Group Importance Sampling for Particle Filtering and MCMC

Luca Martino[⋆], Víctor Elvira[†], Gustau Camps-Valls[⋆]

[⋆] Image Processing Laboratory, Universitat de València (Spain).

[†] IMT Lille Douai CRISTAL (UMR 9189), Villeneuve d'Ascq (France).

Abstract

Importance Sampling (IS) is a well-known Monte Carlo technique that approximates integrals involving a posterior distribution by means of weighted samples. In this work, we study the assignation of a single weighted sample which compresses the information contained in a population of weighted samples. Part of the theory that we present as Group Importance Sampling (GIS) has been employed implicitly in different works in the literature. The provided analysis yields several theoretical and practical consequences. For instance, we discuss the application of GIS into the Sequential Importance Resampling framework and show that Independent Multiple Try Metropolis schemes can be interpreted as a standard Metropolis-Hastings algorithm, following the GIS approach. We also introduce two novel Markov Chain Monte Carlo (MCMC) techniques based on GIS. The first one, named Group Metropolis Sampling method, produces a Markov chain of sets of weighted samples. All these sets are then employed for obtaining a unique global estimator. The second one is the Distributed Particle Metropolis-Hastings technique, where different parallel particle filters are jointly used to drive an MCMC algorithm. Different resampled trajectories are compared and then tested with a proper acceptance probability. The novel schemes are tested in different numerical experiments such as learning the hyperparameters of Gaussian Processes, the localization problem in a wireless sensor network and the tracking of vegetation parameters given satellite observations, where they are compared with several benchmark Monte Carlo techniques. Three illustrative Matlab demos are also provided.

Keywords: Importance Sampling, Markov Chain Monte Carlo (MCMC), Particle Filtering, Particle Metropolis-Hastings, Multiple Try Metropolis, Bayesian Inference

1 Introduction

Monte Carlo methods are state-of-the-art tools for approximating complicated integrals involving multidimensional distributions, which is often required in science and technology [17, 23, 24, 39]. The most popular classes of MC methods are the Importance Sampling (IS) techniques and the Markov chain Monte Carlo (MCMC) algorithms [23, 39]. IS schemes produce a random discrete

approximation of the posterior distribution by a population of weighted samples [6, 28, 24, 39]. MCMC techniques generate a Markov chain (i.e., a sequence of correlated samples) with a pre-established target probability density function (pdf) as invariant density [23, 24].

In this work, we introduce theory and practice of a novel approach, called Group Importance Sampling (GIS), where the information contained in different sets of weighted samples is compressed by using only one, yet properly selected, particle, and one suitable weight. This general idea supports the validity of different Monte Carlo algorithms in the literature: interacting parallel particle filters [5, 34, 38], particle island schemes and related techniques [43, 44, 45], particle filters for model selection [15, 33, 42], nested Sequential Monte Carlo (SMC) methods [35, 36, 41] are some examples. We point out some consequences of the application of GIS in Sequential Importance Resampling (SIR) schemes, allowing partial resampling procedures and the use of different marginal likelihood estimators. Then, we show that the Independent Multiple Try Metropolis (I-MTM) techniques and the Particle Metropolis-Hastings (PMH) algorithm can be interpreted as a classical Independent Metropolis-Hastings method by the application of GIS.

Furthermore, we present two novel techniques based on GIS. The first one is the *Group Metropolis Sampling* (GMS) algorithm that generates a Markov chain of sets of weighted samples. All these resulting sets of samples are jointly exploited to obtain a unique particle approximation of the target distribution. On the one hand, GMS can be considered an MCMC method since it produces a Markov chain of sets of samples. On the other hand, the GMS can be also considered as an iterated importance sampler where different estimators are finally combined in order to build a unique IS estimator. This combination is obtained *dynamically* through random repetitions given by MCMC-type acceptance tests. GMS is closely related to Multiple Try Metropolis (MTM) techniques and Particle Metropolis-Hastings (PMH) algorithms [2, 3, 7, 11, 32, 30], as we discuss below. The GMS algorithm can be also seen as an extension of the method in [8], for recycling auxiliary samples in a MCMC method.

The second novel algorithm based on GIS is the Distributed PMH (DPMH) technique where the outputs of several parallel particle filters are compared by an MH-type acceptance function. The proper design of DPMH is a direct application of GIS. The benefit of DPMH is twofold: different type of particle filters (for instance, with different proposal densities) can be jointly employed, and the computational effort can be distributed in several machines speeding up the resulting algorithm. As the standard PMH method, DPMH is useful for filtering and smoothing the estimation of the trajectory of a variable of interest in a state-space model. Furthermore, the marginal version of DPMH can be used for the joint estimation of dynamic and static parameters. When the approximation of only one specific moment of the posterior is required, like GMS, the DPMH output can be expressed as a chain of IS estimators. The novel schemes are tested in three different numerical experiments: hyperparameter tuning for Gaussian Processes, localization in a sensor network and filtering of Leaf Area Index (LAI), a parameter widely used to monitor vegetation from satellite observations. The comparisons with other benchmark Monte Carlo methods show the benefits of the proposed algorithms.[1]

The remainder of the paper has the following structure. Section 2 recalls some background material. The basis of the GIS theory is introduced in Section 3. The applications of GIS in

[1]Three illustrative Matlab demos are also provided at `https://github.com/lukafree/GIS.git`.

particle filtering and Multiple Try Metropolis algorithms are discussed in Section 4. In Section 5, we introduce the novel techniques based on GIS. Section 6.1 provides the numerical results and in Section 7 we discuss some conclusions.

2 Problem statement and background

In many applications, the goal is to infer a variable of interest, $\mathbf{x} = x_{1:D} = [x_1, x_2, \ldots, x_D] \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$, where $x_d \in \mathbb{R}^{\xi}$ for all $d = 1, \ldots, D$, given a set of related observations or measurements, $\mathbf{y} \in \mathbb{R}^{d_Y}$. In the Bayesian framework all the statistical information is summarized by the posterior probability density function (pdf), i.e.,

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

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the marginal likelihood (a.k.a., Bayesian evidence). In general, $Z \equiv Z(\mathbf{y})$ is unknown and difficult to estimate in general, so we assume to be able to evaluate the unnormalized target function,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \tag{2}$$

The computation of integrals involving $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ is often intractable. We consider the Monte Carlo approximation of these complicated integrals involving the target $\bar{\pi}(\mathbf{x})$ and an integrable function $h(\mathbf{x})$ with respect to $\bar{\pi}$, i.e.,

$$I = E_{\bar{\pi}}[h(\mathbf{X})] = \int_{\mathcal{X}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, \tag{3}$$

where we denote $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$. The basic Monte Carlo (MC) procedure consists in drawing N independent samples from the target pdf, i.e., $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim \bar{\pi}(\mathbf{x})$, so that $\widehat{I}_N = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{x}_n)$ is an estimator of I [24, 39]. However, in general, direct methods for drawing samples from $\bar{\pi}(\mathbf{x})$ do not exist so that alternative procedures are required. Below, we describe perhaps one of the most popular.

2.1 Importance Sampling

Let us consider the use of a simpler proposal pdf, $q(\mathbf{x})$, and rewrite the integral I in Eq. (3) as

$$I = E_{\bar{\pi}}[h(\mathbf{X})] = E_q[h(\mathbf{X})w(\mathbf{X})] = \frac{1}{Z}\int_{\mathcal{X}} h(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}, \tag{4}$$

where $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})} : \mathcal{X} \to \mathbb{R}$. This suggests an alternative procedure. Indeed, we can draw N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from $q(\mathbf{x})$,[2] and then assign to each sample the unnormalized weights

$$w_n = w(\mathbf{x}_n) = \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}, \quad n = 1, \ldots, N. \tag{5}$$

[2] We assume that $q(\mathbf{x}) > 0$ for all $\mathbf{x}$ where $\bar{\pi}(\mathbf{x}) \neq 0$, and $q(\mathbf{x})$ has heavier tails than $\bar{\pi}(\mathbf{x})$.

If Z is known, an unbiased IS estimator [24, 39] is defined as $\widehat{I}_N = \frac{1}{ZN}\sum_{n=1}^{N} w_n h(\mathbf{x}_n)$, where $\mathbf{x}_n \sim q(\mathbf{x})$. If Z is unknown, defining the normalized weights, $\bar{w}_n = \frac{w_n}{\sum_{i=1}^{N} w_i}$ with $n = 1, \ldots, N$, an alternative asymptotically unbiased IS estimator is given by

$$\overline{I}_N = \sum_{n=1}^{N} \bar{w}_n h(\mathbf{x}_n). \tag{6}$$

Both $\widehat{I}_N$ and $\overline{I}_N$ are consistent estimators of I in Eq. (3) [24, 39]. Moreover, an unbiased estimator of marginal likelihood, $Z = \int_{\mathcal{X}} \pi(\mathbf{x}) d\mathbf{x}$, is given by $\widehat{Z} = \frac{1}{N}\sum_{i=1}^{N} w_i$. More generally, the pairs $\{\mathbf{x}_i, w_i\}_{i=1}^{N}$ can be used to build a particle approximation of the posterior distribution,

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

where $\delta(\mathbf{x})$ denotes the Dirac delta function. Table 1 summarizes the main notation of the work. Note that the words *sample* and *particle* are used as synonyms along this work. Moreover, Table 14 shows the main used acronyms.

2.2 Concept of proper weighting

The standard IS weights in Eq. (5) are broadly used in the literature. However the definition of *properly weighted sample* can be extended as suggested in [39, Section 14.2], [24, Section 2.5.4] and in [16]. More specifically, given a set of samples, they are properly weighted with respect to the target $\bar{\pi}$ if, for any integrable function h,

$$E_q[w(\mathbf{x}_n)h(\mathbf{x}_n)] = cE_{\bar{\pi}}[h(\mathbf{x}_n)], \quad \forall n \in \{1, \ldots, N\}, \tag{8}$$

where $c > 0$ is a constant value, also independent from the index n, and the expectation of the left hand side is performed, in general, w.r.t. to the joint pdf of $w(\mathbf{x})$ and $\mathbf{x}$, i.e., $q(w, \mathbf{x})$. Namely, the weight $w(\mathbf{x})$, conditioned to a given value of $\mathbf{x}$, could even be considered a random variable. Thus, in order to obtain consistent estimators, one can design any joint $q(w, \mathbf{x})$ as long as the restriction of Eq. (8) is fulfilled.

3 Group Importance Sampling: weighting a set of samples

In this section, we use the general definition in Eq. (8) for designing proper weights and summary samples assigned to different sets of samples. Let us consider M sets of weighted samples, $\mathcal{S}_1 = \{\mathbf{x}_{1,n}, w_{1,n}\}_{n=1}^{N_1}$, $\mathcal{S}_2 = \{\mathbf{x}_{2,n}, w_{2,n}\}_{n=1}^{N_2}$, ..., $\mathcal{S}_M = \{\mathbf{x}_{M,n}, w_{M,n}\}_{n=1}^{N_M}$, where $\mathbf{x}_{m,n} \sim q_m(\mathbf{x})$, i.e., a different proposal pdf for each set $\mathcal{S}_m$ and in general $N_i \neq N_j$, for all $i \neq j$, $i, j \in \{1, ..., M\}$. In some applications and different Monte Carlo schemes, it is convenient (and often required) to compress the statistical information contained in each set using a pair of summary sample, $\widetilde{\mathbf{x}}_m$, and summary weight, W_m, $m = 1, \ldots, M$, in such a way that the following expression

$$\widetilde{I}_M = \frac{1}{\sum_{j=1}^{M} W_j} \sum_{m=1}^{M} W_m h(\widetilde{\mathbf{x}}_m), \tag{9}$$

Table 1: Main notation of the work.

$\mathbf{x} = [x_1, \ldots, x_D]$	Variable of interest, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$, with $x_d \in \mathbb{R}^{\xi}$ for all d
$\bar{\pi}(\mathbf{x})$	Normalized posterior pdf, $\bar{\pi}(\mathbf{x}) = p(\mathbf{x}\|\mathbf{y})$
$\pi(\mathbf{x})$	Unnormalized posterior function, $\pi(\mathbf{x}) \propto \bar{\pi}(\mathbf{x})$
$\widehat{\pi}(\mathbf{x}\|\mathbf{x}_{1:N})$	Particle approximation of $\bar{\pi}(\mathbf{x})$ using the set of samples $\mathbf{x}_{1:N} = \{\mathbf{x}_n\}_{n=1}^N$
$\widetilde{\mathbf{x}}$	Resampled particle, $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}\|\mathbf{x}_{1:N})$ (note that $\widetilde{\mathbf{x}} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$)
$w_n = w(\mathbf{x}_n)$	Unnormalized standard IS weight of the particle $\mathbf{x}_n$
$\bar{w}_n = \bar{w}(\mathbf{x}_n)$	Normalized weight associated to $\mathbf{x}_n$
$\widetilde{w}_m = \widetilde{w}(\widetilde{\mathbf{x}}_m)$	Unnormalized proper weight associated to the resampled particle $\widetilde{\mathbf{x}}_m$
W_m	Summary weight of m-th set $\mathcal{S}_m$
$\overline{I}_N$	Standard self-normalized IS estimator using N samples
$\widetilde{I}_N$	Self-normalized estimator using N samples and based on GIS theory
Z	Marginal likelihood; normalizing constant of $\pi(\mathbf{x})$
$\widehat{Z}$, $\overline{Z}$	Estimators of the marginal likelihood Z

is still a consistent estimator of I, for a generic integrable function $h(\mathbf{x})$. Thus, although the compression is lossy, we still have a suitable particle approximation $\widehat{\pi}$ of the target $\bar{\pi}$ as shown below. In the following, we denote the importance weight of the n-th sample in the m-th group as $w_{m,n} = w(\mathbf{x}_{m,n}) = \frac{\pi(\mathbf{x}_{m,n})}{q_m(\mathbf{x}_{m,n})}$, the m-th marginal likelihood estimator as

$$\widehat{Z}_m = \frac{1}{N_m} \sum_{n=1}^{N_m} w_{m,n}, \tag{10}$$

and the normalized weights within a set, $\bar{w}_{m,n} = \frac{w_{m,n}}{\sum_{j=1}^{N_m} w_{m,j}} = \frac{w_{m,n}}{N_m \widehat{Z}_m}$, for $n = 1, \ldots, N_m$ and $m = 1, \ldots, M$.

Definition 1. *A resampled particle, i.e.,*

$$\widetilde{\mathbf{x}}_m \sim \widehat{\pi}_m(\mathbf{x}) = \widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N_m}) = \sum_{n=1}^{N_m} \bar{w}_{m,n} \delta(\mathbf{x} - \mathbf{x}_{m,n}), \tag{11}$$

is a summary particle $\widetilde{\mathbf{x}}_m$ for the m-group. Note that $\widetilde{\mathbf{x}}_m$ is selected within $\{\mathbf{x}_{m,1}, \ldots, \mathbf{x}_{m,N_m}\}$ according to the probability mass function (pmf) defined by $\bar{w}_{m,n}$, $n = 1, \ldots, N_m$.

It is possible to use the Liu's definition in order to assign a proper importance weight to a resampled particle [26], as stated in the following theorem.

Theorem 1. *Let us consider a resampled particle $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}_m(\mathbf{x}) = \widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N_m})$. A proper unnormalized weight following the Liu's definition in Eq.* (8) *for this resampled particle is $\widetilde{w}_m = \widehat{Z}_m$, defined in Eq.* (10)*.*

The proof is given in Appendix A. Note that two (or more) particles, $\widetilde{\mathbf{x}}'_m$, $\widetilde{\mathbf{x}}''_m$, resampled with

replacement from the same set and hence from the same approximation, $\widetilde{\mathbf{x}}'_m, \widetilde{\mathbf{x}}''_m \sim \widehat{\pi}_m(\mathbf{x})$, have the same weight $\widetilde{w}(\widetilde{\mathbf{x}}'_m) = \widetilde{w}(\widetilde{\mathbf{x}}''_m) = \widehat{Z}_m$, as depicted in Figure 1. Note that the classical importance weight cannot be computed for a resampled particle, as explained in Appendix A and pointed out in [22, 26, 35], [28, App. C1].

