

Differentiable Particle Filtering via Entropy-Regularized Optimal Transport

Adrien Corenflos^{*1} James Thornton^{*2} George Deligiannidis² Arnaud Doucet²

Abstract

Particle Filtering (PF) methods are an established class of procedures for performing inference in non-linear state-space models. Resampling is a key ingredient of PF, necessary to obtain low variance likelihood and states estimates. However, traditional resampling methods result in PF-based loss functions being non-differentiable with respect to model and PF parameters. In a variational inference context, resampling also yields high variance gradient estimates of the PF-based evidence lower bound. By leveraging optimal transport ideas, we introduce a principled differentiable particle filter and provide convergence results. We demonstrate this novel method on a variety of applications.

1. Introduction

In this section we provide a brief introduction to state-space models (SSMs) and PF methods. We then illustrate one of the well-known limitations of PF (Kantas et al., 2015): resampling steps are required in order to compute low-variance estimates, but these estimates are not differentiable w.r.t. to model and PF parameters. This hinders end-to-end training. We discuss recent approaches to address this problem in econometrics, statistics and machine learning (ML), outline their limitations and our contributions.

1.1. State-Space Models

SSMs are an expressive class of sequential models, used in numerous scientific domains including econometrics, ecology, ML and robotics; see e.g. (Chopin & Papaspiliopoulos, 2020; Douc et al., 2014; Doucet & Lee, 2018; Kitagawa & Gersch, 1996; Lindsten & Schön, 2013; Thrun et al., 2005). SSM may be characterized by a latent \mathcal{X} -valued Markov

^{*}Equal contribution ¹Department of Electrical Engineering and Automation, Aalto University ²Department of Statistics, University of Oxford. Correspondence to: Adrien Corenflos <adrien.corenflos@aalto.fi>, James Thornton <james.thornton@spc.ox.ac.uk>.

process $(X_t)_{t \geq 1}$ and \mathcal{Y} -valued observations $(Y_t)_{t \geq 1}$ satisfying $X_1 \sim \mu_\theta(\cdot)$ and for $t \geq 1$

$$X_{t+1} | \{X_t = x\} \sim f_\theta(\cdot|x), \quad Y_t | \{X_t = x\} \sim g_\theta(\cdot|x), \quad (1)$$

where $\theta \in \Theta$ is a parameter of interest. Given observations $(y_t)_{t \geq 1}$ and parameter values θ , one may perform state inference at time t by computing the posterior of X_t given $y_{1:t} := (y_1, \dots, y_t)$ where

$$\begin{aligned} p_\theta(x_t | y_{1:t-1}) &= \int f_\theta(x_t | x_{t-1}) p_\theta(x_{t-1} | y_{1:t-1}) dx_{t-1}, \\ p_\theta(x_t | y_{1:t}) &= \frac{g_\theta(y_t | x_t) p_\theta(x_t | y_{1:t-1})}{\int g_\theta(y_t | x_t) p_\theta(x_t | y_{1:t-1}) dx_t}, \end{aligned}$$

with $p_\theta(x_1 | y_0) := \mu_\theta(x_1)$.

The log-likelihood $\ell(\theta) = \log p_\theta(y_{1:T})$ is then given by

$$\ell(\theta) = \sum_{t=1}^T \log p_\theta(y_t | y_{1:t-1}),$$

with $p_\theta(y_1 | y_0) := \int g_\theta(y_1 | x_1) \mu_\theta(x_1) dx_1$ and for $t \geq 2$

$$p_\theta(y_t | y_{1:t-1}) = \int g_\theta(y_t | x_t) p_\theta(x_t | y_{1:t-1}) dx_t.$$

The posteriors $p_\theta(x_t | y_{1:t})$ and log-likelihood $p_\theta(y_{1:T})$ are available analytically for only a very restricted class of SSM such as linear Gaussian models. For non-linear SSM, PF provides approximations of such quantities.

1.2. Particle Filtering

PF are Monte Carlo methods entailing the propagation of N weighted particles $(w_t^i, X_t^i)_{i \in [N]}$, here $[N] := \{1, \dots, N\}$, over time to approximate the filtering distributions $p_\theta(x_t | y_{1:t})$ and log-likelihood $\ell(\theta)$. Here $X_t^i \in \mathcal{X}$ denotes the value of the i^{th} particle at time t and $\mathbf{w}_t := (w_t^1, \dots, w_t^N)$ are weights satisfying $w_t^i \geq 0$, $\sum_{i=1}^N w_t^i = 1$. Unlike variational methods, PF methods provide consistent approximations under weak assumptions as $N \rightarrow \infty$ (Del Moral, 2004). In the general setting, particles are sampled according to proposal distributions $q_\phi(x_1 | y_1)$ at time $t = 1$ and $q_\phi(x_t | x_{t-1}, y_t)$ at time $t \geq 2$ prior to weighting and resampling. One often chooses $\theta = \phi$ but this is not necessarily the case (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018).

Algorithm 1 Standard Particle Filter

```

1: Sample  $X_1^i \stackrel{\text{i.i.d.}}{\sim} q_\phi(\cdot | y_1)$  for  $i \in [N]$ 
2: Compute  $\omega_1^i = \frac{p_\theta(X_1^i, y_1)}{q_\phi(X_1^i | y_1)}$  for  $i \in [N]$ 
3:  $\hat{\ell}(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N \omega_1^i$ 
4: for  $t = 2, \dots, T$  do
5:   Normalize weights  $w_{t-1}^i \propto \omega_{t-1}^i$ ,  $\sum_{i=1}^N w_{t-1}^i = 1$ 
6:   Resample  $\tilde{X}_{t-1}^i \sim \sum_{i=1}^N w_{t-1}^i \delta_{X_{t-1}^i}$  for  $i \in [N]$ 
7:   Sample  $X_t^i \sim q_\phi(\cdot | \tilde{X}_{t-1}^i, y_t)$  for  $i \in [N]$ 
8:   Compute  $\omega_t^i = \frac{p_\theta(X_t^i, y_t | \tilde{X}_{t-1}^i)}{q_\phi(X_t^i | \tilde{X}_{t-1}^i, y_t)}$  for  $i \in [N]$ 
9:   Compute  $\hat{p}_\theta(y_t | y_{1:t-1}) = \frac{1}{N} \sum_{i=1}^N \omega_t^i$ 
10:   $\hat{\ell}(\theta) \leftarrow \hat{\ell}(\theta) + \log \hat{p}_\theta(y_t | y_{1:t-1})$ 
11: end for
12: Return: log-likelihood estimate  $\hat{\ell}(\theta) = \log \hat{p}_\theta(y_{1:T})$ 

```

A generic PF is described in Algorithm 1 where $p_\theta(x_1, y_1) := \mu_\theta(x_1)g_\theta(y_1|x_1)$ and $p_\theta(x_t, y_t | x_{t-1}) := f_\theta(x_t | x_{t-1})g_\theta(y_t | x_t)$. Resampling is performed in step 6 of Algorithm 1; it ensures particles with high weights are replicated and those with low weights are discarded, allowing one to focus computational efforts on ‘promising’ regions. The scheme used in Algorithm 1 is known as multinomial resampling and is unbiased (as are other traditional schemes such as stratified and systematic (Chopin & Papaspiliopoulos, 2020)), i.e.

$$\mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N \psi(\tilde{X}_t^i) \right] = \mathbb{E} \left[\sum_{i=1}^N w_t^i \psi(X_t^i) \right], \quad (2)$$

for any $\psi : \mathcal{X} \rightarrow \mathbb{R}$. This property guarantees $\exp(\hat{\ell}(\theta))$ is an unbiased estimate of the likelihood $\exp(\ell(\theta))$ for any N .

Henceforth, let $\mathcal{X} = \mathbb{R}^{d_x}$, $\theta \in \Theta = \mathbb{R}^{d_\theta}$ and $\phi \in \Phi = \mathbb{R}^{d_\phi}$. We assume here that $\theta \mapsto \mu_\theta(x)$, $\theta \mapsto f_\theta(x' | x)$ and $\theta \mapsto g_\theta(y_t | x)$ are differentiable for all x, x' and $t \in [T]$ and $\theta \mapsto \ell(\theta)$ is differentiable. These assumptions are satisfied by a large class of SSMs. We also assume that we can use the reparameterization trick (Kingma & Welling, 2014) to sample the particles; i.e. we have $\Gamma_\phi(y_1, U) \sim q_\phi(x_1 | y_1)$, $\Psi_\phi(y_t, x_{t-1}, U) \sim q_\phi(x_t | x_{t-1}, y_t)$ for some mappings Γ_ϕ, Ψ_ϕ differentiable w.r.t. ϕ and $U \sim \lambda$, λ being independent of ϕ .

1.3. Related Work and Contributions

Let \mathbf{U} be the set of all random variables used to sample and resample the particles. The distribution of \mathbf{U} is (θ, ϕ) -independent as we use the reparameterization trick¹. However, even if we sample and fix $\mathbf{U} = \mathbf{u}$, resampling involves sampling from an atomic distribution and introduces discontinuities in the particles selected when θ, ϕ vary.

¹For example, multinomial resampling relies on N uniform random variables.

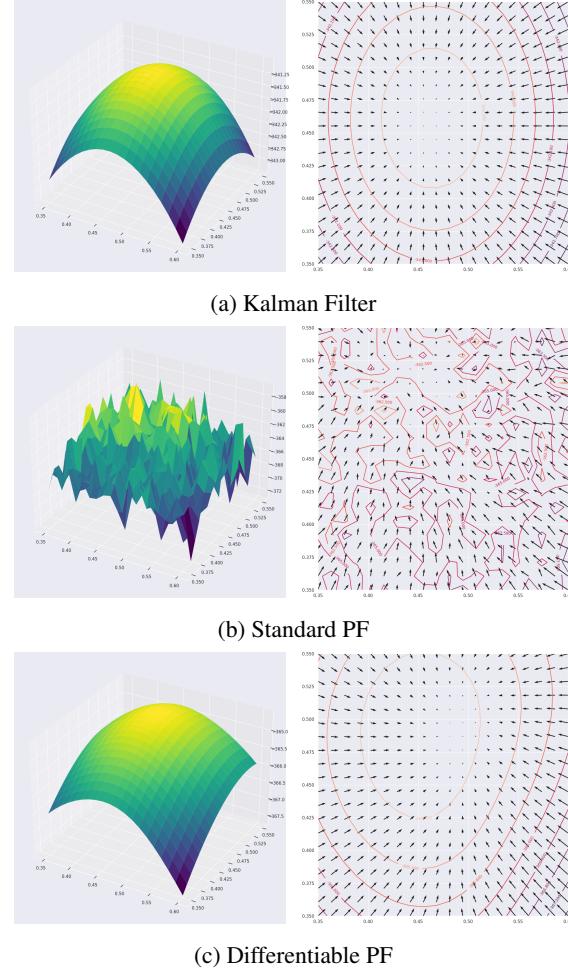


Figure 1. Left: Log-likelihood $\ell(\theta)$ and PF estimates $\hat{\ell}(\theta; \phi, \mathbf{u})$ for linear Gaussian SSM, given in Section 5.1, with $d_\theta = 2$, $d_x = 2$, and $T = 150$, $N = 50$. Right: $\nabla_\theta \ell(\theta)$ and $\nabla_\theta \hat{\ell}(\theta; \phi, \mathbf{u})$.

For $d_x = 1$, Malik & Pitt (2011) make $\theta \mapsto \hat{\ell}(\theta; \phi, \mathbf{u})$ continuous w.r.t. θ by sorting the particles and then sampling from a smooth approximation of their cumulative distribution function. For $d_x > 1$, Lee (2008) proposes a smoother but only piecewise continuous estimate. De-Jong et al. (2013) returns a differentiable log-likelihood estimate $\hat{\ell}(\theta; \phi, \mathbf{u})$ by using a marginal PF (Klaas et al., 2005), where importance sampling is performed on a collapsed state-space. However, the standard marginal PF uses the proposal $q_\phi(x_t) := \sum_{i=1}^N w_{t-1}^i q_\phi(x_t | X_{t-1}^i, y_t)$ from which one cannot generally sample smoothly for arbitrary mixture components. As a consequence they instead suggest using a simple Gaussian distribution for $q_\phi(x_t)$, which can lead to poor estimates for multimodal posteriors. Moreover, in contrast to standard PF, this marginal PF cannot be applied in scenarios where the transition density can only be sampled from (e.g. using the reparameterization trick) but not evaluated pointwise (Murray et al., 2013), as the importance weight would be intractable. The implicit

reparameterization method of [Graves \(2016\)](#) may be used to obtain low variance gradient estimates with a mixture proposal. This method however is only compatible for component distributions with tractable conditional CDFs, such as Gaussian distributions. An alternative unbiased estimate of the likelihood based on dynamic programming may also be obtained ([Finke et al., 2016](#); [Aitchison, 2019](#)). As emphasized by [Aitchison \(2019\)](#), this estimate is differentiable. This approach is again limited however to a restricted class of proposal distributions, such as an unweighted mixture proposal, which may perform poorly for slow-mixing time-series.

In the context of robot localization, a modified resampling scheme has been proposed in ([Karkus et al., 2018](#); [Ma et al., 2020a;b](#)) referred to as ‘soft-resampling’ (SPF). SPF has parameter $\alpha \in [0, 1]$ where $\alpha = 1$ corresponds to regular PF resampling and $\alpha = 0$ is essentially sampling particles uniformly at random. The resulting PF-net is said to be differentiable but computes gradients that ignore the non-differentiable component of the resampling step. [Jonschkowski et al. \(2018\)](#) proposed another PF scheme which is said to be differentiable but simply ignores the non-differentiable resampling terms and proposes new states based on the observation and some neural network. This approach however does not propagate gradients through time. Finally, [Zhu et al. \(2020\)](#) propose a differentiable resampling scheme based on transformers but they report that the best results are achieved when not backpropagating through it, due to exploding gradients. Hence no fully differentiable PF is currently available in the literature ([Kloss et al., 2020](#)).

