Sampling from mixtures with negative weights: application to density approximation by Gaussian processes

Luca Martino*

* Università di Catania, Italy.

email: luca.martino@unict.it

Abstract

In this work, we focus on mixtures with negative coefficients and their applications in computational statistics. Mixtures of probability densities are widely used in statistics and machine learning. While classical mixtures restrict weights to be non-negative, allowing negative weights enables more flexible density approximation. However, negative weights introduce challenges in handling and sampling such distributions. For this purpose, we propose efficient Monte Carlo (MC) methods (including MC quadratures, rejection sampling and importance sampling schemes) for computing integrals and generating samples from these mixtures. A tailored proposal density ensures accurate and efficient generation of (unweighted) samples. Furthermore, we introduce an IS scheme which employs a mixture with negative coefficients as a proposal density, yielding samples with both positive and negative importance weights. Applications in Gaussian process-based density estimation demonstrate the practical relevance and efficiency of proposed schemes. An adaptive importance sampling procedure based on GP-regression is also proposed. The numerical results provide clear empirical evidence of the accuracy and computational efficiency of the proposed methods.

Keywords: Non-convex mixtures, mixtures with negative weights, Gaussian processes, rejection sampling, adaptive importance sampling

1 Introduction

Mixtures of probability densities are fundamental tools in statistics, signal processing, and machine learning [3]. A mixture model represents a probability distribution as a convex combination of simpler component distributions, such as Gaussians, exponentials, or Gamma probability density function (pdf), to name a few. Mixture models provide a powerful and flexible framework for modeling complex data [15, 34, 35].

While classical mixture models restrict weights to be non-negative, allowing negative weights opens new theoretical and practical possibilities. When weights can be negative, the resulting function may no longer be a proper probability density. In this work, we focus on the case where the

mixture remains positive and proper. Mixtures with negative components are also referred to as non-convex or pseudo-convex mixtures [2, 17, 16]. In statistics and machine learning, mixtures with negative weights can be particularly useful for density approximation [24, 27, 49]. For example, Gaussian process (GP) regressors, often used for density estimation, can lead to expansions with both positive and negative coefficients [27, 36, 37, 38]. In this context, negative weights can enable better approximation of sharp features, heavy tails, and periodic behaviors/patterns that may be difficult to capture with strictly non-negative mixtures. However, negative weights also introduce significant challenges. The resulting function may not always be a proper density, and classical sampling methods cannot be directly applied [20, 32, 41].

In this work, we describe several Monte Carlo quadrature and sampling methods involving mixtures with (possibly) negative coefficients (Mix-NCs). First, in Section 3, we focus on the efficient computation of integrals involving non-convex mixtures. Second, in Section 4, we propose an efficient proposal density to be used within rejection sampling (RS) and/or importance sampling with resampling (IS+R) schemes. In both cases, we obtain (unweighted) samples (exactly in RS, or asymptotically in IS+R) that are distributed according to the target mixture with negative weights. The proposal density introduced here ensures good performance in both RS and importance sampling (IS) schemes, as it is itself a "piece" of the target density. In Section 5 we describe how to use a mixture with negative coefficients as a proposal density within an IS scheme. In this setting, some of the generated samples carry negative importance weights. As a result, the proposed procedure admits a physical analogy in which samples with positive importance weights correspond to "matter", while samples with negative importance weights correspond to "anti-matter" [42, 43]. We also describe in Section 6 the application of these methods to GP-based density approximation and apply this idea for designing an adaptive importance sampling (AIS) based on a regression procedure, that can be useful in several frameworks [24, 27, 49]. Theoretical discussions are also provided. The numerical simulations given in Section 8 demonstrate the efficiency and accuracy of the proposed techniques. Related code is also provided.[1]

2 Framework and main notation

Let consider a finite mixture of densities with potentially negative associated weights, i.e.,

$$\bar{p}(\mathbf{x}) \propto p(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x}), \tag{1}$$

$$= p_+(\mathbf{x}) + p_-(\mathbf{x}), \tag{2}$$

$$= \sum_{i=1}^{M} \alpha_i^+ \phi_i(\mathbf{x}) + \sum_{k=1}^{N-M} \alpha_k^- \phi_k(\mathbf{x}) \geq 0, \qquad \forall \mathbf{x} \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^{d_X}, \tag{3}$$

[1] Related Matlab code is given at `http://www.lucamartino.altervista.org/public_code_NegMix2025.zip`.

where $\alpha_i^+ > 0$ and $\alpha_i^- < 0$. Moreover, each $\phi_n(\mathbf{x})$ represents a component density. We have also set

$$p_+(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^+ \phi_i(\mathbf{x}) \geq 0, \quad \text{and} \tag{4}$$

$$p_-(\mathbf{x}) = \sum_{k=1}^{N-M} \alpha_k^- \phi_k(\mathbf{x}) \leq 0. \tag{5}$$

We have assumed that:

$$\alpha_1 = \alpha_1^+ > 0, ..., \alpha_M = \alpha_M^+ > 0, \quad \alpha_{M+1} = \alpha_1^- < 0, ..., \alpha_N = \alpha_{N-M}^- < 0.$$

Namely, without loss of generality, we are assuming that the components are ordered: the first M components are associated to positive weights, α_i^+, and the rest of $N - M$ components have assigned to the negative weights, α_i^-.

Furthermore, additional assumptions are:

(a) $\phi_n(\mathbf{x}) \geq 0$ and $\int_{\mathcal{X}} \phi_n(\mathbf{x}) d\mathbf{x} = 1$, for all n, for instance,

$$\phi_n(\mathbf{x}) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{||\mathbf{x} - \boldsymbol{\mu}_n||^2}{2\lambda^2}\right), \tag{6}$$

where $\boldsymbol{\mu}_n$, λ^2 represent the mean vector and variance value. Clearly, this only a possible example.

(b) We can evaluate and we can draw samples from each component $\phi_n(\mathbf{x})$.

Given the assumptions above, and since we consider a proper/normalized mixture density, i.e., $\bar{p}(\mathbf{x}) \geq 0$ and $\int_{\mathcal{X}} \bar{p}(\mathbf{x}) d\mathbf{x} = 1$, we can write

$$\bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \frac{\sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \sum_{n=1}^{N} \bar{\alpha}_n \phi_n(\mathbf{x}), \tag{7}$$

where we have defined

$$\bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N} \alpha_j}, \qquad n = 1, ..., N. \tag{8}$$

Note that:

- $\sum_{n=1}^{N} \bar{\alpha}_n = 1$ (since $\int_{\mathcal{X}} \bar{p}(\mathbf{x}) d\mathbf{x} = 1$ and $\int_{\mathcal{X}} \phi_n(\mathbf{x}) d\mathbf{x} = 1$ for all n),
- even if we have $\bar{\alpha}_n > 0$ for $n = 1, ..., M$,
- and $\bar{\alpha}_n < 0$ for $n = M + 1, ..., N$.

Hence, Eq. (7) is *not* a convex combination of $\phi_n(\mathbf{x})$. Thus, the inequality

$$\sum_{i=1}^{M} \alpha_i^+ > \sum_{k=1}^{N-M} \alpha_k^-,$$

also holds, since we need $\sum_{n=1}^{N} \alpha_n > 0$ to have $p(\mathbf{x}) \geq 0$. We can also define the two *partial-mixtures*,

$$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+} = \sum_{m=1}^{M} \bar{\alpha}_m^+ \phi_m(\mathbf{x}), \quad \bar{p}_-(\mathbf{x}) = \frac{p_-(\mathbf{x})}{\sum_{i=1}^{N-M} \alpha_i^-} = \sum_{k=1}^{M} \bar{\alpha}_k^- \phi_k(\mathbf{x}), \tag{9}$$

where

$$\bar{\alpha}_m^+ = \frac{\alpha_m^+}{\sum_{i=1}^{M} \alpha_i^+} > 0, \qquad \bar{\alpha}_k^- = \frac{\alpha_k^-}{\sum_{j=1}^{N-M} \alpha_j^-} > 0, \tag{10}$$

As a summary, note that:

- $\bar{p}_+(\mathbf{x}) \geq 0$ and $\bar{p}_-(\mathbf{x}) \geq 0$, despite $\alpha_j^- < 0$ and $p_-(\mathbf{x}) \leq 0$,
- $\bar{p}_+(\mathbf{x})$ and $\bar{p}_-(\mathbf{x})$ are both proper classical mixtures with non-negative weights, i.e., $\sum_{m=1}^{M} \bar{\alpha}_m^+ = 1$, with $\bar{\alpha}_m^+ > 0$, and $\sum_{k=1}^{N-M} \bar{\alpha}_k^- = 1$ with $\bar{\alpha}_k^- > 0$, despite $\alpha_j^- < 0$. This is due to the fact that $\sum_{j=1}^{N-M} \alpha_j^- < 0$ (since $\alpha_j^- < 0$), but the ratio of two negative values is positive, i.e., $\bar{\alpha}_k^- = \dfrac{\alpha_k^-}{\sum_{j=1}^{N-M} \alpha_j^-} > 0$.

