Statistics and Computing (2026) 36:14
https://doi.org/10.1007/s11222-025-10763-y

ORIGINAL PAPER

BOB: Bayesian optimized bootstrap for approximate posterior sampling in Gaussian mixture models

Santiago Marin[1] · **Bronwyn Loong**[1] · **Anton H. Westveld**[1,2]

Received: 2 June 2024 / Accepted: 22 October 2025

Abstract

The posterior distribution of a Gaussian mixture model (GMM) provides a natural framework to infer the model parameters or make predictions about a population of interest. That said, sampling from the posterior distribution of GMMs via standard Markov chain Monte Carlo (MCMC) imposes several computational challenges, which have slowed down the adoption of a broader full Bayesian implementation of these models. A growing body of literature has introduced the weighted likelihood bootstrap and the weighted Bayesian bootstrap as alternatives to MCMC sampling. The core idea of these methods is to repeatedly compute maximum *a posteriori* (MAP) estimates from many randomly weighted posterior densities. These MAP estimates then can be treated as approximate posterior draws. Nonetheless, a central question remains unanswered: How to select the random weights under arbitrary sample sizes. We, therefore, introduce the Bayesian optimized bootstrap (BOB), a computational method to automatically tune these random weights by minimizing, through Bayesian optimization, a black-box and noisy version of the reverse Kullback–Leibler (KL) divergence between the Bayesian posterior and an approximate posterior obtained via random weighting. Our proposed method outperforms competing approaches in recovering the Bayesian posterior, while retaining key asymptotic properties from established methodologies. BOB's performance is demonstrated through extensive simulations, along with real-world data analyses.

Keywords Bayesian optimization · finite mixture models · multimodal posterior sampling · weighted Bayesian bootstrap

1 Introduction

Gaussian mixture models (GMMs) are powerful and flexible tools, which turn up naturally when the population of sampling units consists of homogeneous clusters or subgroups (Van Havre et al. 2015), and can be applied in a wide range of scientific problems including model-based clustering (Fraley and Raftery 2002), density estimation (Izenman and Sommer 1988), or as flexible semi-parametric model-based approaches for analyzing complex data (Omori et al. 2007), making GMMs essential tools in modern Statistics and Machine Learning.

Nonetheless, despite the flexibility and wide applicability of GMMs, the large computational costs of Bayesian analysis have slowed down the adoption of a broader full Bayesian implementation of these models. For instance, one could sample from the posterior distribution of GMMs via standard Markov chain Monte Carlo (MCMC). However, due to the nonconcavity of the log-likelihood function, the resulting posterior ends up being multimodal. The multimodality of the posterior, combined with the serial nature of MCMC algorithms, results in a sampler that might get trapped in local regions of high posterior density, failing to explore the entire parameter space and leading to mixing limitations between posterior modes (Celeux et al. 2000; Fong et al. 2019). Moreover, in the context of mixture models, MCMC algorithms scale poorly to large datasets (Ni et al. 2020) and their serial nature has prevented the adoption of recent developments in parallel computing (Newton et al. 2021; Pompe 2021), exacerbating even further the computational limitations of these

✉ Santiago Marin
santiago.marinardila@anu.edu.au

Bronwyn Loong
bronwyn.loong@anu.edu.au

Anton H. Westveld
anton.westveld@anu.edu.au

[1] Research School of Finance, Actuarial Studies and Statistics, The Australian National University, Canberra, ACT, Australia

[2] Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, VA, USA

Published online: 04 November 2025

methods. Thus, new and refined posterior samplers are still required.

This research joins a growing body of literature, dating back to the 1990s, when Newton and Raftery (1994) introduced the weighted likelihood bootstrap (WLB) as an alternative to MCMC. The main idea behind WLB is to generate approximate posterior draws by independently maximizing a series of randomly weighted log-likelihood functions. WLB, however, does not naturally incorporate any prior information in the sampling procedure. With this in mind, researchers have been recently extending the WLB framework to accommodate prior information by weighting not only the likelihood but also the prior. A few examples include the weighted Bayesian bootstrap (WBB), proposed by Newton et al. (2021) and Ng and Newton (2022), and the Bayesian bootstrap spike-and-slab Lasso, proposed by Nie and Ročková (2023a). In the context of generalized Bayesian analysis and model misspecification, we can find the loss-likelihood bootstrap (Lyddon et al. 2019), the posterior bootstrap (Fong et al. 2019; Pompe 2021), and the deep Bayesian bootstrap (Nie and Ročková 2023b).

Motivated by these recent developments within the WLB and WBB literature, in this article, we develop an algorithm to optimize a randomly weighted posterior density associated with a GMM. This optimization allows us to carry out a Bayesian implementation of GMMs within a WBB framework. Additionally, we also address the problem of selecting adequate random weights in order to obtain better posterior approximations. Up until now, one key question has not been addressed: *how to select the random weights under arbitrary—especially under fixed—sample sizes*. Few analyses have been proposed from the lenses of asymptotic theory but, to the best of our knowledge, the core of this question remains unanswered. In fact, as discussed by Newton and Raftery (1994) and Nie and Ročková (2023b), "no general recipe (to select the random weights) yet exists." Thus, we develop the Bayesian optimized bootstrap (BOB), a computational approach to automatically tune the random weights by minimizing a black-box and noisy version of the reverse Kullback–Leibler (KL) divergence between the Bayesian posterior and an approximate posterior obtained via random weighting. Since this divergence is expensive to evaluate and we do not have access to a closed-form expression for the divergence or its derivatives, the minimization is carried out via Bayesian optimization, or BO (and thus the name BOB), as BO is one of the most efficient approaches for optimizing a black-box objective function, requiring only a few evaluations of the objective itself (Jones et al. 1998; Jones 2001). We show that BOB can not only be seen as a generalization of WLB and WBB, but it also leads to a better approximation of the Bayesian posterior, and, unlike standard MCMC algorithms, it is capable of incorporating recent developments in parallel computing.

The rest of this article is organized as follows. Section 2 revisits WBB and describes how a Bayesian implementation of GMMs can be carried out within a WBB framework. In Section 3 we formally introduce BOB and draw connections with existing bootstrap-based posterior samplers. Simulation studies are carried out in Section 4. In Section 5, we demonstrate the applicability of BOB on real-world data. We conclude with a discussion in Section 6.

2 Revisiting weighted Bayesian bootstrap

2.1 Problem setup

For our sampling model, let $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be a random sample from a Gaussian mixture with K components. More precisely, let the density of $\mathbf{y}_i \in \mathbb{R}^d$, for $i \in \{1, \ldots, n\}$, be

$$p\left(\mathbf{y}_i \middle| \left\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right\}_{k=1}^K\right) = \sum_{k=1}^K \pi_k \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{1}$$

where $\pi_k \in [0, 1]$ is the weight of component k so that $\sum_{k=1}^K \pi_k = 1$, $K \geq 2$ is a known integer, $\varphi(\cdot\,;\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the Gaussian density with mean vector $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma}_k \in \mathbb{M}_+^d$, and $\mathbb{M}_+^d$ denotes the set of symmetric positive semidefinite matrices of size $d \times d$. Then, the distribution of the observed data, $\mathbf{Y} = (\mathbf{y}_1', \ldots, \mathbf{y}_n')' \in \mathbb{R}^{n \times d}$, would be given by

$$p\left(\mathbf{Y} \middle| \left\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right\}_{k=1}^K\right) = \prod_{i=1}^n \left\{\sum_{k=1}^K \pi_k \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right\}. \tag{2}$$

For our prior specification, let, for $k \in \{1, \ldots, K\}$,

$$\begin{aligned}
&\boldsymbol{\mu}_k | \boldsymbol{\Sigma}_k \sim N_d(\boldsymbol{\beta}_k,\ \boldsymbol{\Sigma}_k/\lambda_k),\\
&\boldsymbol{\Sigma}_k \sim \text{inverse-Wishart}(\nu_k,\ \boldsymbol{\Psi}_k^{-1}),\\
&\boldsymbol{\pi} \sim \text{Dirichlet}(a_1, \ldots, a_K),
\end{aligned}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ is a vector of mixture proportions and $\{\boldsymbol{\Psi}_k\}_{k=1}^K$ are inverse-Wishart scale matrices. Additionally, let $\boldsymbol{\theta} = \left\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right\}_{k=1}^K \in \boldsymbol{\Theta}$ be the collection of all model parameters so that, under the sampling distribution and priors introduced above, the Bayesian posterior distribution would be, up to a proportionality constant, given by

$$p(\boldsymbol{\theta}|\mathbf{Y}) \propto p(\boldsymbol{\pi}) p(\mathbf{Y}|\boldsymbol{\theta}) \prod_{k=1}^K p(\boldsymbol{\Sigma}_k) p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k). \tag{3}$$

2.2 Posterior sampling via random weighting

We now illustrate how one can approximately sample from the posterior distribution in (3) via WBB. Following Newton et al. (2021), WBB can be summarized in three simple steps:

- Step 1: Start by sampling nonnegative random weights $\mathbf{u} = (u_1, \ldots, u_n)' \in \mathbb{R}_+^n$ from some weight distribution F_u and construct the weighted log-likelihood function of the observed data as

$$\log p_u(\mathbf{Y}|\boldsymbol{\theta}) = \sum_{i=1}^{n} u_i \left\{ \log \sum_{k=1}^{K} \pi_k \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right\}. \tag{4}$$

- Step 2: Sample $\tilde{\mathbf{u}} = (\tilde{u}_{\mu_1}, \ldots, \tilde{u}_{\mu_K}, \tilde{u}_{\Sigma_1}, \ldots, \tilde{u}_{\Sigma_K}, \tilde{u}_{\pi_1}, \ldots, \tilde{u}_{\pi_K})' \in \mathbb{R}_+^{3K}$ from some weight distribution $F_{\tilde{u}}$ and construct the weighted log-prior density as

$$\begin{aligned} \log p_{\tilde{u}}(\boldsymbol{\theta}) \propto \sum_{k=1}^{K} \Bigg\{ & \tilde{u}_{\pi_k} (a_k - 1) \log \pi_k - \tilde{u}_{\Sigma_k} \\ & \left[\left(\frac{\nu_k + d}{2} + 1 \right) \log |\boldsymbol{\Sigma}_k| + \frac{\operatorname{tr}\left(\boldsymbol{\Psi}_k \boldsymbol{\Sigma}_k^{-1}\right)}{2} \right] \\ & - \tilde{u}_{\mu_k} \left[\frac{\lambda_k}{2} (\boldsymbol{\mu}_k - \boldsymbol{\beta}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\beta}_k) \right] \Bigg\}. \end{aligned} \tag{5}$$

- Step 3: The weighted log-posterior density of $\boldsymbol{\theta}$ would be, up to an additive constant, given by $\log p_u(\boldsymbol{\theta}|\mathbf{Y}) \propto \log p_u(\mathbf{Y}|\boldsymbol{\theta}) + \log p_{\tilde{u}}(\boldsymbol{\theta})$. WBB proceeds to maximize this posterior density seeking

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max} \left\{ \log p_u(\boldsymbol{\theta}|\mathbf{Y}) \right\}. \tag{6}$$

Under a traditional Bayesian framework, the randomness in $\boldsymbol{\theta}$ arises from treating it as a random variable with a prior distribution. On the other hand, as discussed in Nie and Ročková (2023b), $\boldsymbol{\theta}^*$ can be seen as an *estimator*, where, given the data, the only source of randomness comes from the random weights. Thus, by fixing the data $\mathbf{Y}$ and repeating steps 1 - 3 many times, one would obtain approximate posterior draws $\{\boldsymbol{\theta}^*_{(s)}\}_{s \in \mathbb{N}}$. This idea is attractive as optimizing can be easier and less computationally intensive than sampling from an intractable posterior. Note, additionally, that the random weights are independent across iterations. Consequently, $\boldsymbol{\theta}^*_{(s)}$ and $\boldsymbol{\theta}^*_{(s')}$, for $s \neq s'$, are also independent and could be sampled in parallel. Moreover, bootstrap-based posterior samplers do not require costly tuning runs, *burn-in* periods, or convergence diagnostics (Fong et al. 2019; Pompe 2021), making random weighting an attractive alternative to MCMC sampling.

Nonetheless, performing the joint optimization in (6) is still a challenge on its own, especially in light of the nonconcavity of $\log p_u(\boldsymbol{\theta}|\mathbf{Y})$. Thus, let us introduce the latent indicator variables $z_{ik} = \mathbb{1}\{i\text{-th observation is drawn from the } k\text{-th component}\}$ so that $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})' \sim \text{Multinomial}(1; \{\pi_k\}_{k=1}^K)$. Then, note that we can rewrite the weighted log-likelihood of the observed data as $\log p_u(\mathbf{Y}|\boldsymbol{\theta}) = \sum_{i=1}^n u_i \log p(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{i=1}^n u_i \{\log p(\mathbf{y}_i, \mathbf{z}_i|\boldsymbol{\theta}) - \log p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta})\}$. As a result, our weighted log-posterior density would be given by

$$\begin{aligned} \log p_u(\boldsymbol{\theta}|\mathbf{Y}) \propto \sum_{i=1}^{n} u_i \Bigg\{ & \sum_{k=1}^{K} z_{ik} \left[\log \pi_k + \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right] \\ & - \sum_{k=1}^{K} z_{ik} \log \pi_k \Bigg\} + \log p_{\tilde{u}}(\boldsymbol{\theta}). \end{aligned}$$

To optimize the above objective function with respect to $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, we treat the latent indicator variables, z_{ik}, as missing data and develop a randomly weighted expectation-maximization (EM) algorithm (Dempster et al. 1977).

2.3 Randomly weighted expectation-maximization

2.3.1 Expectation step

We start by computing the expected value of each z_{ik} conditional on $\mathbf{Y}$, $\mathbf{u}$, and $\boldsymbol{\theta}^{(t)}$, where $\boldsymbol{\theta}^{(t)}$ is the current value of $\boldsymbol{\theta}$ within the EM algorithm. By Bayes' rule, we have that

$$\begin{aligned} & \mathbb{E}\left[z_{ik} | \mathbf{Y}, \mathbf{u}, \boldsymbol{\theta}^{(t)} \right] \\ & = \frac{\exp\left\{ u_i \log \mathbb{P}\left(z_{ik} = 1 | \pi_k^{(t)} \right) + u_i \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)} \right) \right\}}{\sum_{r=1}^{K} \exp\left\{ u_i \log \mathbb{P}\left(z_{ir} = 1 | \pi_r^{(t)} \right) + u_i \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_r^{(t)}, \boldsymbol{\Sigma}_r^{(t)} \right) \right\}} \\ & = \frac{\left[\pi_k^{(t)} \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)} \right) \right]^{u_i}}{\sum_{r=1}^{K} \left[\pi_r^{(t)} \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_r^{(t)}, \boldsymbol{\Sigma}_r^{(t)} \right) \right]^{u_i}} = q_{ik}. \end{aligned} \tag{7}$$

2.3.2 Maximization step

In the maximization step, we then maximize the surrogate objective function $\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\mathbf{Z}|\mathbf{Y},\mathbf{u},\boldsymbol{\theta}^{(t)}} \left[\log p_u(\mathbf{Y}, \mathbf{Z}|\boldsymbol{\theta}) + \log p_{\tilde{u}}(\boldsymbol{\theta}) \right]$ with respect to $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. More formally, our surrogate objective function is given by

$$\begin{aligned}\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = -\frac{1}{2}\sum_{i=1}^{n} u_i \Bigg\{ \sum_{k=1}^{K} q_{ik} \Big[\log|\boldsymbol{\Sigma}_k| \\ +(\mathbf{y}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k)\Big]\Bigg\} \\ - \sum_{k=1}^{K}\Bigg\{ \tilde{u}_{\mu_k}\left[\frac{\lambda_k}{2}(\boldsymbol{\mu}_k - \boldsymbol{\beta}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\beta}_k)\right] \\ + \tilde{u}_{\Sigma_k}\left[\left(\frac{\nu_k + d}{2} + 1\right)\log|\boldsymbol{\Sigma}_k| + \frac{\operatorname{tr}\left(\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_k^{-1}\right)}{2}\right] \\ - \left[\tilde{a}_k - 1 + \sum_{i=1}^{n} u_i q_{ik}\right]\log\pi_k\Bigg\}, \end{aligned} \tag{8}$$

where $\tilde{a}_k = (a_k - 1)\tilde{u}_{\pi_k} + 1$.

Proposition 1 *Choosing $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ to increase our surrogate objective function, defined in (8), results in an increase at least as large in the weighted log-posterior density,* $\log p_u(\boldsymbol{\theta}|\mathbf{Y})$.

The proof of Proposition 1 is given in Appendix A.1. In summary, Proposition 1 suggests that increases in $\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ indicate progress towards maximizing our weighted log-posterior density, $\log p_u(\boldsymbol{\theta}|\mathbf{Y})$. Moreover, as we will illustrate shortly, our surrogate objective function, $\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$, leads to well-known maximization problems, which can be solved relatively cheaply.

Proposition 2 *Let us define* $\tilde{n}_k = \sum_{i=1}^{n} u_i q_{ik}$, $\tilde{\lambda}_k = \tilde{u}_{\mu_k}\lambda_k$, $\tilde{\boldsymbol{\Psi}}_k = \tilde{u}_{\Sigma_k}\boldsymbol{\Psi}_k$, $\tilde{\nu}_k = \tilde{u}_{\Sigma_k}(\nu_k + d + 2) - 2 - d$, $\bar{\mathbf{y}}_k = (\tilde{n}_k)^{-1}\sum_{i=1}^{n}(u_i q_{ik})\mathbf{y}_i$ *and* $\tilde{\mathbf{S}}_k = \sum_{i=1}^{n}(u_i q_{ik})(\mathbf{y}_i - \bar{\bar{\mathbf{y}}}_k)(\mathbf{y}_i - \bar{\bar{\mathbf{y}}}_k)'$, *for all* $k \in \{1, \ldots, K\}$. *Additionally, let* $\bar{\boldsymbol{\beta}}_k = \tilde{\lambda}_k/(\tilde{\lambda}_k + \tilde{n}_k)\boldsymbol{\beta}_k + \tilde{n}_k/(\tilde{\lambda}_k + \tilde{n}_k)\bar{\bar{\mathbf{y}}}_k$, $\bar{\boldsymbol{\Psi}}_k = (\tilde{\boldsymbol{\Psi}}_k + \tilde{\mathbf{S}}_k + \tilde{\lambda}_k\tilde{n}_k/(\tilde{\lambda}_k + \tilde{n}_k)(\bar{\bar{\mathbf{y}}}_k - \boldsymbol{\beta}_k)(\bar{\bar{\mathbf{y}}}_k - \boldsymbol{\beta}_k)')$, $\bar{\lambda}_k = \tilde{\lambda}_k + \tilde{n}_k$, *and* $\bar{\nu}_k = \tilde{\nu}_k + \tilde{n}_k$. *Then, the update of* $\boldsymbol{\mu}_k$ *and* $\boldsymbol{\Sigma}_k$ *would be given by*

$$\left(\boldsymbol{\mu}_k^{(t+1)}, \boldsymbol{\Sigma}_k^{(t+1)}\right) = \underset{\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k}{\arg\max}\left\{h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\}, \tag{9}$$

where

$$\begin{aligned} h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\left(\frac{\bar{\nu}_k + d}{2} + 1\right)\log|\boldsymbol{\Sigma}_k| \\ - \frac{1}{2}\operatorname{tr}\left(\bar{\boldsymbol{\Psi}}_k\boldsymbol{\Sigma}_k^{-1}\right) - \frac{\bar{\lambda}_k}{2}(\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k). \end{aligned}$$

Details on the derivation of $h : \mathbb{R}^d \times \mathbb{M}_+^d \to \mathbb{R}$ are presented in Appendix A.2. To optimize h, we make use of a two-stage procedure in which we first update $\boldsymbol{\Sigma}_k$ and then update $\boldsymbol{\mu}_k$.

