AN ADAPTIVE POPULATION IMPORTANCE SAMPLER: LEARNING FROM UNCERTAINTY

Luca Martino[⋆], *Víctor Elvira*[†], *David Luengo*[‡], *Jukka Corander*[⋆]

[⋆] Dep. of Mathematics and Statistics, University of Helsinki, 00014 Helsinki (Finland).
[†] Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, 28911 Leganés (Spain).
[‡] Dep. of Circuits and Systems Engineering, Universidad Politécnica de Madrid, 28031 Madrid (Spain).

ABSTRACT

Monte Carlo (MC) methods are well-known computational techniques, widely used in different fields such as signal processing, communications and machine learning. An important class of MC methods is composed of importance sampling (IS) and its adaptive extensions, such as population Monte Carlo (PMC) and adaptive multiple IS (AMIS). In this work, we introduce a novel adaptive and iterated importance sampler using a population of proposal densities. The proposed algorithm, named adaptive population importance sampling (APIS), provides a global estimation of the variables of interest iteratively, making use of all the samples previously generated. APIS combines a sophisticated scheme to build the IS estimators (based on the deterministic mixture approach) with a simple temporal adaptation (based on epochs). In this way, APIS is able to keep all the advantages of both AMIS and PMC, while minimizing their drawbacks. Furthermore, APIS is easily parallelizable. The cloud of proposals is adapted in such a way that local features of the target density can be better taken into account compared to single global adaptation procedures. The result is a fast, simple, robust and high-performance algorithm applicable to a wide range of problems. Numerical results show the advantages of the proposed sampling scheme in four synthetic examples and a localization problem in a wireless sensor network.

Index Terms— Monte Carlo methods, adaptive importance sampling, population Monte Carlo, iterative estimation.

1. INTRODUCTION

Monte Carlo (MC) methods are widely used in signal processing and communications for statistical inference and stochastic optimization [1, 2, 3, 4, 5]. Importance sampling (IS) [6, 7] is a well-known MC methodology to compute efficiently integrals involving a complicated multidimensional target probability density function (pdf), $\pi(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^{D_x}$. Moreover, it is often used in order to calculate the normalizing constant of $\pi(\mathbf{x})$ (also called partition function), which is required in several applications, like model selection [6, 7, 8]. The standard IS technique draws samples from a simple proposal pdf, $q(\mathbf{x})$, assigning weights to them according to the ratio between the target and the proposal, i.e., $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}$. However, although the validity of this approach is guaranteed under mild assumptions, the variance of the estimator depends notably on the discrepancy between the shape of the proposal and the target [6, 7].

Many other techniques to approximate integrals for Bayesian inference have been developed, including asymptotic methods, multiple quadrature rules and Markov Chain Monte Carlo (MCMC) algorithms [9, 10]. In particular, MCMC algorithms are another well-known class of MC techniques which generate a Markov chain converging to the target distribution [6, 11]. MCMC techniques often lead to random walks of the samples generated around the regions of high probability. This exploratory behaviour is responsible for MCMC methods being usually preferred in high-dimensional applications [5, 12, 13]. Nevertheless, MCMC algorithms also suffer from several important shortcomings [6, 7]: the diagnostic of the convergence is often difficult and it is not straightforward to estimate the partition function (i.e., the normalizing constant of the target) given the generated samples, although several algorithms that can address this issue have been recently developed [14, 15, 16].

In the sequel we focus on IS schemes, which are often the preferred approach for the approximation of multi-dimensional integrals in statistics [9, 10]. In order to overcome the problems of standard IS, substantial effort has been devoted to the design of adaptive IS schemes [7], where the proposal density is updated by learning from all the previously generated samples. The population Monte Carlo (PMC) [17] and the adaptive multiple importance sampling (AMIS) [18] methods are two general schemes that combine the proposal adaptation idea with the cooperative use of a cloud of proposal pdfs. On the one hand, in PMC a population of proposals is updated using propagation and resampling steps [7, Chapter 14]. The IS estimator is built as in the standard IS approach, but using a mixture of different proposals [19, 20, 21]. PMC schemes have been widely used in signal processing applications due to their simplicity and flexibility [22, 23, 24].

On the other hand, in AMIS a single proposal is adapted in a standard adaptive IS fashion, but the sequence of all the previous proposals is used to build the importance weights, and the global estimator is constructed according to the so-called *deterministic mixture* approach [25, 26]. This implies that all the previous proposals must be evaluated at the new samples, and also that the new proposal pdf must be evaluated at all the previous samples, thus yielding an increase in computational cost as the algorithm evolves in time. This single proposal could also be a mixture of pdfs, but the adaptation in this case involves more complicated methodologies (such as clustering), increasing the computational cost even more [27]. AMIS has been successfully applied to genetic inference problems recently [28]. Finally, let us remark that the update of the proposals in both methodologies (AMIS and PMC) can also be carried out according to some optimality criterion, such as the minimization of the Kullback-Leibler divergence [19, 20, 21], although at the expense of an increased complexity.

In this work, we introduce a novel population scheme, *adaptive population importance sampling* (APIS).[1] APIS draws samples from different proposal densities at each iteration, weighting these samples according to the deterministic mixture approach [25, 26], which was originally developed for a fixed (i.e., non-adaptive) setting. At each iteration, the APIS algorithm computes iteratively a global IS estimator, taking into account all the generated samples up to that point. The main difference w.r.t. the existing AMIS and PMC schemes lies in the more streamlined adaptation procedure of APIS, as well as in the approach followed to build the estimators. APIS starts with a cloud of N proposals, initialized randomly or according to the prior information available, with the location parameters spread throughout the state space. The initial location parameter for each proposal should be different, and different scale parameters can also be used.[2] The algorithm is then divided into groups (epochs) of T_a iterations, where the proposals are fixed and T_a samples are drawn from each one. At the end of every epoch, the T_a samples drawn from each proposal are used to update its location parameter (using partial IS estimators), and the adaptation memory is "refreshed". This approach allows each proposal to concentrate on some particular region of the state space, thus modelling specific and localized features of the target. In this way, APIS can obtain a very good global approximation of the target by combining all the local approximations. This is achieved without any additional computation in terms of evaluation of the target and proposal pdfs.

Unlike PMC, the novel technique does not require resampling steps to prevent the degeneracy of the mixture, thus avoiding the loss of diversity in the population. This is a common problem for sampling-importance resampling type algorithms, where additional MCMC moves are occasionally applied [30]. Indeed, in [31] the authors attempt to diminish this negative effect by forcing artificially a pre-defined amount of the highest importance weights to be equal to control the loss of diversity caused by resampling. Following the previous observations, we also propose a possible interaction among the proposal locations applying MCMC moves, allowing us to preserve a higher degree of diversity in the population than through the use of resampling steps. We call the resulting technique *Markov APIS (MAPIS)*. MAPIS contains two sources of movement: the APIS movements with the addition of MCMC iterations.

In APIS, at each iteration, the cloud of proposals partake jointly in the construction of an IS estimator using the deterministic mixture approach [25, 26], that introduces more stability in the estimation. This estimator is combined with the past estimators using a simpler strategy than in AMIS: a standard (simpler than the deterministic mixture) IS estimator using multiple proposals is built. Therefore, in this sense, APIS follows an approach "in-between" PMC and AMIS (for further clarifications see Appendix A). Numerical results show that APIS improves the performance of a standard non-adaptive multiple importance sampler regardless of the initial conditions and parameters. We have also compared the performance of APIS to that of several AMIS and PMC schemes, showing that APIS outperforms both approaches in terms of robustness to the choice of the initial parameters.

The paper is organized as follows. The general problem statement is provided in Section 2, and the novel APIS algorithm is described in detail in Section 3. In Section 4, we introduce the additional MCMC adaptation which leads to the MAPIS algorithm. Then, Section 5 is devoted to analyzing differences and similarities among APIS, AMIS and PMC methods. An exhaustive numerical comparison among these three methods is provided in Section 6, where a bidimensional toy example is considered, and Section 7, where two high-dimensional examples are addressed. Section 8 is devoted to comparing APIS with other MCMC approaches (particle splitting methods), whereas the application of APIS to a localization problem in wireless sensor networks is considered in Section 9. Finally, we conclude with a brief summary in Section 10.

[1] A preliminary version of this work has been published in [29]. With respect to that paper, here we propose an interacting adaptation using an MCMC technique, discuss the construction and the consistency of the estimators, and provide more exhaustive numerical simulations, including a localization example in wireless sensor networks. Comparisons with other sampling algorithms are also included.

[2] Since the adaptation of the scale parameters can be an issue for the performance of the sampler, here we focus only on the update of the location parameters in order to ensure the robustness of the algorithm. The development of a robust update mechanism for the scales is left for a future work.

2. PROBLEM STATEMENT AND AIM OF THE WORK

In many applications, we are interested in inferring a variable given a set of observations or measurements. Let us consider the variable of interest, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{D_x}$, and let $\mathbf{y} \in \mathbb{R}^{D_y}$ be the observed data. The posterior pdf is then

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

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the model evidence or partition function (useful in model selection). In general, $Z(\mathbf{y})$ is unknown, so we consider the corresponding (usually unnormalized) target pdf,

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

Our goal is computing efficiently the expected value of $f(\mathbf{X})$, where $\mathbf{X} \sim \frac{1}{Z}\pi(\mathbf{x})$, i.e., an integral measure w.r.t. the target pdf,

$$I = E[f(\mathbf{X})] = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{3}$$

where $Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}$. Our goal is to design a sampling algorithm able to estimate jointly I and Z. Furthermore, we would like to obtain a sampler as efficient and robust as possible, so that the interested user can apply it easily to different problems without having to perform an exhaustive fine tuning of the proposed approach.

3. THE APIS ALGORITHM

The adaptive population importance sampling (APIS) algorithm attempts to estimate jointly Z and I (for an arbitrary function $f(\mathbf{x})$) by drawing samples from a population of adaptive proposals.

3.1. Motivation

Our motivation in designing the APIS algorithm has been trying to exploit the strong points of other adaptive importance sampling algorithms (such as PMC or AMIS), while minimizing their drawbacks. First of all, we consider a cloud of proposal pdfs as in PMC and unlike in AMIS. Moreover, we include the deterministic mixture (DM) approach for building the estimators, since the DM strategy presents advantages in terms of stability and variance w.r.t. the standard IS approach, as shown in Appendix A. However, unlike AMIS (which also exploits the DM technique), we follow a much more efficient approach, dividing the set of iterations of the algorithm into epochs and using the DM scheme to construct the partial IS estimators. In this way, we avoid the increase in computational cost of AMIS with the number of iterations without sacrificing the performance (as shown in the results section). Finally note that, unlike PMC, no resampling step is required in APIS, thus avoiding the loss of diversity in the population.

3.2. Description of the algorithm

For the sake of simplicity, here we consider proposal pdfs defined by two parameters: a location and a scale parameter. However, any other class of proposals can be used, as long as their tails are fatter than the tails of the target density. Currently the adaptation mechanism of APIS focuses exclusively on the location (i.e., first order) parameters, whereas the scale (i.e., second order) parameters are fixed. We have decided not to adapt second or higher order parameters, in order to reinforce the robustness of the sampler.[3]

The APIS algorithm is summarized in Table 1. First of all, the class of the N proposal pdfs, the initial location parameters $\boldsymbol{\mu}_i^{(0)}$ and their scale parameters $\mathbf{C}_i$ have to be fixed. The number of epochs $M = \frac{T}{T_a}$ (or directly the parameter $T_a \geq 2$) also has to be selected. The algorithm works on two different time scales: at each iteration $t = 1, \ldots, T = MT_a$, the global estimates $\hat{I}_t$ and $\hat{Z}_t$ are updated; whereas, at the end of every epoch $m = 1, \ldots, M = \frac{T}{T_a}$, the location parameters $\boldsymbol{\mu}_i^{(m)}$ of the N proposals are updated using the partial IS estimates in Eq. (8).

More specifically, at t-th iteration, one sample from every proposal pdf is generated. The resulting N samples are jointly used, providing a current IS estimator $\hat{J}_t$ obtained by the DM approach. The global estimators $\hat{I}_t$ and $\hat{Z}_t$ are built iteratively as in Eq. (6). Alternative expressions of $\hat{I}_t$ and $\hat{Z}_t$ are given in Eqs. (13)-(15).

[3] It is well-known that the adaptation of second or higher order parameters in adaptive importance sampling schemes is a very delicate issue that can compromise the performance of the algorithm [17]. However, in APIS we can use different scale parameters for the cloud of proposals (as in PMC).

Table 1. APIS algorithm.

1. **Initialization:** Set $t = 1$, $m = 0$, $\hat{I}_0 = 0$, $H_0 = 0$, and $L_0 = 0$. Choose N *normalized* proposal pdfs,

   $$q_i^{(0)}(\mathbf{x}) = q_i(\mathbf{x}; \boldsymbol{\mu}_i^{(0)}, \mathbf{C}_i), \qquad i = 1, \ldots, N,$$

   with location (i.e., mean) vectors $\boldsymbol{\mu}_i^{(0)}$ and scale (i.e., covariance) matrices $\mathbf{C}_i$ ($i = 1, \ldots, N$). Select the number of iterations per epoch, $T_a \geq 2$, and the total number of iterations, $T = MT_a$, with $M \leq \frac{T}{2} \in \mathbb{Z}^+$ denoting the number of epochs. Set also $\boldsymbol{\eta}_i = \mathbf{0}$ and $W_i = 0$ for $i = 1, \ldots, N$.
2. **IS steps:**
   (a) Draw $\mathbf{z}_{i,t} \sim q_i^{(m)}(\mathbf{x})$ for $i = 1, \ldots, N$.

