Compressed Monte Carlo with application in particle filtering

Compressed Monte Carlo with application in particle filtering

Luca Martino[⊤], Víctor Elvira[*]

[⊤] Dep. of Signal Processing, Universidad Rey Juan Carlos (URJC) and Universidad Carlos III de Madrid (UC3M)

[*] School of Mathematics, University of Edinburgh (UK)

***Abstract*—Bayesian models have become very popular over the last years in several fields such as signal processing, statistics, and machine learning. Bayesian inference requires the approximation of complicated integrals involving posterior distributions. For this purpose, Monte Carlo (MC) methods, such as Markov Chain Monte Carlo and importance sampling algorithms, are often employed. In this work, we introduce the theory and practice of a Compressed MC (C-MC) scheme to compress the statistical information contained in a set of random samples. In its basic version, C-MC is strictly related to the stratification technique, a well-known method used for variance reduction purposes. Deterministic C-MC schemes are also presented, which provide very good performance. The compression problem is strictly related to the moment matching approach applied in different filtering techniques, usually called as Gaussian quadrature rules or sigma-point methods. C-MC can be employed in a distributed Bayesian inference framework when cheap and fast communications with a central processor are required. Furthermore, C-MC is useful within particle filtering and adaptive IS algorithms, as shown by three novel schemes introduced in this work. Six numerical results confirm the benefits of the introduced schemes, outperforming the corresponding benchmark methods. A related code is also provided.**[1]

***Index Terms*—Bayesian inference, MCMC, importance sampling, particle filtering, Gaussian quadrature, sigma points, herding Algorithms, distributed algorithms**

I. Introduction

An essential problem in signal processing, statistics, and machine learning is the estimation of unknown parameters in probabilistic models from noisy observations. Within the Bayesian inference framework, these problems are addressed by constructing posterior probability density functions (pdfs) of the unknowns [4], [45]. Unfortunately, the computation of statistical quantities related to these posterior distributions (such as moments or credible intervals) is analytically impossible in most real-world applications. As a consequence, the design of efficient computational algorithms is of utmost interest. Monte Carlo (MC) techniques come to the rescue for solving the most difficult problems of inference [27], [44]. They are benchmark tools for approximating complicated integrals involving sophisticated multidimensional target densities, based on drawing of random samples [44], [34]. Markov Chain Monte Carlo (MCMC) algorithms, Importance Sampling (IS) schemes, and its sequential version (particle filtering) are the most important classes of MC methods [45].

E-mail: luca.martino@urjc.es.

[1]The code is provided at http://www.lucamartino.altervista.org/CMC_CODE_pub_EX1.zip

Determinism and support points. In order to reduce the computational demand of the Monte Carlo methods and the variance of the corresponding estimators, deterministic procedures have been included within the sampling algorithms. In the so-called variance reduction techniques (e.g., conditioning, stratification, antithetic sampling, and control variates), negative correlation is induced among the generated samples, hence obtaining more efficient estimators [41], [49]. In Quasi-Monte Carlo (QMC) methods, deterministic sequences of samples are employed, based on the concept of *low-discrepancy*, avoiding all kinds of randomness [15], [16], [39]. In the same line, deterministic approximations of the posterior distribution based on quadrature, cubature rules, or unscented transformations are often applied, when are available [1], [21], [50], [45]. These techniques provide a set of particles deterministically chosen (often called *sigma points*), to match perfectly the estimation of a pre-established number of moments of the posterior density. Most of them are derived for integrals that involve a Gaussian distribution [45]. These techniques are usually used in filtering applications as an extension of the standard Kalman filtering and as an alternative to the particle filtering techniques based on MC sampling. The quadrature rules are very efficient since with N weighted particles summarized exactly the first $2N$ non-central moments. However, quadrature approximations are available only for certain target densities. Indeed, the true values of the moments must be known and a solution of a highly non-linear system must be provided. This is possible only for specific target densities. More generally, the idea of sigma points is strictly connected to the need of *summarizing* a given distribution (and/or function) with a set of *representative, support points*, deterministically selected [30], [29]. This is an important topic is in computational statistics and has gained increasing attention in the last years: some relevant examples are the herding algorithms [9], [10], [24], [18], the studies about the representative points previously mentioned [29], [30], as well as space-filling and experimental designs [42]. Some of them have been applied jointly with MC schemes or used for numerical integration problems [24], [18].

Contribution. In this work, we introduce different schemes for compressing the information contained in N Monte Carlo samples into $M < N$ weighted particles. They are based on the so-called stratification approach [41], [44]. In the Compressed Monte Carlo (C-MC) schemes, we replace

the particle MC approximation obtained by N unweighted samples (e.g., generated by an MCMC algorithm) or weighted samples (e.g., generated by an IS algorithm), with another particle approximation with $M < N$ summary weighted samples. We desire to reduce the loss of information in terms of moment matching, in the same fashion of the quadrature rules. In this sense, the M summary particles can be considered as approximate sigma points. Furthermore, for a specific choice of the partition (specifically, see the case of unweighted C-MC samples in Section IV-C), an approximate low-discrepancy sequence is obtained, i.e., a QMC sequence is generated. Several alternatives and extensions are presented, including the random or deterministic selection of the summary particles.

The C-MC approach has a direct application in a parallel or distributed Bayesian framework with a centralized node, as discussed in Section V-A and graphically represented in Figure 2 . In this scenario, different local low-power nodes must transmit to a central node the results of their local Bayesian analysis, to provide a common complete inference [38], [3], [43]. The transmission should have the minimum possible cost and contain the maximum amount of information. Hence, the information must be properly compressed before being transmitted (see Section V for further details). C-MC can be considered an improvement of the bootstrap strategy, applied in different works regarding parallel sequential Monte Carlo schemes, where several resampled particles are transmitted jointly with a proper aggregated weight [3], [43], [48], [31]. However, the range of application of C-MC is not only restricted to the distributed scenario. We introduce two novel particle filtering schemes based on the C-MC approach. The first scheme enhances the well-known Gaussian particle filter (GPF) [23]. This proposed algorithm contains the GPF as a special case (with $M = 1$) and the regularized particle filter (with $M = N$) [12]. The second proposed scheme, called *compressed particle filter* (C-PF), requires the evaluation of the measurement model only M times instead of N. Therefore, the C-PF is faster than a standard particle filter and is particularly convenient when the likelihood evaluation is costly. We also provide an example of C-MC in modern adaptive IS schemes to allow the use of expensive mixtures as the denominator of the importance weights [33], [46]. More details are provided in Section V. Finally, note that similar and related ideas have been presented in different works and several applications, such as diffusion estimation [8], [40], smoothing techniques [13], and as alternative resampling procedures in particle filtering [25], [26]. The benefits of the proposed schemes are shown in six different numerical experiments.

Structure of the work. Section II introduces the basic setup of the Bayesian inference problem and describes the goal of the paper jointly with some possible solutions already presented in the literature. In Section III, we introduce the C-MC method whereas, in Section IV, we provide further analyses. In Section V, we describe different applications of C-MC, several novel algorithms, and further extensions. Section VI provides six numerical experiments, and some conclusions are contained in Section VII. The main acronyms of the work are summarized in Table I.

Table I
MAIN ACRONYMS OF THE WORK.

pdf	probability density function
MC	Monte Carlo
QMC	Quasi-Monte Carlo
MCMC	Markov Chain Monte Carlo
IS	Importance Sampling
C-MC	Compressed Monte Carlo
C-PF	Compressed Particle Filter
MSE	Mean Square Error

II. BACKGROUND

A. Problem statement

In many real-world applications, the interest lies in obtaining information about the posterior density of a set of unknown parameters given the observed data. Mathematically, denoting the vector of unknowns as $\mathbf{x} = [x_1, ..., x_{d_x}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$ and the observed data as $\mathbf{y} \in \mathbb{R}^{d_Y}$, the pdf is defined as

$$\bar{\pi}(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})} \propto \pi(\mathbf{x}|\mathbf{y}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}), \qquad (1)$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the normalization factor, that is usually called marginal likelihood or Bayesian model evidence. From now on, we remove the dependence on $\mathbf{y}$ to simplify the notation. A particular integral involving the random variable $\mathbf{X} \sim \bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ is then given by

$$I(h) \triangleq E_{\bar{\pi}}[h(\mathbf{X})] = \int_{\mathcal{D}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{D}} h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \qquad (2)$$

where $h(\mathbf{x})$ can be any integrable function of $\mathbf{x}$.[2] For simplicity, we assume that the functions $h(\mathbf{x})$ and $\bar{\pi}(\mathbf{x})$ are continuous in $\mathcal{D}$, and the integrand function, $h(\mathbf{x})\bar{\pi}(\mathbf{x})$, in Eq. (2) is integrable. More generally, we are interested in finding a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ of the measure of $\bar{\pi}(\mathbf{x})$ [27]. In many practical scenarios, we cannot obtain an analytical solution for the integral in Eq. (2). One possible alternative is to use different deterministic quadrature rules or formulas based on sigma points for approximating the integral $I(h)$ [1], [21], [45]. However, these deterministic techniques are available only in specific scenarios, i.e., for some particular pdfs $\bar{\pi}(\mathbf{x})$. Hence, Monte Carlo schemes are often preferred and applied to estimate I and provide a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$.

B. Monte Carlo (MC) sampling techniques

If it is possible to draw N independent samples, $\{\mathbf{x}_n\}_{n=1}^N$, directly from $\bar{\pi}(\mathbf{x})$, then we can construct a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^N \delta(\mathbf{x} - \mathbf{x}_n)$ of the measure of $\bar{\pi}$

[2]To simplify the notation, we have assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$ and the integral $I(h) \in \mathbb{R}$ is a scalar value. However, a more proper assumption is $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^\nu$ and $\mathbf{I}(\mathbf{h}) \in \mathbb{R}^\nu$ where $\nu \geq 1$. All the techniques and results in this work are valid for the more general mapping with $\nu \geq 1$, but we keep the simpler notation for $\nu = 1$. With $\nu > 1$, we would have a vector of integrals $\mathbf{I}(\mathbf{h})$. For instance, if $\mathbf{h}(\mathbf{x}) = \mathbf{x}$ we have $\nu = d_X$, and we have one integral for each component of $\mathbf{x}$.

[44]. This is the foundation of MC methods, denote as standard or direct MC. Therefore, replacing $\bar{\pi}(\mathbf{x})$ with $\widehat{\pi}^{(N)}(\mathbf{x})$ in Eq. (2), we obtain the standard Monte Carlo estimator of I,

$$\widehat{I}^{(N)}(h) = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{x}_n). \tag{3}$$

However, when sampling from $\bar{\pi}(\mathbf{x})$ is not possible, alternative MC methods are used [27], [44]. For instance, the MCMC algorithms generate correlated samples $\{\mathbf{x}_n\}_{n=1}^N$ that, after a burn-in period, are distributed according to $\bar{\pi}(\mathbf{x})$. Another possible approach is based on the importance sampling (IS) technique [44], [4]. In the following, we describe the basic ideas behind the IS schemes. Consider N samples $\{\mathbf{x}_n\}_{n=1}^N$ drawn from a proposal pdf, $q(\mathbf{x})$, with heavier tails than the target, $\bar{\pi}(\mathbf{x})$. We assign a weight to each sample and then we can be normalized them as follows,

$$w_i = \frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i)}, \qquad \bar{w}_i = \frac{w_i}{\sum_{j=1}^N w_j}, \tag{4}$$

with $i = 1, ..., N$. Therefore, the moment of interest can be approximated as

$$\widehat{I}^{(N)}(h) = \frac{1}{N\widehat{Z}}\sum_{i=1}^{N} w_i h(\mathbf{x}_i) \tag{5}$$

$$= \sum_{i=1}^{N} \bar{w}_i h(\mathbf{x}_i), \tag{6}$$

where $\widehat{Z} = \frac{1}{N}\sum_{j=1}^N w_j$ is a unbiased estimator of $Z = \int_{\mathcal{D}} \pi(\mathbf{x})d\mathbf{x}$ [44]. One can consider that, in the standard Monte Carlo and MCMC methods, the normalized weights are $\bar{w}_i = 1/N$. Then, all the described Monte Carlo estimators can be summarized by Eq. (6), and the particle approximation of the measure of $\bar{\pi}$ is given by

$$\widehat{\pi}^{(N)}(\mathbf{x}) = \sum_{n=1}^{N} \bar{w}_n \delta(\mathbf{x} - \mathbf{x}_n), \tag{7}$$

where $\delta(\mathbf{x})$ is the Dirac delta function. This formulation encompasses jointly MCMC and IS, and in the former case, we have access to the values of the unnormalized weights w_n. Hence, in the IS setting, an estimator $\widehat{Z} = \frac{1}{N}\sum_{n=1}^N w_n$ of the marginal likelihood Z is also available.