- Update $\boldsymbol{\Sigma}_k$: Start by noting that $\boldsymbol{\Sigma}_k \mapsto \log\int_{\mathbb{R}^d}\exp(h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))\mathrm{d}\boldsymbol{\mu}_k$ is a monotone transformation of $\boldsymbol{\Sigma}_k \mapsto h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ (Schilling 2017, Ch. 9). Thus, optimizing (9) with respect to $\boldsymbol{\Sigma}_k$ yields

$$\boldsymbol{\Sigma}_k^{(t+1)} = \underset{\boldsymbol{\Sigma}_k \in \mathbb{M}_+^d}{\arg\max}\left\{-\frac{\bar{\nu}_k + d + 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\operatorname{tr}\left(\bar{\boldsymbol{\Psi}}_k\boldsymbol{\Sigma}_k^{-1}\right)\right\}. \tag{10}$$

  One can identify the solution to (10) as the mode of an inverse-Wishart($\bar{\nu}_k$, $\bar{\boldsymbol{\Psi}}_k^{-1}$) distribution. This is,

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\bar{\boldsymbol{\Psi}}_k}{\bar{\nu}_k + d + 1}. \tag{11}$$

- Update $\boldsymbol{\mu}_k$: Again, note that $\boldsymbol{\mu}_k \mapsto \int_{\mathbb{M}_+^d}\exp(h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))\mathrm{d}\boldsymbol{\Sigma}_k$ is a monotone transformation of $\boldsymbol{\mu}_k \mapsto h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, so optimizing (9) with respect to $\boldsymbol{\mu}_k$ reduces to

$$\boldsymbol{\mu}_k^{(t+1)} = \underset{\boldsymbol{\mu}_k \in \mathbb{R}^d}{\arg\max}\left\{\left[1 + \bar{\lambda}_k(\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k)'\bar{\boldsymbol{\Psi}}_k^{-1}(\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k)\right]^{-(\bar{\nu}_k+1)/2}\right\}, \tag{12}$$

  which corresponds to the mode of a $t_{(\bar{\nu}_k - d + 1)}(\bar{\boldsymbol{\beta}}_k, \bar{\boldsymbol{\Psi}}_k/(\bar{\lambda}_k(\bar{\nu}_k - d + 1)))$ distribution, where $t_\nu(\mathbf{m}, \boldsymbol{\Lambda})$ denotes the t-distribution with ν degrees of freedom, location vector $\mathbf{m}$ and scale matrix $\boldsymbol{\Lambda}$ (Bishop 2006, Ch. 2). More precisely, the update of $\boldsymbol{\mu}_k$ is given by $\boldsymbol{\mu}_k^{(t+1)} = \bar{\boldsymbol{\beta}}_k$.
- Update $\boldsymbol{\pi}$: Lastly, note from (8) that the update of $\boldsymbol{\pi}$ corresponds to the mode of a Dirichlet$(\bar{a}_1, \ldots, \bar{a}_K)$ distribution, where $\bar{a}_k = \tilde{a}_k + \tilde{n}_k$. Namely, for $k \in \{1, \ldots, K\}$,

$$\pi_k^{(t+1)} = \frac{\bar{a}_k - 1}{\left(\sum_{r=1}^{K}\bar{a}_r\right) - K}. \tag{13}$$

Following Proposition 1, our EM algorithm keeps iterating, until convergence, between the expectation and the maximization steps, with the aim of maximizing our weighted log-posterior density.

2.3.3 Dealing with suboptimal modes: Tempered EM

Despite the simplicity of our EM algorithm, a well-known drawback of nonconvex optimization methods is that they might converge to a local optima (i.e., a suboptimal mode). Random restarts have been widely used to increase the parameter space exploration and escape suboptimal modes. However, this approach is too computationally intensive and yet, does not guarantee that one would reach the global mode.

Tempering and annealing (Kirkpatrick et al. 1983; Sambridge 2014), on the other hand, are optimization techniques

which also increase the parameter space exploration at a lower computational cost. More precisely, let $\{T_t\}_{t\in\mathbb{N}}$ be a *tempering profile*, i.e., a sequence of positive numbers such that $\lim_{t\to\infty} T_t = 1$, where t is the t-th iteration of the EM algorithm. Then, a tempered EM algorithm modifies the E step in equation (7) as $\tilde{q}_{ik,t} = q_{ik}^{1/T_t}/(\sum_{r=1}^{K} q_{ir}^{1/T_t})$, where larger values of T_t allow the algorithm to explore more of the target posterior by flattening the distribution of $\tilde{q}_{ik,t}$, and as $t \to \infty$ one would recover the original objective function, progressively attracting the solution towards the global mode (Allassonnière and Chevallier 2021; Lartigue et al. 2022). Thus, we let

$$T_t = 1 + a^{\tau} + b\frac{\sin(\tau)}{\tau}, \qquad \text{where} \qquad \tau = \frac{t + cr}{r}, \quad t \in \mathbb{N}, \tag{14}$$

with $a \in [0, 1)$, $b \in \mathbb{R}$, and $c, r > 0$, which corresponds to the oscillatory tempering profile with gradually decreasing amplitudes from Allassonnière and Chevallier (2021). To select a suitable combination of a, b, c, and r, we use a grid search over a range of possible values and choose the combination that yields the largest objective using an unweighted EM algorithm. Then, we fix such a solution throughout the entire sampler. As discussed in Lartigue et al. (2022), the tempering hyper-parameters can remain fixed across different optimization problems with similar characteristics, as in the case of our randomly weighted EM algorithm.

3 Introducing BOB

So far, we have developed and presented a randomly weighted EM algorithm to approximately sample from the posterior distribution of GMMs within a WBB framework, but we have not yet answered a key and important question: *How to select the random weights*. Newton and Raftery (1994) showed that under low-dimensional settings (this is, when the number of unknown parameters is small and does not grow with the sample size) and assuming the square of Jeffreys prior, likelihood weights drawn from the uniform Dirichlet distribution yield approximate posterior draws that are first order correct, i.e., consistent (they tend to concentrate around a small neighbourhood of the maximum likelihood estimator—MLE) and asymptotically normal. More recently, Ng and Newton (2022) established first order correctness and model selection consistency for a wide range of weight distributions in linear models with Lasso priors, and Nie and Ročková (2023a) showed that, for a number of weight distributions, the approximate posterior from Bayesian bootstrap spike-and-slab Lasso concentrates at the same rate as the actual Bayesian posterior. Note, however, that these results are all based on the assumption that $n \to \infty$. Considering the case of a fixed n is important because, under a small to medium sample size, the prior, $p(\boldsymbol{\theta})$, would have more influence on the posterior, and as the effect of $p(\boldsymbol{\theta})$ gets bigger, the relationship between $p_u(\boldsymbol{\theta}|\mathbf{Y})$ and the random weights becomes less clear (Nie and Ročková 2023b), making the choice of adequate random weights a much more important and difficult task. Additionally, as we will show later (in Sections 4 and 5), standard exponential weighting effectively recovers the posterior distribution of location parameters. However, it struggles much more in recovering the posterior distribution of scale parameters. Thus, we propose BOB as an alternative methodology for selecting adequate random weights under arbitrary sample sizes.

Before illustrating our proposed method, though, let us present a brief summary of variational Bayes (VB) as we draw inspiration from it. VB aims to obtain an approximation, $g_{VB}(\boldsymbol{\theta}|\hat{\boldsymbol{\nu}})$, with variational parameters $\hat{\boldsymbol{\nu}} \in \boldsymbol{\Theta}$, such that $\hat{\boldsymbol{\nu}} = \arg\min_{\boldsymbol{\nu}\in\boldsymbol{\Theta}} \mathbb{E}_{g_{VB}(\boldsymbol{\theta}|\boldsymbol{\nu})}[\log g_{VB}(\boldsymbol{\theta}|\boldsymbol{\nu}) - \log p(\boldsymbol{\theta}|\mathbf{Y})]$, i.e., VB aims to minimize the reverse KL divergence between the Bayesian posterior and a variational approximation (Blei et al. 2017).

With this in mind, we propose the following weighting scheme: Draw the likelihood weights as $\{u_i\}_{i=1}^{n} \overset{\text{iid}}{\sim}$ Exponential(1). Similarly, draw the weights associated with the mean vectors and the mixture proportions as $\{\tilde{u}_{\mu_k}\}_{k=1}^{K} \overset{\text{iid}}{\sim}$ Exponential(1) and $\{\tilde{u}_{\pi_k}\}_{k=1}^{K} \overset{\text{iid}}{\sim}$ Exponential(1), respectively. For the weights associated with the covariance matrices, on the other hand, we set $\{\tilde{u}_{\Sigma_k}\}_{k=1}^{K} \overset{\text{iid}}{\sim} \delta_x$, where δ_x denotes the Dirac measure at x, this is, $\mathbb{P}(\tilde{u}_{\Sigma_k} = x) = 1$. Hence, our problem is reduced to finding an appropriate value for $x \in \mathcal{X} \subset \mathbb{R}$, where $\mathcal{X}$ is a compact search space. Inspired by VB, we propose to select the optimal x as

$$x^* = \arg\min_{x\in\mathcal{X}} \{\mathcal{L}(x)\}, \tag{15}$$

with

$$\begin{aligned}\mathcal{L}(x) &= \int_{\boldsymbol{\Theta}} g_u(\boldsymbol{\theta}|x, \mathbf{Y}) \log\left(\frac{g_u(\boldsymbol{\theta}|x, \mathbf{Y})}{p(\boldsymbol{\theta}|\mathbf{Y})}\right) \mathrm{d}\boldsymbol{\theta} \\ &= \mathbb{E}_{g_u(\boldsymbol{\theta}|x,\mathbf{Y})}\left[\log g_u(\boldsymbol{\theta}|x, \mathbf{Y}) - \log\left(\frac{p(\boldsymbol{\theta})p(\mathbf{Y}|\boldsymbol{\theta})}{p(\mathbf{Y})}\right)\right] \\ &\propto \mathbb{E}_{g_u(\boldsymbol{\theta}|x,\mathbf{Y})}\left[\log g_u(\boldsymbol{\theta}|x, \mathbf{Y})\right] \\ &\quad - \mathbb{E}_{g_u(\boldsymbol{\theta}|x,\mathbf{Y})}\left[\log p(\boldsymbol{\theta}) + \log p(\mathbf{Y}|\boldsymbol{\theta})\right],\end{aligned}$$

where the latter line is known as the negative evidence lower bound (ELBO) (Blei et al. 2017). In other words, we propose to select the optimal x^* so that we minimize the reverse KL divergence between the Bayesian posterior and its approximation induced by random weighting, denoted as $g_u(\boldsymbol{\theta}|x, \mathbf{Y})$. As discussed in Section 2, given the data $\mathbf{Y}$, the only source

of variation in $\boldsymbol{\theta}^*$ comes from the random weights, which, in our weighting scheme, depend on x.

Note, additionally, that our proposed weighting scheme can be seen as a generalization of WLB and WBB. More precisely, by letting $x^* = 1$, we would recover WBB (with a fixed prior weight on the covariance matrices (Ng and Newton 2022)). Moreover, if we let, for $k \in \{1, \ldots, K\}$, $a_k = a_k(n)$, $\lambda_k = \lambda_k(n)$, $\boldsymbol{\Psi}_k = \boldsymbol{\Psi}_k(n)$, and $\nu_k = \nu_k(n)$, such that, as $n \to \infty$, $a_k(n) \to 1$, $\lambda_k(n) \to 0$, $\boldsymbol{\Psi}_k(n) \to \mathbf{0}_{d\times d}$, and $\nu_k(n) \to -(d+2)$, we would recover WLB. Thus, if our search space, $\mathcal{X}$, contains $x = 1$ as a potential solution and our prior beliefs become less informative as the sample size grows large, our proposed method would effectively nest existing bootstrap-based posterior samplers.

That being said, in an ideal setting, we would like to compute x^* as in (15). In practice, however, we cannot optimize $\mathcal{L}(x)$ directly because: (a) we do not know the form of the joint density $g_u(\boldsymbol{\theta}|x, \mathbf{Y})$ and (b) the expectation $\mathbb{E}_{g_u(\boldsymbol{\theta}|x,\mathbf{Y})}[\cdot]$ is analytically intractable. Conveniently, though, we can make use of Sklar's theorem (Sklar 1959) and express $g_u(\boldsymbol{\theta}|x, \mathbf{Y})$ as

$$g_u(\boldsymbol{\theta}|x, \mathbf{Y}) = c(G_{u,1}(\theta_1|x, \mathbf{Y}), \ldots, G_{u,m}(\theta_m|x, \mathbf{Y})) \prod_{j=1}^{m} g_{u,j}(\theta_j|x, \mathbf{Y}),$$

where $c(G_{u,1}(\theta_1|x, \mathbf{Y}), \ldots, G_{u,m}(\theta_m|x, \mathbf{Y}))$ is the copula density of $\boldsymbol{\theta}$ (i.e., c represents the dependence structure in the approximate posterior), $g_{u,j}(\theta_j|x, \mathbf{Y})$ denotes the marginal density of the j-th entry of $\boldsymbol{\theta}$ with cumulative distribution function $G_{u,j}(\theta_j|x, \mathbf{Y})$, and $m = \dim(\boldsymbol{\theta})$. Given posterior draws, $\{\boldsymbol{\theta}^*_{(s)}\}_{s\in\mathbb{N}}$, one could think about estimating such a copula density (see e.g., Provost and Zang (2024) for a review of copula density estimation in the bivariate case); however, obtaining accurate copula density estimates in high dimensional settings is remarkably difficult. Note that the number of unique parameters in our GMM is given by $dK + \frac{d(d+1)}{2}K + (K-1) = \mathcal{O}(d^2)$, which grows quadratically with d. For instance, letting $d = 15$ and $K = 4$ would feature 543 unique parameters, leading to inaccurate and unstable copula density estimates. Thus, we assume that the approximate posterior parameters are mutually independent. More precisely, we let $c(G_{u,1}(\theta_1|x, \mathbf{Y}), \ldots, G_{u,m}(\theta_m|x, \mathbf{Y})) \approx 1$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, so that

$$\log g_u(\boldsymbol{\theta}|x, \mathbf{Y}) \approx \sum_{j=1}^{m} \log g_{u,j}(\theta_j|x, \mathbf{Y}).$$

Therefore, to overcome (a), we propose to obtain a batch of approximate posterior draws, denoted by $\{\boldsymbol{\theta}^*_{(s)}\}_{s=1}^{S_b}$, and estimate each marginal density $g_{u,j}$ with their corresponding kernel density estimate (KDE), denoted by $\hat{g}_{u,j}$. These KDEs can be evaluated efficiently as in Hofmeyr (2021, 2022). To overcome (b), we propose to use the sample mean as an unbiased estimator of the expected value. More formally, for any $x \in \mathcal{X}$, we obtain a batch of approximate posterior draws and estimate $\mathcal{L}(x)$ with

$$\hat{\mathcal{L}}(x) = \frac{1}{S_b}\sum_{s=1}^{S_b}\left\{\left(\sum_{j=1}^{m}\log \hat{g}_{u,j}(\theta^*_{j,(s)}|x, \mathbf{Y})\right) - \log p(\boldsymbol{\theta}^*_{(s)}) - \log p(\mathbf{Y}|\boldsymbol{\theta}^*_{(s)})\right\}. \tag{16}$$

Hence, we can think about $\hat{\mathcal{L}}(x)$ as a noisy and black-box approximation of $\mathcal{L}(x)$, in the sense that, given an input $x \in \mathcal{X}$, we can evaluate the objective, but we do not have access to a closed-form expression for neither the objective nor its derivatives. One could use grid search methods to minimize $\hat{\mathcal{L}}(x)$; however, evaluating $\hat{\mathcal{L}}(x)$ is notably expensive because, for each x, we need to sample a batch of approximate posterior draws, compute univariate KDEs of the approximate marginals, and compare those to the Bayesian posterior. As a consequence, exhaustive grid search approaches would be infeasible to implement. To overcome this, we use BO to minimize $\hat{\mathcal{L}}(x)$ as it is one of the most efficient approaches to optimize a noisy black-box objective with little evaluations (Jones et al. 1998; Jones 2001).

Comments on the independence assumption between approximate posterior parameters: To construct $\hat{\mathcal{L}}(x)$, we assume that the approximate posterior parameters are independent, as this is a widely implemented technique in the Bayesian literature. For instance, mean-field variational Bayes (MFVB) assumes that the variational distribution factorizes as $g_{VB}(\boldsymbol{\theta}|\boldsymbol{\nu}) = \prod_{j=1}^{m} g_{VB,j}(\theta_j|\nu_j)$. In such cases, the MFVB objective function is given by

$$\mathbb{E}_{g_{VB}(\boldsymbol{\theta}|\boldsymbol{\nu})}\left[\left(\sum_{j=1}^{m}\log g_{VB,j}(\theta_j|\nu_j)\right) - \log p(\boldsymbol{\theta}) - \log p(\mathbf{Y}|\boldsymbol{\theta})\right],$$

which closely resembles our noisy black-box objective in equation (16). In a GMM context, however, one could argue that the independence assumption between model parameters might be unreasonable. For instance, the latent indicator variables, $\mathbf{Z}$, depend on each other (i.e., if an observation belongs to one cluster, it cannot belong to any other clusters), as well as on the mean vectors and covariance matrices. In such cases, the MFVB distribution, $\prod_{j=1}^{m} g_{VB,j}(\theta_j|\nu_j)$, might not contain the true posterior as it is not able to capture such a complicated dependence structure, negatively affecting the entire analysis. Thankfully, this is not our case. Unlike MFVB, our goal is not to obtain a variational distribution to approximate the true posterior, but to obtain adequate random weights within a WBB framework. We obtain our approximate posterior draws through a randomly weighted EM

algorithm, where the latent variables, $\mathbf{Z}$, are integrated out in the E-step, incorporating those dependencies in the posterior draws. Additionally, recall from the M-step that the updates of $\boldsymbol{\beta_k}$, $\boldsymbol{\Sigma}_k$, and $\boldsymbol{\pi}$ do not depend on each other, giving room to assume that $\log \hat{g}_u(\boldsymbol{\theta}|x, \mathbf{Y}) \approx \sum_{j=1}^{m} \log \hat{g}_{u,j}(\theta_j|x, \mathbf{Y})$.

3.1 Minimizing $\hat{\mathcal{L}}$ via Bayesian optimization

We now illustrate how one can minimize $\hat{\mathcal{L}}$ via BO. First up, though, note that maximizing $\Upsilon(x) = -\hat{\mathcal{L}}(x)$ is equivalent to minimizing $\hat{\mathcal{L}}(x)$. Thus, we consider the problem

$$x^* = \arg\max_{x \in \mathcal{X}} \{\Upsilon(x)\}.$$

At first sight, the above maximization problem might look relatively simple. However, there are three main limitations. (1) The function $\Upsilon : \mathcal{X} \to \mathbb{R}$ is a black-box, i.e., we do not have access to a closed-form expression for neither Υ nor its derivatives; (2) evaluating Υ is notably expensive; and (3) the function Υ generates outputs that are contaminated by noise, i.e., the outputs are stochastic. Limitations two and three are due to the fact that each evaluation of Υ requires random sampling from an approximate posterior distribution.

Unlike grid search approaches, which evaluate the objective function over an exhaustive predefined grid of x values, BO aims to solve optimization problems of this type, *sequentially*. The idea is to specify a prior over Υ and then keep refining this surrogate model as new data (i.e., new evaluations) are observed (Shahriari et al. 2016).

Thus, we need to start by specifying a prior on our objective function. Due to its flexibility, a Gaussian process (GP) seems like a natural choice for modeling Υ. To be precise, GPs are capable of modeling a wide range of functions, they can naturally incorporate the random noise in the outputs $\Upsilon(x)$, and their respective posterior distributions exist in closed-form, allowing for more efficient computations. Consequently, we let

$$\Upsilon \sim \text{GP}(\phi, \kappa),$$

where $\phi : \mathcal{X} \to \mathbb{R}$ is a mean function and $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric positive semidefinite kernel (covariance) function, so that, for all $x \in \mathcal{X}$, $\Upsilon(x) \sim N(\phi(x), \kappa(x, x))$.

After observing the evaluations $\{(x_l,\ \hat{y}_l)\}_{l=1}^{L}$, the resulting posterior $\Upsilon \mid \{(x_l,\ \hat{y}_l)\}_{l=1}^{L}$ would also be a GP with mean and covariance functions given by

$$\hat{\phi}(x) = \boldsymbol{k}'(\mathbf{K} + \eta^2 \mathbf{I}_L)^{-1} \hat{\mathbf{y}}, \tag{17}$$

$$\hat{\kappa}(x, \tilde{x}) = \kappa(x, \tilde{x}) - \boldsymbol{k}'(\mathbf{K} + \eta^2 \mathbf{I}_L)^{-1} \boldsymbol{v}, \tag{18}$$

where $\hat{y}_l = \Upsilon(x_l) + \epsilon_l$, $\epsilon_l \sim N(0, \eta^2)$, $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_L)' \in \mathbb{R}^L$, $\boldsymbol{k} = (k_1, \ldots, k_L)' \in \mathbb{R}^L$, $\boldsymbol{v} = (v_1, \ldots, v_L)' \in \mathbb{R}^L$, $k_l = \kappa(x, x_l)$, $v_l = \kappa(\tilde{x}, x_l)$, and $(\mathbf{K})_{l,l'} = \kappa(x_l, x_{l'})$ (Rasmussen and Williams 2005). Additionally, as suggested by Snoek et al. (2012), we consider a Matèrn 2.5 kernel function. This is,

$$\kappa(x, \tilde{x}) = \zeta_0 \left(1 + \sqrt{5r^2(x, \tilde{x})} + \frac{5}{3} r^2(x, \tilde{x})\right) \exp\left\{-\sqrt{5r^2(x, \tilde{x})}\right\},$$

where ζ_0 is a covariance amplitude, $r^2(x, \tilde{x}) = (x - \tilde{x})^2 / \zeta^2$, and ζ is a GP length scale.

