Foundations of Data Science

doi:10.3934/fods.2024017

AN ADAPTIVE MIXTURE VIEW OF PARTICLE FILTERS

Nicola Branchini✉* and Víctor Elvira✉

School of Mathematics, University of Edinburgh, United Kingdom

Abstract. Particle filters (PFs) are algorithms that approximate the so-called filtering distributions in complex state-space models. We present a unified view on PFs as importance sampling with adaptive mixture proposals. Existing PFs can be derived as special cases by making specific choices for the components of the mixture proposals and for the importance weights. Our perspective clarifies that the existing PFs implicitly construct particular mixture proposals where the components are chosen independently of each other. We exploit the introduced flexibility of our perspective to propose a class of algorithms, adaptive mixture particle filters (AM-PF). Following IS arguments, the aim is to optimize the mixture proposal to match (an approximation of) the filtering posterior. We discuss two particular cases of the framework, the improved APF (IAPF) and the optimized APF (OAPF). In both linear and nonlinear dynamical systems models, our mixture particle filters consistently show improved performance compared to widely used algorithms such as the bootstrap particle filter (BPF) and the auxiliary particle filter (APF). We conclude by outlining promising future directions opened by our framework. [1]

1. **Introduction.** State-space models (SSMs) are widely used mathematical descriptions of dynamical systems. Real-world tasks in SSMs reduce to computing posterior distributions describing the uncertainty in the evolution of the underlying dynamical system. Posteriors can be obtained without approximations only for a restricted class of models making strong assumptions. Particle filters (PFs) address this shortcoming, providing algorithms that approximate posterior distributions in a much wider range of SSMs. The underlying principle behind PFs is that of approximating posteriors with weighted samples, which are drawn from user-chosen distributions called proposals. The publication of the bootstrap PF (BPF) [38] sparked decades of methodological research into PFs, which are impossible to summarize fully and spanned several communities including statistics [35], signal processing [18], and machine learning [34]. [21] and later [24] described the use of optimal proposals within PFs. [17] introduced sequential Monte Carlo samplers (SMC), i.e., a framework that generalizes PFs and makes the methodology applicable to general sequences of distributions outside the context of SSMs. [7] proved foundational theoretical results for SMC including a central limit theorem. For recent broad surveys on PFs and SMC, see [42, 23, 32, 36, 72, 56, 14, 75, 76]. A large number of real-world applications use PFs to approximate a sequence of

2020 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Particle filters, sequential Monte Carlo, importance sampling, nonlinear filtering, state-space models.

*Corresponding author: Nicola Branchini.

distributions, including mobile robot localization [68, 69, 67], simultaneous localization and mapping (SLAM) [25, 6], indoor tracking [15], weather forecasting [72], option pricing [11], Bayesian phylogenetic inference [3, 45], and more.
The BPF is arguably the most popular PF. Its implementation is simple and easily parallelizable over the number of particles. While being generally robust, some drawbacks of the BPF have been studied, including its poor performance when observations are very informative. [59] address this weakness by incorporating the observation into the resampling probabilities by interpreting them as auxiliary variables, which leads to a class of algorithms named auxiliary particle filters (APFs). The APF framework has received much attention and has been rederived under several interpretations [37, 43, 22, 73]. Within the framework of APFs, locally optimal choices for the resampling probabilities and the proposals lead to the fully adapted APF (FA-APF) [8, Chapter 10.3.3], which however is intractable in practice. Previous approaches resort to approximating the resampling probabilities and proposals of the FA-APF to develop PFs that can be implemented in practice.
In this paper, we introduce a framework for PFs where the auxiliary variables in APF and the standard PF proposals are viewed as weights and components of an overall mixture proposal. More specifically, we make the following contributions.

Contributions. Our contributions consist of the following. [2]

- We propose a framework where previous PF algorithms (BPF, APF, marginal PF) can be interpreted as constructing a mixture proposal where each component and weight is chosen to approximate certain locally optimal choices (i.e., the choices of the FA-APF), independently of each other.
- These insights lead us to propose a framework where the mixture weights and/or the components are treated as free parameters. The parameters can then be *jointly* optimized or selected to minimize a measure of discrepancy between the mixture proposal and the filtering posterior which leads to new practical filtering algorithms.
- We connect and unify our framework with other previous PFs working with mixtures, in particular the multiple importance sampling views of [28] and the marginal particle filtering of [44]. We extend these PFs to allow for a mixture whose parameters can be adapted jointly.
- We outline several promising directions opened up by our framework, which could lead to new efficient particle filters where mixture weights and components depend on all the particles from the previous iteration.

Structure of the paper. We start in Section 1.1 by introducing state-space models with the associated notation used throughout the paper. Then, we introduce PFs in a generic form leading to Algorithm 1. We explain the original perspective on the auxiliary PF (APF) of [59] to highlight the underlying assumptions that are used in practice to select resampling probabilities. We introduce our framework interpreting PFs as importance sampling with mixtures in Section 2. The framework leads to a class of algorithms, adaptive mixture particle filters (AM-PF), that allows for the adaptation of the mixture proposal, summarized in Algorithm 3. In Section 2.1 we expand on the justification behind the proposed AM-PF framework and we connect it to the fully adapted APF (FA-APF). In Section 3 we discuss several

[2] The present work is an extended version of [4]. With respect to that paper, we: (1) establish a unified framework that includes previous mixture-based PFs, (2) connect extensively with the related literature on PFs using marginal importance weights, (3) connect with the fully-adapted APF, and (4) explore further the choice of free parameters experimentally.

concrete choices for the various components of Algorithm 3, which lead to concrete algorithms such as the improved APF (IAPF) [27] and the optimized APF (OAPF) [4]. Concluding in Section 5, we discuss promising future directions opened up by our framework.

1.1. **State-space models and particle filtering.** The temporal evolution of a dynamical system can be modelled with the framework of state-space models (SSMs). The key component of an SSM is a stochastic discrete-time Markovian process of a hidden state $\{\mathbf{x}_t\}_{t\geqslant 1}$ where $\mathbf{x}_t \in \mathbb{R}^{d_{\mathbf{x}_t}}$, which can only be observed via corresponding noisy measurements $\{\mathbf{y}_t\}_{t\geqslant 1}$, where $\mathbf{y}_t \in \mathbb{R}^{d_{\mathbf{y}_t}}$. SSMs are fully specified by a prior probability density function (pdf), $p(\mathbf{x}_0)$, and by the transition and observation densities, $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $g(\mathbf{y}_t|\mathbf{x}_t)$, respectively, defined for $t \geqslant 1$. Note that $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a conditional pdf in $\mathbf{x}_t$, while $g(\mathbf{y}_t|\mathbf{x}_t)$ is a likelihood function since the observations are fixed. We focus on models where densities (i.e., pdfs) exist. In these models, the filtering task consists of the sequential estimation (i.e., online) of the filtering measures $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ for $t = 1, \ldots$ (whose corresponding pdf is $p(\mathbf{x}_t|\mathbf{y}_{1:t})$), as well as expectations of the form

$$\mathbb{E}_{\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})}[h_t(\mathbf{x}_t)] = \int h_t(\mathbf{x}_t)\pi(d\mathbf{x}_t|\mathbf{y}_{1:t}), \tag{1}$$

for (integrable) functions of interest h_t. Most SSMs of interest require approximate inference, due to several reasons, for example: (a) $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ has unknown parameters $\boldsymbol{\theta}$, which need to be estimated from observations; (b) $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is not available in closed form (c) even if $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is a Gaussian, $h(\mathbf{x}_t)$ is nonlinear.
In this context, particle filters (PFs) are the most popular class of algorithms to approximate the distributions of interest and their associated integrals such as Eq. (1). PFs are a sequential implementation of importance sampling (IS) [57, Chapter 9], generating at each time step t a set of M particles (Monte Carlo samples) $\{\mathbf{x}_t^{(m)}\}_{m=1}^M$ from a proposal pdf and assigning them normalized importance weights. First, we describe how the importance weights are computed, then how the weighted samples can be used to approximate the filtering distributions.

Derivation of the importance weights using smoothing densities. Many particle filtering algorithms are derived by considering a sequence of IS target densities (pdfs) $\{p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})\}_{t\geqslant 1}$, which are smoothing pdfs [64, 53]. The reason that smoothing pdfs are used, even if filtering pdfs $\{p(\mathbf{x}_t|\mathbf{y}_{1:t})\}_{t\geqslant 1}$ are ultimately of interest, is that they allow for a convenient (recursive) decomposition of the importance weights, which are defined as target $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ divided by proposal

$$w_t(\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})} = \frac{p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_t,\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{y}_t,\mathbf{x}_{t-1})} = w_{t-1}(\mathbf{x}_{1:t-1})\frac{g(\mathbf{y}_t|\mathbf{x}_t)f(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{y}_t,\mathbf{x}_{t-1})}, \tag{2}$$

where a proposal for the trajectory $\mathbf{x}_{1:t}$, i.e. $q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{y}_t,\mathbf{x}_{t-1})$ is used. The decomposition in Eq. (2) is convenient because it allows obtaining new weights by multiplying the previous weights by a factor that is constant in t. Sampling from the proposal in the space of particle trajectories,i.e., $\mathbf{x}_{1:t}^{(m)} \sim q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$, amounts to sampling from the so-called proposal kernel $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t|\mathbf{y}_t,\mathbf{x}_{t-1}^{(m)})$ and appending it to the previously simulated particles at $t-1, t-2, \ldots$. From now, we denote as $w_t^{(m)}$ the expression in Eq. (2) evaluated at the particular trajectory $\mathbf{x}_{1:t}^{(m)}$, i.e., $w_t^{(m)} = w_t(\mathbf{x}_{1:t}^{(m)})$. The aim of PFs is to use samples from the proposal and their associated weights to construct approximations of the filtering distribution. However, the weights given by Eq. (2) assume that we can point-wise evaluate the normalizing constant of $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$, i.e. $p(\mathbf{y}_{1:t})$, which

Algorithm 1: Particle Filter

1. **Initialization.** Obtain initial particle approximation $\{1/M, \mathbf{x}_0^{(m)}\}_{m=1}^M$ as a sample from the prior pdf $p(\mathbf{x}_0)$. Recall, $\mathbf{x}_t$ denotes a resampled particle.
2. **Recursive step.** For $t = 1, \dots, T$
   - **Sampling.** Sample $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$, for $m = 1, \dots, M$
   - **Weighting.** Set the unnormalized importance weights as

$$\widetilde{w}_t^{(m)} = \widetilde{w}_{t-1}^{(m)} \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)} \quad m = 1, \dots, M, \tag{5}$$

     where $\widetilde{w}_{t-1}^{(m)} = 1/M$ since resampling is performed at every iteration. The normalized weights are given by $\bar{w}_t^{(m)} = \widetilde{w}_t^{(m)} / \sum_{j=1}^M \widetilde{w}_t^{(j)}$
   - **Resampling.** Resample $\{\bar{w}_t^{(m)}, \mathbf{x}_{1:t}^{(m)}\}_{m=1}^M$ to obtain $\{1/M, \mathbf{x}_{1:t}^{(m)}\}_{m=1}^M$, the particles used for $t+1$.
3. Output $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m,t=1}^{M,T}$ as particle approximations of $\{\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})\}_{t=1}^T$

is usually not known.

Particle approximations with self-normalized weights. In these cases, we define weights with a weight function $\widetilde{w}(\mathbf{x}_{1:t})$ that uses the unnormalized smoothing distribution, $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \cdot p(\mathbf{y}_{1:t})$, in the numerator of the IS weights. The evaluation of $\widetilde{w}(\mathbf{x}_{1:t})$ at the m-th trajectory $\mathbf{x}_{1:t}^{(m)}$ is then

$$\widetilde{w}_t^{(m)} = \widetilde{w}(\mathbf{x}_{1:t}^{(m)}) = \frac{p(\mathbf{x}_{1:t}^{(m)}, \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(m)}|\mathbf{y}_{1:t})} = \widetilde{w}_{t-1}^{(m)} \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)}. \tag{3}$$

Note. Eq. (3) differs from Eq. (2) as the unnormalized target is used in Eq. (3). The weights $\widetilde{w}_t^{(m)}$ can be used to consistently estimate expectations (Eq. (1)) with self-normalization [57, Chapter 9], i.e., normalizing the weights such that they sum to one as $\bar{w}_t^{(m)} = \widetilde{w}_t^{(m)} / \sum_{j=1}^M \widetilde{w}_t^{(j)}$. With normalized weights, the particle approximations to filtering and smoothing measures are

$$\pi(d\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \approx \sum_{m=1}^M \bar{w}_t^{(m)} \delta_{\mathbf{x}_{1:t}^{(m)}}(d\mathbf{x}_{1:t}) \Rightarrow \pi(d\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^M \bar{w}_t^{(m)} \delta_{\mathbf{x}_t^{(m)}}(d\mathbf{x}_t), \tag{4}$$

where $\delta_{\mathbf{x}_t^{(m)}}(d\mathbf{x}_t)$ is the Dirac delta measure for the random variable $\mathbf{x}_t$, placing all of its mass at $\mathbf{x}_t^{(m)}$. We denote these empirical measures with a "hat", i.e., $\widehat{\pi}(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{m=1}^M \bar{w}_t^{(m)} \delta_{\mathbf{x}_t^{(m)}}(d\mathbf{x}_t)$, since they are (random) approximations of the true measures. A simple proof that Eq. (4) converges in distribution to the true filtering measure can be found in [31, Chapter 2].