C. Goal

In this work, we address the problem of summarizing the information contained in a set of N weighted or unweighted samples generated by a Monte Carlo sampling technique, with a smaller amount $M < N$ of weighted samples. This problem is strictly related to the more general challenge: summarizing the required information of a given target density $\bar{\pi}(\mathbf{x})$, using a particle approximation (with the smallest amount of weighted particles). Generally, there is a loss of information. More precisely, given a Monte Carlo approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ in Eq. (7), with N samples, we desire to construct another particle approximation

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \bar{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \tag{8}$$

where $M < N$, $\sum_{m=1}^M \bar{a}_m = 1$, and $\mathbf{s}_m \in \mathcal{D}$, sharing with $\widehat{\pi}^{(N)}$ the required properties. The goal is to compress the statistical information contained in $\widehat{\pi}^{(N)}(\mathbf{x})$, reducing as much as possible the loss of information. We refer to $\bar{a}_m$ as summary weights and, to $\mathbf{s}_m$, as summary particles. The rate of compression is clearly given by $\eta = \frac{N}{M}$. Note that when $\eta = 1$ we have no compression whereas, when $\eta = N$, we have the maximum compression ($1 \leq \eta \leq N$).

D. Related works

In the literature, two families of possible solutions have been proposed for different but related purposes. The first one is based on a bootstrap technique, and can be always used. The second one is the moment-matching approach, and is available only for a limited type target pdfs $\bar{\pi}(\mathbf{x})$.

Bootstrap solution. Let assume that we have N unweighted samples. A simple approach for compression consists in choosing uniformly M samples within the N possible ones. Similarly, in the case of weighted samples, this strategy consists in resampling M times within the set $\{\mathbf{x}_n\}_{n=1}^N$ according to the normalized weights $\bar{w}_n$, $n = 1, ..., N$ [3]. Then, a proper aggregated weight is associated to the resampled particles [3], [32], [31]. This kind of compression scheme has been widely used in different works (explicitly or implicitly), from distributed particle filtering methods and other sophisticated Monte Carlo algorithms [3], [43], [36], [48].

Moment-matching solution. For simplicity and without loss of generality, let us consider $d_X = 1$, i.e., $x \in \mathbb{R}$. For some specific types of target pdfs $\bar{\pi}(x)$ and specific domains $\mathcal{D}$, it is possible to obtain a deterministic particle approximation $\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^M \rho_m \delta(x - s_m)$ where the weights ρ_m and the particles s_m are solutions of the nonlinear moment-matching system below,

$$\sum_{m=1}^{M} \rho_m s_m^r = \int_{\mathcal{D}} x^r \bar{\pi}(x)dx \quad \text{for} \quad r = 1, ..., R = 2M, \tag{9}$$

where the true values of the first $2M$ non-central moments, $\int_{\mathcal{D}} x^r \bar{\pi}(x)dx$, must be known. Hence we have $2M$ unknowns (the M weights ρ_m and the M particles s_m) and $R = 2M$ equations. Since the system is highly nonlinear, in general, the analytical solution is available only in few particular cases. These solutions are known as *Gaussian Quadratures* [45], the corresponding deterministic particle approximation provide a perfect-matching with the first $2M$ moments (zero loss of information in the approximation of these moments). Quadrature rules and related sigma point methods have been widely applied within several generalized Kalman filtering techniques [1], [21], [45].

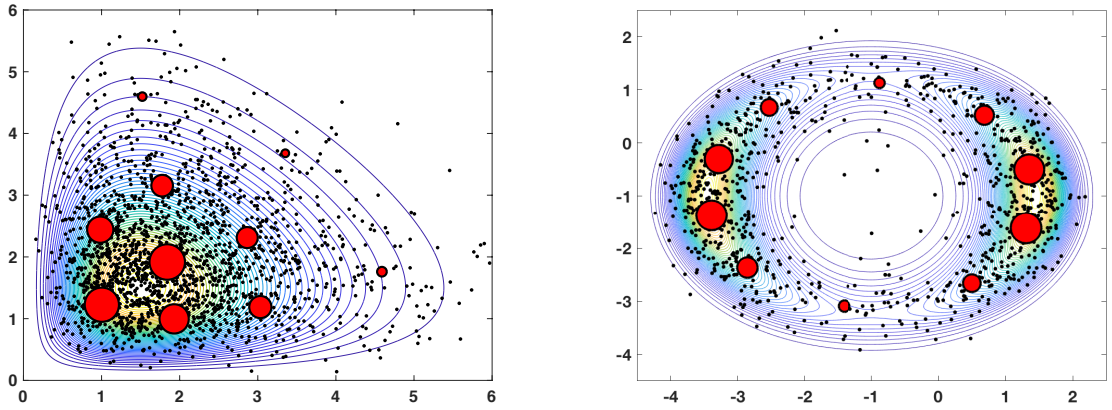


Figure 1. One run of a C-MC scheme with $M = 10$, for two different clouds of $N = 10^3$ samples (represented by dots). Each figure represents a different target $\bar{\pi}(\mathbf{x})$ (shown by the contour plots). The size of the circles is proportional to the corresponding summary weight.

III. Compressed Monte Carlo (C-MC)

In this work, we introduce a compression approach that improves the bootstrap strategy and extends the applicability of the moment-matching scheme, both described above. We consider the cases of compressing unweighted and weighted samples, e.g., the N samples have been previously generated by an MCMC algorithm or an IS technique, respectively. Figure 1 shows two examples of C-MC approximation with $M = 10$ summary particles. The size of the circles is proportional to the corresponding summary weight.

A. Stratification

The underlying grounds of C-MC are based on the so-called stratified sampling [28], [41]. The idea is to divide the support domain $\mathcal{D}$ of the random variable $\mathbf{X}$ into M separate and mutually exclusive regions. More specifically, let us consider an integer $M \in \mathbb{N}^+$, and a partition $\mathcal{P} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ of the state space with M disjoint subsets,

$$\begin{aligned} &\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_M = \mathcal{D}, \\ &\mathcal{X}_i \cap \mathcal{X}_k = \emptyset, \qquad i \neq k, \quad \forall i, j \in \{1, ..., M\}. \end{aligned} \tag{10}$$

We assume that all $\mathcal{X}_m$ are convex sets. Then, in the simplest version of the stratification approach, one sample is drawn from each sub-region, and finally all the generated samples are combined for providing an estimator of $I(h)$. We also denote the area of $\bar{\pi}(\mathbf{x})$ restricted in $\mathcal{X}_m$ as

$$\begin{aligned} \bar{a}_m = \mathbb{P}(\mathbf{X} \in \mathcal{X}_m) &= \int_{\mathcal{X}_m} \bar{\pi}(\mathbf{x}) d\mathbf{x} = \frac{1}{Z} \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}, \\ &= \frac{Z_m}{Z} = \frac{Z_m}{\sum_{j=1}^{M} Z_j}, \end{aligned} \tag{11}$$

where $Z_m = \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}$ and $Z = \sum_{j=1}^{M} Z_j = \int_{\mathcal{D}} \pi(\mathbf{x}) d\mathbf{x}$. Note that $\sum_{m=1}^{M} \bar{a}_m = 1$. The target density can be expressed as a mixture of M non-overlapped densities,

$$\bar{\pi}(\mathbf{x}) = \sum_{m=1}^{M} \bar{a}_m \left[\frac{1}{\bar{a}_m} \bar{\pi}(\mathbf{x}) \mathbb{I}_{\mathcal{X}_m}(\mathbf{x}) \right] = \sum_{m=1}^{M} \bar{a}_m \bar{\pi}_m(\mathbf{x}), \tag{12}$$

where

$$\bar{\pi}_m(\mathbf{x}) = \frac{1}{\bar{a}_m} \bar{\pi}(\mathbf{x}) \mathbb{I}_{\mathcal{X}_m}(\mathbf{x}) = \frac{1}{Z_m} \pi(\mathbf{x}) \mathbb{I}_{\mathcal{X}_m}(\mathbf{x}), \tag{13}$$

is the m-th density in the mixture, and $\mathbb{I}_{\mathcal{X}_m}(\mathbf{x})$ is an indicator function that is 1 when $\mathbf{x} \in \mathcal{X}_m$ and 0 otherwise.

Stratified MC estimators. In order to simulate a sample $\mathbf{x}^*$ from $\bar{\pi}(\mathbf{x})$, we can draw an index $j^* \in \{1, ..., M\}$ according to the probability mass function $\bar{a}_m$, $m = 1, ..., M$ and the draw $\mathbf{x}^* \sim \bar{\pi}_{j^*}(\mathbf{x})$. Alternatively, we could yield an approximation of the measure of $\bar{\pi}$, drawing one sample from each region, i.e., $\mathbf{s}_m \sim \bar{\pi}_m(\mathbf{x})$, and then assign to each sample the weight $\bar{a}_m$, $m = 1, ..., M$. Therefore, in this scenario, the corresponding estimator of the integral $I(h)$ in Eq. (2) and the particle approximation are, respectively,

$$\widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \bar{a}_m h(\mathbf{s}_m), \text{ and} \tag{14}$$

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \bar{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \tag{15}$$

where $\mathbf{s}_m \sim \bar{\pi}_m(\mathbf{x}) = \frac{1}{Z_m} \pi(\mathbf{x}) \mathbb{I}_{\mathcal{X}_m}(\mathbf{x})$, hence $\mathbf{s}_m \in \mathcal{X}_m$. See the `Supplementary Material`, for extensions and further details.

B. C-MC algorithms

Let consider N weighted samples $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^{N}$ generated by a MC scheme, and let M be a constant value such that $M < N$. Given the partition in Eq. (10), i.e., $\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_M = \mathcal{D}$ formed by convex, disjoint sub-regions $\mathcal{X}_m$, we denote the subset of the set of indices $\{1, ..., N\}$,

$$\mathcal{J}_m = \{i = 1, ..., N : \ \mathbf{x}_i \in \mathcal{X}_m\},$$

which are associated with the samples in the m-th sub-region $\mathcal{X}_m$. The cardinality $|\mathcal{J}_m|$ denotes the number of samples in $\mathcal{X}_m$, and we have $\sum_{m=1}^{M} |\mathcal{J}_m| = N$.

C-MC approximation. We can compress the information

contained in the particle approximation of Eq. (7), constructing an empirical stratified approximation based on M weighted particles $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$, i.e.,

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \tag{16}$$

so that for a specific moment the resulting estimator is

$$\widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m), \tag{17}$$

where $\widehat{a}_m$ is an approximation of $\bar{a}_m = \int_{\mathcal{X}_m} \bar{\pi}(\mathbf{x}) d\mathbf{x}$ in Eq. (11), considering the given samples.

Normalized C-MC weights. We can write

$$\begin{aligned} \widehat{a}_m &= \int_{\mathcal{X}_m} \widehat{\pi}^{(N)}(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{N} \bar{w}_i \int_{\mathcal{X}_m} \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x}, \\ &= \sum_{i \in \mathcal{J}_m} \bar{w}_i. \end{aligned} \tag{18}$$

Hence, in the case of compressing samples generated by a standard MC or MCMC schemes, since $\bar{w}_i = \frac{1}{N}$, we obtain $\widehat{a}_m = \frac{|\mathcal{J}_m|}{N}$ that is again an estimate of the probability $\bar{a}_m = \mathbb{P}(\mathbf{X} \in \mathcal{X}_m)$ in Eq. (11). In the IS case, we can also obtain the expression $\widehat{a}_m$ as the ratio of the MC estimators

$$\widehat{Z}_m = \frac{1}{N} \sum_{i \in \mathcal{J}_m} w_i, \qquad \widehat{Z} = \sum_{m=1}^{M} \widehat{Z}_m = \frac{1}{N} \sum_{n=1}^{N} w_n, \tag{19}$$

i.e.,

$$\widehat{a}_m = \frac{\widehat{Z}_m}{\widehat{Z}} = \sum_{i \in \mathcal{J}_m} \frac{w_i}{\sum_{n=1}^{N} w_n} = \sum_{i \in \mathcal{J}_m} \bar{w}_i, \tag{20}$$

as suggested by Eq. (11). Note that, in all cases, we have $0 \leq \widehat{a}_m \leq 1$ and $\sum_{m=1}^{M} \widehat{a}_m = 1$.

