

---

# Optimized Auxiliary Particle Filters: adapting mixture proposals via convex optimization

---

Nicola Branchini<sup>1</sup>

Víctor Elvira<sup>1</sup>

<sup>1</sup>University of Edinburgh

## Abstract

Auxiliary particle filters (APFs) are a class of sequential Monte Carlo (SMC) methods for Bayesian inference in state-space models. In their original derivation, APFs operate in an extended state space using an auxiliary variable to improve inference. In this work, we propose *optimized auxiliary particle filters*, a framework where the traditional APF auxiliary variables are interpreted as weights in a importance sampling mixture proposal. Under this interpretation, we devise a mechanism for proposing the mixture weights that is inspired by recent advances in multiple and adaptive importance sampling. In particular, we propose to select the mixture weights by formulating a convex optimization problem, with the aim of approximating the filtering posterior at each timestep. Further, we propose a weighting scheme that generalizes previous results on the APF (Pitt et al. 2012), proving unbiasedness and consistency of our estimators. Our framework demonstrates significantly improved estimates on a range of metrics compared to state-of-the-art particle filters at similar computational complexity in challenging and widely used dynamical models.

## 1 INTRODUCTION

State-space models (SSMs) allow a mathematical description of complex dynamical systems which are very relevant in computational statistics, machine learning and signal processing, among many other fields [Särkkä, 2013]. Particle filters (PF) or sequential Monte Carlo methods (SMC) are the *de facto* family of algorithms to perform inference tasks in virtually any SSM, e.g., filtering, prediction, or parameter estimation [Doucet et al., 2001]. PFs have been used for solving complex real-world problems in robotics [Thrun,

2002], object tracking [Vlassis et al., 2002, Wardhana et al., 2013] and image processing [Nummiaro et al., 2003]. PFs are also used for problems beyond the classical SSM setting. For instance, they have been recently applied in reinforcement learning [Maddison et al., 2017, Wang et al., 2020, Piché et al., 2019], generative modelling [Lawson et al., 2018, Le et al., 2018], and more generally for approximate Bayesian inference in large probabilistic models [Gu et al., 2015, Naesseth et al., 2018, Ma et al., 2020]. PFs are Monte Carlo methods that approximate probability density functions (pdfs) of interest with  $M$  particles. The *bootstrap PF* (BPF) [Gordon et al., 1993] is the most popular algorithm, because of its simplicity and reasonable performance in several settings. However, alternatives are needed for challenging applications that require models with complex posterior distributions. Most notably, the *auxiliary PF* (APF) [Pitt and Shephard, 1999] was designed to make use of the observation before the simulation of the particles.

In this paper, we develop a framework named *optimized APF* (OAPF) for accurate inference in SSMs. The OAPF framework implements a mixture proposal sampling and an associated weighting scheme at each time step within the PF, allowing for variance reduction in the importance weights, the key aim in SMC methods [Doucet and Johansen, 2009].

The structure of the paper is as follows. In Section 2, we review SSMs and give a brief overview on PFs. In Section 3, we derive our OAPF framework, discussing the design choices and providing a theoretical analysis of its estimators. In Section 5, we show improved results against common particle filters and the recent improved APF [Elvira et al., 2018] in challenging and widely used nonlinear state-space models such as a stochastic Lorenz 63 model and a multivariate stochastic volatility model. We conclude the paper in Section 6 with some final remarks.

**Contributions.** (1) We develop the optimized auxiliary particle filter (OAPF) framework, which encompasses other particle filters as special cases and allows the development of new algorithms with improved estimators. Our framework

has a flexible mixture proposal distribution which appears in the importance weights, provably reducing their variance.

(2) We prove that the resulting marginal likelihood estimators are unbiased and consistent, generalizing the APF estimator in [Pitt et al., 2012].

(3) We propose strategies to select kernels and mixture weights in the proposal. The mixture weights are optimized by matching proposal and posterior at a set of relevant points. Crucially, this allows us to find mixture weights as a solution to a *convex* optimization problem. Therefore, our strategy allows for optimizing the proposal in very generic models (transition and observation pdfs), while avoiding black-box non-convex optimization methods that are common in for instance in variational inference [Archer et al., 2015, Dieng et al., 2017]. Further, we allow for a flexible choice of the number of kernels, detaching this choice from the number of particles unlike previous works (see for instance [Elvira et al., 2019a]).

(4) We propose specific implementations of our framework and show their effectiveness with widely used state-space models. We compare to BPF, APF and to the improved APF (IAPF) [Elvira et al., 2018], a recent algorithm which provides the state-of-the-art in terms of importance weight variance. We show evidence for better estimates in OAPF with similar computational complexity.

## 2 BACKGROUND

### 2.1 STATE-SPACE MODELS AND PARTICLE FILTERING

State-space models (SSM) describe the temporal evolution of a system in a probabilistic manner. They are composed of a stochastic discrete-time Markovian process of a (potentially multivariate) *hidden state*  $\{\mathbf{x}_t\}_{t \geq 1}$ , which can only be observed via corresponding noisy measurements  $\{\mathbf{y}_t\}_{t \geq 1}$ . SSMs are fully specified by a prior probability density function (pdf),  $p(\mathbf{x}_0)$ , and by the *transition* and *observation* kernels,  $f(\mathbf{x}_t|\mathbf{x}_{t-1})$  and  $g(\mathbf{y}_t|\mathbf{x}_t)$ , respectively, defined for  $t \geq 1$ . In these models, the *filtering* task consists in the sequential estimation of the filtering density  $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ , as well as expectations of the form  $I(h_t) = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[h_t(\mathbf{x}_t)] = \int h_t(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t$ , for (integrable) functions of interest. For most SSMs of interests, the filtering pdf is intractable and one needs to resort to approximate inference. In this context, particle filters (PFs) are the most popular inferential methods, approximating the filtering pdf with a set of random particles (Monte Carlo samples). PFs are a sequential implementation of importance sampling (IS), generating at each time step  $M$  particles  $\{\mathbf{x}_t^{(m)}\}_{m=1}^M$  from a proposal pdf  $q(\mathbf{x}_t)$  and assigning them normalized importance weights  $w_t^{(m)}$ . The unnormalized importance weights can be computed by updating the

previous weights as

$$\tilde{w}_t^{(m)} = w_{t-1}^{(m)} \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{y}_t, \mathbf{x}_{t-1}^{(m)})}, \quad (1)$$