Note that in Eq. (4), by constructing a particle representation of the smoothing measure $\pi(d\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$, we were able to obtain one for the filtering distribution $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ as a consequence of the fact that for particle approximations $\int^{(t)} \cdots \int \delta_{\mathbf{x}_{0:t}^{(m)}}(d\mathbf{x}_{0:t}) = \delta_{\mathbf{x}_t^{(m)}}(d\mathbf{x}_t)$.The marginalization over previous states $\mathbf{x}_{1:t-1}$ does

not affect the importance weight $\widetilde{w}_t^{(m)}$, which is the same in both expressions in Eq. (4). As we will see in Section 2, marginalizing in the IS weights also leads to a different class of PFs.

Resampling. Before simulating the particles at time $t+1$ from the proposal(s) $\{q(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)})\}_{m=1}^M$, a particle filter performs resampling, i.e., it replaces the weighted particle set $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m=1}^M$ with a new, equally weighted particle set $\{1/M, \mathbf{x}_t^{(m)}\}_{m=1}^M$. The resampled particles $\mathbf{x}_t^{(m)}$ are obtained by sampling with replacement from the previous particle set, with probabilities $\bar{w}_t^{(m)}$. Many resampling schemes with different properties exist [19, 41, 50], and it is often good practice not to resample at every iteration, but only when a certain condition is satisfied. This is called adaptive resampling (see, e.g., [8, Chapter 10.2]). The most popular choice for $q(\mathbf{x}_t|\mathbf{y}_t, \mathbf{x}_{t-1}^{(m)})$ is $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ and leads to the bootstrap particle filter (BPF) [38]. An advantage of this choice is that the weights in Eq. (3) simplify to $\widetilde{w}_{t-1}^{(m)} g(\mathbf{y}_t|\mathbf{x}_t^{(m)})$, so that the transition densities $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ do not need to be point-wise evaluated.

Pseudocode. See Algorithm 1 for the pseudocode of a PF algorithm with generic proposal $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ that is allowed to depend on $\mathbf{y}_t$. In summary, a PF sequentially updates a set of normalized weights and particles $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m=1}^M$ as a representation of the filtering pdf, updating weights at each time step as in Eq. (2) and sampling current particles the M available proposals (or " proposal kernels") $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$.

Metrics. PFs are generally evaluated by measures of accuracy of the particle approximations to the T filtering distributions involved. Concretely, this reduces to approximate notions such as (approximations of the) effective sample size (ESS) [54, 64, 30] when the true filtering distribution is not available in closed form, which is related to the variance of the PF estimator of the likelihood $p(\mathbf{y}_{1:t})$. It is important to note that approximations of the ESS, especially in the context of PFs, can be unreliable; see [30].

Standing assumptions and notation. We assume for simplicity of presentation that resampling is performed at each iteration, although many possibilities for adaptive resampling exist [8] and we also assume multinomial resampling is used. This article does not deal with parameter estimation, i.e., transition and observation densities are assumed to be known. For two functions $f(n)$ and $g(n)$ (usually on $\mathbb{N}_{\geqslant 0}$), we write $f(n) = \mathcal{O}(g(n))$ as $n \to \infty$ if there are constants C, n_0 such that $|f(n)| \leqslant Cg(n)$ whenever $n \geqslant n_0$. Similarly, $f(n) = \Omega(g(n))$ as $n \to \infty$ if there are constants C, n_0 such that $|f(n)| \geqslant Cg(n)$ whenever $n \geqslant n_0$. Hence, we write $f(n) = \Theta(n)$ if both $f(n) = \mathcal{O}(n)$ and $f(n) = \Omega(g(n))$.

1.2. **Auxiliary Particle Filters.** In this section, we review the auxiliary PF (APF) which was introduced by [59] to address certain weaknesses of the BPF. Algorithmically, as shown in Algorithm 2, the only two differences with Algorithm 1 are in the resampling and weighting steps. Our focus in this Section will be on the original derivation and justification of the APF, to highlight how it implicitly restricts the choices of proposals, resampling weights and IS weights to certain kinds of approximations of the so-called fully adapted APF (FA-APF). Our framework in Section 2 will relax this restriction.

Algorithm 2: Auxiliary Particle Filter

1. **Initialization.** Obtain initial particle approximation $\{1/M, \mathbf{x}_0^{(m)}\}_{m=1}^M$ as a sample from the prior pdf $p(\mathbf{x}_0)$. Recall, $\mathbf{x}_t^{(m)}$ denotes a resampled particle.
2. **Recursive step.** For $t = 1, \dots, T$
   - Select $\{\widetilde{\lambda}_t^{(m)}\}_{m=1}^M$ to approximate $\{\bar{w}_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}$, normalize them to get probabilities $\{\lambda_t^{(m)}\}_{m=1}^M$
   - **Resampling.** Resample $\{\lambda_t^{(m)}, \mathbf{x}_{1:t-1}^{(m)}\}_{m=1}^M$ to obtain $\{1/M, \mathbf{x}_{1:t-1}^{(m)}\}_{m=1}^M$. Let $\{i^{(m)}\}_{m=1}^M$ be the resampled indices.
   - **Sampling.** Sample $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$, for $m = 1, \dots, M$
   - **Weighting.** Set the unnormalized importance weights as

$$\widetilde{w}_t^{(m)} = \widetilde{w}_{t-1}^{(m)} \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{\lambda_t^{(i^{(m)})} q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)} \quad \text{, for } m = 1, \dots, M, \tag{6}$$

     where $\widetilde{w}_{t-1}^{(m)} = 1/M$ since resampling is performed at every iteration.
3. Output $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m,t=1}^{M,T}$ as particle approximations of $\{\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})\}_{t=1}^T$

Original derivation of APF [59] noticed that informative observations (i.e., outliers, or when $g(\mathbf{y}_t|\mathbf{x}_t)$ has little uncertainty as a pdf in $\mathbf{y}_t$) lead to high variance in the resulting importance weights $\bar{w}_t^{(m)}$ from Algorithm 1. It is easy to show that in this case, the distributions $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ and $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t-1})$ (so-called predictive) are far apart, which complicates the filtering task. To address these issues, they consider the following approximation to the unnormalized filtering density

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1}) \approx g(\mathbf{y}_t|\mathbf{x}_t) \sum_{m=1}^M \bar{w}_{t-1}^{(m)} f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \tag{7}$$

[59] then cast the aim of a PF as simulating particles approximately distributed according to the normalized version of Eq. (7). However, they stop further developing this solution due to the computational cost implied by the evaluation of Eq. (7), which requires $\Theta(M^2)$ point-wise evaluations of densities, when maintaining M particles. The approach was later revisited and studied further in auxiliary marginal PFs and multiple importance sampling PFs (see Section 2).

Remark (1). Arguably, the setting where the likelihood $g(\mathbf{y}_t|\mathbf{x}_t)$ is informative, i.e., the distributions $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ and $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t-1})$ differ significantly, is the more interesting scenario for online filtering. When $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ and $\pi(d\mathbf{x}_t|\mathbf{y}_{1:t-1})$ are similar, the filtering distribution is changing little from time $t-1$ to t.

Introduction of the auxiliary variable. [59] proposed instead to attempt (approximate) simulation from a certain extended space joint $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ in $\mathbb{R}^{d_{\mathbf{x}_t}} \times \{1, \dots, M\}$, which has marginals the pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and a pmf $\{i^{(m)}, \lambda_t^{(m)}\}_{m=1}^M$,

with $\lambda_t^{(m)} = \mathbb{P}[i^{(k)} = m]$, whose values we discuss below. The joint function $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ is a hybrid discrete/continuous object, which often appears in simulation methods (e.g., [78]) [3]. While $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ is not rigorously defined, we resort to an operational definition, being able to define sampling and point-wise evaluation straightforwardly. The key observation is that, if the pair $\mathbf{x}_t, i^{(m)} \sim p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$, then by discarding the sampled index $i^{(m)}$, we obtain $\mathbf{x}_t$ distributed according to the filtering pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. In practice, it will be possible to do so only approximately. Crucially, this process avoids sampling from or evaluating the expensive mixture in Eq. (7). We now discuss how to construct $p(i^{(m)}|\mathbf{y}_{1:t})$, which is the pmf corresponding to a certain empirical measure, and how approximations of this pmf are used to sample from an approximation of $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$.

Remark (2). In the APF, the goal of particle filtering can be interpreted as sampling from an approximation of the extended space joint function $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ for a discrete auxiliary random variable $i^{(m)}$.

Note that $i^{(m)}$ is a r.v. and we will often abuse notation slightly (analogously to pdfs) by writing, e.g., $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})})$ instead of $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)}=k)})$ for some value k.

Approximations in APF. Sampling from the joint function $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ can be decomposed with ancestor sampling, i.e., since $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t}) = p(\mathbf{x}_t|i^{(m)}, \mathbf{y}_{1:t})p(i^{(m)}|\mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)p(i^{(m)}|\mathbf{y}_{1:t})$, one can sample an index $i^{(m)}$ from the pmf $p(i^{(m)}|\mathbf{y}_{1:t})$ and then sample from the kernel corresponding to that index $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)$. However, both steps are intractable in general. The kernel $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ is known as optimal kernel [24] and it is usually impossible to sample from. The M probabilities of the pmf $p(i^{(m)}|\mathbf{y}_{1:t})$, i.e., $\lambda_t^{(m)}$, are set as the weights of a particle approximation of $\pi(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t})$ (a one-step smoothing distribution) as

$$\mathbb{P}[i^{(k)} = m] = \lambda_t^{\star,(m)}, \ \lambda_t^{\star,(m)} = \frac{\widetilde{\lambda}_t^{\star,(m)}}{\sum_{p=1}^{M} \widetilde{\lambda}_t^{\star,(p)}}, \ k = 1, \ldots, M, m = 1, \ldots, M, \tag{8}$$

where we now use the $\star$ to denote that these weights will need to be approximated, and

$$\widetilde{\lambda}_t^{\star,(m)} = \bar{w}_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}), \ m = 1 \ldots M, \tag{9}$$

so that

$$\sum_{m=1}^{M} \lambda_t^{\star,(m)} \delta_{\mathbf{x}_{t-1}^{(m)}}(d\mathbf{x}_{t-1}) \approx \pi(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t}). \tag{10}$$

The validity of Eq. (9) can be noted by inspecting the identity of the corresponding density for the smoothing distribution $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t}) = p(\mathbf{y}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$. Since the term $p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)})$ involves an intractable integral, $\widetilde{\lambda}_t^{\star,(m)}$ needs to be approximated. Therefore, in practice, an APF scheme selects an approximation of the optimal kernels $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t) \approx p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ that can be sampled from and

[3] Here, m can be interpreted as a mixture index of Eq. (7), or it can be also viewed as an auxiliary variable without statistical meaning.

an approximation of the weights $\widetilde{\lambda}_t^{(m)} \approx \widetilde{\lambda}_t^{\star,(m)}$ in Eq. (9), which constitute the elements of an approximation $q(i^{(m)}|\mathbf{y}_{1:t})$ of the pmf $p(i^{(m)}|\mathbf{y}_{1:t})$. Once these approximations are chosen, we obtain a particle approximation of the filtering distribution $\widehat{\pi}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ as in Section 1.1 using (a normalized version of) the following IS weights as target divided by proposal (in the extended space)

$$w_t^{(m)} = \frac{p(\mathbf{x}_t^{(m)}, i^{(m)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_t^{(m)}, i^{(m)}|\mathbf{y}_{1:t})} = \frac{p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)\ \tilde{\lambda}_t^{\star,i^{(m)}}}{q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)\ \tilde{\lambda}_t^{(i^{(m)})}} = \frac{w_{t-1}^{(m)} g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)\ \tilde{\lambda}_t^{(i^{(m)})}}, \tag{11}$$

We emphasize that both terms in the denominator of Eq. (11) are part of an overall proposal and, therefore can be chosen by the user.

Remark (3). The choices of $\widetilde{\lambda}_t^{(m)} = \widetilde{\lambda}_t^{\star,(m)}$ and $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ correspond to the so-called fully adapted APF (FA-APF) [73] and minimize the conditional variance of the IS weights in Eq. (11)

Since the FA-APF choices are intractable for most models, we now discuss in particular how to choose $\lambda_t^{(m)}$ in practice.

