Differentiable Particle Filtering via Entropy-Regularized Optimal Transport

Adrien Corenflos [*1] **James Thornton** [*2] **George Deligiannidis** [2] **Arnaud Doucet** [2]

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 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 $t \geq 1$

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

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 $y_{1:t} \coloneqq (y_1, ..., y_t)$ where

$$p_\theta(x_t|y_{1:t-1}) = \int f_\theta(x_t|x_{t-1}) p_\theta(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1},$$

$$p_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t) p_\theta(x_t|y_{1:t-1})}{\int g_\theta(y_t|x_t) p_\theta(x_t|y_{1:t-1}) \mathrm{d}x_t},$$

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

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 $p_\theta(y_1|y_0) \coloneqq \int g_\theta(y_1|x_1)\mu_\theta(x_1)\mathrm{d}x_1$ and for $t \geq 2$

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

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 $[N] \coloneqq \{1, ..., N\}$, 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 $\mathbf{w}_t \coloneqq (w_t^1, ..., w_t^N)$ 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 $N \to \infty$ (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).

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

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Algorithm 1 Standard Particle Filter

1: Sample $X_1^i \overset{\text{i.i.d.}}{\sim} q_\phi(\cdot|y_1)$ 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** $t = 2, ..., T$ **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.

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

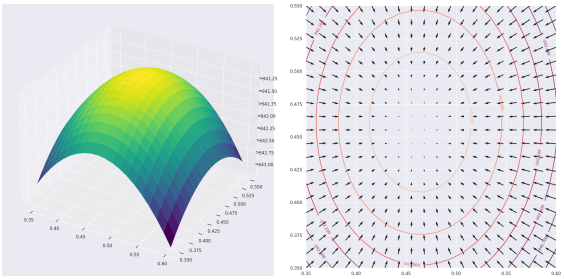
for any $\psi : \mathcal{X} \to \mathbb{R}$. This property guarantees $\exp(\hat{\ell}(\theta))$ is an unbiased estimate of the likelihood $\exp(\ell(\theta))$ for any N.

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.

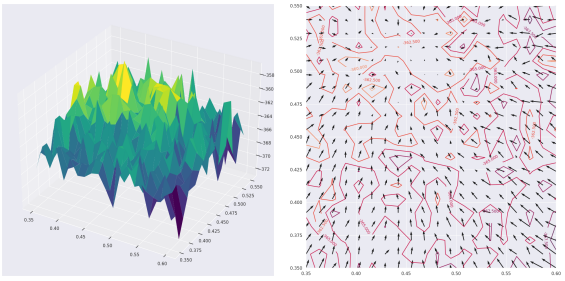
1.3. Related Work and Contributions

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 trick[1]. 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.

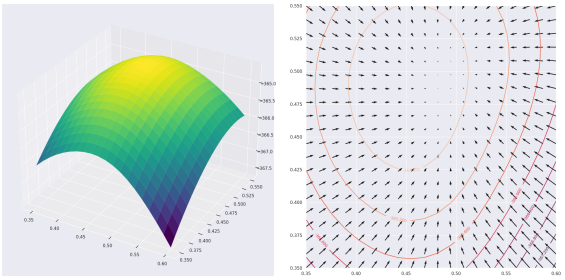
[1]For example, multinomial resampling relies on N uniform random variables.



(a) Kalman Filter



(b) Standard PF



(c) Differentiable PF

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. DeJong 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 proposal distribution typically used in the marginal PF is the mixture distribution $q_\phi(x_t) := \frac{1}{N}\sum_{i=1}^N q_\phi(x_t|\tilde{X}_{t-1}^i, y_t)$ from which one cannot sample smoothly in general. 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 be sampled from (e.g. using the reparameterization trick) but not evaluated pointwise (Murray et al., 2013), as the importance weight would be intractable.

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 $\mathbb{E}_{\mathbf{U}}[\exp\left(\hat{\ell}(\theta;\phi,\mathbf{U})\right)] = \exp(\ell(\theta))$ is an unbiased estimate of $\exp(\ell(\theta))$ for any N, ϕ for standard PF, then one has indeed by Jensen's inequality

$$\ell^{\mathrm{ELBO}}(\theta,\phi) \coloneqq \mathbb{E}_{\mathbf{U}}[\hat{\ell}(\theta;\phi,\mathbf{U})] \leq \ell(\theta). \tag{3}$$

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 $\ell^{\mathrm{ELBO}}(\theta,\phi) \to \ell(\theta)$ as $N \to \infty$ 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).

The contributions of this paper are four-fold.

