Nearly Consistent Finite Particle Estimates in Streaming Importance Sampling

Nearly Consistent Finite Particle Estimates in Streaming Importance Sampling

Alec Koppel*, Amrit Singh Bedi*, Brian M. Sadler*, and Víctor Elvira†

***Abstract*—In Bayesian inference, we seek to compute information about random variables such as moments or quantiles on the basis of available data and prior information. When the distribution of random variables is intractable, Monte Carlo (MC) sampling is usually required. Importance sampling is a standard MC tool that approximates this unavailable distribution with a set of weighted samples. This procedure is asymptotically consistent as the number of MC samples (particles) go to infinity. However, retaining infinitely many particles is intractable. Thus, we propose a way to only keep a *finite representative subset* of particles and their augmented importance weights that is *nearly consistent*. To do so in an online manner, we (1) embed the posterior density estimate in a reproducing kernel Hilbert space (RKHS) through its kernel mean embedding; and (2) sequentially project this RKHS element onto a lower-dimensional subspace in RKHS using the maximum mean discrepancy, an integral probability metric. Theoretically, we establish that this scheme results in a bias determined by a compression parameter, which yields a tunable tradeoff between consistency and memory. In experiments, we observe the compressed estimates achieve comparable performance to the dense ones with substantial reductions in representational complexity.**

I. Introduction

Bayesian inference is devoted to estimating unknowns by considering as them random variables and constructing a posterior distribution. This posterior distribution incorporates the information of available observations (likelihood function) which is merged with prior knowledge about the unknown (prior distribution) [3]. Its application is widespread, spanning statistics [4], signal processing [5], machine learning [6], genetics [7], communications [8], econometrics [9], robotics [10], among many other examples. Bayesian inference is only possible in a very small subset of problems (e.g., when the underlying model between observations and the hidden state is linear and corrupted by Gaussian noise [11]). In some cases, it is possible to linearize nonlinear models and still obtain closed-form (but approximate) solutions [12]. Beyond linearity, Bayesian inference may be tackled by Gaussian Processes when smoothness and unimodality are present [13]. However, in most models of interest, closeed-form expressions are not available, and the posterior distribution of the unknowns must be approximated.

The standard approximation methodologies in Bayesian inference are either (a) Monte Carlo (MC) algorithms [14], which include Markov chain Monte Carlo (MCMC) methods [15], importance sampling (IS) [16], and particle filters (PFs) [17]; or (b) variational algorithms [18]. The later approach approximates the posterior by using an optimization algorithm to select within a parametrized family (e.g., by minimizing the KL divergence). Variational algorithms can optimize the parameters sequentially via the solution of a stochastic optimization problem [19]. In recent years, efforts to approximate the expectation by sampling have been investigated, e.g., see [20]. Unfortunately, unless the prior belongs to a simple exponential family, the parametric update is defined by a non-convex objective, meaning that asymptotic unbiasedness is mostly beyond reach. Mixture models have been considered [21], but their convergence is challenging to characterize and the subject of recent work on resampling [22].

In contrast, MC is a general approach to Bayesian inference based upon sampling (i.e., the simulation of samples/particles), and is known to generate (weighted) samples that eventually converge to the true distribution/quantity of interest [23], [24]. However, the scalability of Monte Carlo methods is still an ongoing challenge from several angles, in that to obtain consistency, the number of particles must go to infinity, and typically its scaling is exponential in the parameter dimension [25]. These scalability problems are the focus of this work, which we specifically study in the context of importance sampling (IS), a very relevant family of MC methods. We focus on IS, as compared with Markov chain Monte Carlo (MCMC), due to advantages such as, e.g., no burn-in period, simple parallelization [26], built-in approximation of the normalizing constant that is useful in many practical problems (e.g., model selection), and the ability to incorporate adaptive mechanisms without compromising its convergence [27]. IS methods approximate expectations of arbitrary functions of the unknown parameter via weighted samples generated from one or several proposal densities [28], [16]. Their convergence in terms of the integral approximation error, which vanishes as the number of samples increases, has been a topic of interest recently [28], [23], and various statistics to quantify their performance have been proposed [29].

Our goal is to alleviate the dependence of the convergence rate on the representational complexity of the posterior estimate. To do so, we propose projecting the posterior density onto a finite statistically significant subset of particles after every particle is generated.[1] However, doing so directly in a measure space is often intractable to evaluate. By embedding

A. Koppel and A.S. Bedi contributed equally to this work. A preliminary version of this work has appeared as [1], [2] with no proofs and a different (heuristic) projection rule for quantifying the difference between distributions.
*U.S. Army Research Laboratory, Adelphi, MD 20783, USA. E-mails: {alec.e.koppel.civ, brian.m.sadler6.civ}@mail.mil, amrit0714@gmail.com
†School of Mathematics, University of Edinburgh, Peter Guthrie Tait Road, Edinburgh, EH9 3FD United Kingdom: victor.elvira@ed.ac.uk

[1]Doing so may be generalized to scenarios where projection can be performed after generating a fixed number T of particles, a form of mini-batching, but this is omitted for simplicity.

this density in a reproducing kernel Hilbert space (RKHS) via its kernel mean embedding [30], we may compute projections of distributions via parametric computations involving the RKHS. More specifically, kernel mean embeddings extend the idea of feature mapping to spaces of probability distributions, which, under some regularity conditions [30, Sec. 3.8], admits a bijection between probability distributions and RKHS elements.

Contributions. Based upon this insight, we propose a compression scheme that operates online within importance sampling, sequentially deciding which particles are statistically significant for the integral estimation. To do so, we invoke the idea of distribution embedding [30] and map our unnormalized distributional estimates to RKHS, in contrast to [31], [32]. We show that the empirical kernel mean embedding estimates in RKHS are parameterized by the importance weights and particles. Then, we propose to sequentially project embedding estimates onto subspaces of the dictionary of particles, where the dictionary is greedily selected to ensure compressed estimates are close to the uncompressed one (according to some metric). This greedy selection is achieved with a custom variant of matching pursuit [33] based upon the Maximum Mean Discrepancy, which is an easily computable way to evaluate an integral probability metric by virtue of the RKHS mean embedding. The underpinning of this idea is similar to gradient projections in optimization, which has been exploited recently to surmount memory challenges in kernel and Gaussian process regression [34], [35].

We establish that the asymptotic bias of this method is a tunable constant depending on the compression parameter. These results yield an approach to importance reweighting that mitigates particle degeneracy, i.e., retaining a large number of particles with small weights [36], by directly compressing the embedding estimate of the posterior in the RKHS domain, rather than statistical tests that require sub-sampling in the distributional space [27], [37]. The compression is performed *online*, without waiting until the total number of samples N are available, which is typically impractical. Experiments demonstrate that this approach yields an effective tradeoff of consistency and memory, in contrast to the classical curse of dimensionality of MC methods.

Additional Context. Dimensionality reduction of nonparametric estimators been studied in disparate contexts. A number of works fix the sparsity dimension and seek the best N-term approximation in terms of estimation error. When a likelihood model is available, one terms the resulting active set a Bayesian *coresets* [38]. Related approaches called "herding" assume a fixed number of particles and characterize the resulting error in a kernel-smoothed density approximation [39], [40], [41], [42]. In these works, little guidance is provided on how to determine the number of points to retain.

In contrast, dynamic memory methods automatically tune the number of particles to ensure small model bias. For instance, in [43], a rule for retainment based on gradient projection error (assuming the likelihood is available) is proposed, similar to those arising in kernel regression [35]. Most similar to our work is the setting where a likelihood/loss is unavailable and one must resort to metrics on density estimates, i.e., statistical tests, for whether new particles are significant. Specifically, in particle filters, multinomial resampling schemes can be used with Chi-squared tests to determine whether the current number of particles should increase/decrease [44], [45]. The performance of these approaches have only characterized when their budget parameter goes to null or sparsity dimension goes to infinity. In contrast, in this work, we are especially focused on finite-sample analysis when budget parameters are left fixed, in order to elucidate the tradeoffs between memory and consistency, both in theory and practice.

II. Elements of Importance Sampling

In Bayesian inference [46][Ch. 7], we are interested in computing expectations

$$I(\phi) := \mathbb{E}_{\mathbf{x}}[\phi(\mathbf{x}) \,|\, \mathbf{y}_{1:K}] = \int_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x}) p(\mathbf{x}|\mathbf{y}_{1:K}) d\mathbf{x} \quad (1)$$

on the basis of a set of available observations $\mathbf{y}_{1:K} := \{\mathbf{y}_{1:K}\}_{k=1}^{K}$, where $\phi : \mathcal{X} \to \mathbb{R}$ is an arbitrary function, $\mathbf{x}$ is a random variable taking values in $\mathcal{X} \subset \mathbb{R}^p$ which is typically interpreted as a hidden parameter, and $\mathbf{y}$ is some observation process whose realizations $\mathbf{y}_{1:K}$ are assumed to be informative about parameter $\mathbf{x}$. For example, $\phi(\mathbf{x}) = \mathbf{x}$ yields the computation of the posterior mean, and $\phi(\mathbf{x}) = \mathbf{x}^p$ denotes the p-th moment. In particular, define the posterior density[2]

$$p\left(\mathbf{x} \,|\, \mathbf{y}_{1:K}\right) = \frac{p\left(\mathbf{y}_{1:K} \,|\, \mathbf{x}\right) p\left(\mathbf{x}\right)}{p\left(\mathbf{y}_{1:K}\right)}. \quad (2)$$

We seek to infer the posterior (2) with K data points $\mathbf{y}_{1:K}$ available at the outset. Even for this setting, estimating (2) has unbounded complexity [47], [48] when the form of the posterior is unknown. Thus, we prioritize efficient estimates of (2) from an online stream of samples from an *importance density* to be subsequently defined. Begin by defining posterior $q(\mathbf{x})$ and un-normalized posterior $\widetilde{q}(\mathbf{x})$ as

$$q(\mathbf{x}) = \widetilde{q}(\mathbf{x})/Z\,, \quad \widetilde{q}(\mathbf{x}) := \widetilde{q}(\mathbf{x} \,|\, \mathbf{y}_{1:K}) = p\left(\mathbf{y}_{1:K} \,|\, \mathbf{x}\right) p\left(\mathbf{x}\right), \quad (3)$$

where $\widetilde{q}(\mathbf{x})$ integrates to normalizing constant $Z := p\left(\mathbf{y}_{1:K}\right)$[3]. In most applications, we only have access to a collection of observations $\mathbf{y}_{1:K}$ drawn from a static conditional density $p(\mathbf{y}_{1:K} \,|\, \mathbf{x})$ and a prior for $p(\mathbf{x})$. Therefore, the integral (1) cannot be evaluated, and hence one must resort to numerical integration such as Monte Carlo methods. In Monte Carlo, we approximate (1) by sampling. Hypothetically, we could draw N samples $\mathbf{x}(n) \sim q(\mathbf{x})$ and estimate the expectation in (1) by the sample average

$$\mathbb{E}_{q(\mathbf{x})}[\phi(\mathbf{x})] \approx \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}(n)), \quad (4)$$

but typically it is difficult to obtain samples $\mathbf{x}(n)$ from posterior $q(\mathbf{x})$ of the random variable. To circumvent this

[2]Throughout, densities are with respect to the Lebesgue measure on $\mathbb{R}^p$.