which can be derived by factorizing a joint proposal  $q(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t})q(\mathbf{x}_t|\mathbf{y}_t, \mathbf{x}_{t-1})$  and targeting joint posterior  $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$  [Särkkä, 2013]. Therefore, a particle filter maintains a set of normalized weights and particles  $\{w_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m=1}^M$  as a representation of the filtering pdf, updating weights at each time step with as in Eq. (1). The most popular choice for  $q(\mathbf{x}_t|\mathbf{y}_t, \mathbf{x}_{t-1})$  is  $f(\mathbf{x}_t|\mathbf{x}_{t-1})$  and leads to the bootstrap particle filter (BPF) [Gordon et al., 1993]. The advantage of this choice is that the weights in (1) simply become  $w_{t-1}^{(m)}g(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ . In practice, particle filters suffer from the *weight degeneracy* effect [Särkkä, 2013], consisting on few normalized weights taking all probability mass (i.e., the posterior is approximated with very few samples). In the BPF, a resampling step is introduced to mitigate this effect. In some implementations, the resampling step is performed only when the effective sample size  $\text{ESS} = \frac{1}{\sum_{m=1}^M (w_t^{(m)})^2}$  is below some threshold [Doucet et al., 2001, Doucet and Johansen, 2009, Särkkä, 2013].

### 2.2 AUXILIARY PARTICLE FILTERS

Auxiliary PFs (APFs) were introduced to alleviate some of the limitations of existing PF methods [Pitt and Shephard, 1999]. For instance, it is well known that informative likelihoods often impact negatively the ability of the standard BPF to reconstruct the filtering pdf [Doucet and Johansen, 2009, Johansen and Doucet, 2008, Whiteley and Johansen, 2011].<sup>1</sup> Intuitively, the reason is that the resampling step at the end of the recursion at time  $t - 1$  does not take into account the new observation  $\mathbf{y}_t$ . In the standard APF, the resampling step at  $t - 1$  is delayed until the new observation  $\mathbf{y}_t$  is available. Then the resampling is performed with modified unnormalized weights

$$\tilde{\lambda}_t^{(m)} = w_{t-1}^{(m)}g(\mathbf{y}_t|\boldsymbol{\mu}_t^{(m)}), \quad \lambda_t^{(m)} = \frac{\tilde{\lambda}_t^{(m)}}{\sum_{i=1}^M \tilde{\lambda}_t^{(i)}}, \quad (2)$$

where  $\boldsymbol{\mu}_t^{(m)} = \mathbb{E}_{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t]$ . Then the particles are propagated using the transition kernel  $f(\mathbf{x}_t|\mathbf{x}_{t-1})$  as in BPF. Finally, the importance weights are chosen as

$$\tilde{w}_{t-1}^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{g(\mathbf{y}_t|\boldsymbol{\mu}_t^{(i^{(m)})})}, \quad (3)$$

where  $i^{(m)}$  denotes the index of the ancestor that generates the  $m$ -th resampled particle. Intuitively, this can be seen

<sup>1</sup>Informally, an informative likelihood refers to a peaky likelihood that heavily influences the shape of the posterior.

as scaling down the BPF weights, taking into account that particles have been already resampled in large number in regions of high likelihood. A different interpretation of APFs<sup>2</sup> is possible from the multiple importance sampling (MIS) perspective [Elvira et al., 2019a]. Note that MIS refers to the different sampling and weighting schemes that are possible in the presence of multiple proposals in IS [Veach and Guibas, 1995, Elvira et al., 2019b]. In this perspective, a resampling step followed by a propagation step is considered to be simply a single sampling step from a mixture pdf. The improved APF (IAPF) [Elvira et al., 2018] exploits the MIS interpretation so that the weight of the  $m$ -th proposal,  $\lambda_t^{(m)}$ , depends on the location of other particles  $j \neq m$ . The MIS perspective is related to auxiliary marginal particle filters (AMPF) [Klaas et al., 2005], where a similar importance weight is derived, but  $\lambda_t^{(m)}$  is chosen as in APF. It is worth noting that [Fearnhead, 1998, Chapter 4, Section 3.2] earlier analysed the basic idea behind AMPF. The AMPF interprets that the inference is performed in the marginal space of  $\mathbf{x}_t$  (marginalizing the auxiliary variable), which guarantees to reduce (in a non-strict sense) the variance of the importance weights (it is a Rao-Blackwellization that can be proved by the variance decomposition lemma).

**Optimality criteria for APF.** A version of the APF known as the *fully adapted* APF (FA-APF) is considered to implement a locally optimal choice. Its implementation requires the computation of the (generally) intractable distributions  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$  and  $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t})$ . While often presented as the *optimal choice*, in Doucet and Johansen [2009] the FA-APF is shown to provide worse estimators than the BPF in one example. The reason is that FA-APF minimizes the variance considering only one step ahead, as explained thoroughly in [Chopin and Papaspiliopoulos, 2020]. Therefore, the intractability of FA-APF as well as its only relative optimality motivates the search for better PFs.

### 3 OPTIMIZED AUXILIARY PARTICLE FILTERS

#### 3.1 THE OAPF FRAMEWORK

In this section, we present our new framework for *optimized auxiliary particle filters* (OAPFs). The OAPF framework extends the MIS perspective, considering a generic mixture as proposal where all samples are (independently) simulated. We consider the generic mixture proposal at each  $t$  as

$$\psi_t(\mathbf{x}_t) = \sum_{k=1}^K \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t), \quad (4)$$

<sup>2</sup>Note that it is also possible to refer to as APF to a generic PF with  $\lambda_t$  being a free choice. Eq. (2) is an approximation to  $p(\mathbf{y}_t|\mathbf{x}_t)$

with associated mixture weights  $\lambda_t^{(k)}$ . To the best of our knowledge, the OAPF is the first method to detach the choice of  $K$  from the number of samples  $M$  (i.e.,  $K \neq M$  in the general case).