PF methods have also been fruitfully exploited in Variational Inference (VI) to estimate θ, ϕ ([Le et al., 2018](#); [Maddison et al., 2017](#); [Naesseth et al., 2018](#)). As $\mathbb{E}_{\mathbf{U}}[\exp(\hat{\ell}(\theta; \phi, \mathbf{U}))] = \exp(\ell(\theta))$ is an unbiased estimate of $\exp(\ell(\theta))$ for any N, ϕ for standard PF, then one has indeed by Jensen’s inequality

$$\ell^{\text{ELBO}}(\theta, \phi) := \mathbb{E}_{\mathbf{U}}[\hat{\ell}(\theta; \phi, \mathbf{U})] \leq \ell(\theta). \quad (3)$$

The standard ELBO corresponds to $N = 1$ and many variational families for approximating $p_{\theta}(x_{1:T}|y_{1:T})$ have been proposed in this context ([Archer et al., 2015](#); [Krishnan et al., 2017](#); [Rangapuram et al., 2018](#)). The variational family induced by a PF differs significantly as $\ell^{\text{ELBO}}(\theta, \phi) \rightarrow \ell(\theta)$ as $N \rightarrow \infty$ and thus yields a variational approximation converging to $p_{\theta}(x_{1:T}|y_{1:T})$. This attractive property comes at a computational cost; i.e. the PF approach trades off fidelity to the posterior with computational complexity. While unbiased gradient estimates of the PF-ELBO (3) can be computed, they suffer from high variance as the resampling steps require having to use REINFORCE gradient estimates ([Williams, 1992](#)). Consequently, [Hirt & Dellaportas \(2019\)](#); [Le et al. \(2018\)](#); [Maddison et al. \(2017\)](#); [Naesseth et al. \(2018\)](#) use biased gradient estimates which ignore these

terms, yet report improvements as N increases over standard VI approaches and Importance Weighted Auto-Encoders (IWAE) ([Burda et al., 2016](#)).

Finally, if one is only interested in estimating θ (and not some distinct ϕ), then particle techniques approximating pointwise the score vector $\nabla_{\theta}\ell(\theta)$ are also available ([Poyiadjis et al., 2011](#); [Kantas et al., 2015](#)).

The contributions of this paper are four-fold.

- We propose the first fully Differentiable Particle Filter (DPF) which can use general proposal distributions. DPF provides a differentiable estimate of $\ell(\theta)$, see Figure 1-c, and more generally differentiable estimates of PF-based losses. Empirically, in a VI context, DPF-ELBO gradient estimates also exhibit much smaller variance than those of PF-ELBO.
- We provide quantitative convergence results on the differentiable resampling scheme and establish consistency results for DPF.
- We show that existing techniques provide inconsistent gradient estimates and that the non-vanishing bias can be very significant, leading practically to unreliable parameter estimates.
- We demonstrate that DPF empirically outperforms recent alternatives for end-to-end parameter estimation on a variety of applications.

Proofs of results are given in the Supplementary Material.

2. Resampling via Optimal Transport

2.1. Optimal Transport and the Wasserstein Metric

Since Optimal Transport (OT) ([Peyré & Cuturi, 2019](#); [Villani, 2008](#)) is a core component of our scheme, the basics are presented here. Given two probability measures α, β on $\mathcal{X} = \mathbb{R}^{d_x}$ the squared 2-Wasserstein metric between these measures is given by

$$\mathcal{W}_2^2(\alpha, \beta) = \min_{\mathcal{P} \in \mathcal{U}(\alpha, \beta)} \mathbb{E}_{(U, V) \sim \mathcal{P}} [||U - V||^2], \quad (4)$$

where $\mathcal{U}(\alpha, \beta)$ the set of distributions on $\mathcal{X} \times \mathcal{X}$ with marginals α and β , and the minimizing argument of (4) is the OT plan denoted \mathcal{P}^{OT} . Any element $\mathcal{P} \in \mathcal{U}(\alpha, \beta)$ allows one to “transport” α to β (and vice-versa) i.e.

$$\beta(dv) = \int \mathcal{P}(du, dv) = \int \mathcal{P}(dv|u)\alpha(du).$$

For atomic probability measures $\alpha_N = \sum_{i=1}^N a_i \delta_{u_i}$ and $\beta_N = \sum_{j=1}^N b_j \delta_{v_j}$ with weights $\mathbf{a} = (a_i)_{i \in [N]}$, $\mathbf{b} = (b_j)_{j \in [N]}$, and atoms $\mathbf{u} = (u_i)_{i \in [N]}$, $\mathbf{v} = (v_j)_{j \in [N]}$, one can show that

$$\mathcal{W}_2^2(\alpha_N, \beta_N) = \min_{\mathbf{P} \in \mathcal{S}(\mathbf{a}, \mathbf{b})} \sum_{i=1}^N \sum_{j=1}^N c_{i,j} p_{i,j}, \quad (5)$$

where any $\mathcal{P} \in \mathcal{U}(\alpha_N, \beta_N)$ is of the form

$$\mathcal{P}(du, dv) = \sum_{i,j} p_{i,j} \delta_{u_i}(du) \delta_{v_j}(dv),$$

$c_{i,j} = \|u_i - v_j\|^2$, $\mathbf{P} = (p_{i,j})_{i,j \in [N]}$ and $\mathcal{S}(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in [0, 1]^{N \times N} : \sum_{j=1}^N p_{i,j} = a_i, \sum_{i=1}^N p_{i,j} = b_j\}$. In such cases, one has

$$\mathcal{P}(dv|u = u_i) = \sum_j a_i^{-1} p_{i,j} \delta_{v_j}(dv). \quad (6)$$

The optimization problem (5) may be solved through linear programming. It is also possible to exploit the dual formulation

$$\mathcal{W}_2^2(\alpha_N, \beta_N) = \max_{\mathbf{f}, \mathbf{g} \in \mathcal{R}(\mathbf{C})} \mathbf{a}^t \mathbf{f} + \mathbf{b}^t \mathbf{g}, \quad (7)$$

where $\mathbf{f} = (f_i)$, $\mathbf{g} = (g_i)$, $\mathbf{C} = (c_{i,j})$ and $\mathcal{R}(\mathbf{C}) = \{\mathbf{f}, \mathbf{g} \in \mathbb{R}^N | f_i + g_j \leq c_{i,j}, i, j \in [N]\}$.

2.2. Ensemble Transform Resampling

The use of OT for resampling in PF has been pioneered by Reich (2013). Unlike standard resampling schemes (Chopin & Papaspiliopoulos, 2020; Doucet & Lee, 2018), it relies not only on the particle weights but also on their locations.

At time t , after the sampling step (Step 7 in Algorithm 1), $\alpha_N^{(t)} = \frac{1}{N} \sum_{i=1}^N \delta_{X_t^i}$ is a particle approximation of $\alpha^{(t)} := \int q_\phi(x_t|x_{t-1}, y_t) p_\theta(x_{t-1}|y_{1:t-1}) dx_{t-1}$ and $\beta_N^{(t)} = \sum w_t^i \delta_{X_t^i}$ is an approximation of $\beta^{(t)} := p_\theta(x_t|y_{1:t})$. Under mild regularity conditions, the OT plan minimizing $\mathcal{W}_2(\alpha^{(t)}, \beta^{(t)})$ is of the form $\mathcal{P}^{\text{OT}}(dx, dx') = \alpha^{(t)}(dx) \delta_{\mathbf{T}^{(t)}(x)}(dx')$ where $\mathbf{T}^{(t)} : \mathcal{X} \rightarrow \mathcal{X}$ is a deterministic map; i.e if $X \sim \alpha^{(t)}$ then $\mathbf{T}^{(t)}(X) \sim \beta^{(t)}$. It is shown in (Reich, 2013) that one can approximate this transport map with the ‘Ensemble Transform’ (ET) denoted $\mathbf{T}_N^{(t)}$. This is found by solving the OT problem (5) between $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ and taking an expectation w.r.t. (6), that is

$$\tilde{X}_t^i = N \sum_{k=1}^N p_{i,k}^{\text{OT}} X_t^k := \mathbf{T}_N^{(t)}(X_t^i), \quad (8)$$

where we slightly abuse notation as $\mathbf{T}_N^{(t)}$ is a function of $X_t^{1:N}$. Reich (2013) uses this update instead of using $\tilde{X}_t^i \sim \sum_{i=1}^N w_t^i \delta_{X_t^i}$. This is justified by the fact that, as $N \rightarrow \infty$, $\mathbf{T}_N^{(t)}(X_t^i) \rightarrow \mathbf{T}^{(t)}(X_t^i)$ in some weak sense (Reich, 2013; Myers et al., 2021). Compared to standard resampling schemes, the ET only satisfies (2) for affine functions ψ .

This OT approach to resampling involves solving the linear program (4) at cost $O(N^3 \log N)$ (Bertsimas & Tsitsiklis, 1997). This is not only prohibitively expensive but moreover the resulting ET is not differentiable. To address these problems, one may instead rely on entropy-regularized OT (Cuturi, 2013).

3. Differentiable Resampling via Entropy-Regularized Optimal Transport

3.1. Entropy-Regularized Optimal Transport

Entropy-regularized OT may be used to compute a transport matrix that is differentiable with respect to inputs and computationally cheaper than the non-regularized version, i.e. we consider the following regularized version of (5) for some $\epsilon > 0$ (Cuturi, 2013; Peyré & Cuturi, 2019)

$$\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N) = \min_{\mathbf{P} \in \mathcal{S}(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^N p_{i,j} \left(c_{i,j} + \epsilon \log \frac{p_{i,j}}{a_i b_j} \right). \quad (9)$$

The function minimized in (9) is strictly convex and hence admits a unique minimizing argument $\mathbf{P}_\epsilon^{\text{OT}} = (p_{\epsilon,i,j}^{\text{OT}})$. $\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N)$ can also be computed using the regularized dual; i.e. $\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N) = \max_{\mathbf{f}, \mathbf{g}} \text{DOT}_\epsilon(\mathbf{f}, \mathbf{g})$ with

$$\text{DOT}_\epsilon(\mathbf{f}, \mathbf{g}) := \mathbf{a}^t \mathbf{f} + \mathbf{b}^t \mathbf{g} - \epsilon \mathbf{a}^t \mathbf{M} \mathbf{b} \quad (10)$$

where $(\mathbf{M})_{i,j} := \exp(\epsilon^{-1}(f_i + g_j - c_{i,j})) - 1$ and \mathbf{f}, \mathbf{g} are now unconstrained. For the dual pair $(\mathbf{f}^*, \mathbf{g}^*)$ maximizing (10), we have $\nabla_{\mathbf{f}, \mathbf{g}} \text{DOT}_\epsilon(\mathbf{f}, \mathbf{g})|_{(\mathbf{f}^*, \mathbf{g}^*)} = \mathbf{0}$. This first-order condition leads to

$$f_i^* = \mathcal{T}_\epsilon(\mathbf{b}, \mathbf{g}^*, \mathbf{C}_{:,i}), \quad g_i^* = \mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}^*, \mathbf{C}_{:,i}), \quad (11)$$

where $\mathbf{C}_{:,i}$ (resp. $\mathbf{C}_{:,i}$) is the i^{th} row (resp. column) of \mathbf{C} . Here $\mathcal{T}_\epsilon : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ denotes the mapping

$$\mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}, \mathbf{C}_{:,i}) = -\epsilon \log \sum_k \exp \{ \log a_k + \epsilon^{-1}(f_k - c_{k,i}) \}.$$

One may then recover the regularized transport matrix as

$$p_{\epsilon,i,j}^{\text{OT}} = a_i b_j \exp(\epsilon^{-1}(f_i^* + g_j^* - c_{i,j})). \quad (12)$$

The dual can be maximized using the Sinkhorn algorithm introduced for OT in the seminal paper of Cuturi (2013). Algorithm 2 presents the implementation of Feydy et al. (2019) where the fixed point updates based on Equation (11) have been stabilized.

Algorithm 2 Sinkhorn Algorithm

- 1: **Function** Potentials($\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{v}$)
- 2: **Local variables:** $\mathbf{f}, \mathbf{g} \in \mathbb{R}^N$
- 3: **Initialize:** $\mathbf{f} = \mathbf{0}, \mathbf{g} = \mathbf{0}$
- 4: Set $\mathbf{C} \leftarrow \mathbf{u}\mathbf{u}^t + \mathbf{v}\mathbf{v}^t - 2\mathbf{u}\mathbf{v}^t$
- 5: **while** stopping criterion not met **do**
- 6: **for** $i \in [N]$ **do**
- 7: $f_i \leftarrow \frac{1}{2}(f_i + \mathcal{T}_\epsilon(\mathbf{b}, \mathbf{g}, \mathbf{C}_{:,i}))$
- 8: $g_i \leftarrow \frac{1}{2}(g_i + \mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}, \mathbf{C}_{:,i}))$
- 9: **end for**
- 10: **end while**
- 11: **Return** \mathbf{f}, \mathbf{g}

The resulting dual vectors $(\mathbf{f}^*, \mathbf{g}^*)$ can then be differentiated for example using automatic differentiation through the Sinkhorn algorithm loop (Flamary et al., 2018), or more efficiently using “gradient stitching” on the dual vectors at convergence, which we do here (see Feydy et al. (2019) for details). The derivatives of $\mathbf{P}_\epsilon^{\text{OT}}$ are readily accessible by combining the derivatives of (11) with the derivatives of (12), using automatic differentiation at no additional cost.