Connection to resampling and the fully adapted APF (FA-APF). The probabilities $\lambda_t^{(m)}$ corresponding to the pmf $q(i^{(m)}|\mathbf{y}_{1:t})$ and used to simulate the auxiliary variables, sometimes called simulation weights or first-stage weights [59, 27], can be seen equivalently as the resampling weight in a standard (non-extended space) PF such as Algorithm 1 [22]. Notice that, in the best case scenario (i.e., when exact simulation from the joint function $p(\mathbf{x}_t, i^{(m)}|\mathbf{y}_{1:t})$ is feasible or equivalently when we can use the FA-APF) we obtain a simulation weight that depends on the observation, since from Eq. (9), $\widetilde{\lambda}_t^{\star,(m)} = \bar{w}_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)})$, which is sometimes presented as the main intuitive motivation behind using APF over BPF [22].

In practice, we need to approximate the FA-APF choice of simulation weights, which means designing approximations to the M intractable integrals in Eq. (9), given by the terms $\{p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$. The most common choice for practical implementations is to approximate each integral at a point associated with the transition kernels $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ (e.g., the mode or the mean) as

$$p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}) = \int g(\mathbf{y}_t|\mathbf{x}_t) f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t \approx \int g(\mathbf{y}_t|\mathbf{x}_t) \delta_{\boldsymbol{\mu}_{t|t-1}^{(m)}}(d\mathbf{x}_t) d\mathbf{x}_t = g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(m)}), \tag{12}$$

where $\boldsymbol{\mu}_{t|t-1}^{(m)} = \mathbb{E}_{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t]$. This reduces to evaluating the observation density $g(\mathbf{y}_t|\mathbf{x}_t)$ at $\boldsymbol{\mu}_{t|t-1}^{(m)}$. Other choices are described in [22, 56], and include, e.g., a Gaussian approximation of a linearized version of the integral Eq. (12). Then, selecting the unnormalized resampling weights as

$$\widetilde{\lambda}_t^{(m)} = \bar{w}_{t-1} g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(m)}), \tag{13}$$

and sampling particles with the transition kernels $\{q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)\}_{m=1}^M = \{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$ leads to IS weights as (by plugging into Eq. (11))

$$\widetilde{w}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(i^{(m)})})} \quad \text{, for } m = 1, \dots, M, \tag{14}$$

which consists of the most common concrete instantiation of the APF framework. In summary, the APF scheme can be seen as a generalization of the standard PF with (extended space) IS weights as in Eq. (2). We recover the BPF by choosing the probabilities of $q(i^{(m)}|\mathbf{y}_{1:t})$, i.e., $\{\lambda_t^{(m)}\}_{m=1}^M$ to be the normalized weights $\{\bar{w}_{t-1}^{(m)}\}_{m=1}^M$, and the proposal kernels to be $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t) = f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$.

Limitation of the APF. A drawback of the presented derivation of the APF is that in practice PFs are designed by selecting each $\lambda_t^{(m)}$ as an approximation to the corresponding FA-APF optimal choice $\lambda_t^{\star,(m)}$ and similarly $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ as an approximation of $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$. However, as we will show in the next Section, the FA-APF can be equivalently thought of as an algorithm sampling from the following mixture

$$\sum_{m=1}^{M} \lambda_t^{\star,(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t), \tag{15}$$

since resampling with weights $\lambda_t^{\star,(m)}$ and propagating particles with kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ is equivalent to sampling from Eq. (15). Indeed, we will interpret all PFs as IS algorithms with a proposal given by a mixture. Since in practice, it is not possible to approximate perfectly each weight $\lambda_t^{\star,(m)}$ and component $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ of the mixture in Eq. (15), we will introduce PFs where the proposed weights $\{\lambda_t^{(m)}\}_{m=1}^M$ and components $\{q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)\}_{m=1}^M$ jointly try to match the full mixture in Eq. (15), as opposed to its individually matching the components.

2. Particle filters based on adaptive mixtures.

In this Section, we introduce our framework for particle filters based on interpreting PFs as sequential importance sampling with mixture proposals. The framework builds on and extends [28] to consider adaptive mixtures. We also expand on the connections with other PFs based on mixtures such as marginals PFs [44] in Section 2.2. In Section 2.1 we summarize the components of the framework and introduce the pseudocode for the resulting class of particle filters, i.e., adaptive mixture particle filters (AM-PF).

2.1. Overview of the framework and justification.

Recall that a PF aims to produce, for each timestep t, an approximation of the filtering distribution $\widehat{\pi}(d\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \pi(d\mathbf{x}_t|\mathbf{y}_{1:t})$ in the form of a weighted empirical measure, summarized by a weighted set of particles $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m=1}^M$. In our framework, for each timestep t, samples are obtained from a single mixture proposal with K components

$$\psi_t(\mathbf{x}_t) = \sum_{k=1}^{K} \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t). \tag{16}$$

Then, the following three steps are carried out in each iteration t:

1. Adaptation of the mixture proposal
2. Sampling from the mixture proposal
3. Weighting of the generated samples.

The three steps generate a weighted set of particles that approximates the filtering distribution at time t. The pseudocode for the full algorithm is given by Algorithm 3 (following pages). The sampling and weighting steps follow the reasoning of the previous multiple importance sampling (MIS) particle filters [28]. The sampling step, therefore, replaces the traditional resampling plus sampling from

Algorithm 2. The weighting is done by following IS arguments as target divided by proposal. The target distribution is an approximation since the true filtering distribution cannot be evaluated point-wise due to an intractable integral, as $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto g(\mathbf{y}_t|\mathbf{x}_t) \int f(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$, for which a typical approximation is $g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{i=1}^{M} w_{t-1}^{(i)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i)})$ [44, 28]. Therefore, we write the IS weight as the approximation to the unnormalized filtering pdf divided by the mixture proposal

$$\widetilde{w}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{i=1}^{M} w_{t-1}^{(i)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i)})}{\sum_{k=1}^{K} \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t^{(m)})} \qquad m = 1, \dots, M. \tag{17}$$

We will refer to Eq. (17) as the MIS weights or the marginal weights.

We now discuss the justification behind the first of the three steps, i.e., the adaptation of the mixture proposal $\psi_t(\mathbf{x}_t)$. We later provide concrete examples of its implementation that go beyond the standard implicit choices made by BPF and APF in Section 3.

Adaptation of the mixture proposal. In order to define a criterion to choose the proposal $\psi_t(\mathbf{x}_t)$, as common in IS [57] we first aim to define an optimal proposal. Ideally, we would target the filtering pdf, but we have seen in Eq. (17) that we can only evaluate an approximation. Additionally, by rearranging the approximation of the filtering pdf as

$$g(\mathbf{y}_t|\mathbf{x}_t) \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}) = \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}) p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t), \tag{18}$$

we can obtain a normalized density by normalizing the mixture weights of the RHS in Eq. (18). Doing so leads to the following mixture approximation of the filtering pdf

$$\psi_t^\star(\mathbf{x}_t) = \sum_{m=1}^{M} \underbrace{\frac{\bar{w}_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)})}{\sum_{k=1}^{M} \bar{w}_{t-1}^{(k)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(k)})}}_{\text{mixture weights}} \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)}_{\text{mixture component}} = \sum_{m=1}^{M} \lambda_t^{\star,(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t). \tag{19}$$

Notice that Eq. (19) is the same as Eq. (15), i.e., the mixture weights are the FA-APF resampling weights $\lambda_t^{\star,(m)}$ and the mixture components are the optimal PF kernels. Because sampling from a mixture is equivalent to resampling with weights given by the mixture weights and propagating particles with the mixture components, the resampling and propagating steps of the FA-APF can be interpreted as sampling from $\psi_t^\star(\mathbf{x}_t)$, which is a particular approximation of the filtering distribution. Notice that we can consider $\psi_t^\star(\mathbf{x}_t)$ as the IS target distribution in our framework, because (up to normalization) it is equivalent to the numerator in Eq. (17). The introduced flexibility of our framework leads to the following important remark.

Remark (4): While previous works implicitly attempt to approximate the optimal mixture term-by-term, we can now consider algorithms where the mixture proposal attempts to match the whole optimal mixture $\psi_t^\star(\mathbf{x}_t)$ which corresponds to the FA-APF.

We exemplify this remark concretely for the APF. **APF as a special case.** Consider the APF framework as per Algorithm 2. By selecting, in our framework,

$K = M$ (recall, M is the number of particles maintained at each time t) and specifically associating each mixture component $q_t^{(m)}(\mathbf{x}_t)$ with one of the particles at time $t-1$, i.e., $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^{M}$, which can be represented as standard in PFs by writing $q_t^{(m)}(\mathbf{x}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$ (adding optional dependence on the observation as well). Consider, for simplicity of the argumentation, that we can select the optimal kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$, so there remains to choose the mixture weights. As discussed in Section 1.2, typical strategies for selecting $\lambda_t^{(m)}$ in APF consist in constructing, for each m, an approximation such that $\lambda_t^{(m)} \approx \lambda_t^{\star,(m)}$. These approximations are imperfect. As we show in Section 3 and later in the experiments, by choosing mixture weights in a joint manner, i.e., where the value of each $\lambda_t^{(m)}$ depends on all of the other values $\lambda_t^{(j)}, j \neq m$ we can obtain a mixture $\psi_t(\mathbf{x}_t)$ that is closer to $\psi_t^{\star}(\mathbf{x}_t)$.

2.2. **Relationship with other particle filters based on mixtures.** Our framework extends the multiple importance sampling (MIS) PFs of [28] by **(a)** allowing for an adaptive mixture **(b)** connecting the choice of the optimal mixture to the fully adapted APF as discussed in Section 2.1 and **(c)** detaching the number of kernels K in the mixture proposal to the number of particles used, M. We discuss the selection of K in Section 3. We use the view of [28] in replacing resampling plus sampling from kernels by sampling from a mixture, as well as their derivation of the IS weights in Eq. (17). [44] presented the marginal PFs (MPFs) also derived the IS weights in Eq. (17) (with a different mixture in the denominator) from a different interpretation, that is by considering the extended space discussed in Section 1.2 and marginalizing the auxiliary variable in the IS weight from Eq. (11). However, the MPF perspective does not offer any additional insights into how to select mixture weights or components w.r.t to the APF, thus it is limited to proposing the use of the marginalized IS weight. Notably, [31, Chapter 4, Section 3.2] earlier presented the marginalized IS weight. Other PFs using the marginal IS weights are reviewed in [13] (see references therein), where the authors also prove several theoretical results on these types of PFs. Further, [12] proposed a particle filter based on Gaussian mixtures where remarkably they can prove an explicit convergence rate to the true filtering measure in terms of the number of mixture components.

Other related works. [46] proposed to simulate particles, instead of using the transition kernel f (as in the BPF), with a mixture between f and a function proportional to the observation density g. Differently from our work, MIS concepts in their perspective help in selecting good proposal kernels (mixture components, in our framework). Their IS weight is based on Eq. (11). Their work could be combined in our framework for a better selection of mixture components. [10] present another perspective on SMC as adaptive importance sampling but do not use the marginal IS weight of [31], [44] and [28] that we also use in this work. Finally, much methodological research has been devoted to improving the BPF and incorporating future observations as early as possible, including online schemes [50] and offline schemes such as twisted APFs [74, 39, 40]. Other recent PFs introduce deterministic transformations to move the particles in high posterior probability regions [1], some even considering the removal of resampling [61, 9] or weighting [77, 52].

2.3. **Theoretical properties of mixture particle filters.** [4] proved that using IS weight as in Eq. (17) still leads to unbiased and consistent estimators of the likelihood $p(\mathbf{y}_{1:T})$ for a generic class of mixture proposals in the context of the optimized auxiliary particle filter (OAPF), which is a special case of the AM-PF framework. We report the result here for completeness.

Let an estimator of the likelihood of all of the observations be constructed by multiplying estimators of the normalizing constants $\widehat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1})$ as

$$\widehat{p}(\mathbf{y}_{1:T}) = \widehat{p}(\mathbf{y}_1)\prod_{t=2}^{T}\widehat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}), \tag{20}$$

where $\widehat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \frac{1}{M}\sum_{m=1}^{M}\widetilde{w}_t^{(m)}$, using the IS weight as per Eq. (17).