Stochastic choice of $\mathbf{s}_m$. We consider different strategies for the selection of the summary particles $\mathbf{s}_m$. The first one is a stochastic approach based on the stratified sampling: each summary particle $\mathbf{s}_m$ is resampled within the set of samples $\mathbf{x}_i \in \mathcal{X}_m$, i.e., $\{\mathbf{x}_i, \text{ with } i \in \mathcal{J}_m\}$, according to the normalized weights,

$$\bar{w}_{m,i} = \frac{w_i}{\sum_{k \in \mathcal{J}_m} w_k} = \frac{\bar{w}_i}{\sum_{k \in \mathcal{J}_m} \bar{w}_k} = \frac{\bar{w}_i}{\widehat{a}_m}, \quad i \in \mathcal{J}_m. \tag{21}$$

In the case of samples generated by standard MC or MCMC schemes, then we obtain $\bar{w}_{m,i} = \frac{1}{|\mathcal{J}_m|}$.

Deterministic choice of $\mathbf{s}_m$. In the same fashion of the deterministic rules and sigma-point construction discussed in Section II-D, we can also set

$$\mathbf{s}_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j, \tag{22}$$

or, if we are interested on the approximation of a specific integral involving a function h, we can set

$$s_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j). \tag{23}$$

These deterministic rules provides a good performance and enjoy interesting properties, as discussed in the next sections and Appendix A.

Other C-MC weights. In some applications, it is required to define an *aggregated weight*

$$W = \sum_{n=1}^{N} w_n = N \widehat{Z}, \tag{24}$$

which is associated to the discrete measure $\widetilde{\pi}^{(M)}$. It is useful in the distributed scenario, as described in Section V [3], [43], [31]. In the case of samples drawn by a standard Monte Carlo or MCMC scheme, the unnormalized weights w_n are unknown, but we can set $W = N$. In the IS scenario, we can also define the unnormalized C-MC weights $a_m = \frac{1}{N} \widehat{a}_m W = \widehat{Z}_m$. These weights, a_m, can be employed for reconstructing the estimator of the marginal likelihood. Indeed, we have

$$\widetilde{Z} = \frac{1}{N} \sum_{m=1}^{M} a_m = \frac{1}{N} \sum_{m=1}^{M} \widehat{Z}_m = \widehat{Z}, \tag{25}$$

recovering perfectly the IS estimator $\widehat{Z}$. Table II summarizes the main expressions introduced in this section.

Table II
SUMMARY OF THE MAIN C-MC EXPRESSIONS.

Scheme	w_i	$\bar{w}_i$	$\widehat{a}_m$	a_m	W
IS	$\frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i)}$	$\frac{w_i}{\sum_{n=1}^{N} w_n}$	$\sum_{i \in \mathcal{J}_m} \bar{w}_i$	$\widehat{Z}_m$	$\sum_{i=1}^{N} w_i$
MCMC	—	$\frac{1}{N}$	$\frac{\|\mathcal{J}_m\|}{N}$	—	N
$\widehat{Z}_m = \frac{1}{N} \sum_{i \in \mathcal{J}_m} w_i$		$\widehat{Z} = \frac{1}{N} \sum_{i=1}^{N} w_i$		$\bar{w}_{m,j} = \frac{\bar{w}_i}{\widehat{a}_m}$	

Additional observation. Note also that the estimator $\widehat{I}^{(N)}(h)$ can be expressed as linear combination of partial estimators, i.e.,

$$\begin{aligned} \widehat{I}^{(N)}(h) = \sum_{i=1}^{N} \bar{w}_i h(\mathbf{x}_i) &= \sum_{m=1}^{M} \sum_{i \in \mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i), \\ &= \sum_{m=1}^{M} \widehat{a}_m \sum_{i \in \mathcal{J}_m} \bar{w}_{m,i} h(\mathbf{x}_i) \\ &= \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_m(h), \end{aligned} \tag{26}$$

where we have used $\bar{w}_{m,i} = \frac{\bar{w}_i}{\widehat{a}_m}$ as shown in Eq. (21), and we

have define the partial estimators $\widehat{I}_m(h) = \sum_{i\in\mathcal{J}_m} \bar{w}_{m,i} h(\mathbf{x}_i)$, for $m = 1, ..., M$. Namely, the MC estimator $\widehat{I}^{(N)}(h)$ of $I(h)$ can be expressed as a *convex* combination of the M partial MC estimators, since $\sum_{m=1}^{M} \widehat{a}_m = 1$. A similar expression is valid for the particle approximations, i.e.,

$$\widehat{\pi}^{(N)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \widehat{\pi}_m(\mathbf{x}), \quad \text{where} \tag{27}$$

$$\widehat{\pi}_m(\mathbf{x}) = \sum_{i\in\mathcal{J}_m} \bar{w}_{m,i} \delta(\mathbf{x} - \mathbf{x}_i). \tag{28}$$

IV. Analysis of C-MC

Proper partition and consistency. Let us focus on the way the partition is formed. A partition rule is proper if, when $M = N$, then $|\mathcal{J}_m| = 1$ (note that $m = n$ in this case), i.e., in the limit case of $M = N$ we consider all the MC samples as summary samples. Recall that, for $M < N$, the C-MC estimators are unbiased as shown in the `Supp. Material` (with $K_m = 1$ and $V = M$). Furthermore, if the partition rule is proper then, for $M = N$, the C-MC estimators will coincide with the non-compressed MC estimators. Hence, as $M \to N$ and $N \to \infty$, the consistency is ensured.

Save in transmission. Let us consider the parallel or distributed framework with a common central node. In C-MC, only the M pairs $\{\widehat{a}_m, \mathbf{s}_m\}_{m=1}^{M}$ are transmitted to the central node, instead of the N pairs. Since, $\mathbf{x}, \mathbf{s} \in \mathbb{R}^{d_X}$, without compression, we need to transmit Nd_X scalar values in case of unweighted samples, or $N(d_X + 1)$ scalar values in the case of weighted samples. With the proposed compression scheme, the transmission of only $M(d_X + 1)$ scalar values is required.

A. Compression Loss

Loss for the deterministic C-MC. Let us consider the deterministic choice of the summary particles as

$$\mathbf{s}_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j, \qquad m = 1, ..., M. \tag{29}$$

Hence, keeping fixed $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^{N}$ and the partition, the summary particles $\mathbf{s}_m$ defined in Eq. (29) are also fixed. Recall that the standard MC estimator and the corresponding C-MC estimator are

$$\widehat{I}^{(N)}(h) = \sum_{n=1}^{N} \bar{w}_n h(\mathbf{x}_n), \quad \widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m).$$

For a specific function h, the information loss for a C-MC scheme can be measured with the squared error, i.e.,

$$\ell(h) = (\widehat{I}^{(N)}(h) - \widetilde{I}^{(M)}(h))^2, \tag{30}$$

or more generally,

$$\ell(h, f) = (\widehat{I}^{(N)}(h) - \widetilde{I}^{(M)}(f))^2, \tag{31}$$

where $f(\mathbf{x})$ is another integrable function. Furthermore, considering a family $\mathcal{H}$ of R functions, i.e., $\mathcal{H} = \{h_1(\mathbf{x}), ..., h_R(\mathbf{x})\}$, we can write we can define the loss as

$$\mathcal{L}_R = \sum_{r=1}^{R} \xi_r^2 \ell(h_r) = \sum_{r=1}^{R} \xi_r^2 \left(\widehat{I}^{(N)}(h_r) - \widetilde{I}^{(M)}(h_r)\right)^2, \tag{32}$$

which is a weighted average of the squared errors, with weights ξ_r^2. For instance, we can set $\xi_r^2 \propto \frac{1}{\left[\widehat{I}^{(N)}(h_r)\right]^2}$ if $\widehat{I}^{(N)}(h_r) \neq 0$, so that $\mathcal{L}_R$ is equivalent to a sum of the relative errors, or simply $\xi_r^2 = \frac{1}{R}$. Moreover, recalling that $\widehat{I}^{(N)}(h) = \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_m(h)$ as shown in Eq. (26), we can write

$$\begin{aligned} \ell(h) &= \left(\widehat{I}^{(N)}(h) - \widetilde{I}^{(M)}(h)\right)^2 \\ &= \left(\sum_{m=1}^{M} \widehat{a}_m \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) - \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m)\right)^2. \end{aligned}$$

We can rewrite it as

$$\ell(h) = \left(\sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) - h(\mathbf{s}_m)\right]\right)^2.$$

Recalling $\widehat{\pi}_m(\mathbf{x}) = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \delta(\mathbf{x} - \mathbf{x}_j)$ and the definition of $\mathbf{s}_m$ in Eq. (29), we can also write

$$\ell(h) = \left(\sum_{m=1}^{M} c_m(h)\right)^2, \tag{33}$$

where

$$c_m(h) = \widehat{a}_m \Big[\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) - h\Big(\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j\Big)\Big], \tag{34}$$

and we have replaced the specific choice $\mathbf{s}_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j$ in Eq. (29). Using also the equalities $\bar{w}_{m,j} = \frac{\bar{w}_j}{\widehat{a}_m}$ and $\widehat{a}_m = \sum_{j\in\mathcal{J}_m} \bar{w}_j$, we obtain

$$\begin{aligned} c_m(h) &= \sum_{j\in\mathcal{J}_m} \bar{w}_j h(\mathbf{x}_j) - \widehat{a}_m h\Big(\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j\Big), & (35) \\ &= \sum_{j\in\mathcal{J}_m} \bar{w}_j \Big[h(\mathbf{x}_j) - h\Big(\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j\Big)\Big]. & (36) \end{aligned}$$

The expressions (34)-(36) only depend on the MC samples $\{\mathbf{x}_n, \bar{w}_n\}$ and the partition, that we have considered pre-established and fixed. Note also that if h is a linear function, then we have a zero-loss compression, i.e., $\ell(h) = 0$. The choice in Eq. (29) is interesting since it provides a very good performance (see Section VI) and also resembles a deterministic quadrature rule with weighted nodes (it can be interpreted an approximate sigma-point construction [21], [45]).

Zero-loss compression. If we are interested only in one specific integral $I(h) = \int_{\mathcal{D}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}$, it is convenient to apply C-MC with the following summary particles

$$s_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j), \tag{37}$$

as highlighted by the theorem below.

Theorem 1. *If s_m as in Eq. (37) is chosen, for $m = 1, ..., M$, and the linear mapping $f(x) = x$, we have $\widehat{I}^{(N)}(h) = \widetilde{I}^{(M)}(f)$, i.e., zero-compression loss $\ell(h, f) = 0$.*

See Appendix A for the proof. Therefore, if we are interested only in one specific integral involving $\bar{\pi}(\mathbf{x})$, we can obtain a perfect compression by choosing the summary particles as in Eq. (37). With the choice in Eq. (37), $s_m \in \mathbb{R}$ is a scalar value since we have assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$ for simplicity, instead of the more general assumption $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^s$, and $s \geq 1$. However, all the presented results are valid for the general case with $s \geq 1$.

Zero-loss estimator of the marginal likelihood. In the weighted sample scenario, we have also the estimator of the marginal likelihood $\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n$. The corresponding C-MC estimator is $\widetilde{I}^{(M)} = \frac{1}{M}\sum_{m=1}^{M} a_m = \frac{1}{M}\sum_{m=1}^{M} Z_m = \widehat{Z}$ as shown in Eq. (25), hence the loss is $\left(\widetilde{I}^{(M)} - \widehat{Z}\right)^2 = 0$. Namely, we always recover the IS estimator of the marginal likelihood, without any loss.