After specifying our prior on Υ, we then need to answer another question: How to efficiently determine the next point for evaluation, i.e., x_{L+1}. By using the GP posterior, $\Upsilon \mid \{(x_l,\ \hat{y}_l)\}_{l=1}^{L}$, BO constructs an acquisition function, $\xi : \mathcal{X} \to \mathbb{R}$, and chooses its maximum as our next point for evaluation, i.e., $x_{L+1} = \arg\max_{x \in \mathcal{X}} \xi(x)$ (Kandasamy et al. 2020). For our acquisition function, we consider the expected improvement over the best current value (Jones et al. 1998) as it is efficient in the number of evaluations required to find the global maximum of a wide range of black-box functions (see e.g., Bull (2011) and Snoek et al. (2012)). More precisely, given the best current value—denoted by x^+, we set

$$\xi(x) = \mathbb{E}\left[\max\left\{0,\ \Upsilon(x) - \Upsilon(x^+)\right\} \mid \{(x_l, \hat{y}_l)\}_{l=1}^{L}\right].$$

In short, at each iteration, we update our GP surrogate model with new data points, while the acquisition function suggests our next point for evaluation. Our BO procedure keeps iterating between these two steps until reaching a maximum number of evaluations, $L_{\max}$. Throughout this article, we set $L_{\max} = 100$, as it nicely balances the trade-off between computational cost and numerical accuracy.

That being said, note that the BO procedure is carried out just once. Then, we use the learned random weights throughout the entire sampling process—as described in Section 2. Additionally, obtaining the batch of approximate posterior draws, computing the univariate KDEs, and evaluating the Bayesian posterior, can all be trivially implemented in parallel, reducing the cost of evaluating and minimizing $\hat{\mathcal{L}}$. Thus, like WLB and WBB, BOB also embraces recent developments in parallel computing. BOB, however, let us select more appropriate random weights.

To construct our Bayesian optimizer, we make use of the well-established "`DiceKriging`" and "`DiceOptim`" R packages (Roustant et al. 2012; Le Riche and Picheny 2021). Our implementation of BOB, as well as of WBB, are all available in the "`BOBgmms`" R package, which can be found in the supplementary materials or online at `github.com/marinsantiago/BOBgmms`. Source code to reproduce the results from this article can also be found in the supplementary materials or online at

github.com/marinsantiago/BOBgmms-examples. We summarize our proposed method in Algorithm 1.

Algorithm 1 Bayesian Optimized Bootstrap

Input:
Data: $\mathbf{Y}$
Total number of posterior draws: S
Prior hyper-parameters: $\{a_k,\ \lambda_k,\ \nu_k,\ \boldsymbol{\beta}_k,\ \boldsymbol{\Psi}_k\}_{k=1}^{K}$
Batch size: S_b
Compact search space: $\mathcal{X}$
Tempering hyper-parameters: $\{a,\ b,\ c,\ r\}$

Output:
Posterior draws: $\left\{\boldsymbol{\theta}^*_{\text{BOB},(s)}\right\}_{s=1}^{S}$

Compute $x^* \leftarrow \arg\min_{x\in\mathcal{X}} \hat{\mathcal{L}}(x)$ via Bayesian Optimization
for $s \in \{1 \ldots, S\}$ **do** ▷ in parallel
  Set $\tilde{u}_{\Sigma_k} \leftarrow x^*,\ \forall k \in [K]$
  Sample $\{\tilde{u}_{\mu_k}\}_{k=1}^{K} \overset{\text{iid}}{\sim} \text{Exponential}(1)$
  Sample $\{\tilde{u}_{\pi_k}\}_{k=1}^{K} \overset{\text{iid}}{\sim} \text{Exponential}(1)$
  Sample $\{u_i\}_{i=1}^{n} \overset{\text{iid}}{\sim} \text{Exponential}(1)$
  Compute $\boldsymbol{\theta}^*_{\text{BOB},(s)} \leftarrow \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \log p_u(\boldsymbol{\theta}|\mathbf{Y})$ via Tempered EM
end for

3.2 Asymptotic properties

Throughout this subsection, we want to show that BOB retains key asymptotic properties from WLB and WBB. Recall, from Section 3, that BOB can be seen as a generalization of WLB and WBB. More formally, by letting $x^* = 1 \in \mathcal{X}$ or if our prior beliefs become less informative as the sample size grows large, we would recover WBB and WLB, respectively. Therefore, let $\hat{\boldsymbol{\theta}}_{\text{MLE}} \in \boldsymbol{\Theta}$ be the MLE for $\boldsymbol{\theta}$ and let $\boldsymbol{\theta}^*_{\text{BOB}} \in \boldsymbol{\Theta}$ be a draw from BOB. It is assumed that the ordering of $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ and $\boldsymbol{\theta}^*_{\text{BOB}}$ is the same. Let also, for $k \in \{1, \ldots, K\}$, $a_k = a_k(n)$, $\lambda_k = \lambda_k(n)$, $\boldsymbol{\Psi}_k = \boldsymbol{\Psi}_k(n)$, and $\nu_k = \nu_k(n)$, such that, as $n \to \infty$, $a_k(n) \to 1$, $\lambda_k(n) \to 0$, $\boldsymbol{\Psi}_k(n) \to \mathbf{0}_{d\times d}$, and $\nu_k(n) \to -(d+2)$. Thus, if $\{\mathbf{y}_i\}_{i=1}^{n}$ are random samples from a sufficiently regular model $p(\mathbf{y}|\boldsymbol{\theta}_0)$, as described in Section 2.1, and if the BOB solution converges to the WBB or WLB solutions, as $n \to \infty$, then $\boldsymbol{\theta}^*_{\text{BOB}}$ would retain key asymptotic properties such as consistency and asymptotic normality. More precisely, following theorems 1 and 2 from Newton and Raftery (1994), along with the results from Newton et al. (2021), we have that if $x^* \to 1$, as $n \to \infty$, then:

1. For any scalar $\varepsilon > 0$, as $n \to \infty$,

$$\mathbb{P}_u(\|\boldsymbol{\theta}^*_{\text{BOB}} - \hat{\boldsymbol{\theta}}_{\text{MLE}}\| > \varepsilon \mid \{\mathbf{y}_i\}_{i=1}^{n}) \to 0,$$

for almost every infinite sequence of data $\{\mathbf{y}_i\}_{i\in\mathbb{N}}$.

2. For all measurable $\mathbf{B} \in \mathcal{B}(\boldsymbol{\Theta})$, as $n \to \infty$,

$$\mathbb{P}_u(\sqrt{n}(\boldsymbol{\theta}^*_{\text{BOB}} - \hat{\boldsymbol{\theta}}_{\text{MLE}}) \in \mathbf{B} \mid \{\mathbf{y}_i\}_{i=1}^{n}) \to \mathbb{P}(\mathbf{T} \in \mathbf{B}),$$

for almost every infinite sequence of data $\{\mathbf{y}_i\}_{i\in\mathbb{N}}$. In this case, $\mathbf{T} \sim N_m(\mathbf{0}, \mathcal{I}(\boldsymbol{\theta}_0)^{-1})$, $\mathcal{I}(\boldsymbol{\theta})$ is the Fisher information, $m = \dim(\boldsymbol{\theta})$, and $\mathcal{B}(\boldsymbol{\Theta})$ denotes the Borel field on $\boldsymbol{\Theta}$.

Remark 1 The probabilities, $\mathbb{P}_u$, from above, refer to the distribution of $\boldsymbol{\theta}^*_{\text{BOB}}$ induced by random weights given the sequence of data $\{\mathbf{y}_i\}_{i=1}^{n}$.

Remark 2 The Bernstein–von Mises theorem states that, under regularity conditions, the actual Bayesian posterior converges to a normal distribution. More formally, as $n \to \infty$, $\{\boldsymbol{\theta}|\mathbf{Y}\} \to N_m(\boldsymbol{\theta}_0, [n\,\mathcal{I}(\boldsymbol{\theta}_0)]^{-1})$. Comparing this result with our previous results, one have that it is possible to approximate posterior credible sets with BOB as $\mathbb{P}(\boldsymbol{\theta}^*_{\text{BOB}} \in \mathbf{B} \mid \mathbf{Y}) \approx \int_{\mathbf{B}} p(\boldsymbol{\theta}|\mathbf{Y})\mathrm{d}\boldsymbol{\theta}$, for all $\mathbf{B} \in \mathcal{B}(\boldsymbol{\Theta})$, and the approximation would get better with a growing n.

On the whole, we have that BOB provides an automatic and much more informed approach to select the random weights, while retaining the asymptotic first order correctness from existing bootstrap-based posterior samplers.

4 Simulation studies

We now evaluate the performance of different approximate posterior samplers through various simulation experiments.

4.1 Simulation settings

To generate the simulated data, we start by sampling each $\mathbf{z}_i$, for $i \in \{1, \ldots, n\}$, from $p(\mathbf{z}_i|K) \propto \prod_{k=1}^{K}(\frac{1}{K})^{z_{ik}}$, where $K \in \{2, 3, 4\}$. We also consider different dimensions $d \in \{5, 10, 15\}$, and generate each mean vector, for $k \in \{1, \ldots, K\}$, as

$$\boldsymbol{\mu}_k = \left((5k-4)\mathbf{1}'_{\lceil 0.6d\rceil}, \mathbf{0}'_{(d-\lceil 0.6d\rceil)}\right)' \in \mathbb{R}^d,$$

so that only 60% of the entries of $\boldsymbol{\mu}_k$ are important parameters in separating the clusters, while the remaining entries are set to zero. Lastly, we set each entry of $\boldsymbol{\Sigma}_k$ to be $(\boldsymbol{\Sigma}_k)_{j,j'} = \{-0.5^{|j-j'|}\}_{j,j'=1}^{d}$ to induce positive and negative correlations between the features. In total, we consider nine different simulations settings, which are summarized in Table 1. In all cases, we center the data so that each feature has mean zero.

The prior hyper-parameters are set, for each $k \in \{1, \ldots, K\}$, as $\boldsymbol{\beta}_k = \mathbf{0}_d$, $\boldsymbol{\Psi}_k = \mathbf{I}_d$, $a_k = 1.1$, $\lambda_k = \lambda$, and $\nu_k = \nu$. To

Table 1 Simulation settings

Simulation parameters	Setting								
	1	2	3	4	5	6	7	8	9
n	50	50	50	100	100	100	150	150	150
d	5	10	15	5	10	15	5	10	15
K	2	2	2	3	3	3	4	4	4

select the regularization parameters, λ and ν, we use a ten-fold likelihood-based cross-validation approach, similar as in Friedman et al. (2008). The idea is to randomly split the data into ten subsets. We then run an unweighted EM algorithm using each pair of λ and ν on nine training subsets of the data. Given the learned model parameters, we then evaluate the log-likelihood at the remaining validation set. This process is repeated ten times, so that each subset is used as a validation set once. We then choose the pair $(\lambda,\ \nu)$ that maximizes the average log-likelihood over the ten validation sets. The same values of λ and ν are then used across all posterior samplers. That being said, the goal of these experiments is not to choose the "best" regularization parameters. Instead, as pointed out by Nie and Ročková (2023b), the goal is to compare different posterior approximations for given values of λ and ν.

We compare BOB against two versions of WBB, namely WBB1 (with random prior weights) and WBB2 (with fixed prior weights). To implement WBB1 we set $\mathbf{u} \overset{\text{iid}}{\sim}$ Exponential(1) and $\tilde{\mathbf{u}} \overset{\text{iid}}{\sim}$ Exponential(1), and to implement WBB2 we set $\mathbf{u} \overset{\text{iid}}{\sim}$ Exponential(1) and $\tilde{\mathbf{u}} \overset{\text{iid}}{\sim} \delta_1$. We also benchmark BOB against MCMC and VB algorithms. More precisely, for our MCMC algorithm we consider a No-U-Turn sampler (NUTS), proposed by Hoffman and Gelman (2014). For our VB algorithm we use a mean-field automatic differentiation variational inference (ADVI) algorithm, proposed by Kucukelbir et al. (2017). Both NUTS and ADVI are implemented in Stan (Carpenter et al. 2017) as it is a highly optimized and off-the-shelf probabilistic programming language.

To initialize our algorithms, we consider a pool of candidate values and choose the initialization that yields the largest log-likelihood. Further details on our initialization strategy can be found in Appendix C. It is worth noting that we initialize all the algorithms (i.e., BOB, WBB, NUTS, and ADVI) at the same starting point.

We obtain $S = 20000$ approximate posterior draws using BOB, WBB, and ADVI, while with NUTS, we run the algorithm for $2S = 40000$ iterations and discard the first half as *burn-in*. In the case of BOB, we use batches of size $S_b = 1000$ to construct $\hat{\mathcal{L}}(x)$ and set our BO search space to be $\mathcal{X} = [10^{-5},\ 1.5]$. We run all our simulations and data analyses on an Apple Silicon-based machine with 48 GB of memory and 16 CPU cores.

4.2 Comparison metrics

To assess the accuracy of different posterior samplers, researchers have traditionally considered various distance metrics between the Bayesian posterior and its approximation. To ease computations, this assessment has usually been driven by the distance between the Bayesian marginal posteriors and the approximate marginal posteriors (see e.g., Stringer et al. (2023) or Ng and Newton (2022)). However, in the context of mixture models, these comparisons are no longer viable because of the so-called *label switching* problem (Diebolt and Robert 1994). One could use relabeling algorithms, as in Stephens (2000b), but the problem would remain as the resulting ordering of the approximate marginals might be different to the ordering of the Bayesian marginals, even after employing a relabeling algorithm, making comparisons between the marginals virtually meaningless.

We, therefore, base our comparisons on the posterior predictive distribution, as the posterior predictive is invariant to any permutation of $\boldsymbol{\theta}$ (i.e., it circumvents the label switching problem). More formally, let $p(\boldsymbol{\theta}|\mathbf{Y})$ be the Bayesian posterior and let $g(\boldsymbol{\theta}|\mathbf{Y})$ be its approximation. Then, the Bayesian posterior predictive distribution would be given by $p(\mathbf{y}_{\text{new}}|\mathbf{Y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y}_{\text{new}}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathbf{Y})\mathrm{d}\boldsymbol{\theta}$, while the approximate posterior predictive distribution would be $g(\mathbf{y}_{\text{new}}|\mathbf{Y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y}_{\text{new}}|\boldsymbol{\theta})\, g(\boldsymbol{\theta}|\mathbf{Y})\mathrm{d}\boldsymbol{\theta}$. Following Geisser (1993), under a correctly specified model and a proper prior, the Bayes risk

$$\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}) \left[\int_{\mathcal{Y}} p(\mathbf{Y}|\boldsymbol{\theta}) \mathrm{D}_{\text{KL}}(p_{\text{true}}(\mathbf{y}_{\text{new}}) \| \, g(\mathbf{y}_{\text{new}}|\mathbf{Y}))\mathrm{d}\mathbf{Y} \right] \mathrm{d}\boldsymbol{\theta}$$

is minimized when $g(\mathbf{y}_{\text{new}}|\mathbf{Y}) = p(\mathbf{y}_{\text{new}}|\mathbf{Y})$, where $\mathrm{D}_{\text{KL}}(p_{\text{true}}(\mathbf{y}_{\text{new}}) \| \, g(\mathbf{y}_{\text{new}}|\mathbf{Y}))$ denotes the KL divergence between the true data generating mechanism, $p_{\text{true}}(\mathbf{y}_{\text{new}})$, and the approximate posterior predictive distribution, $g(\mathbf{y}_{\text{new}}|\mathbf{Y})$. In other words, we minimize the above Bayes risk when our approximate posterior predictive distribution is close to the true Bayesian posterior predictive distribution, making $p(\mathbf{y}_{\text{new}}|\mathbf{Y})$ a natural benchmark when one is interested in recovering the true data generating mechanism.

Thus, we consider the sliced 2-Wasserstein (SW_2) distance between $p(\mathbf{y}_{\text{new}}|\mathbf{Y})$ and $g(\mathbf{y}_{\text{new}}|\mathbf{Y})$ as a comparison metric, where a smaller distance would suggest a better approximation of the Bayesian posterior and a better recovery of the true data generating mechanism. We make use

Table 2 Median SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, WBB2, NUTS, and ADVI, based on 30 independent replications per setting

Method	Setting								
	1	2	3	4	5	6	7	8	9
BOB	0.055	**0.058**	**0.055**	**0.067**	**0.062**	**0.067**	**0.083**	**0.071**	**0.078**
	(0.030)	(0.019)	(0.006)	(0.035)	(0.014)	(0.014)	(0.031)	(0.030)	(0.022)
WBB1	0.156	0.210	0.276	0.127	0.175	0.238	0.124	0.168	0.227
	(0.008)	(0.014)	(0.008)	(0.017)	(0.015)	(0.036)	(0.024)	(0.020)	(0.009)
WBB2	0.183	0.262	0.342	0.148	0.216	0.279	0.138	0.197	0.254
	(0.016)	(0.012)	(0.013)	(0.014)	(0.014)	(0.031)	(0.014)	(0.020)	(0.013)
NUTS	**0.039**	0.541	0.541	0.070	0.934	1.811	0.078	–	–
	(0.011)	(0.329)	(0.190)	(0.019)	(0.398)	(0.978)	(0.046)	–	–
ADVI	0.503	0.999	0.850	0.768	1.007	0.977	0.984	1.447	1.451
	(0.034)	(0.550)	(0.086)	(0.079)	(0.122)	(0.124)	(0.348)	(0.182)	(0.149)

Note: Interquartile ranges are provided in parentheses. The best performance is presented in bold. Entries denoted with "–" indicate that the corresponding method was not able to run within a computing budget of 10800 seconds

of the SW_2 distance as it is computationally efficient while being a proper distance between $p(\mathbf{y}_{\text{new}}|\mathbf{Y})$ and $g(\mathbf{y}_{\text{new}}|\mathbf{Y})$ (i.e., it satisfies symmetry, subadditivity, and the coincidence axioms (Kolouri et al. 2016)). More formally, we consider

$$\begin{aligned} &\text{SW}_2(P_{y_{\text{new}|Y}}, G_{y_{\text{new}|Y}}) \\ &= \left\{ \int_{\mathbb{S}^{d-1}} \text{W}_2^2 \left(\langle \omega, P_{y_{\text{new}|Y}} \rangle, \langle \omega, G_{y_{\text{new}|Y}} \rangle \right) \text{d}\lambda_S(\omega) \right\}^{1/2}, \end{aligned} \tag{19}$$

where $P_{y_{\text{new}|Y}} : \mathcal{B}(\mathbb{R}^d) \to [0, 1]$ and $G_{y_{\text{new}|Y}} : \mathcal{B}(\mathbb{R}^d) \to [0, 1]$ denote the corresponding probability measures of $p(\mathbf{y}_{\text{new}}|\mathbf{Y})$ and $g(\mathbf{y}_{\text{new}}|\mathbf{Y})$, respectively. Additionally, $\mathbb{S}^{d-1}$ denotes the $(d-1)$-dimensional unit hypersphere, λ_S is the uniform distribution on $\mathbb{S}^{d-1}$, and W_2 is the one-dimensional 2-Wasserstein distance (Villani 2008, Ch. 6). To compute the SW_2 distance, we make use of the function `"swdist"` from the `"T4transport"` R package (You 2023). The function takes as input 20000 draws from the Bayesian posterior predictive distribution and 20000 approximate posterior predictive draws. Details on how one can sample from the posterior predictive distributions and the target Bayesian posterior (when the true latent variables, $\mathbf{Z}$, are known), can be found in Appendix B.