[3]Note that $q(\mathbf{x})$ and $\widetilde{q}(\mathbf{x})$ depend on the data $\{\mathbf{y}_{1:K}\}_{k\leq K}$, although we drop the dependence to ease notation.

issue, define the *importance density* $\pi(\mathbf{x})$[4] with the same (or larger) support as true density $q(\mathbf{x})$, and multiply and divide by $\pi(\mathbf{x})$ inside the integral (1), yielding

$$\int_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x})q(\mathbf{x})d\mathbf{x} = \int_{\mathbf{x}\in\mathcal{X}} \frac{\phi(\mathbf{x})q(\mathbf{x})}{\pi(\mathbf{x})}\pi(\mathbf{x})d\mathbf{x}, \tag{5}$$

where the ratio $q(\mathbf{x})/\pi(\mathbf{x})$ is the Radon-Nikodym derivative, or unnormalized density, of the target q with respect to the proposal π. Then, rather than requiring samples from the true posterior, one may sample from the importance density $\mathbf{x}(n) \sim \pi(\mathbf{x}),\ n = 1, ..., N$, and approximate (1) as

$$\begin{aligned} I_N(\phi) :=& \frac{1}{N}\sum_{n=1}^{N} \frac{q(\mathbf{x}(n))}{\pi(\mathbf{x}(n))}\phi(\mathbf{x}(n)) \\ =& \frac{1}{NZ}\sum_{n=1}^{N} g(\mathbf{x}(n))\phi(\mathbf{x}(n)), \end{aligned} \tag{6}$$

where

$$g(\mathbf{x}(n)) \equiv \frac{\widetilde{q}(\mathbf{x}(n))}{\pi(\mathbf{x}(n))}, \tag{7}$$

are the importance weights. Note that (6) is unbiased, i.e., $\mathbb{E}_{\pi(\mathbf{x})}[I_N(\phi)] = \mathbb{E}_{q(\mathbf{x})}[\phi(\mathbf{x})]$ and consistent with N. Moreover, its variance depends on the importance density $\pi(\mathbf{x})$ approximation of the posterior [16].

Example priors and measurement models include Gaussian, Student's t, and uniform. Which choice is appropriate depends on the context [46]. The normalizing constant Z can be also estimated with IS as

$$\hat{Z} := \frac{1}{N}\sum_{n=1}^{N} g(\mathbf{x}(n)). \tag{8}$$

Note that in Eq. (6), the unknown Z can be replaced by $\hat{Z}$ in 8. Then, the new estimator is given by

$$\begin{aligned} I_N(\phi) :=& \frac{1}{N\hat{Z}}\sum_{n=1}^{N} g(\mathbf{x}(n))\phi(\mathbf{x}(n)) \\ =& \frac{1}{\sum_{j=1}^{N} g(\mathbf{x}(j))}\sum_{n=1}^{N} g(\mathbf{x}(n))\phi(\mathbf{x}(n)) \\ =& \sum_{n=1}^{N} \overline{w}(n)\phi(\mathbf{x}(n)), \end{aligned} \tag{9}$$

where the normalized $\overline{w}(n)$ weights are defined

$$\overline{w}(n) \equiv \frac{g(\mathbf{x}(n))}{\sum_{u=1}^{N} g(\mathbf{x}(u))}, \tag{10}$$

for all n. The whole IS procedure is summarized in Algorithm 1. The function $I_N(\phi)$ is the normalized importance sampling (NIS) estimator. It is important to note that the estimator $I_N(\phi)$ can be viewed as integrating a function ϕ with respect to density μ_N defined as

$$\mu_N(\mathbf{x}) := \sum_{n=1}^{N} \overline{w}(n)\delta_{\mathbf{x}(n)}, \tag{11}$$

[4]In general, the importance density could be defined over any observation process $\pi(\mathbf{x} \,|\, \{\mathbf{y}_k\})$, not necessarily associated with time indices $k = 1, \ldots, K$. We define it this way for simplicity.

Algorithm 1 IS: Importance Sampling with streaming samples

Require: Observation model $p(\mathbf{y} \,|\, \mathbf{x})$ and prior $p(\mathbf{x})$ or target density $q(\mathbf{x})$ (if known), importance density $\pi(\mathbf{x})$. Set of observations $\{\mathbf{y}_{1:K}\}_{k=1}^{K}$.

for $N = 0, 1, 2, \ldots$ **do**

Simulate one sample from importance dist. $\mathbf{x}(n) \sim \pi(\mathbf{x})$

Compute weight $g(\mathbf{x}(n))$ [cf. (6)]

Compute normalized weights $\overline{w}(n)$ by dividing by estimate for summand (8):

$$\overline{w}(n) = \frac{g(\mathbf{x}(n))}{\sum_{u=1}^{N} g(\mathbf{x}(u))} \quad \text{for all } n.$$

Estimate the expectation with the self-normalized IS as

$$I_N(\phi) = \sum_{n=1}^{N} \overline{w}(n)\phi(\mathbf{x}(n))$$

The posterior density estimate is given by

$$\mu_N = \sum_{n=1}^{N} \overline{w}(n)\delta_{\mathbf{x}(n)}$$

end for

which is called the particle approximation of q. Here $\delta_{\mathbf{x}(n)}$ denotes the Dirac delta measure evaluated at $\mathbf{x}(n)$. This delta expansion is one reason importance sampling is also referred to as a histogram filter, as they quantify weighted counts of samples across the space. Subsequently, we leave the argument (an event, or measurable subset) of the delta $\delta_{\mathbf{x}(n)}$ as implicit.

As stated in [28], [24], [23], for consistent estimates of (1), we require that N, the number of samples $\mathbf{x}(n)$ generated from the importance density, and hence the parameterization of the importance density, to go to infinity $N \to \infty$. Therefore, when we generate an infinite stream of particles, the parameterization of the importance density grows unbounded as it accumulates every particle previously generated. We are interested in allowing N, the number of particles, to become large (possibly infinite), while the importance density's complexity is moderate, thus overcoming an instance of the curse of dimensionality in Monte Carlo methods. In the next section, we propose a method to do so.

III. Compressing the Importance Distribution

In this section, we detail our proposed sequential compression scheme for overcoming the curse of dimensionality in importance sampling. However, to develop such a compression scheme, we first rewrite importance sampling estimates in vector notation to illuminate the dependence on the number of past particles generated. Then, because directly defining projections over measure spaces is intractable, we incorporate a distributional approximation called a mean embedding [30], over which metrics can be easily evaluated. This permits us to develop our main projection operator.

Begin by noting the curse of dimensionality in importance sampling can be succinctly encapsulated by rewriting the last step of Algorithm 1 in vector notation. Specifically, define $\mathbf{g}_n \in \mathbb{R}^n$, $\overline{\mathbf{w}}_n \in \mathbb{R}^n$ and $\mathbf{X}_n = [\mathbf{x}(1); \cdots \mathbf{x}(n)] \in \mathbb{R}^{p\times n}$.

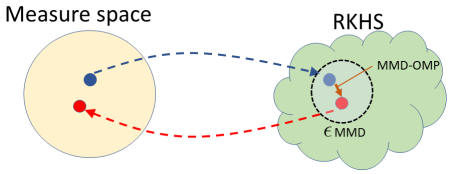


Fig. 1: Approximating the distributions via kernel mean embedding.

Then, after each new sample $\mathbf{x}(n)$ is generated from the importance distribution, we incorporate it into the empirical measure (11) through the parameter updates

$$\begin{aligned} \mathbf{g}_n &= [\mathbf{g}_{n-1}; g(\mathbf{x}(n))] \ , \\ \overline{\mathbf{w}}_n &= z_n \mathbf{g}_n \ , \qquad \mathbf{X}_n = [\mathbf{X}_{n-1}; \mathbf{x}(n)], \end{aligned} \tag{12}$$

where we define $z_n := 1/(\mathbf{1}_n^T \mathbf{g}_n)$ and $\mathbf{1}_n$ is the all ones column vector with dimension n. The unnormalized posterior density estimate parameterized by $\mathbf{g}_n$ and *dictionary* $\mathbf{X}_n$ is given by

$$\tilde{\mu}_n = \sum_{u=1}^{n} \mathbf{g}_n(u) \delta_{\mathbf{x}(u)}, \tag{13}$$

where we define $\mathbf{g}_n(u) := g(\mathbf{x}(u))$ is the importance weight (6) and $\delta_{\mathbf{x}(u)}$ is the Dirac delta function, both evaluated at sample $\mathbf{x}(u)$. Denote as $\Omega_{\mathbf{X}_n}$ the measure space "spanned" by Dirac measures centered at the samples stored in dictionary $\mathbf{X}_n$. More specifically, given measurable space $(\Omega, \Sigma, \lambda)$, where Ω is a set of outcomes, Σ is a σ-algebra whose elements are subsets, and $\lambda : \Sigma \to \mathbb{R}$ denotes the Lebesgue measure, we define the restricted σ-algebra as $\Sigma_{\mathbf{X}_n} = \{F \cap \{\mathbf{x}(u)\}_{u \leq n} : F \in \Sigma\}$. The measure space associated with $\Sigma_{\mathbf{X}_n}$ and the Lebesgue measure $\lambda_{\mathbf{X}_n}$ over this restricted σ-algebra we denote in shorthand as $\Omega_{\mathbf{X}_n}$ (see [49] for more details).

We note that the unnormalized posterior density in (13) is a linear combination of (nonnegative) Dirac measures with mass $\mathbf{g}_n(u)$ for each sample $\mathbf{x}(u)$. The question is how to select a subset of columns of $\mathbf{X}_n$ and modify the weights $\mathbf{g}_n$ such that with an infinite stream of $\mathbf{x}(n)$, we ensures the number of columns of $\mathbf{X}_n$ is finite and the empirical integral estimate tends to its population counterpart, i.e., the integral estimation error becomes very small (or goes to zero) as n tends to infinity [50]. Henceforth, we refer to the number of columns in matrix $\mathbf{X}_n$ which parameterizes (13) as the *model order* denoted by M_n.

A. Kernel Mean Embedding

In this subsection, we introduce kernel mean embedding which maps the measure estimate in the measure space $\Omega_{\mathbf{X}_n}$ to the corresponding reproducing kernel Hilbert space (RKHS) denoted by $\mathcal{H}_{\mathbf{X}_n}$ as shown in Fig. 1. There is a kernel associated with with RKHS defined as $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\kappa(\mathbf{x}, \cdot) \in \mathcal{H}_{\mathbf{X}_n}$. In order to appropriately select a subset of $\mathbf{X}_n$ which would define an approximator for $\tilde{\mu}_n$ in (13), we employ the notion of kernel mean embedding [30] of Dirac measures, and then perform approximation in the corresponding RKHS $\mathcal{H}_{\mathbf{X}_n}$. Doing so is motivated by the fact that operations involving distributions embedded in the RKHS may be evaluated in closed form, whereas in general measure spaces it is often intractable. The explicit value of the map for $\tilde{\mu}_n$ to RKHS is given by

$$\beta_n = \sum_{u=1}^{n} \mathbf{g}_n(u) \kappa(\mathbf{x}(u), \cdot), \tag{14}$$

and $\beta_n \in \mathcal{H}_{\mathbf{X}_n}$. We remark here that the associated kernel κ with RKHS is assumed to be a characteristic kernel which ensure that the mapping $\tilde{\mu}_n \to \beta_n$ is injective [51, Def. 3.2]. A characteristic kernel is imperative to make sure that $\|\beta_{\mathbb{P}} - \beta_{\mathbb{Q}}\|_{\mathcal{H}} = 0$ if and only if $\mathbb{P} = \mathbb{Q}$ for measures $\mathbb{P}$ and $\mathbb{Q}$, i.e., that it satisfies the identity of indiscernibles. This makes sure that there is no information loss by introducing the mapping via kernel mean embedding.