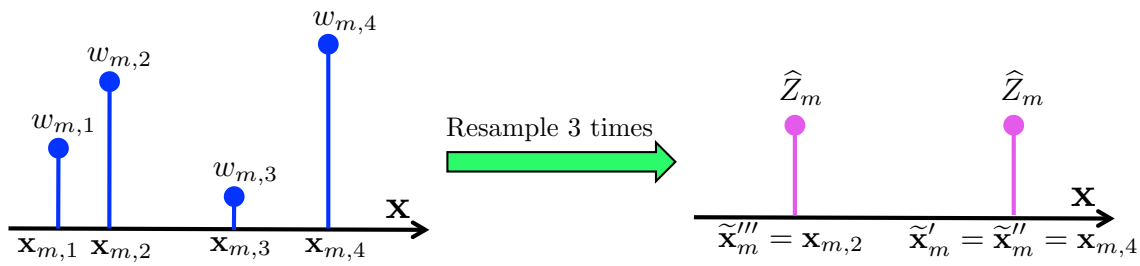

Figure 1: Example of generation (one run) and proper weighting of 3 resampled particles (with replacement), $\widetilde{\mathbf{x}}'_m$, $\widetilde{\mathbf{x}}''_m$ and $\widetilde{\mathbf{x}}'''_m$, from the m-th group, where $N_m = 4$ and $\widehat{Z}_m = \frac{1}{4}\sum_{n=1}^{4} w_{m,n}$.

Definition 2. *The summary weight for the m-th group of samples is $W_m = N_m \widetilde{w}_m = N_m \widehat{Z}_m$.*

Particle approximation. Figure 2 represents graphically an example of GIS with $M = 2$ and $N_1 = 4$, $N_2 = 3$. Given the M summary pairs $\{\widetilde{\mathbf{x}}_m, \widetilde{w}_m\}_{m=1}^{M}$ in a common computational node, we can obtain the following particle approximation of $\bar{\pi}(\mathbf{x})$, i.e.,

$$\widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_{1:M}) = \frac{1}{\sum_{j=1}^{M} N_j \widehat{Z}_j} \sum_{m=1}^{M} N_m \widehat{Z}_m \delta(\mathbf{x} - \widetilde{\mathbf{x}}_m), \tag{12}$$

involving M weighted samples in this case (see App. B). For a given function $h(\mathbf{x})$, the corresponding specific GIS estimator in Eq. (9) is

$$\widetilde{I}_M = \frac{1}{\sum_{j=1}^{M} N_j \widehat{Z}_j} \sum_{m=1}^{M} N_m \widehat{Z}_m h(\widetilde{\mathbf{x}}_m). \tag{13}$$

It is a consistent estimator of I, as we show in Appendix B. The expression in Eq. (13) can be interpreted as a standard IS estimator where $\widetilde{w}(\widetilde{\mathbf{x}}_m) = \widehat{Z}_m$ is a proper weight of a resampled particle [26], and we give more importance to the resampled particles belonging to a set with higher cardinality. See DEMO-2 at `https://github.com/lukafree/GIS.git`.

Combination of estimators. If we are only interested in computing the integral I for a specific function $h(\mathbf{x})$, we can summarize the statistical information by the pairs $\{\overline{I}_{N_m}^{(m)}, \widetilde{w}_m\}$ where

$$\overline{I}_{N_m}^{(m)} = \sum_{n=1}^{N_m} \bar{w}_{m,n} h(\mathbf{x}_{m,n}), \tag{14}$$

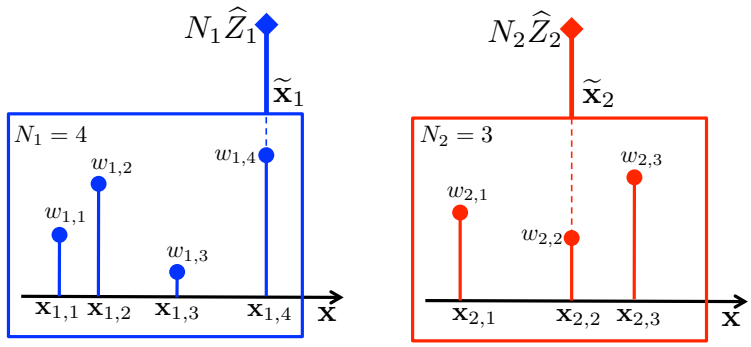

Figure 2: Graphical representation of GIS. In this case, $M = 2$ groups of $N_1 = 4$ and $N_2 = 3$ weighted samples are summarized with a resampled particle and one summary weight $\widetilde{w}_m = N_m\widehat{Z}_m$, $m = 1, 2$.

is the m-th partial IS estimator obtained by using N_m samples in $\mathcal{S}_m$. Given all the $S = \sum_{j=1}^{M} N_j$ weighted samples in the M sets, the complete estimator $\overline{I}_S$ in Eq. (6) can be written as a convex combination of the M partial IS estimators, $\overline{I}_{N_m}^{(m)}$, i.e.,

$$\overline{I}_S = \frac{1}{\sum_{j=1}^{M} N_j\widehat{Z}_j}\sum_{m=1}^{M}\sum_{n=1}^{N_m} w_{m,n}h(\mathbf{x}_{m,n}), \tag{15}$$

$$= \frac{1}{\sum_{j=1}^{M} N_j\widehat{Z}_j}\sum_{m=1}^{M} N_m\widehat{Z}_m\sum_{n=1}^{N_m} \bar{w}_{m,n}h(\mathbf{x}_{m,n}), \tag{16}$$

$$= \frac{1}{\sum_{j=1}^{M} W_m}\sum_{m=1}^{M} W_m\overline{I}_{N_m}^{(m)}. \tag{17}$$

The equation above shows that the summary weight W_m measures the importance of the m-th estimator $\overline{I}_{N_m}^{(m)}$. This confirms that W_m is a proper weight the group of samples $\mathcal{S}_m$, and also suggests another valid compression scheme.

Remark 1. *In order to approximate only one specific moment I of $\bar{\pi}(\mathbf{x})$, we can summarize the m-group with the pair $\{\overline{I}_{N_m}^{(m)}, W_m\}_{m=1}^{M}$, thus all the M partial estimators can be combined following Eq.* (17).

In this case, there is no loss of information w.r.t. storing all the weighted samples. However, the approximation of other moments of $\bar{\pi}(\mathbf{x})$ is not possible. Figures 3-4 depict the graphical representations of the two possible approaches for GIS.

4 GIS in other Monte Carlo schemes

4.1 Application in particle filtering

In Section 2.1, we have described the IS procedure in a batch way, i.e., generating directly a D-dimensional vector $\mathbf{x}' \sim q(\mathbf{x})$ and then compute the weight $\frac{\pi(\mathbf{x}')}{q(\mathbf{x}')}$. This procedure can be performed

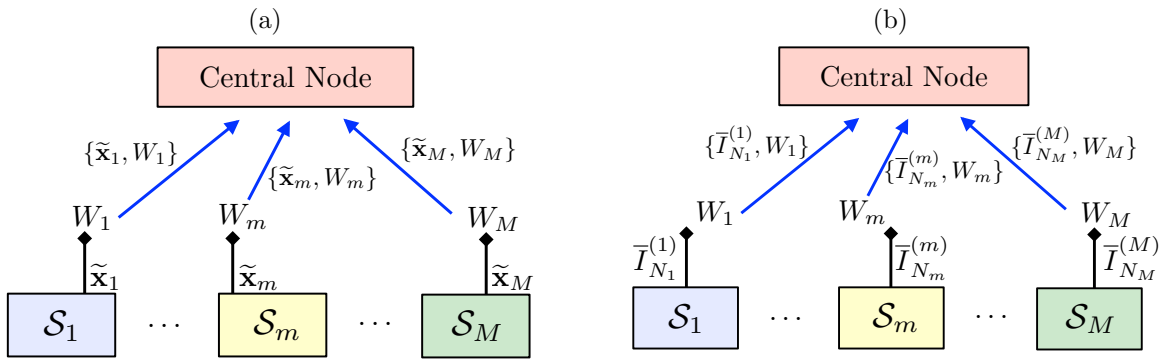

Figure 3: Graphical overview of GIS in a parallel/distributed framework. **(a)** The central node obtains all the pairs $\{\widetilde{\mathbf{x}}_m, W_m\}_{m=1}^M$, and provides $\widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_{1:M})$ or $\overline{I}_M$. Note that only M particles, $\widetilde{\mathbf{x}}_m \in \mathbb{R}^D$, and M scalar weights, $W_m \in \mathbb{R}$, are transmitted, instead of S samples and S weights, with $S = \sum_{m=1}^M N_m$. **(b)** Alternatively, if we are interested only in a specific moment of the target, we can transmit the pairs $\{\overline{I}^{(m)}_{N_m}, W_m\}_{m=1}^M$ and then combine them as in Eq. (17).

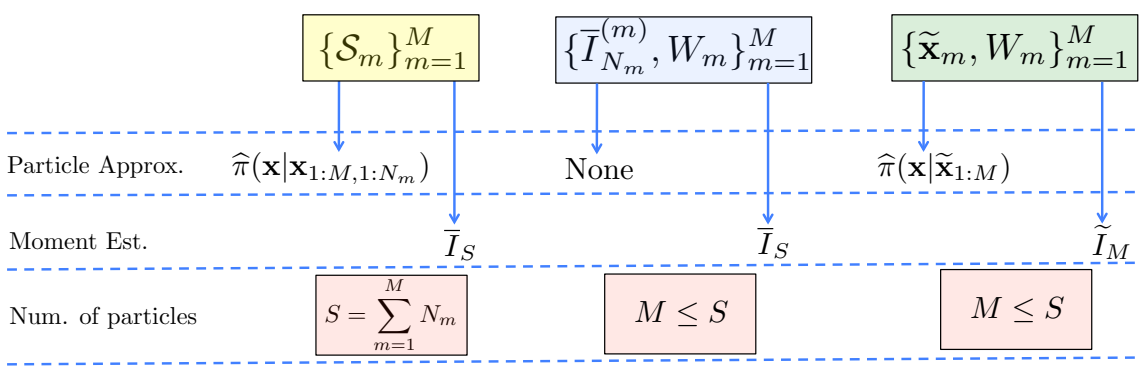

Figure 4: Possible outputs of different GIS compression schemes. On the left, $\{\mathcal{S}_m\}_{m=1}^M$, no compression is applied. In the center, $\{\overline{I}^{(m)}_{N_m}, W_m\}_{m=1}^M$, we can perfectly reconstruct the estimator $\overline{I}_S$ in Eq. (15) where $S = \sum_{m=1}^M N_m$, but we cannot approximate other moments. Using $\{\widetilde{\mathbf{x}}^{(m)}_{N_m}, W_m\}_{m=1}^M$, we always obtain a lossy compression, but any moments of $\bar{\pi}(\mathbf{x})$ can be approximated, as shown in Eqs. (12)-(13).

sequentially if the target density is factorized. In this case, the method is known as Sequential Importance Sampling (SIS). It is the basis of particle filtering, along with the use of the resampling procedure. Below, we describe the SIS method.

4.1.1 Sequential Importance Sampling (SIS)

Let us that recall $\mathbf{x} = x_{1:D} = [x_1, x_2, \ldots, x_D] \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$ where $x_d \in \mathbb{R}^{\xi}$ for all $d = 1, \ldots, D$, and let us consider a target pdf $\bar{\pi}(\mathbf{x})$ factorized as

$$\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x}) \quad = \quad \frac{1}{Z}\gamma_1(x_1)\prod_{d=2}^{D}\gamma_d(x_d|x_{1:d-1}), \tag{18}$$

where $\gamma_1(x_1)$ is a marginal pdf and $\gamma_d(x_d|x_{1:d-1})$ are conditional pdfs. We can also consider a proposal pdf decomposed in the same way, $q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{d-1})$. In a batch IS scheme, given the n-th sample $\mathbf{x}_n = x_{1:D}^{(n)} \sim q(\mathbf{x})$, we assign the importance weight

$$w(\mathbf{x}_n) \quad = \quad \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)} = \frac{\gamma_1(x_1^{(n)})\gamma_2(x_2^{(n)}|x_1^{(n)})\cdots\gamma_D(x_D^{(n)}|x_{1:D-1}^{(n)})}{q_1(x_1^{(n)})q_2(x_2^{(n)}|x_1^{(n)})\cdots q_D(x_D^{(n)}|x_{1:D-1}^{(n)})} = \prod_{d=1}^{D}\beta_d, \tag{19}$$

where $\beta_1^{(n)} = \frac{\pi(x_1^{(n)})}{q(x_1^{(n)})}$ and $\beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{1:d-1}^{(n)})}{q_d(x_d^{(n)}|x_{1:d-1}^{(n)})}$, with $d = 1, \ldots, D$. Let us also denote the joint probability of $[x_1, \ldots, x_d]$ as $\bar{\pi}_d(x_{1:d}) = \frac{1}{Z_d}\pi_d(x_{1:d}) = \frac{1}{Z_d}\gamma_1(x_1)\prod_{j=2}^{d}\gamma_j(x_j|x_{1:j-1})$, so that $\bar{\pi}_D(x_{1:D}) \equiv \bar{\pi}(\mathbf{x})$ and $Z_D \equiv Z$. Thus, we can draw samples generating sequentially each component $x_d^{(n)} \sim q_d(x_d|x_{1:d-1}^{(n)})$, $d = 1, \ldots, D$, so that $\mathbf{x}_n = x_{1:D}^{(n)} \sim q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{d-1})$, and compute recursively the corresponding IS weight as in Eq. (19). The SIS technique is also given in Table 2 by setting $\eta = 0$.

Remark 2. *In SIS, there are two possible formulations of the estimator of the marginal likelihoods* $Z_d = \int_{\mathbb{R}^{d\eta}} \pi_d(x_{1:d})dx_{1:d}$,

$$\widehat{Z}_d \quad = \quad \frac{1}{N}\sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N}\sum_{n=1}^{N} w_{d-1}^{(n)}\beta_d^{(n)}, \tag{20}$$

$$\overline{Z}_d \quad = \quad \prod_{j=1}^{d}\left[\sum_{n=1}^{N}\bar{w}_{j-1}^{(n)}\beta_j^{(n)}\right]. \tag{21}$$

In SIS, both estimators are equivalent $\overline{Z}_d \equiv \widehat{Z}_d$. *See Appendix C for further details.*

4.1.2 Sequential Importance Resampling (SIR)

The expression in Eq. (19) suggests a recursive procedure for generating the samples and computing the importance weights, as shown in Steps 2a and 2b of Table 2. In Sequential Importance Resampling (SIR), a.k.a., standard particle filtering, resampling steps are incorporated

during the recursion as in step 2(c)ii of Table 2 [12, 13]. In general, the resampling steps are applied only in certain iterations in order to avoid the path degeneration, taking into account an approximation $\widehat{ESS}$ of the Effective Sampling Size (ESS) [20, 27]. If $\widehat{ESS}$ is smaller than a pre-established threshold, the particles are resampled. Two examples of ESS approximation are $\widehat{ESS} = \frac{1}{\sum_{n=1}^{N}(\bar{w}_d^{(n)})^2}$ and $\widehat{ESS} = \frac{1}{\max \bar{w}_d^{(n)}}$ where $\bar{w}_d^{(n)} = \frac{w_d^{(n)}}{\sum_{i=1}^{N} w_d^{(i)}}$. Note that, in both cases,

$$0 \leq \widehat{ESS} \leq N. \tag{22}$$

Hence, the condition for the adaptive resampling can be expressed as $\widehat{ESS} < \eta N$ where $\eta \in [0, 1]$. SIS is given when $\eta = 0$ and SIR for $\eta \in (0, 1]$. When $\eta = 1$, the resampling is applied at each iteration and in this case SIR is often called *bootstrap particle filter* [12, 13, 14]. If $\eta = 0$, no resampling is applied, we only apply Steps 2a and 2b, and we have the SIS method described above, that after D iterations is completely equivalent to the bath IS approach, since $w_n = w(\mathbf{x}_n) \equiv w_D^{(n)}$ where $\mathbf{x}_n = x_{1:D}$.

Partial resampling. In Table 2, we have considered that only a *subset* of $R \leq N$ particles are resampled. In this case, step 2(c)iii including the GIS weighting is strictly required in order to provide final proper weighted samples and hence consistent estimators. The partial resampling procedure is an alternative approach to prevent the loss of particle diversity [26]. In the classical description of SIR [40], we have $R = N$ (i.e., all the particles are resampled) and the weight recursion follows setting the unnormalized weights of the resampled particles to any equal value. Since *all* the N particles have been resampled, the selection of this value has no impact in the weight recursion and in the estimation of I.

Marginal likelihood estimators. Even in the case $R = N$, i.e., all the particle are resampled as in the standard SIR method, without using the GIS weighting only the formulation $\overline{Z}_d$ in Eq. (21) provides a consistent estimator of Z_d, since it involves the normalized weights $\bar{w}_{d-1}^{(n)}$, instead of the unnormalized ones, $w_{d-1}^{(n)}$.

