamic models with fully connected neural networks of two hidden layers containing 11 nodes each. During training we use a population of 200 total particles, which we increase to 2000 for testing. This is reduced to 80 particles in training and 800 in testing for the DIMMPF-OT due to memory constraints. We generate 2000 trajectories of 51 time-steps and use them in ratio $2:1:1$ for training, validation and testing, respectively. We train in mini-batches of 100 trajectories, but test on the full 500 trajectory batches. Each experiment is repeated 20 times with independent data generations. All experiments are performed using an NVIDA RTX 3090 GPU.

5.3. Results

We present the main results in Table 1, and the computation times in Table 2. The DIMMPF is the best performing algorithm in all experiments. The filtering approaches far outperform the generic sequence-to-sequence techniques in mean accuracy, however, the LSTM is computationally the cheapest. In training, the DIMMPF is faster than the DIMMPF-OT. But, it is slower than the DIMMPF-N due requiring more terms to be computed. The RLPF further saves time through ignoring gradient terms that the DIMMPF and DIMMPF-N evaluate. During inference, the DIMMPF and DIMMPF-N are equivalent so achieve similar timings.

6. Conclusions

In this paper, we have presented a novel differentiable particle filter, the DIMMPF, that addresses the problem of learning to estimate the state of a regime switching state space process. Our algorithm improves over the previous state-of-the-art, the RSDBPF, in three respects. Firstly, the RSDBPF required that the switching dynamic be fully known a priori, whereas our algorithm can learn it from data. Secondly, the DIMMPF takes account of the assigned regime when resampling particles, thereby concentrating computation on more promising regions. Thirdly, the gradient estimates returned by the DIMMPF are consistent.

We evaluated our algorithm on a set of numerical experiments. The three settings, Markov, Pólya, and Erlang are designed to test the learning of short-term strong dependency, long-range weak dependency, and both simultaneously, respectively. The proposed DIMMPF leads to the smallest filtering errors on all three settings. The DIMMPF is computationally expensive during training, both compared to its simpler variants and especially out-of-the-box sequence-to-sequence techniques. However, during inference it achieves a similar speed to the other DPF approaches.

The key limitations of this work are that the DIMMPF is only capable of learning model parameters and not, additionally, an efficient proposal process; and that the number of regimes must be known. We leave addressing these limitations to future work.

Another important direction for future work is towards more challenging environments, including real-world data. We propose a simple architecture to parameterise the switching dynamic; future work might consider more advanced design patterns such as attention.

CRediT authorship contribution statement

John-Joseph Brady: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Yuhui Luo:** Writing – review & editing, Supervision. **Wenwu Wang:** Writing – review & editing, Supervision. **Víctor Elvira:** Writing – review & editing. **Yunpeng Li:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge funding from the UK Department for Science, Innovation and Technology through the 2024 National Measurement System programme. JJ. Brady was supported in this work through the National Physical Laboratory and University of Surrey partnership via an Engineering and Physical Sciences Research Council studentship. The work of V. E. is supported by ARL/ARO under grant W911NF-22-1-0235.

Appendix A. Calculating the losses

Mean squared error loss: The mean squared error of the filtering estimates is calculated as follows, with $\tilde{x}_t$ denoting the ground truth latent state.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T+1}\sum_{t=0}^{T}\left(\sum_{n=1}^{N}\bar{w}_t^n\left(x_t^n - \tilde{x}_t\right)\right)^2. \tag{A.1}$$

Data-likelihood loss: Calculating the data-likelihood requires a modification to the inputs of Algorithm 2. We replace the observation model,

$G^\theta\left(y_t \mid x_t^n, k_t^n\right)$, with the marginal data-likelihood conditioned on the previous time-step,

$$G^\theta\left(y_t \mid \tilde{x}_t, k_t^n\right) M^\theta\left(\tilde{x}_t \mid \tilde{x}_{t-1}, k_t^n\right). \tag{A.2}$$

