Journal of the Royal Statistical Society Series B: Statistical Methodology, 2022, **00**, 1–24
https://doi.org/10.1093/jrsssb/qkac007

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS
Journal of the Statistics Society
Series B
Statistical Methodology
B

Original Article

Bayesian fusion: scalable unification of distributed statistical analyses

Hongsheng Dai[1], Murray Pollock[2,3] and Gareth O. Roberts[3,4]

[1]Department of Mathematical Sciences, University of Essex, Colchester, UK
[2]School of Mathematics, Statistics and Physics, Newcastle University, Newcastle-upon-Tyne, UK
[3]The Alan Turing Institute, London, UK
[4]Department of Statistics, University of Warwick, Coventry, UK

Address for correspondence: Hongsheng Dai, Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK. Email: hdaia@essex.ac.uk

Abstract

There has been considerable interest in addressing the problem of unifying distributed analyses into a single coherent inference, which arises in big-data settings, when working under privacy constraints, and in Bayesian model choice. Most existing approaches relied upon approximations of the distributed analyses, which have significant shortcomings—the quality of the inference can degrade rapidly with the number of analyses being unified, and can be substantially biased when unifying analyses that do not concur. In contrast, recent Monte Carlo fusion approach is exact and based on rejection sampling. In this paper, we introduce a practical Bayesian fusion approach by embedding the Monte Carlo fusion framework within a sequential Monte Carlo algorithm. We demonstrate theoretically and empirically that Bayesian fusion is more robust than existing methods.

Keywords: Bayesian inference, distributed data, fork-and-join, Langevin diffusion, sequential Monte Carlo

1 Introduction

There has recently been considerable interest in developing methodology to combine *distributed* statistical inferences, into a single (Bayesian) inference. This distributed scenario can arise for a number of practically compelling reasons. For instance, it occurs in large data settings where, to circumvent the memory constraints on a single machine, we split the available data set across C machines (which we term *cores*) and conduct C separate inferences (Scott et al. 2016). Other modern instances appear when working under confidentiality constraints, where pooling the underlying data would be deemed a data privacy breach (e.g., Yıldırım & Ermiş 2019), and in model selection (Buchholz et al. 2019). More classical instances of this common scenario appear in Bayesian meta-analysis (see, e.g., Fleiss 1993; Smith et al. 1995), and in constructing priors from multiple expert elicitation (Berger 1980; Genest & Zidek 1986).

In particular, in this article, we are interested in finding a sample approximation of the following d-dimensional *product-pooled* target density (which we term the *fusion density*):

$$f(\boldsymbol{x}) \propto f_1(\boldsymbol{x}) \dots f_C(\boldsymbol{x}). \tag{1}$$

Here $\boldsymbol{x} \in \mathbb{R}^d$, and f_c for $c \in \{1, \dots, C\}$ represent the C densities up to a normalising constant (which we term *subposteriors*) which we wish to unify—in what we term the *fusion problem*.

For typical Bayesian problems, $\boldsymbol{x}$ is our parameter space, and f_c can be thought of as the posterior distribution from a Bayesian analysis of the data on the cth core (on a parameter space shared

Received: February 5, 2021. **Revised:** September 9, 2022. **Accepted:** November 26, 2022

by all the subposteriors). If an uninformative prior is specified in the analysis, then the fusion density is simply the posterior given all data. For more general specifications of the prior, a minor adjustment of Equation (1) is required (e.g., by using fractional priors on each subposterior). More specifically, if the prior is denoted as $\pi(\boldsymbol{x})$, the prior on each subposterior can be chosen as $\pi(\boldsymbol{x})^{a_c}$ where $a_c \in (0, 1)$, $\sum_{c=1}^{C} a_c = 1$. Furthermore, we have the cth subposterior proportional to $f_c(\boldsymbol{x}) = l_c(\boldsymbol{x})\pi(\boldsymbol{x})^{a_c}$ where $l_c(\boldsymbol{x})$ is the likelihood function for the model parameter $\boldsymbol{x}$ based on the samples in the cth subdataset. For our purposes, we assume any Bayesian analysis on each core is complete, and we have access to a sample approximation of each subposterior (obtained by, e.g., conducting MCMC on each core). We further assume that we are able to evaluate each f_c point-wise.

Specific applications, such as those we used to introduce the fusion problem, have a number of specific constraints and considerations unique to them. For instance, in the large data setting particular consideration may be given to latency and computer architectures (Scott et al. 2016), whereas in the confidentiality setting of Yıldırım and Ermiş (2019) one may be constrained in the number and type of mathematical operations conducted. Indeed, the majority of the current literature addressing the fusion problem has been developed to address specific applications. Our focus in this paper will not concern any particular application, but rather on methodology for the general fusion problem, which in principle could be applied and adapted to the statistical contexts we describe. Some general discussion on particular applications is given in Section 3.7.

The methodologies proposed in the literature to address the fusion problem are mostly approximate, often supported by underpinning theory which ensures their limiting unbiasedness in an appropriate asymptotic limit. While these methods are often computationally efficient and generally effective, it is generally difficult to assess the extent of the biases introduced by these methods, and equally difficult to correct for these biases. One of the earliest, and most widely used method for dealing with the fusion problem is the Consensus Monte Carlo (CMC) method (Scott 2017; Scott et al. 2016). This method weights samples from individual subposteriors in a way which would be completely unbiased if each subposterior was indeed Gaussian. This is attractive in the large data context which motivated their work. On the other hand, outside the Gaussian context CMC can be very biased (Srivastava et al. 2016; X. Wang et al. 2015). An alternative method involving aggregation techniques based on Weierstrass transforms to each subposterior was proposed in X. Wang and Dunson (2013). In comparison to CMC, the so-called Weierstrass rejection sampler (WRS) is computationally more expensive, although it tends to produce less biased results in the context of non-Gaussian subposteriors. We shall use these two methods as benchmarks to compare our methodology.

Much of the existing approximate literature has been focused on distributed large data settings, and as a consequence there has been particular attention on developing *embarrassingly parallel* procedures, where communication between cores is limited to a single unification step. Often termed as *divide-and-conquer* approaches (although strictly speaking *fork-join* approaches), recent contributions include Neiswanger et al. (2013) who construct a kernel density estimate for each subposterior to reconstruct the posterior density. Other approaches which construct approximations directly from subposterior draws include Minsker et al. (2014), Srivastava et al. (2016), X. Wang et al. (2015), Stamatakis and Aberer (2013), Agarwal and Duchi (2011), Neiswanger et al. (2013), Xue and Liang (2019) and X. Wang and Dunson (2013). Alternative nonembarrassingly parallel approaches are discussed extensively in Jordan et al. (2018) and Xu et al. (2014). Within a hierarchical framework Rendell et al. (2018) (and subsequently Vono et al. 2019) introduce a methodology in which a smoothed approximation to Equation (1) can be obtained if increased communication between the cores is permitted.

In contrast to approximate methods, the Monte Carlo fusion (MCF) approach recently introduced by Dai et al. (2019) provides a theoretical framework to sample independent draws from Equation (1) *exactly* (without any form of approximation). MCF is based upon constructing a rejection sampler on an auxiliary space which admits Equation (1) as a marginal. However, unlike approximate approaches there are considerable computational challenges with MCF. In particular, the scalability of the methodology in terms of the number of subposteriors to be unified, increasing dis-similarity in the subposteriors, and the dimensionality of the underlying fusion target density, all inhibit the practical adoption of the methodology. The challenge that we address successfully in the present paper is to devise a methodology which shares the consistency properties of MCF while sharing the scalability behaviour of the approximate alternatives.