Remark 3. *If the GIS weighting is applied in SIR, both formulations $\widehat{Z}_d$ and $\overline{Z}_d$ in Eqs.* (20)-(21) *provide consistent estimator of Z_d and they are equivalent, $\widehat{Z}_d \equiv \overline{Z}_d$ (as in SIS). See an exhaustive discussion in Appendix C.*
See DEMO-1 at `https://github.com/lukafree/GIS.git`.

GIS in Sequential Monte Carlo (SMC). The idea of summary sample and summary weight have been implicitly used in different SMC schemes proposed in literature, for instance, for the communication among parallel particle filters [5, 34, 38], and in the particle island methods [43, 44, 45]. GIS also appears indirectly in particle filtering for model selection [15, 33, 42] and in the so-called Nested Sequential Monte Carlo techniques [35, 36, 41].

Table 2: SIR with partial resampling

1. Choose N the number of particles, $R \leq N$ the number of particles to be resampled, the initial particles $x_0^{(n)}$, $n = 1, \ldots, N$, an ESS approximation $\widehat{ESS}$ [27] and a constant value $\eta \in [0, 1]$.

2. For $d = 1, \ldots, D$:

   (a) **Propagation:** Draw $x_d^{(n)} \sim q_d(x_d | x_{1:d-1}^{(n)})$, for $n = 1, \ldots, N$.

   (b) **Weighting:** Compute the weights

$$w_d^{(n)} = w_{d-1}^{(n)} \beta_d^{(n)} = \prod_{j=1}^{d} \beta_j^{(n)}, \qquad n = 1, \ldots, N, \tag{23}$$

   where $\beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)} | x_{1:d-1}^{(n)})}{q_d(x_d^{(n)} | x_{1:d-1}^{(n)})}$.

   (c) if $\widehat{ESS} < \eta N$ then:

      i. Select randomly a set of particles $\mathcal{S} = \{x_d^{(j_r)}\}_{r=1}^R$ where $R \leq N$, $j_r \in \{1, \ldots, N\}$ for all r, and $j_r \neq j_k$ for $r \neq k$.

      ii. **Resampling:** Resample R times within the set $\mathcal{S}$ according to the probabilities $\bar{w}_d^{(j_r)} = \frac{w_d^{(j_r)}}{\sum_{k=1}^R w_d^{(j_k)}}$, obtaining $\{\bar{x}_d^{(j_r)}\}_{r=1}^R$. Then, set $x_d^{(j_r)} = \bar{x}_d^{(j_r)}$, for $r = 1, \ldots, R$.

      iii. **GIS weighting:** Compute $\widehat{Z}_{\mathcal{S}} = \frac{1}{R} \sum_{r=1}^{R} w_{d-1}^{(j_r)}$ and set $w_d^{(j_r)} = \widehat{Z}_{\mathcal{S}}$ for all $r = 1, \ldots, R$.

3. Return $\{\mathbf{x}_n = x_{1:D}^{(n)}, w_n = w_D^{(n)}\}_{n=1}^N$.

4.2 Multiple Try Metropolis schemes as a Standard Metropolis-Hastings method

The Metropolis-Hastings (MH) method is one of the most popular MCMC algorithm [23, 24, 39]. It generates a Markov chain $\{\mathbf{x}_t\}_{t=1}^{\infty}$ where $\bar{\pi}(\mathbf{x})$ is the invariant density. Considering a proposal pdf $q(\mathbf{x})$ independent from the previous state $\mathbf{x}_{t-1}$, the corresponding Independent MH (IMH) scheme is formed by the steps in Table 3 [39].

Observe that $\alpha(\mathbf{x}_{t-1}, \mathbf{v}') = \min\left[1, \frac{w(\mathbf{v}')}{w(\mathbf{x}_{t-1})}\right]$ in Eq. (24) involves the ratio between the importance weight of the proposed sample $\mathbf{v}'$ at the t-th iteration, and the importance weight of the previous state $\mathbf{x}_{t-1}$. Furthermore, note that at each iteration only one new sample $\mathbf{v}'$ is

Table 3: The Independent Metropolis-Hastings (IMH) algorithm

1. Choose an initial state $\mathbf{x}_0$.
2. For $t = 1, \ldots, T$:
   (a) Draw a sample $\mathbf{v}' \sim q(\mathbf{x})$.
   (b) Accept the new state, $\mathbf{x}_t = \mathbf{v}'$, with probability
$$\alpha(\mathbf{x}_{t-1}, \mathbf{v}') = \min\left[1, \frac{\pi(\mathbf{v}')q(\mathbf{x}_{t-1})}{\pi(\mathbf{x}_{t-1})q(\mathbf{v}')}\right] = \min\left[1, \frac{w(\mathbf{v}')}{w(\mathbf{x}_{t-1})}\right], \tag{24}$$
   where $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}$ (standard importance weight). Otherwise, set $\mathbf{x}_t = \mathbf{x}_{t-1}$.
3. Return $\{\mathbf{x}_t\}_{t=1}^T$.

generated and compared with the previous state $\mathbf{x}_{t-1}$ by the acceptance probability $\alpha(\mathbf{x}_{t-1}, \mathbf{v}')$ (in order to obtain the next state $\mathbf{x}_t$). The Particle Metropolis-Hastings (PMH) method[3] [2] and the alternative version of the Independent Multiply Try Metropolis technique [29] (denoted as I-MTM2) are jointly described in Table 4. They are two MCMC algorithms where at each iteration several candidates $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ are generated. After computing the IS weights $w(\mathbf{v}_n)$, one candidate is selected $\mathbf{v}_j$ within the N possible values, i.e., $j \in \{1, \ldots, N\}$, applying a resampling step according to the probability mass $\bar{w}_n = \frac{w(\mathbf{v}_n)}{\sum_{i=1}^N w(\mathbf{v}_i)} = \frac{w(\mathbf{v}_n)}{N\widehat{Z}'}$, $n = 1, \ldots, N$. Then the selected sample $\mathbf{v}_j$ is tested with a proper probability $\alpha(\mathbf{x}_{t-1}, \mathbf{v}_j)$ in Eq. (25).

The difference between PMH and I-MTM2 is the procedure employed for the generation of the N candidates and for the construction of the weights. PMH employs a sequential approach, whereas I-MTM2 uses a standard batch approach [29]. Namely, PMH generates sequentially the components $v_{j,k}$ of the candidates, $\mathbf{v}_j = [v_{j,1}, \ldots, v_{j,D}]^\top$, and compute recursively the weights as shown in Section 4.1. Since resampling steps are often used, then the resulting candidates $\mathbf{v}_1, \ldots, \mathbf{v}_N$ are correlated, whereas in I-MTM2 they are independent. I-MTM2 coincides with PMH if the candidates are generated sequentially but without applying resampling steps, so that I-MTM2 can be considered a special case of PMH.

Note that $\widetilde{w}(\mathbf{v}_j) = \widehat{Z}'$ and $\widetilde{w}(\mathbf{x}_{t-1}) = \widehat{Z}_{t-1}$ are the GIS weights of the resampled particles $\mathbf{v}_j$ and $\mathbf{x}_{t-1}$ respectively, as stated in Definition 1 and Theorem 1.[4] Hence, considering the GIS

[3]PMH is used for filtering and smoothing a variable of interest in state-space models (see, for instance, Figure 11). The *Particle Marginal MH* (PMMH) algorithm [2] is an extension of PMH employed in order to infer both dynamic and static variables. PMMH is described in Appendix D.

[4]Note that the number of candidates per iteration is constant (N), so that $\frac{W_t}{W_{t-1}} = \frac{N\widetilde{w}(\mathbf{v}_j)}{N\widetilde{w}(\mathbf{x}_{t-1})} = \frac{\widetilde{w}(\mathbf{v}_j)}{\widetilde{w}(\mathbf{x}_{t-1})}$.

Table 4: PMH and I-MTM2 techniques

1. Choose an initial state $\mathbf{x}_0$ and $\widehat{Z}_0$.
2. For $t = 1, \ldots, T$:
   (a) Draw N particles $\mathbf{v}_1, \ldots, \mathbf{v}_N$ from $q(\mathbf{x})$ and weight them with the proper importance weight $w(\mathbf{v}_n)$, $n = 1, \ldots, N$, using a sequential approach (PMH), or a batch approach (I-MTM2). Thus, denoting $\widehat{Z}' = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{v}_n)$, we obtain the particle approximation $\widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:N}) = \frac{1}{N\widehat{Z}'}\sum_{n=1}^{N} w(\mathbf{v}_n)\delta(\mathbf{x}-\mathbf{v}_n)$.
   (b) Draw $\mathbf{v}_j \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:N})$.
   (c) Set $\mathbf{x}_t = \mathbf{v}_j$ and $\widehat{Z}_t = \widehat{Z}'$, with probability
   $$\alpha(\mathbf{x}_{t-1}, \mathbf{v}_j) = \min\left[1, \frac{\widehat{Z}'}{\widehat{Z}_{t-1}}\right]. \tag{25}$$
   Otherwise, set $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $\widehat{Z}_t = \widehat{Z}_{t-1}$.
3. Return $\{\mathbf{x}_t\}_{t=1}^{T}$.

theory, we can write

$$\alpha(\mathbf{x}_{t-1}, \mathbf{v}_j) = \min\left[1, \frac{\widehat{Z}'}{\widehat{Z}_{t-1}}\right] = \min\left[1, \frac{\widetilde{w}(\mathbf{v}_j)}{\widetilde{w}(\mathbf{x}_{t-1})}\right], \tag{26}$$

which has the form of the acceptance function of the classical IMH method in Table 3. Therefore, PMH and I-MTM2 algorithms can be also summarized as in Table 5.

Remark 4. *The PMH and I-MTM2 algorithms take the form of the classical IMH method employing the equivalent proposal pdf $\widetilde{q}(\mathbf{x})$ in Eq. (27) (depicted in Figure 5; see also Appendix A), and using the GIS weight $\widetilde{w}(\widetilde{\mathbf{x}}')$ of a resampled particle $\widetilde{\mathbf{x}}' \sim \widetilde{q}(\mathbf{x})$, within the acceptance function $\alpha(\mathbf{x}_t, \widetilde{\mathbf{x}}')$.*

5 Novel MCMC techniques based on GIS

In this section, we provide two examples of novel MCMC algorithms based on GIS. First of all, we introduce a Metropolis-type method producing a chain of sets of weighted samples. Secondly, we present a PMH technique driven by M parallel particle filters. In the first scheme, we exploit the concept of summary weight and all the weighted samples are stored. In the second one, both concepts of summary weight and summary particle are used. The consistency of the resulting estimators and the ergodicity of both schemes is ensured.

Table 5: Alternative description of PMH and I-MTM2

1. Choose an initial state $\mathbf{x}_0$.
2. For $t = 1, \ldots, T$:
   (a) Draw $\widetilde{\mathbf{x}}' \sim \widetilde{q}(\mathbf{x})$, where
   $$\widetilde{q}(\mathbf{x}) = \int_{\mathcal{X}^N} \left[\prod_{i=1}^{N} q(\mathbf{v}_i) \right] \widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:N}) d\mathbf{v}_{1:N}, \tag{27}$$
   is the equivalent proposal pdf associated to a resampled particle [22, 26].
   (b) Set $\mathbf{x}_t = \widetilde{\mathbf{x}}'$, with probability
   $$\alpha(\mathbf{x}_{t-1}, \widetilde{\mathbf{x}}') = \min \left[1, \frac{\widetilde{w}(\widetilde{\mathbf{x}}')}{\widetilde{w}(\mathbf{x}_{t-1})} \right]. \tag{28}$$
   Otherwise, set $\mathbf{x}_t = \mathbf{x}_{t-1}$.
3. Return $\{\mathbf{x}_t\}_{t=1}^{T}$.

5.1 Group Metropolis Sampling

Here, we describe an MCMC procedure that yields a sequence of sets of weighted samples. All the samples are then employed for a joint particle approximation of the target distribution. The Group Metropolis Sampling (GMS) is outlined in Table 6. Figures 6(a)-(b) give two graphical representations of GMS outputs (with $N = 4$ in both cases). Note that the GMS algorithm uses the idea of summary weight for comparing sets. Let us denote as $\rho_{n,t}$ the importance weights assigned to the samples $\mathbf{x}_{n,t}$ contained in the current set $\mathcal{S}_t$. Given the generated sets $\mathcal{S}_t = \{\mathbf{x}_{n,t}, \rho_{n,t}\}_{n=1}^{N}$, for $t = 1, \ldots, T$, GMS provides the global particle approximation

$$\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N,1:T}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}} \delta(\mathbf{x} - \mathbf{x}_{n,t}), \tag{29}$$

$$= \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \bar{\rho}_{n,t} \delta(\mathbf{x} - \mathbf{x}_{n,t}). \tag{30}$$

Thus, the estimator of a specific moment of the target is

$$\widetilde{I}_{NT} = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \bar{\rho}_{n,t} h(\mathbf{x}_{n,t}) = \frac{1}{T} \sum_{t=1}^{T} \widetilde{I}_N^{(t)}. \tag{31}$$

If the N candidates, $\mathbf{v}_1, \ldots, \mathbf{v}_N$, and the associated weights, $w_1, \ldots, w_N$, are built sequentially by a particle filtering method, we have a Particle GMS (PGMS) algorithm (see Section 6.3) and

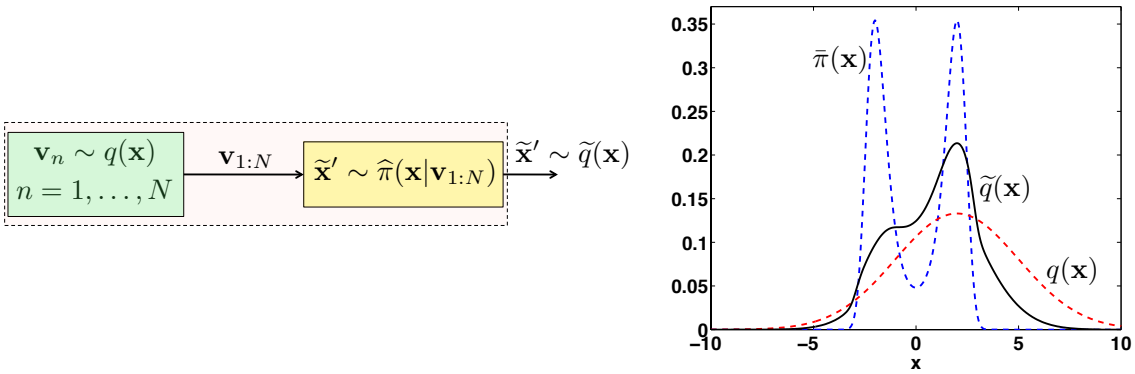

Figure 5: (Left) Graphical representation of the generation of one sample $\mathbf{x}'$ from the equivalent proposal pdf $\widetilde{q}(\mathbf{x})$ in Eq. (27).**(Right)** Example of the equivalent density $\widetilde{q}(\mathbf{x})$ (solid line) with $N = 2$. The target, $\bar{\pi}(\mathbf{x})$, and proposal, $q(\mathbf{x})$, pdfs are shown with dashed lines. See DEMO-3 at `https://github.com/lukafree/GIS.git`.

marginal versions can be also considered (see Appendix D).

Table 6: Group Metropolis Sampling

1. Build an initial set $\mathcal{S}_0 = \{\mathbf{x}_{n,0}, \rho_{n,0}\}_{n=1}^N$ and $\widehat{Z}_0 = \frac{1}{N}\sum_{n=1}^N \rho_{n,0}$.
2. For $t = 1, \ldots, T$:
   (a) Draw N samples, $\mathbf{v}_1, \ldots, \mathbf{v}_N \sim q(\mathbf{x})$ following a sequential or a batch procedure.
   (b) Compute the weights, $w_n = \frac{\pi(\mathbf{v}_n)}{q(\mathbf{v}_n)}$, $n = 1, \ldots, N$; define $\mathcal{S}' = \{\mathbf{v}_n, w_n\}_{n=1}^N$; and compute $\widehat{Z}' = \frac{1}{N}\sum_{n=1}^N w_n$.
   (c) Set $\mathcal{S}_t = \{\mathbf{x}_{n,t} = \mathbf{v}_n, \rho_{n,t} = w_n\}_{n=1}^N$, i.e., $\mathcal{S}_t = \mathcal{S}'$, and $\widehat{Z}_t = \widehat{Z}'$, with probability
   $$\alpha(\mathcal{S}_{t-1}, \mathcal{S}') = \min\left[1, \frac{\widehat{Z}'}{\widehat{Z}_{t-1}}\right]. \tag{32}$$
   Otherwise, set $\mathcal{S}_t = \mathcal{S}_{t-1}$ and $\widehat{Z}_t = \widehat{Z}_{t-1}$.
3. Return $\{\mathcal{S}_t\}_{t=1}^T$, or $\{\widetilde{I}_N^{(t)}\}_{t=1}^T$ where $\widetilde{I}_N^{(t)} = \sum_{n=1}^N \frac{\rho_{n,t}}{\sum_{i=1}^N \rho_{i,t}} h(\mathbf{x}_{n,t})$.