For practical purpose, we are interested in obtaining the value of the underlying posterior density associated with the mean embedding, which necessitates a way to invert the embedding. Doing so is achievable by computing the distributional pre-image [52], which for a given $\beta_n \in \mathcal{H}_{\mathbf{X}_n}$ is given by

$$\mathbf{g}^\star := \underset{\mathbf{g} \in \mathbb{R}^n}{\operatorname{argmin}} \left\| \beta_n - \sum_{u=1}^{n} \mathbf{g}(u) \kappa(\mathbf{x}(u), \cdot) \right\|_{\mathcal{H}}^2, \tag{15}$$

where $\sum_{u=1}^{n} \mathbf{g}(u) \kappa(\mathbf{x}(u), \cdot)$ is the kernel mean embedding for the Dirac measure $\sum_{u=1}^{n} \mathbf{g}(u) \delta_{\mathbf{d}_{\mathbf{x}(u)}}$ and $\mathbf{g} = \mathbf{g}^\star$ is obtained as the solution of (15). Therefore, for a given $\beta_n \in \mathcal{H}_{\mathbf{X}_n}$, we could recover the corresponding distribution in the measure space $\Omega_{\mathbf{X}_n}$ by solving (15). Note that (15) exhibits a closed form solution with $\mathbf{g}^\star = \mathbf{g}_n$, as the distributional measures have a Dirac measure structure. This motivates us to perform the compression in the RKHS $\mathcal{H}_{\mathbf{X}_n}$ parameterized by $\mathbf{X}_n$.

Specifically, given past particles collected in a dictionary $\mathbf{X}_n$, we seek to select the subset of columns of $\mathbf{X}_n$ to formulate its compressed variant $\mathbf{D}_n$. We propose to project the kernel mean embedding β_n onto subspaces $\mathcal{H}_{\mathbf{D}_n} = \text{span}\{\kappa(\mathbf{d}_u, \cdot)\}_{u=1}^{M_n}$ that consist only of functions that can be represented using dictionary $\mathbf{D}_n = [\mathbf{d}_1, \ldots, \mathbf{d}_{M_n}] \in \mathbb{R}^{p \times M_n}$. More precisely, $\mathcal{H}_{\mathbf{D}_n}$ is defined as a subspace in the RKHS $\mathcal{H}_{\mathbf{X}_n}$ that can be expressed as a linear combination of kernel evaluations at points $\{\mathbf{d}_u\}_{u=1}^{M_n}$. We enforce efficiency by selecting dictionaries $\mathbf{D}_n$ such that $M_n \ll n$. The eventual goal is to achieve a finite memory $M_n = \mathcal{O}(1)$ as $n \to \infty$ or $\frac{M_n}{n} \to 0$ as $n \to \infty$, while ensuring the underlying integral estimate has minimal bias, i.e., that we can obtain nearly consistent finite particle estimates.

B. Greedy Subspace Projections

Next, we develop a way to only retain statistically significant particles by appealing to ideas from subspace based projections [53], inspired by [35], [34]. To do so, we begin by rewriting the evolution of the mean embedding in (14) as

$$\beta_n = \beta_{n-1} + \mathbf{g}_n(n) \kappa(\mathbf{x}(n), \cdot). \tag{16}$$

We note that the update in (16) can be written as

$$\beta_n = \underset{f \in \mathcal{H}_{\mathbf{X}_n}}{\operatorname{argmin}} \ \| f - \big(\beta_{n-1} + \mathbf{g}_n(n) \kappa(\mathbf{x}(n), \cdot)\big) \|_{\mathcal{H}}^2 \ , \tag{17}$$

Algorithm 2 Compressed Kernelized IS (CKIS)

Require: Unnormalized target distribution $\widetilde{q}(\mathbf{x})$, importance distribution $\pi(\mathbf{x})$.
for $n = 0, 1, 2, \ldots, N$ **do**
Simulate one sample from importance dist. $\mathbf{x}(n) \sim \pi(\mathbf{x})$
Compute the importance weight $g(\mathbf{x}(n)) \equiv \frac{\tilde{q}(\mathbf{x}(n))}{\pi(\mathbf{x}(n))}$
Normalize weights $w(n)$ by estimate for summand (8):

$$\overline{w}(j) := \frac{w(j)}{z_n}, j = 1, ..., n. \, , z_n = \sum_{u=1}^{n} w(u)$$

Update the mean embedding via last sample & weight [cf. (16)]

$$\tilde{\beta}_n = \beta_{n-1} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot).$$

Append dictionary $\tilde{\mathbf{D}}_n = [\mathbf{D}_{n-1}; \mathbf{x}(n)]$ and importance weights $\tilde{\mathbf{g}}_n = [\mathbf{g}_{n-1}; g(\mathbf{x}(n))]$
Compress the mean embedding as (cf. Algorithm 3)

$$(\beta_n, \mathbf{D}_n, \mathbf{g}_n) = \textbf{MMD-OMP}(\tilde{\beta}_n, \tilde{\mathbf{D}}_n, \tilde{\mathbf{g}}_n, \epsilon_n)$$

Evaluate the pre-image to calculate $\hat{\mu}_n$ using (15)
Estimate the expectation as $\hat{I}_n = \sum_{u=1}^{|\mathbf{D}_n|} \overline{w}(u)\phi(\mathbf{x}(u))$
end for

where the equality employs the fact that β_n can be represented using only the elements in $\mathcal{H}_{\mathbf{X}_n} = \text{span}\{\kappa(\mathbf{x}(u), \cdot)\}_{u \leq n}$. Observe that (17) defines a projection of the update $\big(\beta_{\tilde{\mu}_{n-1}} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot)\big)$ onto the subspace defined by $\mathcal{H}_{\mathbf{X}_n}$, which we propose to replace at each iteration by a projection onto a subspace defined by dictionary $\mathbf{D}_n$, which is extracted from the particles observed thus far. The process by which we select $\mathbf{D}_n$ will be discussed next. To be precise, we replace the update (17) in which the number of particles grows at each step by the subspace projection onto $\mathcal{H}_{\mathbf{D}_n}$ as

$$\begin{aligned}\beta_n =& \underset{f \in \mathcal{H}_{\mathbf{D}_n}}{\text{argmin}} \ \|f - \big(\beta_{\tilde{\mu}_{n-1}} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot)\big)\|_{\mathcal{H}}^2 \\ :=& \mathcal{P}_{\mathcal{H}_{\mathbf{D}_n}}[\beta_{n-1} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot)]. \end{aligned} \tag{18}$$

Let us define $\tilde{\beta}_n := \beta_{n-1} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot)$, which means that $\beta_n = \mathcal{P}_{\mathcal{H}_{\mathbf{D}_n}}[\tilde{\beta}_n]$. Let us denote the corresponding dictionary update as

$$\begin{aligned}\tilde{\mathbf{g}}_n =& [\mathbf{g}_{n-1}; g(\mathbf{x}(n))] \, , \qquad \tilde{\mathbf{D}}_n = [\mathbf{D}_{n-1}; \mathbf{x}(n)] \\ \overline{\mathbf{w}}_n =& z_n \mathbf{g}_n \, , \end{aligned} \tag{19}$$

where $\mathbf{D}_{n-1}$ has M_{n-1} number of elements and $\tilde{\mathbf{D}}_n$ has $\tilde{M}_n = M_{n-1} + 1$. Using the expression for mean embedding in (14), we may write the projection in (18) as follows

$$\begin{aligned}\mathbf{g}_n :=& \underset{\mathbf{g} \in \mathbb{R}^{M_n}}{\text{argmin}} \Big\| \sum_{s=1}^{M_n} \mathbf{g}(s)\kappa(\mathbf{d}_s, \cdot) - \sum_{u=1}^{\tilde{M}_n} \tilde{\mathbf{g}}_n(u)\kappa(\tilde{\mathbf{d}}_u, \cdot) \Big\|_{\mathcal{H}}^2 \\ =& \underset{\mathbf{g} \in \mathbb{R}^{M_n}}{\text{argmin}} \Big(\mathbf{g}^T \mathbf{K}_{\mathbf{D}_n, \mathbf{D}_n} \mathbf{g} - 2\mathbf{g}^T \mathbf{K}_{\mathbf{D}_n, \tilde{\mathbf{D}}_n} \tilde{\mathbf{g}}_n + \tilde{\mathbf{g}}_n \mathbf{K}_{\tilde{\mathbf{D}}_n, \tilde{\mathbf{D}}_n} \tilde{\mathbf{g}}_n \Big), \end{aligned} \tag{20}$$

where we expand the square and define the kernel covariance matrix $\mathbf{K}_{\mathbf{D}_n, \tilde{\mathbf{D}}_n}$ whose $(s, u)^{\text{th}}$ entry is given by $\kappa(\mathbf{d}_s, \tilde{\mathbf{d}}_u)$. The other matrices $\mathbf{K}_{\tilde{\mathbf{D}}_n, \tilde{\mathbf{D}}_n}$ and $\mathbf{K}_{\mathbf{D}_n, \mathbf{D}_n}$ are similarly defined. The problem in (20) may be solved explicitly by computing gradients and solving for $\mathbf{g}_n$ to obtain

$$\mathbf{g}_n = \mathbf{K}_{\mathbf{D}_n, \mathbf{D}_n}^{-1} \mathbf{K}_{\mathbf{D}_n \tilde{\mathbf{D}}_n} \tilde{\mathbf{g}}_n. \tag{21}$$

Then, the projected estimate of the mean embedding $\tilde{\beta}_n$ is given by

$$\beta_n = \mathcal{P}_{\mathcal{H}_{\mathbf{D}_n}}[\tilde{\beta}_n] = \sum_{s=1}^{M_n} \mathbf{g}_n(s)\kappa(\mathbf{d}_s, \cdot), \tag{22}$$

where $\mathbf{g}_n$ is obtained as a solution to (21). Now, for a given dictionary $\mathbf{D}_n$, we know how to obtain the projected version of the mean embedding for each n. Next, we discuss the procedure to obtain $\mathbf{D}_n$ at each n.

Algorithm 3 MMD based Orthogonal Matching Pursuit (MMD-OMP)

Require: kernel mean embedding $\tilde{\beta}_n$ defined by dict. $\tilde{\mathbf{D}}_n \in \mathbb{R}^{p \times \tilde{M}_n}$, coeffs. $\tilde{\mathbf{g}} \in \mathbb{R}^{\tilde{M}_n}$, approx. budget $\epsilon_n > 0$
initialize $\beta = \tilde{\beta}_n$, dictionary $\mathbf{D} = \tilde{\mathbf{D}}_n$ with indices $\mathcal{I}$, model order $M = \tilde{M}_n$, coeffs. $\mathbf{g} = \tilde{\mathbf{g}}_n$.
while candidate dictionary is non-empty $\mathcal{I} \neq \emptyset$ **do**
for $j = 1, \ldots, M$ **do**
Find minimal approx. error with dictionary element $\mathbf{d}_j$ removed

$$\gamma_j = \text{MMD}\Big[\tilde{\beta}_n, \sum_{k \in \mathcal{I} \setminus \{j\}} \mathbf{g}(k)\kappa(\mathbf{d}_k, \cdot)\Big] \, .$$

end for
Find dictionary index minimizing approximation error: $j^\star = \arg\min_{j \in \mathcal{I}} \gamma_j$
if minimal approx. error exceeds threshold $\gamma_{j^\star} > \epsilon_n$
stop
else
Prune dictionary $\mathbf{D} \leftarrow \mathbf{D}_{\mathcal{I} \setminus \{j^\star\}}$, remove the columns associated with index $j^\star$
Revise set $\mathcal{I} \leftarrow \mathcal{I} \setminus \{j^\star\}$ and model order $M \leftarrow M - 1$.
Update weights $\mathbf{g}$ defined by current dictionary $\mathbf{D}$

$$\mathbf{g} = \underset{\mathbf{w} \in \mathbb{R}^M}{\text{argmin}} \big\| \tilde{\beta}_n - \sum_{u=1}^{M} \mathbf{w}(u)\kappa(\mathbf{d}_u, \cdot) \big\|_{\mathcal{H}}$$

end
end while
Assign $\mathbf{g}_n = \mathbf{g}$ and Evaluate the projected kernel mean embedding as $\beta_n = \sum_{u=1}^{M} \mathbf{g}_n(u)\kappa(\mathbf{d}_u, \cdot)$
return $\beta_n, \mathbf{D}_n, \mathbf{g}_n$ of complexity $M \leq \tilde{M}$ s.t. $\text{MMD}[\tilde{\beta}_n, \beta_n] \leq \epsilon_n$