In practice we do not track a latent state x_t, but the same effect can be achieved in Algorithm 2, by setting $M^\theta\left(x_t|x_{t-1}, k_t\right)$ to be uniform over some set that does not depend on x_{t-1} or k_t. Then the negative log data-likelihood is estimated as

$$\mathcal{L}_{\text{data-likelihood}} = -\sum_{t=0}^{T} \log \sum_{n=1}^{N} w_T^n - (T+1)\log N. \tag{A.3}$$

Appendix B. Proof of Theorem 1

The following proof is based, in part, on the proofs found in Chapters 8 and 11 of [13].

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathcal{F}_t(\psi) - \mathbb{P}_t(\psi)\right)^2\right] \\
&= \mathbb{E}\left[\left(\mathcal{F}_t(\psi) - \int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right) + \int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right) - \mathbb{P}_t(\psi)\right)^2\right] \\
&\leq 2\left\{\mathbb{E}\left[\left(\mathcal{F}_t(\psi) - \int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right)\right)^2\right] + \mathbb{E}\left[\left(\int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right) - \mathbb{P}_t(\psi)\right)^2\right]\right\}.
\end{aligned} \tag{B.1}$$

The first term can be bounded using the standard convergence result for an auto-normalised importance sampler. For some factor c_t that is independent of N,

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathcal{F}_t(\psi) - \int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right)\right)^2\right] \\
&\leq \mathbb{E}\left[\left\|\left(\mathcal{F}_t(\psi) - \int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right)\right)\right\|_\infty^2 \mid \hat{x}_{t-1}^{1:N}, w_{t-1}^{1:N}\right] \leq \frac{1}{N} c_t,
\end{aligned} \tag{B.2}$$

where we have used the fact that the particles are conditionally i.i.d. given the particles at the previous time-step. For $t = 0$, the second term in Eq. (B.1) is zero, so MSE convergence is guaranteed *i.e.*Theorem 1 holds for $t = 0$. For $t > 0$, we may bound the MSE by induction with $t = 0$ as the base case. To perform the inductive step we first introduce the identity:

$$\mathbb{P}_{t-1}\left(\int_{\mathcal{X}} \psi\left(\hat{x}_t\right) p\left(d\hat{x}_t \mid \hat{x}_{t-1}, \hat{y}_t\right)\right) = \mathbb{P}_t(\psi), \tag{B.3}$$

which may be proved using basic probability rules and the conditional independence structure of the SSM (Eq. (1)). One may write:

$$\begin{aligned}
\int_{\mathcal{X}} \mu_t\left(d\hat{x}_t\right)\psi\left(\hat{x}_t\right) &= \sum_{i=1}^{N} \bar{w}_{t-1}^i \int_{\mathcal{X}} \psi\left(\hat{x}_t\right) p\left(d\hat{x}_t \mid \hat{x}_{t-1}^i, \hat{y}_{0:t}\right) \\
&= \mathcal{F}_{t-1}\left(\int_{\mathcal{X}} \psi\left(\hat{x}_t\right) p\left(d\hat{x}_t \mid \hat{x}_{t-1}, \hat{y}_t\right)\right).
\end{aligned} \tag{B.4}$$

This implies, using Eq. (B.3), that the second term in (B.1) is bounded by a term of order N^{-1} assuming that Theorem 1 holds for $t-1$. Then, the MSE becomes the sum of two bounded by order N^{-1} terms. So, it is bounded by $\frac{1}{N} c_t'$ for some constant c_t' and therefore the MSE converges to zero in the large sample size limit.

Corollary B.1. *The IMMPF with deterministic sampling exhibits L^2 convergence under the same conditions as Theorem 1,*

$$\left(\mathcal{F}_{Det}\right)_t(\psi) \xrightarrow{L^2} \mathbb{P}_t(\psi), \tag{B.5}$$

implying the weak consistency of $\left(\mathcal{F}_{Det}\right)_t(\psi)$.