In this paper, we substantially reformulate the theoretical underpinnings of the auxiliary construction used in Dai et al. (2019) to support the use of scalable Monte Carlo methodology. There are a number of substantial and novel contributions:

- We show that it is possible to sample from Equation (1) by means of simulating from the probability measure of a forward stochastic differential equation (SDE).
- Based upon the SDE formulation, we further develop a sequential Monte Carlo (SMC) sampler for Equation (1), in a methodology which we term Bayesian fusion (BF), and which inherits SMC consistency properties (Del Moral 2004; Kunsch 2005).
- We develop extensive theory to show that BF is robust with increasing C, and in settings where the subposteriors lack similarity with one another (which is common in many practical Bayesian settings). The gain of BF on robustness is at the cost of a limited number of extra communications between cores, while the existing methodologies require a single communication.
- For practitioners we provide practical guidance for setting algorithm hyperparameters, which will (approximately) optimise the efficiency of our approach.
- Finally, we provide extensive pedagogical examples and real-data applications to contrast our methodology and scaling with existing approximate and exact approaches.

In the next section, we present the theory that underpins BF, together with methodology and pseudo-code for its implementation in Section 2.1. We provide guidance on implementing BF in Section 3, which includes selection of user-specified parameters in Sections 3.1 and 3.2, studies of the robustness of the algorithm with respect to how similar the subposteriors are in Sections 3.3 and 3.4, and extensive discussion of practical considerations in Sections 3.5, 3.6, and 3.7. Section 4 studies the performance of BF in a number of real data set applications. We conclude in Section 5 with discussion and future directions. We suppress all proofs from the main text, which are instead collated in the appendices. The appendices also include some discussion of the underlying diffusion theory and assumptions (see online supplementary material, Appendix A), theory to support implementations for distributed environments in the online supplementary material, Appendix D, and discussion on the application of the methodology to large data settings in the online supplementary material, Appendix E, and are referenced as appropriate in the main text. Online supplementary material, Appendix G, studies the performance of our methodology in comparison to competing methodologies for idealised models and a synthetic data set.

2 Bayesian fusion

A simple approach for finding a sample approximation of Equation (1) is to note that if we sampled the random variables $\boldsymbol{X}^{(c)} \sim f_c$ for $c \in \{1, \ldots, C\}$, then conditional on $\boldsymbol{X}^{(1)} = \cdots = \boldsymbol{X}^{(C)}$, $\boldsymbol{X}^{(1)}$ has density f as given in Equation (1). Of course, in practice this approach is naive as the conditioning event is of probability 0.

An extension of this naive approach would be to instead simulate C independent stochastic processes initialised at $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(C)}$ respectively at time 0, with invariant densities $f_1, \ldots, f_C$, respectively. Now, we would have a sample from Equation (1) if at some point in time these C independent stochastic processes coincided with one another. Of course, this is again too rare an event for the resultant methodology to be practical.

The MCF approach of Dai et al. (2019) is to instead simulate these C stochastic processes in such a way that they coalesce at a fixed time T. Coercing the processes to merge changes the joint distribution of the processes at any time in a fundamental way. In particular, note that they are no longer independent. As such, a key aspect of the MCF approach is to construct the C stochastic processes in such a way that the marginal distribution at the coalescence time T is the fusion density of Equation (1). Sampling from this object is not possible directly, and so Dai et al. (2019) construct an elaborate rejection sampler. However, in common with many rejection-sampling schemes, there are practical limitations to the MCF approach: in our setting this is robustness with increasing numbers of cores, and the level of (dis-)similarity of the subposteriors.

To devise a more practical version of the rejection-sampling-based MCF, it is natural to replace it with a sequential importance scheme which steps through a sequence of neighbouring distributions between the initial proposal distribution and the target fusion density. In the case of MCF this is not direct, but is in essence what we achieve in this paper with the BF approach we introduce. Here, our sequence of neighbouring distributions will be the joint distribution of our C dependent stochastic processes at times in-between 0 and the coalescence time T. The challenge in this paper is to find tractable dynamics for these C stochastic processes. Our approach is to first derive tractable dynamics for a baseline case (which we term the *proposal measure*, denoted by $\mathbb{P}$) where each of the C stochastic processes is Brownian motion before conditioning, and correct for this using suitable importance weights to find the *fusion measure*, $\mathbb{F}$.

To introduce the fusion measure, $\mathbb{F}$, we first present some notation and terminology. We term the proposal measure, $\mathbb{P}$, to be the probability law induced by C interacting d-dimensional parallel continuous-time Markov processes in $[0, T]$, where each process $\boldsymbol{X}_t^{(c)}$, $c \in \{1, \ldots, C\}$ is described by the following d-dimensional SDE,

$$\mathrm{d}\boldsymbol{X}_t^{(c)} = \frac{\bar{\boldsymbol{X}}_t - \boldsymbol{X}_t^{(c)}}{T-t}\,\mathrm{d}t + \mathrm{d}\boldsymbol{W}_t^{(c)}, \quad \boldsymbol{X}_0^{(c)} := \boldsymbol{x}_0^{(c)} \sim f_c, \quad t \in [0, T], \tag{2}$$

where $\{\boldsymbol{W}_t^{(c)}\}_{c=1}^C$ are independent Brownian motions, and $\bar{\boldsymbol{X}}_t := C^{-1}\sum_{c=1}^C \boldsymbol{X}_t^{(c)}$. Typical realisations of the proposal measure are denoted as $X := \{\vec{\boldsymbol{x}}_t, t \in [0, T]\}$, where $\vec{\boldsymbol{x}}_t := \boldsymbol{x}_t^{(1:C)}$ is the dC-dimensional vector of all processes at time t, with one such realisation illustrated in Figure 1a.

Interaction of the C processes in a realisation of X occurs through their average at a given time marginal ($\bar{\boldsymbol{X}}_t$), and note that we have *coalescence* at time T ($\boldsymbol{x}_T^{(1)} = \cdots = \boldsymbol{x}_T^{(C)} =: \boldsymbol{y}$) which shown via the Doob h-transforms (Rogers & Williams 2000, Section IV.6.39) in the proof of Theorem 1. We describe in detail in Section 2.1 how to simulate from $\mathbb{P}$, but note that (critically) initialisation of the proposal measure at $t = 0$ only requires independent draws from the C available subposteriors.