Finally, we can also write the complete mixture $\bar{p}(\mathbf{x})$, in Eq. (7), as a **function (not a mixture)** of the partial mixtures $\bar{p}_+(\mathbf{x})$ and $\bar{p}_-(\mathbf{x})$ in Eq. (9), i.e.,

$$\begin{aligned} \bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} &= \frac{p_+(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} + \frac{p_-(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j}, \\ &= \frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{i=1}^{M} \alpha_i^+} \cdot \frac{p_+(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} + \frac{\sum_{i=1}^{N-M} \alpha_i^-}{\sum_{i=1}^{N-M} \alpha_i^-} \cdot \frac{p_-(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j}, \end{aligned} \tag{11}$$

where we have multiplied each factor for 1, i.e., $\frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{i=1}^{M} \alpha_i^+} = 1$ and $\frac{\sum_{i=1}^{N-M} \alpha_i^-}{\sum_{i=1}^{N-M} \alpha_i^-} = 1$. Recalling that $\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+}$ and $\bar{p}_-(\mathbf{x}) = \frac{p_-(\mathbf{x})}{\sum_{i=1}^{N-M} \alpha_i^-}$ as given in Eq. (9), we can rewrite the expression above as:

$$\begin{aligned} \bar{p}(\mathbf{x}) &= \frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{j=1}^{N} \alpha_j} \bar{p}_+(\mathbf{x}) + \frac{\sum_{i=1}^{N-M} \alpha_i^-}{\sum_{j=1}^{N} \alpha_j} \bar{p}_-(\mathbf{x}) \\ &= \underbrace{\beta^+}_{>1} \bar{p}_+(\mathbf{x}) + \underbrace{(1-\beta^+)}_{<0} \bar{p}_-(\mathbf{x}), \end{aligned} \tag{12}$$

where $\bar{p}_+(\mathbf{x}) \geq 0$, $\bar{p}_-(\mathbf{x}) \geq 0$ are two classical mixtures but

$$\beta^+ = \frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{j=1}^{N} \alpha_j} > 1, \quad 1 - \beta^+ < 0, \tag{13}$$

so that is not a convex combination, as remarked below.

Remark 1. *Even if the sum of two weights β^+ and $1 - \beta^+$ is one, the linear combination $\beta^+\bar{p}_+(\mathbf{x}) + (1 - \beta^+)\bar{p}_-(\mathbf{x})$ is* **not** *a convex combination.*

Remark 2. *Thus, note that:*

- *$\bar{p}(\mathbf{x}) \geq 0$, $\bar{p}_+(\mathbf{x}) \geq 0$ and $\bar{p}_-(\mathbf{x}) \geq 0$ are normalized proper densities (hence, also non-negative functions). More precisely, they are classical mixtures of pdfs with non-negative coefficients.*
- *$p(\mathbf{x}) \geq 0$ and $p_+(\mathbf{x}) \geq 0$ are a non-negative functions (more precisely, an unnormalized densities),*
- *$p_-(\mathbf{x}) \leq 0$ is a non-positive function.*

Table 1 provides all the details regarding the notation.

3 Quadratures for integral approximations involving Mix-NCs

The best procedure for approximating integral involving to a mixture $\bar{p}(\mathbf{x})$ of densities with possibly negative weights (Mix-NCs) is related to a quadrature trick [25, 37]. Indeed, if we are interested to approximate a generic moment or any integral involving to the distribution $\bar{p}(\mathbf{x})$, i.e.,

$$I_{\bar{p}} = \mathbb{E}_{\bar{p}}[f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x})\bar{p}(\mathbf{x})d\mathbf{x}, \tag{14}$$

$$= \frac{1}{\sum_{j=1}^{N} \alpha_j} \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}} f(\mathbf{x})\phi_n(\mathbf{x})d\mathbf{x}, \tag{15}$$

$$= \sum_{n=1}^{N} \bar{\alpha}_n J_n. \tag{16}$$

where $f(\mathbf{x})$ is a generic integrable function. Note that above we have set $J_n = \int_{\mathcal{X}} f(\mathbf{x})\phi_n(\mathbf{x})d\mathbf{x}$ and $\bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N} \alpha_j}$. Since we are able to draw from $\phi_n(\mathbf{x})$, we can approximate each J_n by a simple Monte Carlo procedure (i.e., a stochastic quadrature rule) [32],

$$\widehat{J}_n = \frac{1}{S} \sum_{s=1}^{S} f(\mathbf{x}_n^{(s)}), \qquad \mathbf{x}_n^{(s)} \sim \phi_n(\mathbf{x}). \tag{17}$$

Table 1: Main notation of the work.

Notation	Description
$\phi_n(\mathbf{x}) \geq 0 \quad \forall n$	Normalized density, that we are able to evaluate and draw from
$p(\mathbf{x}) = p_+(\mathbf{x}) + p_-(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x}) \geq 0$	Unnormalized Mix-NCs (non-negative function)
$\bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \sum_{n=1}^{N} \bar{\alpha}_n \phi_n(\mathbf{x}) \geq 0$	Normalized Mix-NCs
$\alpha_n = \alpha_n^+ > 0, \quad n = 1, ..., M$ $\alpha_n = \alpha_n^- < 0, \quad n = N+M, ..., N$	Unnormalized coefficients of the complete Mix-NCs
$0 \leq \bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N} \alpha_j} \leq 1$	Normalized coefficients of the complete Mix-NCs
$p_+(\mathbf{x}) = \sum_{n=1}^{M} \alpha_n^+ \phi_n(\mathbf{x}) \geq 0$	Non-negative function
$p_-(\mathbf{x}) = \sum_{n=1}^{M} \alpha_n^- \phi_n(\mathbf{x}) \leq 0$	Non-positive function
$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+} = \sum_{m=1}^{M} \bar{\alpha}_m^+ \phi_m(\mathbf{x}) \geq 0$	Normalized partial mixture (corresponding to the positive part of $\bar{p}(\mathbf{x})$)
$\bar{p}_-(\mathbf{x}) = \frac{p_-(\mathbf{x})}{\sum_{i=1}^{N-M} \alpha_i^-} = \sum_{m=1}^{M} \bar{\alpha}_k^- \phi_k(\mathbf{x}) \geq 0$	Normalized partial mixture (corresponding to the negative part of $\bar{p}(\mathbf{x})$)
$\bar{p}(\mathbf{x}) = \beta^+ \bar{p}_+(\mathbf{x}) + (1-\beta^+) \bar{p}_-(\mathbf{x}) \geq 0$	Normalized Mix-NCs as function of the partial mixtures
$\beta^+ = \frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{j=1}^{N} \alpha_j} > 1$	Coefficient of the partial mixtures combination

For any generic variables $\bar{\psi}_k$ or functions $\bar{\psi}(\mathbf{x})$ denoted with an *upper bar*, the following conditions hold:

$$\sum_k^K \bar{\psi}_n = 1, \qquad \int_{\mathcal{X}} \bar{\psi}(\mathbf{x}) d\mathbf{x} = 1.$$

Therefore, replacing in the expressions above into $I = \sum_{n=1}^{N} \bar{\alpha}_n J_n$, we obtain the final estimator:

$$\widehat{I}_{\bar{p}} = \sum_{n=1}^{N} \bar{\alpha}_n \widehat{J}_n = \frac{1}{S} \sum_{n=1}^{N} \sum_{s=1}^{S} \bar{\alpha}_n f(\mathbf{x}_n^{(s)}). \tag{18}$$

As $S \to \infty$, we have $\widehat{I}_{\bar{p}} \to I_{\bar{p}}$ [20, 32, 41]. Recall that $\bar{\alpha}_n > 0$ for $n = 1, ..., M$ and $\bar{\alpha}_n < 0$ for $n = M+1, ..., N$. The consistency of $\widehat{I}_{\bar{p}}$ is ensured as $S \to \infty$ by standard MC arguments, since $\widehat{J}_n \to J_n$ and, as a consequence, $\widehat{I}_{\bar{p}} \to I_{\bar{p}}$ [1]. For further details about other convergence properties, involving also N, see [25, Section 7].