4.3 Simulation results

Table 2 presents medians (and interquartile ranges) of the SW_2 distances between the target Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, WBB2, NUTS, and ADVI, based on 30 independent replications per setting. Table 3, on the other hand, presents the elapsed (wall-clock) times of each method, based on the same 30 replications. What is more, Figure 1 displays boxplots of the distributions of the SW_2 distances produced by BOB, WBB1, and WBB2. Details about the BOB-derived x^* values are presented in Appendix D.

Table 2 shows that across all settings, BOB tends to produce the best approximation of the Bayesian posterior predictive distribution. In simpler settings (e.g., setting 1), the accuracy of BOB is comparable to the accuracy of NUTS; however, as the simulation settings get more complicated, BOB becomes significantly more accurate than all the other competitors. For instance, in setting 6, the median SW_2 distance obtained via BOB is 3.55, 4.16, 27.03, and 14.58 times smaller than the median SW_2 distances obtained via WBB1, WBB2, NUTS, and ADVI, respectively. It is also clear that, across all settings, ADVI tends to produce the least accurate results.

We can also observe, in Table 3, that the fastest algorithms are WBB1 and WBB2. That being said, we can observe that the median running times of BOB are constantly around the 35-second mark. NUTS, on the other hand, results on the longest running times. For instance, in setting 6, the median running time of BOB is 43.99 seconds, while the median running time of NUTS is 3720.11 seconds. In other words, the median running time of BOB is 84.57 times smaller than the median running time of NUTS. It is worth mentioning, however, that in settings 8 and 9, NUTS was not able to run within a computing budget of 10800 seconds, while BOB run in 36.24 and 48.56 seconds, respectively, suggesting that BOB can be more than 300 times faster than NUTS. This clearly illustrates the limited scalability of MCMC algorithms to large-scale data problems.

Additionally, we can clearly observe, in Figure 1, that across all settings, the distribution of the SW_2 distances produced by BOB tends to be significantly below the distributions produced by WBB1 and WBB2, illustrating that BOB is a reliable posterior sampler. On the whole, even if WBB is the fastest algorithm, BOB constantly produces the

Table 3 Median elapsed (wall-clock) times, in seconds, for BOB, WBB1, WBB2, NUTS, and ADVI, based on 30 independent replications per setting

Method	Setting								
	1	2	3	4	5	6	7	8	9
BOB	25.612	30.218	35.648	22.988	31.737	43.994	27.705	36.238	48.568
	(0.169)	(0.506)	(1.605)	(0.379)	(0.404)	(1.248)	(1.350)	(1.242)	(2.016)
WBB1	**0.261**	**0.333**	**0.398**	**0.251**	**0.365**	**0.527**	**0.307**	**0.476**	**0.702**
	(0.008)	(0.004)	(0.010)	(0.009)	(0.008)	(0.012)	(0.007)	(0.015)	(0.025)
WBB2	0.296	0.359	0.421	0.262	0.387	0.621	0.319	0.495	0.761
	(0.006)	(0.017)	(0.014)	(0.009)	(0.013)	(0.300)	(0.007)	(0.236)	(0.295)
NUTS	46.677	602.957	1745.412	258.987	2787.522	3720.117	579.731	–	–
	(3.924)	(167.503)	(205.134)	(209.163)	(1904.040)	(1277.173)	(610.817)	–	–
ADVI	1.942	4.643	9.067	4.449	9.957	19.152	7.430	18.087	35.183
	(0.066)	(0.194)	(0.174)	(0.117)	(0.507)	(3.066)	(0.784)	(0.343)	(1.161)

Note: Interquartile ranges are provided in parentheses. The best performance is presented in bold. Entries denoted with "–" indicate that the corresponding method was not able to run within a computing budget of 10800 seconds

closest approximation to the Bayesian posterior predictive distribution in a timely fashion, nicely balancing the trade-off between computational cost and numerical accuracy.

To provide a clearer demonstration of what is happening, let us bring up an illustrative example with $n = 50$, $d = 15$, and $K = 2$. Table 4 presents the SW_2 distances between the Bayesian posterior predictive distribution and its approximations, as well as elapsed times from each method, based on our illustrative example. In particular, note that BOB is not only 62 times faster than NUTS, but it is also 10 times more accurate. What is more, Figure 2 presents KDEs of the first two dimensions of the Bayesian posterior predictive distribution and its different approximations. The Bayesian posterior predictive density (i.e., the red contour) clearly indicates the existence of two clusters. We can also observe that WBB1 and WBB2 correctly capture the locations of such clusters, but do not correctly capture the dispersion of the Bayesian posterior predictive distribution. NUTS and ADVI, on the other hand, fail to identify the two clusters in the data, providing the worst approximations of the Bayesian posterior predictive distribution. These results are not surprising as the reparameterization used by ADVI forces the posterior to be unimodal (Morningstar et al. 2021). In fact, similar results for NUTS and ADVI have also been reported in Fong et al. (2019), where the authors show that the samplers produce unimodal posteriors. Overall, it is clear that BOB produces the closest approximation to the Bayesian posterior predictive distribution. Additional posterior predictive density plots are presented in Appendix F, where we can observe similar results. The target Bayesian posterior predictive distribution is obtained following the procedure described in Appendix B.

Lastly, we would like to investigate how the sample size affects both WBB and BOB. As discussed in Section 3.2, both WBB and BOB are first order correct for the Bayesian posterior. In other words, their approximations should get better with a growing sample size. Thus, we consider an additional experiment where we fix $d = 15$ and $K = 2$, and consider various sample sizes spanning from $n = 50$ to $n = 500$. Figure 3 displays the SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, and WBB2, as a function of n. We can observe that all the distances decrease as n increases, which is expected. However, it is clear that BOB consistently outperforms WBB. Asymptotically, though, we can observe that WBB and BOB tend to converge to the same limiting distribution, which is desirable as WBB possesses appealing asymptotic properties.

Additional simulation experiments are presented in Appendix G, where we consider different initialization strategies and stronger priors on the mixture proportions, $\boldsymbol{\pi}$. Altogether, we have that BOB is a reliable method for approximate posterior sampling, which can be applied in a wide range of problems.

Table 4 SW_2 distance between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, WBB2, NUTS, and ADVI, as well as elapsed (wall-clock) times in seconds for each method, when $K = 2$, $d = 15$, and $n = 50$

Metric	Method				
	BOB	WBB1	WBB2	NUTS	ADVI
SW_2	**0.054**	0.274	0.339	0.536	0.865
Elapsed (secs)	26.875	1.098	**0.369**	1681.446	9.529

Note: The best performance is presented in bold

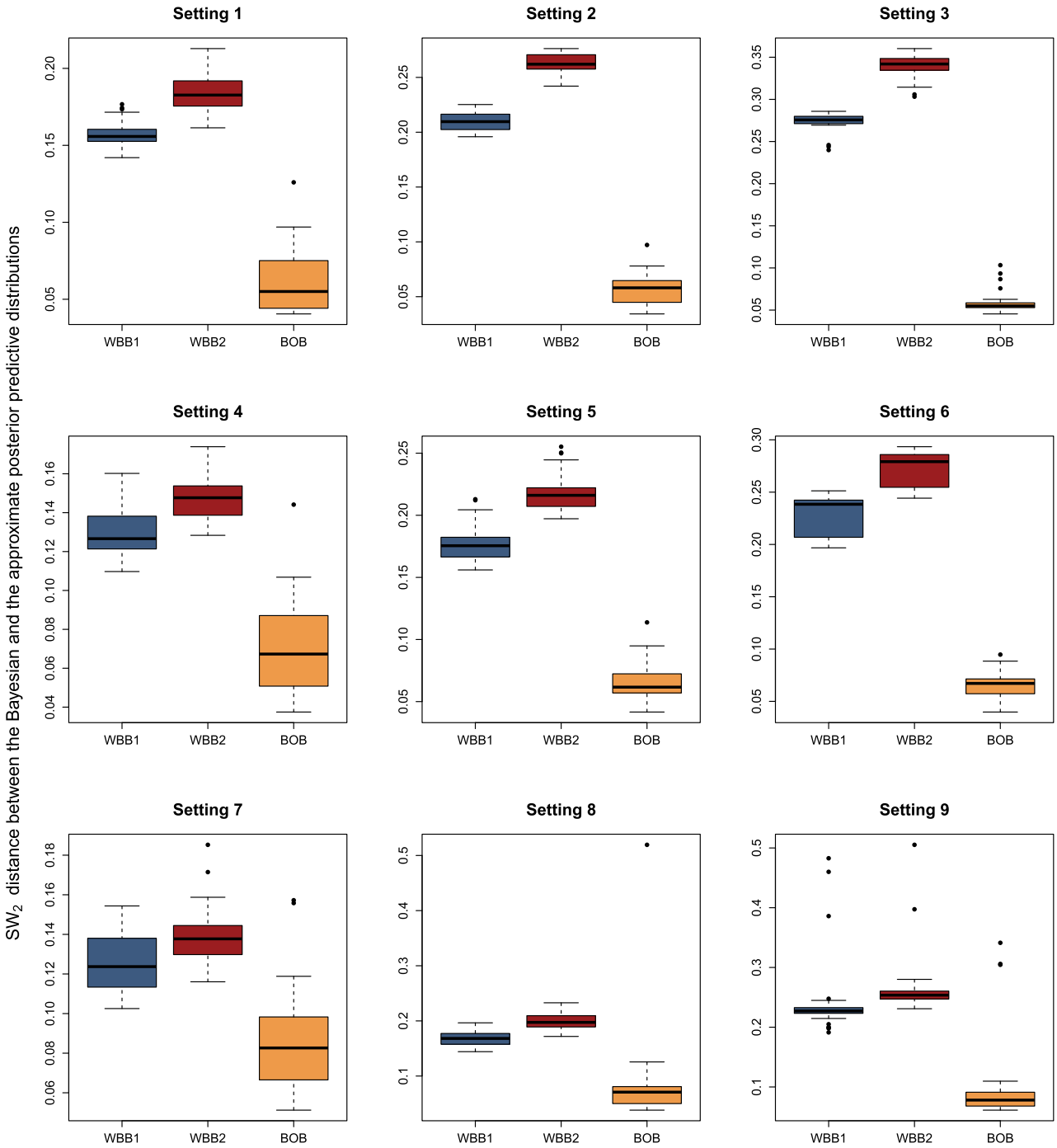

Fig. 1 Boxplots of the distributions of the SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, and WBB2, based on 30 independent replications per setting

5 Analysis of real-world data

To further demonstrate the performance and practical utility of our proposed methods, we apply them to the widely analyzed *Wine* (Aeberhard et al. 1994) and *Seeds* (Charytanowicz et al. 2010) datasets. Both datasets are publicly available from the UC Irvine Machine Learning Repository. The idea is to compare different posterior approximations under real-world data.

5.1 Wine data

The data consists of $d = 13$ observed features for 178 Italian wines, belonging to $K = 3$ different types, namely Barbera, Barolo, and Grignolino. From the Barbera type we have 48 specimens, from Barolo we have 59 specimens, and from Grignolino we have 71 specimens. A detailed description of all the features from each wine is presented in Appendix E. In all our analyses, we center the data so that each feature has mean zero. To set the prior hyper-parameters, we follow the same approach as in Section 4.1. We obtain $S = 20000$ approximate posterior draws using BOB, WBB1, WBB2, and ADVI. In the case of NUTS, we run the algorithm for 40000 iterations and discard the first half as *burn-in*. For BOB, we use batches of size $S_b = 1000$ to construct $\hat{\mathcal{L}}(x)$. As before, we set the BO search space to be $\mathcal{X} = [10^{-5}, 1.5]$.

5.2 Seeds data

Our second dataset is made up of $d = 7$ observed measurements of geometrical properties for 210 kernels belonging to $K = 3$ different varieties of seeds, namely, Kama, Rosa and Canadian. There are 70 kernels from each variety. Further details on these observed measurements are presented in Appendix E. All other configurations are set as in Section 5.1.

5.3 Results

Table 5 presents the SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, WBB2, NUTS, and ADVI, based on the wine and seeds datasets. Table 5 also presents the elapsed times from each method. The Bayesian posterior predictive distribution used as a target is computed as in Appendix B.

We can observe that BOB returns the smallest SW_2 distance under both datasets; however, the differences are much more clear in the wine data. In the seeds dataset, on the other hand, the performance of BOB is comparable to the one of WBB. These results are consistent with our previous simulation experiments, where we showed that under larger dimensions, as in the wine data, BOB tends to outperform WBB, but under smaller dimensions and larger sample sizes, both methods tend to converge to the same limiting distribution. It is also clear that NUTS and ADVI offer the worst recovery of the Bayesian posterior predictive distribution.

Note, additionally, that in the wine dataset, NUTS takes 7501 seconds to run. BOB, on the other hand, takes only 31.5 seconds. In other words, BOB is 238 times faster than NUTS. This suggests that BOB not only outperforms competing approaches in recovering the Bayesian posterior predictive distribution, but it can also be substantially faster than MCMC samplers.

To further understand the differences between BOB and WBB, Table 6 presents the optimal x^* values derived from BOB, based on the wine and seeds datasets. We can observe that under both datasets, the BOB-derived x^* values are smaller than one, suggesting that standard exponential weighting tends to underestimate the posterior distribution of the dispersion parameters. That being said, we can also observe that in the seeds dataset (i.e., when the number of unknown parameters is smaller relative to the sample size), the BOB-derived x^* value is close to one. These results are also expected as setting $x^* = 1$ results in a WBB scheme, which possess excellent asymptotic properties. Altogether, this suggests that BOB is capable of retaining the asymptotic first order correctness of WBB. In small sample sizes, however, BOB might be preferred as it offers a better recovery of the Bayesian posterior distribution.

Figures 4 and 5 present posterior predictive density plots for selected variables from the wine and seeds datasets, respectively. Additional posterior density plots are provided in Appendix F. As before, red contours represent the target Bayesian posterior predictive density, while blue KDEs represent the different approximations. To assess the validity of our model specification and model fitting, scatter plots depict the observed data. This can help us identify systematic differences between the observed data and the posterior predictive distribution. If the observed data look plausible under the posterior predictive distribution, one could say that the model specification and model fitting are reasonable (Gelman et al. 2014, Ch. 6).

In Figures 4 and 5, the Bayesian posterior predictive density clearly indicates the existence of three clusters, which align with the observed data, suggesting that the model assumptions are reasonable. We can also observe that both versions of WBB correctly capture the location of these clusters, but do not correctly capture the dispersion of the Bayesian posterior predictive (especially in Figure 4). NUTS and ADVI produce the least accurate approximations of the Bayesian posterior predictive distribution and are unable to identify the three clusters in the data. It is clear, then, that BOB produces the most accurate approximation of the Bayesian posterior predictive distribution. Note that these results are consistent with the results from Section 4. Additional posterior density plots are provided in Appendix F, where we can observe similar patterns across different variables from the wine and seeds datasets. On the whole, this illustrates BOB's practical utility in real-world problems.

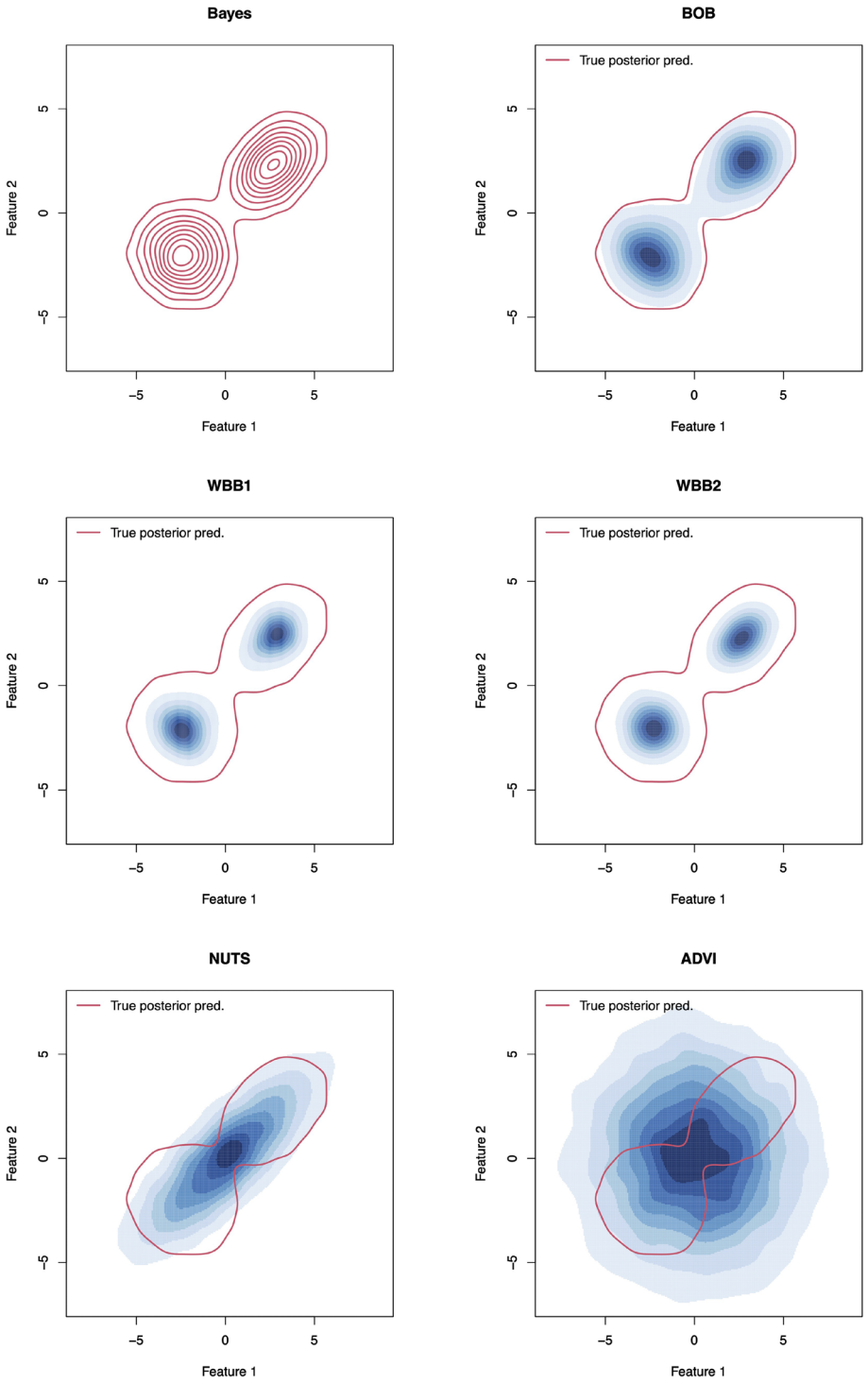

Fig. 2 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, when $K = 2$, $d = 15$, and $n = 50$

6 Discussion

In this article, we have explored alternatives to MCMC algorithms in order to sample from the posterior distribution of GMMs. More precisely, we have built upon WLB and WBB ideas, which are based on the premise that optimizing randomly weighted posterior densities can be faster and less computationally intensive than sampling from an intractable posterior. With this in mind, we have developed a randomly weighted EM algorithm, which allows us to introduce prior information into a GMM within a WBB framework. Additionally, we have also introduced BOB, a novel computational method to automatically tune the random weights from a WBB procedure by minimizing, through

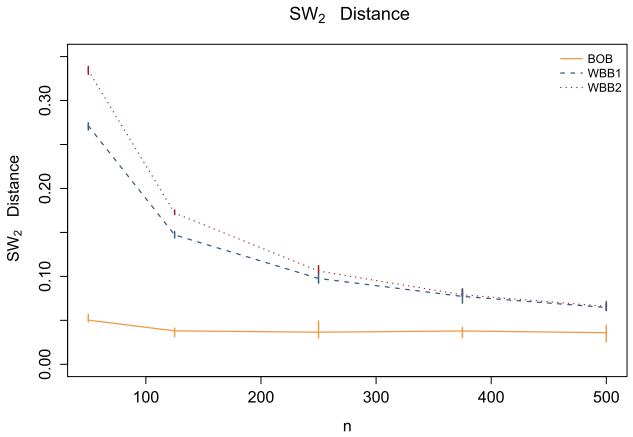

Fig. 3 SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB (golden continuous line), WBB1 (blue dashed line), and WBB2 (red dotted line), as a function of n, with $d = 15$ and $K = 2$. The curves were produced by computing the median across 30 independent replications. Error bars indicate the 25th and 75th percentiles, respectively

Bayesian optimization, a black-box and noisy version of the reverse KL divergence between the Bayesian posterior and an approximate posterior induced by random weighting. We have demonstrated that BOB consistently outperforms competing approaches in recovering the Bayesian posterior, while retaining key asymptotic properties from existing methods. Additionally, we showed that BOB can also be substantially faster than MCMC algorithms, making BOB a competitive approximate posterior sampler.