Loss for the stochastic C-MC. Let us consider the case when $\mathbf{s}_m$ is resampled randomly in each partition, according to the weights $\bar{w}_{m,j}$ in Eq. (21). Given the set of weighted samples $\mathcal{S} = \{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$, we can define the conditional expected mean-square error,

$$\ell(h) = \mathrm{E}_{\widehat{\pi}_m}[(\widetilde{I}^{(M)}(h) - \widehat{I}^{(N)}(h))^2|\mathcal{S}]. \tag{38}$$

Note that, in this case, we have

$$\mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}] = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) = \widehat{I}_m(h). \tag{39}$$

Given Eq. (26), we can also write as

$$\widehat{I}^{(N)}(h) = \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_m(h) = \sum_{m=1}^{M} \widehat{a}_m \mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}], \tag{40}$$

so that

$$\widetilde{I}^{(M)}(h) - \widehat{I}^{(N)}(h) = \sum_{m=1}^{M} \widehat{a}_m \Big(h(\mathbf{s}_m) - \mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S})]\Big).$$

Taking the expectation of both sides, we have

$$\begin{aligned}&\mathrm{E}_{\widehat{\pi}_m}[\widetilde{I}^{(M)}(h) - \widehat{I}^{(N)}(h)|\mathcal{S}] \\ &= \sum_{m=1}^{M} \widehat{a}_m \Big(\mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m|\mathcal{S}] - \mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}]\Big) = 0,\end{aligned}$$

where we have also used the property $\mathrm{E}[\mathrm{E}[Z]] = \mathrm{E}[Z]$. Therefore, the conditional mean error is zero, then conditional expected mean-square error can be easily expressed as

$$\begin{aligned}\ell(h) &= \mathrm{E}_{\widehat{\pi}_m}[(\widetilde{I}^{(M)}(h) - \widehat{I}^{(N)}(h))^2|\mathcal{S}] \\ &= \sum_{m=1}^{M} \widehat{a}_m^2 \mathrm{E}_{\widehat{\pi}_m}\Big[\Big(h(\mathbf{s}_m) - \mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}]\Big)^2\Big|\mathcal{S}\Big].\end{aligned}$$

Finally, noting that the term $\mathrm{E}_{\widehat{\pi}_m}\big[\big(h(\mathbf{s}_m) - \mathrm{E}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}]\big)^2\big|\mathcal{S}\big]$ is the definition of the variance of the random variable $h(\mathbf{s}_m)$, we obtain

$$\ell(h) = \sum_{m=1}^{M} \widehat{a}_m^2 \mathrm{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}]. \tag{41}$$

Recalling that $\widehat{a}_m = \sum_{i\in\mathcal{J}_m} \bar{w}_i$ and expressing the variance $\mathrm{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}]$ in terms of weights and samples as shown in Appendix B, we can also write the expected loss as

$$\begin{aligned}\ell(h) &= \sum_{m=1}^{M}\left[\sum_{i\in\mathcal{J}_m}\bar{w}_i\sum_{i\in\mathcal{J}_m}\bar{w}_i|h(\mathbf{x}_i)|^2 - \Big|\sum_{i\in\mathcal{J}_m}\bar{w}_i h(\mathbf{x}_i)\Big|^2\right], \\ \ell(h) &= \sum_{m=1}^{M} c_m(h),\end{aligned} \tag{42}$$

where

$$c_m(h) = \sum_{i\in\mathcal{J}_m}\bar{w}_i\sum_{i\in\mathcal{J}_m}\bar{w}_i|h(\mathbf{x}_i)|^2 - \Big|\sum_{i\in\mathcal{J}_m}\bar{w}_i h(\mathbf{x}_i)\Big|^2. \tag{43}$$

Note that the expression of $\ell(h)$ above is independent from the stochastically-chosen summary particles $\mathbf{s}_m$. This motivates an adaptive procedure for building a good partition, as discussed below.

B. Compression by kernel density estimation

In Eq. (16), we can replace the delta functions with kernel functions $K(\mathbf{x}|\mathbf{s}_m, \boldsymbol{\Sigma}_m)$, for instance Gaussian kernels $\mathcal{N}(\mathbf{x}|\mathbf{s}_m, \boldsymbol{\Sigma}_m)$, of mean $\mathbf{s}_m$ and with a $d_X \times d_X$ covariance matrix $\boldsymbol{\Sigma}_m$ the $d_X \times d_X$ obtained by an empirical estimation considering the samples in $\mathcal{X}_m$, i.e.,

$$\boldsymbol{\Sigma}_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j}(\mathbf{x}_j - \mathbf{s}_m)(\mathbf{x}_j - \mathbf{s}_m)^\top + \delta\mathbf{I}, \tag{44}$$

where $\mathbf{s}_m$ is defined in Eq. (22) and $\delta > 0$. Thus, we also have

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m K(\mathbf{x}|\mathbf{s}_m, \boldsymbol{\Sigma}_m), \tag{45}$$

where $K(\cdot)$ represents a so-called kernel function with location parameter $\mathbf{s}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$. In a distributed scenario, the M triplets $\{\widehat{a}_m, \mathbf{s}_m, \boldsymbol{\Sigma}_m\}_{m=1}^M$ must be transmitted in the central node. The transmission of $M(\frac{1}{2}d_X^2 + \frac{3}{2}d_X + 1)$ scalar values are required. Alternatively, we can use

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma} = \mathrm{diag}(\widehat{\sigma}_1^2, ..., \widehat{\sigma}_{d_X}^2) + \delta\mathbf{I}, \tag{46}$$

where $\widehat{\sigma}_i^2 = \mathrm{var}_{\widehat{\pi}}[x_{i,n}]$ with $i = 1, ..., d_X$ and $n = 1, ..., N$. Hence, only $M(2d_X + 1)$ scalar values must be transmitted.

C. Choice of the partition

In this section, we discuss some examples of practical choices of the partition, and then a possible adaptive procedure. Given the N samples $\mathbf{x}_n = [x_{n,1}, ..., x_{n,d_X}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$, with $n = 1, ..., N$. Then, we list three practical choices from the simplest to the more sophisticated strategy:

P1 Random grid, where each component of the elements of the grid are contained within the intervals $\min_{n=1,...,N} x_{n,i}$ and $\max_{n=1,...,N} x_{n,i}$, for each $i = 1, ..., d_X$.

P2 Uniform deterministic grid, where each component of the elements of the grid are contained within the intervals $\min_{n=1,...,N} x_{n,i}$ and $\max_{n=1,...,N} x_{n,i}$, for each $i = 1, ..., d_X$.

P3 Voronoi partition obtained by a clustering algorithm with M clusters (e.g., the well-known k-means algorithm).

Adaptive procedure. Set $t = 0$ and choose an initial partition $\mathcal{P}_0 = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_{M_0}\}$ of the domain $\mathcal{D}$, with $M_0 = |\mathcal{P}_0|$ disjoint sub-regions, obtained applying the procedure **P2**, for instance. Decide also the stopping condition, choosing a maximum number of sub-regions $M_{\max} < N$ or a threshold for the loss, L. Therefore, while $M_t \leq M_{\max}$ or $\ell(h) \geq L$ (where $\ell(h)$ is computed as in Eq. (33) or (42)), split the m^*-th sub-region, with

$$m^* = \arg\max_m c_m(h). \tag{47}$$

Repeat the procedure above, until the desired stopping condition is reached. For the stochastic C-MC scheme, this procedure can be extended jointly for several functions h. Recall that we define as a proper partition rule, *any* partition rule such that when $M = N$, then $\mathbf{s}_m = \mathbf{x}_n$ and $\widehat{a}_m = \bar{w}_n$ (note that $m = n$ in this case), i.e., in the limit case with $M = N$ we consider all the MC samples as summary samples.

Unweighted C-MC particles. Let us consider to have N samples generated by a standard MC or an MCMC algorithm, i.e., we have $\bar{w}_i = \frac{1}{N}$ for $i = 1, ..., N$. We can choose a partition such that the C-MC weights, $\widehat{a}_m$, are equals. Indeed, if the partition is chosen such that $|\mathcal{J}_m| = \frac{M}{N}$ for all m, then $\widehat{a}_m = \frac{1}{M}$. In this case, the partition is related to the empirical quantiles of the target distribution. In this scenario, we can interpret the C-MC particles as an approximate quasi-Monte Carlo (QMC) samples. Indeed, as the number of MC samples N grows, the distribution of the nodes $\mathbf{s}_m$ follows the definition of low-discrepancy [39]. Furthermore, since $\widehat{a}_m = \frac{1}{M}$ for all m then, in a distributed scenario, the transmission of summary weights can be avoided: the only information still required is the aggregated weight $W = N$, as we show in the next section. However, we recall that the performance in terms of information loss (see Section IV-A) depends on the cost $c_m(h)$ in each sub-region.

V. Application of C-MC and extensions

A. Application to distributed inference

Distributed algorithms have become a very active topic during the past years favored by fast technological developments (e.g., see [7]). In this section, we consider L independent computational nodes where the Monte Carlo computation is performed in parallel. In the literature, specific techniques have been designed for providing a distributed or diffused inference depending on whether a central node is available or not, respectively [37], [14], [17]. Here, we focus on a centralized distributed framework, i.e., we consider a central node where the transmitted local information is properly combined, as represented in Figure 2. We distinguish three different scenarios. In the first one, from now on referred to as the parallel framework, the same dataset $\mathbf{y} \in \mathbb{R}^{d_Y}$ and the same model is shared by all the local nodes [3], [43], [36]. Thus, all the L nodes address the same inference problem, i.e., they deal with the same posterior density. In the second scenario, referred to as model selection case, all the nodes have access to the entire dataset $\mathbf{y}$, but each local node considers a different possible model (different likelihood and/or prior functions), hence they deal with different posteriors [35]. The third case is the distributed scenario, where the observed data are divided over the L local nodes, $\mathbf{y} = [\mathbf{y}_1, ...\mathbf{y}_L]^\top$. Hence, each node addresses a different sub-posterior density which considers only a subset of the data, $\mathbf{y}_\ell \in \mathbb{R}^{d_\ell}$ (note that $\sum_{\ell=1}^L d_\ell = d_Y$) [38], [47]. In these frameworks, a particle compression is often required for reducing the computational and the transmission cost. Below, we develop the three frameworks.

Parallel framework. We assume the use of N_ℓ particles $\{\mathbf{x}_n^{(\ell)}\}_{n=1}^{N_\ell}$ in each local node. First of all, we consider the transmission of all the particles of the central node, without any compression. In this case, the complete Monte Carlo approximation with $N = \sum_{\ell=1}^L N_\ell$ particles can be expressed as

$$\widehat{\pi}_{\mathtt{tot}}^{(N)}(\mathbf{x}) = \sum_{\ell=1}^{L} \frac{W_\ell}{\sum_{j=1}^{N_\ell} W_j} \sum_{n=1}^{N_\ell} \bar{\beta}_n^{(\ell)} \delta(\mathbf{x} - \mathbf{x}_n^{(\ell)}) \tag{48}$$

$$= \sum_{\ell=1}^{L} \bar{\rho}_\ell \; \widehat{\pi}_\ell^{(N_\ell)}(\mathbf{x}), \tag{49}$$

where $\bar{\rho}_\ell = \frac{W_\ell}{\sum_{j=1}^{N_\ell} W_j}$, and $\bar{\beta}_n^{(\ell)} = \frac{1}{N_\ell}$, $W_\ell = N_\ell$ in the case of unweighted samples, or $\bar{\beta}_n^{(\ell)} = \frac{1}{N_\ell \widehat{Z}^{(\ell)}} \frac{\pi(\mathbf{x}_n^{(\ell)})}{q(\mathbf{x}_n^{(\ell)})}$, $W_\ell = N_\ell \widehat{Z}^{(\ell)}$ in the case of weighted samples. Therefore, the complete Monte Carlo approximation $\widehat{\pi}_{\mathtt{tot}}^{(N)}(\mathbf{x})$ is a convex combination of the L local particle approximations $\widehat{\pi}_\ell^{(N_\ell)}(\mathbf{x})$. If we apply a compression scheme transmitting $M_\ell < N_\ell$ samples, $\widetilde{\pi}_\ell^{(M_\ell)}(\mathbf{x})$ as in Eq. (16) or (45), then the joint particle approximation in the central node is

$$\widetilde{\pi}_{\mathtt{tot}}^{(M)}(\mathbf{x}) = \sum_{\ell=1}^{L} \bar{\rho}_\ell \; \widetilde{\pi}_\ell^{(M_\ell)}(\mathbf{x}). \tag{50}$$

with $M = \sum_{\ell=1}^L M_\ell$. We aim to have a small loss of information between the particle approximations, $\widetilde{\pi}_{\mathtt{tot}}^{(M)}$ and $\widehat{\pi}_{\mathtt{tot}}^{(N)}$. In [3], [43], [48], [31], the bootstrap strategy described in Section II-D is applied for the compression. In the numerical experiments, we compare the performance of this strategy with the C-MC approach.