4 Sample generation from Mix-NCs

In this section, we discuss two sampling schemes: a rejection sampling method and an importance sampling technique. Note that we are able to generate samples from

$$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+} = \sum_{m=1}^{M} \bar{\alpha}_m^+ \phi_m(\mathbf{x}), \qquad \bar{\alpha}_m^+ = \frac{\alpha_m^+}{\sum_{i=1}^{M} \alpha_i^+}, \tag{19}$$

in a classical way, since $0 \leq \bar{\alpha}_m^+ \leq 1$ and $\sum_{m=1}^{M} \bar{\alpha}_m^+ = 1$. Moreover, by construction, we have

$$p_+(\mathbf{x}) \geq p(\mathbf{x}), \tag{20}$$

that is, the inequality required for applying the RS technique [32, Chapter 3]. Furthermore, $\bar{p}_+(\mathbf{x})$ represents a part (a "piece") of the target density $\bar{p}(\mathbf{x})$ (recall that $\bar{p}(\mathbf{x}) \propto p(\mathbf{x})$). Namely, a generic proposal density $q(\mathbf{x})$ must satisfy the inequality $q(\mathbf{x}) \geq p(\mathbf{x})$ in order to apply rejection sampling correctly. If this condition is violated, the samples generated by the algorithm are no longer distributed according to the target density $\bar{p}(\mathbf{x})$, but instead follow $\widetilde{p}(\mathbf{x}) \propto \min\{p(\mathbf{x}), q(\mathbf{x})\}$ [32, Chapter 3].

This observation highlights the relevance and ability of constructing proposals that satisfy the bound (20). In particular, the proposed choice $q(\mathbf{x}) = p_+(\mathbf{x})$ automatically fulfills this requirement. Moreover, $p_+(\mathbf{x})$ is typically an efficient choice of proposal function within an RS scheme, since it is assembled directly from components of the target density $p(\mathbf{x})$, thereby preserving its main structural features and generally remaining close to it in shape. Additionally, we are able to draw from $\bar{p}_+(\mathbf{x}) \propto p_+(\mathbf{x})$ since it is a classical mixture of pdfs with positive coefficients. Hence, $\bar{p}_+(\mathbf{x})$ constitutes an appropriate choice of proposal density in Monte Carlo methods, specially in a RS technique. Below we outline the proposed RS scheme.

Rejection sampling (RS) for Mix-NCs:

1. Set $s = 1$.
2. Draw a candidate $\mathbf{z}' \sim \bar{p}_+(\mathbf{x})$,
3. With probability

$$p_A(\mathbf{z}') = \frac{p(\mathbf{z}')}{p_+(\mathbf{z}')}, \tag{21}$$

   set $\mathbf{x}^{(s)} = \mathbf{z}'$ and increase $s \leftarrow s + 1$. Otherwise, with prob. $1 - p_A(\mathbf{z}')$ discard $\mathbf{z}'$.
4. if $s \leq S$, repeat from step 2.

Note that $p_A \in [0, 1]$. The algorithm provides exact samples from $\bar{p}(\mathbf{x})$ and its validity is ensured by the inequality (20) [32, Chapter 3]. The acceptance rate A_r is:

$$A_r = \int_{\mathcal{X}} \frac{p(\mathbf{x})}{p_+(\mathbf{x})} \bar{p}_+(\mathbf{x}) d\mathbf{x} = \frac{1}{\sum_{i=1}^{M} \alpha_i^+} \int_{\mathcal{X}} p(\mathbf{x}) d\mathbf{x}, \tag{22}$$

$$= \frac{\sum_{n=1}^{N} \alpha_n}{\sum_{i=1}^{M} \alpha_i^+}, \tag{23}$$

$$= 1 - \frac{\sum_{k=1}^{N-M} \alpha_k^-}{\sum_{i=1}^{M} \alpha_i^+} = 1 - \rho, \tag{24}$$

where we set $\rho = \frac{\sum_{k=1}^{N-M} \alpha_k^-}{\sum_{i=1}^{M} \alpha_i^+}$. If $\rho \to 0$ then $A_r \to 1$ and we have a perfect sampler [32, 41]. Hence, the acceptance rate A_r is close to 1 if the sum of negative weights is close to zero, or the sum of positive weights is much larger than the sum of negative weights. This is particularly interesting for the Gaussian process application: in a GP approximation of a density, the number of negative coefficients should tend to disappear as the number of points in the regression grows, and the hyper-parameters are updated and optimized.
The corresponding importance sampling (IS) with resampling (IS+R) scheme using $\bar{p}_+(\mathbf{x})$ as proposal density is described below.

Importance sampling with resampling (IS+R) for Mix-NCs:

1. Draw $\mathbf{z}_1, ..., \mathbf{z}_S \sim \bar{p}_+(\mathbf{x})$,

2. Assign the weight

$$w_s = \frac{p(\mathbf{z}_s)}{\bar{p}_+(\mathbf{z}_s)} = \left[\sum_{i=1}^{M} \alpha_i^+\right] p_A(\mathbf{z}_s), \quad s = 1, ..., S, \tag{25}$$

   where we have used $p_A(\mathbf{z})$ given in Eq. (21).

3. Define the normalized weights

$$\bar{w}_s = \frac{w_s}{\sum_{i=1}^{S} w_i}, \qquad s = 1, ..., S. \tag{26}$$

4. Resample S times within the set $\{\mathbf{z}_1, ..., \mathbf{z}_S\}$ according to the probability mass function defined by the normalized weights $\bar{w}_s$, with $s = 1, ..., S$, obtaining a new set of unweighted samples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(S)}\}$.

Unlike RS, the IS+R does not return exact samples, but provides samples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(S)}\}$ that are approximately and asymptotically distributed as $\bar{p}(\mathbf{x})$ [20, 41]. However, the quality of this approximation improves as S grows [20, 22, 41]. Furthermore, no samples are rejected/discarded as in the RS scheme. It is also important to remark that the weights w_s present good theoretically properties due to the choice of the proposal density. For instance, since $w_s \propto p_A(\mathbf{z}_s)$, the IS weights are bounded

$$w_s \in \left[0, \sum_{i=1}^{M} \alpha_i^+\right], \tag{27}$$

hence their distribution has not heavy tails, and the variance of the weights is always bounded [11, 24, 44, 48]. Standard IS can produce highly variable estimates, especially when weights have heavy right tails [46]. Namely, extreme values of the weights lead to unstable estimates [44, 46] or yield estimators with infinite variance (see numerical experiments in [23]). However, these undesirable scenarios cannot occur in the proposed IS scheme, due to the property in Eq. (27). Furthermore, note that

$$\bar{w}_s = \frac{w_s}{\sum_{i=1}^{S} w_i} = \frac{p_A(\mathbf{z}_s)}{\sum_{i=1}^{S} p_A(\mathbf{z}_i)}.$$

Recall that the proposal density, described in this work, provides good performance within RS and IS schemes since it is itself a piece of the target density [22].

Histograms. Histograms can be constructed in the usual way using the unweighted samples, that are generated by the RS or by IS+R schemes as well. Alternatively, one can directly use the

weighted samples $\{\mathbf{z}_s, w_s\}_{s=1}^S$ obtained by the IS procedure. Indeed, in each bin of the histogram, instead of considering the number of samples inside the bin, we can sum the corresponding weights. Namely, let consider the bin $\mathcal{B} \subset \mathcal{X}$. The value of the histogram in the bin must be $\sum_{\mathbf{z}_j \in \mathcal{B}} w_j$, where we sum each w_j with j such that $\mathbf{z}_j \in \mathcal{B}$ (for the underlying theory see [29]). The histogram can be normalized dividing all the values by the complete sum of the weights, i.e., $\sum_{s=1}^S w_s$.