- We propose the first fully Differentiable Particle Filter (DPF), which unlike (DeJong et al., 2013), 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

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

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 $\mathcal{P}^{\mathrm{OT}}$. Any element $\mathcal{P} \in \mathcal{U}(\alpha,\beta)$ allows one to "transport" α to β (and vice-versa) i.e.

$$\beta(\mathrm{d}v) = \int \mathcal{P}(\mathrm{d}u, \mathrm{d}v) = \int \mathcal{P}(\mathrm{d}v|u)\alpha(\mathrm{d}u).$$

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

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

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

$$\mathcal{P}(\mathrm{d}u,\mathrm{d}v) = \textstyle\sum_{i,j} p_{i,j}\delta_{u_i}(\mathrm{d}u)\delta_{v_j}(\mathrm{d}v),$$

$c_{i,j} = ||u_i - v_j||^2$, $\mathbf{P} = (p_{i,j})_{i,j\in[N]}$ and $\mathcal{S}(\mathbf{a},\mathbf{b})$ is the following set of matrices

$$\mathcal{S}(\mathbf{a},\mathbf{b}) = \left\{\mathbf{P}\in[0,1]^{N\times N} : \sum_{j=1}^N p_{i,j} = a_i,\ \sum_{i=1}^N p_{i,j} = b_j\right\}.$$

In such cases, one has

$$\mathcal{P}(\mathrm{d}v|u = u_i) = \textstyle\sum_j a_i^{-1}p_{i,j}\delta_{v_j}(\mathrm{d}v). \tag{6}$$

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

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

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 $\alpha^{(t)} := \int q_\phi(x_t|x_{t-1}, y_t) p_\theta(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1}$ 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 $\mathcal{P}^{\mathrm{OT}}(\mathrm{d}x, \mathrm{d}x') = \alpha^{(t)}(\mathrm{d}x)\delta_{\mathbf{T}^{(t)}(x)}(\mathrm{d}x')$ where $\mathbf{T}^{(t)} : \mathcal{X} \to \mathcal{X}$ 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 one 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

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

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 $N \to \infty$, $\mathbf{T}_N^{(t)}(X_t^i) \to \mathbf{T}^{(t)}(X_t^i)$ 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)

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

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

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

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

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

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

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

One may then recover the regularized transport matrix as

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

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

Function $\mathbf{Potentials}(\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{v})$
Local variables: $\mathbf{f}, \mathbf{g} \in \mathbb{R}^N$
Initialize: $\mathbf{f} = \mathbf{0}, \mathbf{g} = \mathbf{0}$
Set $\mathbf{C} \leftarrow \mathbf{u}\mathbf{u}^t + \mathbf{v}\mathbf{v}^t - 2\mathbf{u}\mathbf{v}^t$
while stopping criterion not met **do**
  for $i \in [N]$ **do**
    $f_i \leftarrow \frac{1}{2}\left(f_i + \mathcal{T}_\epsilon(\mathbf{b}, \mathbf{g}, \mathbf{C}_{i:})\right)$
    $g_i \leftarrow \frac{1}{2}\left(g_i + \mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}, \mathbf{C}_{:i})\right)$
  end for
end while
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 $\mathbf{P}_\epsilon^{\mathrm{OT}}$ 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

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

Algorithm 3 DET Resampling

Function $\mathbf{EnsembleTransform}(\mathbf{X}, \mathbf{w}, N)$
$\mathbf{f}, \mathbf{g} \leftarrow \mathbf{Potentials}(\mathbf{w}, \frac{1}{N}\mathbf{1}, \mathbf{X}, \mathbf{X})$
for $i \in [N]$ **do**
  for $j \in [N]$ **do**
    $p^{\mathrm{OT}}_{\epsilon,i,j} = \frac{w_i}{N} \exp\left(\frac{f_i + g_j - c_{i,j}}{\epsilon}\right)$
  end for
end for
Return $\tilde{\mathbf{X}} = N\mathbf{P}^{\mathrm{OT}}_{\epsilon}\mathbf{X}$

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 $\mathbf{P}^{\mathrm{OT}}$, $\mathbf{P}^{\mathrm{OT}}_{\epsilon}$ is sensitive to the scale of $\mathbf{X}_t$. To mitigate this sensitivity, one may compute $\delta(\mathbf{X}_t) = \sqrt{d_x}\max_{k\in[d_x]} \mathrm{std}_i(X^i_{t,k})$ 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.