---

#### Algorithm 1: Optimized Auxiliary Particle Filter

---

**Input:** prior, transition, and observation pdfs  $p(\mathbf{x}_0)$ ,  $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ ,  $g(\mathbf{y}_t|\mathbf{x}_t)$ , and sequence of observations  $\mathbf{y}_{1:T}$

**Output:** set of weighted samples for each time step  $\{\mathbf{x}_t^{(m)}, \tilde{w}_t^{(m)}\}_{m=1, t=1}^{M, T}$

- 1 Draw  $M$  samples from prior:  $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$  and set  $w_0^{(m)} = 1/M$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     (a) *optimization step*: optimize the mixture proposal by selecting  $K$  kernels  $q_t^{(k)}$  and choosing their associated mixture weight  $\lambda_t^{(k)}$  that compose the mixture proposal  $\psi_t$  (see Section 3.3)
- 4     (b) *sampling step*: simulate  $M$  particles  $\mathbf{x}_t^{(m)}$  from the proposal as

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t) \quad (5)$$

- 5     (c) *weighting step*: calculate new importance weights as:

$$\tilde{w}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{i=1}^M w_{t-1}^{(i)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i)})}{\sum_{k=1}^K \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t^{(m)})} \quad (6)$$

6 **end**

---

The OAPF framework is described in Algorithm 1. The method starts by simulating  $M$  samples from the prior pdf, and then at each time  $t$ , it consists of the three following stages: (a) optimization, (b) sampling, and (c) weighting steps. Note that this structure keeps also some ties with adaptive IS (AIS) algorithms. In particular, the optimization step can be seen as an adaptive procedure of the mixture proposal with one iteration (see [Bugallo et al., 2017] for more details). First, the optimization step adapts the mixture proposal of Eq. (4). This procedure is discussed in detail in the next Section. Second, the new  $M$  particles are simulated from the mixture proposal. Third, the importance weights are calculated as in Eq. (6). It is worth remarking that the numerator does not evaluate the true filtering pdf but only an (unnormalized) approximation. However, the importance weights are still *proper* ([Liu, 2004]), as we show below.

#### 3.2 OAPF IMPORTANCE WEIGHTS

The importance weights play a crucial role both in the estimators of generic moments of the approximate distributions and also in the behavior of the PF for the next time step.

Hence, reducing the variance of the importance weights is the ultimate goal in PF. Since this variance depends on the discrepancy between the proposal and target pdfs [Ryu and Boyd, 2014] the benefit of considering a mixture proposal in Eq. 4 and for the importance weights in Eq. (6) is twofold. First, mixtures are a flexible way to approximate a large collection of pdfs. Second, while PFs work implicitly with mixture proposal, only few works use them in the denominator of the importance weights [Klaas et al., 2005, Elvira et al., 2018, 2019a]. Moreover, to the best of our knowledge these works did not extend the consistency results for the APF (Pitt 2012) to this importance weight. Placing the whole mixture in the denominator, as we do in OAPF, is known to reduce variance in MIS [Elvira et al., 2019b], even yielding zero-variance weights in the case of perfect matching between the mixture proposal and target pdfs.

In OAPF, the standard IS estimators can be built. More precisely, moments of the filtering pdf can be approximated by the self-normalized IS (SNIS) estimator as

$$\hat{I}(h_t) = \sum_{m=1}^M w_t^{(m)} h_t(\mathbf{x}_t^{(m)}), \quad (7)$$

where  $w_t^{(m)} = \frac{\tilde{w}_t^{(m)}}{\sum_{j=1}^M \tilde{w}_t^{(j)}}$  are the normalized weights. Finally, the weights of OAPF can be used to build an unbiased estimator of  $p(\mathbf{y}_{1:t})$ , which is crucial for many statistical tasks such as model selection [Luengo et al., 2020]. We build the OAPF estimator as:

$$\hat{p}(\mathbf{y}_{1:T}) = \hat{p}(\mathbf{y}_1) \prod_{t=2}^T \hat{p}(\mathbf{y}_t | \mathbf{y}_{1:t-1}), \quad (8)$$

where  $\hat{p}(\mathbf{y}_t | \mathbf{y}_{1:t-1}) = \frac{1}{M} \sum_{m=1}^M \tilde{w}_t^{(m)}$ . The functional form of the OAPF estimator is similar to other PFs and can be justified by standard IS arguments, but the computation of the importance weights  $\tilde{w}_t^{(m)}$  differs from other methods as discussed above. In the following, we prove that the estimator  $\hat{p}(\mathbf{y}_{1:T})$  is unbiased and consistent, which turns the SNIS estimator of Eq. (7) consistent.

**Theorem 1** *For any set of mixture proposals  $\{\psi_t(\mathbf{x}_t)\}_{t=1}^T$  fulfilling standard regularity conditions in IS, the normalizing constant estimator in Eq. (8) is unbiased and consistent, i.e.,  $\mathbb{E}[\hat{p}(\mathbf{y}_{1:T})] = p(\mathbf{y}_{1:T})$  and  $\lim_{M \rightarrow \infty} \hat{p}(\mathbf{y}_{1:T}) = p(\mathbf{y}_{1:T})$  a.s. for any  $T \in \mathbb{R}^+$ .*

*Proof:* The proof is presented in the supplementary material as well as a description of the regularity conditions.  $\square$

Note that the consistency of the SNIS estimator in Eq. (7) is also guaranteed by standard IS arguments (we complete this discussion in the supplement).

Finally, note that the minimization of the variance of the normalizing constant is equivalent to minimizing the variance of the importance weights  $\tilde{w}_t$  [Doucet and Johansen,

2009]. The OAPF explicitly aims at reducing this variance by minimizing the mismatch between the target pdf and the mixture proposal. In the supplement, we also present a proof showing that the variance the OAPF weights in Eq. (6) is always less than those of APF in Eq. (3), when  $K = M$  and the mixture weights are the same.

### 3.3 OPTIMIZATION OF THE MIXTURE WEIGHTS

In this section we discuss an approach to select the weights of the mixture proposal  $\psi_t$ . The ultimate goal is to select them so that the proposal is a good approximation of the approximate filtering posterior. To achieve this, we impose these two distributions to be pointwise close at a set of  $E$  evaluation points  $\{\mathbf{z}_t^{(e)}\}_{e=1}^E$ . We will show that this approach is flexible and brings several advantages. For simplicity, we start considering the case with  $K = E$ , where the evaluation points are the centers of the  $K$  kernels  $q^{(k)}$  in the proposal (4), i.e.,  $\{\mathbf{z}_t^{(e)}\}_{e=1}^E = \{\boldsymbol{\mu}_t^{(k)}\}_{k=1}^K$ . More precisely, the  $K$  kernels could be chosen as a subset  $K$  elements from the set of  $M$  transition kernels (from the previous  $M$  particles). Note that APF and improved APF [Elvira et al., 2019a] also use the center of the transition kernels. Alternatively, we could also use “optimal” SMC kernels [Doucet and Johansen, 2009] which are defined via the intractable function  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$ . Our framework allows for generic choices so these restrictions are not necessary. We continue this section in a generic setting, expanding the discussion on how many kernels and evaluation points to choose in Section 3.4.