Relation with IMH. The acceptance probability α in Eq. (32) is the extension of the acceptance probability of IMH in Eq. (24), by considering the proper GIS weighting of a set of weighted samples. Note that, in this version of GMS, all the sets contain the same number of

samples.

Relation with MTM methods. GMS is strictly related to Multiple Try Metropolis (MTM) schemes [7, 31, 32, 29] and Particle Metropolis Hastings (PMH) techniques [2, 29]. The main difference is that GMS uses no resampling steps at each iteration for generating summary samples, indeed GMS uses the entire set. However, considering a sequential of a batch procedure for generating the N tries at each iteration, we can recover an MTM (or the PMH) chain by the GMS output applying one resampling step when $\mathcal{S}_t \neq \mathcal{S}_{t-1}$,

$$\widetilde{\mathbf{x}}_t = \begin{cases} \widetilde{\mathbf{v}}_t \sim \sum_{n=1}^{N} \bar{\rho}_{n,t}\delta(\mathbf{x}-\mathbf{x}_{n,t}), & \text{if} \quad \mathcal{S}_t \neq \mathcal{S}_{t-1}, \\ \widetilde{\mathbf{x}}_{t-1}, & \text{if} \quad \mathcal{S}_t = \mathcal{S}_{t-1}, \end{cases} \tag{33}$$

for $t = 1, \ldots, T$. Namely, $\{\widetilde{\mathbf{x}}_t\}_{t=1}^T$ is the chain obtained by one run of the MTM (or PMH) technique. Figure 6(b) provides a graphical representation of a MTM chain recovered by GMS outputs.

Ergodicity. As also discussed above, (a) the sample generation, (b) the acceptance probability function and hence (c) the dynamics of GMS exactly coincide with the corresponding steps of PMH or MTM (with a sequential or batch particle generation, respectively). Hence, the ergodicity of the chain is ensured [7, 32, 2, 29]. Indeed, we can recover the MTM (or PMH) chain as shown in Eq. (33).

Recycling samples. The GMS algorithm can be seen as a method of recycling auxiliary weighted samples in MTM schemes (or PMH schemes, if the candidates are generated by SIR). In [8], the authors show how recycling and including the samples rejected in one run of a standard MH method into a unique consistent estimator. GMS can be considered an extension of this technique where $N \geq 1$ candidates are drawn at each iteration.

Iterated IS. GMS can be also interpreted as an iterative importance sampling scheme where an IS approximation of N samples is built at each iteration and compared with the previous IS approximation. This procedure is iterated T times and all the accepted IS estimators $\widetilde{I}_N^{(t)}$ are finally combined to provide a unique global approximation of NT samples. Note that the temporal combination of the IS estimators is obtained dynamically by the random repetitions due to the rejections in the MH test. Hence, the complete procedure for weighting the samples generated by GMS can be interpreted as the composition of two weighting schemes: (a) by an importance sampling approach building $\{\rho_{n,t}\}_{n=1}^N$ and (b) by the possible random repetitions due to the rejections in the MH test.

Consistency of the GMS estimator. Recovering the MTM chain $\{\widetilde{\mathbf{x}}_t\}_{t=1}^T$ as in Eq. (33), the estimator $\widetilde{I}_T = \frac{1}{T}\sum_{t=1}^T h(\widetilde{\mathbf{x}}_t)$ is consistent, i.e., $\widetilde{I}_T$ converges almost-surely to $I = E_{\bar{\pi}}[h(\mathbf{x})]$ as $T \to \infty$, since $\{\widetilde{\mathbf{x}}_t\}_{t=1}^T$ is an ergodic chain [39]. Note that $E_{\widehat{\pi}}[h(\widetilde{\mathbf{x}}_t)|\mathcal{S}_t] = \sum_{n=1}^N \bar{\rho}_{n,t} h(\mathbf{x}_{n,t}) = \widetilde{I}_N^{(t)}$

for $\mathcal{S}_t \neq \mathcal{S}_{t-1}$, where $\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N,t}) = \sum_{n=1}^{N} \bar{\rho}_{n,t}\delta(\mathbf{x}-\mathbf{x}_{n,t})$. If $\mathcal{S}_t = \mathcal{S}_{t-1}$, then $E_{\widehat{\pi}}[h(\widetilde{\mathbf{x}}_t)|\mathcal{S}_t] = E_{\widehat{\pi}}[h(\widetilde{\mathbf{x}}_{t-1})|\mathcal{S}_{t-1}] = \widetilde{I}_N^{(t-1)}$ and, since $\widetilde{I}_N^{(t)} = \widetilde{I}_N^{(t-1)}$, we have again $E_{\widehat{\pi}}[h(\widetilde{\mathbf{x}}_t)|\mathcal{S}_t] = \widetilde{I}_N^{(t)}$. Therefore, we have

$$E[\widetilde{I}_T|\mathcal{S}_{1:T}] = \frac{1}{T}\sum_{t=1}^{T} E_{\widehat{\pi}}[h(\widetilde{\mathbf{x}}_t)|\mathcal{S}_t] \tag{34}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \widetilde{I}_N^{(t)} = \widetilde{I}_{NT}. \tag{35}$$

Thus, the GSM estimator $\widetilde{I}_{NT}$ in Eq. (31) can be expressed as $\widetilde{I}_{NT} = E[\widetilde{I}_T|\mathcal{S}_{1:T}]$, where $\mathcal{S}_{1:T}$ are all the weighted samples obtained by GMS, and it is consistent for $T \to \infty$ since $\widetilde{I}_T$ is consistent. Furthermore, fixing T, $\widetilde{I}_{NT}$ is consistent for $N \to \infty$ for standard IS arguments [24].

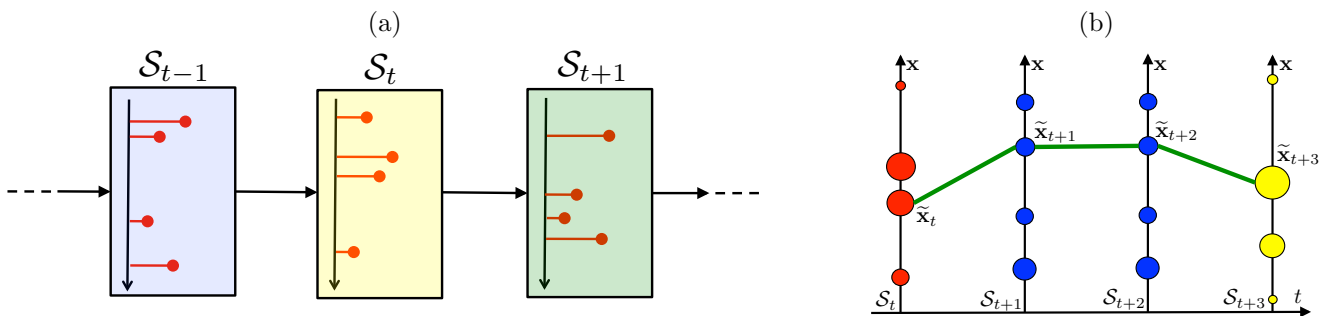

Figure 6: (a) Chain of sets $\mathcal{S}_t = \{\mathbf{x}_{n,t}, \rho_{n,t}\}_{n=1}^{N}$ generated by the GMS method (graphical representation with $N = 4$). **(b)** Graphical examples of GMS outputs, $\mathcal{S}_t$, $\mathcal{S}_{t+1}$, $\mathcal{S}_{t+2}$ and $\mathcal{S}_{t+3}$, where $\mathcal{S}_{t+2} = \mathcal{S}_{t+1}$. The weights of the samples are denoted by the size of the circles. A possible recovered MTM chain is also depicted with solid line, where the states are $\widetilde{\mathbf{x}}_\tau$ with $\tau = t, t+1, t+2, t+3$ and $\widetilde{\mathbf{x}}_{t+2} = \widetilde{\mathbf{x}}_{t+1}$.

5.2 Distributed Particle Metropolis-Hastings algorithm

The PMH algorithm is an MCMC technique particularly designed for filtering and smoothing a dynamic variable in a state-space model [2, 29] (see for instance Figure 11). In PMH, different trajectories obtained by different runs of a particle filter (see Section 4.1) are compared according to suitable MH-type acceptance probabilities, as shown in Table 4. In this section, we show how several parallel particle filters (for instance, each one consider a different proposal pdf) can drive a PMH-type technique.

The classical PMH method uses a single proposal pdf $q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{1:d-1})$, employed in single SIR method in order to generate new candidates before of the MH-type test (see Table 4). Let us consider the problem of tracking a variable of interest $\mathbf{x} = [x_1, \ldots, x_D]^\top \in \mathbb{R}^{D\times\xi}$ with target pdf $\pi(\mathbf{x}) = \pi_1(x_1)\prod_{d=2}^{D} \pi_d(x_d|x_{1:d-1})$. We assume that M independent processing units are available jointly with a central node as shown Fig. 7. We use M parallel particle filters, each one with a different proposal pdf, $q_m(\mathbf{x}) = q_{m,1}(x_1)\prod_{d=2}^{D} q_{m,d}(x_d|x_{1:d-1})$, one per each processor. Then,

after one run of the parallel particle filters, we obtain M particle approximations $\widehat{\pi}_m(\mathbf{x})$. Since, we aim to reduce the communication cost to the central node (see Figs. 3 and 7), we consider that each machine only transmits the pair $\{\widehat{Z}_m, \widetilde{\mathbf{x}}_m\}$, where $\widetilde{x}_m \sim \widehat{\pi}_m(\mathbf{x})$ (we set $N_1 = \ldots = N_M$, for simplicity). Applying the GIS theory, then it is straightforward to outline the method, called *Distributed Particle Metropolis-Hastings* (DPMH) technique, shown in Table 7.

Table 7: Distributed Particle Metropolis-Hastings algorithm

1. Choose an initial state $\mathbf{x}_0$ and $\widehat{Z}_{m,0}$ for $m = 1, \ldots, M$.
2. For $t = 1, \ldots, T$:
   (a) **(Parallel Processors)** Draw N particles $\mathbf{v}_{m,1}, \ldots, \mathbf{v}_{m,N}$ from $q_m(\mathbf{x})$ and weight them with IS weights $w(\mathbf{v}_{m,n})$, $n = 1, \ldots, N$, using a particle filter (or a batch approach), for each $m = 1, \ldots, M$. Thus, denoting $\widehat{Z}_m = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{v}_{m,n})$, we obtain the M particle approximations $\widehat{\pi}_m(\mathbf{x}) = \widehat{\pi}(\mathbf{x}|\mathbf{v}_{m,1:N}) = \frac{1}{N\widehat{Z}_m}\sum_{n=1}^{N} w(\mathbf{v}_{m,n})\delta(\mathbf{x} - \mathbf{v}_{m,n})$.
   (b) **(Parallel Processors)** Draw $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_{m,1:N})$, for $m = 1, \ldots, M$.
   (c) **(Central Node)** Resample $\widetilde{\mathbf{x}} \in \{\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_M\}$ according to the pmf $\frac{\widehat{Z}_m}{\sum_{j=1}^{M}\widehat{Z}_j}$, $m = 1, \ldots, M$, i.e., $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_M)$.
   (d) **(Central Node)** Set $\mathbf{x}_t = \widetilde{\mathbf{x}}$ and $\widehat{Z}_{m,t} = \widehat{Z}_m$, for $m = 1, \ldots, M$, with probability

$$\alpha(\mathbf{x}_{t-1}, \widetilde{\mathbf{x}}) = \min\left[1, \frac{\sum_{m=1}^{M}\widehat{Z}_m}{\sum_{m=1}^{M}\widehat{Z}_{m,t-1}}\right]. \tag{36}$$

   Otherwise, set $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $\widehat{Z}_{m,t} = \widehat{Z}_{m,t-1}$, for $m = 1, \ldots, M$.

The method in Table 7 has the structure of a Multiple Try Metropolis (MTM) algorithm using different proposal pdfs [7, 32]. More generally, in step 2a, the scheme described above can even employ different kinds of particle filtering algorithms. In step 2b, M total resampling steps are performed,one per processor. Then, one resampling step is performed in the central node (step 2c). Finally, the resampled particle is accepted as new state with probability α in Eq. (36).

Ergodicity. The ergodicity of DPMH is ensured since it can be interpreted as a standard PMH method considering a single particle approximation[5]

$$\widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:M,1:N}) = \sum_{m=1}^{M} \frac{\widehat{Z}_m}{\sum_{j=1}^{M} \widehat{Z}_j} \widehat{\pi}(\mathbf{x}|\mathbf{v}_{m,1:N}) = \sum_{m=1}^{M} \overline{W}_m \widehat{\pi}(\mathbf{x}|\mathbf{v}_{m,1:N}), \tag{37}$$

and then we resample once, i.e., draw $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:M,1:N})$. Then, the proper weight of this resampled particle is $\widehat{Z} = \sum_{m=1}^{M} \widehat{Z}_m$, so that the acceptance function of the equivalent classical PMH method is $\alpha(\mathbf{x}_{t-1}, \widetilde{\mathbf{x}}) = \min\left[1, \frac{\widehat{Z}}{\widehat{Z}_{t-1}}\right] = \min\left[1, \frac{\frac{1}{M}\sum_{m=1}^{M} \widehat{Z}_m}{\frac{1}{M}\sum_{m=1}^{M} \widehat{Z}_{m,t-1}}\right]$, where $\widehat{Z}_{t-1} = \sum_{m=1}^{M} \widehat{Z}_{m,t-1}$ (see Table 4).

Using partial IS estimators. if we are interested in approximating only one moment of the target pdf, as shown in Figures 3-4, at each iteration we can transmit the M partial estimators $\overline{I}_N^{(m)}$ and combine them in the central node as in Eq. (17), obtaining $\widetilde{I}'_{NM} = \frac{1}{\sum_{j=1}^{M} \widehat{Z}_j} \sum_{m=1}^{M} \widehat{Z}_m \overline{I}_N^{(m)}$. Then, a sequence of estimators, $\widetilde{I}_{NM}^{(t)}$, is created according to the acceptance probability α in Eq. (36). Finally, we obtain the global estimator

$$\widetilde{I}_{NMT} = \frac{1}{T} \sum_{t=1}^{T} \widetilde{I}_{NM}^{(t)}. \tag{38}$$

This scheme is depicted in Figure 7(b).

Benefits. The main advantage of the DPMH scheme is that the generation of samples can be parallelized (i.e., fixing the computational cost, DPMH allows the use of M processors in parallel) and the communication to the central node requires the transfer of only M particles, $\widetilde{\mathbf{x}}'_m$, and M weights, $\widehat{Z}'_m$, instead of NM particles and NM weights. Figure 7 provides a general sketch of DPMH. Its marginal version is described in Appendix D. Another benefit of DPMH is that different types of particle filters can be jointly employed, for instance, different proposal pdfs can be used.

Special cases and extensions. The classical PMH method is as a special case of the proposed algorithm of Table 7 when $M = 1$. If the partial estimators are transmitted to the central node, as shown in Figure 7(b), DPMH coincides with PGMS when $M = 1$. Adaptive versions of DPMH can be designed in order select automatically the best proposal pdf among the M densities, based of the weights $\overline{W}_m = \frac{\widehat{Z}_m}{\sum_{j=1}^{M} \widehat{Z}_j}$, $m = 1, \ldots, M$. For instance, Figure 12(b) shows that DPMH is able to detect the best scale parameters within the M used values.

[5]This particle approximation can be interpreted as being obtained by a single particle filter splitting the particles in M disjoint sets and then applying the partial resampling described in Section 4.1, i.e., performing resampling steps within the sets. See also Eq. (56).