Now we define the *fusion measure*, $\mathbb{F}$, to be the probability measure induced by the following Radon–Nikodým derivative,

$$\frac{\mathrm{d}\mathbb{F}}{\mathrm{d}\mathbb{P}}(X) \propto \rho_0(\vec{\boldsymbol{x}}_0) \cdot \prod_{c=1}^{C}\left[\exp\left\{-\int_0^T \phi_c(\boldsymbol{x}_t^{(c)})\,\mathrm{d}t\right\}\right], \tag{3}$$

where $\{\boldsymbol{x}_t^{(c)}, t \in [0, T]\}$ is a Brownian bridge from $\boldsymbol{x}_0^{(c)}$ to $\boldsymbol{x}_T^{(c)}$,

$$\phi_c(\boldsymbol{x}) := \triangle f_c(\boldsymbol{x})/2f_c(\boldsymbol{x}), \tag{4}$$

where $\triangle$ is the Laplacian operator, and

$$\rho_0 := \rho_0(\vec{\boldsymbol{x}}_0) = \exp\left\{-\sum_{c=1}^{C}\frac{\|\boldsymbol{x}_0^{(c)} - \bar{\boldsymbol{x}}_0\|^2}{2T}\right\} \in (0, 1], \quad \text{where } \bar{\boldsymbol{x}}_0 = C^{-1}\sum_{c=1}^{C}\boldsymbol{x}_0^{(c)}. \tag{5}$$

We now establish that we can access the fusion density f, by means of the temporal marginal of $\mathbb{F}$ given by common value of the C trajectories at time T. First, we introduce the following regularity assumptions. Let ∇ be the usual gradient operator.

Assumption 2.1 $\nabla \log f_c(\boldsymbol{x})$ is once continuously differentiable.

Assumption 2.2 $\phi_c(\boldsymbol{x})$ is bounded below by some $\Phi_c \le \inf\{\phi_c(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^d\} \in \mathbb{R}$.

Theorem 1. Under Assumptions 2.1 and 2.2, with probability 1 we have that under the fusion measure, $\mathbb{F}$, the ending points of these parallel processes have a common value $\boldsymbol{y} := \boldsymbol{x}_T^{(1)} = \cdots = \boldsymbol{x}_T^{(C)}$ which has density f.

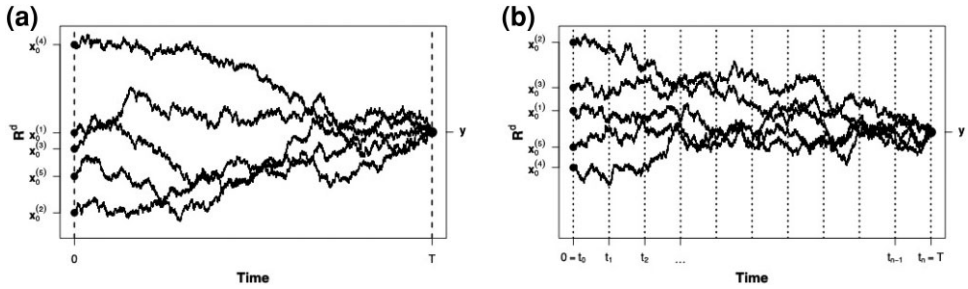

Figure 1. Left plot shows a typical realisation of $\mathfrak{X}$ (C interacting Markov processes), whereas the right plot shows the $d(nC+1)$-dimensional density corresponding to the marginal of $\mathfrak{X}$ given by the temporal partition $\mathcal{P}$. (a) Typical $\mathfrak{X}$. (b) Typical $\mathfrak{X}$ with temporal partition $\mathcal{P}$.

Proof. See online supplementary material, Appendix A. □

2.1 Simulation of f by means of simulating from the fusion measure $\mathbb{F}$

As suggested by Theorem 1 we could simulate from the desired f in Equation (1) by simulating $X \sim \mathbb{F}$ and simply retaining its time T marginal, $\boldsymbol{y}$. However, direct simulation of $\mathbb{F}$ will typically not be possible, and so we now outline general methodology to simulate $\mathbb{F}$ indirectly (and so by extension f). In particular, we show that we can simulate from $\mathbb{F}$ by means of a rejection sampler with proposals $X \sim \mathbb{P}$ which are accepted with probability proportional to the Radon–Nikodým derivative given in Equation (3).

For the purposes of the efficiency of the methodology we will subsequently develop, we will consider the simulation of $\mathbb{P}$ and $\mathbb{F}$ at discrete time points given by the following auxiliary temporal partition:

$$\mathcal{P}\{t_0, t_1, \ldots, t_n : 0 =: t_0 < t_1 < \cdots < t_n := T\}, \tag{6}$$

noting that ultimately we only require the time T marginal corresponding to the nth temporal partition. For simplicity we suppress subscripts when considering the Markov processes at times coinciding with the partition, denoting $\boldsymbol{x}_{t_j}^{(c)}$ as $\boldsymbol{x}_j^{(c)}$, and $\vec{\boldsymbol{x}}_{t_j}$ as $\vec{\boldsymbol{x}}_j$. We further denote $\Delta_j := t_j - t_{j-1}$.

We begin by considering simulating exactly $X \sim \mathbb{P}$ at the points given by the temporal partition, $\mathcal{P}$. To do so, note that the SDE given in Equation (2) is linear and therefore describes a Gaussian process, and its finite-dimensional distributions are explicitly available.

Theorem 2. If X satisfies Equation (2) then under the proposal measure, $\mathbb{P}$, we have

(a) For $s < t$

$$\vec{\boldsymbol{X}}_t \mid (\vec{\boldsymbol{X}}_s = \vec{\boldsymbol{x}}_s) \sim \mathcal{N}(\vec{\boldsymbol{M}}_{s,t}, \boldsymbol{V}_{s,t}), \tag{7}$$

where $\mathcal{N}$ is a multivariate Gaussian density, $\vec{\boldsymbol{M}}_{s,t} = (\boldsymbol{M}_{s,t}^{(1)}, \ldots \boldsymbol{M}_{s,t}^{(C)})$ with

$$\boldsymbol{M}_{s,t}^{(c)} = \frac{T-t}{T-s}\boldsymbol{x}_s^{(c)} + \frac{t-s}{T-s}\bar{\boldsymbol{x}}_s, \tag{8}$$

and where $\boldsymbol{V}_{s,t} = \boldsymbol{\Sigma} \otimes \mathbf{I}_{d\times d}$ with $\boldsymbol{\Sigma} = (\Sigma_{ij})$ being a $C \times C$ matrix given by

$$\Sigma_{ii} = \frac{(t-s)\cdot(T-t)}{T-s} + \frac{(t-s)^2}{C(T-s)}, \quad \Sigma_{ij} = \frac{(t-s)^2}{C(T-s)}. \tag{9}$$

(b) For every $c \in \{1, \ldots, C\}$, the distribution of $\{\boldsymbol{X}_t^{(c)}, s \le u \le t\}$ given endpoints $\boldsymbol{X}_s^c = \boldsymbol{x}_s^{(c)}$ and $\boldsymbol{X}_t^{(c)} = \boldsymbol{x}_t^{(c)}$ is a Brownian bridge, so that

$$\boldsymbol{X}_u^c \mid (\boldsymbol{x}_s^{(c)}, \boldsymbol{x}_t^{(c)}) \sim \mathcal{N}\left(\frac{(t-u)\boldsymbol{x}_s^{(c)} + (u-s)\boldsymbol{x}_t^{(c)}}{t-s}, \frac{(u-s)(t-u)}{t-s}\mathbf{I}_{d\times d}\right). \tag{10}$$