Now that we have fixed the  $K$  mixture kernels and the  $E$  evaluation points, we can satisfy the condition previously mentioned and build a *linear* system of  $E$  equations as:

$$\sum_{k=1}^K \lambda_t^{(k)} q_t^{(k)}(\mathbf{z}_t^{(e)}) = g(\mathbf{y}_t | \mathbf{z}_t^{(e)}) \sum_{m=1}^M w_{t-1}^{(m)} f(\mathbf{z}_t^{(e)} | \mathbf{x}_{t-1}^{(m)}), \quad e = 1, \dots, E, \quad (9)$$

where the  $K$  mixture weights  $\lambda_t^{(k)}$  are unknown at each time  $t$ . For a unique solution to exist is necessary that  $K = E$ , but in general we do not need to restrict to this case. Below, we show how to turn this problem into a (constrained) *convex* optimization problem. Let us define the the vectors  $\boldsymbol{\lambda} = (\lambda_t^{(1)}, \dots, \lambda_t^{(K)})^\top$ ,  $\mathbf{w} = (w_{t-1}^{(1)}, \dots, w_{t-1}^{(M)})^\top$ ,  $\mathbf{f}^{(e)} = (f(\mathbf{z}_t^{(e)} | \mathbf{x}_{t-1}^{(1)}), \dots, f(\mathbf{z}_t^{(e)} | \mathbf{x}_{t-1}^{(M)}))^\top$ , and  $\mathbf{q}^{(e)} = (q_t^{(1)}(\mathbf{z}_t^{(e)}), \dots, q_t^{(K)}(\mathbf{z}_t^{(e)}))^\top$ . Then, we can re-write Eq. (9) as

$$\mathbf{q}^{(e)\top} \boldsymbol{\lambda} = \underbrace{g(\mathbf{y}_t | \mathbf{z}_t^{(e)}) \odot \mathbf{w}^\top \mathbf{f}^{(e)}}_{\tilde{\pi}^{(e)}}, \quad (10)$$



for  $e = 1, \dots, E$ , where  $\odot$  is elementwise multiplication and defining additionally the right-hand side to be  $\tilde{\pi}^{(e)}$ . More compactly, Eq. (10) can be re-expressed in matrix form as:

$$\begin{aligned} & \begin{bmatrix} \text{---} & \mathbf{q}^{(1)\top} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \mathbf{q}^{(E)\top} & \text{---} \end{bmatrix} \begin{bmatrix} \lambda \\ \vdots \\ \lambda \end{bmatrix} = \\ & \begin{bmatrix} \text{---} & g(\mathbf{y}_t | \mathbf{z}_t^{(1)}) \odot \mathbf{f}^{(1)\top} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & g(\mathbf{y}_t | \mathbf{z}_t^{(M)}) \odot \mathbf{f}^{(E)\top} & \text{---} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \vdots \\ \mathbf{w} \end{bmatrix}, \end{aligned} \quad (11)$$

defining  $\mathbf{Q}$  as the  $E \times K$  matrix on the left-hand side of (11) and  $\tilde{\pi}$  as the resulting  $E \times 1$  vector on the right-hand side. We now define a generic constrained optimization problem as for the mixture weights as:

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{Q}\boldsymbol{\lambda}, \tilde{\pi}), \quad (12)$$

where  $\mathcal{L}(\cdot)$  a generic loss function. The optimization will be constrained since  $\boldsymbol{\lambda}$  will be used for resampling, and therefore needs to have non-negative elements.<sup>3</sup> In the next Section, we present a possible strategy to implement  $\mathcal{L}(\cdot)$  and solve the optimization problem.

### Optimization via Non-Negative Least Squares (NNLS)

The previous problem can be encoded as a non-negative least squares problem by taking the squared distance of the pdfs at the  $E$  evaluation points  $\{\mathbf{z}_t^{(e)}\}_{e=1}^E$ . Taking squared differences between left-hand side and right-hand side of (11) leads to:

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \|\mathbf{Q}\boldsymbol{\lambda} - \tilde{\pi}\|_2^2 \quad \text{subject to : } \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^K.$$

This problem 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. Theoretical results on NNLS have shown that, especially when the dimension of  $\mathbf{Q}$  is large (large  $K$  and  $E$  in our case), the solutions tend to be *very sparse* [Slawski et al., 2013, Meinshausen et al., 2013]. The optimization problem can be solved by the widely used algorithm in [Lawson and Hanson, 1995], as well as by concurrent work on exact sparse NNLS [Nadisic et al., 2020] and even strong *GPU accelerations* could be exploited [Luo and Duraiswami, 2011, Kysenko et al., 2012]. There are other possible choices to implement Eq. (12). For instance, it is possible to formulate a linear program and solve it with the *Simplex* algorithm. We tried this approach, but found that the stability of the algorithm may be endangered above 3 to 5 dimensions.

<sup>3</sup>The resulting values can be normalized afterwards so they parametrize the mixture proposal in Eq. (4).

### 3.4 SELECTION OF KERNELS, EVALUATION POINTS, AND COMPUTATIONAL COMPLEXITY

The generic OAPF framework also allows for the choice of the number and type of kernels. In our experiments, we choose the transition kernel for simplicity (as it is done in APF or BPF), but other choices are equally valid. Regarding the number of kernels  $K$ , the novel MIS perspective allows for an extra degree of freedom unlike in standard filters. In particular, we have found that it is in general possible to reduce  $K$  dramatically w.r.t.  $M$ , without a loss of performance; however, the ESS as a degeneracy measure seems more sensitive to the decrease in  $K$ . Further work could develop a formal analysis to explain this behavior. Moreover, we are also able to reduce the number of target evaluations  $E$  w.r.t.  $M$  at the optimization step (see more details in Section 3.3). Unlike in the APF where the number of pre-weights is necessarily  $K = M$ , in OAPF this is a choice. The reason is that the purpose of the evaluations is to evaluate the target pdf so the mixture proposal can place probability mass in relevant parts of the space. Therefore, the number of evaluation points ( $E$ ) can be much smaller than the number of particles (more details are provided in the Supplement).