5 Mix-NCs as proposal density within IS methods

Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ denote the parameter of interest, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. In Bayesian analysis, all the relevant statistical information is encapsulated in the posterior distribution, given by

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{p(\mathbf{y})}, \tag{28}$$

where $\ell(\mathbf{y}|\mathbf{x})$ denotes the likelihood function, $g(\mathbf{x})$ is the prior density, and $Z = p(\mathbf{y})$ is the Bayesian model evidence (also known as the marginal likelihood). Generally, Z is unknown, so we are able to evaluate the unnormalized target function, $\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$. The analytical computation of integrals involving the posterior density

$$\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x}), \tag{29}$$

is often unfeasible, hence numerical approximations are needed. Hence, in order to extract information about the posterior $\bar{\pi}(\mathbf{x})$, often we are interested in computing integrals which generally involve the product of a generic function f and the posterior $\bar{\pi}$,

$$I_{\bar{\pi}} = \mathbb{E}_{\bar{\pi}}[f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}. \tag{30}$$

Note that the expectation above $\mathbb{E}_{\bar{\pi}}[f(\mathbf{x})]$ is different from the expectation $\mathbb{E}_{\bar{p}}[f(\mathbf{x})]$ given in Eq. (14), since in Eq. (30) the density involved is the posterior $\bar{\pi}$ instead of the mixture $\bar{p}$, i.e., $I_{\bar{\pi}} = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}$. Note that we can write:

$$\begin{aligned}
I_{\bar{\pi}} = \mathbb{E}_{\bar{\pi}}[f(\mathbf{x})] &= \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, && (31)\\
&= \int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}(\mathbf{x})d\mathbf{x},\\
&= \int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\left(\beta^+\bar{p}_+(\mathbf{x}) + (1-\beta^+)\bar{p}_-(\mathbf{x})\right)d\mathbf{x},\\
&= \beta^+\int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}_+(\mathbf{x})d\mathbf{x} + (1-\beta^+)\int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}_-(\mathbf{x})d\mathbf{x},\\
&= \beta^+\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\bar{\pi}(\mathbf{x})] + (1-\beta^+)\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\bar{\pi}(\mathbf{x})], && (32)\\
&= \frac{\beta^+}{Z}\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\pi(\mathbf{x})] + \frac{1-\beta^+}{Z}\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\pi(\mathbf{x})], && (33)
\end{aligned}$$

The expression (32) induces the design of an importance sampling scheme with positive and negative IS weights. The idea is to apply the MC approach to approximate the expectations $\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\pi(\mathbf{x})]$ and $\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\pi(\mathbf{x})]$, as shown below.

Importance sampling with an Mix-NCs as proposal density:

1. Draw S samples from $\bar{p}_+(\mathbf{x})$ and S samples from $\bar{p}_-(\mathbf{x})$, i.e.,

$$\mathbf{x}_s^+ \sim \bar{p}_+(\mathbf{x}), \qquad \mathbf{x}_s^- \sim \bar{p}_-(\mathbf{x}), \qquad s = 1, ..., S. \tag{34}$$

2. To each sample, assign the weights

$$w_s^+ = \beta^+ \frac{\pi(\mathbf{x}_s^+)}{\bar{p}(\mathbf{x}_s^+)} \geq 0, \qquad w_s^- = (1-\beta^+) \frac{\pi(\mathbf{x}_s^-)}{\bar{p}(\mathbf{x}_s^-)} \leq 0, \tag{35}$$

   with $s = 1, ..., S$. Note that, both denominators of weights in Eq. (35) contain the complete mixture $\bar{p}(\mathbf{x})$.

3. **If Z is known**, the resulting estimator is given by the formula:

$$\widehat{I}_{\bar{\pi}} = \frac{1}{SZ} \left(\sum_{j=1}^{S} w_j^+ f(\mathbf{x}_j^+) + \sum_{j=1}^{S} w_j^- f(\mathbf{x}_j^-) \right). \tag{36}$$

4. **If Z is unknown**, estimate and replace Z above with:

$$\widehat{Z} = \frac{1}{S} \left(\sum_{j=1}^{S} w_j^+ + \sum_{j=1}^{S} w_j^- \right). \tag{37}$$

The estimator in Eq. (37) is based on the following equality:

$$\begin{aligned} Z &= \beta^+ \int_{\mathcal{X}} \frac{\pi(\mathbf{x})}{\bar{p}(\mathbf{x})} \bar{p}_+(\mathbf{x}) d\mathbf{x} + (1-\beta^+) \int_{\mathcal{X}} \frac{\pi(\mathbf{x})}{\bar{p}(\mathbf{x})} \bar{p}_-(\mathbf{x}) d\mathbf{x}, \\ &= \beta^+ \mathbb{E}_{\bar{p}_+}[\pi(\mathbf{x})] + (1-\beta^+) \mathbb{E}_{\bar{p}_-}[\pi(\mathbf{x})]. \end{aligned} \tag{38}$$

Note that the samples $\mathbf{x}_s^-$ generated from $\bar{p}_-(\mathbf{x})$ have associated negative weights w_s^-. Thus, the above procedure admits a physical analogy, wherein the samples $\mathbf{x}_s^+$ correspond to "matter" samples, and the samples $\mathbf{x}_s^-$ correspond to "anti-matter" samples, or any other similar physical analogy involving negative-signed particles [42, 43].

Remark 3. *In this designed IS scheme, we draw samples from both partial mixtures, $\bar{p}_+(\mathbf{x})$ and $\bar{p}_-(\mathbf{x})$. The samples drawn from $\bar{p}_-(\mathbf{x})$ (denoted $\mathbf{x}_s^-$) carry negative importance weights.*

6 Applications to GP approximation of a distribution

6.1 Emulation of a posterior density

In many applications, there is the need of constructing (via regression) a non-parametric density (a.k.a., emulator/surrogate model), which mimics a posterior distribution [24, 27, 37]. The resulting emulator can be employed in different ways in Bayesian inference. A first possibility is to apply MC sampling methods considering the surrogate model as an approximate posterior pdf within the MC schemes [7, 47, 9][21, Chapter 9.4.3] or within different quadrature rules [19, 39, 26], instead of the evaluation of a costly true posterior. For instance, this is also the case of the strategy known as *calibrate, emulate, sample*, currently in vogue [8]. In order to improve the efficiency of MC algorithms, a second option is to use the emulator as a proposal density within an MC technique. Here, we focus on the last approach. More precisely, let us consider again a target-posterior distribution,

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) \geq 0,$$

where $\int_{\mathcal{X}} \bar{\pi}(\mathbf{x}) d\mathbf{x} = 1$. We can have access to noiseless evaluations of this target density in a set of points $\mathcal{S} = \{\mathbf{x}_n, t_n\}_{n=1}^N$ with $n = 1, ..., N$,

$$t_n = \pi(\mathbf{x}_n) \geq 0, \tag{39}$$

or noisy evaluations, for instance, with a multiplicative and positive noise perturbation ϵ_n,

$$t_n = \pi(\mathbf{x}_n)\epsilon_n \geq 0, \qquad \epsilon_n \geq 0. \tag{40}$$

In any case, we always have $t_n = t(\mathbf{x}_n) \geq 0$. For more details see [24, 27, 37]. In the general case, fixing $\mathbf{x}$, we have access to $t(\mathbf{x})$ that is a random variable with mean $\mathbb{E}[t(\mathbf{x})] = m(\mathbf{x}) = \pi(\mathbf{x}) - \mu(\mathbf{x})$, and variance $\text{var}[t(\mathbf{x})] = s^2(\mathbf{x})$, where $\mu(\mathbf{x})$ is a possible bias [27].

Emulation versus estimation of a density. GP-based emulation is a *supervised* approach: an unnormalized version of a density is evaluated at selected input locations $\mathbf{x}_n$ and considering also the correspoding outputs t_n, a GP regressor is trained to approximate this function over the domain. In contrast, kernel density estimation (KDE) is an *unsupervised* technique that constructs a density estimate directly from samples drawn from the unknown distribution, without requiring point-wise evaluations of the density itself. In this work, we are focusing in the first setting, i.e., density emulation.