   (b) Compute the importance weights,

   $$w_{i,t} = \frac{\pi(\mathbf{z}_{i,t})}{\frac{1}{N}\sum_{j=1}^{N} q_j^{(m)}(\mathbf{z}_{i,t})}, \quad i = 1, \ldots, N, \tag{4}$$

   and normalize them, $\bar{w}_{i,t} = \frac{w_{i,t}}{S_t}$, where $S_t = \sum_{j=1}^{N} w_{j,t}$.
3. **Iterative IS estimation:** Calculate the current estimate of $I = E[f(\mathbf{X})]$,

   $$\hat{J}_t = \sum_{i=1}^{N} \bar{w}_{i,t} f(\mathbf{z}_{i,t}) \approx I, \tag{5}$$

   and the *global estimate*, using the recursive formula

   $$\hat{I}_t = \frac{1}{H_{t-1} + S_t}\left(H_{t-1}\hat{I}_{t-1} + S_t\hat{J}_t\right) \approx I, \tag{6}$$

   where $H_t = H_{t-1} + S_t$. Note that $\hat{Z}_t = \frac{1}{Nt}H_t$.
4. **Learning:**
   (a) Compute

   $$\rho_i = \frac{\pi(\mathbf{z}_{i,t})}{q_i^{(m)}(\mathbf{z}_{i,t})}, \qquad i = 1, \ldots, N. \tag{7}$$

   (b) Calculate the *partial* estimations of the mean of the target,

   $$\boldsymbol{\eta}_i = \frac{1}{W_i + \rho_i}\left(W_i\boldsymbol{\eta}_i + \rho_i\mathbf{z}_{i,t}\right), \tag{8}$$

   and set $W_i = W_i + \rho_i$ for $i = 1, \ldots, N$.
5. **Proposal adaptation:** If $t = kT_a$ ($k = 1, 2, \ldots, M$):
   (a) Change the location parameters $\boldsymbol{\mu}_i^{(m)}$ according to their partial estimates of the mean of the target, i.e., set

   $$\boldsymbol{\mu}_i^{(m+1)} = \boldsymbol{\eta}_i, \qquad i = 1, \ldots, N, \tag{9}$$

   and $q_i^{(m+1)} = q_i(\mathbf{x}; \boldsymbol{\mu}_i^{(m+1)}, \mathbf{C}_i)$.

   (b) "Refresh memory" by setting $\boldsymbol{\eta}_i = \mathbf{0}$ and $W_i = 0$ for $i = 1, \ldots, N$. Set also $m = m + 1$.
6. **Stopping rule:** If $t < T$, set $t = t + 1$ and repeat from step 2. Otherwise, end.
7. **Outputs:** Return the random measure $\{\mathbf{z}_{i,t}, w_{i,t}\}$, for $i = 1, ..., N$ and $t = 1, ..., T_a$, as approximation of $\bar{\pi}(x)$. More specifically, return the estimate of the desired integral,

   $$\hat{I}_T \approx I = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{10}$$

   as well as the normalizing constant of the target pdf,

   $$\hat{Z}_T \approx Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}. \tag{11}$$

At the end of each epoch, i.e., $t = mT_a$, the locations of the N proposal pdfs are adapted. The update of the location parameter $\boldsymbol{\mu}_i^{(m)}$ is done using only the last T_a samples drawn from the i-th proposal, and building the standard IS estimate $\boldsymbol{\eta}_i$ in Eq. (8) of the expected value $E[\mathbf{X}]$, with $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$.

The underlying idea of APIS is providing a good configuration for the location parameters $\boldsymbol{\mu}_i^{(m)}$, $i = 1, \ldots, N$. Indeed, in

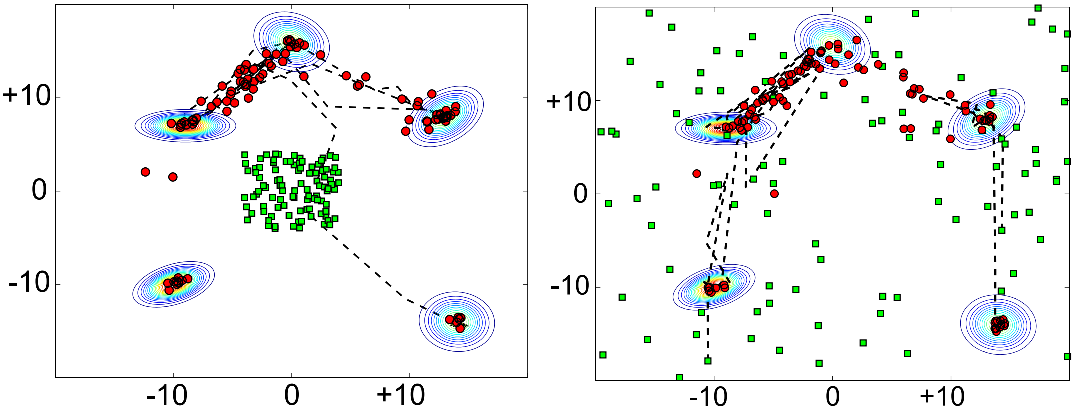


Fig. 1. Contour plot of $\pi(\mathbf{x})$ in Section 6.1, the initial means $\boldsymbol{\mu}_i^{(0)}$ (squares) and the final means $\boldsymbol{\mu}_i^{(T)}$ (circles) obtained in a single run of APIS ($\sigma = 5$, $N = 100$, $T = 2000$, $T_a = \frac{T}{M} = 50$). The trajectories of two means in the sample population are depicted in dashed line. Left figure: $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4,4] \times [-4,4])$ (**In1**). Right figure: $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-20,20] \times [-20,20])$ (**In2**).

APIS we can equivalently state that the proposal consists of an equally weighted mixture of N pdfs:

$$\psi^{(m)}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} q_i^{(m)}(\mathbf{x}|\boldsymbol{\mu}_i^{(m)}, \mathbf{C}_i). \tag{12}$$

In every epoch, NT_a samples are drawn from (12) deterministically by taking exactly one sample from each pdf in the mixture. This mixture can be seen as a kernel density approximation of the target pdf, $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$, where the proposals q_i play the role of the kernels [32, Chapter 6]. Thus, following kernel density estimation arguments, the best configuration for the location parameters is $\boldsymbol{\mu}_i^{(m)} \sim \bar{\pi}(\boldsymbol{\mu})$. Therefore, in general, a good configuration of $\boldsymbol{\mu}_i^{(m)}$, $i = 1, \ldots, N$, is around the modes of the target, as shown in Fig. 1. This ensures a good estimation of the desired integral measure for any arbitrary function $f(\mathbf{x})$ (since $\psi^{(m)}(\mathbf{x})$ approximates $\bar{\pi}(\mathbf{x})$, diminishing the variance of the IS weights $w_{i,t}$). For related considerations and an alternative view of APIS see also Appendix B.

An example of the behaviour of APIS is shown in Fig. 1, which displays a contour plot of a multimodal target $\pi(\mathbf{x})$ and the evolution of the location parameters $\boldsymbol{\mu}_i$, after a run of $T = 2000$ iterations of APIS with $M = 40$, $T_a = \frac{T}{M} = 50$. Gaussian proposals with $\mathbf{C}_i = \sigma^2\mathbf{I}_2$, $\sigma = 5$ have been used and two initializations (shown with squares) have been considered. In the left figure, the initialization is $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4,4] \times [-4,4])$ (**In1**), whereas in the right figure $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-20,20] \times [-20,20])$ (**In2**). The second initialization is better than the first one, since it covers all the areas of high probability of the target and in particular it spans all its modes, towards which the proposals converge. Fig. 1 also depicts the final locations of the means, $\boldsymbol{\mu}_i^{(T)}$ (circles), after T iterations of APIS. Furthermore, the trajectories of two means in the population are depicted by a dashed line. Note that a random walk among different modes is induced in some cases, whereas the corresponding mean remains trapped (after some iterations) around a local mode in other cases.

3.3. Remarks and observations

In this section we provide several remarks on important aspects of the APIS algorithm:

1) All the different proposal pdfs must be normalized.

2) APIS provides a procedure to update the location parameters $\boldsymbol{\mu}_i$, $i = 1, \ldots, N$, in the mixture $\psi^{(m)}(\mathbf{x})$ of Eq. (12).

3) The adaptive mechanism of APIS is driven by the uncertainty in the partial IS estimators (as quantified by their variance). The memoryless feature of APIS facilitates that each proposal pdf can describe local features of the target. Typically, the proposals remain invariable in some regions or a random walk is generated around areas of high probabilities, in the state space $\mathcal{X} \subset \mathbb{R}^{D_x}$.

4) Steps 4 and 5 of APIS do not require additional evaluations of the target and the proposal pdfs since they are already evaluated at $\mathbf{z}_i$, $i = 1, \ldots, N$, in step 2.

5) The global estimators, $\hat{I}_T$ and $\hat{Z}_T$, are iteratively obtained by an importance sampling approach using NT samples drawn from NM different proposals: N initial proposals chosen by the user, and $N(M-1)$ proposals adapted by the algorithm.

Indeed, recall that $\mathbf{z}_{i,t}$ denotes the sample from i-th proposal at the t-th iteration with unnormalized weights in Eq. (4), i.e., $w_{i,t} = \frac{\pi(\mathbf{z}_{i,t})}{\psi^{(m)}(\mathbf{z}_{i,t})}$, where $\psi^{(m)}$ is defined in Eq. (12) and $m = \lfloor \frac{t}{T_a} \rfloor$. Then, the final global estimator $\hat{I}_T$ can be written as

$$\begin{aligned} \hat{I}_T &= \sum_{t=1}^{T} \sum_{i=1}^{N} \bar{\delta}_{i,t} f(\mathbf{z}_{i,t}), \\ &= \frac{1}{\hat{Z}_T} \left(\frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} w_{i,t} f(\mathbf{z}_{i,t}) \right), \end{aligned} \tag{13}$$

where

$$\bar{\delta}_{i,t} = \frac{w_{i,t}}{\sum_{t=1}^{T} \sum_{i=1}^{N} w_{i,t}} = \frac{w_{i,t}}{NT\hat{Z}_T}, \tag{14}$$

and

$$\hat{Z}_T = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} w_{i,t}. \tag{15}$$

The expressions (13), (14), and (15) above show that the global estimator is built using a standard multiple IS approach with the mixtures $\psi^{(m)}(\mathbf{x})$ in Eq. (12) as proposal pdfs, with $m = 1, \ldots, M$, and drawing T_a samples from each of them.

6) Note that the adaptation procedure is independent from each proposal function. Hence, APIS can be completely parallelized if Steps 2(b) and 3 in Table 1 are computed in batch at the end of the algorithm. All the evaluations of the target are performed in parallel in Step 4(a).

7) APIS degenerates into a static algorithm when $T_a = T$ (i.e., $M = 1$). In this scenario, where the adaptation of the proposals never occurs, an iterated multiple IS algorithm is performed. We denote this algorithm, which combines the deterministic mixture idea and the standard IS approach to build the global estimators, and thus is different from a standard multiple IS scheme, as *static APIS* or *PIS*.

3.4. Choice of the parameters

As in any other Monte Carlo technique, the performance of APIS depends on a proper initialization and choice of the parameters, although this sensitivity is reduced w.r.t. a standard IS approach, as illustrated in the simulations. Hence, if some prior information about the target is available, it should be used to choose the initial parameters. In the following, we briefly discuss how to select the main parameters of the algorithm: $\boldsymbol{\mu}_i^{(0)}$, $\mathbf{C}_i$ and T_a.

3.4.1. Initial location parameters $\boldsymbol{\mu}_i^{(0)}$

If no prior information about the target is available, then the initial locations for the proposals should be chosen in order to cover as much as possible of the target's domain, $\mathcal{X} \subseteq \mathbb{R}^{D_x}$. Otherwise, they should be distributed according to the prior.

3.4.2. Scale parameters $\mathbf{C}_i$

Since the scale parameters are not adapted, it is advisable to use different scales for the proposals. The simplest possibility is to choose them randomly (within a range of acceptable scales) for each proposal. Another possibility is associating more than one variance to each proposal pdf. For instance, selecting N_μ initial location parameters and N_σ different scale parameters for each one, implying that the total number of different proposals is $N = N_\mu N_\sigma$.

3.4.3. Samples from each proposal per epoch $T_a = \frac{T}{M}$

T_a is the number of samples used to choose the new location parameters at the end of every epoch. As T_a grows, each partial IS estimator $\boldsymbol{\eta}_i$ (used to adapt the proposals) provides a better estimation, closer to the expected value of the target $\pi(\mathbf{x})$, and also closer to the estimates provided by other proposals. Although this is clearly a good scenario, it is not the best situation for APIS, as the proposals would tend to cover the same region of the target's domain, thus losing diversity in the population. For smaller values of T_a the proposal pdfs tend to be spread out around the regions of high probability, which is a better configuration for

APIS. However, if T_a is too small, large and almost random movements of the proposals are encouraged throughout the state space.[4]

In any case, even with a bad choice of the parameters APIS always provides a consistent IS estimator; even in the worst cases APIS provides better performance than an adaptive IS scheme using a single proposal and a static multiple IS scheme with a random choice of the parameters, as shown in the simulations. Furthermore, the numerical results in Section 6 also suggest the existence of an optimal value T_a^* (or equivalently $M^* = \frac{T}{T_a^*}$), which depends of the scale parameters $\mathbf{C}_i$ of the proposals and the dimension D_x. In general, proposals with small variances provide a better performance using smaller values of T_a, whereas big variances work better with larger values of T_a. Different values of $T_a^{(i)}$ (one for each proposal) could be applied according to their scale parameters and they could even be changed with the time step: smaller values for a more explorative behaviour at the beginning, and larger values to reduce the uncertainty of the proposals as time evolves.