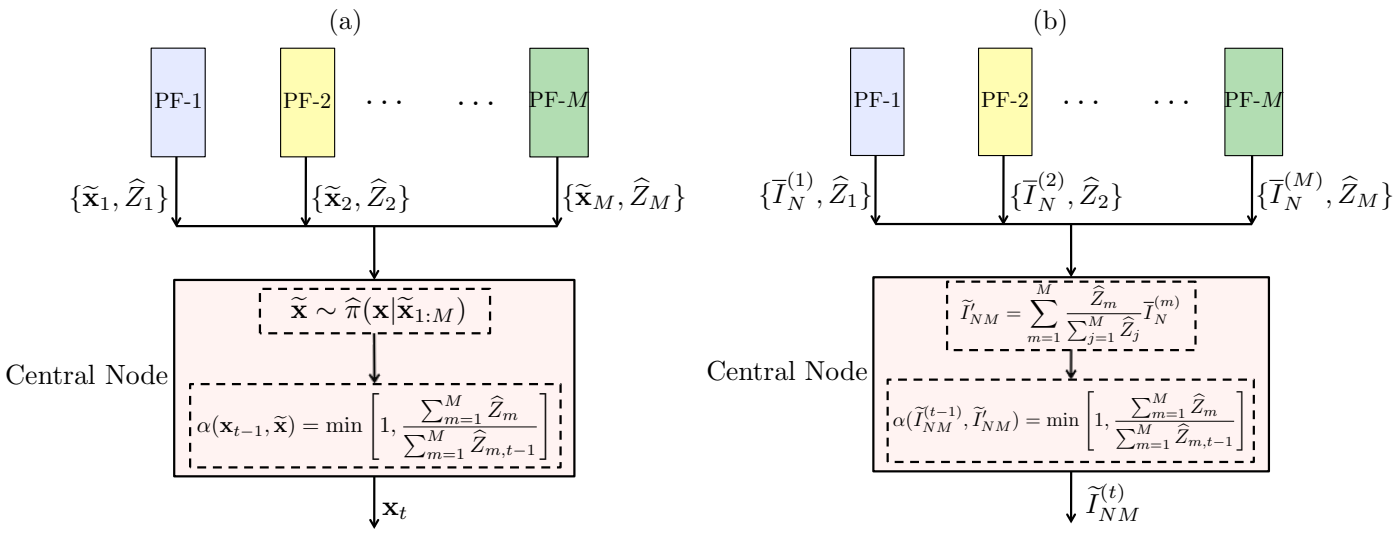

Figure 7: Graphical representation of Distributed Particle Metropolis-Hastings (DPMH) method, **(a)** for estimating a generic moment, or **(b)** for estimating of a specific moment of the target.

6 Numerical Experiments

In this section, we test the novel techniques considering several experimental scenarios and three different applications: hyperparameters estimation for Gaussian Processes ($D = 2$, $\xi = 1$), a localization problem jointly with the tuning of the sensor network ($D = 8$, $\xi = 1$), and the online filtering of a remote sensing variable called Leaf Area Index (LAI; $D = 365$, $\xi = 1$). We compare the novel algorithms with different benchmark methods such adaptive MH algorithm, MTM and PMH techniques, parallel MH chains with random walk proposal pdfs, IS schemes, and the Adaptive Multiple Importance Sampling (AMIS) method.

6.1 Hyperparameter tuning for Gaussian Process (GP) regression models

We test the proposed GMS approach for the estimation of hyperparameters of a Gaussian process (GP) regression model [4], [37]. Let us assume observed data pairs $\{y_j, \mathbf{z}_j\}_{j=1}^{P}$, with $y_j \in \mathbb{R}$ and $\mathbf{z}_j \in \mathbb{R}^L$. We also denote the corresponding $P \times 1$ output vector as $\mathbf{y} = [y_1, \ldots, y_P]^\top$ and the $L \times P$ input matrix as $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_P]$. We address the regression problem of inferring the unknown function f which links the variable y and $\mathbf{z}$. Thus, the assumed model is $y = f(\mathbf{z}) + e$, where $e \sim N(e; 0, \sigma^2)$, and that $f(\mathbf{z})$ is a realization of a GP [37]. Hence $f(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{r}))$ where $\mu(\mathbf{z}) = 0$, $\mathbf{z}, \mathbf{r} \in \mathbb{R}^L$, and we consider the kernel function

$$\kappa(\mathbf{z}, \mathbf{r}) = \exp\left(-\sum_{\ell=1}^{L} \frac{(z_\ell - r_\ell)^2}{2\delta^2}\right). \tag{39}$$

Given these assumptions, the vector $\mathbf{f} = [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_P)]^\top$ is distributed as $p(\mathbf{f}|\mathbf{Z}, \delta, \kappa) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K})$, where $\mathbf{0}$ is a $P \times 1$ null vector, and $\mathbf{K}_{ij} := \kappa(\mathbf{z}_i, \mathbf{z}_j)$, for all $i, j = 1, \ldots, P$, is a $P \times P$ matrix. Therefore, the vector containing all the hyperparameters of the model is $\mathbf{x} = [\delta, \sigma]$, i.e., all the parameters of the kernel function in Eq. (39) and standard deviation σ of the observation noise. In this experiment, we focus on the marginal posterior density of the hyperparameters, $\bar{\pi}(\mathbf{x}|\mathbf{y}, \mathbf{Z}, \kappa) \propto \pi(\mathbf{x}|\mathbf{y}, \mathbf{Z}, \kappa) = p(\mathbf{y}|\mathbf{x}, \mathbf{Z}, \kappa)p(\mathbf{x})$, which can be evaluated analytically, but we cannot compute integrals involving it [37]. Considering a uniform prior within $[0, 20]^2$, $p(\mathbf{x})$ and since $p(\mathbf{y}|\mathbf{x}, \mathbf{Z}, \kappa) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$, we have

$$\log\left[\pi(\mathbf{x}|\mathbf{y}, \mathbf{Z}, \kappa)\right] = -\frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log\left[\det\left(\mathbf{K} + \sigma^2\mathbf{I}\right)\right],$$

where clearly $\mathbf{K}$ depends on δ [37]. The moments of this marginal posterior cannot be computed analytically. Then, in order to compute the Minimum Mean Square Error (MMSE) estimator $\widehat{\mathbf{x}} = [\widehat{\delta}, \widehat{\sigma}]$, i.e., the expected value $E[\mathbf{X}]$ with $\mathbf{X} \sim \bar{\pi}(\mathbf{x}|\mathbf{y}, \mathbf{Z}, \kappa)$, we approximate $E[\mathbf{X}]$ via Monte Carlo quadrature. More specifically, we apply a the novel GMS technique and compare with an MTM sampler, a MH scheme with a longer chain and a static IS method. For all these methodologies, we consider the same number of target evaluations, denoted as E, in order to provide a fair comparison.

We generated $P = 200$ pairs of data, $\{y_j, \mathbf{z}_j\}_{j=1}^P$, according to the GP model above setting $\delta^* = 3$, $\sigma^* = 10$, $L = 1$, and drawing $z_j \sim \mathcal{U}([0, 10])$. We keep fixed these data over the different runs, and the corresponding posterior pdf is given in Figure 9(b). We computed the ground-truth $\widehat{\mathbf{x}} = [\widehat{\delta} = 3.5200, \widehat{\sigma} = 9.2811]$ using an exhaustive and costly grid approximation, in order to compare the different techniques. For both GMS, MTM and MH schemes, we consider the same adaptive Gaussian proposal pdf $q_t(\mathbf{x}|\boldsymbol{\mu}_t, \lambda^2\mathbf{I}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \lambda^2\mathbf{I})$, with $\lambda = 5$ and $\boldsymbol{\mu}_t$ is adapted considering the arithmetic mean of the outputs after a training period, $t \geq 0.2T$, in the same fashion of [25, 19] ($\boldsymbol{\mu}_0 = [1, 1]^\top$). First, we test both techniques fixing $T = 20$ and varying the number of tries N. Then, we set $N = 100$ and vary the number of iterations T. Figure 8 (log-log plot) shows the Mean Square Error (MSE) in the approximation of $\widehat{\mathbf{x}}$ averaged over 10^3 independent runs. Observe that GMS always outperforms the corresponding MTM scheme. These results confirm the advantage of recycling the auxiliary samples drawn at each iteration during an MTM run. In Figure 9(a), we show the MSE obtained by GMS keeping invariant the number of target evaluations $E = NT = 10^3$ and varying $N \in \{1, 2, 10, 20, 50, 100, 250, 10^3\}$. As a consequence, we have $T \in \{10^3, 500, 100, 50, 20, 10, 4, 1\}$. Note that the case $N = 1$, $T = 10^3$, corresponds to an adaptive MH (A-MH) method with a longer chain, whereas the case $N = 10^3$, $T = 1$, corresponds to a static IS scheme (both with the same posterior evaluations $E = NT = 10^3$). We observe that the GMS always provides smaller MSE than the static IS approach. Moreover, GMS outperforms A-MH with the exception of two cases where $T \in \{1, 4\}$.

6.2 Localization of a target and tuning of the sensor network

We consider the problem of positioning a target in $\mathbb{R}^2$ using a range measurements in a wireless sensor network [1, 21]. We also assume that the measurements are contaminated by noise with different unknown power, one per each sensor. This situation is common in several practical

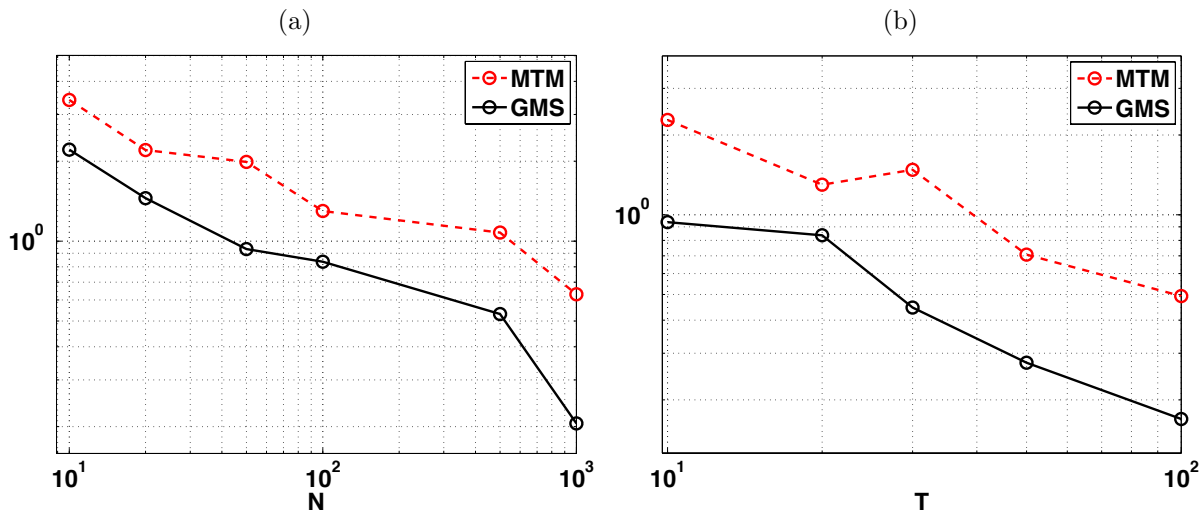

Figure 8: MSE (loglog-scale; averaged over 10^3 independent runs) obtained with the MTM and GMS algorithms (using the same proposal pdf and the same values of N and T) **(a)** as function of N with $T = 20$ and **(b)** as function of T with $N = 100$.

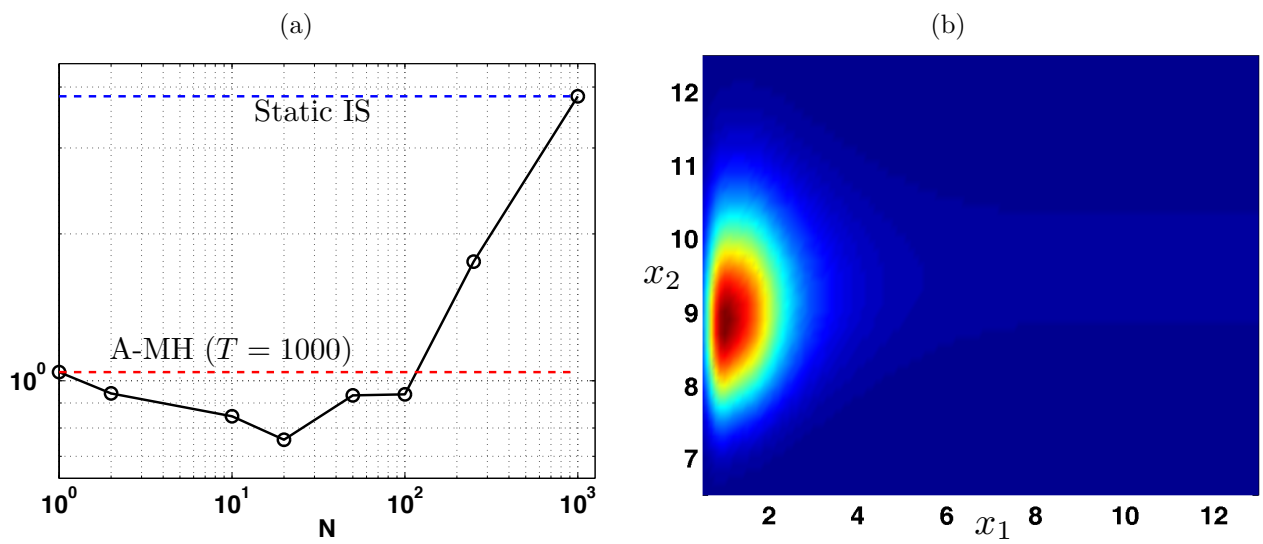

Figure 9: **(a)** MSE (loglog-scale; averaged over 10^3 independent runs) of GMS (circles) versus the number of candidates $N \in \{1, 2, 10, 20, 50, 100, 250, 10^3\}$, but keeping fixed the total number of posterior evaluations $E = NT = 1000$, so that $T \in \{1000, 500, 100, 50, 20, 10, 4, 1\}$. The MSE values the extreme cases $N = 1$, $T = 1000$, and $N = 1000$, $T = 1$, are depicted with dashed lines. In first case, GMS coincides with an adaptive MH scheme (due the adaptation of the proposal, in this example) with a longer chain. The second one represents a static IS scheme (clearly, using the sample proposal than GMS). We can observe the benefit of the dynamic combination of IS estimators obtained by GMS. **(b)** Posterior density $\pi(\mathbf{x}|\mathbf{y}, \mathbf{Z}, \kappa)$.

scenarios. Actually, even if the sensors have the same construction features, the noise perturbation of each the sensor can vary with the time and depends on the location of the sensor. This occurs owing to different causes: manufacturing defects, obstacles in the reception, different physical environmental conditions (such as humidity and temperature) etc. Moreover, in general, these conditions change along time, hence it is necessary that the central node of the network is able to re-estimate the noise powers jointly with position of the target (and other parameters of the models if required) whenever a new block of observations is processed. More specifically, let us denote the target position with the random vector $\mathbf{Z} = [Z_1, Z_2]^\top$. The position of the target is then a specific realization $\mathbf{Z} = \mathbf{z}$. The range measurements are obtained from $N_S = 6$ sensors located at $\mathbf{h}_1 = [3, -8]^\top$, $\mathbf{h}_2 = [8, 10]^\top$, $\mathbf{h}_3 = [-4, -6]^\top$, $\mathbf{h}_4 = [-8, 1]^\top$, $\mathbf{h}_5 = [10, 0]^\top$ and $\mathbf{h}_6 = [0, 10]^\top$ as shown in Figure 10(a). The observation models are given by

$$Y_j = 20 \log \left(||\mathbf{z} - \mathbf{h}_j|| \right) + B_j, \quad j = 1, \ldots, N_S, \tag{40}$$

where B_j are independent Gaussian random variables with pdfs, $\mathcal{N}(b_j; 0, \lambda_j^2)$, $j = 1, \ldots, N_S$. We denote $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{N_S}]$ the vector of standard deviations. Given the position of the target $\mathbf{z}^* = [z_1^* = 2.5, z_2^* = 2.5]^\top$ and setting $\boldsymbol{\lambda}^* = [\lambda_1^* = 1, \lambda_2^* = 2, \lambda_3^* = 1, \lambda_4^* = 0.5, \lambda_5^* = 3, \lambda_6^* = 0.2]$ (since $N_S = 6$), we generate $N_O = 20$ observations from each sensor according to the model in Eq. (40). Then, we finally obtain a measurement matrix $\mathbf{Y} = [y_{k,1}, \ldots, y_{k,N_S}] \in \mathbb{R}^{d_Y}$, where $d_Y = N_O N_S = 120$, $k = 1, \ldots, N_O$. We consider uniform prior $\mathcal{U}(\mathcal{R}_z)$ over the position $[z_1, z_2]^\top$ with $\mathcal{R}_z = [-30 \times 30]^2$, and a uniform prior over λ_j, so that $\boldsymbol{\lambda}$ has prior $\mathcal{U}(\mathcal{R}_\lambda)$ with $\mathcal{R}_\lambda = [0, 20]^{N_S}$. Thus, the posterior pdf is

$$\bar{\pi}(\mathbf{x}|\mathbf{Y}) = \bar{\pi}(\mathbf{z}, \boldsymbol{\lambda}|\mathbf{Y}) = \ell(\mathbf{y}|z_1, z_2, \lambda_1, \ldots, \lambda_{N_S}) \prod_{i=1}^{2} p(z_i) \prod_{j=1}^{N_S} p(\lambda_j), \tag{41}$$

$$= \left[\prod_{k=1}^{N_O} \prod_{j=1}^{N_S} \frac{1}{\sqrt{2\pi\lambda_j^2}} \exp\left(-\frac{1}{2\lambda_j^2} (y_{k,j} + 10 \log \left(||\mathbf{z} - \mathbf{h}_j|| \right)^2 \right) \right] \mathbb{I}_z(\mathcal{R}_z) \mathbb{I}_\lambda(\mathcal{R}_\lambda), \tag{42}$$

where $\mathbf{x} = [\mathbf{z}, \boldsymbol{\lambda}]^\top$ is a vector of parameters of dimension $D = N_S + 2 = 8$ that we desire to infer, and $\mathbb{I}_c(\mathcal{R})$ is an indicator variable that is 1 if $c \in \mathcal{R}$, otherwise is 0.