Model Selection. The model selection application is an extension of the parallel framework. Indeed, all the nodes process the entire set of data $\mathbf{y}$, but each local node considers a different possible model $\mathcal{M}_\ell$, hence they address different posterior distributions $\bar{\pi}(\mathbf{x}|\mathbf{y}, \mathcal{M}_\ell)$. In order to tackle this problem, based on the Bayesian Model Averaging (BMA) approach, we need an estimation of the marginal likelihood of each model $\widehat{Z}^{(\ell)}$ (e.g., see [35]). For this reason, it is preferable to apply an IS scheme where an estimator of the

marginal likelihood is easily provided. In this scenario, we have again $\widehat{\pi}_{\mathtt{tot}}^{(N)}(\mathbf{x}) = \sum_{\ell=1}^{L} \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^{L} N_k \widehat{Z}^{(k)}} \widehat{\pi}_{\ell}^{(N_\ell)}(\mathbf{x})$ without compression, and $\widetilde{\pi}_{\mathtt{tot}}^{(M)}(\mathbf{x}) = \sum_{\ell=1}^{L} \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^{L} N_k \widehat{Z}^{(k)}} \widetilde{\pi}_{\ell}^{(M_\ell)}(\mathbf{x})$, with compression. In this scenario, $\bar{\rho}_\ell = \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^{L} N_k \widehat{Z}^{(k)}}$, for $\ell = 1, ..., L$, represents an approximation of the posterior probability mass function (pmf) of the model given the data, i.e., $p(\mathcal{M}_\ell|\mathbf{y})$.

Distributed framework. For simplicity, let us consider $N_\ell = \frac{N}{L}$ and $M_\ell = \frac{M}{L}$, for all $\ell = 1, ..., L$. In this case, all the nodes consider the same model as in the parallel scenario, but each local node can process only a portion of the observed data, $\mathbf{y}_\ell \in \mathbb{R}^{d_\ell}$, with $\sum_{\ell=1}^{L} d_\ell = d_Y$. Considering a disjoint subsets of data and a split contribution of the prior as in [38], the complete posterior can be factorized as

$$\bar{\pi}_{\mathtt{tot}}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \bar{\pi}_\ell(\mathbf{x}). \tag{51}$$

In different works [38], [47], local approximations of the sub-posteriors are provided and transmitted to the central node, obtaining

$$\widehat{\pi}_{\mathtt{tot}}^{(N)}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \widehat{\pi}_{\ell}^{(N_\ell)}(\mathbf{x}). \tag{52}$$

The simplest approach considers Gaussian local approximations [38], [47]. A more sophisticated approach proposed in [38, Section 3.2] considers a mixture of Gaussian pdfs as KDE local approximation using all the $N_\ell = \frac{N}{L}$ samples in each node, i.e.,

$$\widehat{\pi}_{\ell}^{(N_\ell)}(\mathbf{x}) = \sum_{n=1}^{N_\ell} \bar{\beta}_n^{(\ell)} \mathcal{N}(\mathbf{x}|\mathbf{x}_n^{(\ell)}, \delta \mathbf{I}), \tag{53}$$

with $\delta > 0$ and $\mathbf{I}$ is a $d_X \times d_X$ identity matrix. It is easy to see that $\widehat{\pi}_{\mathtt{tot}}^{(N)}(\mathbf{x})$ in Eq. (52) can be expressed as a mixture of N_ℓ^L Gaussian components [38], [20]. It is possible to draw from this mixture of densities, but clearly the cost depends of the number of N_ℓ^L components [20]. Therefore, here the advantage of using a compressed local mixture, $\widetilde{\pi}^{(M_\ell)}(\mathbf{x}) = \sum_{m=1}^{M_\ell} \widehat{a}_m \mathcal{N}(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m)$ with $M_\ell < N_\ell$, is even more apparent than in the parallel scenarios described above. Indeed, using C-MC, we obtain $\widehat{\pi}_{\mathtt{tot}}^{(M)}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \widetilde{\pi}_{\ell}^{(M_\ell)}(\mathbf{x})$, that can be expressed as a mixture of M_ℓ^L Gaussian pdfs [38], [20].

B. Application to particle filtering

In this section, we show how C-MC can be employed for a performance improvement or a decrease of the computational cost within particle filtering (PF) algorithms. Let us consider the following state-space model

$$\begin{cases} \mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ \mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t) \end{cases}, \qquad t = 1, \ldots, T, \tag{54}$$

described by the propagation kernel, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. Below, we provide two novel schemes based on C-MC.

Improved Gaussian particle filter (I-GPF). The Gaussian particle filter (GPF) is a well-known benchmark PF algorithm proposed in [22]. The GPF outperforms of conventional Gaussian filters (like the Extended Kalman filter and its variants) in many scenarios and presents lower complexity than standard particle filters. The resampling steps In the GPF are replaced by a sampling step from an adapted Gaussian density. Table III describes the novel scheme based on C-MC, where the pdf in Eq. (45) plays the role of the Gaussian density in the standard GPF. Note that, with $M = 1$, we recover the standard GPF whereas, with $M = N$, the I-GPF is equivalent to the well-known regularized particle filter [12]. Moreover, resampling from $\widehat{\pi}^{(N)}$ is more costly than resampling from $\widetilde{\pi}^{(M)}$ if $M < N$. Related ideas can be found in the literature [23], [26]. The performance of GPF and I-GPF are compared in Section VI-E.

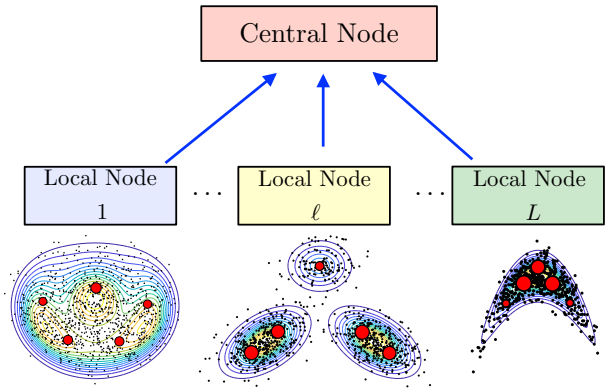


Figure 2. Graphical representation of a distributed Bayesian inference framework with L local computational nodes, and a central node. Each local node addresses a posterior density, which is generally different in each node. If we consider just a parallel framework each node addresses the same posterior.

Table III
IMPROVED GAUSSIAN PARTICLE FILTER (I-GPF)

Initialization: Choose N, M and $\bar{\mathbf{x}}_0^{(i)}$, with $i = 1, ..., N$.
For $t = 1, ..., T$:

1) Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\bar{\mathbf{x}}_{t-1}^{(i)})$, with $i = 1, ..., N$.
2) Compute the M weights
$$w_n = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}), \quad i = 1, ..., N. \tag{55}$$
3) Apply a C-MC scheme for obtaining $\widetilde{\pi}^{(M)}(\mathbf{x})$ in Eq. (45).
4) Draw $\bar{\mathbf{x}}_t^{(n)} \sim \widetilde{\pi}^{(M)}(\mathbf{x})$ with $n = 1, ..., N$.

Compressed particle filter (C-PF). If the compression is applied before the evaluation of the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$, we have an additional reduction of the computational cost. Indeed, in this case, we need to evaluate the likelihood, only $M < N$ times at the summary particles $\mathbf{s}_m$. This is particularly convenient if the evaluation of the likelihood is costly (due to the number of data, or a complex measurement model). The C-PF is given in Table IV. As in I-GPF, the resampling step is performed over M weighted samples instead of N. Thus, C-PF is cheaper and faster than a standard particle filter. Note that the C-MC weights $\widehat{a}_m$ are included in particle weights in Eq.

(56). The weighted points $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$ play a similar role than the sigma points in the Unscented Kalman filter (UKF) [21], [45].

Other possible applications of C-MC are within the so-called parallel partitioned particle filters and multiple particle filters, as an alternative to the use of first moment estimators (or sigma points) for approximating marginal posterior distributions [11]. Similar ideas has been also applied within particle smoothing techniques [13].

Table IV
COMPRESSED PARTICLE FILTER (C-PF)

Initialization: Choose N, M and $\bar{\mathbf{x}}_0^{(i)}$, with $i = 1, ..., N$.
For $t = 1, ..., T$:

1) Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\bar{\mathbf{x}}_{t-1}^{(i)})$, with $i = 1, ..., N$.
2) Apply a C-MC scheme obtaining $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$.
3) Compute the M weights
$$w_m = \widehat{a}_m p(\mathbf{y}_t|\mathbf{s}_m), \quad m = 1, ..., M. \tag{56}$$
and normalized them $\bar{w}_m = \frac{w_m}{\sum_{k=1}^M w_k}$.
4) Obtain $\{\bar{\mathbf{x}}_t^{(n)}\}_{n=1}^N$, by resampling N times within $\{\mathbf{s}_m\}_{m=1}^M$ according to $\bar{w}_m$, with $m = 1, ..., M$.

C. Application to adaptive importance sampling

In the so-called layered adaptive importance sampling (LAIS) algorithm [33] and similar methods [46], an MCMC algorithm is used for obtaining a set of mean parameters $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_T\}$. Then, one sample $\mathbf{x}_t$ is drawn from a proposal density with mean $\boldsymbol{\mu}_t$, i.e., $\mathbf{x}_t \sim q(\mathbf{x}_t|\boldsymbol{\mu}_t, \mathbf{C})$ where $\mathbf{C}$ is a covariance matrix and $t = 1, ..., T$. One possible choice of the weights is

$$w_t = \frac{\pi(\mathbf{x}_t)}{\frac{1}{T}\sum_{k=1}^T q(\mathbf{x}_t|\boldsymbol{\mu}_k, \mathbf{C})}, \tag{57}$$

where a temporal mixture is used in the denominator [33], [46]. With this choice, very good performance can be obtained, but the computational cost of evaluating the weight denominator increases with T^2 [33]. If T is large, the evaluation of the weights in Eq. (57) can be costly. Hence, the C-MC scheme can be applied to the set $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_T\}$ are shown in Table V. More generally, C-MC can be also applied within adaptive MC schemes to obtain a good construction of the adaptive proposal density [6], [5], [4].

D. Extensions: Least Squares CMC (LS-CMC)

If we relax the assumption that the weights $\widehat{a}_m$ must be non-negative, we can obtain better performance in terms of loss in compression. Indeed, given the summary particles $\{\mathbf{s}_m\}_{m=1}^M$ considering a family of $R+1$ functions, i.e., $\mathcal{H} = \{h_0(\mathbf{x}) = 1, h_1(\mathbf{x}), ..., h_R(\mathbf{x})\}$, we can write the following linear system,

Table V
COMPRESSED LAIS (CLAIS)

1) Generate a chain $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_T$ using an MCMC technique (with target π or a tempered version).
2) Draw T samples from $\mathbf{x}_t \sim q(\mathbf{x}_t|\boldsymbol{\mu}_t, \mathbf{C})$, with $t = 1, ..., T$, and where $\mathbf{C}$ is a covariance matrix.
3) Considering the samples $\{\mathbf{x}_t\}_{t=1}^T$, obtain $\widetilde{\pi}^{(M)}(\mathbf{x})$ in Eq. (45) by C-MC, with $M < T$.
4) To each $\mathbf{x}_t$, assign the weight
$$w_t = \frac{\pi(\mathbf{x}_t)}{\widetilde{\pi}^{(M)}(\mathbf{x}_t)}. \tag{58}$$

$$\begin{cases} \sum_{m=1}^M \widehat{a}_m = 1, \\ \sum_{m=1}^M \widehat{a}_m h_1(\mathbf{s}_m) = \widehat{I}^{(N)}(h_1), \\ \vdots \\ \sum_{m=1}^M \widehat{a}_m h_R(\mathbf{s}_m) = \widehat{I}^{(N)}(h_R). \end{cases} \tag{59}$$

with M unknowns and $R+1$ equations. If $M \leq R+1$ the system is overdetermined, and it has in general no solution. However, we can still find a Least Squares (LS) solution for this problem. Indeed, the system in Eq. (59) can be rewritten as

$$\mathbf{H}\widehat{\mathbf{a}} \approx \mathbf{v},$$

where $\mathbf{H}$ is a $(R+1) \times M$ matrix with entries $\mathbf{H}_{ij} = h_i(\mathbf{s}_j)$, $\mathbf{a} = [\widehat{a}_1, ..., \widehat{a}_M]^\top$ is the vector of the unknowns, and $\mathbf{v} = [1, \widehat{I}^{(N)}(h_1), ..., \widehat{I}^{(N)}(h_R)]^\top$. The well-known LS solution is then given by

$$\widehat{\mathbf{a}} = (\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{v}. \tag{60}$$

Note that the weights in the vector $\widehat{\mathbf{a}} = [\widehat{a}_1, ..., \widehat{a}_M]^\top$ could be also negative. For this reason, the range of application of LS-CMC is reduced but, for instance, LS-CMC can be still applied to the pure parallel framework, described in Section V-A.