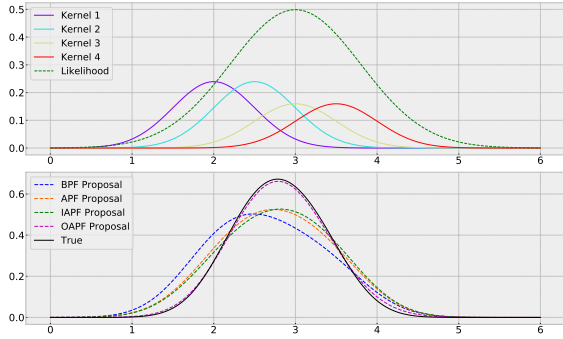
As a general guideline, our starting point is setting  $K = E$  and choosing the evaluation points (deterministically) as the center of each  $q_t^{(k)}$ , retaining those associated with the  $E$  *greatest values of the approximate filtering pdf* in the RHS of Eq. (9): this simple scheme worked well in our experiments. Ultimately, we remark how more evaluation points can only improve the quality of the approximation to the filtering posterior. More strategies in evaluation points selection (number, deterministic selection vs sampling) could be explored in further work.

The computational complexity of OAPF can be decomposed in weighting and optimization steps. In the former, the numerator of Eq. (6) has been shown to have very good approximations in Marginal PFs [Klaas et al., 2005] in time  $\mathcal{O}(M \log M)$  using *dual-tree* methods for weighted kernel density estimation problems. Further, due to the sparsity properties in the recovered  $\boldsymbol{\lambda}$  (see Section 3.3), our method improves the effective runtime in the calculation of the importance weights.

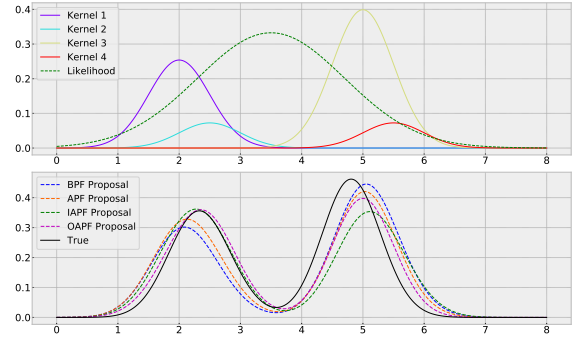
The optimization step can be accelerated significantly by implementing standard methods of the rich literature on fast and accurate approximations to constrained least-squares problems [Pilanci and Wainwright, 2016], or via direct application of Frank-Wolfe algorithms [Jaggi, 2013].

## 4 RELATED WORK

The OAPF follows a different approach w.r.t. most papers in the PF literature by interpreting the  $M$  samples to be sim-



(a) In this first example we choose a unimodal posterior. OAPF substantially outperforms the other algorithms.



(b) OAPF is the only algorithm who can match well *both* modes simultaneously with this multimodal posterior.

Figure 1: **Experiment 1 (Toy Example.)** In this experiment we show that OAPF proposals are closer to true posteriors compared to its competitors. We calculated  $\chi^2$ -divergence for these examples in Table 1. Note that here OAPF uses transition kernels for the proposal and their centers as evaluation points. We provide all parameters for reproducibility in the supplement.

ulated from a mixture proposal with  $K$  components. Moreover, unlike other popular PFs, we allow for a reduction of the number of the components, exploiting the sparsity behavior of the optimization algorithm. This perspective is connected to the auxiliary marginal PF (AMPF) [Klaas et al., 2005] and improved APF (IAPF) [Elvira et al., 2018] algorithms, and is supported by recent advances in MIS [Elvira et al., 2019b] (see also the discussion of the variance reduction in [Klaas et al., 2005]), and it also links with the re-interpretation of BPF and APF [Elvira et al., 2019a]. The selection of the mixture weights has connections with other works. For instance, a flexible framework named *twisted* APFs is developed in [Guarniero et al., 2017], where APFs are interpreted as a special case of changing the distribution targeted in IS (this interpretation appeared first in [Johansen and Doucet, 2008, Doucet and Johansen, 2009]). In this method, the computation is done in an offline fashion (see an extension of this line in [Heng et al., 2020]). Resampling weights were also found in [Reich, 2013] via a convex optimization problem derived via optimal transport arguments, which is however more computationally expensive than ours, and scales worse to higher dimensions. Further, our selection can be connected to *black-box* importance sampling [Liu and Lee, 2017], which computes IS weights<sup>4</sup> in a static setting with convex optimization. In [Akyildiz and Míguez, 2020], they develop a PF framework with a different approach, preemptively moving a subset of particles to a region of high likelihood with gradient methods. In [Cornebise et al., 2014], the method adapts a mixture of kernels in a more generic setup (sequential Monte Carlo samplers), focusing in the choice of kernels. Further, the approach in [Kronander and Schön, 2014] propagates particles associated with kernels that are placed high likelihood regions, which could be combined in our framework.

<sup>4</sup>rather than simulation weights, which are specific to APF

## 5 EXPERIMENTS

We compare OAPF with BPF, APF as well as the recent improved APF (IAPF) [Elvira et al., 2018], which also uses a mixture in the denominator of the importance weights and can be seen as a special case in our framework. The IAPF strictly improves over APF and BPF in most settings [Elvira et al., 2018]. Note that the simple BPF can sometimes perform unexpectedly well, as it is well known in the PF community. In the linear Gaussian model, we additionally compare with the fully adapted APF (FA-APF). We evaluate our framework in 4 sets of experiments.

Our aim is to show the benefits of OAPF in terms of variance of importance weights, which is crucial in particle filters: the weights are used not only for approximating integrals of interest but also for building better particle approximations in the next time steps. Therefore, we choose metrics that are directly connected to the variance of the importance weights:  $\chi^2$ -divergence between mixture proposal and filtering pdfs, error in the estimation of the posterior mean and marginal likelihood, and effective sample size (ESS) [Särkkä, 2013]. The setup of the experiments is as follows:

- **Experiment 1: Toy example.** We show visually that the mixture proposal in OAPF reconstructs the posterior better than its competitors, both with unimodal and multimodal posteriors. Numerically we show an improved  $\chi^2$ -divergence between proposal and filtering pdfs, which directly translates into lower variance of importance weights.
- **Experiment 2: Linear dynamical model.** We exploit the closed-form solution of the linear dynamical model, perhaps the most known SSM and widely used for instance in object tracking [Särkkä, 2013]. This allows comparison with sampling from the Kalman Filter, as well as a closed form FA-APF. We show that OAPF

reaches better solutions with highly reduced runtime w.r.t IAPF thanks to our selection of  $K$  and  $E$ .