Proof. See online supplementary material, Appendix A. □

To simplify the presentation of the methodology, we now restrict our attention to the $d(nC+1)$-dimensional density of the C d-dimensional Markov processes at the $(n+1)$ time marginals given by the temporal partition under $\mathbb{P}$. An illustration of this is given in Figure 1b. As a consequence of Theorem 2 we have

$$h(\vec{\boldsymbol{x}}_0, \ldots, \vec{\boldsymbol{x}}_{n-1}, \boldsymbol{y}) \propto \prod_{c=1}^{C} [f_c(\boldsymbol{x}_0^{(c)})] \cdot \prod_{j=1}^{n} \mathcal{N}(\vec{\boldsymbol{x}}_j; \vec{\boldsymbol{M}}_j, \boldsymbol{V}_j), \tag{11}$$

where to simplify notation we have set

$$\vec{\boldsymbol{M}}_j := \vec{\boldsymbol{M}}_{t_{j-1},t_j} \quad \text{and} \quad \boldsymbol{V}_j := \boldsymbol{V}_{t_{j-1},t_j}. \tag{12}$$

By factorising Equation (3) according to the temporal partition $\mathcal{P}$, the equivalent $d(nC+1)$-dimensional density under $\mathbb{F}$ is simply

$$g(\vec{\boldsymbol{x}}_0, \ldots, \vec{\boldsymbol{x}}_{n-1}, \boldsymbol{y}) \propto h(\vec{\boldsymbol{x}}_0, \ldots, \vec{\boldsymbol{x}}_{n-1}, \boldsymbol{y}) \cdot \prod_{j=0}^{n} \rho_j, \tag{13}$$

where ρ_0 is as given in Equation (5), for $j \in \{1, \ldots, n\}$,

$$\rho_j := \rho_j(\vec{\boldsymbol{x}}_{j-1}, \vec{\boldsymbol{x}}_j) = \prod_{c=1}^{C} \mathbb{E}_{\mathbb{W}_{j,c}}\left[\exp\left\{-\int_{t_{j-1}}^{t_j} (\phi_c(\boldsymbol{x}_t^{(c)}) - \Phi_c)\, \mathrm{d}t\right\}\right] \in (0, 1], \tag{14}$$

and where $\mathbb{W}_{j,c}$ is the law of a Brownian bridge $\{\boldsymbol{x}_t^{(c)}, t \in (t_{j-1}, t_j)\}$ from $\boldsymbol{x}_{j-1}^{(c)}$ to $\boldsymbol{x}_j^{(c)}$, and Φ_c is a constant such that $\inf_{\boldsymbol{x}} \phi_c(\boldsymbol{x}) \ge \Phi_c > -\infty$ (see Assumption 2.2).

As we are interested in sampling from the *fusion density* f (corresponding to the time T marginal of the $d(nC+1)$-dimensional density g), it is sufficient to simulate g rather than the more complicated object $X \sim \mathbb{F}$. As suggested by Equation (13), simulation from g can be achieved by rejection sampling by first simulating a proposal from the density h, and accepting this proposal with probability equal to $\prod_{j=0}^{n} \rho_j$.

Simulating a proposal from h is made possible by Theorem 2 and Equation (11). In particular, we first simulate a single draw from each subposterior and compose them to obtain a proposal at the time 0 marginal of the temporal partition $\mathcal{P}$ (in particular, $\vec{\boldsymbol{x}}_0 := \boldsymbol{x}_0^{(1:C)}$ where $\boldsymbol{x}_0^{(c)} \sim f_c$ for $c \in \{1, \ldots, C\}$). Here we assume we have access to independent realisations from each subposterior. As discussed in the introduction, we may naturally have only *sample approximations* of each subposterior obtained by some other scheme (e.g., MCMC). We reserve discussion of this scenario to Section 3.6 following the introduction of our (more idealised) methodology here. Our initial draw $\vec{\boldsymbol{x}}_0 := \boldsymbol{x}_0^{(1:C)}$ can then be iteratively propagated n-times using Gaussian transitions (as given in Equation (11)) to compose the entire draw from h.

Now, considering the computation of the acceptance probability of the proposal, note that although ρ_0 is computable, direct computation of $\rho_1, \ldots, \rho_n$ is impossible as it requires the evaluation of path integrals of functionals of Brownian motion. However, it is sufficient for our

purposes to construct unbiased estimators of these intractable quantities. In particular, we do this by first introducing $\tilde{\rho}_j$ for $j \in \{1, \ldots, n\}$ where

$$\tilde{\rho}_j(\vec{\boldsymbol{x}}_{j-1,\cdot}, \vec{\boldsymbol{x}}_{j,\cdot}) := \tilde{\rho}_j = \prod_{c=1}^{C} \left(\frac{\Delta_j^{\kappa_c} \cdot e^{-U_j^{(c)}\Delta_j}}{\kappa_c! \cdot p(\kappa_c \mid R_c)} \prod_{k_c=1}^{\kappa_c} \left(U_j^{(c)} - \phi_c\left(\boldsymbol{x}_{\chi_{c,k_c}}^{(c)}\right) \right) \right). \tag{15}$$

Here R_c, $U_j^{(c)}$, κ_c, $p(\kappa_c \mid R_c)$, and $\chi_{c,i}$ for $c \in \{1, \ldots, C\}$ are additional notations we motivate and introduce in the next two paragraphs.

R_c is the *layered Brownian bridge* construction of Beskos et al. (2008), which can be thought of as a discretisation of the random variable

$$\sup_{t \in [t_{j-1}, t_j]} \left| \frac{\boldsymbol{x}_{j-1}^{(c)} + \boldsymbol{x}_j^{(c)}}{2} - \boldsymbol{x}_t^{(c)} \right|$$

for t in the range $[t_{j-1}, t_j]$ thus giving a measure of the extent to which this particular Brownian path meanders away from the average of its endpoints. Beskos et al. (2008) provide efficient algorithms for simulating from R_c as well as the Brownian path conditional on R_c. Conditional on R_c, an upper bound $U_j^{(c)}$ for $\phi_c(\boldsymbol{x}_t^{(c)})$ can be identified, i.e., the variable $U_j^{(c)}$ is chosen as a deterministic function of R_c and such that $\phi_c(\boldsymbol{x}_t^{(c)}) \leq U_j^{(c)}$ for all $\boldsymbol{x}_t^{(c)} \sim \mathbb{W}_{j,c} \mid R_c$. Note that the variable $U_j^{(c)}$ is a random value in the sense that it depends on R_c, which is actually random.

We further let κ_c be a nonnegative integer-valued random variable with probabilities conditional on R_c denoted by $p(\cdot \mid R_c)$. A full discussion of choosing $p(\kappa_c \mid R_c)$ is given in the online supplementary material, Appendix B, however a common choice is that of a Poisson distribution with parameter $\Delta_j(U_j^{(c)} - \Phi_c)$ as that considerably simplifies Equation (15). Finally, $\{\chi_1, \ldots, \chi_{\kappa_c}\} \overset{i.i.d.}{\sim} \mathcal{U}[t_{j-1}, t_j]$.

The precise construction of Equation (15) can be found in the online supplementary material, Appendix B, and in particular its simulation is possible by means of Algorithm 4. Returning to finding unbiased estimators of $\rho_1, \ldots, \rho_n$, then this is established by the following theorem:

Theorem 3. For every $j \in \{1, \ldots, n\}$, $a_j\tilde{\rho}_j$ is an unbiased estimator of ρ_j, where $a_j := \exp\{\sum_{c=1}^{C} \Phi_c \Delta_j\}$.