4. MCMC INTERACTION: MARKOV APIS

In APIS the adaptation of the location parameter of a proposal is done independently from the rest of the population. Here we propose a possible interaction procedure among the location parameters of the proposal pdfs that avoids the loss of diversity in the population caused by a resampling step (another form of interaction). We propose to share information among proposals by applying an MCMC technique over the cloud of means, $\{\boldsymbol{\mu}_i\}_{i=1}^N$, at every transition between two epochs, i.e., $t = mT_a$ with $m = 1, \dots, M$. An appropriate MCMC technique for this purpose is the *Sample Metropolis-Hastings* (SMH) algorithm [11, Chapter 5]. We denote the SMH-enhanced APIS algorithm as *Markov APIS (MAPIS)*. The proposed MCMC iterations are applied after step 5 of APIS. Thus, MAPIS contains two sources of movement for the proposals: step 5 of APIS plus the SMH iterations. Observe that, unlike steps 4 and 5 of APIS, these SMH steps require new evaluations of the target pdf. For the sake of simplicity, in this section we remove the super-index denoting the current epoch from the location parameters $\boldsymbol{\mu}_i$.

4.1. Sample Metropolis-Hastings (SMH) algorithm

Consider the extended target pdf

$$\bar{\pi}_g(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N) \propto \prod_{i=1}^{N} \pi(\boldsymbol{\mu}_i), \tag{16}$$

where each marginal $\pi(\boldsymbol{\mu}_i)$, $i = 1, ..., N$, coincides with the target pdf in Eq. (2). Let us denote as $\tau = 1, \dots, \Upsilon$ the SMH iteration index. At the τ-iteration, we consider *the population* of samples

$$\mathcal{P}_\tau = \{\boldsymbol{\mu}_{1,\tau}, ..., \boldsymbol{\mu}_{N,\tau}\}.$$

At each iteration, the underlying idea of SMH is to replace one "bad" sample in the population with a "better" one, according to certain suitable probabilities. The algorithm is designed so that, after a burn-in period τ_b, the elements in $\mathcal{P}_{\tau'}$ ($\tau' > \tau_b$) are distributed according to $\pi_g(\boldsymbol{\mu}_{1,\tau'}, \dots, \boldsymbol{\mu}_{N,\tau'})$, i.e., $\boldsymbol{\mu}_{i,\tau'}$ are i.i.d. samples from $\pi(\mathbf{x})$. For $\tau = 1, ..., \Upsilon$, the SMH algorithm consists of the following steps:

1. Draw $\boldsymbol{\mu}_{0,\tau} \sim \varphi(\boldsymbol{\mu})$, where φ is another proposal density, chosen by the user, which could be based on the information obtained from the previous steps of APIS.

2. Choose a "bad" sample, $\boldsymbol{\mu}_{k,\tau}$ with $k \in \{1, ..., N\}$, from the population according to a probability proportional to $\frac{\varphi(\boldsymbol{\mu}_{k,\tau})}{\pi(\boldsymbol{\mu}_{k,\tau})}$, which corresponds to the inverse of the importance sampling weights.

3. Accept the new population, $\mathcal{P}_{\tau+1} = \{\boldsymbol{\mu}_{1,\tau+1} = \boldsymbol{\mu}_{1,\tau}, ..., \boldsymbol{\mu}_{k,\tau+1} = \boldsymbol{\mu}_{0,\tau}, ..., \boldsymbol{\mu}_{N,\tau+1} = \boldsymbol{\mu}_{N,\tau}\}$, with probability

$$\alpha(\mathcal{P}_\tau, \boldsymbol{\mu}_{0,\tau}) = \frac{\sum_{i=1}^{N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}{\sum_{i=0}^{N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0 \le i \le N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}. \tag{17}$$

   Otherwise, set $\mathcal{P}_{\tau+1} = \mathcal{P}_\tau$.

4. If $\tau < \Upsilon$, set $\tau = \tau + 1$ and repeat from step 1.

[4] A physical analogy could help the reader in understanding the behavior: defining an *energy* variable $E = \frac{1}{T_a}$, increasing T_a means to cool down the cloud (less energy E) whereas decreasing T_a means to heat up the system, i.e., rise the energy E in the cloud of particles, increasing the total entropy.

Observe that the difference between $\mathcal{P}_\tau$ and $\mathcal{P}_{\tau+1}$ is at most one sample. The ergodicity can be proved considering the extended target function $\bar{\pi}_g(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N)$ and proving the detailed balance condition, as shown in Appendix C. Furthermore, for $N = 1$ it is possible to show that SMH becomes the standard MH method with an independent proposal pdf [11].

4.2. Benefits of the interaction via MCMC

The use of the MCMC step facilitates the movement of the means towards the high-probability regions of the target, regardless of the choice of the initial parameters. Indeed, it can help to reallocate "lost" means in a better position. This step could stop an explorative random walk behaviour of some proposal and reallocate it around a mode. This effect is particularly advantageous when the chosen value T_a is smaller than the optimal one T_a^* and complements the basic adaptive mechanism of APIS, allowing us to avoid the degeneracy problem characteristic of particle systems.

Note that only one new importance weight needs to be evaluated at each iteration, since the rest of the weights have already been computed in the previous steps (except for the initial iteration, where all need to be computed). Finally, we note also that the locations of the proposals hardly ever change if the parameters of φ are not properly chosen, since new points are never accepted. However, this issue can be easily solved by adapting these parameters using some of the existing adaptive MCMC strategies [13, 33].

5. RELATIONSHIP WITH AMIS AND PMC AND CONSISTENCY OF THE ESTIMATORS

5.1. Estimators in PMC and AMIS: relationship with APIS

Table 2. Comparison among the AIS, PMC, AMIS and APIS algorithms.

Algorithm	**Approaches in the estimation**		**Adaptation**		
	Space	**Time**	**type of IS weights**	**Memory**	**Equilibrium**
Standard Adaptive IS	none (single proposal)	Standard IS	Standard IS	Long	Static
(Basic) Generic PMC [7, 17]	Standard IS	Standard IS	Standard IS (Resampling)	Short (Resampling done on the current cloud)	Dynamic
Modified PMC (as in [19, 20])	Deterministic mixture	Standard IS	Standard IS (Resampling)	Short (Resampling done on the current cloud)	Dynamic
AMIS [18]	none (single proposal)	Deterministic mixture	Deterministic mixture	Long	Static
Modified AMIS [34]	none (single proposal)	Deterministic mixture	Standard IS	Short (like epochs in APIS)	Both Dynamic/ Pseudo-static (with N fixed)
APIS	Deterministic mixture	Standard IS	Standard IS	Short (epochs)	Both Dynamic/ Pseudo-static
Markov APIS	Deterministic mixture	Standard IS	Standard IS (+ MCMC)	Short (epochs)	Dynamic

To clarify the different estimators used in PMC, AMIS and APIS, we distinguish two different stages w.r.t. the exchange of statistical information among the proposal pdfs:

- *In space* ($\mathbf{x} \in \mathcal{X}$)*:* creating an estimator by sharing information among different proposal pdfs (i.e., forming a *population*) for a given time step.

- *In time* ($t \in \mathbb{N}$): combining information obtained in different iterations to create a global estimator.

PMC schemes use a cloud of proposal pdfs in each iteration (spread throughout the *state space* of the variable of interest), following the standard IS approach (see Appendix A.1) to construct the estimator. The temporal combination of the information (i.e., the global estimator) can be built in different ways, but the importance weights are based on the standard IS approach in general [35]. In the numerical simulations, we also consider a *modified version* of PMC (M-PMC), where the spatial sharing of information is performed through a deterministic mixture (see Appendix A.2). This idea is based on the *Rao-Blackwellised version of the D-kernel PMC* algorithm [19, 20]. However, the way in which the mixture of proposals is updated in this algorithm (using the Kullback-Leibler divergence) is much more complicated than in APIS. Furthermore, in [21] the authors suggest a procedure to adapt all the parameter of a mixture of pdfs. However, the resulting algorithm is extremely sensitive to the initial conditions, and thus quite unstable.

The AMIS algorithm uses a single proposal pdf at each iteration: N samples are drawn at every step from the same proposal. However, *all* of the previously adapted proposal pdfs are considered to build a global estimator, following the DM approach. This is clearly the most stable way to construct the global estimator, but it is also the most costly, since all the past proposal pdfs need to be re-evaluated at every iteration. Consequently, the computational cost of AMIS grows as the algorithm evolves and the pool of previous proposals becomes larger, thus becoming unfeasible for a medium/large number of iterations in practice. In this sense, APIS lies "in between" PMC and AMIS: we use the deterministic mixture idea in *space* at each iteration, as shown in Eqs. (4)-(5), but keep the standard IS approach to build the global estimator (in *time*), as shown in Eq. (6). Therefore, the computational cost of APIS is reduced w.r.t. AMIS, since APIS does not need to re-evaluate past proposal pdfs, thus being able to maintain a fixed computational cost per iteration (unlike AMIS).

5.2. Consistency of the estimators

The consistency of the global estimator provided by APIS must be ensured when number of samples per time step (N) and/or the number of iterations of the algorithm (T) grow to infinity. In APIS, the global estimator, $\hat{I}_T$, is given by (13), with the estimator of the normalizing constant, $\hat{Z}_T$, given by (15). For $N \to \infty$ and a fixed number of iterations $T < \infty$, the consistency can be guaranteed by standard IS arguments, since it is well known that $\hat{Z}_T \to Z$ and $\hat{I}_T \to I$ as $N \to \infty$ [7]. For $T \to \infty$ and $N < \infty$, we have a convex combination of independent, consistent and biased IS estimators [7].[5] However, $\hat{Z}_T \to Z$ as $T \to \infty$, as discussed in [7, Chapter 14] for PMC schemes. This scenario also applies to APIS, since a standard IS approach is applied *in time* to build the global estimator. Hence, we can ensure that $\hat{I}_T$ is asymptotically unbiased and consistent as $T \to \infty$.

In AMIS, the analysis for $T \to \infty$ is much more complicated [34], since a long memory dependence among the samples is introduced by the use of the deterministic mixture approach *in time*. Indeed, the IS weights (built using the DM scheme) are also used to adapt the proposal pdf in AMIS, yielding a bias that cannot be easily controlled from a theoretical point of view. A similar and well-known problem appears in adaptive MCMC techniques: even if the kernel of the algorithm is valid at every step, a wrong adaptation (change of the kernel) using all the previously generated samples can easily jeopardize the convergence of the chain. Thus, in order to prove the consistency of AMIS, the authors in [34] suggest a simplification in the adaptive structure of AMIS. In this modified approach, the adaptation is performed using only the more recently generated samples (in APIS terminology this corresponds to one epoch) and standard IS weights, whereas the global estimation still uses the deterministic mixture approach. Note that this resembles the adaptive structure of APIS, thus reinforcing the idea that APIS is a robust technique, partly thanks to the memoryless feature of its adaptation.

5.3. Evolution of the proposals

In AMIS, the parameters of the proposal are updated and they converge to fixed values after a certain number of iterations (as in a standard adaptive IS). Thus, the "distance" between two proposals at different time steps diminishes as $T \to \infty$. In the basic PMC schemes, the location parameters of the cloud of proposals are updated via resampling. In this case, the positions of the proposals change at every iteration, moving around the modes of the target as in a "dynamic equilibrium". In APIS both situations can occur, as shown in Fig. 1. On the one hand, random walks around high probability regions can be generated due to partial memoryless IS estimates or MCMC iterations (unlike PMC, where they are due to the resampling procedure and can result in loss of diversity). On the other hand, some proposal could also reach a pseudo-static equilibrium as in AMIS, for instance becoming trapped in a local mode. Both behaviours present certain advantages and APIS benefits from both features, attaining a trade off between explorative search and stability in the estimation.

[5] The locations of the proposals depend on the previous configuration of the cloud, but the samples drawn at each iteration are independent of the previous ones and each other, thus leading to independent IS estimators. The bias is due to the estimation of Z, the normalizing constant of π.