Our goal is to compute the Minimum Mean Square Error (MMSE) estimator, i.e., the expected value of the posterior $\bar{\pi}(\mathbf{x}|\mathbf{Y}) = \bar{\pi}(\mathbf{z}, \boldsymbol{\lambda}|\mathbf{Y})$. Since the MMSE estimator cannot be computed analytically, we apply Monte Carlo methods for approximating it. We compare GMS, the corresponding MTM scheme, the Adaptive Multiple Importance Sampling (AMIS) technique [10], and N parallel MH chains with a random walk proposal pdf. For all of them we consider Gaussian proposal densities. For GMS and MTM, we set $q_t(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \sigma^2\mathbb{I}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \sigma^2\mathbb{I})$ where is adapted considering the empirical mean of the generated samples after a training period, $t \geq 0.2T$ [25, 19], $\boldsymbol{\mu}_0 \sim \mathcal{U}([1, 5]^D)$ and $\sigma = 1$. For AMIS, we have $q_t(\mathbf{x}|\boldsymbol{\mu}_t, \mathbf{C}_t) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \mathbf{C}_t)$, where $\boldsymbol{\mu}_t$ is as previously described (with $\boldsymbol{\mu}_0 \sim \mathcal{U}([1, 5]^D)$) and $\mathbf{C}_t$ is also adapted using the empirical covariance matrix, starting $\mathbf{C}_0 = 4\mathbf{I}$. We also test the use of N parallel Metropolis-Hastings (MH) chains (we also consider the case of $N = 1$, i.e., a single chain), with a Gaussian random-walk proposal pdf, $q_n(\boldsymbol{\mu}_{n,t}|\boldsymbol{\mu}_{n,t-1}, \sigma^2\mathbf{I}) = \mathcal{N}(\boldsymbol{\mu}_{n,t}|\boldsymbol{\mu}_{n,t-1}, \sigma^2\mathbf{I})$ with $\boldsymbol{\mu}_{n,0} \sim \mathcal{U}([1, 5]^D)$ for all n and $\sigma = 1$.

We fix the total number of evaluations of the posterior density as $E = NT = 10^4$. Note that, generally, the evaluation of the posterior is the most costly step in MC algorithms (however, AMIS has the additional cost of re-weighting all the samples at each iteration according to the deterministic mixture procedure [6, 10, 16]). We recall that T denotes the total number of iterations and N the number of samples drawn from each proposal at each iteration. We consider $\mathbf{x}^* = [\mathbf{z}^*, \boldsymbol{\lambda}^*]^\top$ as the ground-truth and compute the Mean Square Error (MSE) in the estimation obtained with the different algorithms. The results are averaged over 500 independent runs and they are provided in Tables 8, 9, and 10 and Figure 10(b). Note that GMS outperforms AMIS for each a pair $\{N, T\}$ (keeping fixed $E = NT = 10^4$), and GMS also provides smaller MSE values than N parallel MH chains (the case $N = 1$ corresponds to a unique longer chain). Figure 10(b) shows the MSE versus N maintaining $E = NT = 10^4$ for GMS and the corresponding MTM method. This figure again confirms the advantage of recycling the samples in a MTM scheme.

Table 8: Results GMS.

MSE	1.30	1.24	1.22	1.21	1.22	**1.19**	1.31	**1.44**
N	10	20	50	100	200	500	1000	2000
T	1000	500	200	100	50	20	10	5
E	$NT = 10^4$							
MSE range	**Min MSE= 1.19** ——— **Max MSE= 1.44**							

Table 9: Results AMIS [10].

MSE	1.58	1.57	1.53	1.48	1.42	**1.29**	1.48	**1.71**
N	10	20	50	100	200	500	1000	2000
T	1000	500	200	100	50	20	10	5
E	$NT = 10^4$							
MSE range	**Min MSE= 1.29** ——— **Max MSE= 1.71**							

Table 10: Results N parallel MH chains with random-walk proposal pdf.

MSE	1.42	**1.31**	1.44	2.32	2.73	**3.21**	3.18	3.15
N	1	5	10	50	100	500	1000	2000
T	10^4	2000	1000	200	100	20	10	5
E	$NT = 10^4$							
MSE range	**Min MSE= 1.31** ——— **Max MSE=3.21**							

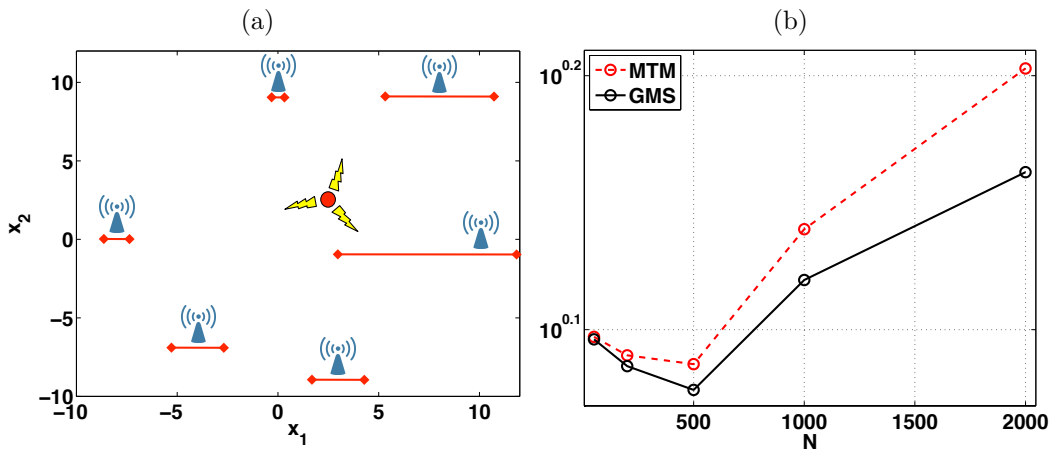

Figure 10: **(a)** Sensor network: the location of the sensors (antennas) and the target (circle) in the numerical example. The solid line represents the different unknown variances of the sensors. **(b)** MSE (log-scale) versus the number of candidates $N \in \{50, 200, 500, 1000, 2000\}$ obtained by GMS and the corresponding MTM algorithm, keeping fixed the total number of evaluations $E = NT = 10^4$ of the posterior pdf, so that $T \in \{200, 50, 20, 10, 5\}$.

6.3 Tracking of biophysical parameters

We consider the challenging problem of estimating biophysical parameters from remote sensing (satellite) observations. In particular, we focus on the estimation of the Leaf Area Index (LAI). It is important to track evolution of LAI through time in every spatial position on Earth because LAI plays an important role in vegetation processes such as photosynthesis and transpiration, and is connected to meteorological/climate and ecological land processes [9]. Let us denote LAI as $x_d \in \mathbb{R}^+$ (where $d \in \mathbb{N}^+$ also represents a temporal index) in a specific region at a latitude of 42° N [18]. Since $x_t > 0$, we consider Gamma prior pdfs over the evolutions of LAI and Gaussian perturbations for the "in-situ" received measurements, y_t. More specifically, we assume the state-space model (formed by propagation and measurement equations),

$$\begin{cases} g_d(x_d|x_{d-1}) & = \mathcal{G}\left(x_d \middle| \frac{x_{d-1}}{b}, b\right) & = \frac{1}{c_d} x_d^{(x_{d-1}-b)/b} \exp\left(-\frac{x_d}{b}\right), \\ \ell_d(y_d|x_d) & = \mathcal{N}(y_d|x_d, \lambda^2) & = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{1}{2\lambda^2}(y_d - x_d)^2\right), \end{cases} \tag{43}$$

for $d = 2, \ldots, D$, with initial probability $g_1(x_1) = \mathcal{G}(x_1|1, 1)$, where $b, \lambda > 0$ and $c_d > 0$ is a normalizing constant. Note that the expected value of the Gamma pdf above is x_{d-1} and the variance is b.

First Experiment. Considering that all the parameters of the model are known, the posterior pdf is

$$\bar{\pi}(\mathbf{x}|\mathbf{y}) \quad \propto \quad \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}) = \left[\prod_{d=2}^{D} \ell_d(y_d|x_d)\right] \left[\left(\prod_{d=2}^{D} g_d(x_d|x_{d-1})\right) g_1(x_1)\right], \tag{44}$$

with $\mathbf{x} = x_{1:D} \in \mathbb{R}^D$. For generating the ground-truth (i.e., the trajectory $\mathbf{x}^* = x^*_{1:D} = [x^*_1, \ldots, x^*_D]$), we simulate the temporal evolution of LAI in one year (i.e., $1 \leq d \leq D = 365$) by using a double logistic function (as suggested in the literature [18]), i.e.,

$$x_d = a_1 + a_2 \left(\frac{1}{1 + \exp(a_3(d - a_4))} + \frac{1}{1 + \exp(a_5(d - a_6))} + 1 \right), \tag{45}$$

with $a_1 = 0.1$, $a_2 = 5$, $a_3 = -0.29$, $a_4 = 120$, $a_5 = 0.1$ and $a_6 = 240$ as employed in [18]. In Figure 11, the true trajectory $x_{1:D}$ is depicted with dashed lines. The observations $\mathbf{y} = y_{2:D}$ are then generated (each run) according to $y_d \sim \ell_d(y_d|x_d) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{1}{2\lambda^2}(y_d - x_d)^2\right)$. First of all, we test the standard PMH, the particle version of GMS (PGMS), and DPMH (fixing $\lambda = 0.1$). For DPMH, we use $M = 4$ parallel filters with different scale parameters $\mathbf{b} = [b_1 = 0.01, b_2 = 0.05, b_3 = 0.1, b_4 = 1]^\top$. Figure 11 shows the estimated trajectories $\widehat{\mathbf{x}}_t = \widehat{x}_{1:D,t} = \frac{1}{t}\sum_{\tau=1}^{t} \widetilde{\mathbf{x}}_\tau$ (averaged over 2000 runs) obtained by DPMH with $N = 5$ at $t \in \{2, 10, 100\}$, in one specific run. Figure 12(a) depicts the evolution of the MSE obtained by DPMH as a function of T and considering different values of $N \in \{5, 7, 10, 20\}$. The performance of DPMH improves as T and N grow, as expected. DPMH detects the best parameters among the four values in $\mathbf{b}$, following the weights $\overline{W}_m$ (see Figure 12(b)) and DPMH takes advantage of this ability. Indeed, we compare DPMH with $N = 10$, $T = 200$, and $M = 4$ using $\mathbf{b}$, with $M = 4$ different standard PMH and PGMS algorithms with $N = 40$ and $T = 200$ (clearly, each one driven by a unique filter, $M = 1$) in order to keep the total number of evaluation of the posterior fixed, $E = NMT = 8 \cdot 10^3$, each one using a parameter b_m, $m = 1, \ldots, M$. The results, averaged over 2000 runs, are shown in Table 11. In terms of MSE, DMPMH always outperforms the 4 possible standard PMH methods. PGMS using two parameters, b_2 and b_3, provides better performance, but DPMH outperforms PGMS averaging the 4 different MSEs obtained by PGMS. Moreover, due to the parallelization, in this case DPMH can save $\approx 15\%$ of the spent computational time.

Second Experiment. Now we consider also the parameter λ unknown, so that the complete variable of interest $[\mathbf{x}, \lambda] \in \mathbb{R}^{D+1}$. Then the posterior is $\bar{\pi}(\mathbf{x}, \lambda|\mathbf{y}) \propto \ell(\mathbf{y}|\mathbf{x}, \lambda) g(\mathbf{x}, \lambda)$ according to the model Eq. (43), where $g(\mathbf{x}, \lambda) = g(\mathbf{x}) g_\lambda(\lambda)$ and $g_\lambda(\lambda)$ is a uniform pdf in $[0.01, 5]$. Then we test the marginal versions of the PMH and DPMH with $q_\lambda(\lambda) = g_\lambda(\lambda)$ (see App. D), for estimating $[\mathbf{x}^*, \lambda^*]$ where $\mathbf{x}^* = x^*_{1:D}$ is given by Eq. (45) and $\lambda^* = 0.7$. Figure 13 shows the MSE in estimation of λ^* (averaged over 1000 runs) obtained by DPMMH as function of T and different number of candidates, $N \in \{5, 10, 20\}$ (with again $M = 4$ and $\mathbf{b} = [b_1 = 0.01, b_2 = 0.05, b_3 = 0.1, b_4 = 1]^\top$). Table 12 compares the standard PMMH and DPMMH for estimating λ^* (we set $E = NMT = 4 \cdot 10^3$ and $T = 100$). Averaging the results of PMMH, we can observe that DPMMH outperforms the standard PMMH in terms of smaller MSE and smaller computational time.

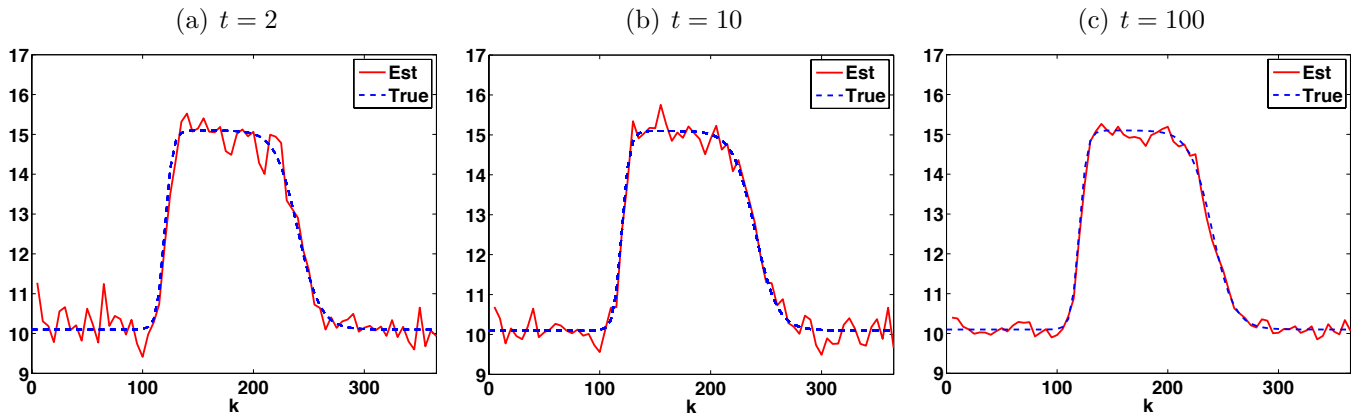

Figure 11: Output of DPMH (with $N = 5$, $\lambda = 0.1$ and $\mathbf{b} = [0.01, 0.05, 0.1, 1]^\top$) at different iterations **(a)** $t = 2$, **(b)** $t = 10$, and **(c)** $t = 100$, in one specific run. The true values, $\mathbf{x}^* = x^*_{1:D}$, are shown dashed lines whereas the estimated trajectories by DPMH, $\widehat{\mathbf{x}}_t = \widehat{x}_{1:D,t}$, with solid lines.