VI. NUMERICAL EXPERIMENTS

In the section, we test the proposed C-MC techniques in six different numerical examples and compare their performance with the corresponding benchmark methods. In the first experiment, we apply the compression techniques to two sets of Monte Carlo samples. In the second experiment, we consider a localization problem in a wireless sensor network and the use of L local parallel processors. We test the performance of the Compressed LAIS (CLAIS) scheme for performing an inference in an exoplanetary model, in the third example. The last three experiments consider the use of particle filtering. In Section VI-D, we test the proposed C-PF obtaining very promising results. Finally, in Sections VI-E and VI-F, we consider two different object tracking problems with different

measurements and propagation models. Moreover, in Section VI-F a centralized distributed inference problem is considered.

A. First numerical analysis

Let start, for simplicity, with a scalar scenario, i.e., $x \in \mathbb{R}$. Furthermore, we consider two possible target densities: the first one is a Gamma pdf

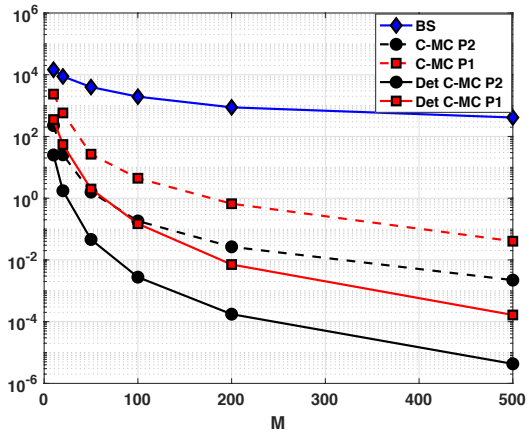
$$\bar{\pi}(x) \propto x^{\alpha-1} \exp\left(-\frac{x}{\kappa}\right), \tag{61}$$

with $\alpha = 4$ and $\kappa = 0.5$, and the second one is a mixture of two Gaussians,

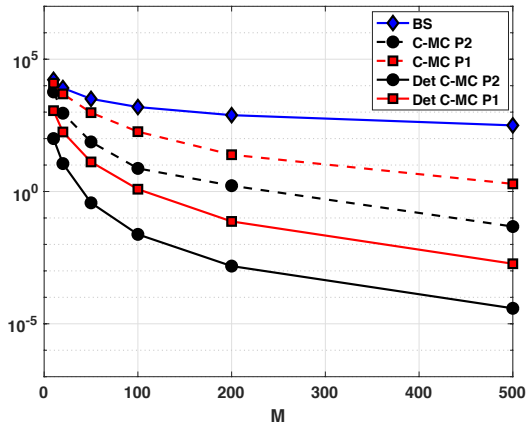
$$\bar{\pi}(x) = \frac{1}{2}\mathcal{N}(x|-2,1) + \frac{1}{2}\mathcal{N}(x|4,0.25). \tag{62}$$

Experiments. We generate $N = 10^5$ Monte Carlo samples from both and compare the bootstrap strategy (BS) with different C-MC schemes. More specifically, we consider two kind of partition procedures: random (P1) and uniform (P2) described in Section IV-C. Furthermore, we compare the stochastic and the deterministic choices of the summary particles $\mathbf{s}_m$, described in Section III. Therefore, for the deterministic C-MC schemes, we consider the use of $\mathbf{s}_m$ in Eq. (22). We repeat the experiment 500 independent runs and average the results. At each run, we compute the loss $\mathcal{L}_5$ with $\xi_r^2 = 1$, for $r = 1, ..., 5$ (i.e., the loss in the first 5 moments) provided by the different techniques. Figure 3 depicts the averaged $\mathcal{L}_5$ as function of the number M of summary particles. Figure 3-**(a)** refers to the Gamma target pdf, whereas Figure 3-**(b)** corresponds to the Gaussian mixture pdf. The results of the BS method are displayed with triangles. The stochastic C-MC schemes are shown with dashed lines, whereas the deterministic C-MC schemes with solid lines.

Discussion. In all cases, C-MC outperforms BS and the deterministic C-MC schemes provide the best results. As expected, the partition P2 (depicted with circles) outperforms P1 (shown with squares). Note that P1 represents the simplest and perhaps the worst possible construction of the partition. However, it is important to remark that the C-MC schemes, even with P1, outperform the BS method.[3]



(a) Gamma target pdf



(b) Mixture target pdf

Figure 3. The loss $\mathcal{L}_5$ as function of M. The results obtained by the bootstrap strategy [3], [43], [31] in Section II-D is depicted with a solid line and rhombuses. The results of C-MC with a random partition (P1), and with a grid partition (P2) are shown by squares and circles, respectively. The results obtained with the deterministic choice of $\mathbf{s}_m$ in Eq. (22) are shown with solid lines (squares and circles), whereas the results corresponding to the random choice of $\mathbf{s}_m$ are provided with dashed lines (squares and circles).

B. Localization in a sensor network with Parallel AIS schemes

In this section, we test the C-MC technique considering the problem of positioning a target in $\mathbb{R}^2$ using a range measurements in a wireless sensor network [19]. Specifically, the target position is modeled as a random vector $\mathbf{X} = [X_1, X_2]^\top$, hence the actual position of the target is a specific realization $\mathbf{X} = \mathbf{x}$. The data (range measurements) are obtained from 3 sensors located at $\mathbf{h}_1 = [3, -8]^\top$, $\mathbf{h}_2 = [10, 0]^\top$, $\mathbf{h}_3 = [0, 10]^\top$, as shown in Figure 4-**(d)**. The likelihood function is induced by the following observation model,

$$Y_j = 20 \log\left(||\mathbf{x} - \mathbf{h}_j||\right) + B_j, \quad j = 1, 2, 3, \tag{63}$$

where $B_j \sim \mathcal{N}(b_j; 0, \lambda_j^2)$. We consider the true position of the target as $\mathbf{x}^* = [x_1^* = 2.5, x_2^* = 2.5]^\top$ and set $\lambda_j = 6$. Then, we generate one measurement y_j from each sensor according to the model in Eq. (63), obtaining the vector $\mathbf{y} = [y_1, y_2, y_3]$. Assuming a uniform prior in the rectangle $\mathcal{R}_z = [-30, 30]^2$, then the posterior density is

$$\bar{\pi}(\mathbf{x}) \propto \left[\prod_{j=1}^{3} \exp\left(-\frac{1}{2\lambda_j^2}(y_j - 20\log\left(||\mathbf{z} - \mathbf{h}_j||\right)^2\right)\right] \mathbb{I}_{\mathcal{R}_z}(\mathbf{x}), \tag{64}$$

where $\mathbb{I}_{\mathcal{R}_z}(\mathbf{x})$ is an indicator function that is 1 if $\mathbf{x} \in \mathcal{R}_z$, otherwise is 0.

Parallel setup. We assume L local computational nodes. At each one, we run an adaptive importance sampler, specifically a standard Population Monte Carlo (PMC) scheme [6]. Each PMC delivers N_ℓ weighted samples as an approximation of the posterior of Eq. (64), after a certain number of iterations [4]. Therefore, we have $\widehat{\pi}_\ell^{(N_\ell)}$ local approximations of N_ℓ particles. In this setting, we have a clear improvement in term of computational times, since the L different PMC algorithms are run in parallel. When all the samples are transmitted to the central

[3]The code of this first example is provided at http://www.lucamartino.altervista.org/CMC_CODE_pub_EX1.zip.

node, we obtain a complete particle approximation $\widehat{\pi}_{\texttt{tot}}^{(N)}$ as in Eq. (48) with $N = \sum_{\ell=1}^{L} N_\ell$ (we set $N_\ell = \frac{N}{L}$). However, in general due to the transmission cost, a particle compression is applied. In this case, we have L local approximations $\widetilde{\pi}_\ell^{(M_\ell)}$, and the central node performs the fusion obtaining $\widetilde{\pi}_{\texttt{tot}}^{(M)}$ as in Eq. (50) with $M = \sum_{\ell=1}^{L} M_\ell$ (we set $M_\ell = \frac{M}{L}$). We measure the quality of the approximation $\widetilde{\pi}_{\texttt{tot}}^{(M)}$ computing the loss (i.e., mean square error) in the estimation of the mean vector, the covariance matrix, skewness, and kurtosis vectors (i.e., overall 9 scalar values) with respect to $\widehat{\pi}_{\texttt{tot}}^{(N)}$. We compare the bootstrap strategy (BS) in [3], [43], [48], [31] and C-MC. For building the partition for C-MC, we perform a k-means clustering with M_ℓ clusters in each local node. The clustering is applied after resampling N_ℓ times within the weighted particles given by PMC. Thus, the partition is given by the M_ℓ Voronoi regions. Then, we consider again the weighted samples produced by the PMC and build the summary weights $\widehat{a}_m$ and summary samples $\mathbf{s}_m$ for each Voronoi region. We average the results over 200 independent runs.

Experiments. The losses of BS (triangles) and C-MC (circles) for different values of M_ℓ and N_ℓ (witht $L = 10$) are depicted in Figures 4 **(a)**-**(b)**-**(c)**. More specifically, in Figure 4-**(a)** we set $N_\ell = 1000$ and vary M_ℓ. In Figure 4-**(b)**, we vary M keeping fixed the compression rate $\eta = \frac{N_\ell}{M_\ell} = 100$, i.e., when M_ℓ grows also N_ℓ is increased. In Figure 4-**(c)**, we set $M_\ell = 10$, and vary N_ℓ. Finally, in Figure 4-**(d)** we set $M_\ell = 10$, $N_\ell = 1000$ and vary L.

Discussion. First of all, we can observe that C-MC always outperforms BS providing the small loss in any scenario. The increase of M_ℓ has always a positive impact as shown in Figures 4-**(a)**-**(b)**. In Figure 4-**(c)**, the compression rate $\eta = \frac{N_\ell}{M_\ell}$ is increasing since M_ℓ is fixed and N_ℓ grows, so that we expect that the performance should become worse as N_ℓ grows. However, in a first moment, the increase of N_ℓ helps both schemes, C-MC and BS, since a better partition can be built with a greater N_ℓ in C-MC by clustering, and the resampling steps used in bootstrap improves its performance with a greater N_ℓ in BS. Moreover, in this scenario, the increase of N_ℓ seems to have a more positive impact on the BS technique. However, Figure 4-**(b)** shows that, if the compression rate $\eta = \frac{N_\ell}{M_\ell}$ is maintained fixed, then C-MC obtains a better improvement. In Figure 4-**(d)**, we can see that the performance improves when L grows.

C. Inference in a exoplanetary model

In this section, we consider the application of the Compressed LAIS (CLAIS) scheme described in Table V to make inference in an exoplanetary system. Let us consider the following simplified observation model of a Keplerian orbit and the radial velocity of the host star,

$$y_j = V + \sum_{i=1}^{N_P} K_i \left[\cos\left(\frac{2\pi}{P_i} t_j + \omega_i\right) + e_i \cos\left(\omega_i\right)\right] + \xi_j, \quad (65)$$

where y_j is the j-th observation, t_j is a known time instants, V is the mean radial velocity, N_p is the number of planets in the system, and K_i is an amplitude, P_i is the period, ω_i is longitude of periastron, e_i the eccentricity of the orbit and $\xi_j \sim \mathcal{N}(0,1)$ [2]. We consider that all the parameters V, K_i, P_i, e_i, ω_i are unknown for $i = 1, ..., N_P$ and also the number of planets N_P is unknown. Note that the dimension of the inference space depends on N_P: if there is no planet in the system $\mathbf{x} = V$ then $d_X = 1$, with $N_P = 1$ we have $\mathbf{x} = [V, K_1, P_1, e_1, \omega_1]^\top$ then $d_X = 5$, with $N_P = 2$ we have $\mathbf{x} = [V, K_1, P_1, e_1, \omega_1, K_2, P_2, e_2, \omega_2]^\top$ hence $d_X = 9$, i.e., generally we have $d_X = 1 + 5N_P$.