C. Dictionary Update

The selection procedure for the dictionary $\mathbf{D}_n$ is based upon greedy sparse approximation, a topic studied in compressive sensing [54]. The function $\tilde{\beta}_n := \beta_{n-1} + \mathbf{g}_n(n)\kappa(\mathbf{x}(n), \cdot)$ is parameterized by dictionary $\tilde{\mathbf{D}}_n$, whose model order is $\tilde{M}_n = M_{n-1} + 1$. We form $\mathbf{D}_n$ by selecting a subset of M_n columns from $\tilde{\mathbf{D}}_n$ that are best for approximating the kernel mean embedding $\tilde{\beta}_n$ in terms of maximum mean discrepancy (MMD). As previously noted, numerous approaches are possible for sparse representation. We make use of destructive

orthogonal matching pursuit (OMP) [55] with allowed error tolerance ϵ_n to find a dictionary matrix $\mathbf{D}_n$ based on the one that includes the latest sample point $\tilde{\mathbf{D}}_n$. With this choice, we can tune the stopping criterion to guarantee the per-step estimates of mean embedding are close to each other. We name the compression procedure MMD-OMP and it is summarized in Algorithm 3. From the procedure in Algorithm 3, note that the projection operation in (18) is performed in a manner that ensures that $\text{MMD}[\tilde{\beta}_n, \beta_n] \leq \epsilon_n$ for all n, and we recall that β_n is the compressed version of $\tilde{\beta}_n$.

IV. Balancing Consistency and Memory

In this section, we characterize the convergence behavior of our posterior compression scheme. Specifically, we establish conditions under which the asymptotic bias is proportional to the kernel bandwidth and the compression parameter using posterior distributions given by Algorithm 2. To frame the discussion, we note that the NIS estimator (9) $I_N(\phi)$, whose particle complexity goes to infinity, is asymptotically consistent [56][Ch. 9, Theorem 9.2], and that the empirical posterior $\mu_N(\cdot)$ contracts to its population analogue at a $\mathcal{O}(1/N)$ rate where N is the number of particles. To establish consistency, we first detail the technical conditions required.

A. Assumptions and Technical Conditions

Assumption 1 *Recall the definition of the target distribution q from Sec. II (following (2)). Denote the integral of test function $\phi : \mathcal{X} \to \mathbb{R}$ as $q(\phi)$.*

(i) Assume that ϕ is absolutely integrable, i.e., $q(|\phi|) < \infty$, and has absolute value at most unit $|\phi| \leq 1$.

(ii) The test function has absolutely continuous second derivative, and $\int_{\mathbf{x}\in\mathcal{X}} \phi''(\mathbf{x})d\mathbf{x} < \infty$.

Assumption 2 *The kernel function associated with RKHS is such that $\int_{\mathbf{x}\in\mathcal{X}} \kappa_{\mathbf{x}^{(n)}}(\mathbf{x}) = 1$, $\int_{\mathbf{x}\in\mathcal{X}} \mathbf{x}\kappa_{\mathbf{x}^{(n)}}(\mathbf{x}) = 0$, and $\sigma_\kappa^2 = \int_{\mathbf{x}\in\mathcal{X}} \mathbf{x}^2\kappa_{\mathbf{x}^{(n)}}(\mathbf{x}) > 0$.*

Assumption 3 *Let $I_N(\phi)$ and $\hat{I}_N(\phi)$ be the integral estimators for test function ϕ associated with the uncompressed and compressed posterior densities. We define the approximation error for $\phi \notin \mathcal{F}$ where $\mathcal{F} := \{f \in \mathcal{H} \mid \|f\|_\mathcal{H} \leq 1\}$, as*

$$I(\phi, f) = \sup_{|\phi|\leq 1}\Big(\mathbb{E}[\hat{I}_N(\phi) - I_N(\phi)]\Big) - \sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f) - I_N(f)]\Big), \tag{23}$$

We assume that $I(\phi, f) \leq G$, where G is a finite constant.

Assumption 1(i) is a textbook condition in the analysis of Monte Carlo methods, and appears in [56]. Assumptions 1(ii) and 2 are required conditions for establishing consistency of kernel density estimates and are standard – see [57][Theorem 6.28]. We begin by noting that under Assumption 1, we have classical statistical consistency of importance sampling as the number of particles becomes large as stated in Lemma 2 in Appendix A. This result enables characterizing the bias of Algorithm 2, given next in Lemma 1. Assumption 3 is non-standard and we use it to bound the error due to continuity conditions imposed by operating in the RKHS. We note that if $\phi \in \mathcal{F}$ which is the case for most practical applications, then $G = 0$.

Lemma 1 *Define the second moment of the true unnormalized density ρ as in Lemma 2. Then, under Assumptions 1-3, the estimator of Alg. 2 satisfies*

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \sum_{n=1}^{N}\epsilon_n + \frac{24}{N}\rho + G. \tag{24}$$

where ϵ_n is the compression budget for each n.

Proof : Inspired by [23], begin by denoting $\hat{I}_N(\phi)$ as the integral estimate given by Algorithm 2. Consider the bias of the integral estimate $\hat{I}_N(\phi) - I(\phi)$, and add and subtract $I_N(\phi)$, the uncompressed normalized importance estimator that is the result of Algorithm 1, to obtain

$$\hat{I}_N(\phi) - I(\phi) = \hat{I}_N(\phi) - I_N(\phi) + I_N(\phi) - I(\phi). \tag{25}$$

Take the expectation on both sides with respect to the population posterior (2) to obtain

$$\mathbb{E}[\hat{I}_N(\phi) - I(\phi)] = \mathbb{E}[\hat{I}_N(\phi) - I_N(\phi)] + \mathbb{E}[I_N(\phi) - I(\phi)]. \tag{26}$$

Let's compute the sup of both sides of (26) over range $|\phi| \leq 1$ and use the fact that a sup of a sum is upper-bounded by the sum of individual terms:

$$\sup_{|\phi|\leq 1}\Big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\Big) \leq \sup_{|\phi|\leq 1}\Big(\mathbb{E}[\hat{I}_N(\phi) - I_N(\phi)]\Big) + \sup_{|\phi|\leq 1}([\mathbb{E}[I_N(\phi) - I(\phi)]). \tag{27}$$

Now add and subtract the supremum over the space $\mathcal{F} := \{f \in \mathcal{H} \mid \|f\|_\mathcal{H} \leq 1\}$ to the first term on the right hand side of (27) to write

$$\begin{aligned}&\sup_{|\phi|\leq 1}\Big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\Big) \qquad (28)\\ &\leq \sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f) - I_N(f)]\Big) + \sup_{|\phi|\leq 1}([\mathbb{E}[I_N(\phi) - I(\phi)])\\ &+ \underbrace{\sup_{|\phi|\leq 1}\Big(\mathbb{E}[\hat{I}_N(\phi) - I_N(\phi)]\Big) - \sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f) - I_N(f)]\Big)}_{I(\phi,f)\leq G}\end{aligned}$$

Observe that the last line on the right-hand side of the preceding expression defines the integral function approximation error $I(\phi, f)$ defined in (23), which is upper-bounded by constant G (Assumption 3). Now, compute the absolute value of both sides of (28), and to the second term on the first line, pull the absolute value inside the supremum. Doing so allows us to apply (49) (Lemma 2) to the second term on the first line, the result of which is:

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f) - I_N(f)]\Big) + \frac{24}{N}\rho + G, \tag{29}$$

where G is defined in Assumption 3 and note that $G = 0$ if $\phi \in \mathcal{H}$. It remains to address the first term on the right

hand side of (29), which noticeably defines an instance of an integral probability metric (IPM) [Muller 1997], i.e.,

$$\sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f)-I_N(f)]\Big) = \sup_{f\in\mathcal{F}}\left(\int f(\mathbf{x})d\hat{\mu}_N - \int f(\mathbf{x})d\tilde{\mu}_N\right), \quad (30)$$

where $\hat{\mu}_N$ is the pre-image unnormalized density estimate obtained by solving (15). The IPM in (30) is exactly equal to Maximum Mean Discrepancy (MMD)[58] for test functions in the RKHS $f \in \mathcal{H}$. This observation allows us to write

$$\sup_{f\in\mathcal{F}}\Big(\mathbb{E}[\hat{I}_N(f)-I_N(f)]\Big) = \|\beta_N - \gamma_N\|_{\mathcal{H}}. \quad (31)$$

where γ_N is the kernel mean embedding corresponding to the uncompressed measure estimate $\tilde{\mu}_N$. Next, consider the term $\|\beta_N - \gamma_N\|_{\mathcal{H}}$ and add and subtract the per step uncompressed estimate $\tilde{\beta}_N$ [cf. (19)] to obtain

$$\begin{aligned}\|\beta_N - \gamma_N\|_{\mathcal{H}} =& \|(\beta_N - \tilde{\beta}_N) + (\tilde{\beta}_N - \gamma_N)\|_{\mathcal{H}} \\ \leq& \|\beta_N - \tilde{\beta}_N\|_{\mathcal{H}} + \|\tilde{\beta}_N - \gamma_{\nu_N}\|_{\mathcal{H}} \\ \leq& \epsilon_N + \|\tilde{\beta}_N - \gamma_N\|_{\mathcal{H}}, \end{aligned} \quad (32)$$

where the last inequality holds from the fact that $\|\beta_N - \tilde{\beta}_N\|_{\mathcal{H}} \leq \epsilon_N$ (cf. Algorithm 3). Now we substitute the values of $\tilde{\beta}_N$ and γ_N using the update in (16), and we get

$$\|\beta_N - \gamma_N\|_{\mathcal{H}} \leq \epsilon_N + \|\beta_{N-1} - \gamma_{N-1}\|_{\mathcal{H}}. \quad (33)$$

Using the above recursion, we easily obtain

$$\|\beta_N - \gamma_N\|_{\mathcal{H}} \leq \sum_{n=1}^{N}\epsilon_n. \quad (34)$$

Hence, using the upper bound of (34) in (31), and then substituting the result into (29) yields

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \sum_{n=1}^{N}\epsilon_n + \frac{24}{N}\rho + G. \quad (35)$$

as stated in Lemma 1. ■

With this technical lemma in place, we are ready to state the main result of this paper.

Theorem 1 *Define $\rho = \frac{\pi(g^2)}{q(g^2)}$ as the variance of the unnormalized importance density with respect to importance weights g as in Lemma 2. Then under Assumptions 1-3, we have the following approximate consistency results:*

(i) for diminishing compression budget $\epsilon_n = \alpha^n$ with $\alpha \in (0,1)$, the estimator of Alg. 2 satisfies

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \frac{\alpha}{1-\alpha} + \mathcal{O}\left(\frac{1}{N}\right) + G. \quad (36)$$

To obtain a δ accurate integral estimate, we need at least $N \geq \mathcal{O}\left(\frac{1}{\delta}\right)$ particles and compression attenuation rate sufficiently large such that $0 < \alpha \leq 1/(1+(2/\delta))$.