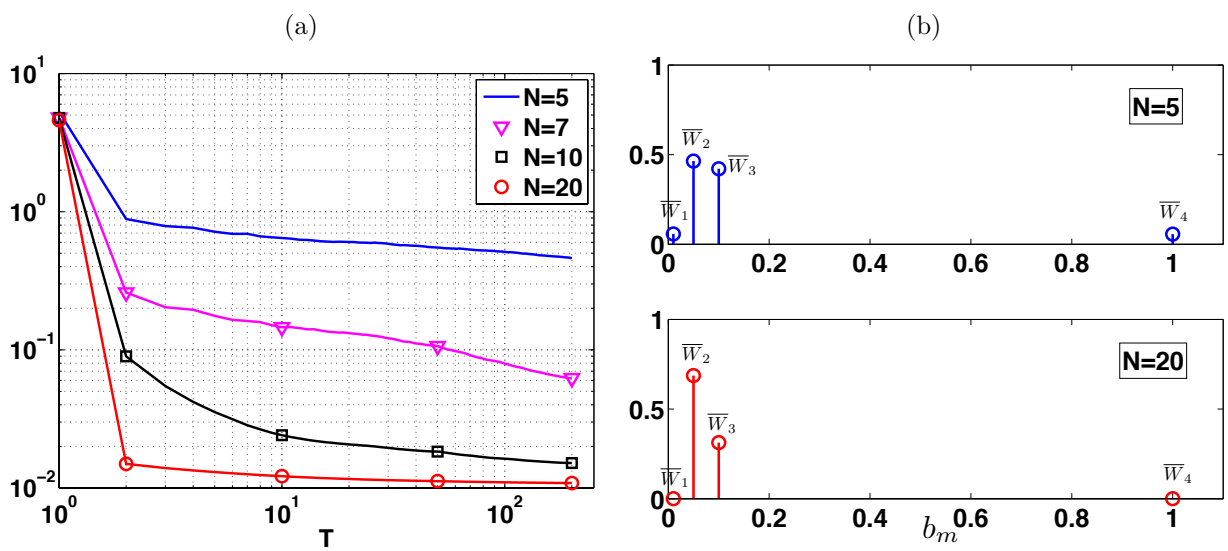

Figure 12: **(a)** MSE in estimation of the trajectory (averaged over 2000 runs) obtained by DPMH as function T and different values of $N \in \{5, 7, 10, 20\}$. As expected, we can see that the performance of DPMH improves as T and N grow. **(b)** Averaged values of the normalized weights $\overline{W}_m = \frac{\widehat{Z}_m}{\sum_{j=1}^{M} \widehat{Z}_j}$ (with $N = 5$ and $N = 10$) associated to each filter. DPMH is able to detect the best variances (b_2 and b_3) of the proposal pdfs among the values $b_1 = 0.01$, $b_2 = 0.05$, $b_3 = 0.1$ and $b_4 = 1$ (as confirmed by Table 11).

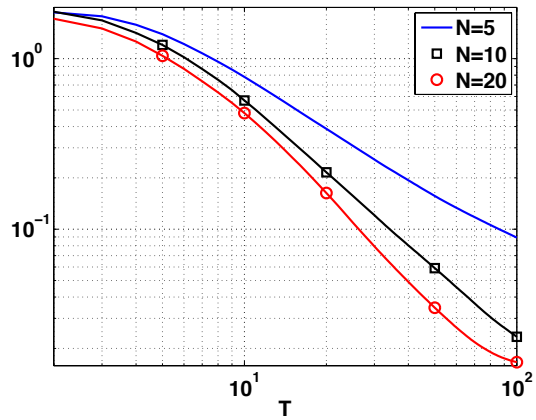

Figure 13: MSE in estimation of $\lambda^* = 0.7$ (averaged over 1000 runs) obtained by DPMMH as function T and different values of $N \in \{5, 10, 20\}$.

Table 11: Comparison among PMH, PGMS and DPMH with $E = NMT = 8 \cdot 10^3$ and $T = 200$ ($\lambda = 0.1$), estimating the trajectory $\mathbf{x}^* = x^*_{1:D}$.

Proposal Var	**Standard PMH**	**PGMS**	**DPMH**
	$N = 40$ ($M = 1$)	$N = 40$ ($M = 1$)	$N = 10$ $M = 4$
	MSE	**MSE**	**MSE**
$b_1 = 0.01$	0.0422	0.0380	0.0108
$b_2 = 0.05$	0.0130	0.0100	
$b_3 = 0.1$	0.0133	0.0102	
$b_4 = 1$	0.0178	0.0140	
Average	0.0216	0.0181	0.0108
Norm. Time	1	1	0.83

7 Conclusions

In this work, we have described the Group Importance Sampling (GIS) theory and its application in other Monte Carlo schemes. We have considered the use of GIS in SIR (a.k.a., particle filtering), showing that GIS is strictly required if the resampling procedure is applied only in a subset of the current population of particles. Moreover we have highlighted that, in the standard SIR method, if GIS is applied there exists two equivalent estimators of the marginal likelihood (one of them is an estimator of the marginal likelihood *only* if the GIS weighting is used), exactly as in Sequential Importance Sampling (SIS). We have also shown that the Independent Multiple Try Metropolis (I-MTM) schemes and the Particle Metropolis-Hastings (PMH) algorithm can be interpreted as a classical Metropolis-Hastings (MH) method taking into account the GIS approach.

Furthermore, two novel methodologies based on GIS have been introduced. One of them (GMS) yields a Markov chain of weighted samples and can be also considered an iterative importance

Table 12: Comparison among PMMH and DMPMH with $E = NMT = 4 \cdot 10^3$ and $T = 100$, for estimating $\lambda^* = 0.7$.

Proposal Var	**PMMH**	**DPMMH**
	$N = 40$ ($M = 1$)	$N = 10$ $M = 4$
	MSE	**MSE**
$b_1 = 0.01$	0.0929	0.0234
$b_2 = 0.05$	0.0186	
$b_3 = 0.1$	0.0401	
$b_4 = 1$	0.0223	
Average	0.0435	0.0234
Norm. Time	1	0.85

sampler. The second one (DPMH) is a distributed version of the PMH where different parallel particle filters can be jointly employed. These filter cooperate for driving the PMH scheme. Both techniques have been applied successfully in three different numerical experiments (tuning of the hyperparameters for GPs, a localization problem in a wireless sensor network and the tracking of the Leaf Area Index), comparing them with several benchmark methods. Marginal versions of GMS and DPMH have been also discussed and tested in the numerical applications. Three Matlab demos have been also given in order to facilitate the comprehension of the reader. As a future line, we plan to design an adaptive DPMH scheme in order to select online the best particle filters among the M run in parallel, and parsimoniously distribute the computational effort.

Acknowledgements

This work has been supported by the European Research Council (ERC) through the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423.

References

[1] A. M. Ali, K. Yao, T. C. Collier, E. Taylor, D. Blumstein, and L. Girod. An empirical study of collaborative acoustic source localization. *Proc. Information Processing in Sensor Networks (IPSN07), Boston*, April 2007.

[2] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

[3] M. Bédard, R. Douc, and E. Mouline. Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, 122:758–786, 2012.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] 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.

[6] M. F. Bugallo, L. Martino, and J. Corander. Adaptive importance sampling in signal processing. *Digital Signal Processing*, 47:36–49, 2015.

[7] R. Casarin, R. V. Craiu, and F. Leisen. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 23(2):185–200, 2013.

[8] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[9] J. M. Chen and T. A Black. Defining leaf area index for non-flat leaves. *Plant, Cell and Environment*, 15:421–429, 1992.

[10] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, December 2012.

[11] R. V. Craiu and C. Lemieux. Acceleration of the Multiple Try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing*, 17(2):109–120, 2007.

[12] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, September 2003.

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

[14] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *technical report*, 2008.

[15] C. C. Drovandi, J. McGree, and A. N. Pettitt. A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design. *Journal of Computational and Graphical Statistics*, 23(1):3–24, 2014.

[16] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized multiple importance sampling. *arXiv:1511.03095*, 2015.

[17] W. J. Fitzgerald. Markov chain Monte Carlo methods with applications to signal processing. *Signal Processing*, 81(1):3–18, January 2001.

[18] J. L. Gomez-Dans, P. E. Lewis, and M. Disney. Efficient emulation of radiative transfer codes using Gaussian Processes and application to land surface parameter inferences. *Remote Sensing*, 8(2), 2016.

[19] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, April 2001.

[20] J. H Huggins and D. M Roy. Convergence of sequential Monte Carlo based sampling methods. *arXiv:1503.00966*, 2015.

[21] 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.

[22] R. Lamberti, Y. Petetin, F. Septier, and F. Desbouvries. An improved sir-based sequential monte carlo algorithm. In *IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5, 2016.

[23] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, England, 2010.

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

[25] D. Luengo and L. Martino. Fully adaptive Gaussian mixture Metropolis-Hastings algorithm. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[26] 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.

[27] L. Martino, V. Elvira, and M. F. Louzada. Effective Sample Size for importance sampling based on the discrepancy measures. *Signal Processing*, 131:386–401, 2017.

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

[29] L. Martino, F. Leisen, and J. Corander. On multiple try schemes and the Particle Metropolis-Hastings algorithm. *viXra:1409.0051*, 2014.

[30] L. Martino and F. Louzada. Issues in the Multiple Try Metropolis mixing. *Computational Statistics*, 32(1):239–252, 2017.

[31] 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.

[32] L. Martino and J. Read. On the flexibility of the design of multiple try Metropolis schemes. *Computational Statistics*, 28(6):2797–2823, 2013.

[33] 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.

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

[35] C. A. Naesseth, F. Lindsten, and T. B. Schon. Nested Sequential Monte Carlo methods. *Proceedings of theInternational Conference on Machine Learning*, 37:1–10, 2015.

[36] C. A. Naesseth, F. Lindsten, and T. B. Schon. High-dimensional filtering using nested sequential monte carlo. *arXiv:1612.09162*, pages 1–48, 2016.

[37] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

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

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

[40] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. *in Bayesian Statistics 3, ads Bernardo, Degroot, Lindley, and Smith. Oxford University Press, Oxford, 1988.*, 1988.

[41] R. Bassi Stern. A statistical contribution to historical linguistics. *Phd Thesis*, 2015.

[42] I. Urteaga, M. F. Bugallo, and P. M. Djurić. Sequential Monte Carlo methods under model uncertainty. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5, 2016.

[43] 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.

[44] C. Verg, P. Del Moral, E. Moulines, and J. Olsson. Convergence properties of weighted particle islands with application to the double bootstrap algorithm. *arXiv:1410.4231*, pages 1–39, 2014.

[45] N. Whiteley, A. Lee, and K. Heine. On the role of interaction in sequential Monte Carlo algorithms. *Bernoulli*, 22(1):494–529, 2016.

A Proper weighting of a resampled particle

Let us consider the particle approximation of $\bar{\pi}$ obtained by the IS approach drawing N particles $\mathbf{x}_n \sim q(\mathbf{x})$,

$$\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) = \sum_{n=1}^{N} \bar{w}(\mathbf{x}_n)\delta(\mathbf{x}-\mathbf{x}_n) = \frac{1}{N\widehat{Z}}\sum_{n=1}^{N} w(\mathbf{x}_n)\delta(\mathbf{x}-\mathbf{x}_n). \tag{46}$$

Given the cloud of particle $\mathbf{x}_{1:N} \sim \prod_{n=1}^{N} q(\mathbf{x}_n)$, we have that $\widetilde{\mathbf{x}}' \sim \widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N})$. Let us denote the joint pdf $\widetilde{Q}(\mathbf{x}, \mathbf{x}_{1:N}) = \widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) \left[\prod_{i=1}^{N} q(\mathbf{x}_i)\right]$. The marginal pdf $\widetilde{q}(\mathbf{x})$ of a resampled particle $\widetilde{\mathbf{x}}'$,

integrating out $\mathbf{x}_{1:N}$ (i.e., $\widetilde{\mathbf{x}}' \sim \widetilde{q}(\mathbf{x})$), is

$$\begin{aligned}
\widetilde{q}(\mathbf{x}) &= \int_{\mathcal{X}^N} \widetilde{Q}(\mathbf{x}, \mathbf{x}_{1:N}) d\mathbf{x}_{1:N} \\
&= \int_{\mathcal{X}^N} \widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) \left[\prod_{i=1}^{N} q(\mathbf{x}_i)\right] d\mathbf{x}_{1:N}, \\
&= \sum_{j=1}^{N} \left(\int_{\mathcal{X}^{N-1}} \frac{w(\mathbf{x})}{N\widehat{Z}} \left[q(\mathbf{x}) \prod_{\substack{i=1 \\ i\neq j}}^{N} q(\mathbf{x}_i) \right] d\mathbf{x}_{\neg j} \right), \\
&= \pi(\mathbf{x}) \sum_{j=1}^{N} \left(\int_{\mathcal{X}^{N-1}} \frac{1}{N\widehat{Z}} \left[\prod_{\substack{i=1 \\ i\neq j}}^{N} q(\mathbf{x}_i) \right] d\mathbf{x}_{\neg j} \right), \\
&= \pi(\mathbf{x}) \int_{\mathcal{X}^{N-1}} \frac{1}{\widehat{Z}} \left[\prod_{\substack{i=1 \\ i\neq j}}^{N} q(\mathbf{x}_i) \right] d\mathbf{x}_{\neg j}, \qquad j \in \{1, \ldots, N\}. \qquad (47)
\end{aligned}$$

Therefore, the standard IS weight of a resampled particle, $\widetilde{\mathbf{x}}' \sim \widetilde{q}(\mathbf{x})$, is

$$w(\widetilde{\mathbf{x}}') = \frac{\pi(\widetilde{\mathbf{x}}')}{\widetilde{q}(\widetilde{\mathbf{x}}')}. \qquad (48)$$

However, generally $\widetilde{q}(\mathbf{x})$ cannot be evaluated, hence the standard IS weight cannot be computed [22, 26, 35], [28, App. C1]. An alternative is to use the Liu's definition of proper weighting in Eq. (8) and look for a weight function $\rho(\widetilde{\mathbf{x}}) = \rho(\widetilde{\mathbf{x}}|\mathbf{x}_{1:N})$ such that

$$E_{\widetilde{Q}(\mathbf{x},\mathbf{x}_{1:N})}[\rho(\mathbf{x}|\mathbf{x}_{1:N})h(\mathbf{x})] = cE_{\overline{\pi}}[h(\mathbf{x})], \qquad (49)$$

where $\widetilde{Q}(\mathbf{x}, \mathbf{x}_{1:N}) = \widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) \left[\prod_{i=1}^{N} q(\mathbf{x}_i)\right]$. Below, we show that a suitable choice is

$$\rho(\widetilde{\mathbf{x}}|\mathbf{x}_{1:N}) = \widehat{Z}(\mathbf{x}_{1:N}) = \frac{1}{N} \sum_{i=1}^{N} w(\mathbf{x}_i), \qquad (50)$$

since it holds in Eq. (49).