In the deterministic case the particles are no longer i.i.d given the previous particles, so we can no longer apply the convergence result for the auto-normalised importance sampler to obtain Eq. (B.2).

The IMMPF sampling algorithm is equivalent to scheme N1 in [27] where each sampling distribution in repeated for $\frac{N}{N_{\text{reg}}}$ copies. Following the derivation in Appendix C, Option 3 of [27] we can see this sampling stratergy is unbiased. Furthermore, they give the variance of the unnormalised sum of weighted samples for a zero mean quantity,

$$\hat{\psi} \overset{\text{def}}{=} \sum_{n=1}^{N} w\left(x^n\right)\psi\left(x^n\right), \tag{B.6}$$

to be

$$\text{Var}\left[\hat{\psi}\right] = \frac{N}{N_{\text{reg}}} \sum_{k=1}^{N_{\text{reg}}} \mathbb{E}\left[\left(w(x)\psi(x)\right)^2 |k\right] \leq \|w\|_\infty^2 \|\psi\|_\infty^2. \tag{B.7}$$

Then by Slutsky's theorem, the asymptotic variance of the proceeding auto-normalised importance sampler is

$$\text{Var}\left[\frac{\hat{\psi}}{\sum_{m=1}^{N} w\left(x^m\right)}\right] \to \frac{\mathbb{E}\left[\left(w\left(x^k\right)\psi\left(x^k\right)\right)^2\right]}{N\mathbb{E}\left[w(x)\right]^2}. \tag{B.8}$$

Which is exactly the same asymptotic variance one obtains using uniform sampling. This convergence result can be substituted for that of the usual auto-normalised importance sampler in (B.2) and the rest of the proof is identical to that of Theorem 1.

We remark that although the asymptotic variances of the auto-normalised importance samplers for the deterministic, Eq. (B.8), and uniform cases are identical, this does not imply the variances of the filtering estimates are.

Appendix C. Proof of Theorem 3

This result and the attached corollaries apply to the deterministic sampling case by applying Corollary B.1 wherever their proofs use Theorem 1 for the uniform case. Throughout this derivation, if the distribution an expectation is taken over is not specified it may be assumed to be the full posterior $p\left(x_T, k_T, r_T|y_{0:T}\right)$.

Consider generating $N \to \infty$ trajectories from the bootstrap particle filter with, Eq. (7), for our system, Eq. (2), with $M^\theta\left(x_t|x_{t-1}\right)$ sampled by the reparameterisation trick. Using the surrogate objectives of [49], we can directly write the appropriate gradient estimator for this stochastic program as:

$$\begin{aligned}
&\nabla_\theta \mathop{\mathbb{E}}_{\text{Boot}}\left[\sum_{n=1}^{N} \Omega_T^n \psi\left(x_T^n\right)\right] = \mathop{\mathbb{E}}_{\text{Boot}}\left[\nabla_\theta \sum_{n=1}^{N} \Omega_T^n \psi\left(x_T^n\right)\right. \\
&\left. + \sum_{n=1}^{N} \Omega_T^n \psi\left(x_t^n\right) \nabla_\theta \sum_{m=1}^{N}\left(\sum_{t=0}^{T-1}\left[\log \Omega_t^m + \log K^\theta\left(k_{t+1}^m | r_t^{a_t^m}\right)\right] + \log K_0^\theta\left(k_0^m\right)\right)\right],
\end{aligned} \tag{C.1}$$

for ancestor variables $a_{0:T-1}^{0:N}$ such that a_t^n is the index at time t that particle n at time $t+1$ descends from and

$$\Omega_t^n \overset{\text{def}}{=} \frac{G^\theta\left(y_t|x_t^n, k_t^n\right)}{\sum_{l=1}^{N} G^\theta\left(y_t|x_t^l, k_t^l\right)}. \tag{C.2}$$