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 $\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)$ ignoring resampling terms considered in (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018) converges as $N \to \infty$ to*

$$\mathbb{E}[\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta,\phi)] \to \int \nabla_\theta \log p_\theta(x_1, y_1)\, p_\theta(x_1|y_1)\mathrm{d}x_1 + \sum_{t=2}^{T} \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1})\, p_\theta(x_{t-1:t}|y_{1:t})\mathrm{d}x_{t-1:t}$$

whereas Fisher's identity yields

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

Hence, whereas we have $\nabla_\theta \ell^{\mathrm{ELBO}}(\theta,\phi) \to \nabla_\theta \ell(\theta)$ as $N \to \infty$ under regularity assumptions, the asymptotic bias of $\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta,\phi)$ 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 $y_{t:T}$, 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 & Nochetto, 2021) and (Weed, 2018). We use the notation $\nu(\psi) \coloneqq \int \psi(x)\nu(\mathrm{d}x)$ 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 $\tilde{\beta}_N = \sum_{i=1}^N a_i \delta_{\tilde{X}^i_{N,\epsilon}}$ where $\tilde{\mathbf{X}}_{N,\epsilon} = \Delta^{-1}\mathbf{P}^{\mathrm{OT}}_{\epsilon}\mathbf{X}$ for $\Delta = \mathrm{diag}(a_1, ..., a_N)$ and $\mathbf{P}^{\mathrm{OT}}_{\epsilon}$ is the transport matrix corresponding to the ϵ-regularized OT coupling, $\mathcal{P}^{\mathrm{OT},N}_{\epsilon}$, 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 $\mathbf{T}$*

[2] Here $\mathbf{U}$ 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.

between them. Then for any bounded 1*-Lipschitz function* ψ*, we have*

$$\left|\beta_N(\psi) - \tilde{\beta}_N(\psi)\right| \leq 2\lambda^{1/2}\mathcal{E}^{1/2}\left[\mathfrak{d}^{1/2} + \mathcal{E}\right]^{1/2} + \max\{\lambda, 1\}\left[\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)\right], \quad (14)$$

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 $\mathcal{W}_2(\alpha_N, \alpha), \mathcal{W}_2(\beta_N, \beta) \to 0$ and we choose $\epsilon_N = o(1/\log N)$ the bound given in (14) vanishes with $N \to \infty$. 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*

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

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

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* $\mathrm{Lip}[f]$ *is the Lipschitz constant of the function* f*, and* $\mathfrak{G}^{(t)}_{N,\epsilon}$, 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*

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

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 & Nochetto, 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, $\ell_\epsilon^{\mathrm{ELBO}}(\theta, \phi) := \mathbb{E}_{\mathbf{U}}[\hat{\ell}_\epsilon(\theta; \phi, \mathbf{U})]$. is actually a valid ELBO. However in all our experiments, see e.g. Section 5.1, $|\ell_\epsilon^{\mathrm{ELBO}}(\theta, \phi) - \ell^{\mathrm{ELBO}}(\theta, \phi)|$ is significantly smaller than $\ell(\theta) - \ell^{\mathrm{ELBO}}(\theta, \phi)$ so $\ell_\epsilon^{\mathrm{ELBO}}(\theta, \phi) < \ell(\theta)$. 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

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

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

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:

$$X_{t+1}|\{X_t = x\} \sim \mathcal{N}\left(\mathbf{A}x, \mathbf{I}_{d_x}\right), \quad (15)$$
$$Y_t|\{X_t = x\} \sim \mathcal{N}(\mathbf{I}_{d_y,d_x}x, \mathbf{I}_{d_y}), \quad (16)$$

with $\mathbf{A} = (0.42^{|i-j|+1})_{1\le i,j\le d_x}$, $\mathbf{I}_{d_y,d_x}$ 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

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

with $\Delta_\phi = \text{diag}(\phi_1, \ldots, \phi_{d_x})$ and a $d_x \times d_y$ matrix $\Gamma_\phi = \text{diag}_{d_x,d_y}(\phi_1, \ldots, \phi_{d_x})$ 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 (15)-(16), 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_θ, h_θ, μ_θ, σ_θ 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 $q_\phi(x_t|x_{t-1}, y_t) = f_\theta(x_t|x_{t-1})$.

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

The VRNN model is trained by maximizing $\ell_\epsilon^{\text{ELBO}}(\theta)$ using DPF. We compare this to the same model trained by

Table 2. ELBO ± Standard Deviation evaluated using Test Data.