(ii) for constant compression budget $\epsilon_n = \epsilon > 0$ and memory $\mathcal{M} := \mathcal{O}\left(\frac{1}{\epsilon^{1/(2p)}}\right)$

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \mathcal{O}\left(\frac{N}{\mathcal{M}^{1/2p}} + \frac{1}{N}\right) \quad (37)$$

and

$$\mathcal{O}\left(\frac{1}{\delta}\right) \leq N \leq \mathcal{O}\left(\delta\mathcal{M}^{1/(2p)}\right), \quad (38)$$

which implies that $\mathcal{M} \geq \mathcal{O}\left(\frac{1}{\delta^{4p}}\right)$.

Proof: Consider the statement of Lemma 24 and to proceed next, we characterize the behavior of the term $\sum_{n=1}^{N}\epsilon_n$ since it eventually determines the final bias in the integral estimation. **Theorem 1(i)**: Diminishing compression budget: Let us consider $\epsilon_n = (\alpha)^n$ with $\alpha = (0,1)$, which implies that

$$\sum_{n=1}^{N}\epsilon_n = \frac{\alpha(1-\alpha^N)}{1-\alpha} \leq \frac{\alpha}{1-\alpha}. \quad (39)$$

Substituting (39) into the right hand side of (35), we get

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \frac{\alpha}{1-\alpha} + \frac{24}{N}\rho. \quad (40)$$

To obtain a δ accurate integral estimate, we need $N \geq \frac{48\rho}{\delta}$ and $0 < \alpha \leq 1/(1+(2/\delta))$.

Theorem 1(ii): Constant compression budget: Let us consider $\epsilon_n = \epsilon$, which implies that

$$\sum_{n=1}^{N}\epsilon_n = N\epsilon. \quad (41)$$

Using (41) in (35), we can write

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq N\epsilon + \frac{24}{N}\rho. \quad (42)$$

For a constant compression budget ϵ, from Theorem 2, we have

$$M_\infty \leq \mathcal{O}\left(\frac{1}{\epsilon^{2p}}\right) := \mathcal{M}. \quad (43)$$

If we are given a maximum memory requirement $\mathcal{M}$, then we can choose ϵ as

$$\epsilon = \frac{G}{\mathcal{M}^{1/(2p)}}, \quad (44)$$

where G is a bound on the unnomalized weight $g(x(u)) \leq G$ for all u. Using this lower bound value of ϵ in (42), we get

$$\Big|\sup_{|\phi|\leq 1}\big(\mathbb{E}[\hat{I}_N(\phi) - I(\phi)]\big)\Big| \leq \frac{NG}{\mathcal{M}^{1/2p}} + \frac{24}{N}\rho. \quad (45)$$

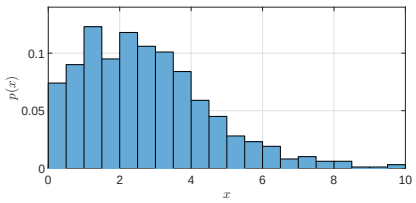
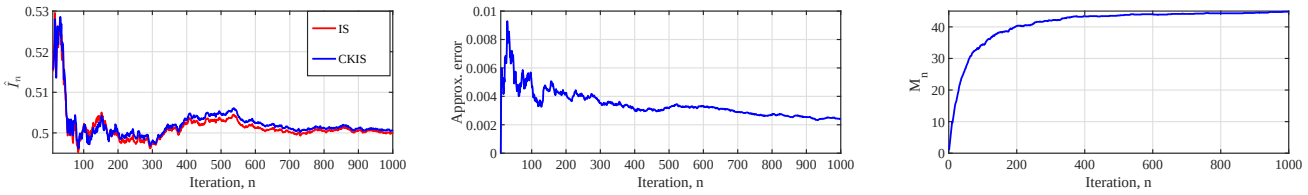


Fig. 2: Particle histogram for direct sampling experiment.



(a) Integral estimate. (b) Integral Error between Algs. 1 - 2. (c) Model Order.

Fig. 3: Simulation results for Alg. 2 run with Gaussian kernel ($h = 0.01$) and compression budget $\epsilon = 3.5$ for the problem (48). The memory-reduction scheme nearly preserves statistical consistency, while yielding reasonable complexity, whereas Alg. 1 attains exact consistency as its memory grows unbounded with index n. All the plots are averaged over 10 iterations.

Note that the first term in the above expression increases with N and the second term decreases with N, hence we obtain a tradeoff between memory and accuracy for the importance sampling based estimator. For a given memory $\mathcal{M}$ (number of elements we could store in the dictionary) and the required accuracy δ, we obtain the following bound on the number of iterations N

$$\frac{48\rho}{\delta} \leq N \leq \frac{\delta M^{1/(2p)}}{2G}, \tag{46}$$

which implies that for increased accuracy we need to run for more iterations but need more memory, and vice versa. ■

Theorem 1 establishes that the compressed kernelized importance sampling scheme proposed in Section III is *nearly* asymptotically consistent. Note that the right hand side in (36) consists of three terms depending upon α, $\frac{1}{N}$, and G. If we ignore G, which actually depends upon the approximation associated with function $\phi(\cdot)$, the other two terms can be made arbitrarily small by making α close to zero and a very high N. Hence, the integral estimation can be made arbitrarily close to exact integral. However, when these parameters are fixed positive constants, they provide a tunable tradeoff between bias and memory. That is, when the compression budget is a positive constant, then the memory of the posterior distribution representation is finite, as we formalize next.

Theorem 2 *Under Assumptions 1-2 (in Section IV-A), for compact feature space $\mathcal{X}$ and bounded importance weights $g(\mathbf{x}^{(n)})$, the model order M_n for Algorithm 2, for all n is bounded by*

$$1 \leq M_n \leq \mathcal{O}\left(\frac{1}{\epsilon^{2p}}\right). \tag{47}$$

Theorem 2 (proof in Appendix B) contrasts with the classical bottleneck in the number of particles required to represent an arbitrary posterior, which grows unbounded [23]. While this is a pessimistic estimate, experimentally we observe orders of magnitude reduction in complexity compared to exact importance sampling, which is the focus of the subsequent section. Observe, however, that this memory-reduction does not come for free, as once the compression budget is fixed, the memory is fixed by the ratio $\frac{1}{\epsilon^{2p}}$ that eventually results in a lower bound on the accuracy of the integral estimate.

V. Experiments

A. Direct Importance Sampling

In this section, we conduct a simple numerical experiment to demonstrate the efficacy of the proposed algorithm in terms of balancing model parsimony and statistical consistency. We consider the problem of estimating the expected value of function $\phi(\mathbf{x})$ with the target $q(\mathbf{x})$ and the proposal $\pi(\mathbf{x})$ given by

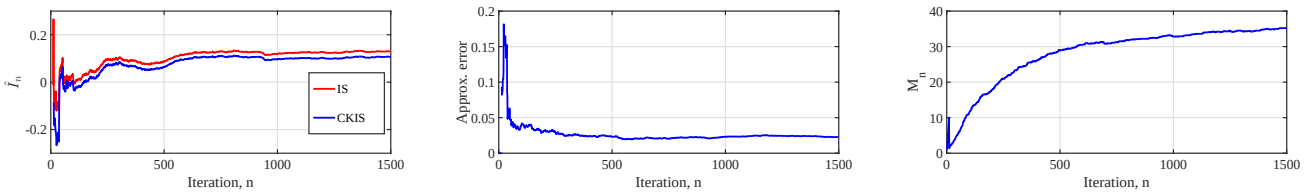
$$\phi(x) = 2\sin\left(\frac{\pi}{(1.5x)}\right), \; q(\mathbf{x}) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-1)^2}{2}\right), \; \pi(\mathbf{x}) = \frac{1}{\sqrt{4\pi}}\exp\left(-\frac{(x-1)^2}{4}\right), \tag{48}$$

to demonstrate that generic Monte Carlo integration allows one to track generic quantities of random variables that are difficult to compute under more typical probabilistic hypotheses. For (48), since $q(\mathbf{x})$ is known, this is referred to as "direct importance sampling". We run Algorithm 1, i.e., classic importance sampling, and Algorithm 2 for the aforementioned problem. For Algorithm 2, we select compression budget $\epsilon = 3$, and used a Gaussian kernel with bandwidth $h = 0.01$. We track the normalized integral estimate (9), absolute integral approximation error, and the number of particles that parameterize the empirical measure (model order).

We first represent the histogram of the particles generated in Fig. 2. In Fig. 3a, we plot the un-normalized integral approximation error for Algorithms 1 - 2, which are close, and the magnitude of the difference depends on the choice of compression budget. Very little error is incurred by kernel mean embedding and memory-reduction. The magnitude of the error relative to the number of particles generated is displayed in Fig. 3b: observe that the error settles on the order of 10^{-3}. In Fig. 3c, we display the number of particles retained by Algorithm 2, which stabilizes to around 56, whereas the complexity of the empirical measure given by Algorithm 1 grows linearly with sample index n, which noticeably grows *unbounded*.

B. Indirect Importance Sampling

As discussed in Sec. II, in practice we do not know the target distribution $q(\mathbf{x})$ and hence we use Bayes rule as described in (7) to calculate $q(\mathbf{x}^{(n)})$ at each instant t. We consider the observation model $y_t = b + \sin(2\pi x) + \eta_t$ where $\eta_t \sim \mathcal{N}(0, \sigma^2)$. From the equality in (7), we need the likelihood



(a) Integral estimate. (b) Integral Error between Algs. 1 - 2. (c) Model Order.

Fig. 4: Simulation results for Alg. 2 with indirect IS, run with Gaussian kernel ($h = 0.012$) and compression budget $\epsilon = 10^{-3}$ for the problem (48). The memory-reduction scheme nearly preserves statistical consistency, while yielding reasonable complexity, whereas Alg. 1 attains exact consistency as its memory grows unbounded with index n. All the plots are averaged over 10 iterations.

and a prior distribution to calculate $q(x^{(n)})$ using Bayes Rule [cf. (2)]. Here we fix the likelihood (measurement model) and prior as $\mathbb{P}\left(\{y_k\}_{k\leq K} \,\middle|\, x^{(n)}\right) = \frac{1}{(2\pi\sigma_1^2)^{K/2}} \exp\left(-\frac{\|\mathbf{y}-x^{(n)}\|^2}{2\sigma_1^2}\right)$, $\mathbb{P}\left(x^{(n)}\right) = \frac{1}{(2\pi\sigma_2^2)} \exp\left(-\frac{(x^{(n)})^2}{2\sigma_2^2}\right)$. We set $K = 10$, $b = 5$, $\sigma = 0.1$, $\sigma_1 = 0.4$, $\sigma_2 = 1.6$, and compression budget $\epsilon = 10^{-3}$. A uniform distribution $\mathcal{U}[3, 7]$ is used as the importance distribution. The results are reported in Fig. 4. We observe a comparable tradeoff to that which may be gleaned from Section V: in particular, we are able to obtain complexity reduction by orders of magnitude with extremely little integral estimation error. This suggests the empirical validity of our compression-rule based on un-normalized importance weights operating in tandem with kernel smoothing.