3.2. Differentiable Ensemble Transform Resampling

We obtain a differentiable ET (DET), denoted $\mathbf{T}_{N,\epsilon}^{(t)}$, by computing the entropy-regularized OT using Algorithm 3 for the weighted particles $(\mathbf{X}_t, \mathbf{w}_t, N)$ at time t

$$\tilde{\mathbf{X}}_t^i = N \sum_{k=1}^N p_{\epsilon,i,k}^{\text{OT}} X_t^k := \mathbf{T}_{N,\epsilon}^{(t)}(X_t^i). \quad (13)$$

Algorithm 3 DET Resampling

```

1: Function EnsembleTransform( $\mathbf{X}, \mathbf{w}, N$ )
2:  $\mathbf{f}, \mathbf{g} \leftarrow \text{Potentials}(\mathbf{w}, \frac{1}{N}\mathbf{1}, \mathbf{X}, \mathbf{X})$ 
3: for  $i \in [N]$  do
4:   for  $j \in [N]$  do
5:      $p_{\epsilon,i,j}^{\text{OT}} = \frac{w_i}{N} \exp\left(\frac{f_i + g_j - c_{i,j}}{\epsilon}\right)$ 
6:   end for
7: end for
8: Return  $\tilde{\mathbf{X}} = N \mathbf{P}_\epsilon^{\text{OT}} \mathbf{X}$ 

```

Compared to the ET, the DET is differentiable and can be computed at cost $O(N^2)$ as it relies on the Sinkhorn algorithm. This algorithm converges quickly (Altschuler et al., 2017) and is particularly amenable to GPU implementation.

The DPF proposed in this paper is similar to Algorithm 1 except that we sample from the proposal q_ϕ using the reparameterization trick and Step 6 is replaced by the DET. While such a differentiable approximation of the ET has previously been suggested in ML (Cuturi & Doucet, 2014; Seguy et al., 2018), it has never been realized before that this could be exploited to obtain a DPF. In particular, we obtain differentiable estimates of expectations w.r.t. the filtering distributions with respect to θ and ϕ and, for a fixed “seed” $\mathbf{U} = \mathbf{u}$ ², we obtain a differentiable estimate of the log-likelihood function $\theta \mapsto \hat{\ell}_\epsilon(\theta; \phi, \mathbf{u})$.

Like ET, DET only satisfies (2) for affine functions ψ . Unlike \mathbf{P}^{OT} , $\mathbf{P}_\epsilon^{\text{OT}}$ is sensitive to the scale of \mathbf{X}_t . To mitigate this sensitivity, one may compute $\delta(\mathbf{X}_t) = \sqrt{d_x} \max_{k \in [d_x]} \text{std}_i(X_{t,k}^i)$ for $\mathbf{X}_t \in \mathbb{R}^{N \times d_x}$ and rescale \mathbf{C} accordingly to ensure that ϵ is approximately independent of the scale and dimension of the problem.

²Here \mathbf{U} denotes only the set of θ, ϕ -independent random variables used to generate particles as, contrary to standard PF, DET resampling does not rely on any additional random variable.

4. Theoretical Analysis

We show here that the gradient estimates of PF-based losses ignoring gradients terms due to resampling are not consistent and can suffer from a large non-vanishing bias. On the contrary, we establish that DPF provides consistent and differentiable estimates of the filtering distributions and log-likelihood function. This is achieved by obtaining novel quantitative convergence results for the DET.

4.1. Gradient Bias from Ignoring Resampling Terms

We first provide theoretical results on the asymptotic bias of the gradient estimates computed from PF-losses, by dropping the gradient terms from resampling, as adopted in (Hirt & Dellaportas, 2019; Jonschkowski et al., 2018; Karkus et al., 2018; Le et al., 2018; Ma et al., 2020b; Maddison et al., 2017; Naesseth et al., 2018). We limit ourselves here to the ELBO loss. Similar analysis can be carried out for the non-differentiable resampling schemes and losses considered in robotics.

Proposition 4.1. *Consider the PF in Algorithm 1 where ϕ is distinct from θ then, under regularity conditions, the expectation of the ELBO gradient estimate $\hat{\nabla}_\theta \ell^{\text{ELBO}}(\theta, \phi)$ ignoring resampling terms considered in (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018) converges as $N \rightarrow \infty$ to*

$$\begin{aligned} \mathbb{E}[\hat{\nabla}_\theta \ell^{\text{ELBO}}(\theta, \phi)] &\rightarrow \int \nabla_\theta \log p_\theta(x_1, y_1) p_\theta(x_1|y_1) dx_1 \\ &+ \sum_{t=2}^T \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1}) p_\theta(x_{t-1:t}|y_{1:t}) dx_{t-1:t} \end{aligned}$$

whereas Fisher’s identity yields

$$\begin{aligned} \nabla_\theta \ell(\theta) &= \int \nabla_\theta \log p_\theta(x_1, y_1) p_\theta(x_1|y_{1:T}) dx_1 \quad (14) \\ &+ \sum_{t=2}^T \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1}) p_\theta(x_{t-1:t}|y_{1:T}) dx_{t-1:t}. \end{aligned}$$

Hence, whereas we have $\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi) \rightarrow \nabla_\theta \ell(\theta)$ as $N \rightarrow \infty$ under regularity assumptions, the asymptotic bias of $\hat{\nabla}_\theta \ell^{\text{ELBO}}(\theta, \phi)$ only vanishes if $p_\theta(x_{t-1:t}|y_{1:t}) = p_\theta(x_{t-1:t}|y_{1:T})$; i.e. for models where the X_t are independent. When $y_{t+1:T}$ do not bring significant information about X_t given $y_{1:T}$, as for the models considered in (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018), this is a reasonable approximation which explains the good performance reported therein. However, we show in Section 5 that this bias can also lead practically to inaccurate parameter estimation.

4.2. Quantitative Bounds on the DET

Weak convergence results for the ET have been established in (Reich, 2013; Myers et al., 2021) and the DET in (Seguy

et al., 2018). We provide here the first quantitative bound for the ET ($\epsilon = 0$) and DET ($\epsilon > 0$) which holds for any $N \geq 1$ by building upon results of (Li & Nocchetto, 2021) and (Weed, 2018). We use the notation $\nu(\psi) := \int \psi(x)\nu(dx)$ for any measure ν and function ψ .

Proposition 4.2. Consider atomic probability measures $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$, with support $\mathcal{X} \subset \mathbb{R}^d$. Let $\tilde{\beta}_N = \sum_{i=1}^N a_i \delta_{\tilde{X}_{N,\epsilon}^i}$ where $\tilde{\mathbf{X}}_{N,\epsilon} = \Delta^{-1} \mathbf{P}_\epsilon^{\text{OT}} \mathbf{X}$ for $\Delta = \text{diag}(a_1, \dots, a_N)$ and $\mathbf{P}_\epsilon^{\text{OT}}$ is the transport matrix corresponding to the ϵ -regularized OT coupling, $\mathcal{P}_\epsilon^{\text{OT},N}$, between α_N and β_N . Let α, β be two other probability measures, also supported on \mathcal{X} , such that there exists a unique λ -Lipschitz optimal transport map \mathbf{T} between them. Then for any bounded 1-Lipschitz function ψ , we have

$$\begin{aligned} |\beta_N(\psi) - \tilde{\beta}_N(\psi)| &\leq 2\lambda^{1/2} \mathcal{E}^{1/2} [\mathfrak{d}^{1/2} + \mathcal{E}]^{1/2} \\ &+ \max\{\lambda, 1\} [\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)], \end{aligned} \quad (15)$$

where $\mathfrak{d} := \sup_{x,y \in \mathcal{X}} |x - y|$ and $\mathcal{E} = \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon \log N}$.

If $\mathcal{W}_2(\alpha_N, \alpha), \mathcal{W}_2(\beta_N, \beta) \rightarrow 0$ and we choose $\epsilon_N = o(1/\log N)$ the bound given in (15) vanishes with $N \rightarrow \infty$. This suggested dependence of ϵ on N comes from the entropic radius, see Lemma C.1 in the Supplementary and (Weed, 2018), and is closely related to the fact that entropy-regularized OT is sensitive to the scale of \mathbf{X} . Equivalently one may rescale \mathbf{X} by a factor $\log N$ when computing the cost matrix. In particular when α_N and β_N are Monte Carlo approximations of α and β , we expect $\mathcal{W}_2(\alpha_N, \alpha), \mathcal{W}_2(\beta_N, \beta) = O(N^{-1/d})$ with high probability (Fournier & Guillin, 2015).

4.3. Consistency of DPF

The parameters θ, ϕ are here fixed and omitted from notation. We now establish consistency results for DPF, showing that both the resulting particle approximations $\tilde{\beta}_N^{(t)} = \frac{1}{N} \sum_{i=1}^N \delta_{\tilde{X}_t^i}$ of $\beta^{(t)} = p(x_t|y_{1:t})$ and the corresponding log-likelihood approximation $\log \hat{p}_N(y_{1:T})$ of $\log p(y_{1:T})$ are consistent. In the interest of simplicity, we limit ourselves to the scenario where the proposal is the transition, $q = f$, so $\omega(x_{t-1}, x_t, y_t) = g(y_t|x_t)$, known as the bootstrap PF and study a slightly non-standard version of it proposed in (Del Moral & Guionnet, 2001); see Appendix D for details. Consistency is established under regularity assumptions detailed in the Supplementary. Assumption B.1 is that the space $\mathcal{X} \subset \mathbb{R}^d$ has a finite diameter \mathfrak{d} . Assumption B.2 implies that the proposal mixes exponentially fast in the Wasserstein sense at a rate κ , which is reasonable given compactness, and essential for the error to not accumulate. Assumption B.3 assumes a bounded importance

weight function i.e. $g(y_t|x_t) \in [\Delta, \Delta^{-1}]$, again not unreasonable given compactness. Assumption B.4 states that at each time step, the optimal transport problem between $\alpha^{(t)}$ and $\beta^{(t)}$ is solved uniquely by a deterministic, globally Lipschitz map. Uniqueness is crucial for the quantitative stability results provided in the following proposition.

Proposition 4.3. Under Assumptions B.1, B.2, B.3 and B.4, for any $\delta > 0$, with probability at least $1 - 2\delta$ over the sampling steps, for any bounded 1-Lipschitz ψ , for any $t \in [1 : T]$, the approximations of the filtering distributions and log-likelihood computed by the bootstrap DPF satisfy

$$|\tilde{\beta}_N^{(t)}(\psi) - \beta^{(t)}(\psi)| \leq \mathfrak{G}_{\epsilon, \delta/T, N, d}^{(t)}(\lambda(c, C, d, T, N, \delta)),$$

$$\begin{aligned} \left| \log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})} \right| &\leq \frac{\kappa}{\Delta} \max_{t \in [1:T]} \text{Lip}[g(y_t \mid \cdot)] \\ &\times \sum_{t=1}^T \mathfrak{G}_{\epsilon, \delta/T, N, d}^{(t)}(\lambda(c, C, d, T, N, \delta)), \end{aligned}$$

for $\lambda(c, C, d, T, N, \delta) = \sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}$ where c, C are finite constants independent of T , and $\text{Lip}[f]$ is the Lipschitz constant of the function f , and $\mathfrak{G}_{N,\epsilon}^{(t)}, f_d$ defined in Appendix D are two functions such that if we set $\epsilon_N = o(1/\log N)$ then we have in probability

$$|\tilde{\beta}_N^{(t)}(\psi) - \beta^{(t)}(\psi)| \rightarrow 0, \quad \left| \log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})} \right| \rightarrow 0.$$

The above bounds are certainly not sharp. A glimpse into the behavior of the above bounds in terms of T can be obtained through careful consideration of the quantities appearing in Proposition D.1 in the supplement. In particular, for κ small enough, it suggests that the bound on the error of the log-likelihood estimator grows linearly with T as for standard PF under mixing assumptions. Sharper bounds are certainly possible, e.g. using a L_1 version of Theorem 3.5 in (Li & Nocchetto, 2021). It would also be of interest to weaken the assumptions, in particular, to remove the bounded space assumption although it is very commonly made in the PF literature to obtain quantitative bounds; see e.g. (Del Moral, 2004; Douc et al., 2014). Although this is not made explicit in the expressions above, there is an exponential dependence of the bounds on the state dimension d_x . This is unavoidable however and a well-known limitation of PF methods.

Finally note that DPF provides a biased estimate of the likelihood contrary to standard PF, so we cannot guarantee that the expectation of its logarithm, $\ell_\epsilon^{\text{ELBO}}(\theta, \phi) := \mathbb{E}_U[\hat{\ell}_\epsilon(\theta; \phi, U)]$, is actually a valid ELBO. However in all our experiments, see e.g. Section 5.1, $|\ell_\epsilon^{\text{ELBO}}(\theta, \phi) - \ell^{\text{ELBO}}(\theta, \phi)|$ is significantly smaller than $\ell(\theta) - \ell^{\text{ELBO}}(\theta, \phi)$ so $\ell_\epsilon^{\text{ELBO}}(\theta, \phi) < \ell(\theta)$. Hence we keep the ELBO terminology.

5. Experiments

In Section 5.1, we assess the sensitivity of the DPF to the regularization parameter ϵ . All other DPF experiments presented here use the DET Resampling detailed in Algorithm 3 with $\epsilon = 0.5$, which ensures stability of the gradient calculations while adding little bias to the calculation of the ELBO compared to standard PF. Our method is implemented in both PyTorch and TensorFlow, the code to replicate the experiments as well as further experiments may be found at <https://github.com/JTT94/filterflow>.

5.1. Linear Gaussian State-Space Model

We consider here a simple two-dimensional linear Gaussian SSM for which the exact likelihood can be computed exactly using the Kalman filter

$$\begin{aligned} X_{t+1} | \{X_t = x\} &\sim \mathcal{N}(\text{diag}(\theta_1 \theta_2)x, 0.5\mathbf{I}_2), \\ Y_t | \{X_t = x\} &\sim \mathcal{N}(x, 0.1\mathbf{I}_2). \end{aligned}$$

We simulate $T = 150$ observations using $\theta = (\theta_1, \theta_2) = (0.5, 0.5)$, for which we evaluate the ELBO at $\theta = (0.25, 0.25)$, $\theta = (0.5, 0.5)$, and $\theta = (0.75, 0.75)$. More precisely, using a standard PF with $N = 25$ particles, we compute the mean and standard deviation of $\frac{1}{T}(\hat{\ell}(\theta; \mathbf{U}) - \ell(\theta))$ over 100 realizations of \mathbf{U} . The mean is an estimate of the ELBO minus the true log-likelihood (rescaled by $1/T$). We then perform the same calculations for the DPF using the same number of particles and $\epsilon = 0.25, 0.5, 0.75$. As mentioned in Section 3.2 and Section 4.3, the DET resampling scheme is only satisfying Equation (2) for affine functions ψ so the DPF provides a biased estimate of the likelihood. Hence we cannot guarantee that the expectation of the corresponding log-likelihood estimate is a true ELBO. However, from Table 1, we observe that the difference between the ELBO estimates computed using PF and DPF is negligible for the three values of ϵ . The standard deviation of the log-likelihood estimates is also similar.