Table III.1

Std Alg.		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		29.56	41.95	64.51	42.84	2.17	0.0454	0.0147	0.0187	0.1914	4.55
PIS ($T_a = T$)		29.28	47.74	75.22	17.61	0.2424	0.0280	0.0124	0.0176	0.1789	0.0651
APIS	$T_a = 100$	22.86	13.70	6.2606	2.47	0.0438	0.0131	0.0129	0.0212	0.1821	0.0110
	$T_a = 50$	17.62	12.14	5.42	1.99	0.0501	0.0118	0.0138	0.0209	0.1750	0.0077
	$T_a = 20$	14.75	11.33	4.77	1.66	0.0361	**0.0108**	0.0146	0.0208	0.1873	0.0056
	$T_a = 5$	13.01	8.50	2.30	0.2831	**0.0074**	0.0114	0.0149	0.0251	0.2027	**0.0045**
	$T_a = 2$	**9.46**	**2.45**	**0.0225**	**0.0170**	0.0103	0.0139	0.0185	0.0354	0.2007	0.0077
AMIS	(best)	124.22	121.21	100.23	54.67	0.8640	0.0124	**0.0121**	**0.0126**	**0.0136**	$- - - - - -$
	(worst)	125.43	123.38	114.82	89.09	16.92	0.3626	0.0128	0.0131	18.66	$- - - - - -$
PMC	$N = 100$	112.99	114.11	47.97	26.32	2.34	0.5217	0.0559	0.4331	2.41	0.3017
M-PMC	$N = 100$	111.92	107.58	26.86	6.03	0.6731	0.1154	0.0744	0.4142	2.42	0.07

Table III.2

MAPIS ($N = 100$)	$T_a = 100$	0.7134	0.0933	0.3213	0.1611	0.0167	0.0101	0.0147	0.0023	0.1765	0.0070
	$T_a = 50$	0.7058	0.1287	0.1136	0.1097	0.0114	**0.0094**	0.0139	0.0020	0.1831	0.0051
	$T_a = 20$	0.6950	0.1319	0.0464	0.1040	**0.0081**	0.0098	0.0152	0.0021	0.1943	**0.0041**
	$T_a = 5$	0.2729	0.0665	0.0319	0.0154	0.0082	0.0123	0.0151	0.0019	0.1946	0.0046
	$T_a = 2$	**0.1708**	**0.0148**	**0.0116**	**0.0138**	0.0105	0.0130	0.0165	0.0027	0.1918	0.0075
PMC	$N = 500$	112.18	113.10	36.63	18.59	2.20	0.4011	0.0134	0.0259	0.8891	0.2964
	$N = 2000$	112.09	112.45	27.91	13.63	2.01	0.1899	**0.0057**	**0.0028**	**0.1120**	0.2802

Table 3. **(Ex-in-Sect 6)** MSE of the estimation of the mean of the target (first component) with the initialization **In1**. In Table III.1, we set $N = 100$ ($T = 2000$; the total number of samples is $L = NT = 2 \cdot 10^5$) for MIS, PIS, APIS and PMC. For AMIS, we show the best results, obtained varying K and M such that $L = KM = 2 \cdot 10^5$.

6. TOY EXAMPLE: NUMERICAL COMPARISON

6.1. Target distribution

In order to test and compare APIS with other algorithms, we first consider a bivariate multimodal target pdf, which is itself a mixture of 5 Gaussians, i.e.,

$$\pi(\mathbf{x}) = \frac{1}{5} \sum_{i=1}^{5} \mathcal{N}(\mathbf{x}; \nu_i, \mathbf{\Sigma}_i), \quad \mathbf{x} \in \mathbb{R}^2, \tag{18}$$

with means $\nu_1 = [-10, -10]^\top$, $\nu_2 = [0, 16]^\top$, $\nu_3 = [13, 8]^\top$, $\nu_4 = [-9, 7]^\top$, $\nu_5 = [14, -14]^\top$, and covariance matrices $\mathbf{\Sigma}_1 = [2,\ 0.6; 0.6,\ 1]$, $\mathbf{\Sigma}_2 = [2,\ -0.4; -0.4,\ 2]$, $\mathbf{\Sigma}_3 = [2,\ 0.8; 0.8,\ 2]$, $\mathbf{\Sigma}_4 = [3,\ 0; 0,\ 0.5]$ and $\mathbf{\Sigma}_5 = [2,\ -0.1; -0.1,\ 2]$. Fig. 1 shows a contour plot of $\pi(\mathbf{x})$. Note that we can compute analytically moments of the target in Eq. (18), so we can easily check the performance of the different techniques.

6.2. Goal, comparisons and initialization

We consider the problem of computing (a) the mean of the target, i.e., $E[\mathbf{X}] = [1.6, 1.4]^\top$ where $\mathbf{X} \sim \frac{1}{Z}\pi(\mathbf{x})$, (b) and the normalizing constant $Z = 1$, using Monte Carlo techniques. We compare the performance in terms of Mean Square Error (MSE) in the estimation using different sampling methodologies: **(1)** standard, non-adaptive, Multiple IS (MIS) approach; **(2)** PIS (or static APIS) scheme; **(3-4)** APIS and MAPIS (APIS with the MCMC interaction) methods; **(5)** the AMIS technique [18]; and **(6)** a PMC acheme [17]. Moreover, we test for all the previous techniques two different initializations:

In1: First, we choose deliberately a "bad" initialization of the initial means in the sense that they are placed far away from the modes. Thus, we can test the robustness of the algorithms and their ability to improve the corresponding *static* approaches. Specifically, the initial location parameters are selected uniformly within a square,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4, 4] \times [-4, 4]),$$

for $i = 1, \ldots, N$. A single realization of $\boldsymbol{\mu}_i^{(0)}$ is depicted by the squares in Fig. 1(a) (jointly with the final locations $\boldsymbol{\mu}_i^{(T)}$, in one specific run).

Table IV.1

Alg. \ Std		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		12.00	9.40	10.26	10.64	7.67	4.40	0.5443	0.0321	0.1764	4.37
PIS ($T_a = T$)		10.14	0.9469	0.0139	0.0085	0.0100	0.0115	0.0146	0.0237	0.1756	0.0106
APIS	$T_a = 100$	0.7741	0.0318	0.0011	0.0017	0.0054	0.0118	**0.0129**	0.0211	0.1794	0.0032
	$T_a = 50$	0.5792	0.0144	0.0007	0.0015	0.0051	0.0112	0.0131	0.0221	0.1772	**0.0029**
	$T_a = 20$	0.4831	0.0401	0.0006	**0.0014**	**0.0047**	**0.0095**	0.0136	0.0245	0.1732	**0.0029**
	$T_a = 5$	0.2552	**0.0008**	**0.0005**	0.0022	0.0064	0.0111	0.0149	0.0270	0.2076	0.0039
	$T_a = 2$	**0.0547**	0.0017	0.0116	0.0051	0.0103	0.0142	0.0182	0.0387	0.1844	0.0080
AMIS	(best)	113.97	112.70	107.85	91.56	44.93	12.75	0.7404	**0.0121**	**0.0141**	− − − − −
	(worst)	116.66	115.62	111.83	104.44	70.62	35.66	9.43	0.0871	18.62	− − − − −
PMC	$N = 100$	111.54	110.78	90.21	46.84	2.29	0.5023	0.0631	0.4273	2.42	0.3082
M-PMC	$N = 100$	23.16	7.43	7.56	3.11	0.6420	0.1173	0.0720	0.4194	2.37	0.0695

Table IV.2

MAPIS ($N = 100$)	$T_a = 100$	0.4753	0.0334	0.0027	0.0017	**0.0059**	0.0092	0.0135	0.0217	0.1762	0.0034
	$T_a = 50$	0.4677	0.0287	0.0007	0.0015	**0.0059**	**0.0091**	0.0133	0.0222	0.1901	0.0031
	$T_a = 20$	0.3110	0.0092	**0.0006**	**0.0014**	0.0061	**0.0091**	0.0141	0.0233	0.1805	**0.0030**
	$T_a = 5$	0.3497	0.0015	0.0007	0.0041	0.0079	0.0122	0.0155	0.0249	0.1933	0.0039
	$T_a = 2$	**0.0870**	**0.0101**	0.0028	0.0060	0.0098	0.0126	0.0154	0.0333	0.2026	0.0078
PMC	$N = 500$	110.58	109.69	64.81	15.99	2.09	0.4841	0.0144	0.0267	0.8924	0.2900
	$N = 2000$	108.22	107.10	27.93	13.21	1.84	0.1912	**0.0054**	**0.0027**	**0.0988**	0.2805

Table 4. **(Ex-in-Sect 6)** MSE of the estimation of the mean of the target (first component) with the initialization **In2**. In Table IV.1, we set $N = 100$ ($T = 2000$; the total number of samples is $L = NT = 2 \cdot 10^5$) for MIS, PIS, APIS and PMC. For AMIS, we show the best results, obtained varying K and M such that $L = KM = 2 \cdot 10^5$.

In2: We also consider a better initialization where some proposals are placed close to the modes. Specifically, the initial means are selected uniformly within a square,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-20, 20] \times [-20, 20]),$$

for $i = 1, \dots, N$. A single realization of $\boldsymbol{\mu}_i^{(0)}$ is depicted by the squares in Fig. 1(b) (jointly with the final locations $\boldsymbol{\mu}_i^{(T)}$, in one specific run).

Below we provide more details of each applied scheme (providing the used parameters).

6.3. Techniques

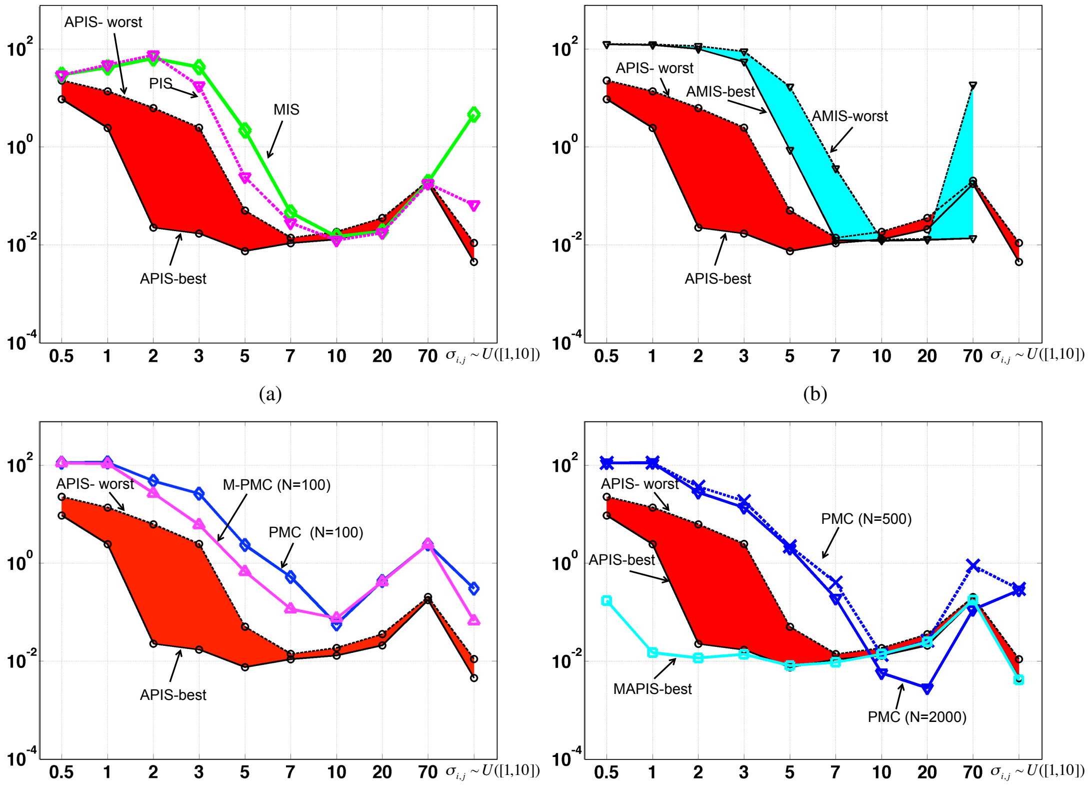
We apply the following techniques proposed in this paper:

- *APIS:* we apply APIS with $N = 100$ Gaussian proposals

$$q_i^{(m)}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(m)}, \mathbf{C}_i), \qquad i = 1, \dots, N.$$

The initial configurations of the means $\boldsymbol{\mu}_i^{(0)}$ are described above. First, we use the same isotropic covariance matrix, $\mathbf{C}_i = \sigma^2 \mathbf{I}_2$, for each proposal. We test different values of $\sigma \in \{0.5, 1, 2, 3, 5, 7, 10, 20, 70\}$, to gauge the performance of APIS. Then, we also try different non-isotropic diagonal covariance matrices, $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)$, where $\sigma_{i,j} \sim \mathcal{U}([1, 10])$ for $j \in \{1, 2\}$ and $i = 1, \dots N$, i.e., different for each proposal. We set $T = 2000$ and $T_a \in \{2, 5, 20, 50, 100\}$, i.e., $M = \frac{T}{T_a} \in \{20, 40, 100, 400, \frac{T}{2} = 1000\}$. We test the performance of APIS with the two initializations described above $\mathbf{In1}$ and $\mathbf{In2}$.

- *PIS (static APIS)*: we also consider the case $M = 1$, which corresponds to a static APIS technique with multiple proposals and no adaptation. PIS combines the deterministic mixture idea and the standard IS approach to build the global estimators (see Appendix A). For this reason, it is different from the standard multiple IS scheme, described below.
- *MAPIS*: for the MCMC interaction, we consider again a Gaussian proposal for the SMH method, i.e., $\varphi(\boldsymbol{\mu}) = \mathcal{N}(\mathbf{x}; [0, 0]^\top, \lambda^2 \mathbf{I}_2)$, with $\lambda = 10$. To maintain a constant computational cost in each simulation, we fix $\Upsilon = T_a = \frac{T}{M}$ (the number of iterations of SMH, at the end of each epoch), i.e., the total number of iterations of SMH in the entire MAPIS method is alway $M\Upsilon = T$.



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

Fig. 2. **(Ex-in-Sect 6)** MSE in log-scale versus the scale parameters, $\sigma_{i,j}$, in the estimation of the first component of the expected value of π. For the APIS and AMIS methods, we show the best and worst results. **(a)** Comparison among MIS (rhombus), PIS (triangles) and APIS. **(b)** Comparison between AMIS and APIS. **(c)** Comparison among M-PMC (rhombus), PMC with $N = 100$ (triangles) and APIS. **(d)** Comparison among PMC with $N = 500$ (X-marks), PMC with $N = 2000$ (triangles), APIS and MAPIS (squares).