One concern, however, is with respect to the sensitivity of our algorithms to the initial parameter values. Despite the incorporation of a tempering profile, both WBB and BOB are still notably sensitive to the initial parameter values. Poorly chosen initial values might lead the algorithm to settle at a sub-optimal solution, and no amount of tempering would help us escape such a local optima (see e.g., Appendix G). Thus, a potential path for future research is the incorporation of nonconvex optimization methods (see e.g., Cong and Li (2024)) in the maximization of our randomly weighted log-posterior density. For now, a potential solution is to make use of recent developments in clustering methods to obtain adequate starting points, effectively mitigating this issue.

Table 6 BOB-derived x^* values for the wine and seeds datasets

	Dataset	
	Wine	Seeds
Optimal x^* value	0.42	0.905

What is more, since EM-type algorithms can be used to fit more general mixture models (beyond the Gaussian mixture case, see e.g., Meng and Rubin (1993) or Karlis and Xekalaki (1999)), another avenue for future research could be to extend the current state of BOB so that it can naturally handle a wider range of finite mixture models. In fact, one can also consider extending BOB to the suite of models presented in the widely-used "`mclust`" R package (Scrucca et al. 2023). One might also consider extending BOB so that it can handle more complicated prior specifications, like sparsity inducing priors on the mean vectors and the covariance matrices (e.g., *global-local* and *spike-and-slab* shrinkage priors (George and McCulloch 1993; Carvalho et al. 2010; Li et al. 2019)). These extensions could make Bayesian inference on finite mixtures more accessible for a wider range of practitioners.

Lastly, throughout this article, we have assumed that the number of components in the mixture, K, is fixed and known. Thus, one additional avenue for future research could be to address the problem of selecting K within a Bayesian paradigm (see e.g. Richardson and Green (1997), Stephens (2000a), Nobile and Fearnside (2007), or Baudry et al. (2010)). A possible solution could be to assume an overfitted mixture and specify a hierarchical prior on the Dirichlet parameters, $\{a_k\}_{k=1}^K$, in the spirit of Malsiner-Walli et al. (2016). This issue, however, is outside the scope of this article.

On the whole, BOB joins a growing body of literature, which aims to draw practitioner's attention toward posterior samplers beyond traditional MCMC algorithms (Nie and Ročková 2023b; Newton et al. 2021). Thus, extending BOB (and BOB-like methodologies) from GMMs with conjugate priors to arbitrary posterior distributions presents an exciting research opportunity.

Table 5 SW_2 distances between the Bayesian posterior predictive distribution and its approximations obtained via BOB, WBB1, WBB2, NUTS, and ADVI, based on the wine and seeds datasets. The elapsed (wall-clock) times in seconds of each method are also displayed

Dataset	Metric	Method				
		BOB	WBB1	WBB2	NUTS	ADVI
Wine	SW_2	**3.201**	4.694	5.957	16.65	18.906
	Elapsed (secs)	31.509	1.283	**0.571**	7501.166	21.307
Seeds	SW_2	**0.07**	0.071	0.071	0.357	0.228
	Elapsed (secs)	24.681	**0.377**	0.505	2224.107	19.204

Note: The best performance is presented in bold

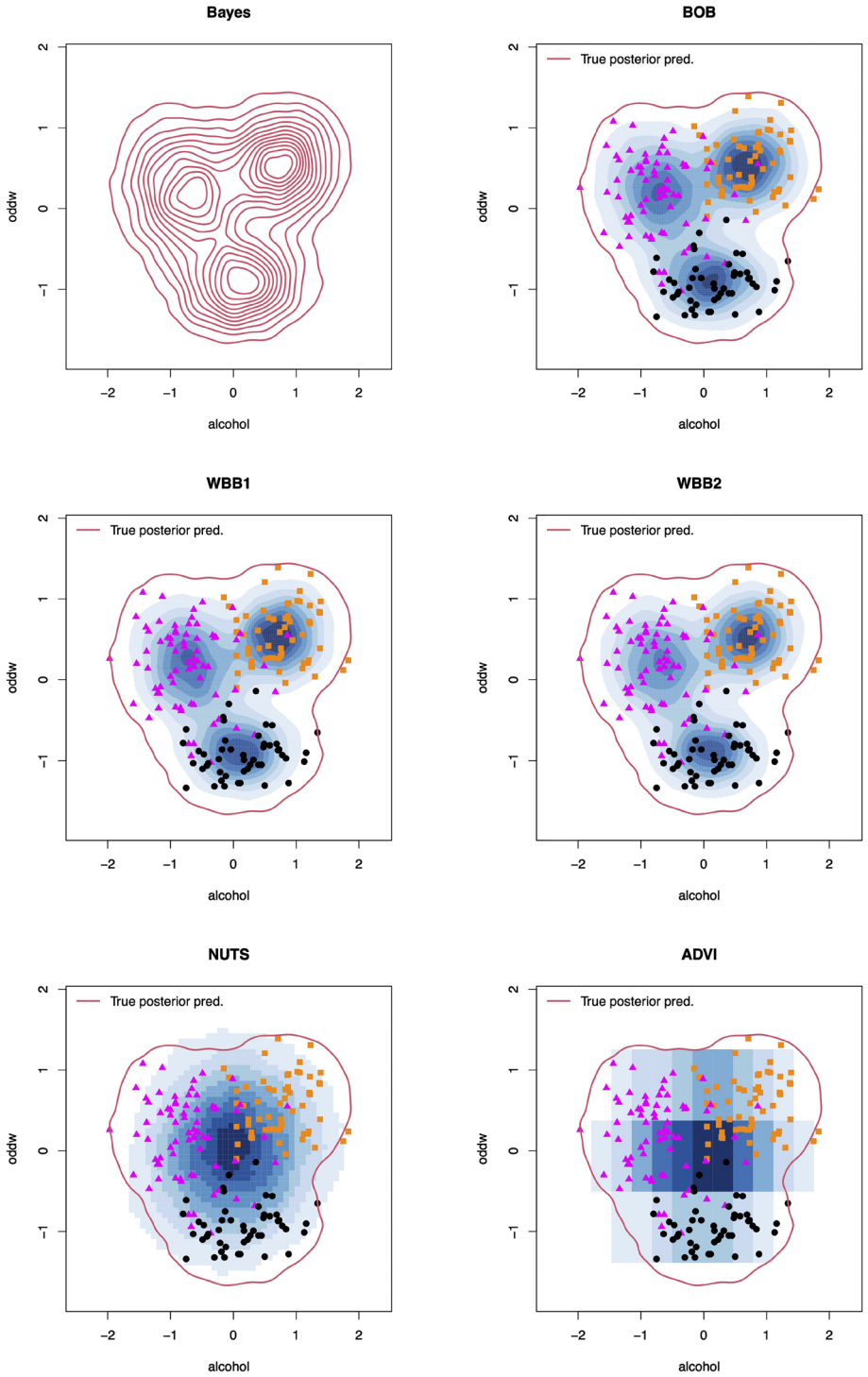

Fig. 4 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for selected variables from the wine data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Barolo, Grignolino, and Barbera wines, respectively

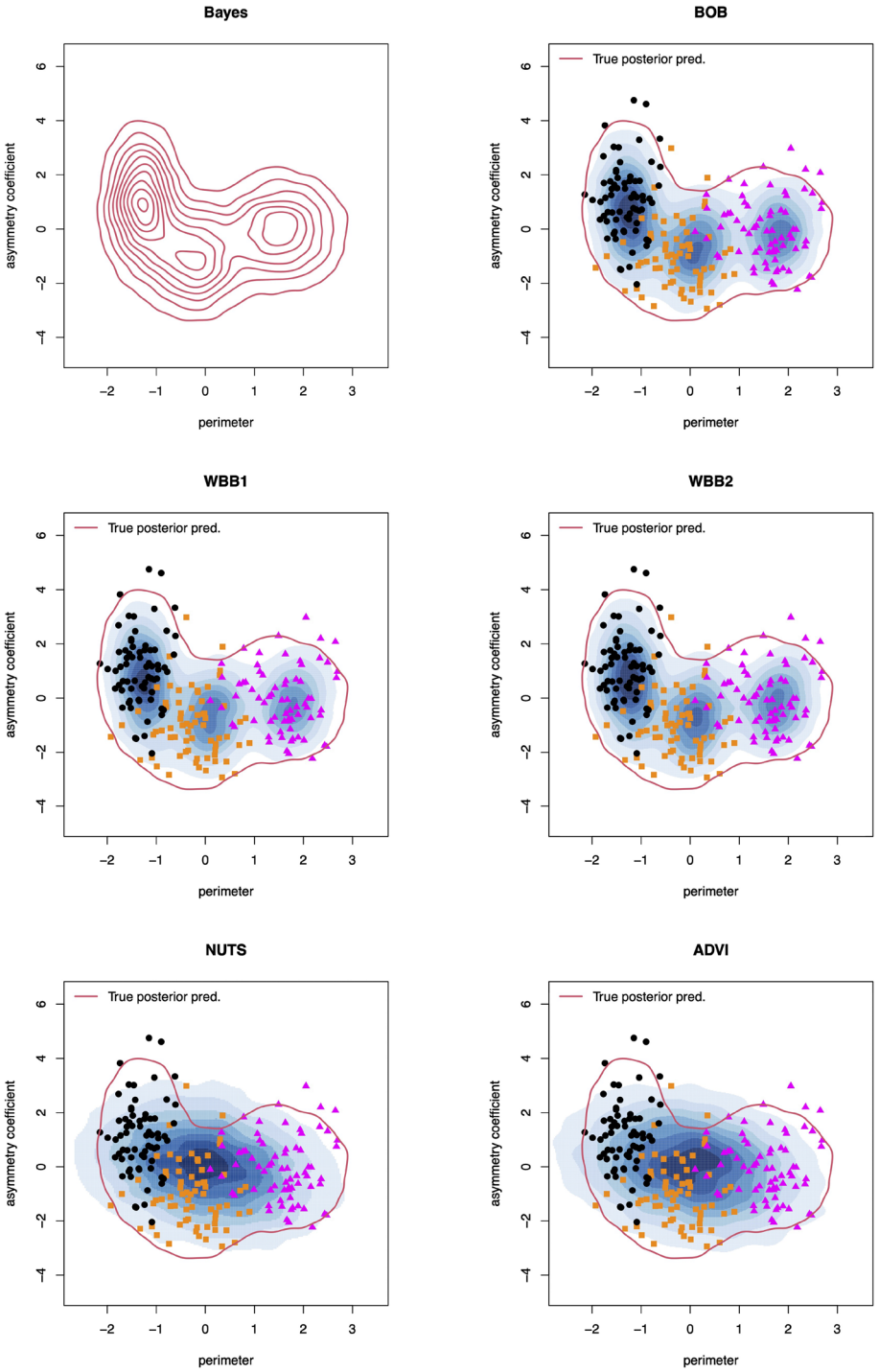

Fig. 5 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for selected variables from the seeds data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Kama, Rosa, and Canadian wheat kernels, respectively

Appendix A Derivations and proofs

A.1 Proof of Proposition 1

Recall from (4) that the weighted log-likelihood function of the observed data is given by

$$\begin{aligned}\log p_u(\mathbf{Y}|\boldsymbol{\theta}) &= \sum_{i=1}^{n} u_i \left\{ \log \sum_{k=1}^{K} \pi_k \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right\} \\ &= \sum_{i=1}^{n} u_i \log p(\mathbf{y}_i|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} u_i \left\{\log p(\mathbf{y}_i, \mathbf{z}_i|\boldsymbol{\theta}) - \log p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta})\right\} \\ &= \sum_{i=1}^{n} u_i \left\{ \sum_{k=1}^{K} z_{ik} \left[\log \pi_k + \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right] \right. \\ &\quad \left. - \sum_{k=1}^{K} z_{ik} \log \pi_k \right\}.\end{aligned}$$

Then, the weighted log-posterior can be written as

$$\begin{aligned}\log p_u(\boldsymbol{\theta}|\mathbf{Y}) &\propto \log p_u(\mathbf{Y}|\boldsymbol{\theta}) + \log p_{\tilde{u}}(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} u_i \left\{ \sum_{k=1}^{K} z_{ik} \left[\log \pi_k + \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right] \right. \\ &\quad \left. - \sum_{k=1}^{K} z_{ik} \log \pi_k \right\} + \log p_{\tilde{u}}(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} u_i \sum_{k=1}^{K} z_{ik} \left[\log \pi_k + \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right] \\ &\quad + \log p_{\tilde{u}}(\boldsymbol{\theta}) - \sum_{i=1}^{n} u_i \sum_{k=1}^{K} z_{ik} \log \pi_k.\end{aligned}$$

We now take the expectation over the latent variables, conditional on $\mathbf{Y}, \mathbf{u}$, and $\boldsymbol{\theta}^{(t)}$. Recall from (7) that $\mathbb{E}[z_{ik}|\boldsymbol{\theta}^{(t)}, \mathbf{Y}, \mathbf{u}] = q_{ik}$; thus, we get

$$\begin{aligned}\log p_u(\boldsymbol{\theta}|\mathbf{Y}) &\propto \sum_{i=1}^{n} u_i \underbrace{\sum_{k=1}^{K} q_{ik} \left[\log \pi_k + \log \varphi\left(\mathbf{y}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right]}_{Q^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})} \\ &\quad + \log p_{\tilde{u}}(\boldsymbol{\theta}) - \sum_{i=1}^{n} u_i \underbrace{\sum_{k=1}^{K} q_{ik} \log \pi_k}_{H^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})} \\ &= \sum_{i=1}^{n} u_i Q^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \log p_{\tilde{u}}(\boldsymbol{\theta}) \\ &\quad - \sum_{i=1}^{n} u_i H^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}).\end{aligned}$$

In the above equation, the left-hand side remains unchanged as it is the expectation of a constant. Then, subtracting the weighted log-posterior evaluated at $\boldsymbol{\theta}^{(t)}$ gives

$$\begin{aligned}&\log p_u(\boldsymbol{\theta}|\mathbf{Y}) - \log p_u(\boldsymbol{\theta}^{(t)}|\mathbf{Y}) \\ &= \sum_{i=1}^{n} u_i \left\{ Q^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q^i(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \right\} \\ &\quad + \left\{ \log p_{\tilde{u}}(\boldsymbol{\theta}) - \log p_{\tilde{u}}(\boldsymbol{\theta}^{(t)}) \right\} \\ &\quad + \sum_{i=1}^{n} u_i \left\{ H^i(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) - H^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\}.\end{aligned}$$

From Gibbs' inequality, we have that $H^i(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \geq H^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. Therefore, since all the weights are nonnegative, we can conclude that

$$\begin{aligned}&\log p_u(\boldsymbol{\theta}|\mathbf{Y}) - \log p_u(\boldsymbol{\theta}^{(t)}|\mathbf{Y}) \\ &\geq \sum_{i=1}^{n} u_i \left\{ Q^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q^i(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \right\} \\ &\quad + \left\{ \log p_{\tilde{u}}(\boldsymbol{\theta}) - \log p_{\tilde{u}}(\boldsymbol{\theta}^{(t)}) \right\}.\end{aligned}$$

Moreover, note that

$$\begin{aligned}\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\mathbf{Z}|\mathbf{Y},\mathbf{u},\boldsymbol{\theta}^{(t)}} \left[\log p_u(\mathbf{Y}, \mathbf{Z}|\boldsymbol{\theta}) + \log p_{\tilde{u}}(\boldsymbol{\theta})\right] \\ &= \sum_{i=1}^{n} u_i Q^i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \log p_{\tilde{u}}(\boldsymbol{\theta}),\end{aligned}$$

which is our surrogate objective function defined in (8). In other words, by maximizing $\tilde{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ with respect to $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, we cannot decrease the weighted log-posterior. Hence, the proof is complete.

A.2 Proof of Proposition 2

From (8), we have that the update of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ is given by

$$\left(\boldsymbol{\mu}_k^{(t+1)}, \boldsymbol{\Sigma}_k^{(t+1)}\right) = \underset{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}{\arg\max} \left\{ h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\},$$

with

$$\begin{aligned}h(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{1}{2} \sum_{i=1}^{n} \left(u_i q_{ik} \left[\log |\boldsymbol{\Sigma}_k| \right.\right. \\ \left.\left. + \left(\mathbf{y}_i - \boldsymbol{\mu}_k\right)' \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{y}_i - \boldsymbol{\mu}_k\right)\right]\right) \\ - \frac{\tilde{u}_{\Sigma_k} \operatorname{tr}(\boldsymbol{\Psi}_k \boldsymbol{\Sigma}_k^{-1})}{2} - \frac{\tilde{u}_{\mu_k} \lambda_k}{2} (\boldsymbol{\mu}_k\end{aligned}$$

$$
\begin{aligned}
&- \boldsymbol{\beta}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\beta}_k) - \tilde{u}_{\Sigma_k} \left(\frac{\nu_k + d}{2} + 1 \right) \log |\boldsymbol{\Sigma}_k| \\
&\overset{(*)}{=} -\frac{\tilde{n}_k}{2} \log |\boldsymbol{\Sigma}_k| \\
&- \frac{1}{2} \operatorname{tr} \left[\left(\tilde{n}_k (\tilde{\bar{\mathbf{y}}}_k - \boldsymbol{\mu}_k)(\tilde{\bar{\mathbf{y}}}_k - \boldsymbol{\mu}_k)' + \tilde{\mathbf{S}}_k \right) \boldsymbol{\Sigma}_k^{-1} \right] \\
&- \left(\frac{\tilde{\nu}_k + d}{2} + 1 \right) \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} \operatorname{tr} \left(\tilde{\boldsymbol{\Psi}}_k \boldsymbol{\Sigma}_k^{-1} \right) \\
&- \frac{\tilde{\lambda}_k}{2} \operatorname{tr} \left[(\boldsymbol{\mu}_k - \boldsymbol{\beta}_k)(\boldsymbol{\mu}_k - \boldsymbol{\beta}_k)' \boldsymbol{\Sigma}_k^{-1} \right] \\
&= -\left(\frac{\bar{\nu}_k + d}{2} + 1 \right) \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} \operatorname{tr} \left(\bar{\boldsymbol{\Psi}}_k \boldsymbol{\Sigma}_k^{-1} \right) \\
&- \frac{\bar{\lambda}_k}{2} (\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu}_k - \bar{\boldsymbol{\beta}}_k),
\end{aligned}
$$

where $(*)$ follows from the fact that $\sum_{i=1}^{n} \mathbf{b}_i' \mathbf{A} \mathbf{b}_i = \operatorname{tr}(\mathbf{B}' \mathbf{B} \mathbf{A})$, where $\mathbf{B} \in \mathbb{R}^{n \times d}$ denotes the matrix whose i-th row is $\mathbf{b}_i' \in \mathbb{R}^{1 \times d}$, and by writing $(\mathbf{y}_i - \boldsymbol{\mu}_k)$ as $(\mathbf{y}_i - \tilde{\bar{\mathbf{y}}}_k + \tilde{\bar{\mathbf{y}}}_k - \boldsymbol{\mu}_k)$. This completes the proof.

Appendix B Posterior and posterior predictive samplers

B.1 Sampling from the posterior predictive distribution

Given posterior draws, $\{\boldsymbol{\theta}_{(s)}\}_{s=1}^{S}$, one can easily obtain posterior predictive draws as described in algorithm 2.