- **Experiment 3: Stochastic Lorenz 63 model.** Transitioning to more challenging non-linear non-Gaussian models, we show an improved performance on discretized version of this popular chaotic dynamical system, which is used for instance in atmospheric models for weather forecasting [Ott et al., 2004, Yeong et al., 2020]. We compare the PFs in terms of the ESS, which is widely used as a proxy for the weight variance.
- **Experiment 4: Stochastic volatility model.** Finally, we perform inference for a multivariate stochastic volatility model used in related work on APFs [Guarniero et al., 2017]. Here, as in Experiment 4, we look at ESS and show improved performance against all other algorithms.

We consider time-series with  $T = 100$  time steps, except otherwise stated. We let  $d_x$  be the dimension of the hidden state, i.e.,  $\mathbf{x}_t \in \mathbb{R}^{d_x}$ . Due to the curse of dimensionality, a general reduction in performance for all methods is expected as  $d_x$  grows. For linear dynamical models, we show improved estimates with significantly reduced runtime than all other algorithms, including IAPF. For the more challenging models where ground truth is not available, we achieve better ESS than the competitors. Note that in those experiments, we set  $K = E = M$ . However, due to the high sparsity of solutions in OAPF, our effective  $K$  is much lower. For  $M = 1000$  particles we report an average of 88% sparsity, while for  $M = 100$  an average of 65%. All averages and standard errors are obtained with 100 independent Monte Carlo runs.<sup>5</sup>

Table 1: **Experiment 1 (Toy example).**  $\chi^2$ -div. between filtering and mixture proposal pdfs in Figure 1.

Method	$\chi^2$ -div. (Fig. 1a)	$\chi^2$ -div. (Fig. 1b)
BPF	0.1662	0.2245
APF	0.0916	0.1633
IAPF	0.0870	0.2402
OAPF	<b>0.0069</b>	<b>0.0819</b>

**1. Toy Example.** The main goal of this toy example is to illustrate that the OAPF mixture proposal better reconstructs the filtering pdf. We also measure the  $\chi^2$ -divergence between both pdfs. We consider a single iteration of each PF algorithm and build artificial proposals by multiplying a mixture of 4 Gaussians with a Gaussian likelihood. Results from the two experiments with the above setting are shown

<sup>5</sup>The code used in the experiments can be found at [https://github.com/nicola144/optimized\\_auxiliary\\_particle\\_filters](https://github.com/nicola144/optimized_auxiliary_particle_filters)

in Figure 1. We select the means of the transition kernel  $f(\cdot)$  as evaluation points, and set  $K = E = M = 4$ . In Figure 1a, we show the results for a unimodal posterior. This setting is advantageous for the IAPF, as transition kernels significantly overlap (see [Elvira et al., 2019a] for more details). The likelihood is sufficiently informative, which explains why APF outperforms BPF. Figure 1b shows a more complex multimodal posterior with a more diffused likelihood. Interestingly, we find that IAPF can perform even worse than APF, while our OAPF does not suffer from this issue. Table 1 quantifies (for both settings) the mismatch between mixture proposal and filtering pdfs in terms of  $\chi^2$ -divergence, confirming the visual analysis of Fig. 1a.

**2. Linear Dynamical Model.** The linear dynamical model is arguably the most popular SSM, routinely being the first choice to assess PFs. It has been applied in a wide range of applications (e.g., robotics [Särkkä, 2013]). This model is particularly useful for validating PFs since it is one of the few models admitting closed-form solutions of the filtering distribution and the normalizing constant (via the celebrated *Kalman filter*). The defining transition and observation equations are standard (see e.g., [Särkkä, 2013]) and require mainly the selection of observation and transition covariances (more details in the supplement). We tested the algorithms for different choices of model parameters, e.g., for high-variance observation noise or high-variance transition noise. In Figure 2a, we calculate the normalized mean-squared error (NMSE)<sup>6</sup> between the estimate of the posterior mean and the true value for  $M \in \{10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000\}$  with  $d_x = 10$ . Here, observation and transition covariances were set to  $5\mathbf{I}$  and  $2.5\mathbf{I}$  respectively; complementary results with other parameters, dimensions, as well as additional results on estimation of the normalizing constant are available in the supplement. The proposed OAPF outperforms all the competitors with a low  $K = E = 5$  in all settings, which is translated into large computational savings. Similar conclusions can be extracted for other choices of the model parameters.

**3. Stochastic Lorenz 63 Model.** The Lorenz 63 is a *chaotic* system, since slightly different initial conditions generate extremely different trajectories. Due to this difficulty, this model is often used to evaluate PFs [Akyildiz and Míguez, 2020]. We consider a discretized version of the state dynamics using an Euler-Maryuyama scheme and observations with additive noise. The hidden state is three dimensional  $\mathbf{x} = [x^{[1]}, x^{[2]}, x^{[3]}]$  and the transition dynamics

<sup>6</sup>We define NSME as mean-squared error divided by true value.