Theorem 2.1. [4] *Assume the following:* ***(1)*** *observations $\{\mathbf{y}_t\}_{t\geq 0}$ are fixed, i.e., not random;* ***(2)*** *likelihoods are bounded, $\sup_{\mathbf{x}_t} g(\mathbf{y}_t|\mathbf{x}_t) < \infty$;* ***(3)*** *likelihoods are nonzero, $g(\mathbf{y}_t|\mathbf{x}_t) > 0$ for all $\mathbf{x}_t$;* ***(4)*** *weights are bounded, i.e., $\mathbb{E}[w(\mathbf{x}_t)] < \infty$. For any set of mixture proposals $\{\psi_t(\mathbf{x}_t)\}_{t=1}^{T}$ the normalizing constant estimator in Eq. (20) is unbiased and consistent, i.e., $\mathbb{E}[\widehat{p}(\mathbf{y}_{1:T})] = p(\mathbf{y}_{1:T})$ and $\lim_{M\to\infty}\widehat{p}(\mathbf{y}_{1:T}) = p(\mathbf{y}_{1:T})$ a.s. for any $T \in \mathbb{R}^+$.*

Proof. The proof is presented in the Appendix. Unbiasedness of the likelihood forms the basis for the inclusion of PF schemes into particle Markov Chain Monte Carlo (PMCMC), which we do not expand on here. [13] recently studied more in-depth the theoretical properties of particle filters using the IS weight in Eq. (17), showing that typical CLT results can be extended to this case.

Algorithm 3: Adaptive Mixture Particle Filter (AM-PF)

1. **Initialization.** Obtain initial particle approximation $\{1/M, \mathbf{x}_0^{(m)}\}_{m=1}^{M}$ as a sample from the prior pdf $p(\mathbf{x}_0)$. Recall, $\mathbf{x}_t^{(m)}$ denotes a resampled particle.
2. **Recursive step.** For $t = 1, \ldots, T$
   - **Adaptation of the mixture proposal.** Adapt weights $\{\lambda_t^{(k)}\}_{k=1}^{K}$ and/or components $\{q_t^{(k)}(\mathbf{x}_t)\}_{k=1}^{K}$ of the mixture proposal
   $$\psi_t(\mathbf{x}_t) = \sum_{k=1}^{K}\lambda_t^{(k)}q_t^{(k)}(\mathbf{x}_t) \tag{21}$$
   using the particle approximation from $t-1$, i.e., $\{\bar{w}_{t-1}^{(m)}, \mathbf{x}_{t-1}^{(m)}\}_{m=1}^{M}$ and the current observation $\mathbf{y}_t$
   - **Sampling.** Obtain M i.i.d. new samples from the mixture proposal
   $$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t), \qquad m = 1, \ldots, M \tag{22}$$
   - **Weighting.** Obtain the unnormalized IS weights as
   $$\widetilde{w}_t^{(m)} = \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(m)}\right)\sum_{j=1}^{M} w_{t-1}^{(j)} f\left(\mathbf{x}_t^{(m)} \mid \mathbf{x}_{t-1}^{(j)}\right)}{\sum_{k=1}^{K}\lambda_t^{(k)}q^{(k)}\left(\mathbf{x}_t^{(m)}\right)}, \tag{23}$$
3. Output $\{\bar{w}_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m,t=1}^{M,T}$

3. **Implementations of the AM-PF framework.** In this Section, we describe two concrete implementations of our framework, the improved APF (IAPF) of [27] and the optimized APF (OAPF) of [4]. We emphasize that differently to the previous mixture-based PFs such as the marginal PFs of [44], both algorithms can choose mixture weights by targeting the full mixture $\psi_t^\star(\mathbf{x}_t)$ in Eq. (19). As we expand on in Section 3.2, we did not study algorithms that adapt the kernel parameters, which we leave to future work.

3.1. **Selecting the mixture weights: Improved APF.** [27] exploits the MIS perspective to derive the that are jointly selected (informally, "cooperate") to approximate the unnormalized filtering pdf, deriving an algorithm that consists of an implementation of Algorithm 3 with a certain choice of mixture weights. The choice is in contrast to common choices of mixture weights derived by algorithms based on Algorithm 2 because each weight $\lambda_t^{(m)}$ depends on all of the particles $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^M$. [27] selects the mixture weight as the ratio of two mixtures, the unnormalized target and the equally weighted mixture of transition kernels $\frac{1}{M}\sum_{j=1}^M p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(j)})$, evaluated at the means of the transition kernels $\{\boldsymbol{\mu}_{t|t-1}^{(m)}\}_{j=1}^M = \{\mathbb{E}_{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t]\}_{j=1}^M$ as

$$\widetilde{\lambda}_t^{(m)} = \frac{g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(m)})\sum_{j=1}^M \bar{w}_{t-1}^{(j)} f(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\frac{1}{M}\sum_{j=1}^M f\left(\boldsymbol{\mu}_{t|t-1}^{(m)} \mid \mathbf{x}_{t-1}^{(j)}\right)} \qquad m = 1, \ldots, M, \tag{24}$$

where $\widetilde{\lambda}_t^{(m)}$ is the unnormalized weight. Note that the RHS of Eq. (24) is a scalar, since all expressions are evaluated at $\{\boldsymbol{\mu}_{t|t-1}^{(m)}\}_{j=1}^M$.

Remark (5). While the above definition uses the transition kernels $\{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$, in the general case these would be replaced with proposal kernels $\{q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)\}_{m=1}^M$.

We, therefore, proceed with the following slightly more general version of Eq. (24), as

$$\widetilde{\lambda}_t^{(m)} = \frac{g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(m)})\sum_{j=1}^M \bar{w}_{t-1}^{(j)} f(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\frac{1}{M}\sum_{j=1}^M q(\boldsymbol{\mu}_{t|t-1}^{(m)}|\mathbf{x}_{t-1}^{(j)}, \mathbf{y}_t)} \qquad m = 1, \ldots, M, \tag{25}$$

By employing the assumption that the proposal kernels are well separated [4], i.e., the particle $i^{(m)}$ which was sampled from the m-th kernel needs to be only evaluated at that kernel (other evaluations are negligible), we can simplify Eq. (25) as

$$\widetilde{\lambda}_t^{(m)} \approx \frac{g(\mathbf{y}_t|\boldsymbol{\mu}_{t|t-1}^{(m)})\bar{w}_{t-1}^{(i^{(m)})} f(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{\frac{1}{M} q(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)}. \tag{26}$$

Note that after this approximation, $\widetilde{\lambda}_t^{(m)}$ does not depend on all the particles $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^M$ anymore. Plugging Eq. (26) into the expression of the IS weight in

[4] (see [28] for details)

Eq. (17), gives

$$\widetilde{w}(\mathbf{x}_t^{(m)}) = \frac{q(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t) g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{g(\mathbf{y}_t|\boldsymbol{\mu}_{t-1}^{(i^{(m)})}) f(\boldsymbol{\mu}_{t-1}^{(i^{(m)})}|\mathbf{x}_{t-1}^{(i^{(m)})}) q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)}. \tag{27}$$

Using the transition kernels to simulate the particles $\{q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)\}_{m=1}^M = \{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$ reduces Eq. (27) to the most common implementation of the APF, with unnormalized IS weights as in Eq. (14).

The weight in Eq. (25) is more costly than Eq. (26), i.e., it requires $\Theta(M^2)$ density evaluations versus $\Theta(M)$, but it is more appropriate when the assumption of separated kernels does not hold. In those cases, it is expected to lead to a mixture proposal that better matches the posterior. As mentioned, in Eq. (25), the value of each $\widetilde{\lambda}_t^{(m)}$ depends not only on $\boldsymbol{\mu}_t^{(m)}$, but on the whole set $\{\boldsymbol{\mu}_t^{(j)}\}_{j=1}^M$. As we show experimentally in Section 4, this often leads to better mixture proposals.

3.2. **Optimizing the mixture weights: OAPF.** Next, we define an optimization objective where the aim is to optimize mixture weights such that our proposal $\psi_t(\mathbf{x}_t)$ is close to $\psi_t^\star(\mathbf{x}_t)$. This can adapt better to a wider range of situations compared to the approach described in the previous subsection. We design a method that can be applied, without additional generation of particles, to optimize the mixture weights. In this way, additional cost w.r.t. algorithms such as APF is only due to additional density evaluations and any other operations that are not sample generation.

To start, recall that in IS we can frame the problem of having a proposal that is close to the optimal pdf as minimizing the variance of the IS weights. In the best case, the IS weights are constant. Since we cannot evaluate the IS weights everywhere in $\mathbf{x}_t$ space, in each timestep t we select a set of points $\{\mathbf{z}_t^{(e)}\}_{e=1}^E$, where the IS weights are going to be evaluated. For concreteness, an example would be the case with $K = E$, where the evaluation points are the centers (means) of the K kernels in the mixture proposal. We now show that setting the IS weights in Eq. (17), evaluated at the points $\{\mathbf{z}_t^{(e)}\}_{e=1}^E$, equal to one is equivalent to obtaining a linear system of equations in the mixture weights.

Let us define the the vectors $\boldsymbol{\lambda} = (\lambda_t^{(1)}, \dots, \lambda_t^{(K)})^\top, \mathbf{w} = (w_{t-1}^{(1)}, \dots w_{t-1}^{(M)})^\top$ and $\mathbf{f}^{(e)} = (f(\mathbf{z}_t^{(e)}|\mathbf{x}_{t-1}^{(1)}), \dots, f(\mathbf{z}_t^{(e)}|\mathbf{x}_{t-1}^{(M)}))^\top$, $\mathbf{q}^{(e)} = (q_t^{(1)}(\mathbf{z}_t^{(e)}), \dots, q_t^{(K)}(\mathbf{z}_t^{(e)}))^\top$. Then, setting the IS weights in Eq. (17) equal to one leads to

$$\mathbf{q}^{(e)^\top}\boldsymbol{\lambda} = \underbrace{g(\mathbf{y}_t|\mathbf{z}_t^{(e)}) \odot \mathbf{w}^\top \mathbf{f}^{(e)}}_{\widetilde{\boldsymbol{\pi}}^{(e)}}, \; e = 1, \dots, E, \tag{28}$$

where $\odot$ is elementwise multiplication and defining additionally the right-hand side to be $\widetilde{\boldsymbol{\pi}}^{(e)}$. Note that this is indeed a system of linear equations where the unknowns are the mixture weights $\boldsymbol{\lambda}$.

Remark (5): The typical APF concrete choice of $\boldsymbol{\lambda}$ of given by Eq. (13) can be obtained from Eq. (28) by setting $K = E = M$, selecting $q_t^{(m)}(\mathbf{x}_t) = f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ and $\{\mathbf{z}_t^{(m)}\}_{m=1}^M = \{\boldsymbol{\mu}_{t|t-1}^{(m)}\}_{m=1}^M$.

To proceed, we express Eq. (28) more compactly in matrix form as

$$\overbrace{\begin{bmatrix} — & \mathbf{q}^{(1)^\top} & — \\ \vdots & \vdots & \vdots \\ — & \mathbf{q}^{(E)^\top} & — \end{bmatrix}}^{E\times K} \overbrace{\begin{bmatrix} | \\ \boldsymbol{\lambda} \\ | \end{bmatrix}}^{K\times 1} = \underbrace{\overbrace{\begin{bmatrix} — & g(\mathbf{y}_t|\mathbf{z}_t^{(1)}) \odot \mathbf{f}^{(1)^\top} & — \\ \vdots & \vdots & \vdots \\ — & g(\mathbf{y}_t|\mathbf{z}_t^{(M)}) \odot \mathbf{f}^{(E)^\top} & — \end{bmatrix}}^{E\times M} \overbrace{\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix}}^{M\times 1}}_{\widetilde{\boldsymbol{\pi}}}, \quad (29)$$

defining $\mathbf{Q}$ as the $E \times K$ matrix on the left-hand side of (29) and $\widetilde{\boldsymbol{\pi}}$ as the resulting $E \times 1$ vector on the right-hand side. We can now obtain the mixture weights $\boldsymbol{\lambda}$ in a joint manner as the solution of the following constrained optimization problem

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{L}\left(\mathbf{Q}\boldsymbol{\lambda}, \boldsymbol{\pi}\right), \text{ subject to } : \boldsymbol{\lambda} \in \mathbb{R}^K_{\geqslant 0}, \quad (30)$$

where $\mathcal{L}$ represents a generic loss function. Note that $\mathcal{L}$ could be any measure of distance or discrepancy between two vectors. We optimize the unnormalized mixture weights to then normalize them afterwards, but alternative approaches could consider optimization directly in the simplex $\{\boldsymbol{\lambda} : \lambda_k \geqslant 0 \ \forall k, \boldsymbol{\lambda}^\top \mathbf{1} = 1\}$.