Algorithm 2 Posterior Predictive Sampler

Input:
Posterior draws: $\{\boldsymbol{\theta}_{(s)}\}_{s=1}^{S}$
Model: $p(\mathbf{y}|\boldsymbol{\theta})$
Output:
Posterior predictive draws: $\{\mathbf{y}_{\text{new},(s)}\}_{s=1}^{S}$
1: **for** $s \in \{1, \ldots, S\}$ **do** ▷ in parallel
2: Sample $\mathbf{y}_{\text{new},(s)} \sim p(\mathbf{y}|\boldsymbol{\theta}_{(s)})$
3: **end for**

B.2 Sampling from the target Bayesian posterior

Following Gelman et al. (2014), under the likelihood and priors specified in Section 2.1, the Bayesian posterior distributions of $\{\boldsymbol{\mu}_k|\mathbf{Y}, \mathbf{Z}\}$, $\{\boldsymbol{\Sigma}_k|\mathbf{Y}, \mathbf{Z}\}$, and $\{\boldsymbol{\pi}|\mathbf{Y}, \mathbf{Z}\}$ are given by

$$
\begin{aligned}
&\boldsymbol{\mu}_k|\mathbf{Y}, \mathbf{Z} \sim t_{(\hat{\nu}_k - d + 1)} \left(\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\Psi}}_k / (\hat{\lambda}_k (\hat{\nu}_k - d + 1)) \right), \\
&\boldsymbol{\Sigma}_k|\mathbf{Y}, \mathbf{Z} \sim \text{inverse-Wishart} \left(\hat{\nu}_k, \hat{\boldsymbol{\Psi}}_k^{-1} \right), \\
&\boldsymbol{\pi}|\mathbf{Y}, \mathbf{Z} \sim \text{Dirichlet} \left(\hat{a}_1, \ldots, \hat{a}_K \right),
\end{aligned}
$$

where $\hat{\boldsymbol{\beta}}_k = \lambda_k / (\lambda_k + n_{z_k}) \boldsymbol{\beta}_k + n_{z_k} / (\lambda_k + n_{z_k}) \bar{\mathbf{y}}_{z_k}$, $\hat{\boldsymbol{\Psi}}_k = \boldsymbol{\Psi}_k + \mathbf{S}_{z_k} + (\lambda_k n_{z_k}) / (\lambda_k + n_{z_k}) (\bar{\mathbf{y}}_{z_k} - \boldsymbol{\beta}_k)(\bar{\mathbf{y}}_{z_k} - \boldsymbol{\beta}_k)'$, $\hat{\lambda}_k = \lambda_k + n_{z_k}$, $\hat{\nu}_k = \nu_k + n_{z_k}$, and $\hat{a}_k = a_k + n_{z_k}$, with $n_{z_k} = \sum_{i=1}^{n} z_{ik}$, $\bar{\mathbf{y}}_{z_k} = (n_{z_k})^{-1} \sum_{i:z_{ik}=1} \mathbf{y}_i$, and $\mathbf{S}_{z_k} = \sum_{i:z_{ik}=1} (\mathbf{y}_i - \bar{\mathbf{y}}_{z_k})(\mathbf{y}_i - \bar{\mathbf{y}}_{z_k})'$. When the true latent indicator variables, $\mathbf{Z}$, are known (e.g., under simulated data or when we know the true cluster labels), one can easily obtain a sequence of random draws from the target Bayesian posterior distribution, $\{\boldsymbol{\theta}_{\text{Bayes},(s)}\}_{s=1}^{S}$, as described in algorithm 3. With the posterior draws, $\{\boldsymbol{\theta}_{\text{Bayes},(s)}\}_{s=1}^{S}$, one can then proceed to obtain random draws from the Bayesian posterior predictive distribution, as described in algorithm 2.

Algorithm 3 Bayesian Posterior Sampler

Input:
Data: $\mathbf{Y}$
True latent indicator variables: $\mathbf{Z}$
Total number of posterior draws: S
Posterior hyper-parameters: $\{\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\Psi}}_k, \hat{\lambda}_k, \hat{\nu}_k, \hat{a}_k\}_{k=1}^{K}$
Output:
Bayesian posterior draws: $\{\boldsymbol{\theta}_{\text{Bayes},(s)}\}_{s=1}^{S}$
1: **for** $s \in \{1, \ldots, S\}$ **do** ▷ in parallel
2: Sample $\boldsymbol{\Sigma}_{k,(s)}|\mathbf{Y}, \mathbf{Z} \sim \text{inverse-Wishart}(\hat{\nu}_k, \hat{\boldsymbol{\Psi}}_k^{-1})$, $\forall k \in [K]$
3: Sample $\boldsymbol{\mu}_{k,(s)}|\mathbf{Y}, \mathbf{Z}, \boldsymbol{\Sigma}_{k,(s)} \sim N_d(\hat{\boldsymbol{\beta}}_k, \boldsymbol{\Sigma}_{k,(s)}/\hat{\lambda}_k)$, $\forall k \in [K]$
4: Sample $\boldsymbol{\pi}_{(s)}|\mathbf{Y}, \mathbf{Z} \sim \text{Dirichlet}(\hat{a}_1, \ldots, \hat{a}_K)$
5: Set $\boldsymbol{\theta}_{\text{Bayes},(s)} \leftarrow \{\pi_{k,(s)}, \boldsymbol{\mu}_{k,(s)}, \boldsymbol{\Sigma}_{k,(s)}\}_{k=1}^{K}$
6: **end for**

Appendix C Warm initialization strategy

To initialize our algorithms, we consider a pool of candidate initial values. These candidate values are obtained via (1) hard-thresholded K-means (Raymaekers and Zamar 2022), where the HTK-means penalty term is selected using AIC, BIC, and a regularization path plot; (2) sparse K-means (Witten and Tibshirani 2010), where the sparse K-means shrinkage parameter is selected using a permutation approach; (3) model-based clustering using the "`mclust`" R package (Scrucca et al. 2023), which itself is initialized by hierarchical model-based agglomerative clustering; and (4) K-means clustering (MacQueen 1967). Then, we choose the point that yields the largest log-likelihood as our warm initialization. To facilitate comparisons, we initialize all algorithms at this starting point.

Appendix D BOB-derived x^* values

Table 7 presents medians (and interquartile ranges) of the BOB-derived x^* values for all the nine settings considered

Table 7 Median x^* values obtained via BOB, based on 30 independent replications per setting

	Setting								
	1	2	3	4	5	6	7	8	9
Optimal x^*	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)	10^{-5} (0.0)

Note: Interquartile ranges are provided in parentheses

in Section 4. It is evident that the Bayesian optimization procedure is constantly choosing the optimal x^* values at the lower boundary of the search space, $\mathcal{X}$. This suggests that in complex settings, where the number of unknown parameters is large and the sample size is small, standard exponential weighting severely underestimates the posterior distribution of dispersion parameters. These results are consistent with the ones presented in Figures 1, 2, 6, and 7, where we can clearly observe that standard exponential weighting does not capture the dispersion of the target Bayesian posterior predictive distribution. That being said, in cases where the sample size is larger (as in the seeds data), BOB is also capable of deriving optimal x^* values close to one, retaining the first order correctness of standard exponential weighting (see e.g., Table 6).

Appendix E Features in benchmark data

Tables 8 and 9 present details and descriptions of the variables in the Wine and Seeds datasets, respectively.

Appendix F Additional density plots

F.1 Additional bivariate posterior predictive density plots

Figures 6 and 7 present additional bivariate posterior predictive density plots from our illustrative example (in which $n = 50$, $d = 15$, and $K = 2$). We can observe similar patterns as in the main article. More precisely, both versions of WBB fail to capture the dispersion of the Bayesian posterior predictive distribution, while NUTS and ADVI produce unimodal densities, failing to identify the clusters in the data. It is clear, then, that BOB produces the best approximation.

Similarly, Figures 8, 9, 10, and 11 present bivariate posterior predictive density plots for additional variables from the wine and seeds datasets. As in the main document, we can observe that both versions of WBB correctly capture the location of the three clusters; however, they tend to produce overconfident posterior predictive distributions. NUTS and ADVI are unable to identify the three clusters in the data. It is clear, then, that BOB produces the best approximation of the Bayesian posterior predictive distribution.

Table 8 Features in the Wine Data

Wine Feature		Wine Feature	
Abbreviation	Description	Abbreviation	Description
`alco`	Alcohol percentage	`nflav`	Nonflavanoid phenols
`mal`	Malic acid	`proa`	Proanthocyanins
`ash`	Ash	`col`	Color intensity
`akash`	Alkalinity of ash	`hue`	Hue
`mag`	Magnesium	`ODdw`	$\frac{OD_{280}}{OD_{315}}$ of diluted wines
`phen`	Total phenols	`prol`	Proline
`flav`	Flavanoids		

Table 9 Features in the Seeds Data

Seed Feature		Seed Feature	
Abbreviation	Description	Abbreviation	Description
`area`	Area of kernel	`k.width`	Width of kernel
`perimeter`	Perimeter of kernel	`asymmetry`	Asymmetry coefficient
`compactness`	Compactness of kernel	`k.gr.len`	Length of kernel groove
`k.len`	Length of kernel		

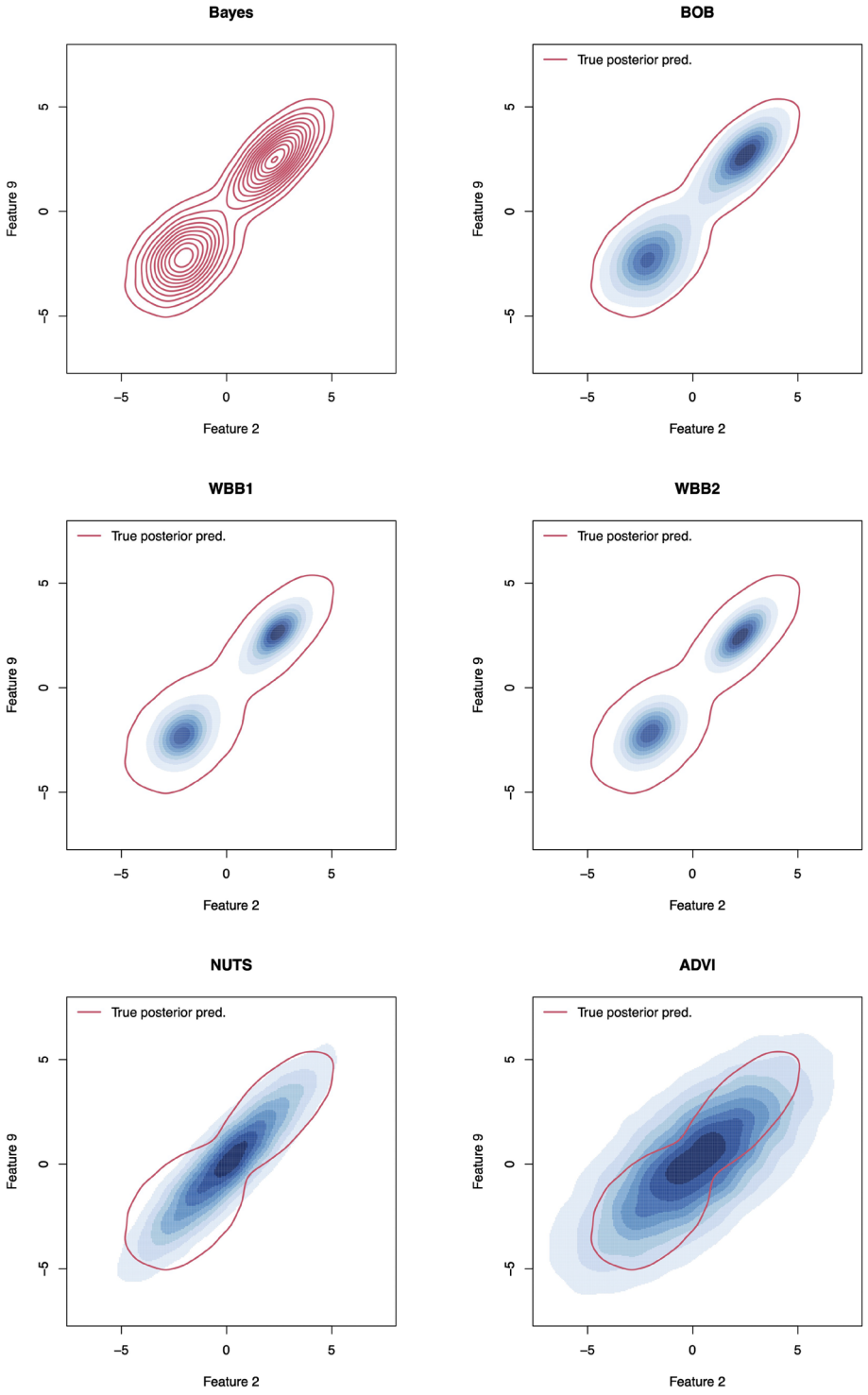

Fig. 6 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, when $K = 2$, $d = 15$, and $n = 50$

F.2 Additional univariate posterior predictive density plots

What is more, Figures 12, 13, and 14 present univariate density plots of each marginal entry from the approximate posterior predictive distributions obtained with BOB, WBB1, WBB2, NUTS, and ADVI, as well as the target Bayesian posterior predictive distribution. To be precise, Figure 12 is based on our illustrative example (in which $n = 50$, $d = 15$, and $K = 2$). Figures 13 and 14 are based on the wine and seeds datasets, respectively. As before, we can observe that BOB tends to produce the closest approximation of the Bayesian posterior predictive distribution.

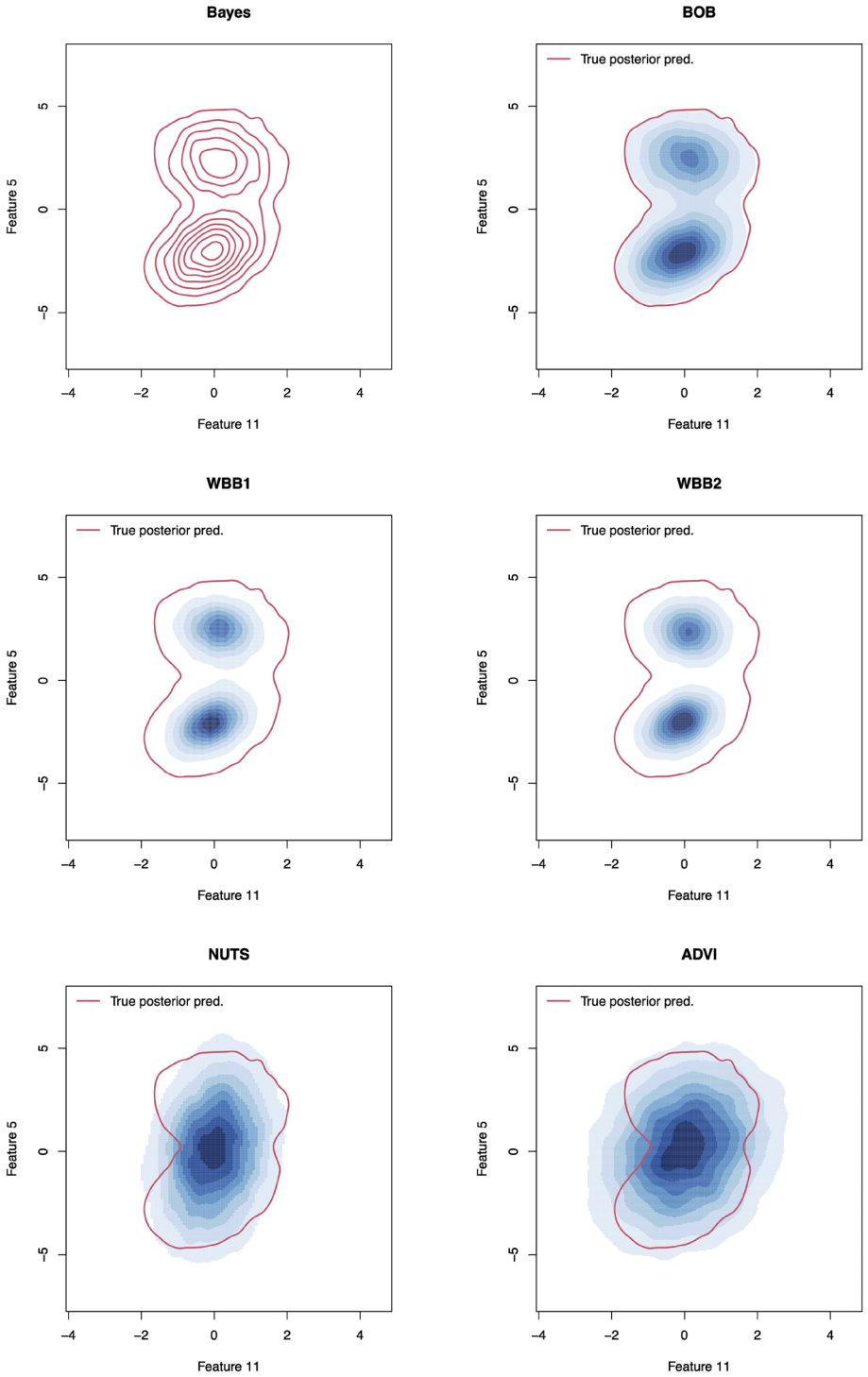

Fig. 7 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, when $K = 2$, $d = 15$, and $n = 50$

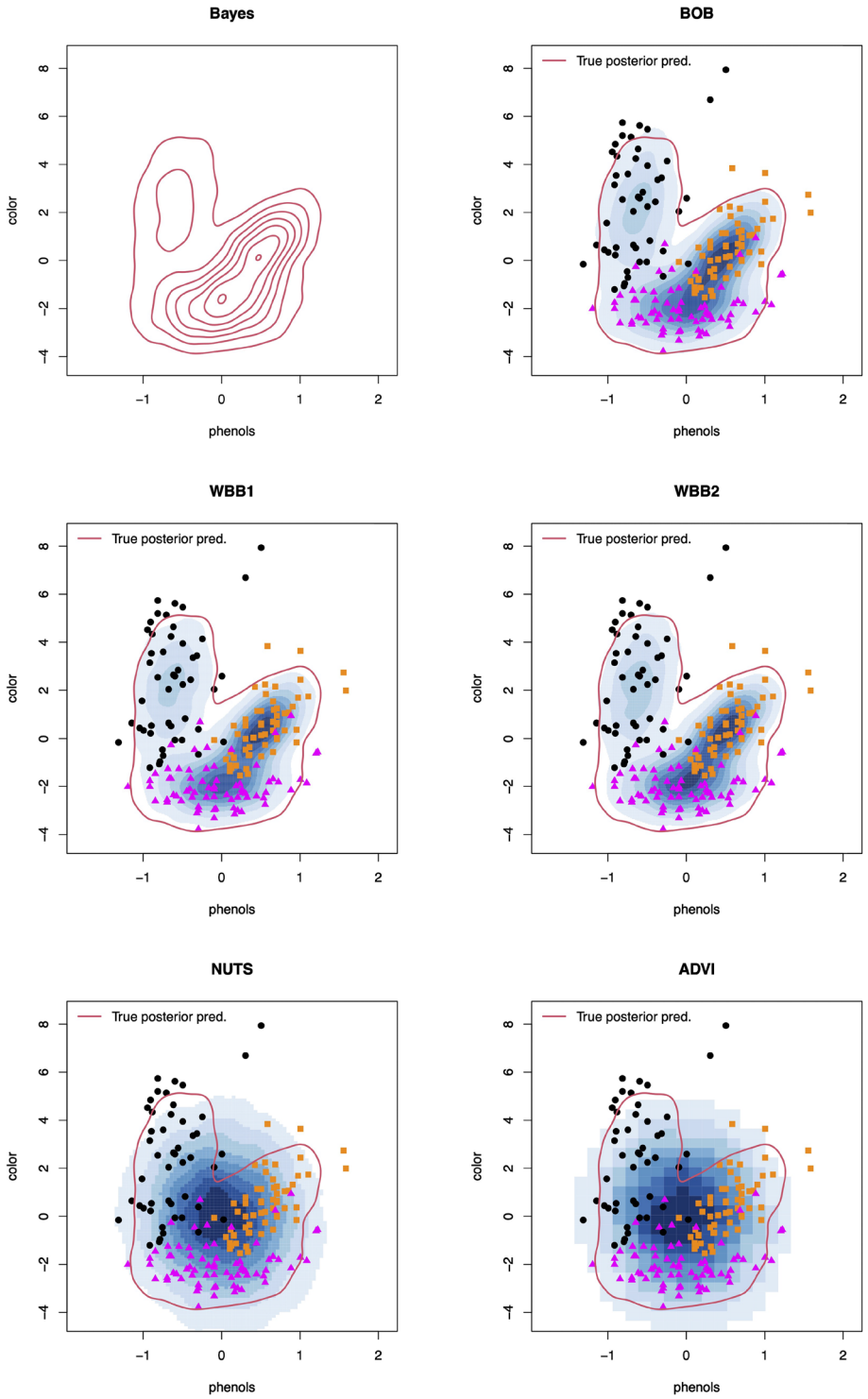

Fig. 8 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for additional variables from the wine data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Barolo, Grignolino, and Barbera wines, respectively