C. Source Localization

In this section we present a sensor network localization experiment based on range measurements.The results illustrate the ability to succinctly represent the unknown distribution of the source signal location, yielding a model that is both parsimonious and nearly consistent. Consider the problem of localizing a static target in two-dimensional space $\mathbb{R}^2$ with range measurements from the source collected in a wireless sensor network (WSN). Since the observation model is *non-linear*, the posterior distribution of the location of the target is intractable, and hence finding the least-squares estimator is not enough. This problem is well-studied in signal processing [59] and robotics [60]. Let $\mathbf{x} = [x, y]^T$ denote the random unknown target location. We assume six sensors with locations $\{\mathbf{h}_i\}_{i=1}^6$ at locations $[1, -8]^T$, $[8, 10]^T$, $[-15, -17]^T$, $[-8, 1]^T$, $[10, 0]^T$, and $[0, 10]^T$, respectively. The true location of the target is at $[3.5, 3.5]^T$. The measurement at each sensor i is related to the true target location $\mathbf{x}$ via the following nonlinear function of range $y_{i,j} = -20\log(\|\mathbf{x} - \mathbf{h}_i\|) + \eta_i$ for $i = 1$ to 6 and $j = 1$ to N, where N_i is the number of measurements collected by sensor i. Here, $\eta_i \sim \mathcal{N}(0, 1)$ models the range estimation error. For the experiment, we consider a Gaussian prior on the target location $\mathbf{x}$ with mean $[3.5, 3.5]^T$ and identity covariance matrix. We use the actual target location as the mean for the Gaussian prior because we are interested in demonstrating that the proposed technique successfully balances particle growth and model bias. In practice, for a general possibly misspecified prior, we can appeal to advanced adaptive algorithms – for example see [37], [27].

Fig. 5 shows the performance of the proposed algorithm compared against classical (uncompressed) normalized importance sampling. Fig. 5a shows that the final estimated value of the target location for compressed and uncompressed versions of the algorithms are close. We plot the squared error in Fig. 5b and both algorithms converge with close limiting estimates. Further, in Fig. 5c we observe that the model order for the compressed distribution settles to 21, whereas the classical algorithm requires its number of particles in its importance distribution to grow unbounded. The memory-reduction comes at the cost of very little estimation error (Fig. 5a).

VI. Conclusions

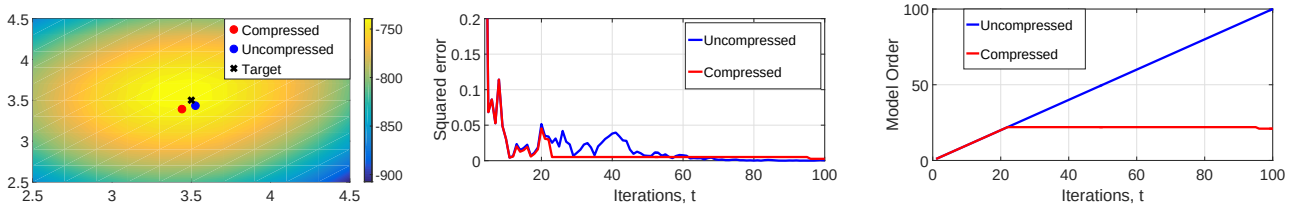
We focused on Bayesian inference where one streams simulated Monte Carlo samples to approximate an unknown posterior via importance sampling. Doing so may consistently approximate any function of the posterior at the cost of infinite memory. Thus, we proposed Algorithm 2 (CKIS) to approximate the posterior by a kernel density estimate (KDE) projected onto a nearby lower-dimensional subspace, which allows online compression as particles arrive in perpetuity. We established that the bias of CKIS depends on kernel bandwidth and compression budget, providing a tradeoff between statistical accuracy and memory. Experiments demonstrated that we attain memory-reduction by orders of magnitude with very little estimation error. This motivates future application to memory-efficient versions of Monte Carlo approaches to nonlinear signal processing problems such as localization, which has been eschewed due to its computational burden.

Appendix A
Proof of Consistency of Importance Sampling

Here we state a result on the sample complexity and asymptotic consistency of IS estimators in terms of integral error. We increase the granularity of the proof found in the literature so that the modifications required for our results on compressed IS estimates are laid bare.

Lemma 2 *[23][Theorem 2.1] Suppose π, the proposal distribution is absolutely continuous w.r.t. q, the population posterior, and both are defined over $\mathcal{X}$. Then define their Radon-Nikodyn derivative: $\frac{dq}{d\pi}(\mathbf{x}) := \frac{g(\mathbf{x})}{\int_{\mathcal{X}} g(\mathbf{x})\pi(d\mathbf{x})}$, $\rho := \frac{\pi(g^2)}{q(g^2)}$ where g is the unnormalized density of q with respect to π. Moreover, ρ is its second moment ("variance" of unnormalized density). Under Assumption 1(i), Alg. 1 contracts to the true posterior as*

$$\sup_{|\phi|\leq 1} |\mathbb{E}[I_N(\phi) - I(\phi)]| \leq \frac{12}{N}\rho, \mathbb{E}\left[(I_N(\phi) - I(\phi))^2\right] \leq \frac{4}{N}\rho\,, \tag{49}$$



(a) Location estimate. (b) Squared error estimate. (c) Model Order.

Fig. 5: Simulation results for Alg. 2 run with Gaussian kernel ($h = 0.0001$) and compression budget $\epsilon = 0.002$. Observe that the memory-reduction scheme (compressed) nearly preserves statistical consistency, while yielding a finite constant limiting model complexity, whereas the original uncompressed version (uncompressed) attains exact consistency but its memory grows linearly with particle index t.

and hence approaches exact consistency as $N \to \infty$.

Proof : This is a more detailed proof than given in [23][Theorem 2.1] develop for greater completeness and coherence. Let us denote the empirical random measure by π^N as $\pi^N := \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{x}(n)}$, and $\mathbf{x}(n) \sim \pi$, where π^N is the occupancy measure, which when weighted, yields the importance sampling empirical measure (11). Note that the integral approximation at N is denoted by $I_N(\phi)$. With the above notation is hand, it holds that

$$\pi^N(g) = \int \frac{1}{N}\sum_{n=1}^{N} g(\mathbf{x})\delta_{\mathbf{x}(n)}(\mathbf{x})dx = \frac{1}{N}\sum_{n=1}^{N} g(\mathbf{x}(n)), \quad (50)$$

and similarly

$$\begin{aligned}\pi^N(\phi g) =& \int \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{x}(n)}(\mathbf{x})\phi(\mathbf{x})g(\mathbf{x})dx \\ =& \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}(n))g(\mathbf{x}(n)). \end{aligned} \quad (51)$$

From the above equalities, we can write the estimator bias as

$$\begin{aligned} I_N(\phi) - I(\phi) =& \frac{\pi^N(\phi g)}{\pi^N(g)} - I(\phi) && (52)\\ =& \frac{\pi^N(\phi g)}{\pi^N(g)} - \left(I(\phi)\frac{\pi^N(g)}{\pi^N(g)}\right) && (53)\\ =& \frac{1}{\pi^N(g)}\left[\pi^N(\phi g) - I(\phi)\pi^N(g)\right] && (54)\\ =& \frac{1}{\pi^N(g)}\pi^N\left((\phi - I(\phi))g\right). && (55)\end{aligned}$$

Let us define $\bar{\phi} := \phi - I(\phi)$ and note that

$$\pi(\bar{\phi}g) = 0. \quad (56)$$

Rewriting the bias, we get

$$\begin{aligned} I_N(\phi) - I(\phi) =& \frac{1}{\pi^N(g)}\pi^N\left(\bar{\phi}g\right) \\ =& \frac{1}{\pi^N(g)}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right], \end{aligned} \quad (57)$$

where the second equality holds from (56). The first term in the bracket is an unbiased estimator for the second one, so that

$$\mathbb{E}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right] = 0. \quad (58)$$

Taking the expectation on both sides of (57), we get

$$\mathbb{E}\left[I_N(\phi) - I(\phi)\right] = \mathbb{E}\left[\frac{1}{\pi^N(g)}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right]\right]. \quad (59)$$

Since it equals zero, we can add the expression in (58) to the right hand side of (59) to obtain

$$\begin{aligned}\mathbb{E}\left[I_N(\phi) - I(\phi)\right] =& \mathbb{E}\left[\frac{1}{\pi^N(g)}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right]\right] \\ &+ \mathbb{E}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right] && (60)\\ =& \mathbb{E}\left[\frac{1}{\pi^N(g)}\left[\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right]\right] \\ &+ \mathbb{E}\left[\frac{1}{\pi(g)}\left(\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right)\right]. && (61)\end{aligned}$$

Taking the expectation operator outside, we get

$$\begin{aligned}\mathbb{E}\left[I_N(\phi) - I(\phi)\right] =& \mathbb{E}\left[\left(\frac{1}{\pi^N(g)} - \frac{1}{\pi(g)}\right)\left(\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right)\right] \\ =& \mathbb{E}\left[\frac{1}{\pi^N(g)\pi(g)}\left(\pi(g) - \pi^N(g)\right)\left(\pi^N\left(\bar{\phi}g\right) - \pi\left(\bar{\phi}g\right)\right)\right]. \end{aligned} \quad (62)$$

Next, we split the set of integration to $A = \{2\pi^N_{MC}(g) > \pi(g)\}$ and its compliment using the property

$$E[f(X)] = E[f(X)1_A(X)] + E[f(X)1_{A^c}(X)],$$

where 1_A the indicator function of the set A selecting $A = \{2\pi^N_{MC}(g) > \pi(g)\}$, which takes value 1 if $x \in A$ and 0 if $x \notin A$. We get

$$\begin{aligned}\left|\mathbb{E}\left[I_N(\phi) - I(\phi)\right]\right| \leq& \left|\mathbb{E}\left[I_N(\phi) - I(\phi)\right]\mathbf{1}_{\{2\pi^N(g) > \pi(g)\}}\right| \\ &+ \left|\mathbb{E}\left[I_N(\phi) - I(\phi)\right]\mathbf{1}_{\{2\pi^N(g) \leq \pi(g)\}}\right|. \end{aligned} \quad (63)$$

Consider the second term of (63), and use the fact that $|\phi| \leq 1$, and so $|\mu^N(\phi)|, |I(\phi)| \leq 1$ since they are mean values w.r.t. probability measures μ^N, q respectively. Then we use $E[1_A] = P(A)$ and obtain

$$\begin{aligned}\left|\mathbb{E}\left[I_N(\phi) - I(\phi)\right]\right| \leq& \left|\mathbb{E}\left[I_N(\phi) - I(\phi)\right]\mathbf{1}_{\{2\pi^N(g) > \pi(g)\}}\right| \\ &+ 2\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right). \end{aligned} \quad (64)$$

The constant 2 comes from the fact that $|I_N(\phi) - I(\phi)| \leq |I_N(\phi)| + |I(\phi)| \leq 2$. For the first term on the right hand side of (64), from the set condition (, it holds that

$$\frac{1}{\pi^N_{MC}(g)\pi(g)} < \frac{2}{\pi^2(g)}, \quad (65)$$

which implies that

$$|\mathbb{E}\left[I_N(\phi)-I(\phi)\right]| \leq \frac{2}{\pi(g)^2}\mathbb{E}\left[|\pi(g)-\pi^N(g)||\pi^N(\bar{\phi}g)-\pi\big(\bar{\phi}g\big)|\right] + 2\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right). \quad (66)$$

Finally, to upper bound the first term on the right hand side of (66), we first bound the expectation using Cauchy-Schwartz

$$E[|\pi(g)-\pi^N(g)||\pi^N(\bar{\phi}g) - \pi(\bar{\phi}g)|] \leq E[(\pi(g) - \pi^N(g))^2]^{\frac{1}{2}}E[(\pi^N(\bar{\phi}g) - \pi(g))^2]^{\frac{1}{2}} \quad (67)$$