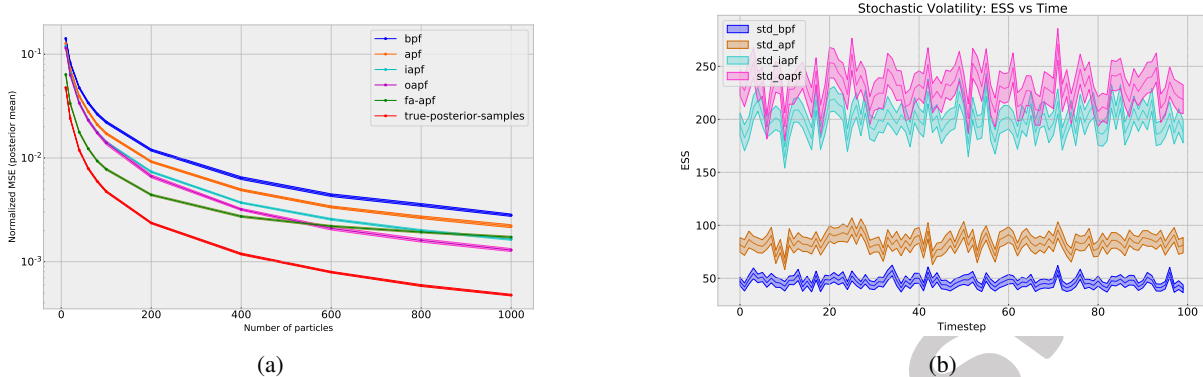


Figure 2: (a) **Experiment 2 (Linear dynamical model)**. NMSE to the true posterior mean as function of the number of particles, with  $d_x = 10$ . Note that OAPF runs with  $K = E = 5$  in all cases, selected by the strategy described in Section 3.4. The FA-APF is only available analytically in this model, and the line true posterior samples (sampled from the Kalman filter) is presented as benchmark. (b) **Experiment 4 (Stochastic volatility)**. ESS as a function of time with  $d_x = 10$ ,  $M = 1000$ .

are defined by the differential equations:

$$dx^{[1]} = \sigma(x^{[2]} - x^{[1]})d\tau + dw_{x^{[1]}} \quad (13)$$

$$dx^{[2]} = (\rho x^{[1]} - x^{[3]}x^{[1]} - x^{[2]})d\tau + dw_{x^{[2]}} \quad (14)$$

$$dx^{[3]} = (x^{[1]}x^{[2]} - \beta x^{[3]})d\tau + dw_{x^{[3]}} \quad (15)$$

where  $\tau$  denotes continuous time,  $w_{x^{[1]}}$ ,  $w_{x^{[2]}}$ ,  $w_{x^{[3]}}$  are independent one-dimensional standard Wiener processes and  $(\sigma, \rho, \beta)$  are parameters of the model. We use the an increment  $\Delta t$  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 the standard values for  $(\sigma, \rho, \beta)$  (see supplement). The results of averaged ESS with two different values of  $\Delta t \in \{0.01, 0.008\}$  are shown in Table 2. Note that even these small changes in  $\Delta t$  cause very different trajectories, as we also show in the supplement.

**4. Stochastic Volatility Model.** We perform inference in a multivariate stochastic volatility model (SVM), a type of stochastic process where the variance is a latent variable that follows itself a stochastic process. These are extremely useful models to apply for many tasks in econometrics, e.g., for predicting the volatility of a heteroskedastic sequence such as returns on equity indices or currency exchanges. SVMs are often used to evaluate particle filters [Pitt and Shephard, 1999, Klaas et al., 2005, Guarniero et al., 2017]. We employ the version in [Chib et al., 2009], which is also used in related work on APFs [Guarniero et al., 2017]. It is defined by the following pdfs:

$$p(\mathbf{x}_0) = \mathcal{N}_{\mathbf{x}_0}(\mathbf{m}, \mathbf{U}_0), \quad (16)$$

$$f(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}_{\mathbf{x}_t}(\mathbf{m} + \text{diag}(\phi)(\mathbf{x}_{t-1} - \mathbf{m}), \mathbf{U}), \quad (17)$$

$$g(y_t|\mathbf{x}_t) = \mathcal{N}_{y_t}(\mathbf{0}, \exp(\text{diag}(\mathbf{x}_t))). \quad (18)$$

Table 3 shows averaged ESS for  $d_x = (2, 5, 10)$ . The parameters for this experiment are set to  $\mathbf{m} = \mathbf{0}$ ,  $\mathbf{U}_0 = \mathbf{I}$ ,  $\mathbf{U} =$

Table 2: **Experiment 3 (Lorenz)**.  $T = 1000$  timesteps,  $M = 100$  particles. Averaged ESS and standard errors.

Method	$\Delta t = 0.01$	$\Delta t = 0.008$
BPF	$57.7 \pm 0.2$	$58.1 \pm 0.2$
APF	$55.1 \pm 0.2$	$55.2 \pm 0.2$
IAPF	$70.1 \pm 0.1$	$71.0 \pm 0.1$
OAPF	<b><math>76.7 \pm 0.1</math></b>	<b><math>76.4 \pm 0.1</math></b>

Table 3: **Experiment 4 (Stochastic Volatility)**. Note that when  $d_x = 10$  then  $M = 1000$ , otherwise  $M = 100$ . Averaged ESS and standard errors.

Method	$d_x = 2$	$d_x = 5$	$d_x = 10$
BPF	$63.5 \pm 0.2$	$33.5 \pm 0.2$	$108.7 \pm 0.8$
APF	$63.5 \pm 0.2$	$34.5 \pm 0.2$	$107.2 \pm 1.0$
IAPF	$73.0 \pm 0.1$	$44.9 \pm 0.2$	$203.5 \pm 0.9$
OAPF	<b><math>88.3 \pm 0.2</math></b>	<b><math>63.5 \pm 0.2</math></b>	<b><math>366.2 \pm 1.8</math></b>

$\mathbf{I}$ ,  $\phi = \mathbf{1}$ . Figure 2b shows the averaged ESS over time for  $d_x = 10$  and same parameters, except  $\phi = \frac{1}{2}\mathbf{1}$ . For additional results, see supplementary.

## 6 CONCLUSIONS

In this paper we have proposed the OAPF, a flexible framework for particle filtering that uses a generic mixture distribution as a proposal and includes it in the importance weighting scheme. The framework allows for the development of particle filters with improved performance, and we



provide an explicit implementation. We have proved the unbiasedness of the OAPF marginal likelihood estimator for any mixture proposal that fulfills standard IS requirements. We also show the effectiveness of OAPF in reducing the variance of the IS estimators. In OAPF, we directly optimize the mixture proposal to the posterior in an online fashion, rather than making specific analytic choices of mixture weights like in AMPF or IAPF. Conversely to most other methods that optimize a proposal (e.g., variational inference), our optimization strategy is convex, directly addressing the ultimate goal of minimizing the variance of the importance weights. Therefore, OAPF can deal with any likelihood and transition models (that admit a density) without resorting to black-box, non-convex methods (see for instance [Archer et al., 2015, Dieng et al., 2017]). We have shown improved performance of the proposed implementation of the OAPF across a series of challenging state-space models and metrics, comparing with BPF, APF, and the competitive IAPF. Finally, the flexibility and the strong theoretical guarantees of OAPF pave the way for new methodological advances within this framework.