6.2 Surrogate construction by Gaussian processes (GPs)

Let us assume that we apply a Gaussian process (GP) regression model [40]. We consider a kernel function, $k(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, then we can define a $N \times N$ kernel matrix $\mathbf{K}$ where each entry is $[\mathbf{K}]_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$, and a $N \times 1$ kernel vector $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^\top$. For the purpose of density approximation, it is more appropriate to consider kernel functions $k(\mathbf{x}, \mathbf{z})$ that are themselves normalized probability densities [25]. For simplicity in the exposition, we assume

Gaussian kernels,

$$k(\mathbf{x}, \mathbf{z}) = \left(\frac{1}{2\pi\lambda^2}\right)^{\frac{d_X}{2}} \exp\left(-\frac{||\mathbf{x}-\mathbf{z}||^2}{2\lambda^2}\right), \quad \lambda > 0. \tag{41}$$

Clearly, many other choice of kernels are possible. Given the pairs $\mathcal{S} = \{\mathbf{x}_n, t_n\}_{n=1}^N$, and defining also the vector $\mathbf{t} = [t_1, ..., t_N]^\top$, the approximation of $\pi(\mathbf{x})$ is given by the formulas:

$$p(\mathbf{x}|\mathcal{S}) = \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \eta \mathbf{I}_N)^{-1}\mathbf{t}, \qquad \eta \geq 0, \tag{42}$$

$$= \mathbf{k}(\mathbf{x})^\top \boldsymbol{\alpha}, \tag{43}$$

$$= \sum_{n=1}^N \alpha_n \underbrace{k(\mathbf{x}, \mathbf{x}_n)}_{\phi_n(\mathbf{x})} = \sum_{n=1}^N \alpha_n \phi_n(\mathbf{x}), \tag{44}$$

where $\mathbf{I}_N$ is a $N \times N$ identity matrix, and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^\top$ is defined as

$$\boldsymbol{\alpha} = (\mathbf{K} + \eta \mathbf{I}_J)^{-1}\mathbf{t}. \tag{45}$$

The values α_n play the same role of the coefficients in Eq. (2), and the kernel functions play the role of densities in Eq. (2), i.e.,

$$\phi_n(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}_n) = \left(\frac{1}{2\pi\lambda^2}\right)^{\frac{d_X}{2}} \exp\left(-\frac{||\mathbf{x}-\mathbf{z}||^2}{2\lambda^2}\right). \tag{46}$$

Recall that $p(\mathbf{x}|\mathcal{S})$ is an unnormalized density that approximates the unnormalized target $\pi(\mathbf{x})$. Below, we derive its normalized version, $\bar{p}(\mathbf{x}|\mathcal{S}) \propto p(\mathbf{x}|\mathcal{S})$, in Eq. (48) below, which serves as an approximation of the normalized target $\bar{\pi}(\mathbf{x})$.

Remark 4. *Note that η in Eq. (42) must be non-negative $\eta \geq 0$, and now assume $\eta = 0$, that corresponds to the interpolation scenario (i.e., maximum overfitting). Even in this setting, it is always possible to choose a value of λ such that $p(\mathbf{x}) \geq 0$. In particular, there exists a threshold λ_{max} such that, for all $\lambda \leq \lambda_{max}$, the condition $p(\mathbf{x}) \geq 0$ is satisfied. Note that, as $\lambda \to 0$, we obtain the limiting case $p(\mathbf{x}) = t_n \cdot \delta(\mathbf{x} - \mathbf{x}_n)$.*

Hence, we can always choose a set of parameters λ, η such that $p(\mathbf{x}) \geq 0$. However, some of the coefficients α_n may still be negative. The condition $p(\mathbf{x}) \geq 0$, nevertheless, implies that

$$\sum_{n=1}^N \alpha_n \geq 0, \tag{47}$$

since $\phi_n(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}_n) \geq 0$ for all n.

Remark 5. *The positive and negative values of the coefficients α_n given in Eq. (45) are not necessarily ordered as assumed in Section 2. However, this does not affect the validity or*

applicability of the proposed methodologies. Indeed, the coefficients α_n and their corresponding nonlinear functions $\phi_n(\mathbf{x})$ can always be reordered by redefining the index $n \in 1, \ldots, N$, without altering the final solution, which depends only on the sum $\sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x})$.

The normalized version of the approximated target is obtained as

$$\bar{p}(\mathbf{x}|\mathcal{S}) = \frac{1}{\sum_{n=1}^{N} \alpha_n} p(\mathbf{x}|\mathcal{S}) = \frac{1}{\sum_{n=1}^{N} \alpha_n} \sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x}). \tag{48}$$

Recall that we choose the hyper-parameters λ, η with the constrains that $p(\mathbf{x}|\mathcal{S}) \geq 0$, so that $\sum_{n=1}^{N} \alpha_n > 0$. The hyper-parameters λ, η can be tuned using different procedures, such as marginal likelihood maximization or the leave-one-out cross-validation [23, 33, 40]. By construction, we have $\bar{p}(\mathbf{x}|\mathcal{S}) \approx \bar{\pi}(\mathbf{x})$ as N grows [27, 40].

Table 2: **Gaussian Process Adaptive Importance Sampling (GP-AIS)**

- Initialization: Choose the initial set $\mathcal{S}_0 = \{\mathbf{x}_{0,n}, t_{0,n}\}_{n=1}^{N}$ of nodes, and the values K.
- For $k = 1, \ldots, K$**:**

1. **Emulator construction:** Given the set $\mathcal{S}_{k-1} = \{\mathbf{x}_{k-1,n}, t_{k-1,n}\}_{n=1}^{N}$, build the emulator/proposal function $\bar{p}_k(\mathbf{x}) = \bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1})$, with a non-parametric regression procedure (see Sect. 6.2).

2. **Sampling:** Draw N samples

$$\mathbf{x}_{k,n} \sim \bar{p}_k(\mathbf{x}) = \bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1}), \tag{49}$$

   using one of the method proposed in Section 4.

3. **Updating:** Evaluate $t_{k,n} = \pi(\mathbf{x}_{t,n})$, for all $n = 1, ..., N$, and update the set of nodes appending $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{\mathbf{x}_{k,n}, t_{k,n}\}_{n=1}^{N}$.

- Final weighting: Assign to each sample the weight

$$w_{t,n} = \frac{\pi(\mathbf{x}_{t,n})}{\frac{1}{K}\sum_{\tau=1}^{K} \bar{p}_\tau(\mathbf{x}_{k,n})} = \frac{t_{k,n}}{\frac{1}{K}\sum_{\tau=1}^{K} \bar{p}_\tau(\mathbf{x}_{k,n})}, \qquad \text{for all} \quad k = 1, ..., K, \quad n = 1, ..., N.$$

- Outputs: Final emulator $\bar{p}_{K+1}(\mathbf{x}) = \bar{p}_{K+1}(\mathbf{x}|\mathcal{S}_K)$, and/or the set of weighted particles $\{\mathbf{x}_{k,n}, w_{k,n}\}_{n=1}^{N}$ for $k = 1, ..., K$.

7 Adaptive importance sampling by emulation

The performance of Monte Carlo (MC) algorithms depends critically on the choice of the proposal density. In adaptive schemes, this proposal is iteratively updated using previously generated

samples. In recent years, a wide range of adaptive importance sampling (AIS) algorithms has been proposed in the literature [4]. In many of these approaches, the overall proposal is modeled as a finite parametric mixture of densities [6, 5, 13, 12, 30]. In contrast, here we adopt a non-parametric proposal based on an interpolating construction (emulation). Several attempts of using an interpolating proposal has been done in the literature [18, 28].
At some iteration index $k \in \mathbb{N}$, given a set of support points $\mathcal{S}_{k-1} = \{\mathbf{x}_{k-1,n}, t_{k-1,n}\}_{n=1}^{N}$, here the idea is to build the emulator/proposal function $\bar{p}_k(\mathbf{x}) = \bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1})$ as described in Sect. 6.2. Then, at each iteration, the emulator $\bar{p}_k(\mathbf{x}) = \bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1})$ is improved by incorporating N additional nodes to the set of support points . This additional points are drawn from the emulator, i.e., $\mathbf{x}_{k,n} \sim \bar{p}_k(\mathbf{x}) = \bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1})$, using one of the method proposed in Section 4, and evaluating $t_{k,n} = \pi(\mathbf{x}_{t,n})$.[2]

Clearly, after K iterations, $\mathcal{S}_K$ the final set of support points will contain NK nodes. Note also that all evaluations of the posterior π are used to build the emulator. At the end of the iterative part, we compute the final IS weights $w_{t,n}$, using all the posterior evaluations $\pi_{t,n} = \pi(\mathbf{x}_{t,n})$, which are stored in the inner layer. More specifically, we assign to each sample (drawn also in the inner stage) the weight

$$w_{t,n} = \frac{\pi(\mathbf{x}_{t,n})}{\frac{1}{K}\sum_{\tau=1}^{K} \bar{p}_\tau(\mathbf{x}_{k,n})}, \qquad \text{for all} \quad k = 1, ..., K, \quad n = 1, ..., N. \tag{50}$$

where we have employed a deterministic mixture weighting scheme [45, 14], i.e., the denominator consists of a temporal mixture (e.g., as also suggested in [10]). Note that the weights $w_{t,n}$ are not required in the iterative steps. Hence, they can be computed after the adaptation and sampling steps are finalized. The complete algorithm is described in Table 2. The output is formed by the final emulator $\bar{p}_{K+1}(\mathbf{x}) = \bar{p}_{K+1}(\mathbf{x}|\mathcal{S}_K)$, and/or the set of weighted particles $\{\mathbf{x}_{k,n}, w_{k,n}\}_{n=1}^{N}$ for $k = 1, ..., K$. The GP construction provides smoother solutions that can be directly applied in unbounded domains $\mathcal{X}$. We call the scheme based on these constructions as Gaussian process adaptive importance sampling (GP-AIS). Clearly, the GP emulation requires the inversion of kernel matrix (with a dimension that increases as the number of nodes grows) and the tuning of the hyper-parameters of the kernel function. As shown in Remark 4, the hyper-parameters are chosen such that $p_k(\mathbf{x}|\mathcal{S}_{k-1}) \geq 0$ (clearly, even in this scenario, some α_n can be negative).