Proof. See online supplementary material, Appendix B. □

Now that we have found unbiased estimators for $\rho_1, \ldots, \rho_n$ and we have an implementable rejection sampler for f in Equation (13). In particular, upon simulating a proposal from h we can simply accept the proposal with probability $\prod_{j=0}^{n} a_j\tilde{\rho}_j \in (0, 1]$. The validity of using $a_j\tilde{\rho}_j$ in place of ρ_j follows from Theorem 3 together with of Beskos and Roberts (2005, Prop. 1): the algorithm is statistically equivalent to the original construction (i.e., outputs from both algorithms have identical probabilities), and there is no detrimental effect from the use of the estimators (such as decreased acceptance probabilities, or inflated variance).

Although we could now proceed and implement a rejection sampler, a rejection-sampling approach can suffer from a number of inefficiencies in settings we are typically interested in. For instance, the acceptance probability in Equation (14) will typically decay geometrically with increasing C as each of the terms in the product of Equation (14) is bounded by 1. As another example of an inefficiency, note that Equation (14) will typically decay exponentially with increasing T. Indeed, a simplified variant of this approach termed MCF was introduced by Dai et al. (2019) (it was based upon methodology developed from a substantial simplification of Theorem 2 without the auxiliary temporal partition, $\mathcal{P}$), and does suffer from these (and other) practical shortcomings. Further discussion of the MCF approach is given in the online supplementary material, Appendix G, and contrasted with the methodology we develop in this section.

An immediate extension of the rejection-sampling approach of Dai et al. (2019) would be an importance sampler, in which importance weights are assigned to each of the proposals from h corresponding to the acceptance probability. This would however ultimately suffer from similar inefficiencies to the rejection-sampling approach manifested by variance in the importance weights. A drawback of both rejection and importance sampling approaches, are the computational complications from the simulation of diffusion bridges (required in Equation (15)) which have computational cost which scales exponentially rather than linearly with T. Indeed, this is one of the motivations for introducing the temporal partition, $\mathcal{P}$ (which is fully discussed and specified in Section 3.2).

The key novelty of Theorem 2 is that the auxiliary temporal partition $\mathcal{P}$ which has been introduced allows g to be simulated using an SMC approach. This mitigates the robustness drawbacks of the MCF approach and allows us to leverage the results and approaches available within the SMC literature. In particular, and as suggested by Equation (13), one could initialise an algorithm by simulating N particles from the time 0 marginal of h in Equation (13), $\vec{x}_{0,1}, \ldots, \vec{x}_{0,N}$ (recalling that $\vec{x}_0 := x_t^{(1:C)}$, where for $c \in \{1, \ldots, C\}$ $x_t^{(c)} \sim f_c$), and assigning each an un-normalised importance weight $w'_{0,\cdot} := \rho_0(\vec{x}_{0,\cdot})$. This initial particle set (which constitutes an approximation of the time 0 marginal of g in Equation (13)), can then be iteratively propagated n times by interlacing Gaussian transitions of the particle set over the jth partition of $\mathcal{P}$ (with mean vector $\vec{M}_j$ and covariance matrix $\boldsymbol{V}_j$ as given in Equation (12)), and updating the particle set weightings by a factor of $a_j\tilde{\rho}_j(\vec{x}_{j-1,\cdot}, \vec{x}_{j,\cdot})$. The weighted particle set obtained after the final nth iteration of the algorithm (which is an approximation of the time T marginal of g) can then be used as a proxy for the desired f (as supported by Theorem 2).

We term the SMC approach outlined above BF and present pseudo-code for it in Algorithm 1. Note that in this setting (unlike the rejection-sampling setting) we need to further consider the construction of the unbiased estimator for ρ_j and its variance, which is fully considered in the online supplementary material, Appendix B.

Algorithm 1 outputs a weighted particle set at the end of each iteration, which are then *re-normalised*. In common with much of the SMC literature, we monitor for weight degeneracy by monitoring the particle weights, and if the estimated effective sample size (ESS) falls below a lower user-specified threshold then *resampling*. For our BF approach we adopted the widely used ESS convention of (Kong et al. 1994), and employed a *multinomial* resampling strategy (Gordon et al. 1993) (although, the resampling step can be modified to a variety of other strategies common in SMC, such as those in Kitagawa 1996 and Doucet et al. 2001). Note that although commonly used, the appropriateness of the ESS heuristic within SMC is disputed within the literature, as it can give misleading or suboptimal results. As acknowledged in Kong et al. (1994), ESS is only a loose approximation. (Elvira et al. 2022) pointed out that it could overestimate the theoretical ESS value under a small particle size. This means that improvement using integrand dependent metric could be achieved. Possible solutions were pointed out by Elvira et al. (2022). Also, when comparing an importance sampling estimator with an estimator based on i.i.d. sampling, the ESS criterion can judge the importance sampling estimator as inferior when the opposite is true. Therefore, as suggested by Elvira et al. (2022), caution should be used when interpreting results based on the ESS formula, which is outwith the scope of this paper.

Note that re-normalisation in Algorithm 1 Step bi of the BF approach removes all contributory components of $\Phi_1, \ldots, \Phi_C$ from $a_j\tilde{\rho}_j(\vec{x}_{j-1,\cdot}, \vec{x}_{j,\cdot})$, as a_j is a constant for all particles and will be cancelled out in the re-normalisation. This conveniently allows us to avoid the computation of $\Phi_1, \ldots, \Phi_C$, and so we only need to evaluate $\tilde{\rho}_j$ in Equation (15).

As suggested by Algorithm 1, the output can be used directly as an approximation for the fusion density, f. The efficiency of the BF approach outlined in Algorithm 1 will depend critically on the user-specified time horizon T, and the resolution of $\mathcal{P}$ (and hence the number of iterations required in the algorithm). In the following section, we provide guidance on selecting these tuning parameters, together with additional practical guidance on implementation.

Algorithm 1 Bayesian Fusion Algorithm.

(a) **Initialisation Step** ($j = 0$)**:**
  (i) **Input:** (Un-normalised) subposteriors, $f_1, \ldots, f_C$, number of particles, N, time horizon, T and set $t_0 = 0$.
  (ii) For i in 1 to N,
    A. $\vec{\boldsymbol{x}}_{0,i}$: For c in 1 to C, simulate $\boldsymbol{x}_{0,i}^{(c)} \sim f_c$. Set $\vec{\boldsymbol{x}}_{0,i} := \boldsymbol{x}_{0,i}^{(1:C)}$.
    B. $w'_{0,i}$: Compute un-normalised weight $w'_{0,i} = \rho_0(\vec{\boldsymbol{x}}_{0,i})$, as per Equation (5).
  (iii) $w_{0,\cdot}$: For i in 1 to N compute normalised weight $w_{0,i} = w'_{0,i} / \sum_{k=1}^{N} w'_{0,k}$.
  (iv) g_0^N: Set $g_0^N(\mathrm{d}\vec{\boldsymbol{x}}_0) := \sum_{i=1}^{N} w_{0,i} \cdot \delta_{\vec{\boldsymbol{x}}_{0,i}}(\mathrm{d}\vec{\boldsymbol{x}}_0)$.
(b) **Iterative Update Steps** ($j = j + 1$) **while** $t_{j-1} < T$:
  (i) **Resample:** If the ESS $:= (\sum_{i=1}^{N} w_{j-1,i}^2)^{-1}$ breaches the lower user-specified threshold, then for i in 1 to N resample $\vec{\boldsymbol{x}}_{j-1,i} \sim g_{j-1}^N$, and set $w_{j-1,i} = 1/N$.
  (ii) t_j: Set Δ_j as guided by, say, Remark 6 and set $t_j = \min\{T, t_{j-1} + \Delta_j\}$.
  (iii) For i in 1 to N,
    A. $\vec{\boldsymbol{x}}_{j,i}$: Simulate $\vec{\boldsymbol{x}}_{j,i} \sim \mathcal{N}(\vec{\boldsymbol{x}}_{j-1,i}; \vec{\boldsymbol{M}}_{j,i}, \boldsymbol{V}_j)$, where $\vec{\boldsymbol{M}}_{j,i}$ and $\boldsymbol{V}_j$ are computed using Theorem 2.
    B. $w'_{j,i}$: Compute weight $w'_{j,i} = w_{j-1,i} \cdot \tilde{\rho}_j(\vec{\boldsymbol{x}}_{j-1,i}, \vec{\boldsymbol{x}}_{j,i})$ as per Equation (15).
  (iv) $w_{j,\cdot}$: For i in 1 to N compute normalised weight $w_{j,i} = w'_{j,i} / \sum_{k=1}^{N} w'_{j,k}$.
  (v) g_j^N: Set $g_j^N(\mathrm{d}\vec{\boldsymbol{x}}_j) := \sum_{i=1}^{N} w_{j,i} \cdot \delta_{\vec{\boldsymbol{x}}_{j,i}}(\mathrm{d}\vec{\boldsymbol{x}}_j)$.
(c) **Output:** $\hat{f}(\mathrm{d}\boldsymbol{y}) := g_n^N(\mathrm{d}\boldsymbol{y}) \approx f(\mathrm{d}\boldsymbol{y})$.