Let consider 50 data stacked in a vector $\mathbf{y}$, generated from the model in Eq. (65). Our goal is to make inference regarding the number of N_P and the corresponding parameters, with $0 < N_P \leq 3$. We consider uniform priors $\mathcal{U}([a, b])$ over the parameters ($a = -20$, $b = 20$ for V, $a = 0$, $b = 365$ for P_i, $a = -\pi$, $b = \pi$ for ω_i, $a = 0$, $b = 1$ for e_i) and a uniform discrete prior $p_i = 1/4$ over the number of planets, N_P. We fix N_P, and apply CLAIS with a random walk Metropolis chain [44], of length $T = 20^5$ and set $M = 10$ (see Table V). The partition is built by the approach P2 given in Section IV-C. With CLAIS we can easily estimate the marginal likelihood $\widehat{Z}^{(i)}$ with $i = 0, ..., 3$, using the corresponding IS estimator. Then, the marginal posterior of N_P is approximated by

$$p(N_P = k|\mathbf{y}) \approx \frac{\widehat{Z}^{(k)}}{\sum_{i=0}^{3} \widehat{Z}^{(i)}}, \quad (66)$$

with $k = 0, ..., 3$. We make two experiments. First, we set $N_P = 1$ and then $N_P = 3$ planets and generate the corresponding data $\mathbf{y}$. Note that for computing $p(N_P = k|\mathbf{y})$ we need to integrate out the rest of parameters. The probabilities $p(N_P = k|\mathbf{y})$ obtained in the two experiments are given in Figure 5. Note that the task of providing a good estimation of $\widehat{Z}^{(i)}$ depends on the ability of the sampling method of exploring properly the state space. For this reason, the need of increasing the length T of the MCMC chain raises as the dimension d_X grows. In Figure 5, we can observe that CLAIS is able to recover the number of planets in each experiments. The results are averaged over 100 independent runs.

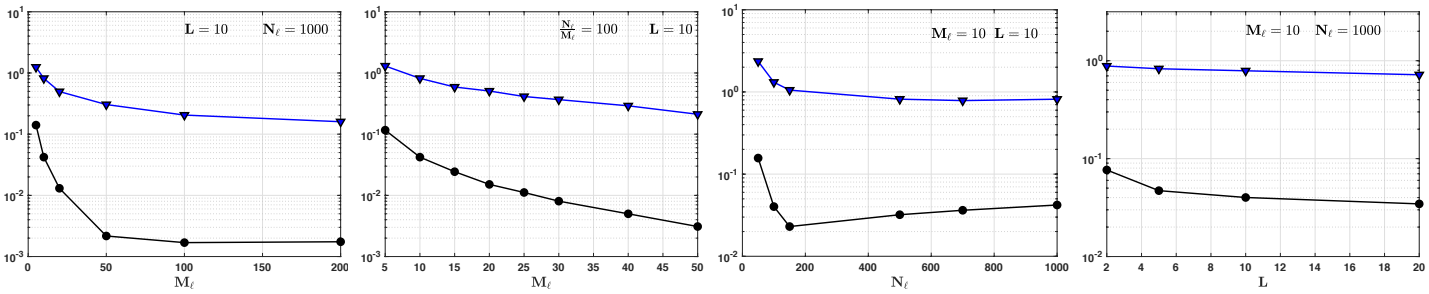
D. Compressed Particle Filtering

This section is devoted to analyzing the performance of the Compressed Particle Filter (C-PF) described in Table IV. Given the following the state-space model

$$\begin{cases} x_t = |x_{t-1}| + v_t \\ y_t = \log(x_t^2) + u_t \end{cases}, \qquad t = 1, \ldots, T, \quad (67)$$

where $v_t \sim \mathcal{N}(0,1)$ and $u_t \sim \mathcal{N}(0,1)$, the goal is to track x_t for $T = 100$ time instants, with a particle filtering algorithm considering $N \in \{100, 1000\}$ particles. We compare the bootstrap particle filter (BPF) [12] and C-PF in terms of the Mean Square Error (MSE) in the estimation of $x_{1:T}$. We apply C-PF with different values of M (clearly, with $M \leq N$). We consider the deterministic C-MC scheme with a uniform construction P2 of the partition.

Figure 6 shows the MSE (averaged over 5000 independent runs) as function of the compression rate, given by the ratio $\frac{M}{N}$. The solid lines represent the MSE obtained by the BPF. The dashed line with squares corresponds to the C-PF (using the deterministic compression) with $N = 100$, whereas the



(a) (b) (c) (d)

Figure 4. **(a)-(b)-(c)-(d)** Results in terms of information loss for the localization problem in wireless sensor network: C-MC is shown with circles and the bootstrap strategy with triangles.

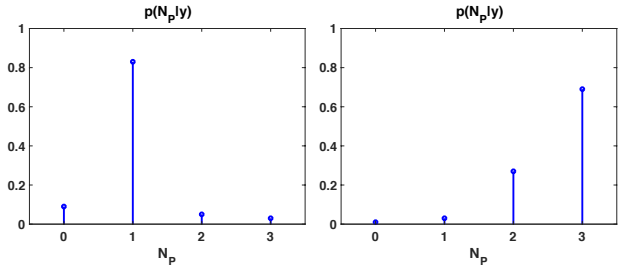


Figure 5. Approximation of the marginal posterior probability mass of number of planets N_P obtained by using CLAIS with $T = 20^5$ and $M = 10$ ($N_P = 1$ on the left and $N_p = 3$ on the right).

dashed line with circles corresponds to the C-PF with $N = 1000$. Note that C-PF virtually obtains the same performance of the BPF with approximately 85% fewer evaluations of the likelihood function. We recall that the N resampling steps are performing over M particles instead of N. Furthermore, fixing the compression rate of $\frac{M}{N}$, It is interesting to note that the performance of C-PF improves when N grows. Finally, we have applied an unscented Kalman filter (UKF) [21], [45], and computed its MSE in estimating $x_{1:T}$. C-PF obtains the same or better MSE for $M \geq 20$ when $N = 1000$.

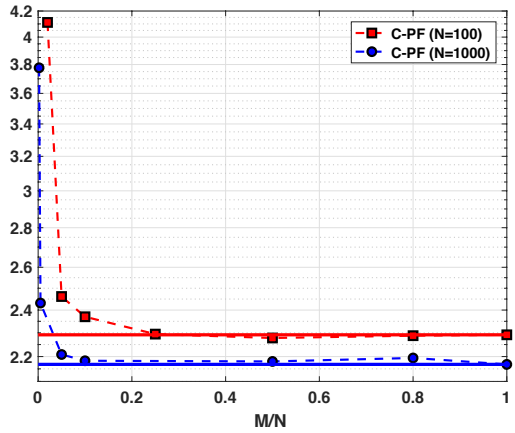


Figure 6. MSE as function of the compression rate $\frac{M}{N}$. The dashed line with squares corresponds to the C-PF with $N = 100$, whereas with circles corresponds to the C-PF with $N = 1000$. The solid lines corresponds to the bootstrap particle filter with $N = 100, 1000$. C-PF virtually obtains the same performance of the bootstrap particle filter with approximately 85% less evaluations of the likelihood function.

E. Tracking with Improved Gaussian particle filtering (I-GPF)

In this section, we compare the performance of the benchmark Gaussian particle filter (GPF) with an Improved GPF (I-GPF) method which employs C-MC, described in Table III. For this comparison, we consider a bearings-only tracking (BOT) model. The BOT model arises in different engineering applications. More specifically, we considers tracking position and velocity of an object moving in a 2D space, $\mathbf{x}_t = [p_{t,1}, p_{t,2}, v_{t,1}, v_{t,2}]^\top$ where $\mathbf{p}_t = [p_{t,1}, p_{t,2}]^\top$ and $\mathbf{v}_t = [v_{t,1}, v_{t,2}]^\top$ are the position and velocity vectors, respectively. The measurements taken by the sensor are the bearings or angles regarding the sensor position, contaminated by noise. The range of the object, that is, the distance from the sensor, is not observed. The transition model is

$$\mathbf{x}_{t+1} = \mathbf{\Phi}\mathbf{x}_t + \mathbf{\Gamma}\boldsymbol{\eta}_{t+1}, \qquad t = 1, ..., T,$$

where

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{\Gamma} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and $\boldsymbol{\eta}_{t+1} = [\eta_{1,t+1}, \eta_{2,t+1}]^\top \sim \mathcal{N}(\mathbf{0}, \sigma_\eta^2 \mathbf{I})$. The measurements consist of the true bearing of the target contaminated by noise, i.e., the measurement equation is

$$y_t = \arctan\left[\frac{p_{t,1}}{p_{t,2}}\right] + \zeta_t$$

where $\zeta_{i,t} \sim \mathcal{N}(0, \sigma_\zeta^2)$. Note that, with this kind of observation model, we obtain no information about the range of the object from the measurement. At the t-th iteration, the GPF algorithm replaces the resampling steps in a standard particle filter by constructing a Gaussian density, given the weighted samples, and sampling from it. In the I-GPF scheme, the pdf in Eq. (45) ($\delta = 0.1$) based on the deterministic C-MC, plays the role of the Gaussian density in the standard GPF (see Table III). We consider a uniform partition P2 (see Section IV-C). We generate trajectories of length $T = 15$ and measurements from

the model with parameters $\mathbf{x}_0 = [-0.05, 0.001, 0.7, -0.055]$, $\sigma_\eta = 0.001$, $\sigma_\eta = 0.005$, and number of particles $N = 1000$. We compute the MSE (averaged in the four components) in estimation of $\mathbf{x}_{1:T}$ (averaged over 10^5 runs) using GPF and I-GPF with $M \in \{5, 10, 20\}$. The results in Table VI, shown that I-GPF outperforms GPF.

Table VI
MSE IN ESTIMATION OF $\mathbf{x}_{1:T}$ (EX. IN SECTION VI-E).

Method	$M = 5$	$M = 10$	$M = 20$	$M = 30$
GPF	0.0186			
I-GPF	0.0157	0.0145	0.0121	0.0098

F. Application to distributed particle filtering (DPF)

In this section, we consider the nearly coordinated turn model, with state $\mathbf{x}_t = [p_{t,1}, p_{t,2}, v_{t,1}, v_{t,2}, \gamma_t]^\top$, i.e., $d_X = 5$, which contains the position and velocity coordinates ($\mathbf{p}_t = [p_{t,1}, p_{t,2}]^\top$ and $\mathbf{v}_t = [v_{t,1}, v_{t,2}]^\top$), as well as the turn rate γ_t. Thus, the transition model is

$$\mathbf{x}_{t+1} = \begin{pmatrix} 1 & 0 & \frac{\sin(\gamma_t)}{\gamma_t} & \frac{\cos(\gamma_t)-1}{\gamma_t} & 0 \\ 0 & 1 & \frac{\cos(\gamma_t)-1}{\gamma_t} & \frac{\sin(\gamma_t)}{\gamma_t} & 0 \\ 0 & 0 & \cos(\gamma_t) & -\sin(\gamma_t) & 0 \\ 0 & 0 & \sin(\gamma_t) & \cos(\gamma_t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_t + \boldsymbol{\eta}_{t+1},$$

where $t = 1, ..., T$, $\boldsymbol{\eta}_{t+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$ with $\mathbf{D} = \text{diag}([0.05, 0.05, 0.04, 0.04, 0])$, and constant turn rate $w_t = 0.139$. The measurement equations is

$$y_i = h_i(\mathbf{x}_t) + \zeta_{i,t} \tag{68}$$

where $h_i(\mathbf{x}_t)$ represents the specific sensor and $\zeta_{i,t} \sim \mathcal{N}(0, \sigma_i^2)$. We consider K sensors distributed uniformly in the square region $\mathcal{R} = [-3, 3] \times [-3, 3]$, with the position denoted as $\mathbf{r}_i = [r_{i,1}, r_{i,2}]^\top$, $i = 1, ..., K$. We consider 4 different types of sensors: $K/4$ of them are bearing-only sensors,

$$h_i(\mathbf{x}_t) = \arctan\left[\frac{p_{t,1} - r_{i,1}}{p_{t,2} - r_{i,2}}\right], \tag{69}$$

with $\sigma_i = 0.175$ and $K/4$ of them are the signal-strength sensors,

$$h_i(\mathbf{x}_t) = \frac{1}{||\mathbf{p}_t - \mathbf{r}_i||^2 + a}, \tag{70}$$

with $a = 10^{-4}$, $\sigma_i = 2$, $K/4$ of them are the range-measurement sensors,

$$h_i(\mathbf{x}_t) = ||\mathbf{p}_t - \mathbf{r}_i||, \tag{71}$$

with $\sigma_i = 0.14$ and $K/4$ of them are the radial-velocity sensors described by

$$h_i(\mathbf{x}_t) = \frac{(\mathbf{p}_t - \mathbf{r}_i) \cdot \mathbf{v}_t}{||\mathbf{p}_t - \mathbf{r}_i||}, \tag{72}$$

where $\sigma_i = 0.004$ and $\cdot$ denotes the scalar product. Each sensor provides one measurement, y_i, per iteration. We consider $L \in \{4, 8\}$ local processors distributed uniformly in the area $\mathcal{R}$ (in a grid form). Each sensor transmit to the closest local processor. Hence, each local processor addresses a partial posterior $\bar{\pi}_\ell^{(t)}$, using Eqs. (45)-(46) ($\delta = 0.1$), with different number of observations. In the central node, we perform the information fusion obtaining $\widehat{\pi}_{\texttt{tot}}^{(t)}$ or, with compression $\widetilde{\pi}_{\texttt{tot}}^{(t)}$. The deterministic C-MC is performed creating a partition of M sets creating a uniform grid strategy P2 suggested in Section IV-C. We compare the deterministic C-MC with the ideas proposed in [40] adapted for the central node scenario that coincides with the first method proposed in [38] but employed within a particle filtering context. We set $T = 10$, $K = \{8, 16, 40\}$ and $M = 4$ and compute the MSE in estimation of $\mathbf{x}_{1:T}$, averaged over 10^4 independent runs. The results are shown in Table VII. The proposed technique obtains the smallest MSE since, in general, provides a more robust estimation of $\mathbf{x}_{1:T}$.