Concrete loss choice: Regression with non-negative least-squares. [4] implement a concrete version of the framework and consider the mean squared loss, leading to the optimization problem

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \|\mathbf{Q}\boldsymbol{\lambda} - \tilde{\boldsymbol{\pi}}\|_2^2 \quad \text{subject to } : \boldsymbol{\lambda} \in \mathbb{R}^K_{\geqslant 0} \quad (31)$$

The above is a constrained quadratic program. Therefore, it is convex and the non-negativity constraints form a convex feasible set. When $\mathbf{Q}$ has full column rank, then there is a unique solution. The resulting algorithm, following these choices of $\boldsymbol{\lambda}$, is named optimized auxiliary PF (OAPF) in [4]. Note that as in the IAPF, described in the previous section, each mixture weight formulated in Eq. (31) λ_k is *not* necessarily a good approximation to the k-th optimal resampling weight corresponding to the FA-APF, which is also only meaningful when $K = M$. Following the new perspective, the mixture weights $\boldsymbol{\lambda}$ are chosen jointly such that the resulting mixture proposal is close to the approximation to the filtering posterior.

Choice of K and kernels. We did not study adaptive algorithms to optimize the kernel parameters. Such an attempt will lead to expectation-maximization style algorithms, similar to [10] or [5], which could be made jointly convex (together with mixture weights) only under certain assumptions. The exploration of algorithms adapting mixture weights and component parameters jointly is beyond this work. The choice of the number of kernels K is driven by computational constraint, as a mixture proposal with more kernels is never going to be worse than one with fewer kernels. Implicitly, frameworks like APF and BPF constrain $K = M$ and each kernel is in some way associated with a particle at time $t-1$. In our experiments, we show that we can sometimes select $K << M$ by selecting the kernels corresponding to the evaluation points leading to largest values of $\widetilde{\boldsymbol{\pi}}^{(e)}$ from Eq. (28), i.e., values of the approximation to the unnormalized posterior.

Choice of evaluation points. The evaluation points determine the locations in the $\mathbf{x}_t$ space where the proposal is going to be matched to the approximate filtering posterior.

A natural choice is to choose the means of the transition kernels $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$, i.e., $\{\boldsymbol{\mu}_t^{(m)}\}_{m=1}^M$. This is the choice implicitly made by concrete implementations of the APF when mixture weights are not jointly optimized but selected as Eq. (13), as mentioned in a previous remark. An important consideration lies in the computational cost required to obtain the evaluation point set. In this work, we assume that the evaluation points are not newly generated samples and therefore do not incur additional costs.

Sparsity of the mixture weights. For the particular choice of loss of Eq. (31), several works [65, 55] show that non-negative least squares problem solutions become sparse as K (in our notation) increases. In our context, this can be advantageous, since if many kernels in the proposals have 0 weight, then sampling from the mixture is faster and evaluation of the IS weight also is. However, since typically we will select $K = M$ or less, the dominant factor in the IS weight cost is M, and therefore reducing K does not contribute in the $\mathcal{O}$ or Θ sense. In a non-asymptotic sense, a low K contributes to reducing the time complexity of the IS weight computation and the sampling step.

3.3. **Computational complexity and statistical efficiency.** The computational complexity as a function of the number of particles M is $\Theta(M^2)$ due to the evaluation of the marginal/MIS weight since the numerator has a sum with M terms. [44] discusses a class of methods that can be used to approximate the sum in time $\Theta(M \log M)$. [66, 70] investigated a different approximation that leads to a fast computation of the IS weights. Further on reducing this computation, [13] provide additional references. In our context, an asymptotic analysis involving E and K as well requires strategies for selecting these in practice, since they will typically both depend on M.

We note that algorithms following the schemes in Algorithm 3 will perform more point-wise density evaluations of the likelihood $g(\mathbf{y}_t|\mathbf{x}_t)$, proposal kernels $q^{(k)}(\mathbf{x}_t)$, and transition densities $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ when compared to the BPF or the standard APF implementations deriving from Algorithm 2. However, the number of particles simulated M is the same. In many applications, sampling a particle $\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t)$ has a very different cost than point-wise evaluating either the transition or the observation density. It is common for some PF algorithms to target a specific setting where some operations are more expensive than others due to a specific application, see, e.g., [48]. As of now, a default implementation of the AM-PFs benefits applications where *sampling* is significantly more costly than other operations (we generate no additional samples, just additional target or proposal evaluations). To distinguish computational cost from the number of particles/samples generated, [47] (among others) discusses the concept of statistical efficiency. An estimator is said to be more statistically efficient than another one if it reduces estimation error, given the same budget of samples. This may involve doing (potentially many) additional computational steps, but this is accepted, as long as new samples are not generated in the process. Efficient estimators in multiple importance sampling have exploited this concept to reduce variance, e.g., [26, 51]. Sampling from a mixture distribution can be implemented concretely by resampling followed by propagation of the resampled particles with the components. The time complexity of resampling schemes as a function of the number of particles is studied in e.g. [50]. A detailed analysis is beyond the scope of the present work.

4. **Numerical examples.** In this section, we present numerical examples to illustrate the performance of the AM-PF framework. The concrete algorithms (instantiations of the framework) are the IAPF and OAPF described in Section 3.1 and Section 3.2. In Section 4.1, we illustrate with a toy example on univariate distributions how the proposal constructed with AM-PF algorithms is qualitatively closer to the true posterior. This is also measured quantitatively by the χ^2 divergence in Table 1. In Section 4.2, we study a typical linear Gaussian SSM (Fig. 2). Finally, in Section 4.3 we demonstrate performance along with a study of computational cost in a Lorenz 63 model (Fig. 3). The code is available at https://github.com/nicola144/optimized_auxiliary_particle_filters.

4.1. **Toy example.** We start illustrating the benefits of our framework with a toy example where a proposal is built choosing mixture weights with the choices made by BPF, APF, IAPF ([27]) and OAPF [4]. The last two PFs exploit our perspective and are special cases of the AM-PF framework. The toy example is reproduced from [4], with some additional examples. In Table 1, we show how OAPF generally constructs proposals that are closer, in the χ^2 divergence sense, to the target distribution compared to the alternatives. In the Appendix, all the details for reproducibility can be found.

4.2. **Linear Gaussian SSM.** We now study the performance of OAPF in a typical linear Gaussian state-space model, where the exact posterior can be obtained with the Kalman Filter (KF). We focus on estimating the mean of the posterior distribution. In this setting, we can also compare against the fully adapted APF (FA-APF) [43], since its choices of optimal resampling weights and transition kernels are available in closed form. Recall that the FA-APF is only "optimal" in terms of conditional variance of the weights up to time t, so it is not guaranteed to outperform any of the algorithms ([43] in particular show an example where it is outperformed by BPF). However, it will generally perform well when compared to APF and BPF. In Fig. 1, we show the results obtained for a linear SSM in 2 dimensions with 1000 Monte Carlo replications, with transition matrix $0.2\mathbf{I}$ and observation matrix $0.3\mathbf{I}$.

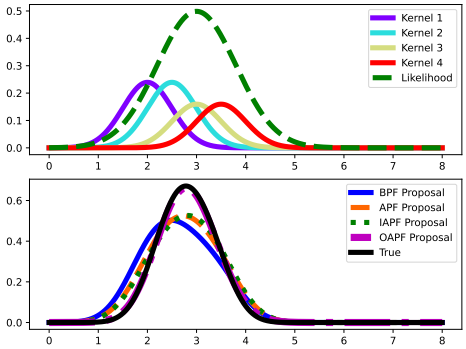
TABLE 1. Values of the Pearson χ^2 divergence between the mixture proposal and the target; lower is better.

Algorithm	χ^2 (Fig. 1a)	χ^2 (Fig. 1b)	χ^2 (Fig. 1c)	χ^2 (Fig. 1d)
BPF	1.89	0.22	2.31	1.72
APF	0.30	0.16	0.27	0.36
IAPF	0.12	0.24	0.31	0.28
OAPF	0.0043	0.09	0.08	0.08

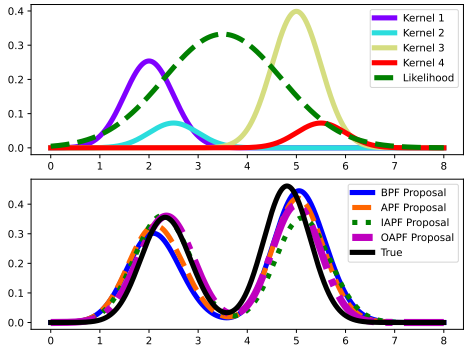
4.3. **Lorenz 63.** We now study a challenging nonlinear example, the Lorenz 63 dynamical system, very often used to benchmark PFs [1, 58, 62]. The system is described by the following stochastic differential equations for the evolution of the hidden state

$$\mathrm{d}x^{[1]} = \sigma(x^{[2]} - x^{[1]})\mathrm{d}\tau + \mathrm{d}w_{x^{[1]}} \tag{32}$$

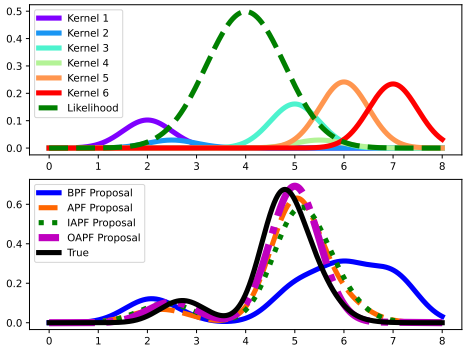
$$\mathrm{d}x^{[2])} = (\rho x^{[1]} - x^{[3]}x^{[1]} - x^{[2]})\mathrm{d}\tau + \mathrm{d}w_{x^{[2]}} \tag{33}$$



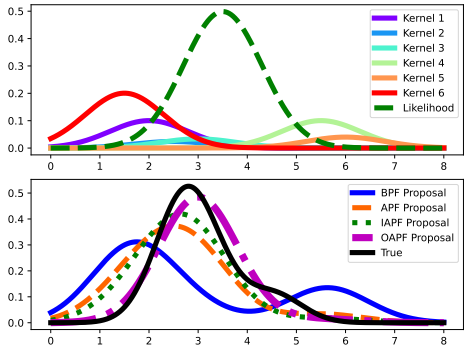
(A) In this first example we choose a unimodal posterior. Note that the choices of the AM-PF-based algorithms (i.e., IAPF and OAPF) lead to proposals that are closer to the true posterior (in black).



(B) In this example, the true posterior (in black) is bimodal. OAPF is the only algorithm that can match well both modes *simultaneously* (others only match one of them).



(C) In this example, the APF and IAPF proposals put too much mass on the right side of their modes compared to OAPF



(D) In this example, the APF and IAPF are not putting enough mass in the true mode and are tilted to the left.

FIGURE 1. **Toy Example.** In this experiment we show that the proposals of AM-PF-based algorithms (IAPF and OAPF) are closer to true posteriors compared to the competitors. In the first row, we show the transition kernels $f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ and the likelihood associated with the true filtering distribution. We calculated χ^2-divergence for these examples in Table Table 1. Note that here OAPF uses transition kernels for the proposal and their means $\boldsymbol{\mu}_{t|t-1}^{(m)}$ as evaluation points, with $K = M$.

$$\mathrm{d}x^{[3]} = (x^{[1]}x^{[2]} - \beta x^{[3]})\mathrm{d}\tau + \mathrm{d}w_{x^{[3]}} \tag{34}$$

where τ denotes continuous time, $w_{x^{[1]}}, w_{x^{[2]}}, w_{x^{[3]}}$ are independent one-dimensional standard Wiener processes and (σ, ρ, β) are parameters of the model. We use the increment Δt in the discretization and partially observe the hidden state (only the first dimension) with scalar $y_t \sim \mathcal{N}_{y_t}(x^{(1)}, \sigma_{y_t}^2 = 1)$, using standard values for (σ, ρ, β). We used standard parameters $(\sigma, \rho, \beta) = (10, 28, 2.667)$. We also used $\frac{1}{2}\mathbf{I}$ for transition covariance and $\mathbf{I}$ for observation covariance. In Fig. 3, we show how

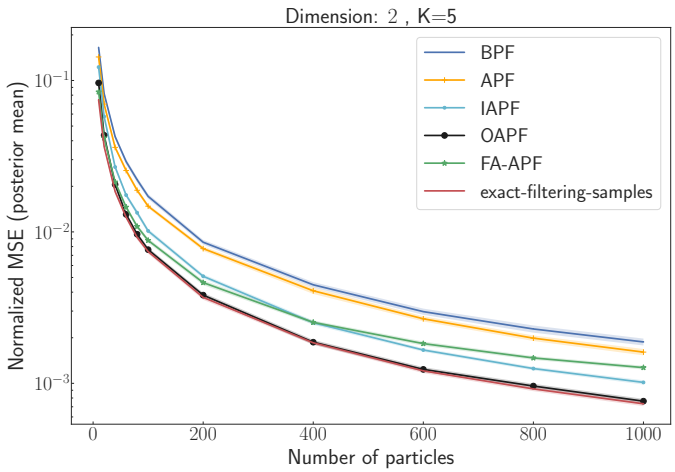


FIGURE 2. **(Linear Gaussian SSM.)** Normalized (divided by the true value) MSE for the mean of a posterior distribution in the linear Gaussian state-space model, with 100 Monte Carlo replications. In this experiment, K was set to 5 for all numbers of particles, retaining performance. We found similar qualitative results for other values of the SSM parameters.

the AM-PFs lead to better ESS values. As Fig. 3 shows, in this more challenging example, reducing the number of kernels K with our heuristic is less effective than in the linear Gaussian system. Further, we show a study of the computational cost for this example in Fig. 4. Clearly, as discussed earlier in Section 3.3, the AM-PF algorithms suffer from the M^2 computational cost. However, our implementation is on CPU (as opposed to GPU) and does not exploit recent research exactly targeted at reducing this cost in mixture PFs, see our Section 5.