Taking the limit $N \to \infty$ and applying the consistency result of Theorem 1 then:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[\psi\left(x_t\right)\right] = \mathbb{E}\left[\nabla_\theta \psi\left(x_t\right) + \psi\left(x_t\right)\nabla_\theta\left(\log K_0^\theta\left(k_0\right) - \log p\left(y_{0:T}\right) + \right.\right. \\
&\left.\left.\sum_{t=0}^{T}\left[\log G^\theta\left(y_t|x_t, k_t\right) + \mathbb{1}_{t\geq 1} \log K^\theta\left(k_t, r_{t-1}\right)\right]\right)\right],
\end{aligned} \tag{C.3}$$

where $\mathbb{1}_{t\geq 1}$ is defined to be equal to 0 for $t = 0$ and 1 otherwise. Since our proposal Eq. (8) contains no additional parameters, if our gradient estimator is asymptotically equal to Eq. (C.3), then by Theorem 1 it is consistent. Applying Theorems 1 and 2,

$$\nabla_\theta \sum_{n=1}^{N} \bar{w}_T^n \psi\left(x_T^n\right) \xrightarrow{L^2} \mathop{\mathbb{E}}_{\hat{x}_T \sim p\left(\hat{x}_T|y_{0:T}\right)}\left[\nabla_\theta \psi\left(x_T\right) + \psi\left(x_t\right)\nabla_\theta \log\left(\bar{w}_T\right)\right]. \tag{C.4}$$

Defining $Z_T \overset{\text{def}}{=} \sum_{t=0}^{T} \log \sum_{n=1}^{N} w_t^n - (T+1)\log N$,

$$\begin{aligned}&\mathbb{E}_{\hat{x}_T \sim p(\hat{x}_T|y_{0:T})}\left[\psi\left(x_T\right)\nabla_\theta \log \bar{w}_T\right] = \\ &-\mathbb{E}\left[\psi\left(x_T\right)\right]\nabla_\theta\left(Z_T - Z_{T-1}\right) + \mathbb{E}_{\hat{x}_T \sim p(\hat{x}_T|y_{0:T})}\left[\psi\left(x_T\right)\nabla_\theta\left(\log G^\theta\left(y_T \mid x_T, k_T\right)\right.\right. \\ &\left.\left.+\frac{1}{p(\hat{x}_T|y_{0:T-1})}\mathbb{E}_{\hat{x}_{T-1}\sim p(\hat{x}_{T-1}|y_{0:T-1})}\left[p\left(\hat{x}_T \mid \hat{x}_{T-1}\right)\left(\log \bar{w}_{T-1} + \log K^\theta\left(k_T \mid r_{T-1}\right)\right)\right]\right)\right]\end{aligned} \tag{C.5}$$

$$= -\mathbb{E}\left[\psi\left(x_T\right)\right]\nabla_\theta Z_T + \mathbb{E}\left[\psi\left(x_T\right)\nabla_\theta\left(\log K_0^\theta\left(k_0\right) + \sum_{t=0}^{T}\log G^\theta\left(y_t|x_t,k_t\right) + \mathbb{1}_{t\geq 1}\log K^\theta\left(k_t|r_{t-1}\right)\right)\right], \tag{C.6}$$

where we have derived the final expression by a recursive application of Slutsky's theorem. Note that we cannot marginalise over the past time-steps in the outer expectation since x_T is a function of all $x_{0:T-1}$.

$$\nabla_\theta Z_T \xrightarrow{L^2} \mathbb{E}\left[\nabla_\theta\left(\log K_0^\theta\left(k_0\right) + \sum_{t=0}^{T}\log G^\theta\left(y_t|x_t,k_t\right) + \mathbb{1}_{t\geq 1}\log K^\theta\left(k_t|r_{t-1}\right)\right)\right] \tag{C.7}$$

$$= \nabla_\theta \log p\left(y_{0:T}\right), . \tag{C.8}$$

Combining (C.8), (C.4) and (C.6) we directly obtain (C.3) so our gradient estimator is consistent. We have proved that estimates of gradients of expectations for bounded functions, by Theorem 1, converges in the L^2 sense to the true expectation under the posterior.