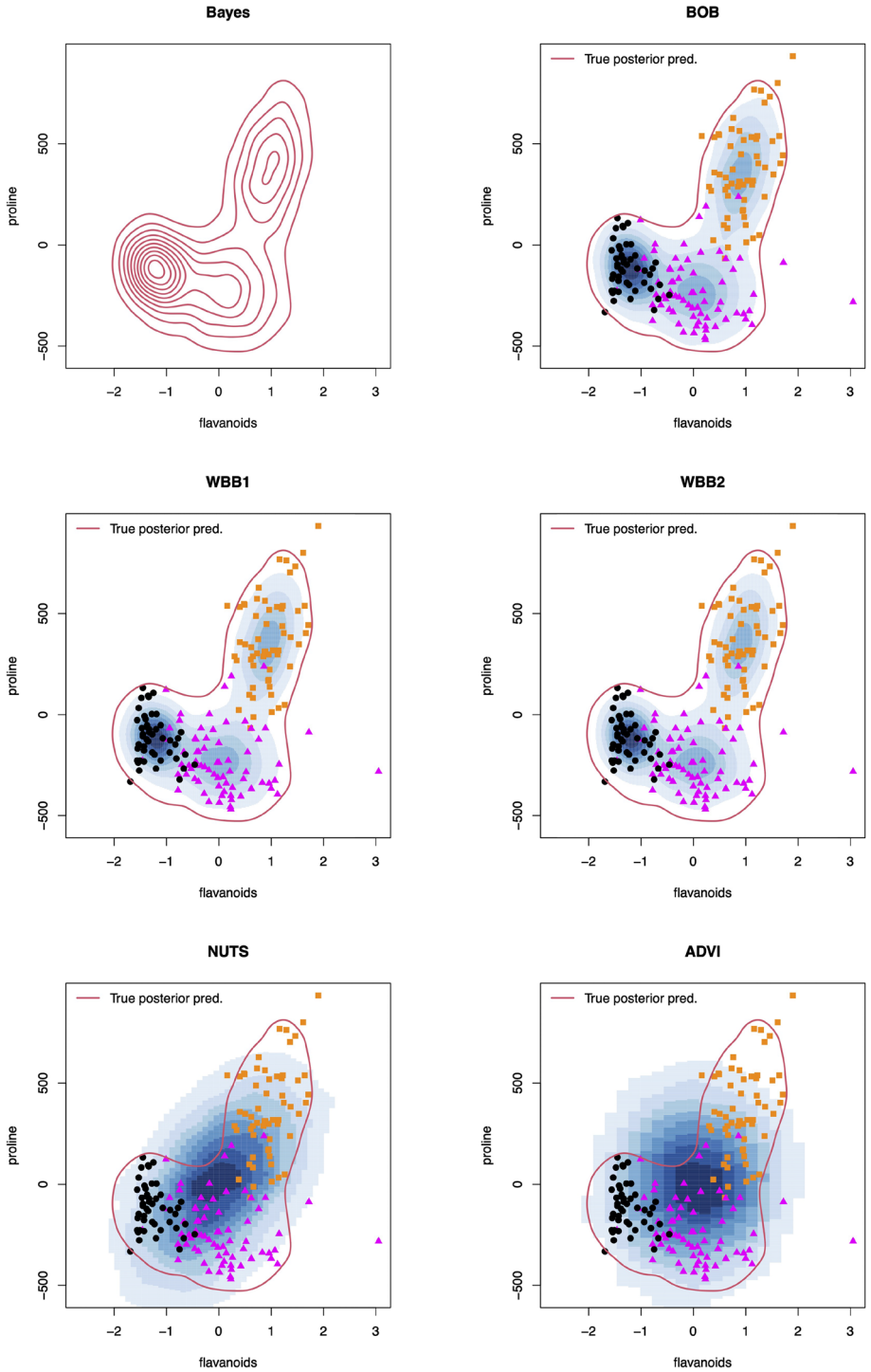

Fig. 9 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for additional variables from the wine data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Barolo, Grignolino, and Barbera wines, respectively

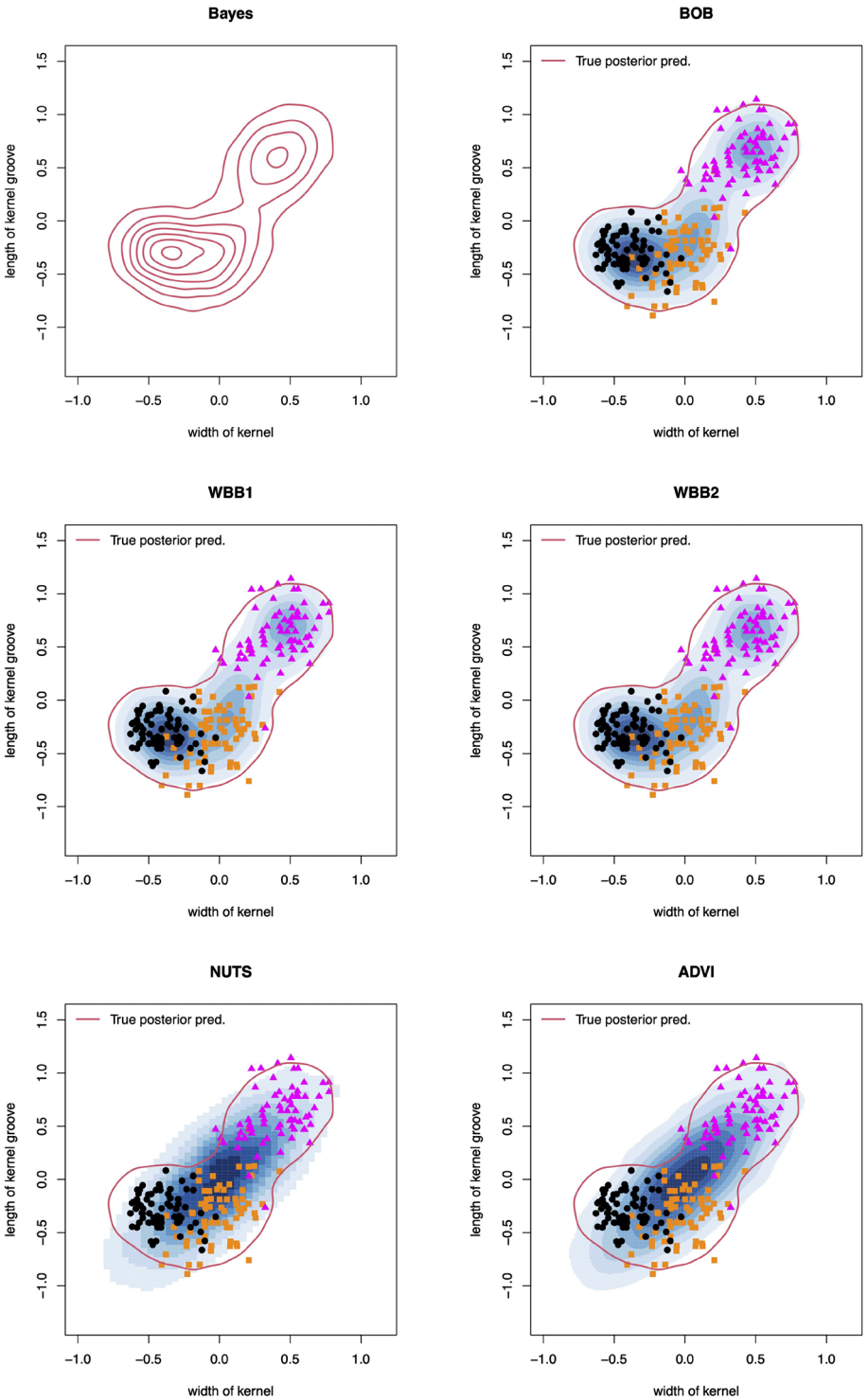

Fig. 10 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for additional variables from the seeds data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Kama, Rosa, and Canadian wheat kernels, respectively

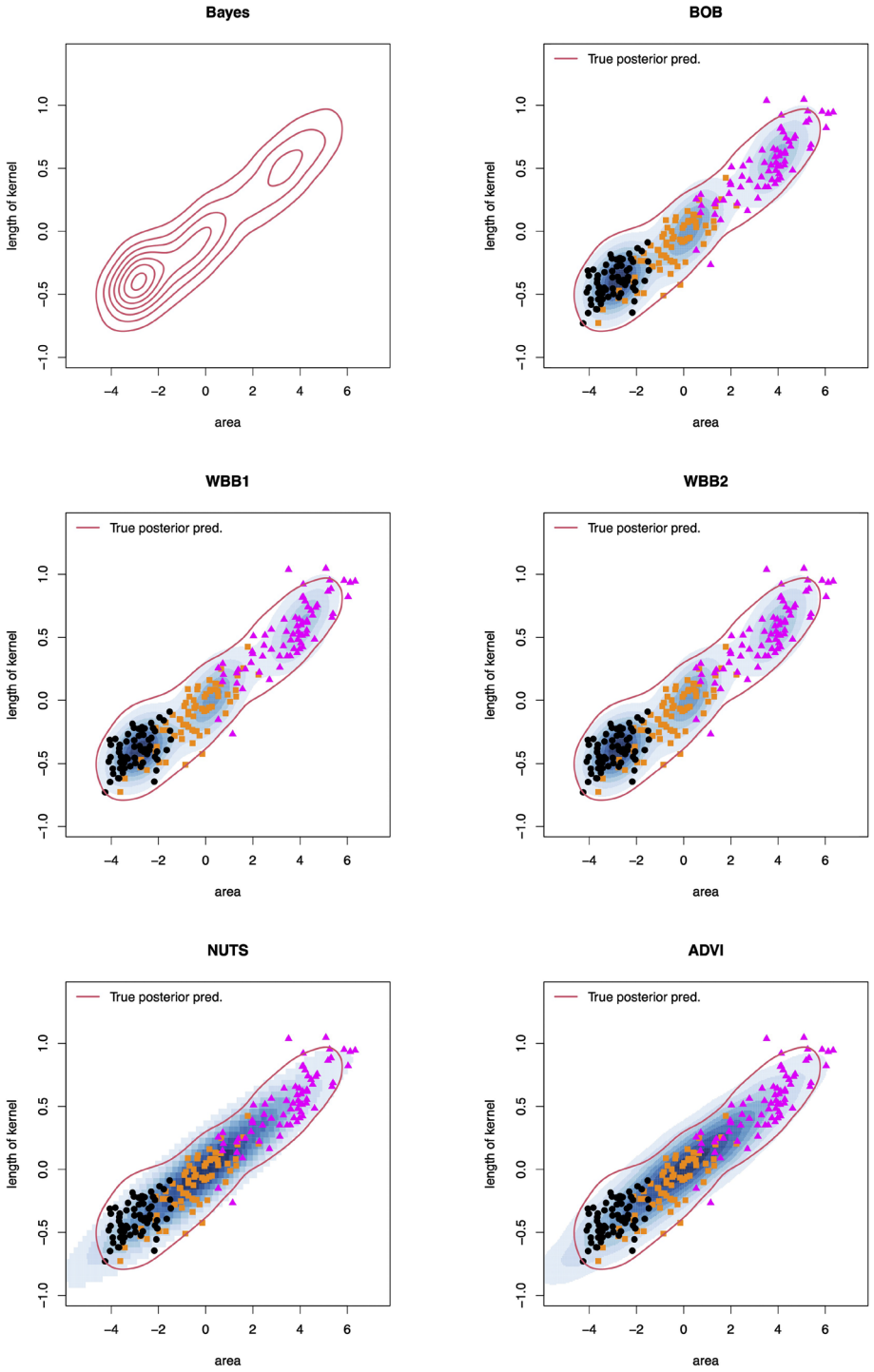

Fig. 11 True Bayesian posterior predictive density (red contours) and its approximations (blue KDEs) obtained via BOB, WBB1, WBB2, NUTS, and ADVI, for additional variables from the seeds data. Scatter plots depict the observed data. Orange squares, pink triangles and dark circles represent Kama, Rosa, and Canadian wheat kernels, respectively

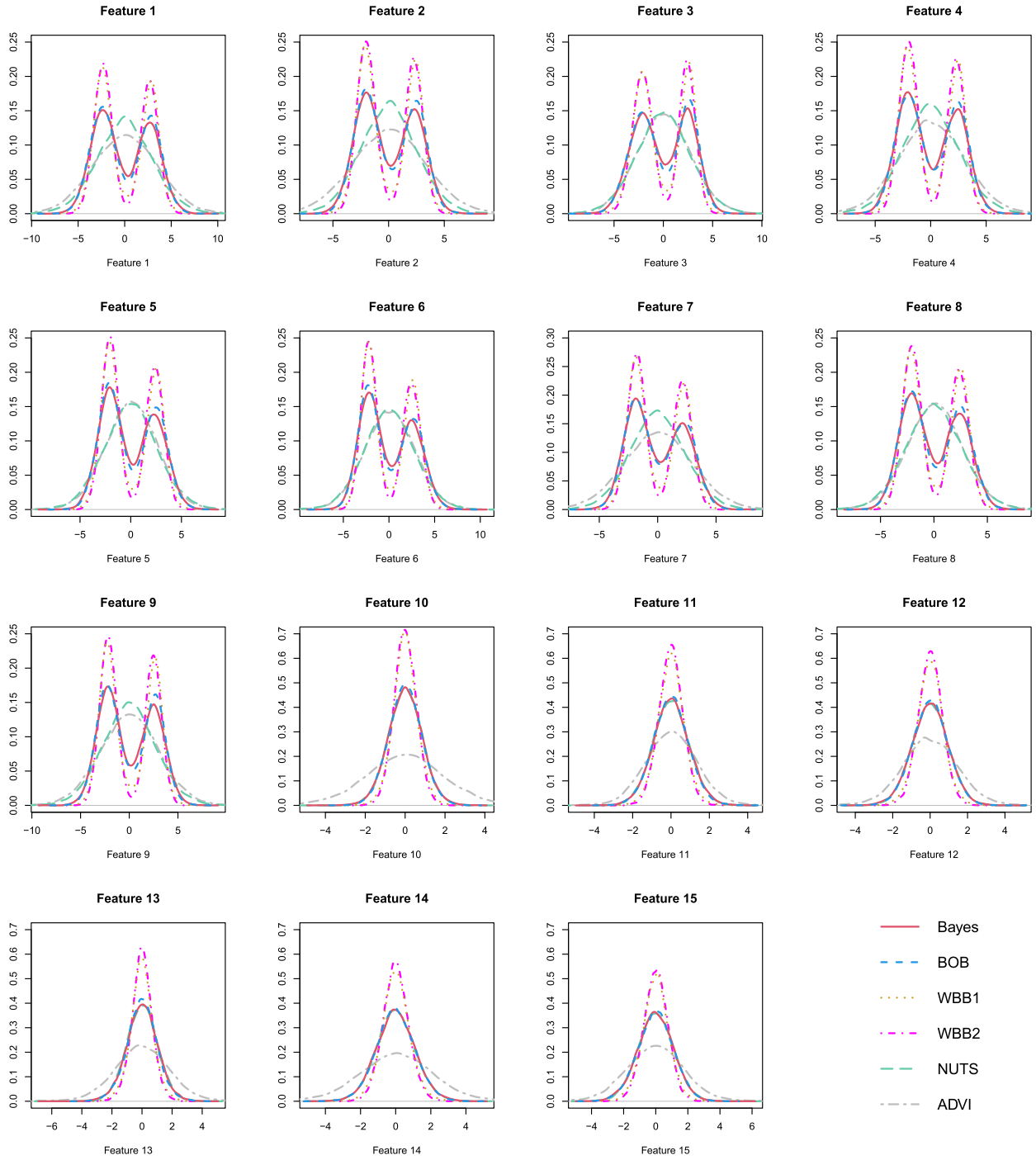

Fig. 12 Univariate density plots of each marginal entry from the approximate posterior predictive distributions obtained with BOB, WBB1, WBB2, NUTS, and ADVI, as well as the target Bayesian posterior predictive distribution, when $K = 2$, $d = 15$, and $n = 50$

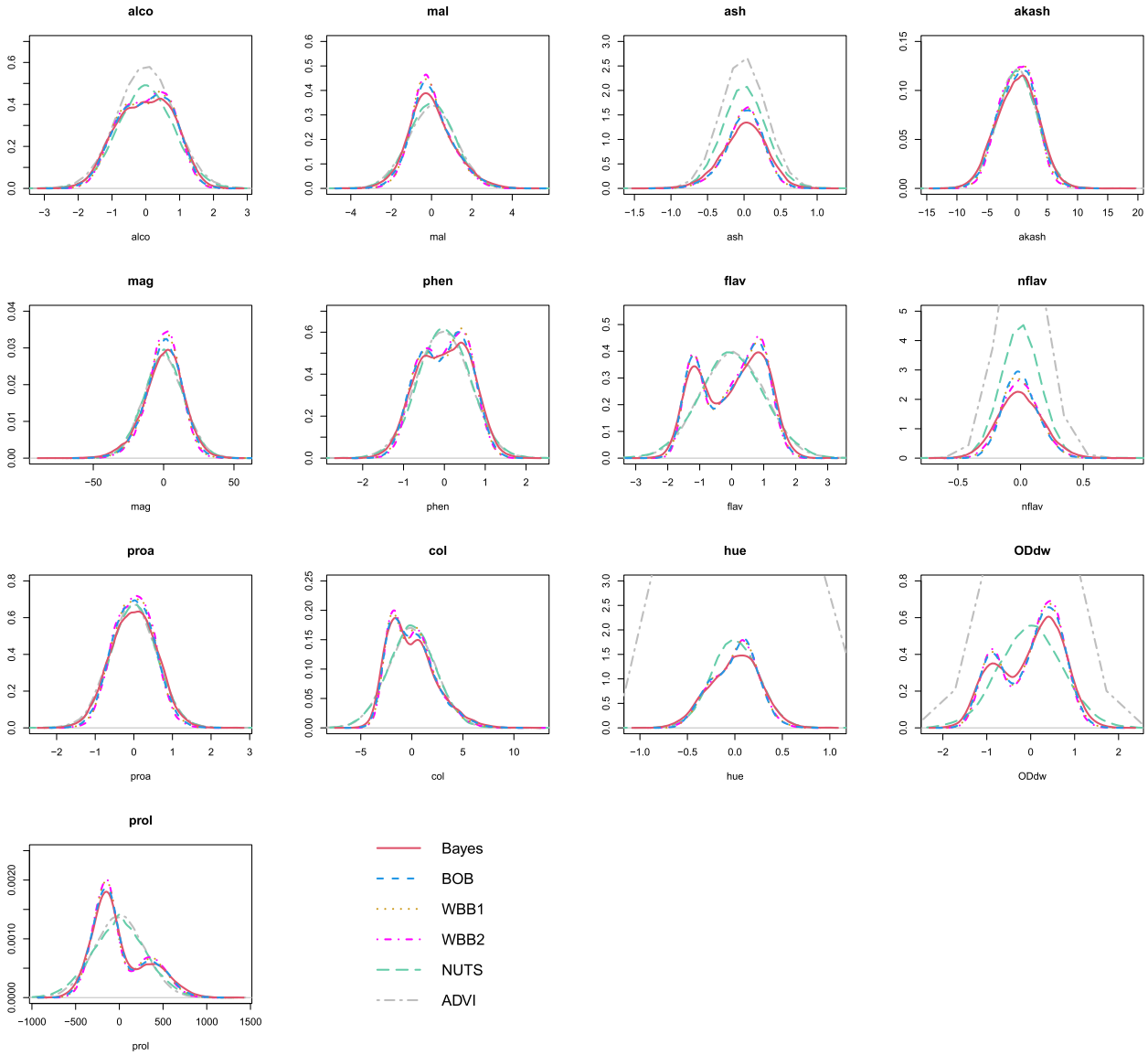

Fig. 13 Univariate density plots of each marginal entry from the approximate posterior predictive distributions obtained with BOB, WBB1, WBB2, NUTS, and ADVI, as well as the target Bayesian posterior predictive distribution, based on the wine dataset

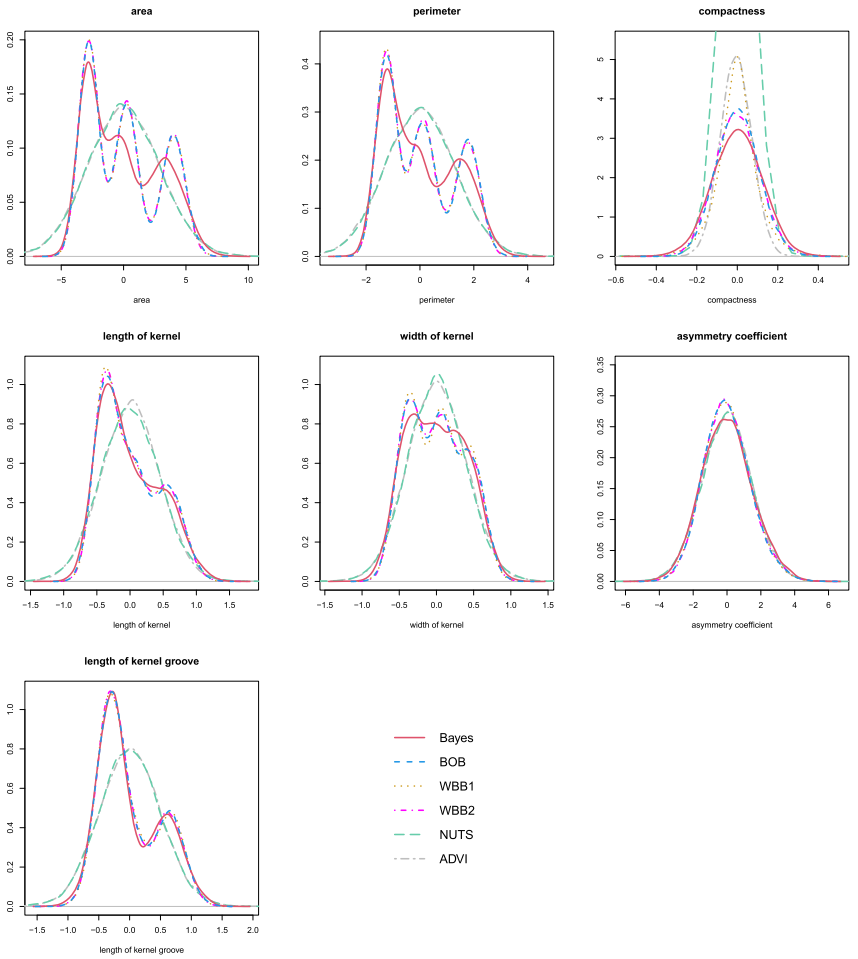

Fig. 14 Univariate density plots of each marginal entry from the approximate posterior predictive distributions obtained with BOB, WBB1, WBB2, NUTS, and ADVI, as well as the target Bayesian posterior predictive distribution, based on the seeds dataset

Appendix G Additional simulation experiments

G.1 Sensitivity to starting values

We now want to explore the sensitivity of WBB and BOB to other starting values. Thus, we consider three additional simulation settings where we fix $n = 100$, $K = 3$, and let $d \in \{5, 10, 15\}$. In all cases, we consider a trivial initialization (TI). In other words, we set, for $k \in \{1, \ldots, K\}$, $\boldsymbol{\mu}_k = (0, \ldots, 0)' \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_k = \mathbb{I}_d \in \mathbb{M}_+^d$, and $\pi_k = 1/K$. Consequently, we call our simulation settings TI 1 (for $d = 5$), TI 2 (for $d = 10$), and TI 3 (for $d = 15$).