The first expectation on the right hand side of (67) is bounded as follows: by definition of π^N we have for $x_n \sim \pi$ independent that

$$E[(\pi(g) - \pi^N(g))^2] = E[(\pi(g) - \frac{1}{N}\sum_{n=1}^{N} g(\mathbf{x}(n))^2)] = \frac{1}{N^2}E[(\sum_{n=1}^{N} g(\mathbf{x}(n)) - N\pi(g))^2], \quad (68)$$

which since $E[g(\mathbf{x}(n))] = \pi(g)$ and by independence of the $\mathbf{x}(n)$ is equal to

$$= \frac{1}{N^2}Var(\sum_{n=1}^{N} g(\mathbf{x}(n))) = \frac{1}{N^2}\sum_{n=1}^{N} Var(g(\mathbf{x}(n))),$$

and since $\mathbf{x}(n)$ is identically distributed, $\mathbf{x}(n) \sim \pi$, this is equal to

$$\frac{N}{N^2}Var_{u\sim\pi}(g) = \frac{1}{N}(\pi(g^2) - \pi(g)^2) \leq \frac{1}{N}\pi(g^2).$$

The second expectation on the right hand side of (67) is bounded in a similar way along with the fact that $|\phi| \leq 1$ so that $|\bar{\phi}| \leq 2$. Then we utilize these upper bounds on the right hand side of (63) to obtain

$$|\mathbb{E}\left[I_N(\phi)-I(\phi)\right]| \leq \frac{2}{\pi(g)^2}\mathbb{E}\left[|\pi(g)-\pi^N(g)||\pi^N(\bar{\phi}g)-\pi\left(\bar{\phi}g\right)|\right] + 2\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right) \quad (69)$$

$$\leq \frac{2}{\pi(g)^2}\frac{1}{\sqrt{N}}\pi(g^2)^{1/2}\frac{2}{\sqrt{N}\pi(g^2)^{1/2}}] + 2\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right) \quad (70)$$

where the inequalities follow from the fact that the test function is bounded $|\phi|$. Next, note that

$$\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right) = \mathbb{P}\left(2\left(\pi^N(g) - \pi(g)\right) \leq -\pi(g)\right) \leq \mathbb{P}\left(2|\pi^N(g) - \pi(g)| \geq \pi(g)\right), \quad (71)$$

where the first equality is obtained by subtracting $-2\pi(g)$ from both sides inside the bracket. Next, we use the Markov inequality, given by $\mathbb{P}\left(X \geq a\right) \leq \frac{\mathbb{E}(X)}{a}$. Utilizing this, we can write

$$\mathbb{P}\left(2|\pi^N(g)-\pi(g)| \geq \pi(g)\right) \leq \frac{2\mathbb{E}\left[|\pi^N(g)-\pi(g)|\right]}{\pi(g)} \leq \frac{4}{N}\frac{\pi(g^2)}{\pi(g)^2}.$$

This implies that

$$\mathbb{P}\left(2\pi^N(g) \leq \pi(g)\right) \leq \frac{4}{N}\frac{\pi(g^2)}{\pi(g)^2}. \quad (72)$$

Finally, using the upper bound in (72) in (70), we obtain

$$\sup_{|\phi|\leq 1} |\mathbb{E}\left[I_N(\phi) - I(\phi)\right]| \leq \frac{12}{N}\frac{\pi(g^2)}{\pi(g)^2} \quad (73)$$

which proves the result. ■

APPENDIX B
PROOF OF THEOREM 2

We begin by presenting a lemma which allows us to relate the stopping criterion of our sparsification procedure to a Hilbert subspace distance.

Lemma 3 *Define the distance of an arbitrary feature vector* $\mathbf{x}$ *evaluated by the feature transformation* $\psi(\mathbf{x}) := \kappa(\mathbf{x}, \cdot)$ *to* $\mathcal{H}_{\mathbf{D}} = span\{\psi(\mathbf{d}_n)\}_{n=1}^{M}$*, the subspace of the real space spanned by a dictionary* $\mathbf{D}$ *of size* M*, as*

$$dist(\psi(\mathbf{x}), \mathcal{H}_{\mathbf{D}}) = \min_{y\in\mathcal{H}_{\mathbf{D}}} |\psi(\mathbf{x}) - \mathbf{v}^T\boldsymbol{\phi}_{\mathbf{D}}|. \quad (74)$$

This set distance simplifies to the following least-squares projection when $\mathbf{D} \in \mathbb{R}^{p\times M}$ *is fixed*

$$dist(\psi(\mathbf{x}), \mathcal{H}_{\mathbf{D}}) = \left|\psi(\mathbf{x}) - \psi(\mathbf{x})\boldsymbol{\psi}_{\mathbf{D}}^T\mathbf{K}_{\mathbf{D},\mathbf{D}}^{-1}\boldsymbol{\psi}_{\mathbf{D}}\right|. \quad (75)$$

Proof: The distance to the subspace $\mathcal{H}_{\mathbf{D}}$ is defined as

$$\text{dist}(\psi(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) = \min_{y\in\mathcal{H}_{\mathbf{D}}} |\psi(\mathbf{x}) - \mathbf{v}^T\boldsymbol{\psi}_{\mathbf{D}}| = \min_{\mathbf{v}\in\mathbb{R}^M} |\psi(\mathbf{x}) - \mathbf{v}^T\boldsymbol{\psi}_{\mathbf{D}}|, \quad (76)$$

where the first equality comes from the fact that the dictionary $\mathbf{D}$ is fixed, so $\mathbf{v} \in \mathbb{R}^M$ is the only free parameter. Now plug in the minimizing weight vector $\tilde{\mathbf{v}}^\star = \psi(\mathbf{x}_n)\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}^{-1}\boldsymbol{\psi}_{\mathbf{D}_n}$ into (76) which is obtained in an analogous manner to the logic which yields (15) - (21). Doing so simplifies (76) to the following

$$\text{dist}(\psi(\mathbf{x}_n), \mathcal{H}_{\mathbf{D}_n}) = \left|\psi(\mathbf{x}_n) - \psi(\mathbf{x}_n)[\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}^{-1}\boldsymbol{\psi}_{\mathbf{D}_n}]^T\boldsymbol{\psi}_{\mathbf{D}_n}\right| = \left|\psi(\mathbf{x}_n) - \psi(\mathbf{x}_n)\boldsymbol{\psi}_{\mathbf{D}_n}^T\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}^{-1}\boldsymbol{\psi}_{\mathbf{D}_n}\right|. \quad (77)$$

■

Next, we establish that the model order is finite.

Proof : Consider the model order of the kernel mean embedding β_n and β_{n-1} generated by Algorithm 2 and denoted by M_n and M_{n-1}, respectively, at two arbitrary subsequent instances n and $n-1$. Suppose the model order of the estimate β_n is less than or equal to that of β_{n-1}, i.e. $M_n \leq M_{n-1}$. This relation holds when the stopping criterion of MMD-OMP (defined in Algorithm 2), stated as $\min_{j=1,\ldots,M_{n-1}+1}\gamma_j > \epsilon$, *is not* satisfied for the updated dictionary matrix with the newest sample point $\mathbf{x}(n)$ appended: $\tilde{\mathbf{D}}_n = [\mathbf{D}_{n-1}; \mathbf{x}(n)]$ [cf. (19)], which is of size $M_{n-1}+1$. Thus, the negation of the termination condition of MMD-OMP in Algorithm 2 must hold for this case, stated as

$$\min_{j=1,\ldots,M_{n-1}+1}\gamma_j \leq \epsilon. \quad (78)$$

Observe that the left-hand side of (78) lower bounds the approximation error $\gamma_{M_{n-1}+1}$ for removing the most recent sample $\mathbf{x}(n)$ due to the minimization over j, that

is, $\min_{j=1,\ldots,M_{n-1}+1}\gamma_j \leq \gamma_{M_{n-1}+1}$. Consequently, if $\gamma_{M_{n-1}+1} \leq \epsilon$, then (78) holds and the model order does not grow. Thus it suffices to consider $\gamma_{M_{n-1}+1}$. The definition of $\gamma_{M_{n-1}+1}$ with the substitution of β_n in (78) allows us to write

$$\begin{aligned}\gamma_{M_{n-1}+1} &= \min_{\mathbf{u}\in\mathbb{R}^{M_{n-1}}}\Big|\beta_{n-1} + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \sum_{k\in\mathcal{I}\setminus\{M_{n-1}+1\}} u_k\kappa_{\mathbf{d}_k}(\mathbf{x})\Big| \\ &= \min_{\mathbf{u}\in\mathbb{R}^{M_{n-1}}}\Big|\sum_{k\in\mathcal{I}\setminus\{M_{n-1}+1\}} g(\mathbf{x}(k))\kappa_{\mathbf{d}_k}(\mathbf{x}) \\ &\qquad + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \sum_{k\in\mathcal{I}\setminus\{M_{n-1}+1\}} u_k\kappa_{\mathbf{d}_k}(\mathbf{x})\Big|, \end{aligned} \tag{79}$$

where we denote $\kappa_{\mathbf{x}(n)}(\mathbf{x}) = \kappa_{(\mathbf{x}(n),\cdot)}$ and the k^{th} column of $\mathbf{D}_n$ as $\mathbf{d}_k$. The minimal error is achieved by considering the square of the expression inside the minimization and expanding terms to obtain

$$\begin{aligned}&\Big|\sum_{k\in\mathcal{I}\setminus\{M_{n-1}+1\}} g(\mathbf{x}(k))\kappa_{\mathbf{d}_k}(\mathbf{x}) + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \sum_{k\in\mathcal{I}\setminus\{M_{n-1}+1\}} u_k\kappa_{\mathbf{d}_k}(\mathbf{x})\Big|^2 \\ &= \Big|\mathbf{g}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}) + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \mathbf{u}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big|^2 \\ &= \mathbf{g}^T\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}\mathbf{g} + g(\mathbf{x}(n))^2 + \mathbf{u}^T\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}\mathbf{u} \\ &\quad + 2g(\mathbf{x}(n))\mathbf{w}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n)) - 2g(\mathbf{x}(n))\mathbf{u}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n)) \\ &\quad - 2\mathbf{w}^T\mathbf{K}_{\mathbf{D}_n,\mathbf{D}_n}\mathbf{u}. \end{aligned} \tag{80}$$

To obtain the minimum, we compute the stationary solution of (80) with respect to $\mathbf{u} \in \mathbb{R}^{M_{n-1}}$ and solve for the minimizing $\tilde{\mathbf{u}}^\star$, which in a manner similar to the logic in (15) - (21), is given as $\tilde{\mathbf{u}}^\star = [g(\mathbf{x}(n))\mathbf{K}^{-1}_{\mathbf{D}_n,\mathbf{D}_n}\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n)) + \mathbf{g}]$. Plug $\tilde{\mathbf{u}}^\star$ into the expression in (79) and, using the short-hand notation $\sum_k u_k\kappa_{\mathbf{d}_k}(\mathbf{x}) = \mathbf{u}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})$. Simplifies (79) to

$$\begin{aligned}&\Big|\mathbf{g}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}) + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \mathbf{u}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big| \\ &= \Big|\mathbf{g}^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}) + g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) \\ &\quad - [g(\mathbf{x}(n))\mathbf{K}^{-1}_{\mathbf{D}_n,\mathbf{D}_n}\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n)) + \mathbf{g}]^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big| \\ &= \Big|g(\mathbf{x}(n))\kappa_{\mathbf{x}(n)}(\mathbf{x}) - [g(\mathbf{x}(n))\mathbf{K}^{-1}_{\mathbf{D}_n,\mathbf{D}_n}\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n))]^T\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big| \\ &= g(\mathbf{x}(n))\Big|\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}^n)^T\mathbf{K}^{-1}_{\mathbf{D}_n,\mathbf{D}_n}\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big| \end{aligned} \tag{81}$$

Notice that the right-hand side of (81) may be identified as the distance to the subspace $\mathcal{H}_{\mathbf{D}_n}$ in (77) defined in Lemma 3 scaled by a factor of $g(\mathbf{x}(n))$. We may upper-bound the right-hand side of (81) as

$$\begin{aligned}&g(\mathbf{x}(n))\Big|\kappa_{\mathbf{x}(n)}(\mathbf{x}) - \boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x}(n))^T\mathbf{K}^{-1}_{\mathbf{D}_n,\mathbf{D}_n}\boldsymbol{\kappa}_{\mathbf{D}_n}(\mathbf{x})\Big| \\ &\qquad = g(\mathbf{x}(n))\text{dist}(\kappa_{\mathbf{x}(n)}(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) \end{aligned} \tag{82}$$

where we have applied (75) regarding the definition of the subspace distance on the right-hand side of (82) to replace the absolute value term. Now, when the MMD-OMP stopping criterion is violated, i.e., (78) holds, this implies $\gamma_{M_{n-1}+1} \leq \epsilon$. Therefore, the right-hand side of (82) is upper-bounded by ϵ, and we can write

$$g(\mathbf{x}(n))\text{dist}(\kappa_{\mathbf{x}(n)}(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) \leq \epsilon. \tag{83}$$

After rearranging the terms in (83), we can write

$$\text{dist}(\kappa_{\mathbf{x}(n)}(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) \leq \frac{\epsilon}{g(\mathbf{x}(n))}, \tag{84}$$

where we have divided both sides by $g(\mathbf{x}(n))$. Observe that if (84) holds, then $\gamma_{M_n} \leq \epsilon$ holds, but since $\gamma_{M_n} \geq \min_j \gamma_j$, we may conclude that (78) is satisfied. Consequently the model order at the subsequent step does not grow which means that $M_n \leq M_{n-1}$ whenever (84) is valid.