Corollary C.1. *Under the same assumptions as Theorem 3 the DIMMPF estimate of the gradient of the MSE of a bounded function is weakly consistent. Precisely, for a true state $\tilde{x}_T$,*

$$\nabla_\theta\left(\sum_{n=1}^{N}\bar{w}_T^n\psi\left(x_T^n\right) - \psi\left(\tilde{x}_T\right)\right)^2 \xrightarrow{L^2} \nabla_\theta\left(\mathbb{E}\left[\psi\left(\tilde{x}_T\right)\right] - \psi\left(\tilde{x}_T\right)\right)^2. \tag{C.9}$$

This result is trivially extended to an average over time-steps and batched trajectories due to the linearity of the gradient operator.

Proof. Define $\tilde{\psi}\left(x_T^n\right) \overset{\text{def}}{=} \psi\left(x_T^n\right) - \psi\left(\tilde{x}_T\right)$.

$$\nabla_\theta\left(\sum_{n=1}^{N}\bar{w}_T^n\psi\left(x_T^n\right) - \psi\left(\tilde{x}_T\right)\right)^2 = 2\left(\sum_{n=1}^{N}\bar{w}_T^n\tilde{\psi}\left(x_t^n\right)\right)\nabla_\theta\left(\sum_{n=1}^{N}\bar{w}_T^n\tilde{\psi}\left(x_T^n\right)\right). \tag{C.10}$$

Applying Theorems 1 and 3, as well as Slutsky's theorem we therefore have:

$$\begin{aligned}&\nabla_\theta\left(\sum_{n=1}^{N}\bar{w}_t^n\psi\left(x_T^n\right) - \psi\left(\tilde{x}_T\right)\right)^2 \xrightarrow{L^2} 2\mathbb{E}\left[\tilde{\psi}\left(x_T^n\right)\right]\nabla_\theta\mathbb{E}\left[\tilde{\psi}\left(x_T^n\right)\right] \\ &= \nabla_\theta\left(\mathbb{E}\left[\psi\left(x_T\right)\right] - \psi\left(\tilde{x}_T\right)\right)^2\end{aligned} \tag{C.11}$$

For our case, Eq. (A.1), ψ is the identity function, which is not bounded. So, formally we have not proved convergence. However, one could instead take ψ to be an arbitrarily wide rectangular function for a formally consistent estimator. □

Corollary C.2. *When the weights and their gradient are upper-bounded, the estimate of the gradient of the log-likelihood, is consistent. I.e.*

$$\sum_{t=0}^{T}\nabla_\theta Z_T \xrightarrow{L^2} \nabla_\theta \log p\left(y_{0:T}\right). \tag{C.12}$$

Proof. This result is directly implied by Eq. (C.8). For the data-likelihood loss, $\sum_{t=0}^{T}\nabla_\theta Z_T = -\nabla_\theta \mathcal{L}_{\text{data-likelihood}}$, by Eq. (A.3). □

Data availability

No data was used. The code is may be accessed on Github from a link in the paper.

References

[1] M. Guidolin, Missing Data Methods: Time-Series Methods and Applications, Emerald Group Publishing Ltd., 2011.

[2] J.D. Hamilton, A new approach to the economic analysis of nonstationary time series and the business cycle, Econometrica 57 (2) (1989) 357–384.

[3] S. McGinnity, G. Irwin, Multiple model bootstrap filter for maneuvering target tracking, IEEE Tran. Aero. Electr. Syst. 36 (3) (2000) 1006–1012.

[4] H.A. Blom, An efficient decision-making-free filter for processes with abrupt changes, in: Proc. IFAC/IFORS Symp. on Ident. and Syst. Param. Est., York, UK, 1985.