	MUSEDATA	JSB	NOTTINGHAM
DPF	$\mathbf{-7.59}_{\pm 0.01}$	$\mathbf{-7.67}_{\pm 0.08}$	$\mathbf{-3.79}_{\pm 0.02}$
PF	$-7.60_{\pm 0.06}$	$-7.92_{\pm 0.13}$	$-3.81_{\pm 0.02}$
SPF	$-7.73_{\pm 0.14}$	$-8.17_{\pm 0.07}$	$-3.91_{\pm 0.05}$

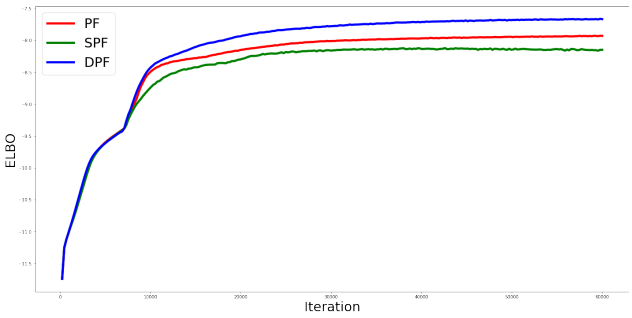


Figure 2. ELBO during training, evaluated on Test Data for JSB.

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 Ber$(\hat{p}_t)$, this recovers the negative log-predictive cross-entropy.

Figure 2 and Table 2 illustrate 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 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

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

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):

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

where $X_t^\star$ 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 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 3. 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.

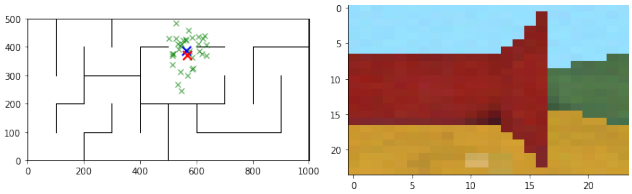
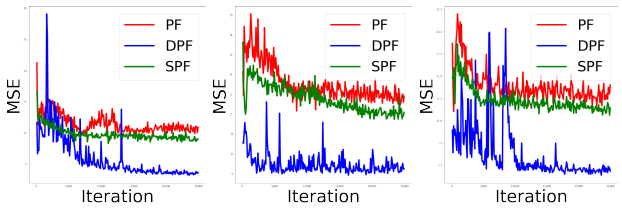


Figure 3. Left: Particles $(X_t^{(1),i}, X_t^{(2),i})$ (green), PF estimate of $\mathbb{E}[X_t|y_{1:t}]$ (blue), true state X_t^* (red). Right: Observation, O_t.

We compare our method, DPF, to regular PF used in (Maddison 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 4, 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 5, 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.

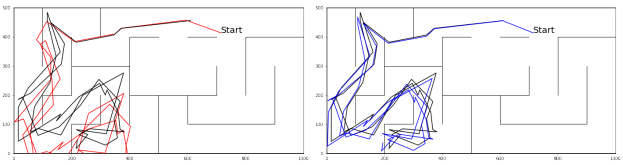
Table 3. MSE and $\pm$ Standard Deviation evaluated on Test Data: Lower is better

	MAZE 1	MAZE 2	MAZE 3
DPF	$\mathbf{3.55}_{\pm \mathbf{0.20}}$	$\mathbf{4.65}_{\pm 0.50}$	$\mathbf{4.44}_{\pm \mathbf{0.26}}$
PF	$10.71_{\pm 0.45}$	$11.86_{\pm 0.57}$	$12.88_{\pm 0.65}$
SPF	$9.14_{\pm 0.39}$	$10.12_{\pm \mathbf{0.40}}$	$11.42_{\pm 0.37}$



(a) Maze 1 (b) Maze 2 (c) Maze 3

Figure 4. MSE of PF (red), SPF (green) and DPF (blue) estimates, evaluated on test data during training.



(a) Standard PF (b) Differentiable PF

Figure 5. Illustrative Example: PF estimate of path compared to true path (black) on a single 50-step trajectory from test data.

6. Discussion

This paper introduces the first principled, fully differentiable PF (DPF) which permits parameter inference in state-space models using end-to-end gradient based optimization. This property 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), or 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 & Greenewald, 2020), could also lead to DPFs of interest.

Acknowledgments

The work of Adrien Corenflos was supported by the Academy of Finland (projects 321900 and 321891). Arnaud Doucet is supported by the EPSRC CoSInES (COmputational Statistical INference for Engineering and Security) grant EP/R034710/1, James Thornton by the OxWaSP CDT through grant EP/L016710/1. Computing resources provided through the Google Cloud Platform Research Credits Programme.

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.

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., Schrittwieser, 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.

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 Greenewald, K. Gaussian-smooth optimal transport: Metric structure and statistical efficiency. *arXiv preprint arXiv 2001.09206*, 2020.

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 Nochetto, 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.

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.