Table 1. Mean & std of $\frac{1}{T}(\hat{\ell}(\theta; \mathbf{U}) - \ell(\theta))$

θ_1, θ_2		0.25	0.5	0.75
PF	mean	-1.13	-0.93	-1.05
	std	0.20	0.18	0.17
DPF ($\epsilon = 0.25$)	mean	-1.14	-0.94	-1.07
	std	0.20	0.18	0.19
DPF ($\epsilon = 0.5$)	mean	-1.14	-0.94	-1.08
	std	0.20	0.18	0.18
DPF ($\epsilon = 0.75$)	mean	-1.14	-0.94	-1.08
	std	0.20	0.18	0.18

Recall here that alternative techniques estimating the score vector $\nabla_\theta \ell(\theta)$ by approximating (14) using particle smooth-

ing algorithms (Poyiadjis et al., 2011; Kantas et al., 2015) could also be used to estimate θ .

5.2. Learning the Proposal Distribution

We consider a similar example as in (Naesseth et al., 2018) where one learns the parameters ϕ of the proposal using the ELBO for the following linear Gaussian SSM:

$$X_{t+1} | \{X_t = x\} \sim \mathcal{N}(\mathbf{A}x, \mathbf{I}_{d_x}), \quad (16)$$

$$Y_t | \{X_t = x\} \sim \mathcal{N}(\mathbf{I}_{d_y, d_x}x, \mathbf{I}_{d_y}), \quad (17)$$

with $\mathbf{A} = (0.42^{|i-j|+1})_{1 \leq i,j \leq d_x}$, \mathbf{I}_{d_y, d_x} is a $d_y \times d_x$ matrix with 1 on the diagonal for the d_y first rows and zeros elsewhere. For $\phi \in \mathbb{R}^{d_x+d_y}$, we consider

$$q_\phi(x_t | x_{t-1}, y_t) = \mathcal{N}(x_t | \Delta_\phi^{-1}(\mathbf{A}x_{t-1} + \Gamma_\phi y_t), \Delta_\phi),$$

with $\Delta_\phi = \text{diag}(\phi_1, \dots, \phi_{d_x})$ and a $d_x \times d_y$ matrix $\Gamma_\phi = \text{diag}_{d_x, d_y}(\phi_1, \dots, \phi_{d_x})$ with ϕ_i on the diagonal for d_x first rows and zeros elsewhere. The locally optimal proposal $p(x_t | x_{t-1}, y_t) \propto g(y_t | x_t) f(x_t | x_{t-1})$ in (Doucet & Johansen, 2009) corresponds to $\phi = \mathbf{1}$, the vector with unit entries of dimension $d_\phi = d_x + d_y$.

For $d_x = 25, d_y = 1, M = 100$ realizations of $T = 100$ observations using (16)-(17), we learn ϕ on each realization using 100 steps of stochastic gradient ascent with learning rate 0.1 on the $\ell^{\text{ELBO}}(\phi)$ using regular PF with biased gradients as in (Maddison et al., 2017; Le et al., 2018; Naesseth et al., 2018) and $\ell^{\text{ELBO}}(\phi)$ with four independent filters using DPF. We use $N = 500$ for regular PF and $N = 25$ for DPF so as to match the computational complexity. While $p(x_t | x_{t-1}, y_t)$ is not guaranteed to maximize the ELBO, our experiments showed that it outperforms optimized proposals. We therefore report the RMSE of $\phi - \mathbf{1}$ and the average Effective Sample Size (ESS) (Doucet & Johansen, 2009) as proxy performance metrics. On both metrics, DPF outperforms regular PF. The RMSE over 100 experiments is 0.11 for DPF vs 0.22 for regular PF while the average ESS after convergence is around 60% for DPF vs 25% for regular PF. The average time per iteration was around 15 seconds for both DPF and PF.

5.3. Variational Recurrent Neural Network (VRNN)

A VRNN is an SSM introduced by (Chung et al., 2015) to improve upon LSTMs (Long Short Term Memory networks) with the addition of a stochastic component to the hidden state, this extends variational auto-encoders to a sequential setting. Indeed let latent state be $X_t = (R_t, Z_t)$ where R_t is an RNN state and Z_t a latent Gaussian variable, here Y_t is a vector of binary observations. The VRNN is detailed as follows. RNN_θ denotes the forward call of an LSTM cell which at time t emits the next RNN state R_{t+1} and output O_{t+1} . $E_\theta, h_\theta, \mu_\theta, \sigma_\theta$ are fully connected neural networks; detailed fully in the Supplementary Material.

This model is trained on the polyphonic music benchmark datasets (Boulanger-Lewandowski et al., 2012), whereby Y_t represents which notes are active. The observation sequences are capped to length 150 for each dataset, with each observation of dimension 88. We chose latent states Z_t and R_t to be of dimension $d_z = 8$ and $d_r = 16$ respectively so $d_x = 24$. We use $q_\phi(x_t|x_{t-1}, y_t) = f_\theta(x_t|x_{t-1})$.

$$\begin{aligned} (R_{t+1}, O_{t+1}) &= \text{RNN}_\theta(R_t, Y_{1:t}, E_\theta(Z_t)), \\ Z_{t+1} &\sim \mathcal{N}(\mu_\theta(O_{t+1}), \sigma_\theta(O_{t+1})), \\ \hat{p}_{t+1} &= h_\theta(E_\theta(Z_{t+1}), O_{t+1}), \\ Y_t|X_t &\sim \text{Ber}(\hat{p}_t). \end{aligned}$$

Table 2. ELBO \pm Standard Deviation evaluated using Test Data.

	MUSEDATA	JSB	NOTTINGHAM
DPF	-7.59 ± 0.01	-7.67 ± 0.08	-3.79 ± 0.02
PF	-7.60 ± 0.06	-7.92 ± 0.13	-3.81 ± 0.02
SPF	-7.73 ± 0.14	-8.17 ± 0.07	-3.91 ± 0.05

The VRNN model is trained by maximizing $\ell_e^{\text{ELBO}}(\theta)$ using DPF. We compare this to the same model trained by maximizing $\ell^{\text{ELBO}}(\theta)$ computed with regular PF (Maddison et al., 2017) and also trained with ‘soft-resampling’ (SPF) introduced by (Karkus et al., 2018) and described in Section 1.3, SPF is used here with parameter $\alpha = 0.1$. Unlike regular resampling, SPF partially preserves a gradient through the resampling step, however SPF still involves a non-differentiable operation, again resulting in a biased gradient. SPF also produces higher variance estimates as the resampled approximation is not uniformly weighted, essentially interpolating between PF and IWAE. Each of the methods are performed with $N = 32$ particles. Although DET is computationally more expensive than the other resampling schemes, the computational times of DPF, PF, and SPF are very similar due to most of the complexity coming from neural network operations. The learned models are then evaluated on test data using multinomial resampling for comparable ELBO results. Due to the fact that our observation model is $\text{Ber}(\hat{p}_t)$, this recovers the negative log-predictive cross-entropy.

Table 2 illustrates the benefit of using DPF over regular PF and SPF for the JSB dataset. Although DPF remains competitive compared to other heuristic approaches, the difference is relatively minor for the other datasets. We speculate that the performance of the heuristic methods is likely due to low predictive uncertainty for the next observation given the previous one.

5.4. Robot Localization

Consider the setting of a robot/agent in a maze (Jonschkowski et al., 2018; Karkus et al., 2018). Given the agent’s initial state, S_1 , and inputs a_t , one would like

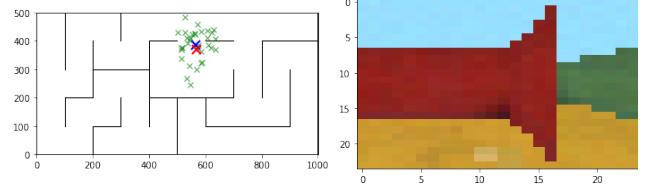


Figure 2. Left: Particles $(X_t^{(1),i}, X_t^{(2),i})$ (green), PF estimate of $\mathbb{E}[X_t|y_{1:t}]$ (blue), true state X_t^* (red). Right: Observation, O_t .

to infer the location of the agent at any specific time given observations O_t . Let the latent state be denoted $S_t = (X_t^{(1)}, X_t^{(2)}, \gamma_t)$ where $(X_t^{(1)}, X_t^{(2)})$ are location coordinates and γ_t the robot’s orientation. In our setting observations O_t are images, which are encoded to extract useful features using a neural network E_θ , where $Y_t = E_\theta(O_t)$. This problem requires learning the relationship between the robot’s location, orientation and the observations. Given actions $a_t = (v_t^{(1)}, v_t^{(2)}, \omega_t)$, we have

$$\begin{aligned} S_{t+1} &= F_\theta(S_t, a_t) + \nu_t, \quad \nu_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \Sigma_F), \\ Y_t &= G_\theta(S_t) + \epsilon_t, \quad \epsilon_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_G^2 \mathbb{I}_{d_d}), \end{aligned}$$

where $\Sigma_F = \text{diag}(\sigma_x^2, \sigma_x^2, \sigma_\theta^2)$ and the relationship between state S_t and image encoding Y_t may be parameterized by another neural network G_θ . We consider here a simple linear model of the dynamics

$$F(S_t, a_t) = \begin{bmatrix} X_t^{(1)} + v_t^{(1)} \cos(\gamma_t) + v_t^{(2)} \sin(\gamma_t) \\ X_t^{(2)} + v_t^{(1)} \sin(\gamma_t) - v_t^{(2)} \cos(\gamma_t) \\ \gamma_t + \omega_t \end{bmatrix}.$$

D_θ denotes a decoder neural network, mapping the encoding back to the original image. E_θ , G_θ and D_θ are trained using a loss function consisting of the PF-estimated log-likelihood $\hat{\mathcal{L}}_{\text{PF}}$; PF-based mean squared error (MSE), $\hat{\mathcal{L}}_{\text{MSE}}$; and auto-encoder loss, $\hat{\mathcal{L}}_{\text{AE}}$, given per-batch as in (Wen et al., 2020):

$$\begin{aligned} \hat{\mathcal{L}}_{\text{MSE}} &:= \frac{1}{T} \sum_{t=1}^T \|X_t^* - \sum_{i=1}^N w_t^i X_t^i\|^2, \quad \hat{\mathcal{L}}_{\text{PF}} := -\frac{1}{T} \hat{\ell}(\theta), \\ \hat{\mathcal{L}}_{\text{AE}} &:= \sum_{t=1}^T \|D_\theta(E_\theta(O_t)) - O_t\|^2, \end{aligned}$$

where X_t^* are the true states available from training data and $\sum_{i=1}^N w_t^i X_t^i$ are the PF estimates of $\mathbb{E}[X_t|y_{1:t}]$. The auto-encoder / reconstruction loss $\hat{\mathcal{L}}_{\text{AE}}$ ensures the encoder is informative and prevents the case whereby networks G_θ , E_θ map to a constant. The PF-based loss terms $\hat{\mathcal{L}}_{\text{MSE}}$ and $\hat{\mathcal{L}}_{\text{PF}}$ are not differentiable w.r.t. θ under traditional resampling schemes.

We use the setup from (Jonschkowski et al., 2018) with data from DeepMind Lab (Beattie et al., 2016). This consists of

Table 3. MSE and \pm Standard Deviation evaluated on Test Data:
Lower is better

	MAZE 1	MAZE 2	MAZE 3
DPF	3.55 ± 0.20	4.65 ± 0.50	4.44 ± 0.26
PF	10.71 ± 0.45	11.86 ± 0.57	12.88 ± 0.65
SPF	9.14 ± 0.39	10.12 ± 0.40	11.42 ± 0.37

3 maze layouts of varying sizes. We have access to ‘true’ trajectories of length 1,000 steps for each maze. Each step has an associated state, action and observation image, as described above. The visual observation O_t consists of 32×32 RGB pixel images, compressed to 24×24 , as shown in Figure 2. Random, noisy subsets of fixed length are sampled at each training iteration. To illustrate the benefits of our proposed method, we select the random subsets to be of length 50 as opposed to length 20 as chosen in (Jonschkowski et al., 2018). Training details in terms of learning rates, number of training steps and neural network architectures for E_θ , G_θ and D_θ are given in the Appendices.

We compare our method, DPF, to regular PF used in (Madison et al., 2017) and Soft PF (SPF) used in (Karkus et al., 2018; Ma et al., 2020a;b), whereby the soft resampling is used with $\alpha = 0.1$. As most of the computational complexity arises from neural network operations, DPF is of similar overall computational cost to SPF and PF. As shown in Table 3 and Figure 3, DPF significantly outperforms previously considered PF methods in this experiment. The observation model becomes increasingly important for longer sequences due to resampling and weighting operations. Indeed, as shown in Figure 4, the error is small for both models at the start of the sequence, however the error at later stages in the sequence is visibly smaller for the model trained using DPF.

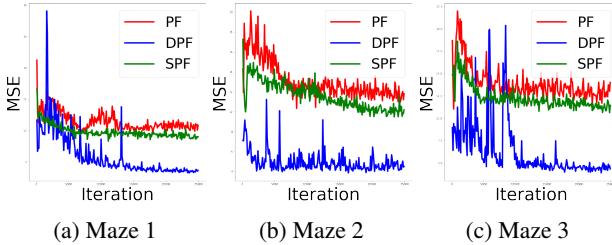


Figure 3. MSE of PF (red), SPF (green) and DPF (blue) estimates, evaluated on test data during training.