Robust scheme. We present some variants/extensions in order to **(a)** reduce the dependence from the initial nodes and **(b)** speed up the convergence of the emulator covering quickly the state space. This is obtained combining the non-parametric proposal emulator $\bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1})$ with a parametric proposal density, $q_{\texttt{par}}(\mathbf{x})$. Hence, the complete proposal, denoted as $\varphi_t(\mathbf{x})$, will be a mixture of densities with a parametric and a non-parametric components,

$$\varphi_k(\mathbf{x}) = \gamma_k \bar{q}_{\texttt{par}}(\mathbf{x}) + (1 - \gamma_k)\bar{p}_k(\mathbf{x}|\mathcal{S}_{k-1}), \tag{51}$$

where $\gamma_k \in [0, 1]$ for all t, and α_t is a non-increasing function t. The idea is to set initially $\gamma_0 = \frac{1}{2}$, and then decrease $\gamma_k \to \beta_\infty$ as $k \to \infty$ (e.g., we can set $\gamma_\infty = 0$). Note that $\varphi_k(\mathbf{x})$ must be evaluated in the denominator of the weights $w_{k,n}$ in (50), taking the place of $\bar{p}_\tau(\mathbf{x}_{k,n})$ in the denominator of Eq. (50), and also used in (49).

[2] Without loss of generality, and to simplify the exposition, we assume noiseless evaluations.

Remark 6. *Choosing $\bar{q}_{par}(\mathbf{x})$ with heavier tails than $\bar{\pi}(\mathbf{x})$ ensures that $\varphi_k(\mathbf{x})$ also has heavier tails than $\bar{\pi}(\mathbf{x})$. Consequently, this avoids the infinite-variance issue in the importance sampling weights.*

8 Numerical simulations

In this section, we first test the proposed RS technique described in Section 4 and then, in the second section, we test the IS scheme introduced in Section 5.[3]

8.1 RS for drawing samples from the emulator

Let us consider that we desire to emulate (i.e., approximate by regression) the following multimodal target density

$$\bar{\pi}(x) \propto \pi(x) = \sin(x)^2 \exp\left(-\frac{x^2}{30}\right),$$

that is shown in solid line in Figure 1(a). We consider three scenarios.

Scenario 1: We consider $N = 9$ points in the regression, more precisely,

$$x_i \in \{-5, -4, -1, 0, 0.5, 1, 2, 5, 10\}, \tag{52}$$

and $t_i = \pi(x_i)$. We apply the GP regressor with Gaussian kernel and hyper-parameters $\lambda = 1$, $\eta = 0$. In this case, we obtain 6 positive coefficients and 3 negative coefficients in the vector $\boldsymbol{\alpha}$. The 3 negative coefficients are associated to the kernels localized at $x_i = 0, 0.5$ and 2. Applying the proposed RS scheme, the histogram of the accepted samples is given in Figure 1(d). The acceptance rate $A_r = 1-\rho = 0.417$. The unnormalized densities $p(x)$, $p_+(x)$ and the corresponding versions are also shown in Figure 1.

Scenario 2: Now, we consider $N = 11$ points in the regression,

$$x_i \in \{-8, -5, -4, -1.2, -0.8, 0.5, 1, 2, 5, 8, 10\}, \tag{53}$$

and again $t_i = \pi(x_i)$, as shown in Figure 2. We apply the GP regressor with Gaussian kernel and hyper-parameters $\lambda = 0.4$, $\eta = 0$. In this case, we have a unique negative coefficient, α_6, corresponding to $x_6 = 0.5$. The acceptance rate $A_r = 1 - \rho = 0.974$, that is sensibly greater than in scenario 1. However, in both scenarios, we obtain more than reasonable acceptance rates, due to the suitable choice of the proposal density.

Scenario 3: We consider again the same $N = 9$ input points in Eq. (52) of Scenario 1. However, now we apply the GP regressor with different hyper-parameters of the Gaussian kernel, more precisely, $\lambda = 2$ and $\eta = 0.5$. Since $\eta > 0$ we have a regression instead of interpolation. In this scenario, we have two negative coefficients, α_4 and α_5. The acceptance rate $A_r = 0.503$ (i.e., 50%), that is again a good acceptance rate.

[3]Related Matlab code is given at `http://www.lucamartino.altervista.org/public_code_NegMix2025.zip`.

8.2 Emulator with negative coefficients as proposal density within IS

Let us consider again the following target density

$$\bar{\pi}(x) \propto \pi(x) = \sin(x)^2 \exp\left(-\frac{x^2}{30}\right).$$

For clarity of exposition and to help the understanding of the interested reader, we consider a simple emulator formulation in which negative coefficients appear and are explicitly shown. We consider an emulator with $N = 2$ that can be expressed as

$$\bar{p}(x) \propto p(x) = \alpha_1 k(x, x_1) + \alpha_2 k(x, x_2) = \alpha_1 \phi_1(x) + \alpha_2 \phi_2(x),$$

following Eq. (44). We consider Gaussian kernels, with $\mathbf{x}_1 = 0$, $\mathbf{x}_2 = 1$, $\lambda = 4$ and $\eta > 0$ is chosen such that $\alpha_1 = 3$ and $\alpha_2 = -1$. Then, we can write the (unnormalized) emulator as:

$$p(x) = 3\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right) - \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right).$$

Since $\bar{\alpha}_1 = \frac{\alpha_1}{\alpha_1+\alpha_2} = 1.5$ and $\bar{\alpha}_2 = \frac{\alpha_2}{\alpha_1+\alpha_2} = -0.5$, the normalized mixture/emulator is:

$$\begin{aligned} \bar{p}(x) &= \frac{p(x)}{\alpha_1 + \alpha_2} = \bar{\alpha}_1 \phi_1(x) + \bar{\alpha}_2 \phi_2(x), \\ &= 1.5\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right) - 0.5\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right). \end{aligned} \tag{54}$$

Note that, in this case, we have

$$\beta^+ = \alpha_1 = 1.5, \qquad 1 - \beta^+ = \alpha_2 = -0.5,$$

and $\bar{p}_+(x) = \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right)$ and $\bar{p}_-(x) = \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right)$, i.e., the partial mixtures are formed by just one component. Hence, with the previous definitions, we have

$$\bar{p}(x) = \beta^+ \bar{p}_+(x) + (1 - \beta^+)\bar{p}_-(x). \tag{55}$$

Given Eqs. (54)-(55), we can apply the IS scheme in Section 5 using $\bar{p}(x)$ are proposal density and approximate the following 3 integrals:

$$Z = \int_{-\infty}^{+\infty} \pi(x)dx, \quad I_1 = \int_{-\infty}^{+\infty} x^2 \bar{\pi}(x)dx, \quad I_2 = \int_{-\infty}^{+\infty} x^4 \bar{\pi}(x)dx. \tag{56}$$

We compute the ground-truth values using tight grids and evaluate the relative absolute errors (rel-AEs) of the estimates. The results are averaged over 10^5 independent runs, and a global relative absolute error is obtained as the arithmetic mean of the three rel-AEs. This procedure is repeated for different numbers of samples, $S \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$, drawn from the $\bar{p}_+(x)$ and $\bar{p}_-(x)$. The results are given in Table 3. We can observe that the relative error decreases quickly as S grows.

Table 3: Global relative absolute error as function of S, averaged over 10^5 runs.