3 Theoretical underpinning and implementational guidance

In common with other fusion approaches, a key consideration of BF is the distributed nature of the cores and respective subposteriors. For instance, in a distributed big-data setting the data remains separated across the cores throughout, and communication *between* cores is computationally expensive. Methods such as CMC (Scott et al. 2016), embarrassingly parallel MCMC (Neiswanger et al. 2013), and double-parallel Monte Carlo (Neiswanger et al. 2013) reduce communication to a single instance, whereas more recent approaches (e.g., Rendell et al. 2018; Vono et al. 2019) permit a limited number of communications in an effort to reduce the level of approximation of Equation (1). Although the *exact* MCF approach of Dai et al. (2019) only requires a single instance of communication, our more robust (yet consistent) BF approach requires a limited number of further communications (in total, n instances). As a consequence, efficiently implementing BF to ensure strong scalability and robustness properties is critical, particularly in terms of the number of cores and discrepancy between subposteriors.

To this end, in this section, we provide guidance on how to select the user-specified time horizon (T) and an appropriate resolution of the auxiliary temporal partition (n and $\mathcal{P}$). This is considered in Sections 3.1 and 3.2, respectively. The robustness of this guidance is considered by means of two extreme possible scenarios in Sections 3.3 and 3.4. Finally, in Sections 3.5–3.7, we provide further practical guidance.

We begin in developing guidance for T and $\mathcal{P}$ (or n), by noting that Algorithm 1 is an SMC algorithm for simulating the extended target density g in Equation (13), which is achieved by approximating successive temporal marginals of g (in particular, g_j^N) by means of propagating and re-weighting the previous temporal marginal (g_{j-1}^N). As such, it is natural to choose T, n and $\mathcal{P}$ to ensure the discrepancy between the sequence of proposal and target distributions is not degenerate, and so ESS is an appropriate quantity to analyse (see Kong et al. 1994). We here use the term conditional effective sample size (CESS), following the convention of Zhou et al. (2016):

$$\mathrm{CESS}_j := \frac{(\sum_{i=1}^{N} \tilde{\rho}_{j,i})^2}{\sum_{i=1}^{N} \tilde{\rho}_{j,i}^2}, \quad j = 1, \ldots, n; \quad \mathrm{CESS}_0 = \frac{(\sum_{i=1}^{N} \rho_{0,i})^2}{\sum_{i=1}^{N} \rho_{0,i}^2}, \tag{16}$$

where $\rho_{0,i}$ as per Equation (5). To develop concrete implementational guidance we consider and analyse the idealised setting of posterior distributions of large sample size m. In particular, throughout this section, we assume that the target density f is multivariate Gaussian with mean vector $\boldsymbol{a}$ and covariance matrix $m^{-1}b\mathbf{I}$ (for some $b > 0$), and each of the subposterior densities $f_c(\boldsymbol{x})$ ($c \in \{1, \ldots, C\}$) is also multivariate Gaussian but with mean vector $\boldsymbol{a}_c$ and covariance matrix $m^{-1}Cb\mathbf{I}$, respectively. Note that we have $\boldsymbol{a} = C^{-1}\sum_{c=1}^{C}\boldsymbol{a}_c$, and we will further reasonably assume $m > C > 1$. To study the robustness of Algorithm 1 we further consider the quantity $\sigma_a^2 := C^{-1}\sum_{c=1}^{C}\|\boldsymbol{a}_c - \boldsymbol{a}\|^2$ which gives a measure of what we term the *subposterior heterogeneity* (the degree to which the individual subposteriors agree or disagree with one another).

3.1 Guidance on selecting T

Considering the selection of T note from Algorithm 1 that its influence appears solely in the initial weighting given to each of the N particles in Equation (5) through ρ_0. As such, we study the *initial* CESS.

Theorem 4. Considering the initial CESS (CESS_0), we have that as $N \to \infty$, the following convergence in probability holds:

$$N^{-1}\mathrm{CESS}_0 \xrightarrow{p} \exp\left\{-\frac{\frac{\sigma_a^2 b}{m}}{\left(\frac{T}{C}+\frac{b}{m}\right)\cdot\left(\frac{T}{C}+\frac{2b}{m}\right)}\right\}\cdot\left[1+\frac{\left(\frac{Cb}{Tm}\right)^2}{1+\frac{2Cb}{Tm}}\right]^{-(C-1)d/2}.$$

Proof. See online supplementary material, Appendix C. □

Theorem 4 shows explicitly how CESS_0 degrades as the level of subposterior heterogeneity (σ_a^2) increases. To explore this dependency we introduce the following conditions which will allow us to clearly identify regimes where CESS_0 is well-behaved.

Condition 1 (SH(λ))

The subposteriors obey the *subposterior homogeneity* SH(λ) condition (for some constant $\lambda > 0$) if, $\sigma_a^2 = b(C-1)\lambda/m$.

Condition 2 (SSH(γ))

The subposteriors obey the *super subposterior heterogeneity* SSH(γ) condition (for some constant $\gamma > 0$) if, $\sigma_a^2 = b\gamma$.

Note that Condition 1 is a very natural condition which would arise in many settings (e.g., if (m/C)th of the data was randomly allocated to each subposterior then $\sigma_a^2 \sim b/m \times \chi^2_{C-1}$ and thereby have mean $b(C-1)/m$). For m/C large we would expect that for $\lambda > 1$ the subposteriors would obey the SH(λ) condition with high probability. Whereas at the other end of the spectrum, the SSH(γ) condition of Condition 2 captures the case where subposterior heterogeneity does not decay with m.

Considering the initial CESS under Conditions 1 and 2 we establish the following corollary.