## Acknowledgements

We would like to thank Theodoros Damoulas for initial comments on an earlier version of the paper, and for the support during the refinement of this work.

## References

- Ömer Deniz Akyildiz and Joaquín Míguez. Nudging the particle filter. *Statistics and Computing*, 30(2):305–330, 2020.
- Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.
- Monica F Bugallo, Víctor Elvira, Luca Martino, David Luengo, Joaquín Míguez, and Petar M Djurić. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.
- Siddhartha Chib, Yasuhiro Omori, and Manabu Asai. Multivariate stochastic volatility. In *Handbook of Financial Time Series*, pages 365–400. Springer, 2009.
- Nicolas Chopin and Omiros Papaspiliopoulos. *An introduction to sequential Monte Carlo*. Springer, 2020.
- Julien Cornebise, Eric Moulines, and Jimmy Olsson. Adaptive sequential monte carlo by means of mixture of experts. *Statistics and Computing*, 24(3):317–337, 2014.
- Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via  $\chi$  upper bound minimization. In *Advances in Neural Information Processing Systems*, pages 2732–2741, 2017.
- Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.
- Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.
- Víctor Elvira, Luca Martino, Mónica F Bugallo, and Petar M Djurić. In search for improved auxiliary particle filters. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1637–1641. IEEE, 2018.
- Victor Elvira, Luca Martino, Monica F Bugallo, and Petar M Djuric. Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]. *IEEE Signal Processing Magazine*, 36(6):145–152, 2019a.
- Víctor Elvira, Luca Martino, David Luengo, Mónica F Bugallo, et al. Generalized multiple importance sampling. *Statistical Science*, 34(1):129–155, 2019b.
- Paul Fearnhead. *Sequential Monte Carlo methods in filter theory*. PhD thesis, University of Oxford, 1998.
- Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.
- Shixiang Shane Gu, Zoubin Ghahramani, and Richard E Turner. Neural adaptive sequential monte carlo. In *Advances in neural information processing systems*, pages 2629–2637, 2015.
- Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.
- Jeremy Heng, Adrian N Bishop, George Deligiannidis, Arnaud Doucet, et al. Controlled sequential monte carlo. *Annals of Statistics*, 48(5):2904–2929, 2020.
- Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013.
- Adam M Johansen and Arnaud Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504, 2008.

- Mike Klaas, Nando de Freitas, and Arnaud Doucet. Toward practical  $n^2$  monte carlo: the marginal particle filter. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 308–315, Arlington, Virginia, 2005. AUAI Press.
- Joel Kronander and Thomas B Schön. Robust auxiliary particle filters using multiple importance sampling. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pages 268–271. IEEE, 2014.
- V. Kysenko, K. Rupp, O. Marchenko, S. Selberherr, and A. Anisimov. Gpu-accelerated non-negative matrix factorization for text mining. In *International Conference on Application of Natural Language to Information Systems*, pages 158–163. Springer, 2012.
- Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.
- Dieterich Lawson, George Tucker, Christian A Naesseth, Chris J Maddison, Ryan P Adams, and Yee Whye Teh. Twisted variational sequential monte carlo. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.
- Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.
- Qiang Liu and Jason Lee. Black-box importance sampling. In *Artificial Intelligence and Statistics*, pages 952–961. PMLR, 2017.
- David Luengo, Luca Martino, Mónica Bugallo, Víctor Elvira, and Simo Särkkä. A survey of monte carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, 2020(1):1–62, 2020.
- Yuancheng Luo and Ramani Duraiswami. Efficient parallel nonnegative least squares on multicore architectures. *SIAM Journal on Scientific Computing*, 33(5):2848–2863, 2011.
- Xiao Ma, Péter Karkus, David Hsu, and Wee Sun Lee. Particle filter recurrent neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 5101–5108. AAAI Press, 2020.
- Chris J. Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Arnaud Doucet, Andriy Mnih, and Yee Whye Teh. Particle value functions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.
- Nicolai Meinshausen et al. Sign-constrained least squares estimation for high-dimensional regression. *Electronic Journal of Statistics*, 7:1607–1631, 2013.
- Nicolas Nadisic, Arnaud Vandaele, Nicolas Gillis, and Jeremy E Cohen. Exact sparse nonnegative least squares. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5395–5399. IEEE, 2020.
- Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. volume 84 of *Proceedings of Machine Learning Research*, pages 968–977, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.
- Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003.
- Edward Ott, Brian R Hunt, Istvan Szunyogh, Aleksey V Zimin, Eric J Kostelich, Matteo Corazza, Eugenia Kalnay, DJ Patil, and James A Yorke. A local ensemble kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):415–428, 2004.
- Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2019.
- Mert Pilanci and Martin J Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *The Journal of Machine Learning Research*, 17(1):1842–1879, 2016.
- Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.
- Michael K Pitt, Ralph dos Santos Silva, Paolo Giordani, and Robert Kohn. On some properties of markov chain monte carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151, 2012.
- Sebastian Reich. A nonparametric ensemble transform method for bayesian inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, 2013.
- Ernest K Ryu and Stephen P Boyd. Adaptive importance sampling via stochastic convex programming. *arXiv preprint arXiv:1412.4845*, 2014.
- Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.
- Martin Slawski, Matthias Hein, et al. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

Sebastian Thrun. Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 511–518. Morgan Kaufmann Publishers Inc., 2002.

E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

Nikos Vlassis, Bas Terwijn, and Ben Krose. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 7–12. IEEE, 2002.

Yunbo Wang, Bo Liu, Jiajun Wu, Yuke Zhu, Simon S Du, Li Fei-Fei, and Joshua B Tenenbaum. Dualsmc: Tunneling differentiable filtering and planning under continuous pomdps. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2020.

Ananta Adhi Wardhana, Evan Clearesta, Augie Widyotriatmo, et al. Mobile robot localization using modified particle filter. In *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*, pages 161–164. IEEE, 2013.

Nick Whiteley and Adam M Johansen. Auxiliary particle filtering: recent developments. *Bayesian time series models*. Cambridge University Press, Cambridge, 2011.

Hoong C Yeong, Ryne T Beeson, N Sri Namachchivaya, and Nicolas Perkowski. Particle filters with nudging in multi-scale chaotic systems: With application to the lorenz’96 atmospheric model. *Journal of Nonlinear Science*, pages 1–34, 2020.