6. Discussion

This paper introduces the first principled, fully differentiable PF (DPF) which can use general proposal distributions. It provides a differentiable estimate of the log-likelihood function and more generally differentiable estimates of PF-based losses. This permits parameter inference in state-space mod-

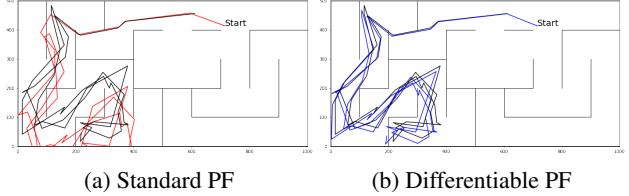


Figure 4. Illustrative Example: PF estimate of path compared to true path (black) on a single 50-step trajectory from test data.

els and proposal distributions, using end-to-end gradient based optimization. This also allows the use of PF routines in general differentiable programming pipelines, in particular as a differentiable sampling method for inference in probabilistic programming languages (Dillon et al., 2017; Ge et al., 2018; van de Meent et al., 2018).

For a given number of particles N , existing PF methods ignoring resampling gradient terms have computational complexity $O(N)$. Training with these resampling schemes however is unreliable and performance cannot be improved by increasing N as gradient estimates are inconsistent and the limiting bias can be significant. DPF has complexity $O(N^2)$ during training. However, this cost is dwarfed when training large neural networks. Additionally, once the model is trained, standard PF may be ran at complexity $O(N)$. The benefits of DPF are confirmed by our experimental results where it was shown to outperform existing techniques, even when an equivalent computational budget was used. Moreover, recent techniques have been proposed to speed up the Sinkhorn algorithm (Altschuler et al., 2019; Scetbon & Cuturi, 2020) at the core of DPF and could potentially be used here to reduce its complexity.

Regularization parameter ϵ was not fine-tuned in our experiments. In future work, it would be interesting to obtain sharper quantitative bounds on DPF to propose principled guidelines on choosing ϵ , further improving its performance. Finally, we have focused on the use of the differentiable ensemble transform to obtain a differentiable resampling scheme. However, alternative OT approaches could also be proposed such as a differentiable version of the second order ET presented in (Acevedo et al., 2017), techniques based on point cloud optimization (Cuturi & Doucet, 2014; Peyré & Cuturi, 2019) relying on the Sinkhorn divergence (Genevay et al., 2018) or the sliced-Wasserstein metric. Alternative non-entropic regularizations, such as the recently proposed Gaussian smoothed OT (Goldfeld & Greenwald, 2020), could also lead to DPFs of interest.

Acknowledgments

Adrien Corenflos was supported by the Academy of Finland (projects 321900 and 321891). Arnaud Doucet is supported by the EPSRC CoSInES (COmputational Sta-

tistical INference for Engineering and Security) grant EP/R034710/1, James Thornton by the OxWaSP CDT through grant EP/L016710/1. Computing resources were provided through the Google Cloud Research Credits Programme. The authors thank Valentin De Bortoli and Yuyang Shi for their comments and Laurence Aitchison for bringing attention to reference (Aitchison, 2019).

References

- Acevedo, W., de Wiljes, J., and Reich, S. Second-order accurate ensemble transform particle filters. *SIAM Journal on Scientific Computing*, 39(5):A1834–A1850, 2017.
- Aitchison, L. Tensor Monte Carlo: particle methods for the GPU era. *Advances in Neural Information Processing Systems*, 2019.
- Altschuler, J., Niles-Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pp. 1964–1974, 2017.
- Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. Massively scalable Sinkhorn distances via the Nyström method. In *Advances in Neural Information Processing Systems*, pp. 4429–4439, 2019.
- Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.
- Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittweis, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. DeepMind Lab, 2016.
- Bertsimas, D. and Tsitsiklis, J. N. *Introduction to Linear Optimization*. Athena Scientific Belmont, MA, 1997.
- Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *International Conference on Machine Learning*, pp. 1881–1888, 2012.
- Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.
- Chopin, N. and Papaspiliopoulos, O. *An Introduction to Sequential Monte Carlo*. Springer, 2020.
- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pp. 2980–2988, 2015.
- Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pp. 2292–2300, 2013.
- Cuturi, M. and Doucet, A. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pp. 685–693, 2014.
- DeJong, D. N., Liesenfeld, R., Moura, G. V., Richard, J.-F., and Dharmarajan, H. Efficient likelihood evaluation of state-space representations. *Review of Economic Studies*, 80(2):538–567, 2013.
- Del Moral, P. *Feynman-Kac Formulae*. Springer, 2004.
- Del Moral, P. and Guionnet, A. On the stability of interacting processes with applications to filtering and genetic algorithms. In *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, volume 37, pp. 155–194, 2001.
- Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.
- Douc, R., Moulines, E., and Stoffer, D. *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC press, 2014.
- Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.
- Doucet, A. and Lee, A. Sequential Monte Carlo methods. *Handbook of Graphical Models*, pp. 165–189, 2018.
- Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-I., Trouvé, A., and Peyré, G. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 2019.
- Finke, A., Doucet, A., and Johansen, A. M. On embedded hidden Markov models and particle Markov chain Monte Carlo methods. *arXiv preprint arXiv:1610.08962*, 2016.
- Flamary, R., Cuturi, M., Courty, N., and Rakotomamonjy, A. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018.
- Fournier, N. and Guillin, A. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015.
- Ge, H., Xu, K. X., and Ghahramani, Z. Turing: A language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 1682–1690, 2018.

- Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, 2018.
- Goldfeld, Z. and Greenwald, K. Gaussian-smooth optimal transport: Metric structure and statistical efficiency. *arXiv preprint arXiv:2001.09206*, 2020.
- Graves, A. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.
- Hirt, M. and Dellaportas, P. Scalable Bayesian learning for state space models using variational inference with SMC samplers. In *International Conference on Artificial Intelligence and Statistics*, pp. 76–86, 2019.
- Jonschkowski, R., Rastogi, D., and Brock, O. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proceedings of Robotics: Science and Systems*, 2018.
- Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015.
- Karkus, P., Hsu, D., and Lee, W. S. Particle filter networks with application to visual localization. In *Conference on Robot Learning*, 2018.
- Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.
- Kitagawa, G. and Gersch, W. *Smoothness Priors Analysis of Time Series*, volume 116. Springer Science & Business Media, 1996.
- Klaas, M., De Freitas, N., and Doucet, A. Toward practical N^2 Monte Carlo: the marginal particle filter. *Uncertainty in Artificial Intelligence*, 2005.
- Kloss, A., Martius, G., and Bohg, J. How to train your differentiable filter. *arXiv preprint arXiv:2012.14313*, 2020.
- Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*, pp. 2101–2109, 2017.
- Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.
- Lee, A. Towards smooth particle filters for likelihood estimation with multivariate latent variables. Master’s thesis, University of British Columbia, 2008.
- Li, W. and Nocchetto, R. H. Quantitative stability and error estimates for optimal transport plans. *IMA Journal of Numerical Analysis*, 2021.
- Lindsten, F. and Schön, T. B. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.
- Ma, X., Karkus, P., Hsu, D., and Lee, W. S. Particle filter recurrent neural networks. In *AAAI Conference on Artificial Intelligence*, 2020a.
- Ma, X., Karkus, P., Ye, N., Hsu, D., and Lee, W. S. Discriminative particle filter reinforcement learning for complex partial observations. In *International Conference on Learning Representations*, 2020b.
- Maddison, C. J., Lawson, D., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. W. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, 2017.
- Malik, S. and Pitt, M. K. Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209, 2011.
- Murray, L. M., Jones, E. M., and Parslow, J. On disturbance state-space models and the particle marginal Metropolis–Hastings sampler. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):494–521, 2013.
- Myers, A., Thiery, A. H., Wang, K., and Bui-Thanh, T. Sequential ensemble transform for Bayesian inverse problems. *Journal of Computational Physics*, 427:110055, 2021.
- Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2018.
- Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5–6): 355–607, 2019.
- Poyiadjis, G., Doucet, A., and Singh, S. S. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.
- Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, pp. 7785–7794, 2018.

Reich, S. A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, 2013.

Scetbon, M. and Cuturi, M. Linear time Sinkhorn divergences using positive features. In *Advances in Neural Information Processing Systems*, 2020.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018.

Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics*. MIT Press, 2005.

van de Meent, J.-W., Paige, B., Hongseok, Y., and Wood, F. An introduction to probabilistic programming. *arXiv preprint arXiv:1809.10756*, 2018.

Villani, C. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

Weed, J. An explicit analysis of the entropic penalty in linear programming. In *Proceedings of the 31st Conference On Learning Theory*, 2018.

Wen, H., Chen, X., Papagiannis, G., Hu, C., and Li, Y. End-to-end semi-supervised learning for differentiable particle filters. *arXiv preprint arXiv:2011.05748*, 2020.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Zhu, M., Murphy, K., and Jonschkowski, R. Towards differentiable resampling. *arXiv preprint arXiv:2004.11938*, 2020.

A. Proof of Proposition 4.1

A particle filter with multinomial resampling is defined by the following joint distribution

$$\bar{q}_{\theta, \phi}(x_{1:T}^{1:N}, a_{1:T-1}^{1:N}) = \prod_{i=1}^N q_\phi(x_1^i) \prod_{t=2}^T \prod_{i=1}^N w_{t-1}^{a_{t-1}^i} q_\phi(x_t^i | x_{t-1}^{a_{t-1}^i})$$

where $a_{t-1}^i \in \{1, \dots, N\}$ is the ancestral index of particle x_t^i and

$$\omega_{\theta, \phi}(x_1, y_1) = \frac{p_\theta(x_1, y_1)}{q_\phi(x_1)}, \quad \omega_{\theta, \phi}(x_{t-1}, x_t, y_t) = \frac{p_\theta(x_t, y_t | x_{t-1})}{q_\phi(x_t | x_{t-1})}.$$

Finally, we have $w_t^i \propto \omega_{\theta, \phi}(x_{t-1}^{a_{t-1}^i}, x_t^i, y_t)$, $\sum_{i=1}^N w_t^i = 1$. We do not emphasize notationally that the weights $w_{t-1}^{a_{t-1}^i}$ are θ, ϕ and observations dependent.

The ELBO is given by

$$\ell^{\text{ELBO}}(\theta, \phi) = \mathbb{E}_{\bar{q}_{\theta, \phi}} [\log \hat{p}_\theta(y_{1:T})] = \mathbb{E}_{\bar{q}_{\theta, \phi}} \left[\log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_1^i, y_1) \right) + \sum_{t=2}^T \log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t) \right) \right].$$

We now compute $\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi)$. We assume from now on that the regularity conditions allowing us to swap the expectation and differentiation operators are satisfied as in (Maddison et al., 2017; Le et al., 2018; Naesseth et al., 2018). We can split the gradient using the product rule and apply the log-derivative trick:

$$\begin{aligned} \nabla_\theta \ell^{\text{ELBO}}(\theta, \phi) &= \mathbb{E}_{\bar{q}_{\theta, \phi}} [\nabla_\theta \log \hat{p}_\theta(y_{1:T})] + \mathbb{E}_{\bar{q}_{\theta, \phi}} [\log \hat{p}_\theta(y_{1:T}) \nabla_\theta \log \bar{q}_{\theta, \phi}(X_{1:T}^{1:N}, A_{1:T-1}^{1:N})] \\ &= \mathbb{E}_{\bar{q}_{\theta, \phi}} \left[\nabla_\theta \log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_1^i, y_1) \right) + \sum_{t=2}^T \nabla_\theta \log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t) \right) \right] \end{aligned} \quad (18)$$

$$+ \mathbb{E}_{\bar{q}_{\theta, \phi}} \left[\log \hat{p}_\theta(y_{1:T}) \left\{ \sum_{t=2}^T \sum_{i=1}^N \nabla_\theta \log w_{t-1}^{A_{t-1}^i} \right\} \right] \quad (19)$$

For the first part of the ELBO gradient (18), we have

$$\nabla_\theta \log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_1^i, y_1) \right) = \sum_{i=1}^N w_1^i \nabla_\theta \log \omega_{\theta, \phi}(X_1^i, y_1) = \sum_{i=1}^N w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1)$$

and

$$\nabla_\theta \log \left(\frac{1}{N} \sum_{i=1}^N \omega_{\theta, \phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t) \right) = \sum_{i=1}^N w_t^i \nabla_\theta \log \omega_{\theta, \phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t) = \sum_{i=1}^N w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i}).$$

This gives

$$\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi) = \mathbb{E}_{\bar{q}_{\theta, \phi}} \left[\sum_{i=1}^N w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1) + \sum_{t=2}^T \sum_{i=1}^N w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i}) \right] \quad (20)$$

$$+ \mathbb{E}_{\bar{q}_{\theta, \phi}} \left[\log \hat{p}_\theta(y_{1:T}) \left\{ \sum_{t=2}^T \sum_{i=1}^N \nabla_\theta \log w_{t-1}^{A_{t-1}^i} \right\} \right]. \quad (21)$$

When we ignore the gradient terms due to resampling corresponding to (21) as proposed in (Naesseth et al., 2018; Le et al., 2018; Maddison et al., 2017; Hirt & Dellaportas, 2019), we only use an unbiased estimate of the first term (20), i.e.

$$\hat{\nabla}_\theta \ell^{\text{ELBO}}(\theta, \phi) := \sum_{i=1}^N w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1) + \sum_{t=2}^T \sum_{i=1}^N w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i}), \quad \text{where } (X_{1:T}^{1:N}, A_{1:T-1}^{1:N}) \sim \bar{q}_{\theta, \phi}(\cdot). \quad (22)$$

Now we assume that the mild assumptions ensuring almost sure convergence of the PF estimates are satisfied (see e.g. (Del Moral, 2004)). Under these assumptions, the estimator (22) converges almost surely as $N \rightarrow \infty$ towards

$$\int \nabla_{\theta} \log p_{\theta}(x_1, y_1) p_{\theta}(x_1 | y_1) dx_1 + \sum_{t=2}^T \int \nabla_{\theta} \log p_{\theta}(x_t, y_t | x_{t-1}) p_{\theta}(x_{t-1:t} | y_{1:t-1}) dx_{t-1:t}. \quad (23)$$

Under an additional uniform integrability condition on $\hat{\nabla}_{\theta} \ell^{\text{ELBO}}(\theta, \phi)$, we thus have that $\mathbb{E}_{\bar{q}_{\theta, \phi}}[\hat{\nabla}_{\theta} \ell^{\text{ELBO}}(\theta, \phi)]$ converges towards (23). We recall that the true score is given by Fisher's identity and satisfies

$$\int \nabla_{\theta} \log p_{\theta}(x_1, y_1) p_{\theta}(x_1 | y_{1:T}) dx_1 + \sum_{t=2}^T \int \nabla_{\theta} \log p_{\theta}(x_t, y_t | x_{t-1}) p_{\theta}(x_{t-1:t} | y_{1:T}) dx_{t-1:t}.$$

This concludes the proof of Proposition 4.1.