[5] H.A. Blom, E.A. Bloem, Exact Bayesian and particle filtering of stochastic hybrid systems, IEEE Tran. Aero. Elect. Syst. 43 (1) (2007) 55–70.

[6] Y. Boers, J. Driessen, Interacting multiple model particle filter, IEE Proc. Radar Sonar Nav. 150 (2003) 344–349.

[7] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE Proc. F Radar Sig. Process. 140 (2) (1993) 107–113.

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

[9] Y. El-Laham, L. Yang, P.M. Djurić, M.F. Bugallo, Particle filtering under general regime switching, in: Proc. Euro. Sig. Process. Conf., EUSIPCO, Amsterdam, NL, 2021, pp. 2378–2382.

[10] A. Doucet, S. Sénécal, Fixed-lag sequential Monte Carlo, in: Proc. Europe. Sig. Process. Conf., EUSIPCO, Vienna, Austria, 2004, pp. 861–864.

[11] N.J. Gordon, S. Maskell, T. Kirubarajan, Efficient particle filters for joint tracking and classification, in: Sig. Data Process. Small Targets, 2002, pp. 439–449.

[12] D. Crisan, A. Doucet, A survey of convergence results on particle filtering methods for practitioners, IEEE Tran. Sig. Process. 50 (3) (2002) 736–746.

[13] N. Chopin, O. Papaspiliopoulos, An Introduction To Sequential Monte Carlo, Springer International Publishing, New York, USA, 2020.

[14] P. Del Moral, Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer, New York, NY, USA, 2004.

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

[16] P. Karkus, D. Hsu, W.S. Lee, Particle filter networks with application to visual localization, in: Proc. Conf. Robot Learn., PMLR, Zurich, CH, 2018, pp. 169–178.

[17] X. Chen, Y. Li, An overview of differentiable particle filters for data-adaptive sequential Bayesian inference, Found. Data Sci. (2023).

[18] N. Kantas, A. Doucet, S. Singh, J. Maciejowski, An overview of sequential Monte Carlo methods for parameter estimation in general state-space models, in: IFAC Proc. Volumes, vol. 42, (10) Saint-Malo, FR, 2009, pp. 774–785.

[19] C. Andrieu, A. Doucet, R. Holenstein, Particle markov chain monte carlo methods, J. R. Stat. Soc. Ser. B: Stat. Method. 72 (3) (2010).

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

[21] I. Urteaga, M.F. Bugallo, P.M. Djurić, Sequential Monte Carlo methods under model uncertainty, in: Proc. IEEE Stat. Sig. Process. Workshop, SSP, Palma de Mallorca, Spain, 2016, pp. 1–5.

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

[23] J.-J. Brady, Y. Luo, W. Wang, V. Elvira, Y. Li, Regime learning for differentiable particle filters, 2024, ArXiv:2405.04865.

[24] S. Mcginnity, G. Irwin, A multiple model extension to the bootstrap filter for manoeuvring targets, in: IFAC Proc. Workshop Alg. Arch. Real Time Contr., AARTC, Cancun, Mexico, 1998, pp. 23–28.

[25] E. Chouzenoux, V. Elvira, Graphical inference in non-Markovian linear-Gaussian state-space models, in: Proc. IEEE Int. Conf. Acoustics, Speech Sig. Process, ICASSP, Seoul, South Korea, 2024.

[26] C. Andrieu, A. Doucet, V. Tadic, On-line parameter estimation in general state-space models, in: Proc. Conf. Decision Control, Seville, Spain, 2005, pp. 332–337.

[27] V. Elvira, L. Martino, D. Luengo, M.F. Bugallo, Generalized multiple importance sampling, Stat. Sci. 34 (1) (2019) 129–155.

[28] N. Whiteley, Stability properties of some particle filters, Ann. App. Prob. 23 (6) (2013) 2500–2537.