Table 10 presents medians (and interquartile ranges) of the SW_2 distances between the Bayesian posterior predictive distribution and its approximations under a trivial initialization, based on 30 independent replications per setting. Additionally, Figure 15 displays boxplots of the distributions of the SW_2 distances produced by BOB, WBB1, and WBB2, all under this trivial initialization.

Table 10 Median SW_2 distances under a trivial initialization (TI), based on 30 independent replications per setting

Method	Setting		
	TI 1 ($d = 5$)	TI 2 ($d = 10$)	TI 3 ($d = 15$)
BOB	**0.703**	**0.560**	**0.587**
	(0.087)	(0.063)	(0.052)
WBB1	0.755	0.821	0.922
	(0.067)	(0.053)	(0.049)
WBB2	0.772	0.851	0.950
	(0.063)	(0.045)	(0.041)

Note: Interquartile ranges are provided in parentheses. The best performance is presented in bold

We can observe, in Table 10 and Figure 15, a huge drop in performance across all samplers, especially if we compare these results to the ones from settings 4, 5, and 6 from Table 2. More precisely, we can observe that a *warm-initialized* BOB is around 8.3 times better than a *trivially-initialized* BOB. Similarly, a *warm-initialized* WBB is around 4.7 times better than a *trivially-initialized* WBB. These results are not

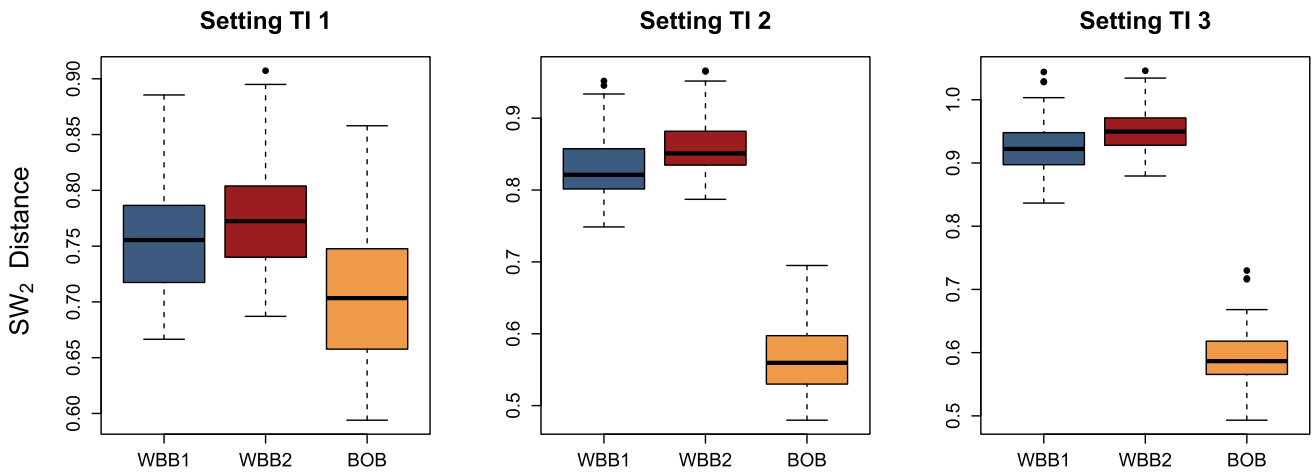

Fig. 15 Boxplots of the distributions of the SW_2 distances under a trivial initialization (TI), based on 30 independent replications per setting

surprising as EM-type algorithms, like the one considered in this article, are notably sensitive to the initial points and poorly chosen initial values might lead the algorithm to saddle at a sub-optimal solution. That being said, it is worth pointing out that even under a trivial initialization, BOB still produces a better approximation of the Bayesian posterior predictive distribution, especially when $d = 10$ and $d = 15$.

G.2 Sensitivity to a strong informative Dirichlet prior

Lastly, we would like to explore the sensitivity of WBB and BOB to a more informative Dirichlet prior on $\boldsymbol{\pi}$. Throughout this article, we have set $\boldsymbol{\pi} \sim \text{Dirichlet}(a_1, \ldots, a_K)$, with $a_k = 1.1$, for all $k \in \{1, \ldots, K\}$. This results in a weak proper prior on $\boldsymbol{\pi}$. However, as pointed out by a reviewer, posterior inference in mixture models is particularly sensitive to the specification of the Dirichlet parameters $\{a_k\}_{k=1}^K$. Thus, we are now considering one last simulation experiment with three settings, where we fix $n = 100$, $K = 3$, and let $d \in \{5, 10, 15\}$. In all cases, we obtain our Dirichlet parameters as

$$\{a_k\}_{k=1}^K \overset{\text{iid}}{\sim} \text{Poisson}(10) + 1.1,$$

which should result in strong informative Dirichlet priors (SIDP). Consequently, we call our simulation settings SIDP 1 (for $d = 5$), SIDP 2 (for $d = 10$), and SIDP 3 (for $d = 15$).

Table 11 Median SW_2 distances under strong informative Dirichlet priors (SIDP), based on 30 independent replications per setting

Method	Setting		
	SIDP 1 ($d = 5$)	SIDP 2 ($d = 10$)	SIDP 3 ($d = 15$)
BOB	**0.172**	**0.131**	**0.172**
	(0.253)	(0.148)	(0.106)
WBB1	0.199	0.260	0.308
	(0.215)	(0.160)	(0.080)
WBB2	0.245	0.308	0.334
	(0.148)	(0.085)	(0.079)

Note: Interquartile ranges are provided in parentheses. The best performance is presented in bold

Table 11 presents medians (and interquartile ranges) of the SW_2 distances between the Bayesian posterior predictive distribution and its approximations under strong informative Dirichlet priors, based on 30 independent replications per setting. Additionally, Figure 16 displays boxplots of the distributions of the SW_2 distances produced by BOB, WBB1, and WBB2, all under strong informative Dirichlet priors.

As in Appendix G.1, we can observe a drop in performance across all samplers, especially if we compare these results to the ones from settings 4, 5, and 6 from Table 2. In this case, however, the drop in performance is less severe. Interestingly, BOB still produces a better approximation of the Bayesian posterior predictive distribution, especially when $d = 10$ and $d = 15$. Overall, this means that incorporating the use of stronger Dirichlet priors could be an avenue for future research.

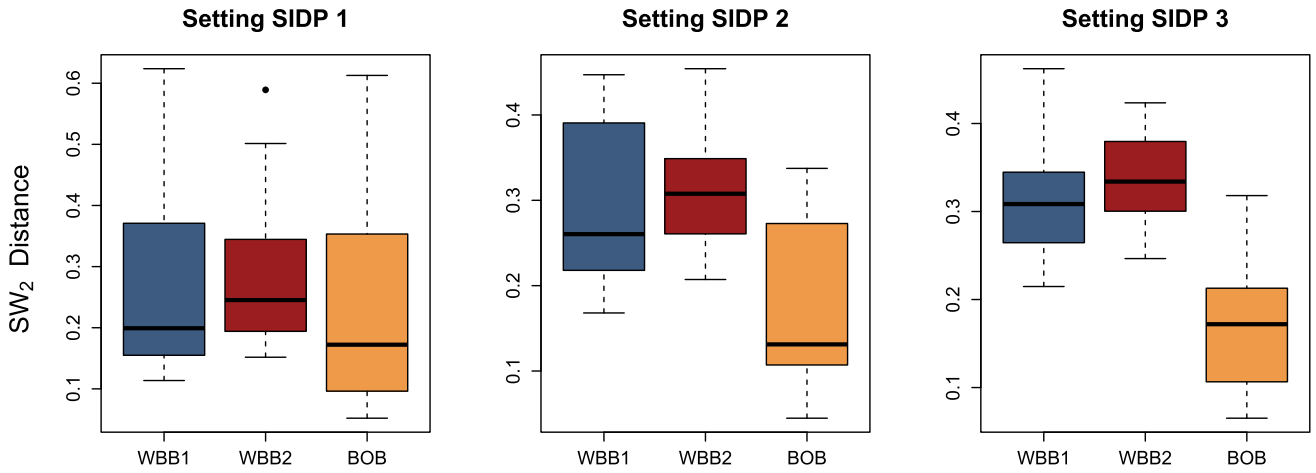

Fig. 16 Boxplots of the distributions of the SW_2 distances under strong informative Dirichlet priors (SIDP), based on 30 independent replications per setting

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-025-10763-y.

Acknowledgements We would like to extend our sincere gratitude to the Editor, an Associate Editor, and two anonymous reviewers for carefully reading our article, and for all their constructive suggestions that have substantially improved the quality of our manuscript. We would also like to thank Michael Martin for many insightful comments and helpful guidance. Any remaining errors are, of course, our own.

Author Contributions S.M. contributed to the paper by conceptualizing and developing the methodology, implementing the software, and writing the initial version of the manuscript. B.L. and A.H.W. provided guidance throughout all stages of the study and critically reviewed the entire article. All authors read and approved the final manuscript.

Data Availability We have used publicly available data from the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu). The *wine* data: (Aeberhard et al. 1994) are available at: https://doi.org/10.24432/C5PC7J. The *seeds* data: (Charytanowicz et al. 2010) are available at: https://doi.org/10.24432/C5H30K. All source code to analyze the data is provided in the supplementary materials.

Declarations

Competing Interests The authors declare no competing interests.

References

Allassonnière, S., Chevallier, J.: A new class of stochastic EM algorithms. Escaping local maxima and handling intractable sampling. Comput. Stat. Data Anal. **159**, 107159 (2021)

Aeberhard, S., Coomans, D., de Vel, O.: Comparative analysis of statistical pattern recognition methods in high dimensional settings. Pattern Recogn. **27**(8), 1065–1077 (1994)

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin, Heidelberg (2006)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)

Baudry, J.-P., Raftery, A.E., Celeux, G., Lo, K., Gottardo, R.: Combining mixture components for clustering. J. Comput. Graph. Stat. **19**(2), 332–353 (2010)

Bull, A.D.: Convergence rates of efficient global optimization algorithms. J. Mach. Learn. Res. **12**(88), 2879–2904 (2011)

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A.: Stan: A probabilistic programming language. J. Stat. Softw. **76**(1), 1–32 (2017)

Celeux, G., Hurn, M., Robert, C.P.: Computational and inferential difficulties with mixture posterior distributions. J. Am. Stat. Assoc. **95**(451), 957–970 (2000)

Cong, Y., Li, S.: Big learning expectation maximization. Proc. AAAI Conference Artificial Intell. **38**(10), 11669–11677 (2024)

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S., Żak, S.: Complete gradient clustering algorithm for features analysis of x-ray images. Information Technologies in Biomedicine 2, 15–24 (2010). Springer

Carvalho, C.M., Polson, N.G., Scott, J.G.: The horseshoe estimator for sparse signals. Biometrika, 465–480 (2010)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Stat Methodol. **39**(1), 1–22 (1977)

Diebolt, J., Robert, C.P.: Estimation of finite mixture distributions through Bayesian sampling. J. R. Stat. Soc. Ser. B Stat Methodol. **56**(2), 363–375 (1994)

Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3), 432–441 (2008)

Fong, E., Lyddon, S., Holmes, C.: Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. Proceed-

ings of the 36th International Conference on Machine Learning 97, 1952–1962 (2019). PMLR
Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. **97**(458), 611–631 (2002)
Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis, 3rd edn. CRC Press, Boca Raton, FL (2014)
Geisser, S.: Predictive Inference, 1st edn. Chapman and Hall/CRC, New York, NY (1993)
George, E.I., McCulloch, R.E.: Variable selection via Gibbs sampling. J. Am. Stat. Assoc. **88**(423), 881–889 (1993)
Hoffman, M.D., Gelman, A.: The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. **15**(47), 1593–1623 (2014)
Hofmeyr, D.P.: Fast exact evaluation of univariate kernel sums. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 447–458 (2021)
Hofmeyr, D.P.: Fast kernel smoothing in R with applications to projection pursuit. J. Stat. Softw. **101**(3), 1–33 (2022)
Izenman, A.J., Sommer, C.J.: Philatelic mixtures and multimodal densities. J. Am. Stat. Assoc. **83**(404), 941–953 (1988)
Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Global Optim. **21**, 345–383 (2001)
Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**, 455–492 (1998)
Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn. Res. **18**(14), 1–45 (2017)
Kandasamy, K., Vysyaraju, K.R., Neiswanger, W., Paria, B., Collins, C.R., Schneider, J., Poczos, B., Xing, E.P.: Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with dragonfly. J. Mach. Learn. Res. **21**(81), 1–27 (2020)
Karlis, D., Xekalaki, E.: Improving the EM algorithm for mixtures. Stat. Comput. **9**(4), 303–307 (1999)
Kolouri, S., Zou, Y., Rohde, G.K.: Sliced Wasserstein kernels for probability distributions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5258–5267 (2016)
Li, Y., Craig, B.A., Bhadra, A.: The graphical horseshoe estimator for inverse covariance matrices. J. Comput. Graph. Stat. **28**(3), 747–757 (2019)
Lartigue, T., Durrleman, S., Allassonnière, S.: Deterministic approximate EM algorithm; application to the Riemann approximation EM and the tempered EM. Algorithms **15**(3), 78 (2022)
Lyddon, S.P., Holmes, C., Walker, S.: General Bayesian updating and the loss-likelihood bootstrap. Biometrika **106**(2), 465–478 (2019)
Le Riche, R., Picheny, V.: Revisiting Bayesian optimization in the light of the coco benchmark. Struct. Multidiscip. Optim. **64**(5), 3063–3087 (2021)
MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. Fifth Berkeley Symposium Math. Stat. Probability **1**(14), 281–297 (1967)
Meng, X.-L., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: A general framework. Biometrika **80**(2), 267–278 (1993)
Morningstar, W., Vikram, S., Ham, C., Gallagher, A., Dillon, J.: Automatic differentiation variational inference with mixtures. International Conference on Artificial Intelligence and Statistics 130, 3250–3258 (2021). PMLR
Malsiner-Walli, G., Frühwirth-Schnatter, S., Grün, B.: Model-based clustering based on sparse finite Gaussian mixtures. Stat. Comput. **26**(1), 303–324 (2016)
Nobile, A., Fearnside, A.T.: Bayesian finite mixtures with an unknown number of components: The allocation sampler. Stat. Comput. **17**, 147–162 (2007)
Ni, Y., Müller, P., Diesendruck, M., Williamson, S., Zhu, Y., Ji, Y.: Scalable Bayesian nonparametric clustering and classification. J. Comput. Graph. Stat. **29**(1), 53–65 (2020)
Ng, T.L., Newton, M.A.: Random weighting in lasso regression. Electronic J. Stat. **16**(1), 3430–3481 (2022)
Newton, M.A., Polson, N.G., Xu, J.: Weighted Bayesian bootstrap for scalable posterior distributions. Canadian J. Stat. **49**(2), 421–437 (2021)
Newton, M.A., Raftery, A.E.: Approximate Bayesian inference with the weighted likelihood bootstrap. J. R. Stat. Soc. Ser. B Stat Methodol. **56**(1), 3–48 (1994)
Nie, L., Ročková, V.: Bayesian bootstrap spike-and-slab lasso. J. Am. Stat. Assoc. **118**(543), 2013–2028 (2023)
Nie, L., Ročková, V.: Deep bootstrap for Bayesian inference. Phil. Trans. R. Soc. A **381**(2247), 20220154 (2023)
Omori, Y., Chib, S., Shephard, N., Nakajima, J.: Stochastic volatility with leverage: Fast and efficient likelihood inference. J. Econometrics **140**(2), 425–449 (2007)
Pompe, E.: Introducing prior information in weighted likelihood bootstrap with applications to model misspecification. arXiv preprint arXiv:2103.14445 (2021)
Provost, S.B., Zang, Y.: Nonparametric copula density estimation methodologies. Mathematics **12**(3), 398 (2024)
Richardson, S., Green, P.J.: On Bayesian analysis of mixtures with an unknown number of components (with discussion). J. R. Stat. Soc. Ser. B Stat Methodol. **59**(4), 731–792 (1997)
Roustant, O., Ginsbourger, D., Deville, Y.: DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. J. Stat. Softw. **51**(1), 1–55 (2012)
Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge, MA (2005)
Raymaekers, J., Zamar, R.H.: Regularized k-means through hard-thresholding. J. Mach. Learn. Res. **23**(93), 1–48 (2022)
Sambridge, M.: A parallel tempering algorithm for probabilistic sampling and multimodal optimization. Geophys. J. Int. **196**(1), 357–374 (2014)
Stringer, A., Brown, P., Stafford, J.: Fast, scalable approximations to posterior distributions in extended latent Gaussian models. J. Comput. Graph. Stat. **32**(1), 84–98 (2023)
Schilling, R.L.: Measures, Integrals and Martingales. Cambridge University Press, Cambridge, UK (2017)
Scrucca, L., Fraley, C., Murphy, T.B., Raftery, A.E.: Model-Based Clustering, Classification, and Density Estimation Using mclust In R. Chapman and Hall/CRC, New York, NY (2023)
Sklar, M.: Fonctions de répartition à N dimensions et leurs marges. Ann. I.S.U.P. **8**(3), 229–231 (1959)
Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. Adv. Neural. Inf. Process. Syst. **25**, 2951–2959 (2012)
Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE **104**(1), 148–175 (2016)
Stephens, M.: Bayesian analysis of mixture models with an unknown number of components–an alternative to reversible jump methods. Ann. Stat. **28**(1), 40–74 (2000)
Stephens, M.: Dealing with label switching in mixture models. J. R. Stat. Soc. Ser. B Stat Methodol. **62**(4), 795–809 (2000)
Van Havre, Z., White, N., Rousseau, J., Mengersen, K.: Overfitting Bayesian mixture models with an unknown number of components. PLoS ONE **10**(7), 0131739 (2015)
Villani, C.: Optimal Transport: Old and New, vol. 338. Springer, Berlin, Heidelberg (2008)
Witten, D.M., Tibshirani, R.: A framework for feature selection in clustering. J. Am. Stat. Assoc. **105**(490), 713–726 (2010)

You, K.: T4transport: Tools for Computational Optimal Transport. (2023). R package version 0.1.2