Moreover, we compare these techniques with the following benchmark schemes:

• *Non-adaptive Multiple IS (MIS):* Given the initial $\boldsymbol{\mu}_i^{(0)}$, these positions never change as in PIS. We set $N = 100$. Thus, $T = 2000$ samples are drawn from each proposal in order to perform a fair comparison with APIS (in APIS we use $L = NT = 2 \cdot 10^5$ samples). The IS weights are built using the standard IS approach described in Appendix A.1.

• *AMIS scheme:* AMIS uses only one proposal pdf in the space fixing the temporal iteration index m, i.e.,

$$h_m(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Phi}_m), \quad m = 0, \ldots, M-1,$$

Both parameters $\boldsymbol{\mu}_m$ and $\boldsymbol{\Phi}_m$ are updated after each iteration. Note that we have used M as the number of adaptive iterations in AMIS since it is equivalent to the number of epochs M used in APIS. The initial mean $\boldsymbol{\mu}_0$ is chosen according to $\mathbf{In1}$ and $\mathbf{In2}$, whereas $\boldsymbol{\Phi}_0 = \sigma^2\mathbf{I}_2$ with $\sigma \in \{0.5, 1, 2, 3, 5, 7, 10, 20, 70\}$. At each iteration m, K samples are drawn from $h_m(\mathbf{x})$. Then, IS weights are associated to these samples using the deterministic mixture idea, taking into account all the previous proposals $h_0(\mathbf{x})$, $h_1(\mathbf{x})$,..., $h_{m-1}(\mathbf{x})$. Therefore, the weights associated to previous samples need to be updated as well. For these reasons, AMIS is more costly than APIS. Then, the parameters $\boldsymbol{\mu}_m$ and $\boldsymbol{\Phi}_m$ are updated according to the IS estimation of the mean and variances of the target. We have considered values of K and M such that $L = KM = NT = 2 \cdot 10^5$, for a fair comparison with APIS. Specifically, we have run different simulations using $K \in \{500, 1000, 2000, 5000\}$ and, as a consequence, $M \in \{40, 20, 10, 4\}$. Obviously, AMIS becomes more costly when M increases. However, depending on the starting value σ, the best results of AMIS in this scenario are usually provided by $M \in \{4, 10\}$ (i.e., $K = 5000$ and $K = 2000$). This is due to the fact that better estimations of the mean and covariance of the target are achieved, so that the adaptation is also improved.

- *PMC schemes:* we also apply the mixture PMC scheme [17]. More precisely, we consider a population of samples

$$\{\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)}\},$$

at the t-th iteration, and propagate them with random walks

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \epsilon_t, \quad i = 1, \ldots, N,$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{x}; [0,0]^\top, \mathbf{\Phi})$, with $\mathbf{\Phi} = \sigma^2 \mathbf{I}_2$ and $\sigma \in \{0.5, 1, 2, 5, 10, 20, 70\}$. At each iteration, the resampling step is performed according to the normalized importance weights. The initial cloud $\{\mathbf{x}_i^{(0)}\}_{i=1}^N$ is chosen according to the same initialization procedure described for $\mathbf{In1}$ and $\mathbf{In2}$. The cumulative mean of the cloud $\{\mathbf{x}_i^{(0)}\}_{i=1,t=1}^{N,T}$, as well as the cumulative estimate of the normalizing constant, are computed until $T = 2000$. We have not been able to apply the adaptive strategy suggested in [17] in order to select suitable scale parameters, within a population of pre-chosen values, since it has been difficult to select these values adequately. More specifically, we have not been able to find a set of parameters for this approach that provides reasonable results in this scenario. We have set $N = 100$ for a fair comparison with APIS, using the same total number of samples $L = NT$. Moreover, we have also run other simulations with $N = 500, 2000$ in order to see the computational cost needed to reach the performance of APIS. Finally, we have also considered a Modified PMC (M-PMC) that, similarly to [19, 20], uses the deterministic mixture for the spatial construction of the global estimator as in APIS. The results are shown in Tables 3-4.

6.4. Results

All the results are averaged over 2000 independent experiments. Tables 3 and 4 show the Mean Square Error (MSE) in the estimation of the mean (first component), with the initialization **In1** and **In2**, for the different algorithms. In AMIS, for the sake of simplicity, we only show the worst and best results among the several simulations made with different parameters (see the detailed description above). The results of MAPIS and PMC with $N \in \{500, 2000\}$ are included in two different subtables since their application entails more computational effort. In each subtable, the best results in each column are highlighted in bold-face.

We can observe that APIS outperforms the other techniques, except for a few values of σ, where APIS has a negligibly larger error. Only with $\sigma = 70$, AMIS has an MSE sensibly smaller than APIS in its best case. However, this result depends strictly on the choice of the parameter: the MSE of AMIS in its worst case is the highest whereas APIS provides always small MSE regardless of the choice of T_a. Moreover, for high values of $\sigma \in \{10, 20, 70\}$, the results of APIS could be easily improved using a higher value of T_a (for instance, $T_a = 500$). Observe also that the robust implementation, choosing randomly the scale parameters $\sigma_{i,j} \sim \mathcal{U}([1, 10])$, provides the best results (with the exception of PMC with $N = 2000$ which provides negligibly smaller MSE, with much higher computational cost). Moreover, MAPIS in general improves the results and the robustness of APIS, although at the expense of a higher computational cost due to the additional MCMC steps. Figure 2 depicts the MSE in log-scale of the estimation of the mean of π versus the choice of the scale parameters $\sigma_{i,j}$, comparing the different techniques.

7. NUMERICAL COMPARISONS IN HIGHER DIMENSION

In this section, we investigate there performance of APIS and MAPIS in higher dimensional problems. As a target density, we consider a mixture of Gaussians $\bar{\pi}(\mathbf{x}) = \frac{1}{3}\sum_{k=1}^3 \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_k, \mathbf{\Sigma}_k)$, with $\mathbf{x} \in \mathbb{R}^{D_x}$, $\boldsymbol{\nu}_k = [\nu_{k,1}, \ldots, \nu_{k,D_x}]^\top$ and $\mathbf{\Sigma}_k = \xi_k \mathbf{I}_{D_x}$, $k = 1, 2, 3$, where $\mathbb{I}_{D_x}$ is the $D_x \times D_x$ identity matrix. In this example, we consider two different cases: $D_x \in \{10, 30\}$. We use Gaussian proposal densities for all the analyzed methodologies: we compare APIS and MAPIS, with PMC and AMIS as in the previous examples. Furthermore, here we also test the mixture AIS scheme in [21]. In this method, weights, means and covariances of a mixture of Gaussians (with a fixed number of component denoted as J) are adapted.

We have tried different combinations of parameters keeping fixed the total number of samples, $L = 4 \cdot 10^5$. The initial means are selected randomly $\mu_i^{(0)} \sim \mathcal{U}([-W \times W]^{D_x})$, for $i = 1, \ldots, N$, and for all the techniques. We set $W = 10$ for $D_x = 10$, whereas $W = 6$ for $D_x = 30$. For APIS, we test $N \in \{10, 100, 200, 500, 10^3\}$ and for PMC $N \in \{100, 200, 10^3, 10^4\}$. For the method in [21], we use $J \in \{10, 20, 100, 500\}$. In APIS, we also test different values of $T_a \in \{100, 200, 500\}$. We recall that in AMIS there is only one proposal. In AMIS, we test different values of samples per iteration $K \in \{500, 10^3, 5 \cdot 10^3, 10^4, 2 \cdot 10^4, 10^5\}$.

We use different initial covariance matrices, $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,10}^2)$. We choose randomly at each run the values of $\sigma_{i,j} \sim \mathcal{U}([1, Q])$, for all $i = 1, \ldots, N$, and $j = 1, \ldots, D_x$. We test $Q = 11$ for $D_x = 10$ whereas $Q \in \{6, 11\}$ for $D_x = 30$.

The total number of iterations is chosen adequately for each simulation in order to keep the computational effort fixed to $L = 4 \cdot 10^5$. For MAPIS, we consider again a Gaussian proposal for the SMH method, i.e., $\varphi(\boldsymbol{\mu}) = \mathcal{N}(\mathbf{x}; [0,0]^\top, \lambda^2 \mathbf{I}_2)$, with $\lambda = 5$. To keep a constant computational cost in each simulation of MAPIS, we fix $\Upsilon = T_a$, that is the number of iterations of SMH. Thus, the total steps of SMH are $\Upsilon M = T_a M = T$. Hence, in MAPIS the total number of evaluations of the target is $L' = T + L$ (where $L = 4 \cdot 10^5$ and T is chosen differently in each run in order to keep L fixed).

7.1. Target specifications for $D_x = 10$

In this case, we set $\nu_{1,j} = 6$, $\nu_{2,j} = -5$ with $j = 1, \ldots, 10$, and $\boldsymbol{\nu}_3 = [1,2,3,4,5,5,4,3,2,1]^\top$. Moreover, we set $\xi_k = 3$, for all $k = 1,2,3$. The expected value of the target $\bar{\pi}(\mathbf{x})$ is $E[\mathbf{X}] = \left[\frac{2}{3}, 1, \frac{4}{3}, \frac{5}{3}, 2, 2, \frac{5}{3}, \frac{4}{3}, 1, \frac{2}{3}\right]^\top$, where $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$.

7.2. Target specifications for $D_x = 30$

For $D_x = 30$, we set $\nu_{1,j} = -5$, $\nu_{2,j} = 3$ and $\nu_{3,j} = 6$, with $j = 1, \ldots, 30$. We set again $\xi_k = 3$, for all $k = 1,2,3$. In this case, $E[\mathbf{X}] = \left[\frac{4}{3}, \ldots, \frac{4}{3}\right]^\top$.

7.3. Results

For each combination of parameters, we have run 10^3 independent simulations and compute the mean square error (MSE) in the estimation of $E[\mathbf{X}]$ (we have averaged the MSEs of all the components). The best and the averaged results in terms of MSE are shown in Table 5. With $D_x = 10$, PMC provides the minimum MSE but APIS obtains the best averaged results. AMIS suffers in this multimodal scenario since it often converges to a specific mode. On the contrary, with $D_x = 30$, AMIS provides the best results. However, in both cases, APIS provides results close to the best performance. The results also show that MAPIS is more robust than APIS, but at the expense of an increased computational effort.

Dim. D_x	Results	PMC	AMIS	AIS in [21]	APIS	MAPIS
10	**best**	**0.0858**	13.70	7.68	0.3857	0.3213
	average	7.52	16.92	10.92	**2.82**	**2.06**
30	**best**	8.34	**4.75**	8.81	5.58	4.81
	average	17.35	**8.73**	13.46	10.20	9.17

Table 5. **(Ex-in-Sect 7)** Averaged and best results in terms of MSE.

8. COMPARISON WITH PARTICLE SPLITTING METHODS

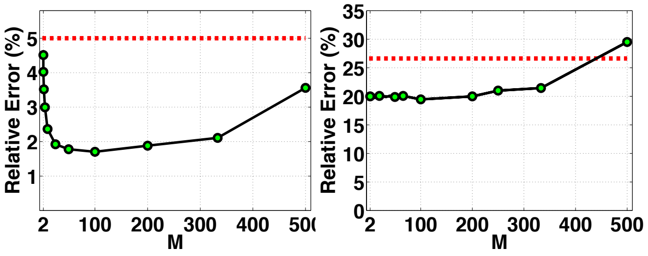
8.1. Bivariate bimodal target function

Let us consider the following target pdf used in [15, pages 495-498],

$$\bar{\pi}(x_1, x_2) \propto \exp\left(-\frac{x_1^2 + x_2^2 + (x_1 x_2)^2 - 24 x_1 x_2}{2}\right),$$

$x_1, x_2 \in \mathbb{R}$. The goal is to compute the normalizing constant $Z = \frac{1}{2.539 \cdot 10^{26}} = 2.825 \cdot 10^{-27}$ (approximated via an exhaustive deterministic method). The authors in [15] apply a particle splitting technique with a computational effort equivalent to $L \approx 1.2 \cdot 10^5$ samples (as stated in [15, page 498]), obtaining an averaged relative error of 5%.

We apply APIS with $N = 100$ Gaussian proposal pdfs and $T = 10^3$, so that $L = NT = 10^5$. In each run, the initial means are chosen randomly $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-6,6] \times [-6,6])$. The covariance matrices are also chosen randomly $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)$, with $\sigma_{i,j} \sim \mathcal{U}([1,6])$, $i = 1, \ldots, N$ and $j = 1, 2$. We run 10^3 different simulations with different epochs $2 \leq M \leq \frac{T}{2}$ (recall $M = \lfloor \frac{T}{T_a} \rfloor$). Figure 3(a) shows the percentage of relative error obtained with APIS (solid line) and particle splitting method (dashed line). APIS outperforms the technique in [15, pages 495-498] for every value of M. Furthermore with $M = 1$, corresponding to the (static) PIS method, we obtain a relative error of 6%. In the other extreme case, with $M = T$, the movements of the means $\boldsymbol{\mu}_i^{(m)}$ are random walks (i.e., non-driven movements) producing a relative error of 78% (this shows the effectiveness in the learning movements of APIS).