B. Notation and Assumptions

B.1. Filtering Notation

Recall $\mathcal{X} = \mathbb{R}^{d_x}$, denote the Borel sets of \mathcal{X} by $\mathcal{B}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X})$ the set of Borel probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. In an abuse of notation, we shall use the same notation for a probability measure and its density w.r.t. Lebesgue measure; i.e. $\nu(dx) = \nu(x)dx$. We also use the standard notation $\nu(\psi) = \int \psi(x)\nu(x)dx$ for any test function ψ . In the interest of notational clarity, we will remove subscript θ, ϕ where unnecessary in further workings.

We denote $\{\alpha^{(t)}\}_{t \geq 0}$ the predictive distributions where $\alpha^{(t)}(x_t) = p(x_t | y_{1:t-1})$ for $t > 1$ and $\alpha^{(1)}(x_1) = \mu(x_1)$ while $\{\beta^{(t)}\}_{t \geq 1}$ denotes the filtering distributions; i.e. $\beta^{(t)}(x_t) = p(x_t | y_{1:t})$ for $t \geq 1$.

Using this notation, we have

$$\alpha^{(t)}(\psi) = \int \psi(x_t) f(x_t | x_{t-1}) \beta^{(t-1)}(x_{t-1}) dx_{t-1} dx_t := \beta^{(t-1)} f(\psi), \quad (24)$$

$$\beta^{(t)}(\psi) = \frac{\alpha^{(t)}(g(y_t | \cdot) \psi)}{\alpha^{(t)}(g(y_t | \cdot))} = \frac{\beta^{(t-1)}(f(g(y_t | \cdot) \psi))}{\beta^{(t-1)}(f(g(y_t | \cdot)))}. \quad (25)$$

More generally, for a proposal distribution $q(x_t | x_{t-1}, y_t) \neq f(x_t | x_{t-1})$ with parameter $\phi \neq \theta$, the following recursion holds

$$\beta^{(t)}(\psi) = \frac{\beta^{(t-1)}(q(\omega_t \psi))}{\beta^{(t-1)}(q(\omega_t))} \quad (26)$$

$$\omega_t(x_{t-1}, x_t) := \omega(x_{t-1}, x_t, y_t) = \frac{g(y_t | x_t) f(x_t | x_{t-1})}{q(x_t | x_{t-1}, y_t)}. \quad (27)$$

To simplify the presentation, we will present the analysis in the scenario where $\phi = \theta$ and $q(x_t | x_{t-1}, y_t) = f(x_t | x_{t-1})$ so we will analyze (24) for which $\omega_t(x_{t-1}, x_t) = g(y_t | x_t)$. In this case, the particle approximations of μ is denoted μ_N and for $t > 1$, $\alpha^{(t)}$ and $\beta^{(t)}$ are given by the random measures

$$\alpha_N^{(t)}(\psi) = \frac{1}{N} \sum_{i=1}^N \psi(X_t^i), \quad \beta_N^{(t)}(\psi) = \sum_{i=1}^N w_t^i \psi(X_t^i), \quad \tilde{\beta}_N^{(t)}(\psi) = \frac{1}{N} \sum_{i=1}^N \psi(\tilde{X}_t^i), \quad (28)$$

where $w_t^i \propto g(y_t | X_t^i)$ with $\sum_{i=1}^N w_t^i = 1$ and particles are drawn from $X_t^i \sim f(\cdot | \tilde{X}_{t-1}^i)$.

Here $\beta_N^{(t)}$ denotes the weighted particle approximation of $\beta^{(t)}$ while $\tilde{\beta}_N^{(t)}$ is the uniformly weighted approximation obtained after the DET transformation described in Section 3.2.

B.2. Optimal Transport Notation

Recall from Section 2.1, $\mathcal{P}_t^{\text{OT}}$ denotes a transport between $\alpha^{(t)}$ and $\beta^{(t)}$ with accompanying map $\mathbf{T}^{(t)}$. $\mathcal{P}_t^{\text{OT},N}$ denotes an optimal transport between particle approximations $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ with corresponding transport matrix, \mathbf{P}^{OT} with i, j entry $p_{i,j}^{\text{OT}}$. To simplify notation, we remove script t when not needed.

Similarly from Section 3.1, $\mathcal{P}_{\epsilon}^{\text{OT},N}$ denotes the regularized transport between $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ with accompanying matrix $\mathbf{P}_{\epsilon}^{\text{OT}}$ with i, j entry $p_{\epsilon,i,j}^{\text{OT}}$. Recall $\tilde{\beta}_N^{(t)} = \frac{1}{N} \sum_{i=1}^N \delta_{\tilde{X}^i}$ is the uniformly weighted particle approximation for $\beta^{(t)}$ under the DET, i.e. $\tilde{X}^i = \mathbf{T}_{N,\epsilon}^{(t)}(X^i) = \int y \mathcal{P}_{\epsilon}^{\text{OT},N}(dy|x^i)$. Note that $\tilde{X}_{N,\epsilon}^i$ will be used where necessary to avoid ambiguity when comparing to other resampling schemes.

Recall also for $p > 0$:

$$\mathcal{W}_p^p(\alpha, \beta) = \min_{\mathcal{U}(\alpha, \beta)} \mathbb{E}_{(U, V) \sim \mathcal{P}} [||U - V||^p] \quad (29)$$

where $\mathcal{U}(\alpha, \beta)$ is the collection of couplings with marginals α and β .

B.3. Assumptions

Our results will rely on the following four assumptions.

Assumption B.1. $\mathcal{X} \subset \mathbb{R}^d$ is a compact subset with diameter

$$\mathfrak{d} := \sup_{x, y \in \mathcal{X}} |x - y|.$$

Assumption B.2. There exists $\kappa \in (0, 1)$ such that for any two probability measures π, ρ on \mathcal{X}

$$\mathcal{W}_k(\pi f, \rho f) \leq \kappa \mathcal{W}_k(\pi, \rho), \quad k = 1, 2.$$

Assumption B.3. The weight function $\omega^{(t)} : \mathcal{X} \rightarrow [\Delta, \Delta^{-1}]$ is 1-Lipschitz for all t .

Assumption B.4. There exists a $\lambda > 0$, such that for all $t \geq 0$ the unique optimal transport plan between $\alpha^{(t)}$ and $\beta^{(t)}$ is given by a deterministic, λ -Lipschitz map $\mathbf{T}^{(t)}$.

C. Auxiliary Results and Proof of Proposition 4.2

We start by establishing a couple of key auxiliary results which will be then used subsequently to establish Proposition 4.2.

C.1. Auxiliary Results

As per section 2.1, let $\mathcal{S}(\alpha_N, \beta_N)$ denote the collection of coupling matrices between $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$. We also denote entropy by H where $H(\mathbf{P}) = \sum_{i,j} p_{i,j} \log(1/p_{i,j})$ for $\mathbf{P} = (p_{i,j})_{i,j} \in \mathcal{S}(\alpha_N, \beta_N)$.

Lemma C.1. *The entropic radius, R_H , of simplex $\mathcal{U}(\alpha_N, \beta_N)$ may be bounded above as follows*

$$R_H := \max_{\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{S}(\alpha_N, \beta_N)} H(\mathbf{P}_1) - H(\mathbf{P}_2) \leq 2 \log(N)$$

Proof. Notice that $-H(\mathbf{P})$ is convex, so that $H(\mathbf{P})$ is concave.

$$\begin{aligned} \sum_{i,j} p_{i,j} \log\left(\frac{1}{p_{i,j}}\right) &= N^2 \sum_{i,j} \frac{1}{N^2} p_{i,j} \log\left(\frac{1}{p_{i,j}}\right) \\ &\leq N^2 H\left(\frac{1}{N^2} \sum_{i,j} p_{i,j}\right) = N^2 H(1/N^2) = N^2 \frac{1}{N^2} \log(N^2) = 2 \log(N). \end{aligned}$$

In addition since $p_{i,j} \leq 1$ for all i, j , we have that $H(\mathbf{P}) \geq 0$ and therefore we can bound

$$R_H = \max_{P_1, P_2 \in \mathcal{P}} H(\mathbf{P}_1) - H(\mathbf{P}_2) \leq \max_{\mathbf{P}_1 \in \mathcal{S}(\alpha_N, \beta_N)} H(\mathbf{P}_1) \leq 2 \log(N).$$

□

Lemma C.2. Let $\mathcal{X} \subset \mathbb{R}^d$ be compact with diameter $\mathfrak{d} > 0$. Suppose we are given two probability measures α, β on \mathcal{X} with a unique deterministic, λ -Lipschitz optimal transport map \mathbf{T} while $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$. We write $\mathcal{P}^{\text{OT},N}$, resp. $\mathcal{P}_\epsilon^{\text{OT},N}$, for an optimal coupling between α_N and β_N , resp. the ϵ -regularized optimal transport plan, between α_N and β_N . Then

$$\left[\int \|y - \mathbf{T}(x)\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(dx, dy) \right]^{1/2} \leq 2\lambda^{1/2} \mathcal{E}^{1/2} \left[\mathfrak{d}^{1/2} + \mathcal{E} \right]^{1/2} + \max\{\lambda, 1\} [\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)],$$

where

$$\mathcal{E} := \mathcal{E}(N, \epsilon, \alpha, \beta) := \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon \log(N)}.$$

Proof. From Corollary 3.8 from (Li & Nocettono, 2021)

$$\left[\int \|\mathbf{T}(x) - y\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(dx, dy) \right]^{1/2} \leq 2\lambda^{1/2} \sqrt{\tilde{e}_{N,\epsilon}} [\mathcal{W}_2(\alpha, \beta) + \tilde{e}_{N,\epsilon}]^{1/2} + \lambda \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta),$$

where λ is the Lipschitz constant of the optimal transport map \mathbf{T} sending α to β , and

$$\tilde{e}_{N,\epsilon} := \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \left[\int \|x - y\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(dx, dy) \right]^{1/2} - \mathcal{W}_2(\alpha_N, \beta_N). \quad (30)$$

From Proposition 4 of (Weed, 2018),

$$\sum_{i,j=1,\dots,N} p_{\epsilon,i,j}^{\text{OT}} |Y_i - X_j|^2 - \mathcal{W}_2^2(\alpha_N, \beta_N) \leq \epsilon R_H,$$

where R_H is the entropic radius as defined in Lemma C.1.

By Lemma C.1 we therefore have that

$$\int \|x - y\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(dx, dy) - \mathcal{W}_2^2(\alpha_N, \beta_N) \leq 2\epsilon \log(N).$$

Since $x \mapsto \sqrt{x}$ is sub-additive, for $r, s > 0$ we have that $\sqrt{r} - \sqrt{s} \leq \sqrt{r-s}$, whence

$$\left[\int \|x - y\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(dx, dy) \right]^{1/2} - \mathcal{W}_2(\alpha_N, \beta_N) \leq \sqrt{2\epsilon \log N}.$$

We thus have

$$\tilde{e}_{N,\epsilon} \leq \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon \log(N)}.$$

In addition, by Assumption B.1 we have that $\mathcal{W}_2(\alpha, \beta) \leq \mathfrak{d}^{1/2}$ and the result follows. □

C.2. Proof of Proposition 4.2

Proof of Proposition 4.2. By definition, we have $\tilde{\beta}_N(d\tilde{x}) = \int \alpha_N(dx) \delta_{\mathbf{T}_{N,\epsilon}(x)}(d\tilde{x})$ with $\mathbf{T}_{N,\epsilon}(x) := \int \tilde{x} \mathcal{P}_\epsilon^{\text{OT},N}(d\tilde{x}|x)$ while, as $\mathcal{P}_\epsilon^{\text{OT},N}$ belongs to $\mathcal{U}(\alpha_N, \beta_N)$, we also have $\beta_N(d\tilde{x}) = \int \alpha_N(dx) \mathcal{P}_\epsilon^{\text{OT},N}(d\tilde{x}|x)$. We then have for any 1-Lipschitz

function

$$\begin{aligned}
 |\beta_N(\psi) - \tilde{\beta}_N(\psi)| &= \left| \int \left[\int (\psi(\tilde{x}) - \psi(\mathbf{T}_{N,\epsilon}(x))) \mathcal{P}_\epsilon^{\text{OT},N}(\mathrm{d}\tilde{x}|x) \right] \alpha_N(\mathrm{d}x) \right| \\
 &\leq \iint |\psi(\tilde{x}) - \psi(\mathbf{T}_{N,\epsilon}(x))| \alpha_N(\mathrm{d}x) \mathcal{P}_\epsilon^{\text{OT},N}(\mathrm{d}\tilde{x}|x) \\
 &\leq \iint \|\tilde{x} - \mathbf{T}_{N,\epsilon}(x)\| \mathcal{P}_\epsilon^{\text{OT},N}(\mathrm{d}x, \mathrm{d}\tilde{x}) \\
 &\leq \left(\iint \|\tilde{x} - \mathbf{T}_{N,\epsilon}(x)\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(\mathrm{d}x, \mathrm{d}\tilde{x}) \right)^{\frac{1}{2}} \\
 &\leq \left(\iint \|\tilde{x} - \mathbf{T}(x)\|^2 \mathcal{P}_\epsilon^{\text{OT},N}(\mathrm{d}x, \mathrm{d}\tilde{x}) \right)^{\frac{1}{2}},
 \end{aligned}$$

where the final inequality follows from the fact that for any random vector V the mapping $v \mapsto \mathbb{E}[\|V - v\|^2]$ is minimized at $v = \mathbb{E}[V]$. The stated result is then obtained using Lemma C.2. \square

D. Proof of Proposition 4.3

For technical reasons, we analyse here a slightly modified PF algorithm where

$$\alpha_N^{(t)} = \frac{1}{N} \sum_{j=1}^N \delta_{X_t^j}, \quad X_t^j \stackrel{\text{i.i.d.}}{\sim} \tilde{\beta}_N^{(t-1)} f = \frac{1}{N} \sum_{j=1}^N f(\cdot | \tilde{X}_{t-1}^j). \quad (31)$$

instead of the standard version where one has

$$\alpha_N^{(t)} = \frac{1}{N} \sum_{j=1}^N \delta_{X_t^j}, \quad X_t^j \sim f(\cdot | \tilde{X}_{t-1}^j).$$

This slightly modified version of the bootstrap PF was analyzed for example in (Del Moral & Guionnet, 2001). The analysis does capture the additional error arising from the use of DET instead of resampling. Similar results should hold for the standard PF algorithm. The main technical reason for analysing this modified algorithm is our reliance on Theorem 2 of (Fournier & Guillin, 2015); analysing the standard PF algorithm requires a version of (Fournier & Guillin, 2015) for stratified sampling and will be done in future work.