Table VII
MSE IN ESTIMATION OF $\mathbf{x}_{1:T}$ (EX. IN SECTION VI-F).

Method		$K = 8$	$K = 16$
$L = 4$	[40], [38]	0.332	0.186
	CMC-DPF	0.161	0.095
Method		$L = 4$	$L = 8$
$K = 16$	[40], [38]	0.186	0.301
	CMC-DPF	0.095	0.143

VII. CONCLUSIONS AND FUTURE WORKS

In this work, we have introduced a novel efficient scheme to summarize the information provided by Monte Carlo sampling algorithms. This problem is related to the moment matching approach used in different filtering methods but applicable only for certain target densities. The proposed technique can be applied in different scenarios, for instance, in the distributed inference framework, within advanced particle filtering schemes, or within adaptive Monte Carlo methods. We have introduced three novel Monte Carlo schemes based on C-MC. Among them, the C-PF is particularly promising, since reducing considerably the number of the likelihood evaluations, C-PF is still able to provide a similar performance of a standard particle filter, with remarkably more evaluations of the likelihood. In the proposed CLAIS method, we have shown that C-MC can be employed for reducing the computational cost of AIS schemes.

The C-MC-based algorithms have been tested in six different numerical experiments, considering several inference problems. The results have shown that C-MC techniques outperform the corresponding benchmark methods. The deterministic C-MC scheme appears particularly efficient. As future research line, we plan to study the connection between C-MC and sigma-points approaches (see, e.g., in C-PF). We also plan to analyze the information loss using the Kullback-Leibler (KL) divergence between the C-MC approximation and the true distribution. The LS-CMC scheme (and its regularized versions) also deserves further studies also from a theoretical point of view, trying to overcome the difficulty due to the possibility of obtaining negative weights. The joint use of LS-CMC and C-PF will be also investigated.

REFERENCES

[1] I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, 2009.

[2] S. T. Balan and O. Lahav. Exofit: orbital parameters of extrasolar planets from radial velocities. *M. N. of the Royal Astronomical Society*, 394(4):1936–1944, 2009.

[3] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions Signal Processing*, 53(7):2442–2450, 2005.

[4] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

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

[6] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[7] M. Cetin, L. Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky. Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, July 2006.

[8] W. Chao, M. Rabbat, and S. Blouin. Particle weight approximation with clustering for gossip- based distributed particle filters. *IEEE Int. Workshop Comp Comput. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, pages 85–88, 2015.

[9] W. Ye Chen, L. Mackey, J. Gorham, F. X. Briol, and C. J. Oates. Stein Points. *arXiv:1803.10161*, pages 1–31, 2018.

[10] Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. *In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 1–8, 2010.

[11] P. M. Djuric, T. Lu, and M. F. Bugallo. Multiple particle filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1181–1184, 2007.

[12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[13] M. Klaas et al. Fast particle smoothing: If I had a million particles. *International conference on Machine learning (ICML)*, pages 481–488, 2006.

[14] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis. Set-membership constrained particle filter: distributed adaptation for sensor networks. *IEEE Transactions on Signal Processing*, 59(9):4122–4138, 2011.

[15] P. Fearnhead. Using random Quasi-Monte Carlo within particle filters, with application to financial time series. *Journal of Computational and Graphical Statistics*, 14(4):751–769, 2005.

[16] M. Gerber and N. Chopin. Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579, 2015.

[17] O. Hlinka, F. Hlawatsch, and P.M. Djuric. Consensus-based distributed particle filtering with distributed proposal adaptation. *IEEE Transactions on Signal Processing*, 62(12):3029–3041, 2014.

[18] F. Huszár and D. Duvenaud. Optimally-weighted herding is Bayesian quadrature. *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pages 377–386, 2012.

[19] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Transactions on Selected Areas in Communications*, 23(4):809–819, April 2005.

[20] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky. Efficient multiscale sampling from products of Gaussian Mixtures. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–8, 2004.

[21] S. J. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(2):401–422, March 2004.

[22] J. Kotecha and Petar M. Djurić. Gaussian particle filtering. *IEEE Transactions Signal Processing*, 51(10):2592–2601, October 2003.

[23] J. Kotecha and Petar M. Djurić. Gaussian sum particle filtering. *IEEE Transactions Signal Processing*, 51(10):2602–2612, October 2003.

[24] S. Lacoste-Julien, F. Lindsten, and F. Bach. Sequential kernel herding: Frank-Wolfe optimization for particle filtering. *In Proc. of the 18th International Conference on Artificial Intelligence and Statistics*, page 544552, 2015.

[25] T. Li, M. Bolic, and P. M. Djuric. Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, 2015.

[26] T. Li, T. P. Sattar, and S. Sun. Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters. *Signal Processing*, 92(7):1637–1645, 2012.

[27] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[28] P. LEcuyer. Efficiency improvement and variance reduction. *In Proceedings of the 1994 Winter Simulation Conference*, pages 122–132, 1994.

[29] S. Mak and V. R. Joseph. Projected support points: a new method for high-dimensional data reduction. *arXiv:1708.06897*, pages 1–48, 2018.

[30] S. Mak and V. R. Joseph. Support points. *(to appear) Annals of Statistics, arXiv:1609.01811*, pages 1–55, 2018.

[31] L. Martino, V. Elvira, and G. Camps-Valls. Group Importance Sampling for Particle Filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[32] L. Martino, V. Elvira, and F. Louzada. Weighting a resampled particle in Sequential Monte Carlo. *IEEE Statistical Signal Processing Workshop, (SSP)*, 122:1–5, 2016.

[33] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[34] L. Martino, V. P. Del Olmo, and J. Read. A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453, 2012.

[35] L. Martino, J. Read, V. Elvira, and F. Louzada. Cooperative parallel particle filters for on-line model selection and applications to urban mobility. *Digital Signal Processing*, 60:172–185, 2017.

[36] J. Míguez and M. A. Vázquez. A proof of uniform convergence over time for a distributed particle filter. *Signal Processing*, 122:152–163, 2016.

[37] A. Mohammadi and A. Asif. Diffusive particle filtering for distributed multisensor estimation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3801–3805, 2016.

[38] W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parallel MCMC. *arXiv:1311.4780*, 2013.

[39] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial Mathematics, 1992.

[40] B. N. Oreshkin and M. J. Coates. Asynchronous distributed particle filter via decentralized evaluation of gaussian products. *International Conference on Information Fusion*, pages 1–8, 2010.

[41] A. Owen. *Monte Carlo theory, methods and examples*. http://statweb.stanford.edu/∼owen/mc/, 2013.

[42] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Societe Franaise de Statistique*, 158(1):7–36, 2017.

[43] J. Read, K. Achutegui, and J. Míguez. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Processing*, 98:121 – 134, 2014.

[44] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[45] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[46] I. Schuster and I. Klebanov. Markov Chain Importance Sampling - a highly efficient estimator for MCMC. *arXiv:1805.07179*, pages 1 – 16, 2018.

[47] Steven L. Scott, Alexander W. Blocker, Fernando V. Bonassi, Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

[48] C. Verg, C. Dubarry, P. Del Moral, and E. Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260, 2015.

[49] J. R. Wilson. Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences*, 4(3):277–312, 1984.

[50] Y. Wu, D. Hu, M. Wu, and X. Hu. A numerical-integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, 54(8):2910–2921, 2006.

APPENDIX A
ZERO-LOSS COMPRESSION FOR A SPECIFIC INTEGRAL $I(h)$

Given a function $h(\mathbf{x})$, Theorem 1 states that, with the choice $s_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j)$ in (23), we have $\widetilde{I}^{(M)}(f) \equiv$

$\widehat{I}^{(N)}(h)$, when $f(\mathbf{x}) = \mathbf{x}$. Indeed, we have

$$\begin{aligned}\widetilde{I}^{(M)}(f) &= \sum_{m=1}^{M} \widehat{a}_m s_m \\ &= \sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) \right]\end{aligned}$$

and replacing $\bar{w}_{m,j} = \frac{\bar{w}_j}{\widehat{a}_m}$ given in Eq. (21), we obtain

$$\begin{aligned}\widetilde{I}^{(M)}(f) &= \sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j \in \mathcal{J}_m} \frac{\bar{w}_j}{\widehat{a}_m} h(\mathbf{x}_j) \right]. \\ &= \sum_{m=1}^{M} \sum_{j \in \mathcal{J}_m} \bar{w}_j h(\mathbf{x}_j) \\ &= \sum_{j=1}^{N} \bar{w}_j h(\mathbf{x}_j) = \widehat{I}^{(N)}(h), \end{aligned} \tag{73}$$

that is the desired result, given in Theorem 1.

APPENDIX B
DERIVATION OF $c_m(h)$

In this Appendix, the goal is to show that

$$\begin{aligned} c_m(h) &= \widehat{a}_m^2 \text{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}] \\ &= \sum_{i \in \mathcal{J}_m} \bar{w}_i \sum_{i \in \mathcal{J}_m} \bar{w}_i |h(\mathbf{x}_i)|^2 - \Big| \sum_{i \in \mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i) \Big|^2. \end{aligned} \tag{74}$$

First of all, we have

$$\text{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}] = \sum_{i \in \mathcal{J}_m} \bar{w}_{m,i} |h(\mathbf{x}_i)|^2 - \Big| \sum_{i \in \mathcal{J}_m} \bar{w}_{m,i} h(\mathbf{x}_i) \Big|^2,$$

and considering the expressions $\bar{w}_{m,j} = \frac{\bar{w}_j}{\widehat{a}_m}$ given in Eq. (21) and $\widehat{a}_m = \sum_{k \in \mathcal{J}_m} \bar{w}_k$ given in Eq. (20), we obtain

$$\text{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}] = \frac{\sum_{i \in \mathcal{J}_m} \bar{w}_i |h(\mathbf{x}_i)|^2}{\sum_{k \in \mathcal{J}_m} \bar{w}_k} - \frac{\Big| \sum_{i \in \mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i) \Big|^2}{\Big| \sum_{k \in \mathcal{J}_m} \bar{w}_k \Big|^2}. \tag{75}$$

Moreover, again since $\widehat{a}_m = \sum_{k \in \mathcal{J}_m} \bar{w}_k$ and replacing above, we can write

$$\begin{aligned} &\widehat{a}_m^2 \text{var}_{\widehat{\pi}_m}[h(\mathbf{s}_m)|\mathcal{S}] = \\ &= \Big| \sum_{k \in \mathcal{J}_m} \bar{w}_k \Big|^2 \frac{\sum_{i \in \mathcal{J}_m} \bar{w}_i |h(\mathbf{x}_i)|^2}{\sum_{k \in \mathcal{J}_m} \bar{w}_k} - \frac{\Big| \sum_{i \in \mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i) \Big|^2}{\Big| \sum_{k \in \mathcal{J}_m} \bar{w}_k \Big|^2}, \\ &= \sum_{k \in \mathcal{J}_m} \bar{w}_k \sum_{i \in \mathcal{J}_m} \bar{w}_i |h(\mathbf{x}_i)|^2 - \Big| \sum_{i \in \mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i) \Big|^2, \end{aligned}$$

that is exactly the expression in Eqs. (43) and (74).