(a) (b)

Fig. 3. **(Ex-in-Sect 8.1-8.2)** Relative error (%) obtained with APIS as a function of the number of epochs $M = \lfloor \frac{T}{T_a} \rfloor$ (solid line) and the particle splitting method [15, pages 495-498] (dashed line), **(a)** for the example in Section 8.1 and **(b)** for the example in Section 8.2.

8.2. Logistic model

Consider a set of binary observations $y_k \in \{0, 1\}$ and the the likelihood function

$$\ell(\mathbf{y}|\mathbf{x}) \propto \prod_{k}^{D_y} p_k^{y_k}(1-p_k)^{1-y_k}, \quad p_k = \left(1+\exp(-\mathbf{s}_k^\top \mathbf{x})\right)^{-1},$$

with $\mathbf{x} = [x_1, x_2, x_3]^\top \in \mathbb{R}^3$ and where $\mathbf{s}_k = [s_{k,1}, s_{k,2}, s_{k,3}]^\top$ is the k-th explanatory variable. We use a Gaussian prior $g(\mathbf{x}) \propto \exp\left(-\frac{1}{2\xi^2}\mathbf{x}^2\right)$ with $\xi = 10$. We generate $D_y = 100$ artificial data $\mathbf{y}^* = [y_1^*, \ldots, y_{D_y}^*]^\top$ from this model given $\mathbf{x}^* = [1, -5.5, 1]^\top$ (vectors $\mathbf{s}_k$ are generated as in [15, page 500]). Let us consider the posterior pdf

$$\bar{\pi}(\mathbf{x}|\mathbf{y}^*) \propto \ell(\mathbf{y}^*|\mathbf{x})g(\mathbf{x}).$$

The goal consists on computing $E[\mathbf{X}]$ with $\mathbf{X} \sim \bar{\pi}(\mathbf{x}|\mathbf{y}^*)$, as an estimate of $\mathbf{x}^*$. We compare the ADAM technique in [36, 14, 15][6] with APIS. We use $N = 10$ Gaussian pdfs in APIS and $T \in 10^3$ (recall that $L = NT = 10^4$ total samples).[7] The initial means are chosen randomly $\mu_i^{(0)} \sim \mathcal{U}([-6 \times 6]^3)$, for $i = 1, \ldots, N$, in each simulation. The covariance matrices $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2, \sigma_{i,3}^2)$ are also chosen randomly in each run, $\sigma_{i,j} \sim \mathcal{U}([1, 6])$. We test APIS considering different number of epochs $M = \lfloor \frac{T}{T_a} \rfloor$.

Figure 3(b) depicts the percentage of the relative error (averaged over the 3 components) for APIS (solid line) and ADAM (dashed line). APIS, in general, outperforms ADAM (for several values of M). The results are averaged over 10^3 independent runs.

9. LOCALIZATION PROBLEM IN A WIRELESS SENSOR NETWORK

We consider the problem of positioning a target in a 2-dimensional space using range measurements. This is a problem that appears frequently in localization applications in wireless sensor networks [37, 38, 39]. Namely, we consider a random vector $\mathbf{X} = [X_1, X_2]^\top$ to denote the target position in the plane $\mathbb{R}^2$. The position of the target is then a specific realization $\mathbf{X} = \mathbf{x}$. The range measurements are obtained from 3 sensors located at $\mathbf{h}_1 = [-10, 2]^\top$, $\mathbf{h}_2 = [8, 8]^\top$ and $\mathbf{h}_3 = [-20, -18]^\top$. The observation equations are given by

$$Y_j = a \log\left(\frac{||\mathbf{x} - \mathbf{h}_j||}{0.3}\right) + \Theta_j, \quad j = 1, \ldots, 3, \tag{19}$$

where Θ_j are independent Gaussian random variables with identical pdfs, $\mathcal{N}(\vartheta_j; 0, \omega^2)$, $j = 1, 2$. We also consider a prior density over ω, i.e., $\Omega \sim p(\omega) = \mathcal{N}(\omega; 0, 25)I(\omega > 0)$, where $I(\omega > 0)$ is 1 if $\omega > 0$ and 0 otherwise. The parameter $A = a$ is

[6] We use the code provided directly by the authors in [15, pages 500-503] considering the generated $D_y = 100$ observations and only one run of ADAM in each simulation.

[7] The authors in [15] use 10^3 particles but it is not straightforward to compute the overall computational effort. The computational cost of ADAM in [15, pages 500-503] is $L \geq 10^4$ (in terms of evaluations of the target) since they also applied a Newton-Raphson method before running the algorithm and at least 10 steps of an hybrid MCMC technique before each iteration of ADAM.

also unknown and we again consider a Gaussian prior $A \sim p(a) = \mathcal{N}(a; 0, 25)$. Moreover, we also apply Gaussian priors over $\mathbf{X}$, i.e., $p(x_i) = \mathcal{N}(x_i; 0, 25)$ with $i = 1, 2$. Thus, the posterior pdf $\pi(x_1, x_2, a, \omega) = p(x_1, x_2, a, \omega|\mathbf{y})$ is

$$\pi(x_1, x_2, a, \omega) \propto \ell(\mathbf{y}|x_1, x_2, a, \omega)p(x_1)p(x_2)p(a)p(\omega),$$

where $\mathbf{y} \in \mathbb{R}^{D_y}$ is the vector of received measurements. We simulate $d = 30$ observations from the model ($D_y/3 = 10$ from each of the three sensors) fixing $x_1 = 3$, $x_2 = 3$, $a = -20$ and $\omega = 5$. With $D_y = 30$, the expected value of the target ($E[X_1] \approx 2.8749$, $E[X_2] \approx 3.0266$, $E[A] \approx 5.2344$, $E[\Omega] \approx 20.1582$)[8] is quite close to the true values.

Our goal is computing the expected value of $(X_1, X_2, A, \Omega) \sim \pi(x_1, x_2, a, \omega)$ via Monte Carlo, in order to provide an estimation of the position of the target, the parameter a and the standard deviation ω of the noise in the system. We apply APIS and PMC schemes both using N Gaussian proposals as in the previous example. For both algorithms, we initialize the cloud of particles spread throughout the space of the variables of interest, i.e.,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, 30^2\mathbf{I}_4), \quad i = 1, ..., N,$$

and the scale parameters $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,4}^2)\mathbf{I}_4$ with $i = 1, \ldots, N$. The values of the standard deviations $\sigma_{i,j}$ are chosen randomly for each Gaussian pdf. Specifically, $\sigma_{i,j} \sim \mathcal{U}([1, Q])$, where we have considered three possible values for Q, i.e., $Q \in \{5, 10, 30\}$.

The MSE of the estimations (averaged over 3000 independent runs) are provided in Tables 6 and 7 for different values of $N \in \{50, 100, 200\}$, $T \in \{1000, 2000, 4000\}$ and $T_a \in \{20, 100\}$. More specifically, in Table 6, we maintain fixed $T = 2000$ whereas in Table 7 we keep fixed the total number of generated samples $NT = 2\ 10^5$. APIS outperforms always PMC when $\sigma_{i,j} \sim \mathcal{U}([1, 5])$ and $\sigma_{i,j} \sim \mathcal{U}([1, 10])$ whereas PMC provides better results for $\sigma_{i,j} \sim \mathcal{U}([1, 30])$ (with the exception of the case $N = 200$ and $T = 2000$ in Table 6). This is owing to APIS needs the use of a greater value of T_a with bigger variances. Therefore, the results show jointly the robustness and flexibility of the APIS technique.

Std **Alg.**			$\sigma_{i,j} \sim \mathcal{U}([1,5])$	$\sigma_{i,j} \sim \mathcal{U}([1,10])$	$\sigma_{i,j} \sim \mathcal{U}([1,30])$
APIS	$N = 50$	$T_a = 20$	**0.0181**	**0.0768**	0.8298
		$T_a = 100$	0.0261	0.0770	0.6808
PMC		—	0.2067	0.5224	**0.2421**
APIS	$N = 100$	$T_a = 20$	**0.0073**	**0.0379**	0.5075
		$T_a = 100$	0.0147	0.0433	0.3702
PMC		—	0.0642	0.4345	**0.1533**
APIS	$N = 200$	$T_a = 20$	**0.0053**	**0.0174**	0.2816
		$T_a = 100$	0.0111	0.0229	**0.1886**
PMC		—	0.0136	0.2741	0.3455

Table 6. **(Ex-in-Sect 9)** MSE of the estimation of $E[(X_1, X_2, A, \Omega)]$ using APIS and PMC with $T = 2000$, for different random choices of the scale parameters and different number of particles N in the population. The best results, in each column and with the same number N, are highlighted with bold-faces.

10. CONCLUSIONS

In this work, we have introduced the adaptive population importance sampling (APIS) algorithm. APIS is an iterative importance sampling (IS) technique which uses multiple adaptive proposal pdfs. On the one hand, the deterministic mixture is used to build the partial IS estimators for the population of proposals in APIS, thus providing an increased robustness w.r.t. the population Monte Carlo (PMC) approach. On the other hand, the temporal evolution makes use of a standard IS estimator, thus avoiding the increase in computational cost as the algorithm evolves occurring in the adaptive multiple importance sampling (AMIS) scheme. Consequently, APIS is able to attain simultaneously the advantages of these two approaches (simplicity and robustness) while minimizing their drawbacks. Unlike PMC, APIS updates the proposal pdfs in an adaptive IS fashion, without using resampling. Hence, there is no loss of diversity in the mixture of proposals. Furthermore, by introducing an MCMC approach on top of APIS (thus leading to the so-called MAPIS algorithm) that diversity may be increased w.r.t. the initial population. Another advantage of APIS is that it is easily parallelizable, thus serving as the basis to develop distributed

[8] These values have been obtained with a deterministic, expensive and exhaustive numerical integration method, using a thin grid.

Std / Alg.				$\sigma_{i,j} \sim \mathcal{U}([1,5])$	$\sigma_{i,j} \sim \mathcal{U}([1,10])$	$\sigma_{i,j} \sim \mathcal{U}([1,30])$
APIS	$N = 50$	$T = 4000$	$T_a = 20$	**0.0067**	0.0330	0.4827
			$T_a = 100$	0.0069	**0.0269**	0.3078
PMC			—	0.2014	0.5151	**0.2214**
APIS	$N = 200$	$T = 1000$	$T_a = 20$	**0.0139**	**0.0498**	0.5800
			$T_a = 100$	0.0453	0.0947	0.5439
PMC			—	0.0153	0.2747	**0.3540**

Table 7. **(Ex-in-Sect 9)** MSE of the estimation of $E[(X_1, X_2, A, \Omega)]$ using APIS and PMC, for different random choices of the scale parameters, keeping constant the total number of drawn samples $NT = 2\ 10^5$. The best results, in each column and with the same number N, are highlighted with bold-faces.

importance sampling estimators. Numerical results confirm that APIS outperforms both techniques (AMIS and PMC) in terms of performance and robustness w.r.t. the choice of the initial parameters.

Acknowledgment

This work has been supported by the Spanish government's projects ALCIT (TEC2012-38800-C03-01), DISSECT (TEC2012-38058-C03-01), OTOSIS (TEC2013-41718-R) and COMPREHENSION (TEC2012-38883-C02-01); by the BBVA Foundation through project MG-FIAR ("I Convocatoria de Ayudas Fundación BBVA a Investigadores, Innovadores y Creadores Culturales"); by the ERC grant 239784 and AoF grant 251170; and by the European Union's 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861.

11. REFERENCES

[1] P. M. Djurić and S. J. Godsill, Eds., *Special Issue on Monte Carlo Methods for Statistical Signal Processing*, IEEE Transactions Signal Processing 50 (3), February 2002.

[2] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[3] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo Bayesian signal processing for wireless communications," *Journal of VLSI Signal Processing*, vol. 30, pp. 89–105, 2002.

[4] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[5] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

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

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

[8] K. H. Knuth, M. Habeck, N. K. Malakar, A. M. Mubeen, and B. Placek, "Bayesian evidence and model selection," *arXiv:1411.3013*, 2014.

[9] M. Evans and T. Swartz, "Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems," *Statistical Science*, vol. 10, no. 3, pp. 254–272, 1995.

[10] M. Evans and T. Swartz, *Approximating Integrals via Monte Carlo and Deterministic Methods*, Oxford University Press, Oxford (UK), 2000.

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

[12] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003.

[13] 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.

[14] Z. I. Botev and D. P. Kroese, "An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting," *Methodology and Computing in Applied Probability*, vol. 10, no. 4, pp. 471–505, December 2008.

[15] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods*, Wiley Series in Probability and Statistics, New York, 2011.

[16] J. Skilling, "Nested sampling for general Bayesian computation," *Bayesian Analysis*, vol. 1, no. 4, pp. 833–860, June 2006.

[17] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.

[18] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert, "Adaptive multiple importance sampling," *Scandinavian Journal of Statistics*, vol. 39, no. 4, pp. 798–812, December 2012.

[19] G.R. Douc, J.M. Marin, and C. Robert, "Convergence of adaptive mixtures of importance sampling schemes," *Annals of Statistics*, vol. 35, pp. 420–448, 2007.

[20] G.R. Douc, J.M. Marin, and C. Robert, "Minimum variance importance sampling via population Monte Carlo," *ESAIM: Probability and Statistics*, vol. 11, pp. 427–447, 2007.

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

[22] M. Bugallo, M. Hong, and P. M. Djuric, "Marginalized population Monte Carlo," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 2925–2928.

[23] M. Hong, M. F Bugallo, and P. M Djuric, "Joint model selection and parameter estimation by population monte carlo simulation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 3, pp. 526–539, 2010.