$S = 10^2$	$S = 10^3$	$S = 10^4$	$S = 10^5$	$S = 10^6$
0.40	0.19	0.11	0.05	0.04

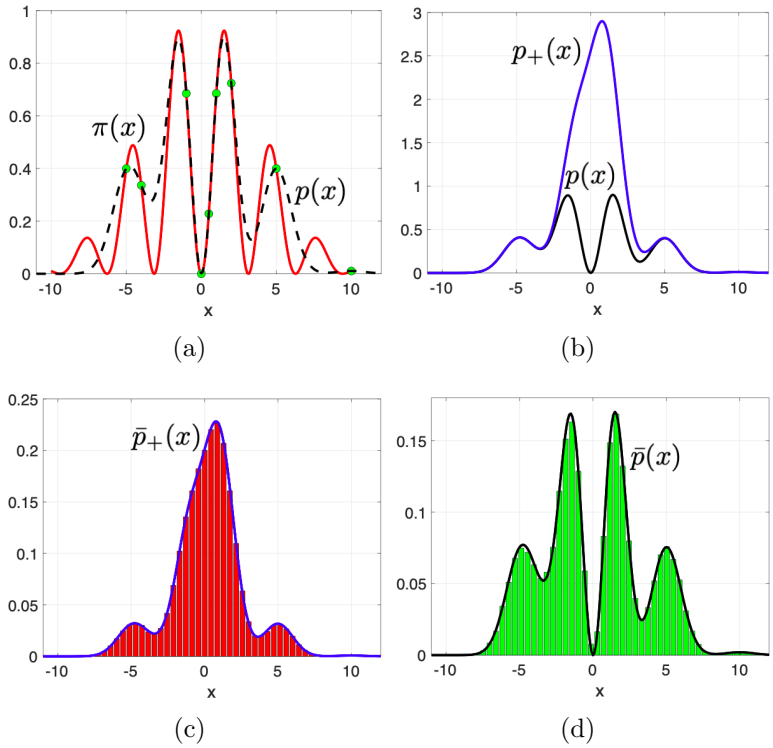


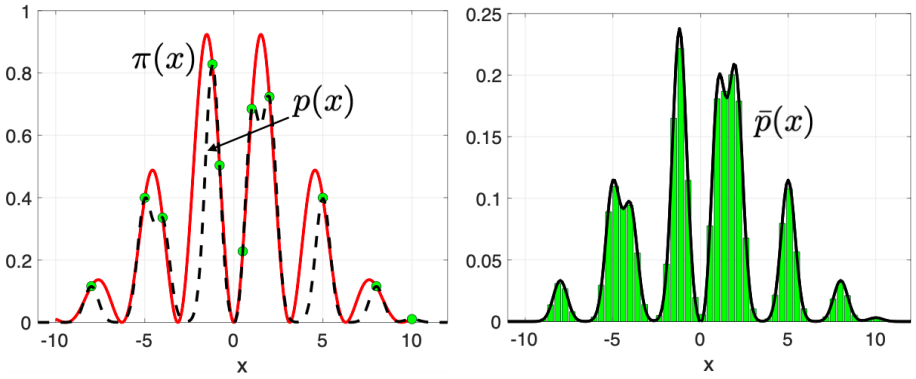
Figure 1: **Scenario 1: (a)** The red solid line shows $\pi(\mathbf{x})$, whereas the black dashed line represents the unnormalized Mix-NCs $p(\mathbf{x})$. The $N = 9$ input points $x_i \in \{-5, -4, -1, 0, 0.5, 1, 2, 5, 10\}$ in regression are depicted in green dots; in this scenario, with $\lambda = 1$, $\eta = 0$, we have 6 positive coefficients and 3 negative coefficients in $\boldsymbol{\alpha}$ (and, hence, in $p(\mathbf{x})$). The 3 negative coefficients are associated to the kernels localized at $x_i = 0$, 0.5 and 2. **(b)** The unnormalized positive partial mixture $p_+(\mathbf{x})$ (blue line) and the unnormalized Mix-NCs $p(\mathbf{x})$ in black line. **(c)** The normalized positive partial mixture $\bar{p}_+(\mathbf{x})$, used as proposal pdf in a RS scheme, and the corresponding histogram. **(d)** Histograms of the accepted samples in the RS scheme distributed according to $\bar{p}(\mathbf{x})$. The theoretical acceptance rate is $A_r = 1 - \rho = 0.417$, and the empirical acceptance rate is ≈ 0.416, in line with the theoretical expression. This means that drawing 20000 samples from $\bar{p}_+(x)$, in one run we obtain $S = 8326$ samples from $\bar{p}(x)$.

8.3 Testing GP-AIS

In this experiment, we consider a multimodal Gaussian target in dimension $d_X = 10$,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3),$$

with $\boldsymbol{\mu}_1 = [5, 0, \ldots, 0]$, $\boldsymbol{\mu}_2 = [-7, 0, \ldots, 0]$, $\boldsymbol{\mu}_3 = [1, \ldots, 1]$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = 4^2\mathbf{I}_{10}$. Note that in this controlled scenario $\bar{\pi}(\mathbf{x}) = \pi(\mathbf{x})$, i.e., $Z = 1$.



(a) (b)

Figure 2: **Scenario 2: (a)** The red solid line shows $\pi(\mathbf{x})$, whereas the black dashed line represents the unnormalized Mix-NCs $p(\mathbf{x})$. The $N = 11$ input points $x_i \in \{-8, -5, -4, -1.2, -0.8, 0.5, 1, 2, 5, 8, 10\}$ in regression are depicted in green dots; ; in this scenario, with $\lambda = 0.6$, $\eta = 0$, we have only one negative coefficient in $\boldsymbol{\alpha}$ (and, hence, in $p(\mathbf{x})$). **(b)** Histograms of the accepted samples in the RS scheme distributed according to $\bar{p}(\mathbf{x})$. The theoretical acceptance rate $A_r = 1 - \rho = 0.974$ which is virtually identical with the empirical acceptance rate obtained, i.e., ≈ 0.974. This means that drawing 20000 samples from $\bar{p}_+(x)$, in one run we obtain $S = 19481$ samples from $\bar{p}(x)$. The corresponding histogram is depicted in figure (b).

Goal. We want to test the performance of the different AIS methods in estimating the normalizing constant $Z = 1$. Specifically, our aim is to test the proposed GP-AIS scheme compared with other AIS algorithms. The total budged of target evaluations is set to $E = 10^3$.

Benchmark AIS methods. We apply the well-known *population Monte Carlo* (PMC)[6], the *deterministic mixture PMC* (DM-PMC) [13] and *layered adaptive IS* (LAIS)[31]. These are adaptive importance sampling (AIS) algorithms in which the proposal distribution(s) are updated at each iteration using information from previously generated samples. In particular, PMC uses multinomial resampling to determine the proposal locations at the next iteration, whereas LAIS updates the proposal location parameters through an MCMC evolution. The goal here is to compare the performance of PMC, LAIS and the GP-AIS scheme with $q_{\texttt{par}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \xi \cdot \mathbf{I}_{10})$ in Eq. (51) (we set and keep fixed $\beta_k = 0.5$). The initial set of N support points are drawn randomly from $q_{\texttt{par}}(\mathbf{x})$.
We use Gaussian kernels/pdfs as the proposal pdfs for all methods. We also need to set the number of these proposals in PMC and LAIS, as well as the dispersion of the Gaussian densities. For PMC, we test different number of proposals $N_{\text{PMC}} \in \{10, 100, 200, 500\}$, whose means are initialized at random in $[-15, 15]^{10}$. At each iteration of PMC, one sample is drawn from each of the N_{PMC} proposal pdfs (and one sample per proposal is drawn in each iteration), hence the algorithm is run for $T_{\text{PMC}} = \frac{E}{N_{\text{PMC}}} = \frac{1000}{N_{\text{PMC}}}$ iterations for a fair comparison. As a second alternative, we consider the deterministic mixture weighting approach for PMC, which is shown to have better overall

performance, denoted DM-PMC [13, 45]. For the LAIS algorithm, we also test different number of proposal density $N_{\text{LAIS}} \in \{10, 100, 200, 500\}$ (again one sample per proposal is drawn in each iteration). We consider the *one-chain* application of LAIS (OC-LAIS), that requires to run one MCMC algorithm targeting $\bar{\pi}(\mathbf{x})$ to obtain the N_{LAIS} location parameters, hence it requires N_{LAIS} evaluations of the target. Then, at each iteration of LAIS, one sample is drawn from the mixture of proposals, hence we run the algorithm for $T_{\text{LAIS}} = E - N_{\text{LAIS}} = 1000 - N_{\text{LAIS}}$ iterations for a fair comparison. For simplicity, we also consider Gaussian random-walk Metropolis to obtain the N_{LAIS} means of the lower IS layer in LAIS.
Regarding the GP-AIS method, we consider $N \in \{10, 100, 200, 500\}$ samples per iterations, so that $K = E/N$, keeping the number of target evaluations $E = 1000$. For all the techniques, the covariance of the Gaussian proposals was set to $\xi^2 \mathbf{I}_{10}$ and we test $\xi = 1, ..., 6$. Recall that in GP-AIS we are using $q_{\texttt{par}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \xi \cdot \mathbf{I}_{10})$. All the methods are compared through the mean absolute error (MAE) in estimating Z, and the results are averaged over 500 independent simulations. The results are given in Table 4. We observe that GP-AIS provides more robust results than the rest of methods. In particular, GP-AIS achieves equal or lower MAE than LAIS, depending on the choice of parameters. Overall, the proposed scheme outperforms all the benchmark AIS methods, such as PMC, DM-PMC, and LAIS.