5. Discussion and conclusions.

We have proposed a framework that unifies particle filters under the perspective of importance sampling with mixture proposals that can be adapted in an online manner. We have reframed the task of selecting resampling probabilities and proposal kernels to that of selecting mixture weights and components in a mixture proposal. The perspective allows us new methodological opportunities in the design of PFs, such as: detaching the number of components in the mixture from the number of particles used in the PF, which in some cases allows for some computational savings; jointly selecting the mixture weights, as opposed to separately like in the APF, so that the mixture proposal is a good approximation of the filtering distribution.

Future avenues. Our framework opens up many directions for future research. On the algorithmic side, more research is needed to define concrete choices of evaluation points, including choices that use information from both the transition densities f and the observation densities g. For example, a wide range of active learning strategies for the evaluation points could be considered [71, 33].

A theoretical analysis in terms of variance of the likelihood estimator could study the benefits of mixture weights as those proposed in this work, even with $K \neq M$, in the setting where kernels cannot be chosen to be the optimal kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(k)}, \mathbf{y}_t)$.

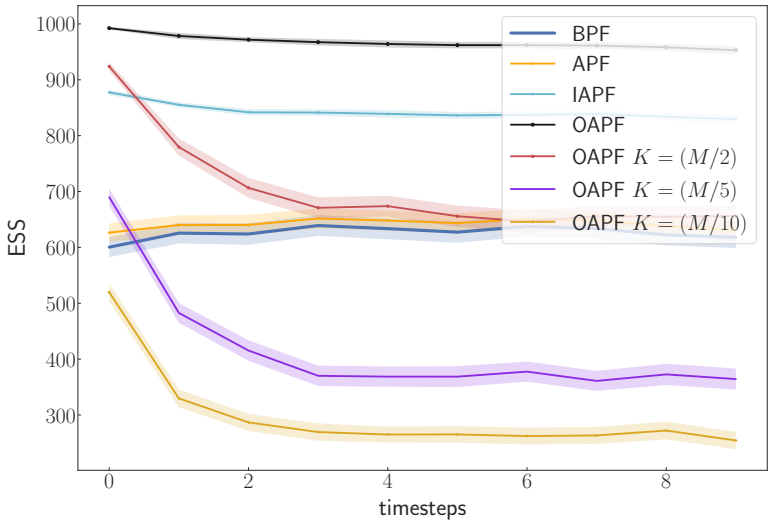


FIGURE 3. **(Lorenz 63 system.)** The plot shows effective sample size (ESS) over timesteps for 1000 Monte Carlo replications. In the legend we show the value of K, which was set based on $M = 1000$, the number of particles. In this example, we show that unlike in the linear Gaussian system, unfortunately reducing too much the number of kernels K impacts performance for these settings of the system parameters. When $K = M$, OAPF outperforms the other algorithms, which suggests that indeed the proposal constructed by the optimization of the mixture weights better matches the posterior distribution.

Optimization of the mixture components with expectation-maximization type algorithms [20, 5, 16] will likely have to consider additional sample generation into the cost of the procedure but could be interesting future work, as suggested by a reviewer. Similarly, the inclusion of gradient-based schemes such as [63, 2] could lead to better mixture proposals with the benefit of theoretical guarantees.

Finally, in the machine learning literature, there has been recent interest in the problem of jointly estimating SSMs parameters and proposal parameters (where both are given by neural networks) [5] with an end-to-end gradient-based algorithm. When a PF algorithm is used for filtering, the introduction of resampling steps poses a challenge for computing gradients w.r.t. proposal parameters, because the resampling operation is not continuous. [9] proposed to replace resampling with a deterministic transformation of the particles derived from an optimal transport problem. However, later [49] show that by employing a recently proposed method for differentiating through mixtures in the machine learning literature, it is sufficient to employ the mixture view of PF to obtain a differentiable algorithm in the abovementioned context. In this context, the AM-PF framework opens up the possibility of reducing the variance of gradient estimates with control variates similarly to [51].

[5] In this setting, called "variational" parameters, due to the objective function being a KL divergence.

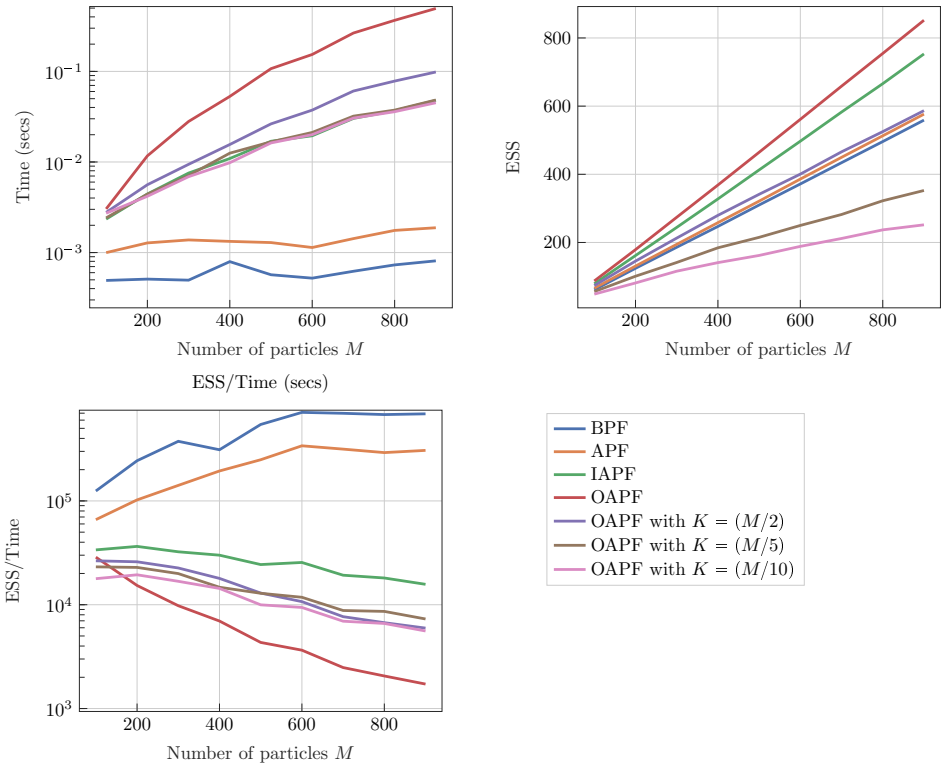


FIGURE 4. Here, we show for the Lorenz 63 experiment, (i) runtime (in seconds) per iteration in t (ii) ESS and (iii) ESS/runtime, vs the number of particles $M \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$. The AM-PF algorithms (IAPF,OAPF) clearly suffer from the M^2 computational cost as discussed in Section 3. Recall that ESS is bounded by M. Also, notice how the ESS for OAPF grows almost linearly as a function of M.

Appendix A. Appendix.

A.1. **Theoretical properties.** The theoretical properties of our estimators are analysed from the importance sampling perspective. In the case of the mixture proposals ψ_t, we assume that each time t, the support of ψ_t is a superset of the support of $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, i.e., that $\psi_t(\mathbf{x}_t) > 0$ for all $\mathbf{x}_t$ where $p(\mathbf{x}_t|\mathbf{y}_{1:t}) > 0$. Let us define the partial normalizing constants as $Z_t \triangleq p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$, the joint normalizing constant as $Z_{1:t} \triangleq p(\mathbf{y}_{1:t})$, and also $Z_{t-h:t} \triangleq p(\mathbf{y}_{t-h:t}|\mathbf{y}_{1:t-h-1})$. In the OAPF framework, we can build estimator of those quantities, e.g., the partial estimator $\widehat{Z}_\tau \triangleq \frac{1}{M}\sum_{m=1}^M \widetilde{w}_\tau^{(m)}$, the joint estimator $\widehat{Z}_{1:t} = \prod_{\tau=1}^t \widehat{Z}_\tau$, and also the estimator $\widehat{Z}_{t-h:t} = \prod_{\tau=t-h}^t \widehat{Z}_\tau$, with and the estimator $\widehat{Z}_t \triangleq \frac{1}{M}\sum_{m=1}^M \widetilde{w}_t^{(m)}$. We also assume that the estimators of all the partial normalizing constants have finite variance (see for instance [29]). We define the set of weighted samples at

time t as $\mathcal{A}_t \triangleq \{\mathbf{x}_t^{(m)}, \widetilde{w}_t^{(m)}\}_{m=1}^M$. In order to avoid ambiguities when evaluating pdfs, we define the functions $g(\mathbf{y}_t|\mathbf{x}_t) \triangleq p(\mathbf{y}_t|\mathbf{x}_t)$, $g(\mathbf{y}_t|\mathbf{x}_{t-1}) \triangleq p(\mathbf{y}_t|\mathbf{x}_{t-1})$, $g(\mathbf{y}_t, \mathbf{x}_t|\mathbf{x}_{t-1}) \triangleq p(\mathbf{y}_t, \mathbf{x}_t|\mathbf{x}_{t-1})$ and $g(\mathbf{y}_{t-h:t}, \mathbf{x}_t|\mathbf{x}_{t-1}) \triangleq p(\mathbf{y}_{t-h:t}, \mathbf{x}_t|\mathbf{x}_{t-1})$.

In the following, we show that OAPF provides an unbiased estimator of the normalizing constant $p(\mathbf{y}_{1:t})$, which follows a proof by induction, in a similar spirit as in [60], but with more generic results. In particular, here the (approximate) filtering distribution is the marginalized version of the one in [60] and is constituted by a mixture in the numerator of the importance weights (see [44] for an explanation). In OAPF the proposal density can be any mixture $\psi_t(\mathbf{x}_t)$ fulfilling the standard regularity conditions described above, hence in the denominator of the importance weights, a second mixture appears. Theorem A.3 is here the main result, and is supported by Lemmas A.1 and A.2 which we present first.

Lemma A.1. *We have that*

$$\mathbb{E}\Big[\widehat{Z}_t|\mathcal{A}_{t-1}\Big] = \sum_{m=1}^M w_{t-1}^{(m)} g(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}). \tag{35}$$

Proof.

$$\mathbb{E}\Big[\widehat{Z}_t|\mathcal{A}_{t-1}\Big] = \mathbb{E}\Big[\frac{1}{M}\sum_{m=1}^M \widetilde{w}_t^{(m)}|\mathcal{A}_{t-1}\Big] \tag{36}$$

$$= \mathbb{E}\Big[\frac{1}{M}\sum_{m=1}^M \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})\sum_{j=1}^M w_{t-1}^{(j)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\psi(\mathbf{x}_t^{(m)})}|\mathcal{A}_{t-1}\Big] \tag{37}$$

$$= \frac{1}{M}\sum_{m=1}^M \mathbb{E}\Big[\frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})\sum_{j=1}^M w_{t-1}^{(j)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\psi(\mathbf{x}_t^{(m)})}|\mathcal{A}_{t-1}\Big]. \tag{38}$$

Now, since given $\mathcal{A}_{t-1}$ the particles at time t are conditionally independent with pdf $\psi_t(\mathbf{x}_t)$, then we have that the integrals within (38) are identical:

$$\mathbb{E}\Big[\widehat{Z}_t|\mathcal{A}_{t-1}\Big] = \int \frac{g(\mathbf{y}_t|\mathbf{x}_t)\sum_{j=1}^M w_{t-1}^{(j)} f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})}{\psi(\mathbf{x}_t)}\psi(\mathbf{x}_t)d\mathbf{x}_t \tag{39}$$

$$= \int g(\mathbf{y}_t|\mathbf{x}_t)\sum_{j=1}^M w_{t-1}^{(j)} f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})d\mathbf{x}_t \tag{40}$$

$$= \sum_{j=1}^M w_{t-1}^{(j)} \int g(\mathbf{y}_t, \mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})d\mathbf{x}_t \tag{41}$$

$$= \sum_{j=1}^M w_{t-1}^{(j)} g(\mathbf{y}_t|\mathbf{x}_{t-1}^{(j)}). \tag{42}$$

□

Lemma A.2. *For any $h \in \{1, ..., t-1\}$ we have that*

$$\mathbb{E}\Big[\widehat{Z}_{t-h:t}|\mathcal{A}_{t-h-1}\Big] = \sum_{m=1}^M w_{t-h-1}^{(m)} g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1}^{(m)}). \tag{43}$$