Now, let's take the contrapositive of the preceding expressions to observe that growth in the model order ($M_n = M_{n-1} + 1$) implies that the condition

$$\text{dist}(\kappa_{\mathbf{x}(n)}(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) > \frac{\epsilon}{g(\mathbf{x}(n))} \tag{85}$$

holds. Therefore, each time a new point is added to the model, the corresponding map $\kappa_{\mathbf{x}}(\mathbf{x}(n))$ is guaranteed to be at least a distance of $\frac{\epsilon}{g(\mathbf{x}(n))}$ from every other feature map in the current model. In canonical works such as [25], [61], the largest self-normalized importance weight is shown to be bounded by a constant. Under the additional hypothesis that the *un-normalized* importance weight is bounded by some constant W, then we have via (85) $\text{dist}(\kappa_{\mathbf{x}(n)}(\mathbf{x}), \mathcal{H}_{\mathbf{D}_n}) > \frac{\epsilon}{W}$. Therefore, For a fixed compression budget ϵ, the MMD-OMP stopping criterion is violated for the newest point whenever distinct dictionary points $\mathbf{d}_k$ and $\mathbf{d}_j$ for $j, k \in \{1, \ldots, M_{n-1}\}$, satisfy the condition $\text{dist}(\kappa_{\mathbf{x}}(\mathbf{d}_j), \kappa_{\mathbf{d}_k}(\mathbf{x})) > \frac{\epsilon}{W}$. Next, we follow a similar argument as provided in the proof of Theorem 3.1 in [62]. Since $\mathcal{X}$ is compact and $\kappa_{\mathbf{x}}$ is continuous, the range $\kappa_{\mathbf{x}}(\mathcal{X})$ of the feature space $\mathcal{X}$ is compact. Therefore, the minimum number of balls (covering number) of radius κ (here, $\kappa = \frac{\epsilon}{W}$) needed to cover the set $\kappa_{\mathbf{x}}(\mathcal{X})$ is finite (see, e.g., [63]) for a finite compression budget ϵ. The finiteness of the covering number implies that the number of elements in the dictionary M_N will be finite and using [62, Proposition 2.2], we can characterize the number of elements in the dictionary as $1 \leq M_N \leq C\left(\frac{W}{\epsilon}\right)^{2p}$, where C is a constant depending upon the space $\mathcal{X}$. ■

REFERENCES

[1] A. S. Bedi, A. Koppel, B. M. Sadler, and V. Elvira, "Compressed streaming importance sampling for efficient representations of localization distributions," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 477–481.

[2] A. Koppel, A. S. Bedi, B. M. Sadler, and V. Elvira, "A projection operator to balance consistency and complexity in importance sampling," in *Neural Information Processing Systems, Symposium on Advances in Approximate Bayesian Inference*, 2019.

[3] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[4] C. Robert, *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[5] J. V. Candy, *Bayesian signal processing: classical, modern, and particle filtering methods*. John Wiley & Sons, 2016, vol. 54.

[6] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[7] M. A. Beaumont and B. Rannala, "The bayesian revolution in genetics," *Nature Reviews Genetics*, vol. 5, no. 4, p. 251, 2004.

[8] E. Karseras, W. Dai, L. Dai, and Z. Wang, "Fast variational bayesian learning for channel estimation with prior statistical information," in *IEEE SPAWC*, June 2015, pp. 470–474.

[9] O. Jangmin, J. W. Lee, S.-B. Park, and B.-T. Zhang, "Stock trading by modelling price trend with dynamic bayesian networks," in *IDEAL*. Springer, 2004, pp. 794–799.

[10] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *ICRA*, vol. 2, 2010.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[12] M. Hoshiya and E. Saito, "Structural identification by extended kalman filter," *Journal of engineering mechanics*, vol. 110, no. 12, pp. 1757–1770, 1984.

[13] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[14] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993.

[15] L. Martino and V. Elvira, "Metropolis sampling," *Wiley StatsRef: Statistics Reference Online*, pp. 1–18, 2014.

[16] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, "Generalized multiple importance sampling," *Statistical Science*, vol. 34, no. 1, pp. 129–155, 02 2019.

[17] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[18] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[19] T. P. Minka, "Expectation propagation for approximate bayesian inference," *arXiv preprint arXiv:1301.2294*, 2013.

[20] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[21] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," in *Advances in neural information processing systems*, 2016, pp. 2378–2386.

[22] J. Zhang, R. Zhang, and C. Chen, "Stochastic particle-optimization sampling and the non-asymptotic convergence theory," *arXiv preprint arXiv:1809.01293*, 2018.

[23] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, A. Stuart *et al.*, "Importance sampling: Intrinsic dimension and computational cost," *Statistical Science*, vol. 32, no. 3, pp. 405–431, 2017.

[24] V. Elvira and L. Martino, "Advances in importance sampling," *Wiley StatsRef: Statistics Reference Online, arXiv:2102.05407*, 2021.

[25] T. Bengtsson, P. Bickel, B. Li *et al.*, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and statistics*. Ints. of Math. Stats., 2008, pp. 316–334.

[26] C. P. Robert, V. Elvira, N. Tawn, and C. Wu, "Accelerating mcmc algorithms," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 10, no. 5, p. e1435, 2018.

[27] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: past, present, and future," *IEEE Signal Proc. Mag.*, vol. 34, no. 4, pp. 60–79, 2017.

[28] A. B. Owen, "Monte carlo theory, methods and examples," 2013.

[29] V. Elvira, L. Martino, and C. P. Robert, "Rethinking the effective sample size," *arXiv preprint arXiv:1809.04129*, 2018.

[30] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *JMLR*, vol. 11, no. Apr, pp. 1517–1561, 2010.

[31] L. Martino and V. Elvira, "Compressed monte carlo with application in particle filtering," *Information Sciences*, vol. 553, pp. 331–352, 2021.

[32] L. Martino, V. Elvira, J. López-Santiago, and G. Camps-Valls, "Compressed particle methods for expensive models with application in astronomy and remote sensing," *IEEE Transactions on Aerospace and Electronic Systems*, 2021.

[33] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *Asilomar Conference*, 1993.

[34] A. Koppel, "Consistent online gaussian process regression without the sample complexity bottleneck," in *ACC*. IEEE, 2019.

[35] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *Journal of Machine Learning Research*, vol. 20, no. 3, pp. 1–44, 2019.

[36] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado, "Fight sample degeneracy & impoverishment in particle filters," *Expert Systems w/ apps.*, vol. 41, no. 8, pp. 3944–3954, 2014.

[37] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, "Improving population monte carlo: Alternative weighting and resampling schemes," *Signal Processing*, vol. 131, pp. 77–91, 2017.

[38] T. Campbell and T. Broderick, "Automated scalable bayesian inference via hilbert coresets," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 551–588, 2019.

[39] Y. Chen, M. Welling, and A. Smola, "Super-samples from kernel herding," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 109–116.

[40] S. Lacoste-Julien, F. Lindsten, and F. Bach, "Sequential kernel herding: Frank-wolfe optimization for particle filtering," in *18th International Conference on Artificial Intelligence and Statistics*, 2015.

[41] N. Keriven, A. Bourrier, R. Gribonval, and P. Pérez, "Sketching for large-scale learning of mixture models," *Information and Inference: A Journal of the IMA*, vol. 7, no. 3, pp. 447–508, 2018.

[42] F. Futami, Z. Cui, I. Sato, and M. Sugiyama, "Bayesian posterior approximation via greedy particle optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3606–3613.

[43] T. Campbell and T. Broderick, "Bayesian coreset construction via greedy iterative geodesic ascent," in *International Conference on Machine Learning*, 2018, pp. 698–706.

[44] S.-H. Jun and A. Bouchard-Côté, "Memory (and time) efficient sequential monte carlo," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 514–522.

[45] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, 2016.

[46] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge, 2013, vol. 3.

[47] B. Li, T. Bengtsson, and P. Bickel, "Curse-of-dimensionality revisited: Collapse of importance sampling in very large scale systems," *Rapport technique*, vol. 85, 205.

[48] S. T. Tokdar and R. E. Kass, "Importance sampling: a review," *Wiley Interdisciplinary Reviews: Comp. Stat.*, vol. 2, no. 1, pp. 54–60, 2010.

[49] R. Wheeden, R. Wheeden, and A. Zygmund, *Measure and Integral: An Introduction to Real Analysis*, ser. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 1977.

[50] S. Ghosal, J. K. Ghosh, A. W. Van Der Vaart *et al.*, "Convergence rates of posterior distributions," *Annals of Statistics*, vol. 28, no. 2, pp. 500–531, 2000.

[51] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, 2017.

[52] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[53] R. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Opt.*, vol. 14, no. 5, pp. 877–898, 1976. [Online]. Available: https://doi.org/10.1137/0314056

[54] D. Needell, J. Tropp, and R. Vershynin, "Greedy signal recovery review," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*. IEEE, 2008, pp. 1048–1050.

[55] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, Jul 2002. [Online]. Available: https://doi.org/10.1023/A:1013955821559

[56] A. B. Owen, *Monte Carlo theory, methods and examples*, 2013.

[57] L. Wasserman, *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[58] R. Fortet and E. Mourier, "Convergence de la répartition empirique vers la répartition théorique," in *Annales scientifiques de l'École Normale Supérieure*, vol. 70, no. 3, 1953, pp. 267–285.

[59] A. M. Ali, S. Asgari, T. C. Collier, M. Allen, L. Girod, R. E. Hudson, K. Yao, C. E. Taylor, and D. T. Blumstein, "An empirical study of collaborative acoustic source localization," *Journal of Signal Processing Systems*, vol. 57, no. 3, pp. 415–436, 2009.

[60] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, 2005.

[61] P. Bickel, B. Li, T. Bengtsson *et al.*, "Sharp failure rates for the bootstrap particle filter in high dimensions," in *Pushing the limits of contemporary statistics*. Ints. of Math. Stats., 2008, pp. 318–329.

[62] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transa. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, 2004.

[63] M. Anthony and P. L. Bartlett, *Neural network learning: Theoretical foundations*. Cambridge, 2009.