Proof. *Note that*

$$\begin{aligned}
E_{\widetilde{Q}(\mathbf{x},\mathbf{x}_{1:N})}[\rho(\mathbf{x}|\mathbf{x}_{1:N})h(\mathbf{x})] &= \int_{\mathcal{X}} \int_{\mathcal{X}^N} \rho(\mathbf{x}|\mathbf{x}_{1:N})h(\mathbf{x})\widetilde{Q}(\mathbf{x}, \mathbf{x}_{1:N}) d\mathbf{x} d\mathbf{x}_{1:N}, \\
&= \int_{\mathcal{X}} \int_{\mathcal{X}^N} h(\mathbf{x})\rho(\mathbf{x}|\mathbf{x}_{1:N})\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) \left[\prod_{i=1}^{N} q(\mathbf{x}_i)\right] d\mathbf{x} d\mathbf{x}_{1:N}. \qquad (51)
\end{aligned}$$

Recalling that $\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:N}) = \frac{1}{N\widehat{Z}}\sum_{j=1}^{N} w(\mathbf{x}_j)\delta(\mathbf{x}-\mathbf{x}_j)$*, where* $\widehat{Z} = \widehat{Z}(\mathbf{x}_{1:N}) = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{x}_n)$ *and* $w(\mathbf{x}_n) = \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}$*, we can rearrange the expectation above as*

$$
\begin{aligned}
E_{\widetilde{Q}(\mathbf{x},\mathbf{x}_{1:N})}[\rho(\mathbf{x}|\mathbf{x}_{1:N})h(\mathbf{x})] &= \int_{\mathcal{X}} h(\mathbf{x}) \left[\sum_{j=1}^{N}\left(\int_{\mathcal{X}^{N-1}} \rho(\mathbf{x}|\mathbf{x}_{1:N})\frac{w(\mathbf{x})}{N\widehat{Z}}\left[q(\mathbf{x})\prod_{\substack{i=1\\ i\neq j}}^{N} q(\mathbf{x}_i)\right] d\mathbf{x}_{\neg j}\right)\right] d\mathbf{x}, \\
&= \int_{\mathcal{X}} h(\mathbf{x})\pi(\mathbf{x}) \left[\sum_{j=1}^{N}\left(\int_{\mathcal{X}^{N-1}} \rho(\mathbf{x}|\mathbf{x}_{1:N})\frac{1}{N\widehat{Z}}\left[\prod_{\substack{i=1\\ i\neq j}}^{N} q(\mathbf{x}_i)\right] d\mathbf{x}_{\neg j}\right)\right] d\mathbf{x},
\end{aligned}
$$

where $\mathbf{x}_{\neg j} = [\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_N]$*. If we choose* $\rho(\mathbf{x}|\mathbf{x}_{1:N}) = \widehat{Z}$ *and replace it in the expression above, we obtain*

$$
\begin{aligned}
E_{\widetilde{Q}(\mathbf{x},\mathbf{x}_{1:N})}[\rho(\mathbf{x}|\mathbf{x}_{1:N})h(\mathbf{x})] &= \int_{\mathcal{X}} h(\mathbf{x})\pi(\mathbf{x}) \left[\sum_{j=1}^{N}\left(\int_{\mathcal{X}^{N-1}} \widehat{Z}\frac{1}{N\widehat{Z}}\left[\prod_{\substack{i=1\\ i\neq j}}^{N} q(\mathbf{x}_i)\right] d\mathbf{x}_{\neg j}\right)\right] d\mathbf{x}, \\
&= \int_{\mathcal{X}} h(\mathbf{x})\pi(\mathbf{x}) N\frac{1}{N} d\mathbf{x}, \\
&= \int_{\mathcal{X}} h(\mathbf{x})\pi(\mathbf{x}) d\mathbf{x} \\
&= cE_{\bar{\pi}}[h(\mathbf{x})], \qquad (52)
\end{aligned}
$$

where $c = Z$*, that is the normalizing constant of* $\pi(\mathbf{x})$*. Note that Eq.* (52) *coincides with* (49)*.* □

B Particle approximation by GIS

Let us consider S samples $\mathbf{x}_{m,n} \sim q_m(\mathbf{x})$, where $S = \sum_{m=1}^{M} N_m$, and weight them $w_{m,n} = \frac{\pi(\mathbf{x}_{m,n})}{q_m(\mathbf{x}_{m,n})}$ with $m = 1, \ldots, M$ and $n = 1, \ldots, N_m$. Moreover, let us define two types of normalized weights, one within the m-th group

$$
\bar{w}_{m,n} = \frac{w_{m,n}}{\sum_{k=1}^{N} w_{m,k}} = \frac{w_{m,n}}{N_m \widehat{Z}_m}, \qquad (53)
$$

and the other one considering all the S samples,

$$
\bar{r}_{m,n} = \frac{w_{m,n}}{\sum_{j=1}^{M}\sum_{k=1}^{N_j} w_{j,k}} = \frac{w_{m,n}}{\sum_{j=1}^{M} N_j \widehat{Z}_j}. \qquad (54)
$$

The complete particle approximation of the target distribution is

$$\begin{aligned}\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:M,1:N}) &= \frac{1}{\sum_{j=1}^{M}\sum_{k=1}^{N_j} w_{j,k}} \sum_{m=1}^{M}\sum_{n=1}^{N_m} w_{m,n}\delta(\mathbf{x}-\mathbf{x}_{m,n}), \\ &= \sum_{m=1}^{M}\sum_{n=1}^{N_m} \bar{r}_{m,n}\delta(\mathbf{x}-\mathbf{x}_{m,n}). \end{aligned} \tag{55}$$

Note that it can be also rewritten as

$$\begin{aligned}\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:M,1:N}) &= \frac{1}{\sum_{j=1}^{M} N_j\widehat{Z}_j} \sum_{m=1}^{M} N_m\widehat{Z}_m \sum_{n=1}^{N} \bar{w}_{m,n}\delta(\mathbf{x}-\mathbf{x}_{m,n}), \\ &= \frac{1}{\sum_{j=1}^{M} N_j\widehat{Z}_j} \sum_{m=1}^{M} N_m\widehat{Z}_m\widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N}), \\ &= \sum_{m=1}^{M} \overline{W}_m\widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N}), \end{aligned} \tag{56}$$

where $\widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N})$ are the m-th particle approximation and $\overline{W}_m = \frac{N_m\widehat{Z}_m}{\sum_{j=1}^{M} N_j\widehat{Z}_j}$ is the normalized weight of the m-th group. If we resample M times $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}(\mathbf{x}|\mathbf{x}_{m,1:N})$ exactly one sample per group, we obtain the particle approximation of Eq. (12), i.e.,

$$\widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_{1:M}) = \sum_{m=1}^{M} \overline{W}_m\delta(\mathbf{x}-\widetilde{\mathbf{x}}_m). \tag{57}$$

Since $\widehat{\pi}(\mathbf{x}|\mathbf{x}_{1:M,1:N})$ is a particle approximation of the target distribution $\bar{\pi}$ (converging to the distribution for $N \to \infty$), then $\widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_{1:M})$ is also a particle approximation of $\bar{\pi}$ (converging for $N \to \infty$ and $M \to \infty$).Therefore, any estimator of the moments of $\bar{\pi}$ obtained using the summary weighted particles as in Eq. (13) is consistent.

C Estimators of the marginal likelihood in SIS and SIR

The classical IS estimator of the normalizing constant $Z_d = \int_{\mathbb{R}^{d\times\eta}} \pi_d(x_{1:d})dx_{1:d}$ at the d-th iteration is

$$\widehat{Z}_d = \frac{1}{N}\sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N}\sum_{n=1}^{N} w_{d-1}^{(n)}\beta_d^{(n)}, \tag{58}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left[\prod_{j=1}^{d}\beta_j^{(n)}\right]. \tag{59}$$

An alternative formulation, denoted as $\overline{Z}_d$, is often used

$$\overline{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right] \tag{60}$$

$$= \prod_{j=1}^{d} \left[\frac{\sum_{n=1}^{N} w_j^{(n)}}{\sum_{n=1}^{N} w_{j-1}^{(n)}} \right] = \widehat{Z}_1 \prod_{j=2}^{d} \left[\frac{\widehat{Z}_j}{\widehat{Z}_{j-1}} \right] = \widehat{Z}_d. \tag{61}$$

where we have employed $\bar{w}_{j-1}^{(n)} = \frac{w_{j-1}^{(n)}}{\sum_{i=1}^{N} w_{j-1}^{(i)}}$ and $w_j^{(n)} = w_{j-1}^{(n)} \beta_j^{(n)}$ [13, 14]. Therefore, given Eq. (61), in SIS these two estimators $\widehat{Z}_d$ in Eq. (58) and $\overline{Z}_d$ in Eq. (60) are equivalent approximations of the d-th marginal likelthood Z_d [33]. Furthermore, note that $\overline{Z}_d$ can be written in a recursive form as

$$\overline{Z}_d = \overline{Z}_{d-1} \left[\sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \beta_d^{(n)} \right]. \tag{62}$$

C.1 Estimators of the marginal likelihood in particle filtering

Sequential Importance Resampling (SIR) (a.k.a., particle filtering) combines the SIS approach with the application of the resampling procedure corresponding to step 2(c)ii of Table 2. If the GIS weighting is not applied, in SIR only

$$\overline{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right].$$

is a consistent estimator of Z_d. In this case, $\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)}$ is not a possible alternative without using GIS. However, considering the proper GIS weighting of the resampled particles (the step 2(c)iii of Table 2), then $\widehat{Z}_d$ is also a consistent estimator of Z_d and it is equivalent to $\overline{Z}_d$. Below, we analyze three cases considering a resampling applied to the entire set of particles:

- **No Resampling ($\eta = 0$):** this scenario corresponds to SIS where $\widehat{Z}_d$, $\overline{Z}_d$ are equivalent as shown in Eq. (61).
- **Resampling at each iteration ($\eta = 1$):** using the GIS weighting, $w_{d-1}^{(n)} = \widehat{Z}_{d-1}$ for all n and for all d, and replacing in Eq. (58) we have

$$\widehat{Z}_d = \widehat{Z}_{d-1} \left[\frac{1}{N} \sum_{n=1}^{N} \beta_d^{(n)} \right], \tag{63}$$

$$= \frac{1}{N} \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \beta_j^{(n)} \right]. \tag{64}$$

Since the resampling is applied to the entire set of particles, we have $\bar{w}_{d-1}^{(n)} = \frac{1}{N}$ for all n. Replacing it in the expression of $\overline{Z}_d$ in (62), we obtain

$$\overline{Z}_d = \frac{1}{N} \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \beta_j^{(n)} \right], \tag{65}$$

that coincides with $\widehat{Z}_d$ in Eq. (64).

- **Adaptive resampling ($0 < \eta < 1$):** for the sake of simplicity, let us start considering a unique resampling step applied at the k-th iteation with $k < d$. We check if both estimators are equal at d-th iteration of the recursion. Due to Eq. (61), we have $\overline{Z}_k \equiv \widehat{Z}_k$,[6] since before the k-th iteration no resampling has been applied. With the proper weighting $w_k^{(n)} = \widehat{Z}_k$ for all n, at the next iteration we have

$$\widehat{Z}_{k+1} = \frac{1}{N} \sum_{n=1}^{N} w_k^{(n)} \beta_{k+1}^{(n)} = \widehat{Z}_k \left[\frac{1}{N} \sum_{n=1}^{N} \beta_{k+1}^{(n)} \right],$$

and using Eq. (62), we obtain

$$\overline{Z}_{k+1} = \overline{Z}_k \left[\sum_{n=1}^{N} \frac{1}{N} \beta_{k+1}^{(n)} \right] = \widehat{Z}_k \left[\frac{1}{N} \sum_{n=1}^{N} \beta_{k+1}^{(n)} \right],$$

so that the estimators are equivalent also at the $(k+1)$-th iteration, $\overline{Z}_{k+1} \equiv \widehat{Z}_{k+1}$. Since we are assuming no resampling steps after the k-th iteration and until the d-th iteration, we have that $\overline{Z}_i \equiv \widehat{Z}_i$ for $i = k+2, \ldots, d$ due to we are in a SIS scenario for $i > k$ (see Eq. (61)). This reasoning can be easily extended for different number of resampling steps.

Figure 14 summarizes the expressions of the estimators in the extreme cases of $\eta = 0$ and $\eta = 1$. Note that the operations of sum and product are inverted. See DEMO-1 at `https://github.com/lukafree/GIS.git`.

D Particle Marginal Metropolis-Hastings (PMMH) algorithms

Let us consider $\mathbf{x} = x_{1:D} = [x_1, x_2, \ldots, x_D] \in \mathcal{X} \subseteq \mathbb{R}^{D \times \xi}$ where $x_d \in \mathbb{R}^{\xi}$ for all $d = 1, \ldots, D$ and an additional model parameter $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ to be inferred as well. Assuming a prior pdf $g_\theta(\boldsymbol{\theta})$ over $\boldsymbol{\theta}$, and a factorized complete posterior pdf $\bar{\pi}(\mathbf{x}, \boldsymbol{\theta})$

$$\bar{\pi}_c(\mathbf{x}, \boldsymbol{\theta}) \propto \pi_c(\mathbf{x}, \boldsymbol{\theta}) = g_\theta(\boldsymbol{\theta}) \pi(\mathbf{x}|\boldsymbol{\theta}),$$

where $\pi(\mathbf{x}|\boldsymbol{\theta}) = \gamma_1(x_1|\boldsymbol{\theta}) \prod_{d=2}^{D} \gamma_d(x_d|x_{1:d-1}, \boldsymbol{\theta})$. Moreover, let us the denote as $\widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:N}, \boldsymbol{\theta}) = \frac{1}{N\widehat{Z}(\boldsymbol{\theta}')} \sum_{n=1}^{N} w(\mathbf{v}_n|\boldsymbol{\theta}') \delta(\mathbf{x} - \mathbf{v}_n)$ a particle approximation of $\pi(\mathbf{x}|\boldsymbol{\theta})$ obtained by one run of a particle

[6] We consider to compute the estimators before the resampling.

No Resampling ($\eta = 0$)	**Always Resampling** ($\eta = 1$)
$\overline{Z}_d \equiv \widehat{Z}_d = \frac{1}{N}\sum_{n=1}^{N}\left[\prod_{j=1}^{d}\beta_j^{(n)}\right]$	$\overline{Z}_d = \widehat{Z}_d = \frac{1}{N}\prod_{j=1}^{d}\left[\sum_{n=1}^{N}\beta_j^{(n)}\right]$
SIS	bootstrap particle filter

Figure 14: Expressions of the marginal likelihood estimators $\overline{Z}_d$ and $\widehat{Z}_d$ in two extreme scenarios: without resampling and applying resampling at each iterations. Note that in the formulations above the operations of sum and product are inverted.

filter approach. The Marginal PMH (PMMH) technique is then summarized in Table 13. PMMH is often used for both smoothing and parameter estimation in state-space models. Note that if $q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = g_\theta(\boldsymbol{\theta})$ then the acceptance function becomes

$$\alpha = \min\left[1, \frac{\widehat{Z}(\boldsymbol{\theta}')}{\widehat{Z}(\boldsymbol{\theta}_{t-1})}\right]. \tag{66}$$

Table 13: Particle Marginal MH (PMMH) algorithm

1. Choose an initial state $\mathbf{x}_0$ and $\widehat{Z}_{m,0}$ for $m = 1, \ldots, M$.
2. For $t = 1, \ldots, T$:
   (a) Choose an initial state $\mathbf{x}_0$, $\boldsymbol{\theta}_0$, $\widehat{Z}(\boldsymbol{\theta}_0)$.
   (b) For $t = 1, \ldots, T$:
      i. Draw $\boldsymbol{\theta}' \sim q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and $\mathbf{v}_j \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:N}, \boldsymbol{\theta}') = \frac{1}{N\widehat{Z}(\boldsymbol{\theta}')}\sum_{n=1}^{N} w(\mathbf{v}_n|\boldsymbol{\theta}')\delta(\mathbf{x}-\mathbf{v}_n)$ (where $\widehat{\pi}$ is obtained with one run of a particle filter).
      ii. Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$, $\mathbf{x}_t = \mathbf{v}_j$, with probability
      $$\alpha = \min\left[1, \frac{\widehat{Z}(\boldsymbol{\theta}')g_\theta(\boldsymbol{\theta}')q_\theta(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\widehat{Z}(\boldsymbol{\theta}_{t-1})g_\theta(\boldsymbol{\theta}_{t-1})q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})}\right]. \tag{67}$$
      Otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$ and $\mathbf{x}_t = \mathbf{x}_{t-1}$.
3. Return $\{\mathbf{x}_t\}_{t=1}^T$ and $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

Distributed Particle Marginal Metropolis-Hastings (DPMMH). We can easily design a marginal version of DPMH in Section 5.2, drawing $\boldsymbol{\theta}' \sim q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and run M particle filters addressing the target pdf $\bar{\pi}(\mathbf{x}|\boldsymbol{\theta}')$. The algorithm follows the steps in Table 13 with the difference that M parallel particle filters are used, and in this case the acceptance probability is

$$\alpha = \min\left[1, \frac{\left[\sum_{m=1}^{M} \widehat{Z}_m(\boldsymbol{\theta})\right] g_\theta(\boldsymbol{\theta}') q_\theta(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\left[\sum_{m=1}^{M} \widehat{Z}_m(\boldsymbol{\theta}_{t-1})\right] g_\theta(\boldsymbol{\theta}_{t-1}) q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})}\right]. \tag{68}$$

Table 14: Main acronyms in the work.

GP	Gaussian Process
MSE	Mean Square Error
IS	Importance Sampling
SIS	Sequential Importance Sampling
SIR	Sequential Importance Resampling
PF	Particle Filter
SMC	Sequential Monte Carlo
AMIS	Adaptive Multiple Importance Sampling
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
IMH	Independent Metropolis-Hastings
MTM	Multiple Try Metropolis
I-MTM	Independent Multiple Try Metropolis
I-MTM2	Independent Multiple Try Metropolis (version 2)
PMH	Particle Metropolis-Hastings
PMMH	Particle Marginal Metropolis-Hastings
GIS	Group Importance Sampling
GSM	Group Metropolis Sampling
PGSM	Particle Group Metropolis Sampling
DPMH	Distributed Particle Metropolis-Hastings
DPMMH	Distributed Particle Marginal Metropolis-Hastings