Proof. We follow a proof by induction. First, note that (43) is true for $h = 0$ due to Lemma A.1. Then, we assume that (43) holds for a given h and we will prove

that it then holds for $h+1$. Let us start developing the left-hand side of (43) for $h+1$ by first noting that $\widehat{Z}_{t-h-1:t} = \widehat{Z}_{t-h:t}\widehat{Z}_{t-h-1}$. Then,

$$\mathbb{E}\Big[\widehat{Z}_{t-h-1:t}|\mathcal{A}_{t-h-2}\Big] = \mathbb{E}\left[\mathbb{E}\Big[\widehat{Z}_{t-h:t}|\mathcal{A}_{t-h-1}\Big]\widehat{Z}_{t-h-1}|\mathcal{A}_{t-h-2}\right] \tag{44}$$

$$= \mathbb{E}\left[\Big[\sum_{m=1}^{M} g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1}^{(m)})w_{t-h-1}^{(m)}\Big]\widehat{Z}_{t-h-1}|\mathcal{A}_{t-h-2}\right] \tag{45}$$

where we have simply substituted Eq. (43) that we assume to hold for h. Next,

$$\mathbb{E}\Big[\hat{Z}_{t-h-1:t}|\mathcal{A}_{t-h-2}\Big]$$

$$= \mathbb{E}\left[\Big[\sum_{m=1}^{M} g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1}^{(m)})\frac{\hat{w}_{t-h-1}^{(m)}}{\sum_{j=1}^{M}\hat{w}_{t-h-1}^{(j)}}\Big]\frac{1}{M}\sum_{j=1}^{M}\hat{w}_{t-h-1}^{(j)}|\mathcal{A}_{t-h-2}\right] \tag{46}$$

$$= \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M} g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1}^{(m)})\frac{g(\mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1}^{(m)})\sum_{j=1}^{M} w_{t-h-2}^{(j)} f(\mathbf{x}_{t-h-1}^{(m)}|\mathbf{x}_{t-h-2}^{(j)})}{\psi_{t-h-1}(\mathbf{x}_{t-h-1}^{(m)})}|\mathcal{A}_{t-h-2}\right] \tag{47}$$

$$= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1}^{(m)})\frac{g(\mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1}^{(m)})\sum_{j=1}^{M} w_{t-h-2}^{(j)} f(\mathbf{x}_{t-h-1}^{(m)}|\mathbf{x}_{t-h-2}^{(j)})}{\psi(\mathbf{x}_{t-h-1}^{(m)})}|\mathcal{A}_{t-h-2}\right] \tag{48}$$

where we have substituted with the importance weights $\widetilde{w}_{t-h-1}^{(m)}$ of Eq. 7 of the manuscript. Since, given $\mathcal{A}_{t-h-2}$, the particles at time t are conditionally independent with pdf $\psi_{t-h-1}(\mathbf{x}_{t-h-1})$, all M expectations are identical:

$$\mathbb{E}\Big[\hat{Z}_{t-h-1:t}|\mathcal{A}_{t-h-2}\Big]$$

$$= \int g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1})\frac{g(\mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1})\sum_{j=1}^{M} w_{t-h-2}^{(j)} f(\mathbf{x}_{t-h-1}|\mathbf{x}_{t-h-2}^{(j)})}{\psi(\mathbf{x}_{t-h-1})}\psi(\mathbf{x}_{t-h-1})d\mathbf{x}_{t-h-1} \tag{49}$$

$$= \int g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1})g(\mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1})\sum_{j=1}^{M} w_{t-h-2}^{(j)} f(\mathbf{x}_{t-h-1}|\mathbf{x}_{t-h-2}^{(j)})d\mathbf{x}_{t-h-1} \tag{50}$$

$$= \int g(\mathbf{y}_{t-h-1:t}|\mathbf{x}_{t-h-1})\sum_{j=1}^{M} w_{t-h-2}^{(j)} f(\mathbf{x}_{t-h-1}|\mathbf{x}_{t-h-2}^{(j)})d\mathbf{x}_{t-h-1} \tag{51}$$

Step (50) to (51) is justified since $\mathbf{y}_{t-h:t} \perp\!\!\!\perp \mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1}$, so we can replace $g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1})$ in 50 with $g(\mathbf{y}_{t-h:t}|\mathbf{y}_{t-h-1}, \mathbf{x}_{t-h-1})$ and then $g(\mathbf{y}_{t-h-1:t}|\mathbf{x}_{t-h-1}) = g(\mathbf{y}_{t-h:t}|\mathbf{x}_{t-h-1})g(\mathbf{y}_{t-h-1}|\mathbf{x}_{t-h-1})$ follows by the chain rule. Next,

$$= \sum_{j=1}^{M} w_{t-h-2}^{(j)} \int g(\mathbf{y}_{t-h-1:t}, \mathbf{x}_{t-h-1}|\mathbf{x}_{t-h-2}^{(j)})d\mathbf{x}_{t-h-1} \tag{52}$$

$$= \sum_{j=1}^{M} w_{t-h-2}^{(j)} g(\mathbf{y}_{t-h-1:t}|\mathbf{x}_{t-h-2}^{(j)}) \tag{53}$$

which is the right-hand side of (43). □

Theorem A.3. *The OAPF estimator of the normalizing constant is unbiased, i.e.,* $\mathbb{E}[\widehat{Z}_{1:t}] = p(\mathbf{y}_{1:t})$.

Proof. The unbiasedness is a consequence of Lemma 2 with $h = t-1$. □

Now we look at the variance of the normalizing constant estimators. First, we establish a superiority in performance (i.e., equal or less variance) of the OAPF importance weights. This result is also used below to prove the convergence of the estimators by standard results in particle filtering.

Let us particularize importance weights in OAPF for the case with $K = M$ as

$$\widetilde{w}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{i=1}^M w_{t-1}^{(i)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i)})}{\sum_{i=1}^M \lambda_t^{(i)} q_t^{(i)}(\mathbf{x}_t^{(m)}|\bar{\mathbf{x}}_{t-1}^{(i)})}. \tag{54}$$

We also consider the generalized APF weights given by

$$\widetilde{v}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}^{(m)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{\lambda_t^{(m)} q_t^{(m)}(\mathbf{x}_t^{(m)}|\bar{\mathbf{x}}_{t-1}^{(m)})}. \tag{55}$$

These are generalized in the sense that the concrete APF described in the main paper is obtained by setting $\lambda_t^{(m)} \propto w_{t-1}^{(m)} g(\mathbf{y}_t|\boldsymbol{\mu}_t^{(m)})$ and propagating particles with transition kernels $f(\cdot)$, thus our following discussion holds for any choice of $\lambda_t^{(m)}$.

Lemma A.4. *The conditional variance of $\widehat{Z}_t^{OAPF}$ using the OAPF weights in (54) is always less or equal than the same estimator $\widehat{Z}_t^{APF}$ using the APF weights in (55).*

Proof. First, note that $\widetilde{v}_t^{(m)}$ can be interpreted as an importance weight in an extended space on $\mathbf{x}_t$ and the auxiliary variable m (see for instance [44, Section 3.1] and [59, 36]). Next, $\widetilde{w}_t^{(m)}$ can be interpreted as a version of $\widetilde{v}_t^{(m)}$ where both in the numerator (approximate filtering pdf) and denominator (proposal pdf), the auxiliary variable has been marginalized. Then, the variance inequality for each importance weight holds from the application of the variance decomposition lemma (also known as law of total variance). This proof generalizes the result in [44] for any set of mixture weights $\{\lambda_t^{(m)}\}_{m=1}^M$, with $\sum_{j=1}^M \lambda_t^{(j)}$ and $\lambda_t^{(m)} \geqslant 0$, for all m. Finally, since both $\hat{Z}_t^{\text{OAPF}}$ and $\hat{Z}_t^{\text{APF}}$ are constructed as the average of the OAPF and APF weights, respectively, the conditional variance of $\hat{Z}_t^{\text{OAPF}}$ is necessarily upper-bounded by that of $\hat{Z}_t^{\text{APF}}$. □

We now address the consistency of the normalizing constant, $\widehat{Z}_{1:t}$, and the self-normalized IS (SNIS) estimator $\widehat{I}(h_t) = \sum_{m=1}^M w_t^{(m)} h_t(\mathbf{x}_t^{(m)})$.

Corollary A.5. *The OAPF estimator of the normalizing constant $\widehat{Z}_{1:t}$ and the SNIS estimator $\widehat{I}(h_t)$ are consistent, i.e., $\lim_{M\to\infty} \widehat{Z}_{1:t} = p(\mathbf{y}_{1:t})$ and $\lim_{M\to\infty} \widehat{I}(h_t) = I(h_t)$ a.s. (almost surely) for a finite t.*

Proof. The consistency of $\widehat{Z}_{1:t}$ is a consequence of its unbiasedness, proved in Theorem A.3, and the variance inequality in Lemma A.4, which ensures the variance convergence to zero a.s. when $N \to \infty$ since the APF, which upper-bounds its variance, is also consistent [22, Section 3.6]. A similar argumentation can be done for the SNIS estimator $\widehat{I}(h_t)$. Note that the SNIS estimator can be re-expressed as $\widehat{I}(h_t) = \sum_{m=1}^M \frac{\widetilde{w}_t^{(m)}}{M\widehat{Z}_t} h_t(\mathbf{x}_t^{(m)}) = \frac{1}{M}\sum_{m=1}^M \frac{\widetilde{w}_t^{(m)}}{\widehat{Z}_t} h_t(\mathbf{x}_t^{(m)})$. Since $\widehat{Z}_t$ is a consistent estimator of $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$, the denominator converges to $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$ while the numerator converges to $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})I(h_t)$, when $N \to \infty$. Therefore, the ratio converges to $I(h_t)$ a.s. □

A.2. **Experimental details.** The Python code is available at https://github.com/nicola144/optimized_auxiliary_particle_filters.

A.2.1. *Toy example.* We provide all necessary parameters to reproduce Fig. 1 in the main paper. We recall that in this toy example we do the Bayesian recursion from $t-1$ to t with $M=4$ particles.

- In Fig. 1a, we have set the particles $\{\bar{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^{M=4} = \{2, 2.5, 3, 3.5\}$, the normalized weights $\{3/10, 3/10, 1/5, 1/5\}$, likelihood centered at 3, and $\sigma_{\text{lik}} = 0.8$, and $\sigma_{\text{kern}} = 0.5$.
- In Fig. 1b, $\{\bar{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^{M=4} = \{2, 2.5, 5, 5.5\}$, the normalized weights are $\{7/22, 1/11, 1/2, 1/11\}$, the likelihood is centered at 3.5, and $\sigma_{\text{lik}} = 1.2$, and $\sigma_{\text{kern}} = 0.5$.
- In Fig. 1c, $\{\bar{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^{M=6} = \{2., 2.5, 5., 5.5, 6, 7\}$, the unnormalized weights are $\{7/22, 1/11, 1/2, 1/11, 9/12, 8/11\}$, the likelihood is centered at 4, and $\sigma_{\text{lik}} = 0.8$, and $\sigma_{\text{kern}} = 0.5$.
- In Fig. 1d, $\{\bar{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^{M=6} = \{2., 2.5, 3., 5.5, 6, 1.5\}$, the unnormalized weights are $\{1, 6/25, 1/3, 1, 4/10, 2\}$, the likelihood is centered at 3.5, and $\sigma_{\text{lik}} = 0.8$, and $\sigma_{\text{kern}} = 0.8$.

The proposals of all algorithms are then calculated as:

$$\sum_{m=1}^{M} \lambda_t^{(m)} f(\mathbf{x}_t|\boldsymbol{\mu}_{t-1}^{(m)}), \tag{56}$$

where the mixture weights $\lambda_t^{(m)}$ for BPF are $w_{t-1}^{(m)}$, for APF are $\propto w_{t-1}^{(m)} g(\mathbf{y}_t|\boldsymbol{\mu}_{t-1}^{(m)})$, for IAPF $\propto g(\mathbf{y}_t|\boldsymbol{\mu}_{t-1}^{(m)}) \sum_{m=1}^{M} w_{t-1}^{(m)} f(\boldsymbol{\mu}_{t-1}^{(m)}|\mathbf{x}_{t-1}^{(m)}) / \sum_{m=1}^{M} f(\boldsymbol{\mu}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})$ and finally for OAPF they are the solution to the non-negative least squares optimization problem. As specified in the main paper and can be seen from (56), we used transition kernels as proposal kernels for OAPF. Moreover, we used the centers of the transition kernels $\boldsymbol{\mu}_t^{(m)}$ as evaluation points, which in this case they correspond to the resampled particles $\{\bar{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^{M=4}$.

REFERENCES

[1] Ö. D. Akyildiz and J. Míguez, Nudging the particle filter, *Statistics and Computing*, **30** (2020), 305-330.