Proposition D.1. Suppose that Assumptions B.1, B.2 and B.3 hold. Suppose also that given $\tilde{\beta}_N^{(t-1)}$, $\alpha_N^{(t)}$ is defined through (31). Define the functions

$$\begin{aligned}
 \mathcal{F}(x) &:= x + \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) x} \\
 f_d(x) &:= \begin{cases} x, & d < 4 \\ \frac{x}{\log(2+1/x)}, & d = 4 \\ x^{d/2}, & d > 4. \end{cases} \\
 \mathcal{F}_{N,\epsilon,\delta,d}(x) &:= \mathcal{F}\left(\kappa x + \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}\right), \\
 \frac{1}{\mathfrak{d}} \mathfrak{G}_{\epsilon,\delta,N,d}^2(x) &:= 2\lambda^{1/2} \left[\mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon \log N} \right]^{1/2} \left[\mathfrak{d}^{1/2} + \mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon \log N} \right]^{1/2} \\
 &\quad + \lambda \kappa \mathcal{F}_{N,\epsilon,\delta,d}(x) + \max\{\lambda, 1\} \mathcal{F}_{N,\epsilon,\delta,d}(x).
 \end{aligned} \quad (32)$$

Then for any $\epsilon, \delta > 0$ we have with probability at least $1 - \delta$, over the sampling step in (31), that

$$\mathcal{W}_2\left(\tilde{\beta}_N^{(t)}, \beta^{(t)}\right) \leq \mathfrak{G}_{\epsilon,\delta,N,d}\left[\mathcal{W}_2\left(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}\right)\right] \quad (33)$$

In particular if $\mathcal{W}_2(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}) \rightarrow 0$ and $\epsilon_N = o(1/\log(N))$ as $N \rightarrow \infty$ we have that

$$\mathcal{W}_2\left(\tilde{\beta}_N^{(t)}, \beta^{(t)}\right) \rightarrow 0,$$

in probability.

Proof of Proposition D.1. To keep notation concise we write for $N \geq 1$

$$\alpha_N := \alpha_N^{(t)}, \quad \alpha'_N := \tilde{\beta}_N^{(t-1)} f, \quad \beta_N := \beta_N^{(t)}, \quad \tilde{\beta}_N := \tilde{\beta}_N^{(t-1)}.$$

Controlling $\mathcal{W}_1(\beta_N, \beta)$. Let ψ be 1-Lipschitz. Without loss of generality we may assume that $\psi(0) = 0$ since otherwise we can remove a constant.

$$\begin{aligned} |\beta_N(\psi) - \beta(\psi)| &= \left| \frac{\alpha_N(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha(\omega)} \right| \\ &\leq \left| \frac{\alpha_N(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha_N(\omega)} \right| + \left| \frac{\alpha(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha(\omega)} \right| \\ &\leq \Delta^{-1} |\alpha_N(\omega\psi) - \alpha(\omega\psi)| + \Delta^{-2} \alpha(\omega\psi) |\alpha_N(\omega) - \alpha(\omega)|. \end{aligned}$$

At this stage notice that

$$|(\omega\psi)'| \leq |\omega'\psi| + |\omega\psi'| \leq \|\psi\|_\infty + \|\omega\|_\infty.$$

Notice that

$$|\psi(x)| = |\psi(x) - \psi(0)| \leq |x - 0| \leq \mathfrak{d}.$$

Therefore we have that

$$|(\omega\psi)'| \leq \mathfrak{d} + \Delta^{-1},$$

and thus $\omega\psi$ is $(\mathfrak{d} + \Delta^{-1})$ -Lipschitz. It follows that

$$\begin{aligned} |\beta_N(\psi) - \beta(\psi)| &\leq \Delta^{-1} |\alpha_N(\omega\psi) - \alpha(\omega\psi)| + \Delta^{-2} \alpha(\omega\psi) |\alpha_N(\omega) - \alpha(\omega)| \\ &\leq \Delta^{-1} (\mathfrak{d} + \Delta^{-1}) \mathcal{W}_1(\alpha_N, \alpha) + \Delta^{-3} \mathfrak{d} \mathcal{W}_1(\alpha_N, \alpha) \\ &=: K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha). \end{aligned}$$

Therefore we have that

$$\mathcal{W}_1(\beta_N, \beta) \leq K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha). \tag{34}$$

Notice that using the compactness of the state space we easily get also that

$$\mathcal{W}_2(\beta_N, \beta) \leq \sqrt{\mathfrak{d} \mathcal{W}_1(\beta_N, \beta)} \leq \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha)} \leq \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) \mathcal{W}_2(\alpha_N, \alpha)}, \tag{35}$$

since clearly $\mathcal{W}_1(\rho, \sigma) \leq \mathcal{W}_2(\rho, \sigma)$ for any two probability measures ρ, σ .

Controlling $\mathcal{W}_1(\tilde{\beta}_{N,\epsilon}, \beta)$. Again supposing ψ is 1-Lipschitz, and $\psi(0) = 0$, consider

$$\begin{aligned} |\tilde{\beta}_N(\psi) - \tilde{\beta}(\psi)| &= \left| \int \psi(\mathbf{T}_{N,\epsilon}(x)) \alpha_N(dx) - \int \psi(\mathbf{T}(x)) \alpha(dx) \right| \\ &\leq \left| \int \psi(\mathbf{T}_{N,\epsilon}(x)) \alpha_N(dx) - \int \psi(\mathbf{T}(x)) \alpha_N(dx) \right| \\ &\quad + \left| \int \psi(\mathbf{T}(x)) \alpha_N(dx) - \int \psi(\mathbf{T}(x)) \alpha(dx) \right| \end{aligned}$$

For the second term, using the fact that \mathbf{T} and ψ are λ - and 1-Lipschitz respectively, we have that $\psi \circ \mathbf{T}$ is λ -Lipschitz and therefore

$$\left| \int \psi(\mathbf{T}(x)) \alpha_N(dx) - \int \psi(\mathbf{T}(x)) \alpha(dx) \right| \leq \lambda \mathcal{W}_1(\alpha_N, \alpha) \leq \lambda \mathcal{W}_2(\alpha_N, \alpha),$$

where we used Assumption B.2 for that last inequality. For the first term recall that using Cauchy-Schwarz and Jensen we get

$$\begin{aligned}
 & \left| \int \psi(\mathbf{T}_{N,\epsilon}(x)) \alpha_N(dx) - \int \psi(\mathbf{T}(x)) \alpha_N(dx) \right| \\
 & \leq \int |\mathbf{T}_{N,\epsilon}(x) - \mathbf{T}(x)| \alpha_N(dx) \\
 & \leq \int \left| \int y \mathcal{P}_{N,\epsilon}(x, dy) - \mathbf{T}(x) \right| \alpha_N(dx) \\
 & \leq \iint |y - \mathbf{T}(x)| \alpha_N(dx) \mathcal{P}_{N,\epsilon}(x, dy) \\
 & \leq \left[\iint |y - \mathbf{T}(x)|^2 \alpha_N(dx) \mathcal{P}_{N,\epsilon}(x, dy) \right]^{1/2}.
 \end{aligned}$$

Here we can directly apply Lemma C.2 to obtain

$$\begin{aligned}
 & \left[\iint |y - \mathbf{T}(x)|^2 \alpha_N(dx) \mathcal{P}_{N,\epsilon}(x, dy) \right]^{1/2} \\
 & \leq 2\lambda^{1/2} \mathcal{E}^{1/2} \left[\mathfrak{d}^{1/2} + \mathcal{E} \right]^{1/2} + \max\{\lambda, 1\} [\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)],
 \end{aligned}$$

where

$$\mathcal{E} := \mathcal{E}(n, \epsilon, \alpha, \beta) := \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon \log(N)}.$$

From (35) we have that

$$\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) \leq \mathcal{W}_2(\alpha_N, \alpha) + \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) \mathcal{W}_2(\alpha_N, \alpha)}.$$

Next we want to bound $\mathcal{W}_2(\alpha_N, \alpha)$. Notice first that

$$\mathcal{W}_2(\alpha_N, \alpha) \leq \mathcal{W}_2(\alpha_N, \alpha'_N) + \mathcal{W}_2(\alpha'_N, \alpha) \leq \mathcal{W}_2(\alpha_N, \alpha'_N) + \kappa \mathcal{W}_2(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}),$$

by Assumption B.2.

To control the other term we use (Fournier & Guillin, 2015) to obtain a high probability bound on $\mathcal{W}_2(\alpha_N, \alpha'_N)$. In particular, using Theorem 2 from (Fournier & Guillin, 2015), with $\alpha = \infty$ since we are in a compact domain, that for some positive constants C, c we have

$$\mathbb{P}[\mathcal{W}_2^2(\alpha_N, \alpha'_N) \geq x] \leq C \exp[-cNf_d^2(x)], \quad (36)$$

where

$$f_d(x) := \begin{cases} x, & d < 4 \\ \frac{x}{\log(2+1/x)}, & d = 4 \\ x^{d/2}, & d > 4. \end{cases} \quad (37)$$

In particular, for any $\delta > 0$, with probability at least $1 - \delta$ over the sampling step in F_N we have that

$$\mathcal{W}_2(\alpha_N, \alpha'_N) \leq \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}. \quad (38)$$

Assuming that $d \geq 4$ the rate then is of order $N^{-1/d}$ as expected.

Therefore with probability at least $1 - \delta$ over the sampling step we have that

$$\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) \leq \mathcal{F}_{N,\epsilon,\delta,d}\left(\mathcal{W}_2(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)})\right),$$

where

$$\mathcal{F}_{N,\epsilon,\delta,d}(x) = \mathcal{F}\left(\kappa x + \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}\right), \quad \mathcal{F}(x) := x + \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) x} \quad (39)$$

Thus overall we have with probability at least $1 - \delta$ over the sample

$$\mathcal{W}_2(\tilde{\beta}_{N,\epsilon}, \tilde{\beta}) \leq \sqrt{\mathfrak{d}\mathcal{W}_1(\tilde{\beta}_{N,\epsilon}, \tilde{\beta})} \leq \mathfrak{G}_{\epsilon,\delta,N,d} \left(\mathcal{W}_2 \left(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)} \right) \right),$$

where

$$\begin{aligned} \frac{1}{\mathfrak{d}} \mathfrak{G}_{\epsilon,\delta,N,d}^2(x) &:= 2\lambda^{1/2} \left[\mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon \log N} \right]^{1/2} \left[\mathfrak{d}^{1/2} + \mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon \log N} \right]^{1/2} \\ &\quad + \lambda\kappa \mathcal{F}_{N,\epsilon,\delta,d}(x) + \max\{\lambda, 1\} \mathcal{F}_{N,\epsilon,\delta,d}(x). \end{aligned}$$

In particular notice that if we set $\epsilon_N = o(1/\log N)$ and $x_N = o(1)$ we have

$$\mathfrak{G}_{\epsilon_N,\delta,N,d}(x_N) \rightarrow 0.$$

Therefore, notice that if $\epsilon_N = o(1/\log N)$ and $\mathcal{W}_2(\mu_N, \mu) \rightarrow 0$, then for any $x > 0$ we have that

$$\mathbb{P} \left[\mathcal{W}_2(\tilde{\beta}_{N,\epsilon}, \tilde{\beta}) \geq x \right] \leq \mathbb{P}[\mathcal{W}_2(\alpha'_N, \alpha_N) \geq x'],$$

for some x' that does not depend on N , where the probability is over the sampling step. The convergence in probability follows. \square

Proposition D.2. Let $\mu_N = \frac{1}{N} \sum_{i=1}^N \delta_{X_1^i}$ where $X_1^i \stackrel{\text{i.i.d.}}{\sim} \mu := q(\cdot | y_1)$ for $i \in [N]$ and suppose that for $t \geq 1$, $\alpha_N^{(t)}$ is defined through (31). Under Assumptions B.1, B.2, B.3 and B.4, for any $\delta > 0$, with probability at least $1 - 2\delta$ over the sampling steps, for any bounded 1-Lipschitz ψ , for any $t \in [1 : T]$, the approximations of the filtering distributions and log-likelihood computed by DPF satisfy

$$|\tilde{\beta}_N^{(t)}(\psi) - \beta^{(t)}(\psi)| \leq \mathfrak{G}_{\epsilon,\delta/T,N,d}^{(t)} \left(\sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)} \right) \quad (40)$$

$$\left| \log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})} \right| \leq \frac{\kappa}{\Delta} \max_{t \in [1:T]} \text{Lip}[g(y_t | \cdot)] \sum_{t=1}^T \mathfrak{G}_{\epsilon,\delta/T,N,d}^{(t)} \left(\sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)} \right) \quad (41)$$

where C is a finite constant independent of T , $\mathfrak{G}_{\epsilon,\delta/T,N,d}$, f_d are defined in (32), and $\text{Lip}[f]$ is the Lipschitz constant of the function f . $\mathfrak{G}_{\epsilon,\delta/T,N,d}^{(t)}$ denotes the t -repeated composition of function $\mathfrak{G}_{\epsilon,\delta/T,N,d}$. In particular, if we set $\epsilon_N = o(1/\log N)$

$$\left| \log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})} \right| \rightarrow 0,$$

in probability.