[29] V. Elvira, L. Martino, M.F. Bugallo, P.M. Djurić, In search for improved auxiliary particle filters, in: Europe. Sig. Process. Conf., EUSIPCO, IEEE, Rome, Italy, 2018, pp. 1637–1641, http://dx.doi.org/10.23919/EUSIPCO.2018.8553361.

[30] V. Elvira, L. Martino, M.F. Bugallo, P.M. Djurić, Elucidating the auxiliary particle filter via multiple importance sampling, IEEE Sig. Process. Mag. 36 (6) (2019) 145–152.
[31] M.K. Pitt, N. Shephard, Filtering via simulation: Auxiliary particle filters, J. Am. Stat. Assoc. 94 (446) (1999) 590–599.
[32] A. Corenflos, J. Thornton, G. Deligiannidis, A. Doucet, Differentiable particle filtering via entropy-regularized optimal transport, in: Proc. Int. Conf. on Machine Learn., ICML, Online, 2021, pp. 2100–2111.
[33] C. Naesseth, S. Linderman, R. Ranganath, D. Blei, Variational sequential monte carlo, in: Proc. Int. Conf. Art. Int. and Stat., AISTATS, PMLR, Lanzarote, Canary Islands, 2018, pp. 968–977.
[34] T. Le, M. Igl, T. Rainforth, T. Jin, F. Wood, Auto-encoding sequential Monte Carlo, in: Proc. Int. Conf. Learn Represent., ICLR, Vancouver, Canada, 2018.
[35] C.J. Maddison, et al., Filtering variational objectives, in: Proc. Adv. in Neural Info. Process. Syst., NeurIPS, Long Beach, CA, USA, 2017.
[36] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.
[37] A. Ścibior, F. Wood, Differentiable particle filtering without modifying the forward pass, 2021, ArXiv:2106.10314.
[38] G. Arya, M. Schauer, F. Schäfer, C. Rackauckas, Automatic differentiation of programs with discrete randomness, 35, 2022, pp. 10435–10447, Vol. 35, New Orleans, LA, USA.
[39] H. Sak, A. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014, arXiv: 1402.1128.
[40] X. Chen, Y. Li, Normalising flow-based differentiable particle filters, 2024, ArXiv: 2403.01499.
[41] A. Younis, E. Sudderth, Differentiable and stable long-range tracking of multiple posterior modes, NeurIPS, in: Proc. Adv. Neural Inf. Process. Syst., vol. 36, New Orleans, LA, USA, 2023.
[42] A. Graves, Stochastic backpropagation through mixture density distributions, 2016, arXiv preprint arXiv:1607.05690.
[43] B. Cox, et al., End-to-end learning of Gaussian mixture proposals using differentiable particle filters and neural networks, in: IEEE Int. Conf. on Acoustics, Speech and Sig. Process., ICASSP, Seoul, South Korea, 2024, pp. 9701–9705.
[44] S. Mohamed, M. Rosca, M. Figurnov, A. Mnih, Monte carlo gradient estimation in machine learning, J. Mach. Learn. Res. 21 (132) (2020).
[45] G. Poyiadjis, A. Doucet, S.S. Singh, Particle approximations of the score and observed information matrix in state space models with application to parameter estimation, Biometrika 98 (1) (2011) 65–80.
[46] J. Carpenter, P. Clifford, P. Fearnhead, Improved particle filter for nonlinear problems, in: IEE Proc. Radar, Sonar and Navi., vol. 146, 1999.
[47] M. Zhu, K.P. Murphy, R. Jonschkowski, Towards differentiable resampling, 2020, arXiv:2004.11938.
[48] A. Vaswani, et al., Attention is all you need, in: Proc. Adv. in Neural Info. Process. Syst., NeurIPS, Long Beach, CA, USA, 2017.
[49] J. Schulman, N. Heess, T. Weber, P. Abbeel, Gradient estimation using stochastic computation graphs, NeurIPS, in: Proc. Adv. in Neural Info. Process. Syst., vol. 28, Montréal, Canda, 2015.