[24] E. Koblents and J. Miguez, "Robust mixture population Monte Carlo scheme with adaptation of the number of components," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*. IEEE, 2013, pp. 1–5.

[25] A. Owen and Y. Zhou, "Safe and effective importance sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 135–143, 2000.

[26] E. Veach and L. Guibas, "Optimally combining sampling techniques for Monte Carlo rendering," *In SIGGRAPH 1995 Proceedings*, pp. 419–428, 1995.

[27] L. Pozzi and A. Mira, "A R adaptive multiple importance sampling (ARAMIS)," *http://cran.r-project.org/web/packages/ARAMIS/vignettes/demoARAMIS.pdf*, 2012.

[28] J. Sirén, P. Marttinen, and J. Corander, "Reconstructing population histories from single nucleotide polymorphism data," *Mol Biol Evol*, vol. 28, no. 1, pp. 673–683, 2011.

[29] L. Martino, V. Elvira, D. Luengo, and J. Corander, "An adaptive population importance sampler," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 8088–8092, 2014.

[30] C. Berzuini and W. Gilks, "Resample-move filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 6. Springer, 2001.

[31] E. Koblents and J. Miguez, "A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models," *Statistics and Computing*, 2014.

[32] David W Scott, *Multivariate density estimation: theory, practice, and visualization*, vol. 383, John Wiley & Sons, 2009.

[33] H. Haario, E. Saksman, and J. Tamminen, “An adaptive Metropolis algorithm,” *Bernoulli*, vol. 7, no. 2, pp. 223–242, April 2001.

[34] J. M. Marin, P. Pudlo, and M. Sedki, “Consistency of the adaptive multiple importance sampling,” *arXiv:1211.2548*, 2012.

[35] Y. Iba, “Population Monte Carlo algorithms,” *Transactions of the Japanese Society for Artificial Intelligence*, vol. 16, pp. 279–286, 2001.

[36] Z. I. Botev, “An algorithm for rare-event probability estimation using the product rule of probability theory,” *Technical report, School of Mathematics and Physics, The University of Queensland*, 2008.

[37] 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.

[38] 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*, vol. 23, no. 4, pp. 809–819, April 2005.

[39] L. Martino and J. Míguez, “Generalized rejection sampling schemes and applications in signal processing,” *Signal Processing*, vol. 90, no. 11, pp. 2981–2995, November 2010.

A. IS APPROACHES USING WITH MULTIPLE PROPOSAL PDFS

Recall that our goal is computing efficiently some moment of $\mathbf{x}$, i.e., an integral measure w.r.t. the target pdf $\frac{1}{Z}\pi(\mathbf{x})$, $I = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$. Let us assume that we have two normalized proposal pdfs, $q_1(\mathbf{x})$ and $q_2(\mathbf{x})$, from which we intend to draw K_1 and K_2 samples respectively: $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_{K_1}^{(1)} \sim q_1(\mathbf{x})$ and $\mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_{K_2}^{(2)} \sim q_2(\mathbf{x})$. Then, there are at least two procedures to build a joint IS estimator: the standard importance sampling (IS) approach and the deterministic mixture (DM) IS technique. Both are briefly reviewed in the following.

A.1. Standard IS approach

The simplest approach [7, Chapter 14] is computing the classical IS weights:

$$w_i^{(1)} = \frac{\pi(\mathbf{x}_i^{(1)})}{q_1(\mathbf{x}_i^{(1)})}, \quad w_j^{(2)} = \frac{\pi(\mathbf{x}_j^{(2)})}{q_2(\mathbf{x}_j^{(2)})}, \tag{20}$$

with $i = 1, \ldots, K_1$ and $j = 1, \ldots, K_2$. The IS estimator is then built by normalizing them jointly, i.e., computing

$$\hat{I}_{IS} = \frac{1}{S_{tot}} \left(\sum_{i=1}^{K_1} w_i^{(1)} f(\mathbf{x}_i^{(1)}) + \sum_{j=1}^{K_2} w_j^{(2)} f(\mathbf{x}_j^{(2)}) \right), \tag{21}$$

where $S_{tot} = S_1 + S_2$ and the two partial sums are given by $S_1 = \sum_{i=1}^{K_1} w_i^{(1)}$ and $S_2 = \sum_{j=1}^{K_2} w_j^{(2)}$. Considering the normalized weights, $\bar{w}_i^{(1)} = \frac{w_i^{(1)}}{S_1}$ and $\bar{w}_j^{(2)} = \frac{w_j^{(2)}}{S_2}$, Eq. (21) can be rewritten as

$$\hat{I}_{IS} = \frac{1}{S_1 + S_2}\left(S_1 \hat{I}_1 + S_2 \hat{I}_2\right) = \frac{S_1}{S_1 + S_2}\hat{I}_1 + \frac{S_2}{S_1 + S_2}\hat{I}_2,$$

where $\hat{I}_1$ and $\hat{I}_2$ are the two *partial IS estimators*, obtained by considering only one proposal pdf. This procedure can be easily extended for $N > 2$ different proposal pdfs, obtaining the complete IS estimator as the convex combination of the N partial IS estimators:

$$\hat{I}_{IS} = \frac{\sum_{n=1}^{N} S_n \hat{I}_n}{\sum_{n=1}^{N} S_n}, \tag{22}$$

where $\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_{K_n}^{(n)} \sim q_n(\mathbf{x})$, $w_i^{(n)} = \pi(\mathbf{x}_i^{(n)})/q_n(\mathbf{x}_i^{(n)})$, $S_n = \sum_{i=1}^{K_n} w_i^{(n)}$ and $\hat{I}_n = \sum_{i=1}^{K_n} w_i^{(n)} f(\mathbf{x}_i^{(n)})$.

A.2. Deterministic mixture

An alternative approach is provided by the so-called *deterministic mixture* [25, 26]. For $N = 2$ proposals, setting

$$\mathbf{Z} = \left[\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_{K_1}^{(1)}, \mathbf{x}_1^{(2)}, \dots, \mathbf{x}_{K_2}^{(2)}\right],$$

with $\mathbf{x}_i^{(j)} \in \mathbb{R}^{D_x \times 1}$ ($j \in \{1, 2\}$ and $1 \leq i \leq K_j$) and $\mathbf{Z} \in \mathbb{R}^{D_x \times (K_1+K_2)}$, the weights are now defined as

$$w_i = \frac{\pi(\mathbf{Z}_i)}{\frac{K_1}{K_1+K_2} q_1(\mathbf{Z}_i) + \frac{K_2}{K_1+K_2} q_2(\mathbf{Z}_i)}, \tag{23}$$

with $\mathbf{Z}_i$ denoting the i-th column of $\mathbf{Z}$ for $i = 1, \dots, K_1 + K_2$. In this case, the *complete* proposal is considered to be a *mixture* of q_1 and q_2, weighted according to the number of samples drawn from each one. Note that, unlike in the standard procedure for sampling from a mixture, a deterministic and fixed number of samples are drawn from each proposal in the DM approach. However, it can be easily proved that the samples drawn in this *deterministic* way is exactly distributed according to the mixture $q(\mathbf{z}) = \frac{K_1}{K_1+K_2} q_1(\mathbf{z}) + \frac{K_2}{K_1+K_2} q_2(\mathbf{z})$ [25]. The DM estimator is finally given by

$$\hat{I}_{DM} = \frac{1}{S_{tot}} \sum_{i=1}^{K_1+K_2} w_i f(\mathbf{Z}_i),$$

where $S_{tot} = \sum_{i=1}^{K_1+K_2} w_i$ and the w_i are given by (23). For $N > 2$ proposal pdfs, the DM estimator can also be easily generalized:

$$\hat{I}_{DM} = \frac{1}{\sum_{j=1}^{K} w_j} \sum_{i=1}^{K} w_i f(\mathbf{Z}_i),$$

with $w_i = \pi(\mathbf{Z}_i) / (\sum_{n=1}^{N} \frac{K_n}{K} q_n(\mathbf{Z}_i))$ and $K = K_1 + K_2 + \ldots + K_N$. On the one hand, the DM approach is more stable than the IS method, thus providing a better performance in terms of a reduced variance of the corresponding estimator, as shown in the following section. On the other hand, it needs to evaluate every proposal K times (i.e., KN total evaluations) instead of K_n times (i.e., K total evaluations), and therefore is more costly from a computational point of view. However, this increased computational cost is negligible when the proposal is much cheaper to evaluate than the target, as it often happens in practical applications.

A.3. Comparison in terms of variance

In this section we prove that the variance of the DM estimator is always lower or equal than the variance of the IS estimator. For the sake of simplicity we focus on the case where $K_n = 1$ for $n = 1, \dots, N$ (and thus $K = N$), as this is the case in APIS, but this result can be easily extended to any value of K_n. We first prove the following lemma and then state our main theorem.

Lemma A.1 *Let $\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}) > 0$ for $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{D_x}$. Then, for all $\mathbf{x} \in \mathcal{X}$ and any α such that $0 \leq \alpha \leq 1$,*

$$\frac{1}{(1-\alpha)\varphi_1(\mathbf{x}) + \alpha\varphi_2(\mathbf{x})} \leq \frac{1-\alpha}{\varphi_1(\mathbf{x})} + \frac{\alpha}{\varphi_2(\mathbf{x})}. \tag{24}$$

Proof: Note that (24) is equivalent to

$$\frac{1}{(1-\alpha)\varphi_1(\mathbf{x}) + \alpha\varphi_2(\mathbf{x})} \leq \frac{(1-\alpha)\varphi_2(\mathbf{x}) + \alpha\varphi_1(\mathbf{x})}{\varphi_1(\mathbf{x})\varphi_2(\mathbf{x})}. \tag{25}$$

Defining $\beta = 1 - \alpha$, (25) can be rewritten as

$$\begin{aligned} \varphi_1(\mathbf{x})\varphi_2(\mathbf{x}) \leq & [\alpha\varphi_1(\mathbf{x}) + \beta\varphi_2(\mathbf{x})][\beta\varphi_1(\mathbf{x}) + \alpha\varphi_2(\mathbf{x})] \\ = & \beta^2\varphi_1(\mathbf{x})\varphi_2(\mathbf{x}) + \alpha\beta[\varphi_1(\mathbf{x})^2 + \varphi_2(\mathbf{x})^2] \\ & + \alpha^2\varphi_1(\mathbf{x})\varphi_2(\mathbf{x}). \end{aligned} \tag{26}$$

Rearranging terms in (26), we obtain

$$\begin{aligned}0 &\leq [\beta^2+\alpha^2-1]\varphi_1(\mathbf{x})\varphi_2(\mathbf{x})+\alpha\beta[\varphi_1(\mathbf{x})^2+\varphi_2(\mathbf{x})^2]\\ &=-2\alpha\beta\varphi_1(\mathbf{x})\varphi_2(\mathbf{x})+\alpha\beta[\varphi_1(\mathbf{x})^2+\varphi_2(\mathbf{x})^2]\\ &=\alpha\beta[\varphi_1(\mathbf{x})-\varphi_2(\mathbf{x})]^2,\end{aligned} \tag{27}$$

which is obviously verified, since $\alpha, \beta \geq 0$. □

Theorem A.2 *Consider a normalized target pdf,* $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$*, and* N *samples drawn from a set of* N *normalized proposal pdfs (one from each pdf),* $\mathbf{x}_i \sim q_i(\mathbf{x})$ *for* $i = 1, 2, \ldots, N$*. In this case, the standard importance sampling (IS) estimator and the deterministic mixture (DM) IS can be expressed as*

$$\hat{I}_{IS} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{x}_i)\pi(\mathbf{x}_i)}{q_i(\mathbf{x}_i)}, \quad \hat{I}_{DM} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{x}_i)\pi(\mathbf{x}_i)}{\frac{1}{N}\sum_{j=1}^{N}q_j(\mathbf{x}_i)}.$$

Moreover, we also consider N *independent samples* $\mathbf{z}_i \sim \frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})$*, drawn from the mixture in a standard way and the corresponding standard mixture estimator*

$$\hat{I}_{SM} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{z}_i)\pi(\mathbf{z}_i)}{\frac{1}{N}\sum_{j=1}^{N}q_j(\mathbf{z}_i)}.$$

The variance of the DM estimator is always lower or equal than the variance of the corresponding standard IS estimators, i.e.,

$$\mathit{Var}(\hat{I}_{DM}) \leq \mathit{Var}(\hat{I}_{SM}) \leq \mathit{Var}(\hat{I}_{IS}). \tag{28}$$

Proof: It is straightforward to see that

$$\text{Var}(\hat{I}_{DM}) \leq \text{Var}(\hat{I}_{SM}),$$

since the DM procedure follows a well-known variance reduction method (such as the *stratified sampling* [7, Chapter 4], [15, Chapter 9]). The variance of the IS estimator is [7]

$$\text{Var}(\hat{I}_{IS}) = \frac{1}{N^2}\sum_{i=1}^{N}\left(\int_{\mathcal{X}}\frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{q_i(\mathbf{x})}d\mathbf{x} - I^2\right), \tag{29}$$

where $I = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}$ is the true value of the integral that we want to estimate. The variance of the standard mixture estimator is given by

$$\text{Var}(\hat{I}_{SM}) = \frac{1}{N^2}\sum_{i=1}^{N}\left(\int_{\mathcal{X}}\frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{\frac{1}{N}\sum_{j=1}^{N}q_j(\mathbf{x})}d\mathbf{x} - I^2\right). \tag{30}$$