Proof of Proposition D.2. Following the proof of Proposition D.1, we define $\alpha_N^{(t)'} = \tilde{\beta}_N^{(t-1)} f$ and for $t \in [1 : T]$, the events

$$A_t := \mathcal{W}_2 \left(\alpha_N^{(t)}, \alpha_N^{(t)'} \right) \leq \sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)}.$$

We know from Theorem 2 in (Fournier & Guillin, 2015) that $\mathbb{P}(A_t) \geq 1 - \delta/T$, where the probability is over the sampling step. In particular we have that

$$\mathbb{P} \left[\bigcap_{t=1}^T A_t \right] = 1 - \mathbb{P} \left[\bigcup_{t=1}^T A_t^c \right] \geq 1 - \sum_{t=1}^N \mathbb{P}[A_t^c] \geq 1 - T \frac{\delta}{T} = 1 - \delta.$$

Notice that on the event $\bigcap_{t=1}^T A_t$, iterating the bound (33) we have

$$\mathcal{W}_2 \left(\tilde{\beta}_N^{(t)}, \beta^{(t)} \right) \leq \mathfrak{G}_{\epsilon,\delta/T,N,d}^{(t)} (\mathcal{W}_2(\mu_N, \mu)),$$

with probability at least $1 - \delta$. Again by Theorem 2 in (Fournier & Guillin, 2015) we have that with probability at least $1 - \delta$

$$\mathcal{W}_2(\mu_N, \mu) \leq \sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)}.$$

Therefore with probability at least $1 - 2\delta$ we have

$$\mathfrak{G}_{\epsilon, \delta/T, N, d}^{(t)} \left(\sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)} \right).$$

It remains to prove (41). Note that $|\log(x) - \log(y)| \leq \frac{|x-y|}{\min\{x,y\}}$ for any $x, y > 0$ so

$$\begin{aligned} |\log \hat{p}(y_{1:T}) - \log p(y_{1:T})| &\leq \sum_{t=1}^T |\log \hat{p}(y_t | y_{1:t-1}) - \log p(y_t | y_{1:t-1})| \\ &\leq \sum_{t=1}^T \left| \frac{\hat{p}(y_t | y_{1:t-1}) - p(y_t | y_{1:t-1})}{\min(\hat{p}(y_t | y_{1:t-1}), p(y_t | y_{1:t-1}))} \right| \\ &\leq \Delta^{-1} \sum_{t=1}^T |\hat{p}(y_t | y_{1:t-1}) - p(y_t | y_{1:t-1})| \end{aligned} \tag{42}$$

where Δ is defined in Assumption B.3.

The term in line (42) may be written as follows

$$\begin{aligned} &\hat{p}(y_t | y_{1:t-1}) - p(y_t | y_{1:t-1}) \\ &= \iint g(y_t | x_t) f(dx_t | \tilde{x}_{t-1}) \tilde{\beta}_N^{(t-1)}(d\tilde{x}_{t-1}) - \iint g(y_t | x_t) f(dx_t | \tilde{x}_{t-1}) \tilde{\beta}^{(t-1)}(d\tilde{x}_{t-1}) \\ &= \tilde{\beta}_N^{(t-1)}(h) - \beta^{(t-1)}(h) \end{aligned}$$

for $\Delta^2 \leq h(x) := \int g(y_t | x') f(x' | x) dx' \leq \Delta^{-2}$. At this point notice also that

$$\begin{aligned} h(x) - h(x') &= \int f(dw | x) g(y_t | w) - \int f(dw | x') g(y_t | w) \\ &= \int \delta_x(dz) \int f(dw | z) g(y_t | w) - \int \delta_{x'}(dz) \int f(dw | z) g(y_t | w) \\ &= [\delta_x f][g(y_t | \cdot)] - [\delta_{x'} f][g(y_t | \cdot)] \\ &\leq \text{Lip}[g(y_t | \cdot)] \mathcal{W}_1(\delta_x f, \delta_{x'} f) \leq \kappa \text{Lip}[g(y_t | \cdot)] \mathcal{W}_1(\delta_x, \delta_{x'}) = \kappa \text{Lip}[g(y_t | \cdot)] |x - x'|, \end{aligned}$$

by Assumption B.2. It follows therefore that h is Lipschitz and therefore that

$$\hat{p}(y_t | y_{1:t-1}) - p(y_t | y_{1:t-1}) = \tilde{\beta}_N^{(t-1)}(h) - \beta^{(t-1)}(h) \leq \kappa \text{Lip}[g(y_t | \cdot)] \mathcal{W}_1(\beta_N^{(t-1)}, \beta^{(t-1)}).$$

Combining (40) and (42), and using the fact that $\mathcal{W}_1 \leq \mathcal{W}_2$, we thus get

$$\begin{aligned} |\log \hat{p}(y_{1:T}) - \log p(y_{1:T})| &\leq \Delta^{-1} \kappa \sum_{t=1}^T \text{Lip}[g(y_t | \cdot)] \mathcal{W}_1(\tilde{\beta}_N^{(t-1)}, \beta^{(t)}) \\ &\leq \Delta^{-1} \kappa \max_{t \in [1:T]} \text{Lip}[g(y_t | \cdot)] \sum_{t=1}^T \mathfrak{G}_{\epsilon, \delta/T, N, d}^{(t)} \left(\sqrt{f_d^{-1} \left(\frac{\log(CT/\delta)}{cN} \right)} \right), \end{aligned}$$

where the last inequality holds with probability at least $1 - \delta$ over the sampling steps.

The convergence in probability follows from the corresponding statement of Proposition D.1. \square

E. Additional Experiments and Details

E.1. Linear Gaussian model

We first consider the following 2-dimensional linear Gaussian SSM for which exact inference can be carried out using Kalman techniques:

$$X_t | \{X_{t-1} = x\} \sim \mathcal{N}(\text{diag}(\theta_1 \theta_2)x, 0.5\mathbf{I}_2), \quad Y_t | \{X_t = x\} \sim \mathcal{N}(x, 0.1 \cdot \mathbf{I}_2). \quad (43)$$

We simulate $T = 150$ observations using $\theta = (\theta_1, \theta_2) = (0.5, 0.5)$. As a result, we expect in these scenarios that the filtering distribution $p_\theta(x_t | y_{1:t})$ is not too distinct from the smoothing distribution $p_\theta(x_t | y_{1:T})$ as the latent process is mixing quickly. From Proposition 4.1, this is thus a favourable scenario for methods ignoring resampling terms in the gradient as the bias should not be very large. Figure 1, displayed earlier, shows $\ell(\theta)$ obtained by Kalman and $\hat{\ell}(\theta; \mathbf{u})$ computed regular PF and DPF for the same number $N = 25$ of particles using $q_\phi(x_t | x_{t-1}, y_t) = f_\theta(x_t | x_{t-1})$. The corresponding gradient vector fields are given in Figure 1, where the gradient is computed using the biased gradient from (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) for regular PF.

We now compare the performance of the estimators $\hat{\theta}_{\text{SMLE}}$ (for DPF) and $\hat{\theta}_{\text{ELBO}}$ (for both regular PF and DPF) learned using gradient with learning rate 10^{-4} on 100 steps, using $N = 25$ for DPF and $N = 500$ for regular PF, to $\hat{\theta}_{\text{MLE}}$ computed using Kalman derivatives. We simulate $M = 50$ realizations of $T = 150$ observations using $\theta = (\theta_1, \theta_2) = (0.5, 0.5)$. The ELBO stochastic gradient estimates are computed using biased gradient estimates of $\ell_{\text{ELBO}}(\theta)$ ignoring the contributions of resampling steps as in (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) (we recall that unbiased estimates suffer from very high variance) and unbiased gradients of $\ell^{\text{ELBO}}(\theta)$ using DPF. We average B parallel PFs to reduce the variance of these gradients of the ELBO and also B PFs (with fixed random seeds) to compute the gradient of $\hat{\ell}_{\text{SMLE}(\theta; \mathbf{u}_{1:B})} := \frac{1}{B} \sum_{b=1}^B \hat{\ell}(\theta; \mathbf{u}_b)$. The results are given in Table 4. For this example, $\hat{\theta}_{\text{ELBO}}^{\text{DPF}}$ maximizing $\ell_{\text{DPF}}^{\text{ELBO}}(\theta)$ outperforms $\hat{\theta}_{\text{ELBO}}^{\text{PF}}$ and $\hat{\theta}_{\text{SMLE}}$. However, as B increases, $\hat{\theta}_{\text{SMLE}}$ gets closer to $\hat{\theta}_{\text{ELBO}}^{\text{DPF}}$ which is to be expected as $\hat{\ell}_{\text{SMLE}(\theta; \mathbf{u}_{1:B})} \rightarrow \ell^{\text{ELBO}}(\theta)$. In Table 4, the Root Mean Square Error (RMSE) is defined as $\sqrt{\frac{1}{M} \sum_{i=1}^M \sum_{k=1}^M (\hat{\theta}_i^k - \hat{\theta}_{\text{MLE},i}^k)^2}$.

Table 4. $10^3 \times \text{RMSE}^4$ over 50 datasets - lower is better

B	$\hat{\theta}_{\text{ELBO}}^{\text{PF}}$	$\hat{\theta}_{\text{ELBO}}^{\text{DPF}}$	$\hat{\theta}_{\text{SMLE}}$
1	1.94	1.30	7.94
4	2.40	1.35	3.28
10	2.80	1.37	2.18

E.2. Variational Recurrent Neural Network

$N = 32$ particles were used for training, with a regularization parameter of $\epsilon = 0.5$. The ELBO (scaled by sequence length) was used as the training objective to maximise for each resampling/ DET procedure. The ELBO evaluated on test data using $N = 500$ particles and multinomial resampling. Resampling / DET operations were carried out when effective sample (ESS) size fell below $N/2$. Learning rate 0.001 was used with the Adam optimizer.

Recall the state-space model is given by

$$\begin{aligned} (R_t, O_t) &= \text{RNN}_\theta(R_{t-1}, Y_{1:t-1}, E_\theta(Z_{t-1})), \\ Z_t &\sim \mathcal{N}(\mu_\theta(O_t), \sigma_\theta(O_t)), \\ \hat{p}_t &= h_\theta(E_\theta(Z_t), O_t), \\ Y_t | X_t &\sim \text{Ber}(\hat{p}_t). \end{aligned}$$

Network architectures and data preprocessing steps were based loosely on (Maddison et al., 2017). Given the low volume of data and sparsity of the observations, relatively small neural networks were considered to prevent overfitting, larger neural

⁴The Root Mean Square Error (RMSE) is defined as $\sqrt{\frac{1}{M} \sum_{i=1}^M \sum_{k=1}^M (\hat{\theta}_i^k - \hat{\theta}_{\text{MLE},i}^k)^2}$.

networks are considered in the more complex robotics experiments. R_t is of dimension $d_r = 16$, Z_t is of dimension $d_z = 8$. E_θ is a single layer fully connected network with hidden layer of width 16, output of dimension 16 and RELU activation.

μ_θ and σ_θ are both fully connected neural networks with two hidden layers, each of 16 units and RELU activation, the activation function is not applied to the final output of μ_θ but the softplus is applied to the output of σ_θ , which is the diagonal entries of the covariance matrix of the normal distribution that is used to sample Z_t .

h_θ is a single layer fully connected network with two hidden layers, each of width 16 and RELU activation. The final output is not put through the RELU and is instead used as the logits for the Bernoulli distribution of observations.

E.3. Robot Localization

Similar to the VRNN example, $N = 32$ particles were used for training, with a regularization parameter of $\epsilon = 0.5$ and resampling / DET operations were carried out when ESS size fell below $N/2$. Learning rate 0.001 was used with the Adam optimizer.

Network architectures and data preprocessing were based loosely on (Jonschkowski et al., 2018). There are 3 neural networks being considered:

- Encoder E_θ maps RBG 24×24 pixel images, hence dimension $3 \times 24 \times 24$, to encoding of size $d_E = 128$. This network consists of a convolutional network (CNN) of kernel size 3 and a single layer fully connected network of hidden width 128 and RELU activation.
- Decoder D_θ maps encoding back to original image. This consists of a fully connected neural network with three hidden layers of width 128 and RELU activation function. This is followed by a transposed convolution network with matching specification to the CNN in the encoder, to return an output with the same dimension as observation images, $3 \times 24 \times 24$.
- Network G_θ maps the state $S_t = (X_t^{(1)}, X_t^{(2)}, \gamma_t)$ to encoding of dimension 128. First angle γ_t was converted to $\sin(\gamma_t), \cos(\gamma_t)$. Then the augmented state $(X_t^{(1)}, X_t^{(2)}, \sin(\gamma_t), \cos(\gamma_t))$ was passed to a 3 layer fully connected network with hidden layers of dimensions 16, 32, 64 and RELU activation function, with final output of dimension 128.