Corollary 1. If for some constant $k_1 > 0$, T is chosen such that

$$T \geq \frac{bC^{3/2}k_1}{m}, \tag{17}$$

then the following lower bounds on CESS_0 hold:

(a) If SH(λ) holds for some $\lambda > 0$, then

$$\lim_{N\to\infty} N^{-1}\mathrm{CESS}_0 \geq \exp\{-\lambda k_1^{-2} - dk_1^{-2}/2\}. \tag{18}$$

(b) If SSH(γ) holds for some $\gamma > 0$, and $T \geq k_2 C^{1/2}$ (for some constant $k_2 > 0$), then

$$\lim_{N\to\infty} N^{-1}\mathrm{CESS}_0 \geq \exp\{-\gamma b k_1^{-1}k_2^{-1} - dk_1^{-2}/2\}. \tag{19}$$

Proof. See online supplementary material, Appendix C. □

Remark 1 (Choosing k_1, k_2)

We will use Corollary 1 to select appropriate choices for k_1 and k_2 to ensure that, with high probability, CESS_0 exceeds a prescribed threshold. To do this we will need to compute estimates of b from the population variance; λ, calculated from the variance of the subposterior means and Condition 1; and σ_a^2, the variance of the subposterior means).

We begin by choosing $\zeta \in (0, 1)$ to be a lower bound on the initial ESS we would tolerate. Then,

(a) If SH(λ) holds, then to guarantee that $N^{-1}\mathrm{CESS}_0 > \zeta$, Equation (18) suggests choosing k_1 such that $\exp(-(\lambda + d/2)k_1^{-2}) = \zeta$, i.e., $k_1 = 1/\sqrt{-\log(\zeta)/(\lambda + d/2)}$.
(b) If SSH(γ) holds, then Equation (19) suggests choosing k_1 and k_2 such that

$$\exp\{-\gamma b k_1^{-1}k_2^{-1} - dk_1^{-2}/2\} = \zeta \tag{20}$$

for the ESS $N^{-1}\mathrm{CESS}_0 > \zeta$. Corollary 1 indicates that in the SSH(γ) setting T should be chosen such that $T \geq \max(bC^{3/2}k_1/m, k_2C^{1/2})$. On the other hand, we do not wish to choose too large a T as in Algorithm 1 the computational cost is driven by Step b, and here we want to make both $bC^{3/2}k_1/m$ and $k_2C^{1/2}$ small. As such, we set $bC^{3/2}k_1/m = k_2C^{1/2}$, i.e., we choose $k_2 = bCk_1/m$. Substituting this into Equation (20), we then choose $k_1^2 = (m\gamma/C + d/2)(-\log\zeta)^{-1}$.

In practice, preliminary runs of BF can be used to refine these initial choices.

Remark 2 (T)

Given k_1 and k_2, then T can be chosen as per Equation (17) in the SH(λ) setting, or as per Corollary 1 if SSH(γ) holds. Choosing T as small as possible within this minimal guidance, minimises the introduction of the additional communication and computation required in Algorithm 1 Step (b).

3.2 Guidance on selecting $\mathcal{P}$

Having selected an appropriate T following the guidance of Corollary 1 and Remarks 1 and 2, we are left with choosing the remaining user-specified parameters n and $\mathcal{P}$ (the resolution and spacing of the auxiliary temporal partition), as required in Algorithm 1. We address this implicitly by considering how to choose the jth interval size (i.e., the interval $(t_{j-1}, t_j]$), which we do so by again considering the CESS in Theorem 5.

Theorem 5. Let k_3 and k_4 be positive constants, and choose $p(\kappa_c \mid R_c)$ to be a Poisson distribution with intensity $[\Delta_j \int_{t_{j-1}}^{t_j} (U_j^{(c)} - \phi_c(\boldsymbol{x}_t^{(c)}))^2\,\mathrm{d}t]^{1/2}$ in specifying $\tilde{\rho}$ in Equation (15). If $\lim_{\Delta_j\to 0}$ is taken over sequences of $\Delta_j \to 0$ with

$$\Delta_j \leq \widetilde{\Delta}_j := \min\left\{\frac{b^2k_3C}{m^2(\mathbb{E}v_j)}, \left(\frac{k_4b^2C}{2dm^2}\right)^{1/2}\right\}, \tag{21}$$

where $v_j := C^{-1} \sum_{c=1}^{C} \|\boldsymbol{x}_{j-1}^{(c)} - \boldsymbol{a}_c\|^2$ and the expectation $\mathbb{E}$ is taken conditional on $\vec{\boldsymbol{x}}_{j-1}$, we have

$$\lim_{\Delta_j \to 0} \lim_{N \to \infty} N^{-1} \mathrm{CESS}_j \geq e^{-k_3 - k_4}, \tag{22}$$

where $\lim_{N\to\infty} N^{-1}\mathrm{CESS}_j$ means convergence in probability.

Proof. See online supplementary material, Appendix C. □

Remark 3 (k_3, k_4)

Choosing $\zeta' \in (0, 1)$ to be a lower bound on the ESS we would tolerate, then we can choose k_3 and k_4 such that $e^{-k_3-k_4} = \zeta'$.

Remark 4 (v_j, $\mathbb{E}v_j$)

In essence, v_j in Theorem 5 describes the average variation of the C trajectories of the distribution of their proposed update locations with respect to their individual subposterior mean (i.e., how different $\boldsymbol{x}_{j-1}^{(c)}$ is from $\boldsymbol{a}_c$). Recalling that Algorithm 1 is coalescing C trajectories initialised independently from their respective subposteriors to a common end point, then v_j will largely be determined by a combination of how close the interval is to the end point T, how large the interval $(t_{j-1}, t_j]$ we are simulating over is, and critically the degree of *subposterior heterogeneity* as determined by variation in their mean. Although $\mathbb{E}v_j$ is not computable, it only depends on the distribution of $\vec{\boldsymbol{x}}_{j-1}$ (at time t_{j-1}), and so a natural estimator is $\widehat{\mathbb{E}v_j} = N^{-1} \sum_{i=1}^{N} (C^{-1} \sum_{c=1}^{C} \|\boldsymbol{x}_{j-1,i}^{(c)} - \boldsymbol{a}_c\|^2)$.

Remark 5 (Δ_j, $\tilde{\Delta}$, n, $\mathcal{P}$)

Recalling $t_0 = 0$, and using the guidance for $\widehat{\mathbb{E}v_j}$ in Remark 4, and k_3 and k_4 in Remark 3, then following Theorem 5 we can iteratively approximate $\tilde{\Delta}$ and so we recommend setting $t_j = \min\{T,\ t_{j-1} + \widetilde{\Delta}_j\}$. Thus, by a recursive argument we find n and specify $\mathcal{P}$. Of course, choosing interval sizes ($\Delta_j = t_j - t_{j-1}$) smaller than this guidance is possible (and may help computationally in the simulation of $\tilde{p}_{\cdot}$ as per Algorithm 4) but leads to an overall increased number of iterations (and so increased communication between different cores) in Algorithm 1. Note that following this guidance we have an irregular temporal mesh, $\mathcal{P}$.

Remark 6 ($\mathcal{P}$ regularity)

In the case where second term on the RHS of Equation (21) is the smaller (e.g., when C is large), the guidance of Remark 5 results in a regular temporal mesh, $\mathcal{P}$. Regular temporal meshes have practical and computational advantages, and behave well and robustly in the examples we have explored. For instance, for large data sets with observations randomly allocated to subposteriors then subposterior heterogeneity will be small, and one would anticipate $\mathbb{E}v_j$ to be small and of the order of $\mathcal{O}(m^{-1})$. Here, for algorithmic simplicity we can impose a *regular mesh* ($\Delta := \Delta_j = t_j - t_{j-1} = (k_4 b^2 C / 2dm^2)^{1/2}, \forall j$), and so $n = T/\Delta$.