Substracting (30) and (29), we get

$$\begin{aligned}&\text{Var}(\hat{I}_{SM}) - \text{Var}(\hat{I}_{IS})\\ &= \int_{\mathcal{X}}\left[\frac{N}{\frac{1}{N}\sum_{j=1}^{N}q_j(\mathbf{x})} - \sum_{i=1}^{N}\frac{1}{q_i(\mathbf{x})}\right]f^2(\mathbf{x})\pi^2(\mathbf{x})d\mathbf{x} \leq 0,\end{aligned}$$

where the last inequality is required to fulfill (28). Hence, since $f^2(\mathbf{x})\pi^2(\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in \mathcal{X}$, in order to prove the theorem it is sufficient to show that

$$\frac{1}{\frac{1}{N}\sum_{j=1}^{N}q_j(\mathbf{x})} \leq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{q_i(\mathbf{x})}, \tag{31}$$

which can be easily proved by induction. Let us consider first the case $N = 2$, where (31) becomes

$$\frac{1}{\frac{1}{2}(q_1(\mathbf{x})+q_2(\mathbf{x}))} \leq \frac{1}{2}\left(\frac{1}{q_1(\mathbf{x})} + \frac{1}{q_2(\mathbf{x})}\right), \tag{32}$$

which can be obtained directly from Lemma A.1, setting $\alpha = \beta = \frac{1}{2}$, $\varphi_1(\mathbf{x}) = q_1(\mathbf{x})$ and $\varphi_2(\mathbf{x}) = q_2(\mathbf{x})$. Now, let us assume that (31) is true for $N-1$, i.e.,

$$\frac{1}{\frac{1}{N-1}\sum_{j=1}^{N-1} q_j(\mathbf{x})} \leq \frac{1}{N-1}\sum_{i=1}^{N-1}\frac{1}{q_i(\mathbf{x})}. \tag{33}$$

Then, for N we have

$$\begin{aligned}\frac{1}{\frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})} &= \frac{1}{\frac{N-1}{N}\frac{1}{N-1}\sum_{j=1}^{N-1} q_j(\mathbf{x}) + \frac{1}{N}q_N(\mathbf{x})}\\ &\leq \frac{(N-1)/N}{\frac{1}{N-1}\sum_{j=1}^{N-1} q_j(\mathbf{x})} + \frac{1/N}{q_N(\mathbf{x})},\end{aligned} \tag{34}$$

where we have applied Lemma A.1 with $\alpha = \frac{1}{N}$, $\beta = 1 - \frac{1}{N} = \frac{N-1}{N}$, $\varphi_1(\mathbf{x}) = \frac{1}{N-1}\sum_{j=1}^{N-1} q_j(\mathbf{x})$ and $\varphi_2(\mathbf{x}) = q_N(\mathbf{x})$. Finally, making use of (33) we obtain

$$\begin{aligned}\frac{1}{\frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})} &\leq \frac{N-1}{N}\frac{1}{N-1}\sum_{i=1}^{N-1}\frac{1}{q_i(\mathbf{x})} + \frac{1}{N}\frac{1}{q_N(\mathbf{x})}\\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{q_i(\mathbf{x})}.\end{aligned}$$

Thus Var($\hat{I}_{SM}$) $\leq$ Var($\hat{I}_{IS}$) and, as a consequence, Var($\hat{I}_{DM}$) $\leq$ Var($\hat{I}_{IS}$). □

B. IDEAL CONFIGURATION: PROPOSALS AND LOCATIONS

From a probabilistic point of view, APIS adaptive approach to update the i-th proposal, within the m-th epoch, can be summarized by the following two steps:

1. Draw a location parameter $\boldsymbol{\mu}_i^{(m)} \sim \phi_i^{(m)}(\boldsymbol{\mu})$.
2. Draw samples $\mathbf{z}_{i,t} \sim q_i^{(m)}(\mathbf{z}|\boldsymbol{\mu}_i^{(m)}, \mathbf{C}_i)$.

The pdf $\phi_i^{(m)}(\boldsymbol{\mu})$ is associated to the IS estimator used to update the mean of the i-th proposal at the m-th epoch,

$$\hat{\boldsymbol{\mu}}_i^{(m)} \propto \sum_{t=1}^{T_a}\frac{\pi(\mathbf{z}_{i,t})}{q_i^{(m-1)}(\mathbf{z}_{i,t})}\mathbf{z}_{i,t}. \tag{35}$$

Hence, $\phi_i^{(m)}(\boldsymbol{\mu})$ is the pdf of $\hat{\boldsymbol{\mu}}_i^{(m)}$ given in (35). This procedure leads to the following *equivalent proposal* pdf:

$$\begin{aligned}\widetilde{q}_i^{(m)}(\mathbf{z}|\mathbf{C}_i) &= \int_{\mathcal{X}} q_i^{(m)}(\mathbf{z}-\boldsymbol{\mu}|\mathbf{C}_i)\phi_i^{(m)}(\boldsymbol{\mu})d\boldsymbol{\mu}\\ &= q_i^{(m)}(\mathbf{z}|\mathbf{C}_i) * \phi_i^{(m)}(\boldsymbol{\mu}),\end{aligned} \tag{36}$$

where we have used the fact that $q_i^{(m)}(\mathbf{z}|\boldsymbol{\mu}, \mathbf{C}_i) = q_i^{(m)}(\mathbf{z}-\boldsymbol{\mu}|\mathbf{C}_i)$, since $\boldsymbol{\mu}_i^{(m)}$ is a location parameter, and $*$ denotes the D_x-dimensional linear convolution operator.

Ideally we would like to have $\widetilde{q}_i^{(m)}(\mathbf{z}|\mathbf{C}_i) = \bar{\pi}(\mathbf{z})$, since this proposal is optimal from the point of view of interpreting APIS as a kernel density estimator (as discussed in the text). Then, from (36) we would have

$$\bar{\pi}(\mathbf{z}) = q_i^{(m)}(\mathbf{z}|\mathbf{C}_i) * \phi_i^{(m)}(\boldsymbol{\mu}). \tag{37}$$

Eq. (37) can be rewritten, in terms of the characteristic functions $\bar{\Pi}(\boldsymbol{\nu}) = E[\bar{\pi}(\mathbf{x})e^{j\boldsymbol{\nu}\mathbf{x}}]$, $Q_i^{(m)}(\boldsymbol{\nu}|\mathbf{C}_i) = E[q_i^{(m)}(\mathbf{x}|\mathbf{C}_i)e^{j\boldsymbol{\nu}\mathbf{x}}]$, $\Phi_i^{(m)}(\boldsymbol{\nu}) = E[\phi_i^{(m)}(\mathbf{x})e^{j\boldsymbol{\nu}\mathbf{x}}]$, as

$$\bar{\Pi}(\boldsymbol{\nu}) = Q_i^{(m)}(\boldsymbol{\nu}|\mathbf{C}_i)\Phi_i^{(m)}(\boldsymbol{\nu}). \tag{38}$$

On the one hand, from (38) we note that the characteristic function of the proposals should ideally be given by

$$Q_i^{(m)}(\boldsymbol{\nu}|\mathbf{C}_i) = \frac{\bar{\Pi}(\boldsymbol{\nu})}{\Phi_i^{(m)}(\boldsymbol{\nu})}.$$

Moreover, since the IS estimator is known to be asymptotically unbiased and consistent [7], $\phi_i^{(m)}(\boldsymbol{\mu}) \to \delta(\boldsymbol{\mu}-\boldsymbol{\mu}_\pi)$ as $T_a \to \infty$ and thus $\Phi_i^{(m)}(\boldsymbol{\nu}) \to 1$. Consequently, in the limit we would have $Q_i^{(m)}(\boldsymbol{\nu}|\mathbf{C}_i) = \bar{\Pi}(\boldsymbol{\nu})$, and the proposals would have to be distributed exactly as the target ideally. On the other hand, the characteristic function associated to the optimal distribution of the means would be

$$\Phi_i^{(m)}(\boldsymbol{\nu}) = \frac{\bar{\Pi}(\boldsymbol{\nu})}{Q_i^{(m)}(\boldsymbol{\nu}|\mathbf{C}_i)}. \tag{39}$$

Particularizing this equation for $m = 0$ we would have the optimal distribution for the prior used to draw the initial means: $\Phi_i^{(0)}(\boldsymbol{\nu}) = \bar{\Pi}(\boldsymbol{\nu})/Q_i^{(0)}(\boldsymbol{\nu}|\mathbf{C}_i)$. Unfortunately, the optimal prior pdf cannot be obtained analytically in general.

C. DETAILED BALANCE CONDITION FOR SMH

For simplicity, in the following we remove the super-index in $\boldsymbol{\mu}_i$ denoting the current epoch, and add the iteration index τ of the SMH technique. Thus, we denote as

$$\mathcal{P}_{\tau-1} = \{\boldsymbol{\mu}_{1,\tau-1}, ..., \boldsymbol{\mu}_{N,\tau-1}\}$$

the population of means at the τ-th iteration. A sufficient condition for proving the ergodicity of the chain (generated by SMH) is given by the detailed balance condition.

Theorem C.1 *The chain yielded by SMH converges to the stationary pdf in Eq.* (16), $\bar{\pi}_g(\mathcal{P}_\tau) = \prod_{i=1}^N \bar{\pi}(\boldsymbol{\mu}_{i,\tau})$, *since the balance condition,*

$$\bar{\pi}_g(\mathcal{P}_{\tau-1})K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1}) = \bar{\pi}_g(\mathcal{P}_\tau)K(\mathcal{P}_{\tau-1}|\mathcal{P}_\tau), \tag{40}$$

is satisfied. The conditional probability $K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1})$ denotes the transition kernel of the SMH method.

Proof: For the case $\mathcal{P}_\tau \neq \mathcal{P}_{\tau-1}$ (the case $\mathcal{P}_\tau = \mathcal{P}_{\tau-1}$ is trivial), the kernel K can be expressed as

$$K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1}) = N\varphi(\boldsymbol{\mu}_{0,\tau})\frac{\frac{\varphi(\boldsymbol{\mu}_{j,\tau})}{\pi(\boldsymbol{\mu}_{j,\tau})}}{\sum_{i=1}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}\alpha(\mathcal{P}_{\tau-1}, \boldsymbol{\mu}_{0,\tau}),$$

where we have considered that the j-th mean has been selected as a candidate for replacement and α is given by Eq. (17). Since $j \in \{1, \dots, N\}$, for the interchangeability we have N equal probabilities (this is the reason of the factor N). Replacing the expression of α in Eq. (17), we obtain

$$\begin{aligned} K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1}) \quad = \quad & N\varphi(\boldsymbol{\mu}_{0,\tau})\frac{\frac{\varphi(\boldsymbol{\mu}_{j,\tau})}{\pi(\boldsymbol{\mu}_{j,\tau})}}{\sum_{i=1}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}} \times \\ & \frac{\sum_{i=1}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}{\sum_{i=0}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0\leq i\leq N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}, \end{aligned}$$

$$K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1}) = \frac{N}{\pi(\boldsymbol{\mu}_{j,\tau})}\frac{\varphi(\boldsymbol{\mu}_{0,\tau})\varphi(\boldsymbol{\mu}_{j,\tau})}{\sum_{i=0}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0\leq i\leq N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}.$$

Now we can also write

$$\begin{aligned} \bar{\pi}_g(\mathcal{P}_{\tau-1})K(\mathcal{P}_\tau|\mathcal{P}_{\tau-1}) \quad = \quad & \left[\prod_{i=1}^N \bar{\pi}(\boldsymbol{\mu}_i)\right]\frac{N}{\pi(\boldsymbol{\mu}_{j,\tau})} \times \\ & \frac{\varphi(\boldsymbol{\mu}_{0,\tau})\varphi(\boldsymbol{\mu}_{j,\tau})}{\sum_{i=0}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0\leq i\leq N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}, \end{aligned}$$

and defining $\gamma(P_{t-1}, \boldsymbol{\mu}_{0,\tau}) = \sum_{i=0}^{N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0 \leq i \leq N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}$, we have

$$\bar{\pi}_g(\mathcal{P}_{\tau-1}) K(\mathcal{P}_\tau | \mathcal{P}_{\tau-1}) = \frac{N}{Z} \left[\prod_{i=1 \neq j}^{N} \bar{\pi}(\boldsymbol{\mu}_i) \right] \frac{\varphi(\boldsymbol{\mu}_{0,\tau}) \varphi(\boldsymbol{\mu}_{j,\tau})}{\gamma(P_{\tau-1}, \boldsymbol{\mu}_{0,\tau})}.$$

This expression above is symmetric w.r.t. $\boldsymbol{\mu}_{0,\tau}$ and $\boldsymbol{\mu}_{j,\tau}$. Since $\mathcal{P}_{\tau-1}$ and $\mathcal{P}_\tau$ differ only in the elements $\boldsymbol{\mu}_{0,\tau}$ and $\boldsymbol{\mu}_{j,\tau}$ ($\mathcal{P}_{\tau-1}$ contains $\boldsymbol{\mu}_{j,\tau}$ whereas $\mathcal{P}_\tau$ contains $\boldsymbol{\mu}_{0,\tau}$), then $\bar{\pi}_g(\mathcal{P}_{\tau-1}) K(\mathcal{P}_\tau | \mathcal{P}_{\tau-1}) = \bar{\pi}_g(\mathcal{P}_\tau) K(\mathcal{P}_{\tau-1} | \mathcal{P}_\tau)$, which is precisely the detailed balance condition. □