[2] Ö. D. Akyildiz and J. Míguez, Convergence rates for optimised adaptive importance samplers, *Statistics and Computing*, **31** (2021), Paper No. 12, 17 pp.

[3] A. Bouchard-Côté, S. Sankararaman and M. I. Jordan, Phylogenetic inference via sequential Monte Carlo, *Systematic Biology*, **61** (2012), 579-593.

[4] N. Branchini and V. Elvira, Optimized auxiliary particle filters: Adapting mixture proposals via convex optimization, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, Proceedings of Machine Learning Research*, **161** (2021), 1289-1299, https://proceedings.mlr.press/v161/branchini21a.html.

[5] O. Cappé, R. Douc, A. Guillin, J.-M. Marin and C. P. Robert, Adaptive importance sampling in general mixture classes, *Statistics and Computing*, **18** (2008), 447-459.

[6] L. Carlone, J. Du, M. K. Ng, B. Bona and M. Indri, Active SLAM and exploration with particle filters using Kullback-Leibler divergence, *Journal of Intelligent & Robotic Systems*, **75** (2014), 291-311.

[7] N. Chopin, Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference, *The Annals of Statistics*, **32** (2004), 2385-2411.

[8] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*, Springer Ser. Statist., Springer, Cham, 2020.

[9] A. Corenflos, J. Thornton, G. Deligiannidis and A. Doucet, Differentiable particle filtering via entropy-regularized optimal transport, *International Conference on Machine Learning*, (2021), 2100-2111.

[10] J. Cornebise, E. Moulines and J. Olsson, Adaptive sequential Monte Carlo by means of mixture of experts, *Statistics and Computing*, **24** (2014), 317-337.
[11] D. Creal, A survey of sequential Monte Carlo methods for economics and finance, *Econometric Reviews*, **31** (2012), 245-296.
[12] D. Crisan and K. Li, Generalised particle filters with Gaussian mixtures, *Stochastic Processes and their Applications*, **125** (2015), 2643-2673.
[13] F. R. Crucinio and A. M. Johansen, Properties of marginal sequential Monte Carlo methods, *Statist. Probab. Lett.*, **203** (2023), Paper No. 109914, 8 pp.
[14] C. Dai, J. Heng, P. E. Jacob and N. Whiteley, An invitation to sequential Monte Carlo samplers, *Journal of the American Statistical Association*, **117** (2022), 1587-1600.
[15] D. Dardari, P. Closas and P. M Djurić, Indoor tracking: Theory, methods, and technologies, *IEEE Transactions on Vehicular Technology*, **64** (2015), 1263-1278.
[16] K. Daudel, Mixture weights optimisation for alpha-divergence variational inference, *Advances in Neural Information Processing Systems*, **34** (2021), 4397-4408.
[17] P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo samplers, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68** (2006), 411-436.
[18] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo and J. Miguez, Particle filtering, *IEEE Signal Processing Magazine*, **20** (2003), 19-38.
[19] R. Douc and O. Cappé, Comparison of resampling schemes for particle filtering, *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, (2005), 64-69.
[20] R. Douc, A. Guillin, J.-M. Marin and C. P. Robert, Minimum variance importance sampling via population Monte Carlo, *ESAIM: Probability and Statistics*, **11** (2007, 427-447.
[21] A. Doucet, *On Sequential Simulation-based Methods for Bayesian Filtering*, Department of Engineering, University of Cambridge, 1998.
[22] A. Doucet and A. M. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering*, **12** (2009), 656-704.
[23] A. Doucet and A. Lee, Sequential Monte Carlo methods, *Handbook of Graphical Models, Chapman & Hall/CRC Handb. Mod. Stat. Methods CRC Press, Boca Raton, FL*, (2018), 165-189.
[24] A. Doucet, S. Godsill and C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, **10** (2000), 197-208.
[25] J. Du, L. Carlone, M. K. Ng, B. Bona and M. Indri, A comparative study on active SLAM and autonomous exploration with particle filters, *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (2011), 916-923.
[26] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo, Efficient multiple importance sampling estimators, *IEEE Signal Processing Letters*, **22** (2015), 1757-1761.
[27] V. Elvira, L. Martino, M. F. Bugallo and P. M. Djurić, In search for improved auxiliary particle filters, *2018 26th European Signal Processing Conference (EUSIPCO)*, (2018), 1637-1641.
[28] V. Elvira, L. Martino, M. F. Bugallo and P. M. Djuric, Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes], *IEEE Signal Processing Magazine*, **36** (2019), 145-152.
[29] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo, Generalized multiple importance sampling, *Statistical Science*, **34** (2019), 129-155.
[30] V. Elvira, L. Martino and C. P. Robert, Rethinking the effective sample size, *International Statistical Review*, **90** (2022), 525-550.
[31] P. Fearnhead, *Sequential Monte Carlo Methods in Filter Theory*, PhD thesis, University of Oxford, 1998.
[32] P. Fearnhead and H. R. Künsch, Particle filters and data assimilation, *Annual Review of Statistics and Its Application*, **5** (2018), 421-449.
[33] F. L. Fernández, L. Martino, V. Elvira, D. Delgado and J. López-Santiago, Adaptive quadrature schemes for Bayesian inference via active learning, *IEEE Access*, **8** (2020), 208462-208483.
[34] D. Fox, KLD-sampling: Adaptive particle filters, *Advances in Neural Information Processing Systems*, (2001), 713.
[35] W. R. Gilks and C. Berzuini, Following a moving target-Monte Carlo inference for dynamic Bayesian models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **63** (2001), 127-146.
[36] S. Godsill, Particle filtering: The first 25 years and beyond, *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2019), 7760-7764.

[37] S. Godsill and T. Clapp, Improvement strategies for Monte Carlo particle filters, *Sequential Monte Carlo Methods in Practice, Stat. Eng. Inf. Sci. Springer-Verlag, New York*, (2001), 139-158.
[38] N. J. Gordon, D. J. Salmond and A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE proceedings-F (Radar and Signal Processing)*, **140** (1993), 107-113.
[39] P. Guarniero, A. M. Johansen and A. Lee, The iterated auxiliary particle filter, *Journal of the American Statistical Association*, **112** (2017), 1636-1647.
[40] J. Heng, A. N. Bishop, G. Deligiannidis and A. Doucet, Controlled sequential Monte Carlo, *Annals of Statistics*, **48** (2020), 2904-2929.
[41] J. D. Hol, T. B. Schon and F. Gustafsson, On resampling algorithms for particle filters, *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, (2006), 79-82.
[42] A. M. Johansen, Sequential monte carlo: Particle filters and beyond, *Wiley StatsRef: Statistics Reference Online*, (2014), 1-16.
[43] A. M. Johansen and A. Doucet, A note on auxiliary particle filters, *Statistics & Probability Letters*, **78** (2008), 1498-1504.
[44] M. Klaas, N. de Freitas and A. Doucet, Toward practical N^2 Monte Carlo: The marginal particle filter, *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, (2005), 308-315.
[45] H. Koptagel, O. Kviman, H. Melin, N. Safinianaini and J. Lagergren, VaiPhy: A variational inference based algorithm for phylogeny, *Advances in Neural Information Processing Systems*, (2022).
[46] J. Kronander and T. B. Schön, Robust auxiliary particle filters using multiple importance sampling, *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, (2014), 268-271.
[47] J. Kuntz, F. R. Crucinio and A. M. Johansen, Product-form estimators: Exploiting independence to scale up Monte Carlo, *Statistics and Computing*, **32** (2022), Paper No. 12, 22 pp.
[48] S. Lacoste-Julien, F. Lindsten and F. Bach, Sequential kernel herding: Frank-wolfe optimization for particle filtering, *Artificial Intelligence and Statistics*, (2015), 544-552.
[49] J. Lai, J. Domke and D. Sheldon, Variational marginal particle filters, *International Conference on Artificial Intelligence and Statistics*, (2022), 875-895.
[50] T. Li, M. Bolic and P. M. Djuric, Resampling methods for particle filtering: Classification, implementation, and strategies, *IEEE Signal Processing Magazine*, **32** (2015), 70-86.
[51] W. Li, R. Chen and Z. Tan, Efficient sequential Monte Carlo with multiple proposals and control variates, *Journal of the American Statistical Association*, **111** (2016), 298-313.
[52] Y. Li, L. Zhao and M. Coates, Particle flow for particle filtering, *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2016), 3979-3983.
[53] F. Lindsten and T. B. Schön, Backward simulation methods for Monte Carlo statistical inference, *Foundations and Trends® in Machine Learning*, **6** (2013), 1-143.
[54] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Ser. Statist., Springer-Verlag, New York, 2001.
[55] N. Meinshausen, Sign-constrained least squares estimation for high-dimensional regression, *Electronic Journal of Statistics*, **7** (2013), 1607-1631.
[56] C. A. Naesseth, F. Lindsten and T. B. Schön, Elements of sequential Monte Carlo, *Foundations and Trends® in Machine Learning*, **12** (2019), 307-392.
[57] A. B. Owen, *Monte Carlo Theory, Methods and Examples*, 2013.
[58] S. Pérez-Vieites and J. Míguez, Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models, *Signal Processing*, **189** (2021), 108295.
[59] M. K. Pitt and N. Shephard, Filtering via simulation: Auxiliary particle filters, *Journal of the American Statistical Association*, **94** (1999), 590-599.
[60] M. K. Pitt, R. dos S. Silva, P. Giordani and R. Kohn, On some properties of Markov chain Monte Carlo simulation methods based on the particle filter, *Journal of Econometrics*, **171** (2012), 134-151.
[61] S. Reich, A nonparametric ensemble transform method for Bayesian inference, *SIAM Journal on Scientific Computing*, **35** (2013), A2013-A2024.
[62] H. Ruzayqat, A. Er-Raiy, A. Beskos, D. Crisan, A. Jasra and N. Kantas, A lagged particle filter for stable filtering of certain high-dimensional state-space models, *SIAM/ASA Journal on Uncertainty Quantification*, **10** (2022), 1130-1161.

[63] E. K. Ryu and S. P. Boyd, Adaptive importance sampling via stochastic convex programming, arXiv preprint, (2014), arXiv:1412.4845.
[64] S. Särkkä. *Bayesian Filtering and Smoothing*, IMS Textb., 3. Cambridge University Press, Cambridge, 2013.
[65] M. Slawski and M. Hein, Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization, *Electronic Journal of Statistics*, **7** (2013), 3004-3056.
[66] O. Straka and J. Duník, Efficient implementation of marginal particle filter by functional density decomposition, *2022 25th International Conference on Information Fusion (FUSION)*, (2022), 01-08.
[67] S. Thrun, Particle filters in robotics, *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc.*, (2002), 511-518.
[68] S. Thrun, D. Fox and W. Burgard, Monte Carlo localization with mixture proposal distribution, *AAAI/IAAI*, (2000), 859-865.
[69] S. Thrun, D. Fox, W. Burgard and F. Dellaert, Robust Monte Carlo localization for mobile robots, *Artificial Intelligence*, **128** (2001), 99-141.
[70] P. Tichavskỳ, O. Straka and J. Duník, Grid-based Bayesian filters with functional decomposition of transient density, *IEEE Transactions on Signal Processing*, **71** (2023), 92-104.
[71] L. M. M. van den Bos, B. Sanderse and W. A. A. M. Bierbooms, Adaptive sampling-based quadrature rules for efficient Bayesian prediction, *Journal of Computational Physics*, **417** (2020), 109537, 27 pp.
[72] P. J. Van Leeuwen, H. R. Künsch, L. Nerger, R. Potthast and S. Reich, Particle filters for high-dimensional geoscience applications: A review, *Quarterly Journal of the Royal Meteorological Society*, **145** (2019), 2335-2365.
[73] N. Whiteley and A. M. Johansen, Auxiliary particle filtering: Recent developments, *Bayesian Time Series Models, Cambridge University Press, Cambridge*, (2011), 52-81.
[74] N. Whiteley and A. Lee, Twisted particle filters, *The Annals of Statistics*, **42** (2014), 115-141.
[75] A. Wigren, J. Wågberg, F. Lindsten, A. G. Wills and T. B. Schön, Nonlinear system identification: Learning while respecting physical models using a sequential Monte Carlo method, *IEEE Control Systems Magazine*, **42** (2022), 75-102.
[76] A. G. Wills and T. B. Schön, Sequential Monte Carlo: A unified review, *Annual Review of Control, Robotics, and Autonomous Systems*, **6** (2023).
[77] T. Yang, P. G. Mehta and S. P. Meyn, Feedback particle filter, *IEEE Transactions on Automatic Control*, **58** (2013), 2465-2480.
[78] G. Zanella and G. Roberts, Scalable importance tempering and Bayesian variable selection, *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **81** (2019), 489-517.

Received May 2023; revised February 2024; early access April 2024.