Table 4: **MAE for estimating Z with $E = 10^3$ total number of evaluations of the target π.** (best and worst MAE of each method are boldfaced)

Methods		$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$	$\xi = 6$
PMC	$N_{\text{PMC}} = 10$	0.9993	0.9526	0.8603	0.6743	0.6024	0.6155
	$N_{\text{PMC}} = 100$	0.9998	0.9896	0.8853	0.6761	0.5192	**0.4544**
	$N_{\text{PMC}} = 200$	**1.0002**	0.9893	0.8816	0.7099	0.6389	0.5384
	$N_{\text{PMC}} = 500$	0.9995	0.9916	0.9741	0.8700	0.7421	0.6544
DM-PMC	$N_{\text{PMC}} = 10$	0.9991	0.9478	0.8505	0.6009	0.5352	0.5814
	$N_{\text{PMC}} = 100$	0.9997	0.8719	0.4490	0.2425	**0.1901**	0.2193
	$N_{\text{PMC}} = 200$	0.9999	0.9321	0.5708	0.3257	0.2374	0.2524
	$N_{\text{PMC}} = 500$	**1.0000**	0.9888	0.7969	0.5009	0.3684	0.3800
OC-LAIS	$N_{\text{LAIS}} = 10$	**1.0000**	1.0000	0.9992	0.9883	0.9468	0.9079
	$N_{\text{LAIS}} = 100$	0.9999	0.8731	0.4434	0.2785	0.2392	0.2870
	$N_{\text{LAIS}} = 200$	0.9982	0.7028	0.2418	0.1243	0.1406	0.2070
	$N_{\text{LAIS}} = 500$	0.9937	0.4949	0.1221	**0.0857**	0.1195	0.1786
GP-AIS	$N = 10$	**0.9511**	0.9278	0.9291	0.9083	0.8965	0.8949
	$N = 100$	0.9252	0.3886	0.1334	0.1487	0.1623	0.1921
	$N = 200$	0.8195	0.2176	**0.1026**	0.1167	0.1418	0.1532
	$N = 500$	0.8417	0.2954	0.1512	0.1178	0.1522	0.1689

9 Conclusions

In this work, we have focused on mixtures with negative coefficients and their applications in computational statistics. These generalized mixture models enable more flexible and accurate density approximations, though they introduce challenges for handling and sampling such distributions. To address these challenges, we proposed efficient Monte Carlo methods (including quadrature techniques, rejection sampling, and importance sampling schemes) capable of accurately approximating integrals and generating (unweighted) samples from these non-convex mixtures. The use of a tailored proposal density ensures both accuracy and efficiency. Furthermore, we have designed how to utilize a mixture with negative coefficients as a proposal density in an importance sampling scheme. In this approach, some generated samples have negative importance weights. Consequently, this method can be likened to a physical analogy where samples with positive importance weights represent "matter", while those with negative importance weights represent "anti-matter". Applications to GP-based density estimation illustrate the practical relevance and effectiveness of the proposed methods, highlighting their potential for broader use in complex density modeling tasks. An adaptive importance sampling (AIS) scheme using a GP approximation as proposal has been also designed. The numerical results show that the proposed GP-AIS scheme outperforms other benchmark AIS methods.

Acknowledgements

This work has been partially supported by the PIACERI Starting Grant BA-GRAPH (UPB 28722052144) and the project PIACERI LikeFree-BA-GRAPH (UPB 28722052159) of the University of Catania.

References

[1] O. D. Akyildiz. Global convergence of optimized adaptive importance samplers. *Foundations of Data Science*, 7(4):944–962, 2025.

[2] D. J. Bartholomew. Sufficient conditions for a mixture of exponentials to be a probability density function. *The Annals of Mathematical Statistics*, 40(6):2183–2188, 1969.

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

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

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

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

[7] J. A. Christen and C. Fox. Markov Chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[8] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *arXiv:2001.03689*, 2020.

[9] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

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

[11] G. Deligiannidis, P. E. Jacob, E. M. Khribch, and G. Wang. On importance sampling and independent Metropolis-Hastings with an unbounded weight function. *arXiv:2411.09514*, pages 1–43, 2025.

[12] Y. El-Laham, P. M. Djurić, and M. F. Bugallo. A variational adaptive population importance sampler. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5052–5056. IEEE, 2019.

[13] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Improving population Monte Carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131:77–91, 2017.

[14] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized multiple importance sampling. *Statistical Science*, 34(1):129–155, 2019.

[15] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[16] M. Felgueiras. Mixtures with negative weights. Technical report, Center for Statistics and Applications, University of Lisbon, 2018.

[17] M. Felgueiras, J. Martins, and R. Santos. Pseudo-convex mixtures. *AIP Conference Proceedings*, 1479(1):1125–1128, 2012.

[18] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[19] M. Kennedy. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8(4):365–375, 1998.

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

[21] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[22] F. Llorente and L. Martino. Optimality in importance sampling: a gentle survey. *arXiv:2502.07396*, 2025.

[23] F. Llorente, L. Martino, D. Delgado, and J. López-Santiago. Marginal likelihood computation for model selection and hypothesis testing: An extensive review. *SIAM Review*, 65(1):3–58, 2023.

[24] F. Llorente, L. Martino, D. Delgado-Gomez, and G. Camps-Valls. Deep importance sampling based on regression for model inversion and emulation. *Digital Signal Processing*, 116:103104, 2021.

[25] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. López-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *IEEE Access*, 8:208462–208483, 2020.

[26] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. Lopez-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *IEEE Access*, 8:208462–208483, 2020.

[27] F. Llorente, L. Martino, J. Read, and D. Delgado-Gómez. A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning and ABC. *International Statistical Review*, 93(1):18–61, 2025.

[28] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing*, 2018(1):5, 2018.

[29] L. Martino, V. Elvira, and G. Camps-Valls. Group importance sampling for particle filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[30] L. Martino, V. Elvira, D. Luengo, and J. Corander. An adaptive population importance sampler: Learning from the uncertanity. *IEEE Transactions on Signal Processing*, 63(16):4422–4437, 2015.

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

[32] L. Martino, D. Luengo, and J. Míguez. *Independent Random Sampling Methods*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[33] L. Martino and J. Read. A joint introduction to Gaussian Processes and relevance vector machines with connections to Kalman filtering and other kernel smoothers. *Information Fusion*, 74:17–38, 2021.

[34] A. Mazza and A. Punzo. Mixtures of multivariate contaminated normal regression models. *Statistical Papers*, 61(2):577–608, 2020.

[35] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[36] I. Murray, D. MacKay, and R. P. Adams. The Gaussian process density sampler. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

[37] C. J. Oates, J. Cockayne, F. X. Briol, and M. Girolami. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

[38] G. Rabusseau and F. Denis. Learning negative mixture models by tensor decompositions. *ArXiv:1403.4224*, 2014.

[39] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

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

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

[42] J.-M. Sellier. A signed particle formulation of non-relativistic quantum mechanics. *Journal of Computational Physics*, 297:48–74, 2015.

[43] M. Troyer and W.-J. Wiese. Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Physical Review Letters*, 94(17):170201, 2005.

[44] M. A. Vazquez and J. Míguez. Importance sampling with transformed weights. *Electronics Letters*, 53(12):783–785, 2017.

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

[46] A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry. Pareto smoothed importance sampling. *J. Mach. Learn. Res.*, 25(1), 2024.

[47] H. Ying, K. Mao, and K. Mosegaard. Moving Target Monte Carlo. *arXiv preprint arXiv:2003.04873*, 2020.

[48] T. Yu, L. Lu, and J. Li. A weight-bounded importance sampling method for variance reduction. *arXiv:1811.09436*, pages 1–14, 2019.

[49] B. Zhang and C. Zhang. Finite mixture models with negative components. In *Machine Learning and Data Mining in Pattern Recognition*, pages 31–41. Springer Berlin Heidelberg, 2005.