Having established guidance for choosing T, n and $\mathcal{P}$ for BF, we now verify that these selections lead to BF being robust to increasing data size (as measured by CESS). We do so by studying the guidance in idealised settings for the posterior distribution under the SH(λ) and SSH(γ) conditions, which we do in Sections 3.3 and 3.4, respectively. Note that we consider more substantial examples and comparisons with competing methodologies in Section 4, and in Appendix G of the online supplementary material. Following Remark 6, we further discuss the temporal regularity of the mesh in Section 3.5.

3.3 Subposteriors with similar mean

We begin by examining the guidance for T and n in BF under the SH(λ) setting of Condition 1. Recall this would be the most common setting of relatively homogeneous subposteriors (as

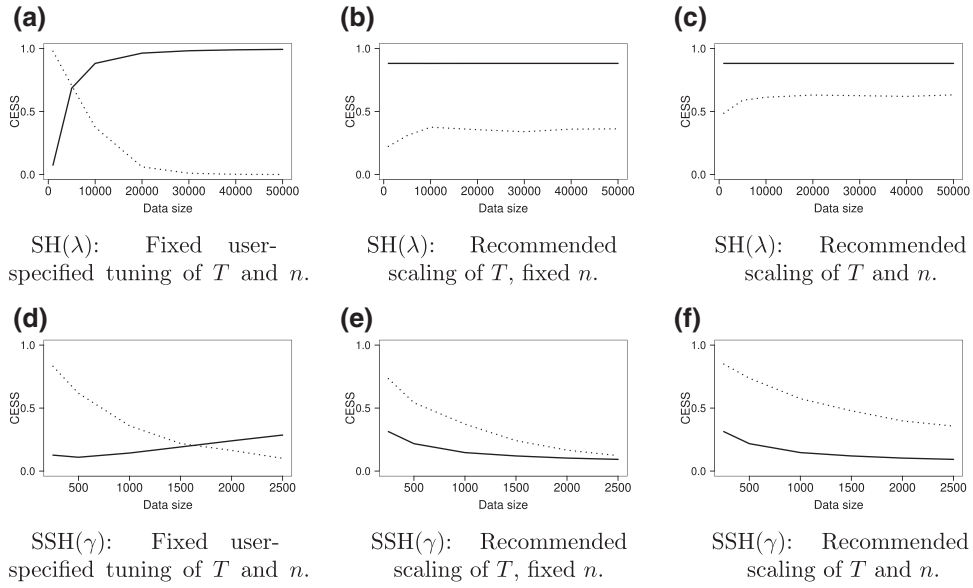

Figure 2. CESS of Algorithm 1 with increasing data size in SH(λ) setting of Section 3.3 (a–c) and SSH(γ) setting of Section 3.4 (d–f). Solid lines denote initial CESS (CESS_0, following Algorithm 1 Step (a)). Dotted lines denote averaged CESS in subsequent iterations of Algorithm 1 ($(\sum_{j=1}^{n} \text{CESS}_j)/n$, following Algorithm 1 Step (b)).

characterised by variation in the subposterior mean), which would occur if for instance we were able to randomly allocate approximately a Cth of the available data to each subposterior. To do so we consider the idealised scenario in which we wish to recover a target distribution f, which is Gaussian with mean $\mu = 0$ and variance $\sigma^2 = m^{-1}$, by applying Algorithm 1 to unify C subposteriors (f_c, $c \in \{1, \ldots, C\}$), which are Gaussian with mean $\mu_c = 0$ and variance $\sigma_c^2 = C\sigma^2$. In this example we consider a range of data sizes from $m = 1,000$ to $m = 50,000$, with a fixed number of subposteriors ($C = 10$), and using a particle set of size $N = 10,000$. In implementing Algorithm 1 we use UE-b (Condition B.2) of the online supplementary material, Appendix B for simulating the unbiased estimator in Step b(iii)B.

Here, we consider CESS_0 and CESS_j ($j \in \{1, \ldots, n\}$) with increasing data size by first considering fixed choices for T and n ($T = 0.005$ and $n = 5$), then choosing a robust scaling of T but with fixed n, and then robustly scaling T and n. This procedure is summarised in Remark 7, and the results are presented in Figures 2a–2c.

Remark 7 (SH(λ) parameter setting)

We set the BF tuning parameters as follows:

1. In line with Remark 1, prior to setting T we determine a lower bound on the initial ESS we would tolerate (ζ). Here, we conservatively chose $\zeta N = \exp(-2)N \approx 0.2N$. In this example, $\lambda \approx 1$ and $d = 1$, and so we set $k_1 = 1$.
2. Now, following Remark 2 and Equation (17), and noting $b = 1$, we choose $T = C^{3/2}k_1/m$.
3. Prior to choosing a temporal partition, following Remark 3 we again choose a lower bound on the CESS we would tolerate. Here, for simplicity we choose $k_3 = k_4 = 1$, and so $\zeta' N = \exp(-2)N \approx 0.14N$.
4. We can now set n and $\mathcal{P}$ following Remark 5. As per Remark 6, we use a regular partition. As such, $\Delta_j = (k_4 b^2 C/2dm^2)^{1/2}$ (i.e., $n = T/\Delta_j = \mathcal{O}(C)$).

Considering the results of fixing T and n in Figure 2a, it is clear in this regime that Algorithm 1 would lack robustness with increasing data size. Although CESS_0 improves with increasing data size as expected with increasingly similar subposteriors from Equation (5) of Theorem 2, this

comes with drastically decreasing CESS_j (as suggested by Theorem 5), which in totality would render the methodology impractical.

Scaling T following the above guidance immediately stabilises both CESS_0 and CESS_j in the SH(λ) setting, making Algorithm 1 robust to increasing data size (as shown in Figure 2b). Additionally scaling n substantively improves CESS_j for all data sizes. In both Figures 2b and 2c the slightly decreased CESS_j for small data sizes can be explained by random variation in the simulation of the subposterior, which leads to slight mis-matching.

3.4 Subposteriors with dissimilar mean

Now we examine the guidance for T and n in BF under the SSH(γ) setting of Condition 2. Recall this would be an extreme setting in which subposterior heterogeneity does not decay with data size, m. To investigate this setting we consider recovering a target distribution f, which is Gaussian with mean $\mu = 0$ and variance $\sigma^2 = m^{-1}$, by using Algorithm 1 to unify $C = 2$ subposteriors with mean $\mu_c = \pm 0.25$ and variance $\sigma_c^2 = 2\sigma^2$. In this scenario as data size increases the subposteriors have increasingly diminishing common support, although our measure of heterogeneity is fixed with $\sigma_a^2 = 0.0625$. In this example we consider a range of data sizes from $m = 250$ to $m = 2,500$, and use a particle set of size $N = 10,000$. We again use UE-b (Condition B.2) of the online supplementary material, Appendix B for simulating the unbiased estimator in Step b(iii)B when implementing Algorithm 1.

As in the SH(λ) setting of Section 3.3, for this SSH(γ) setting we consider CESS_0 and CESS_j ($j \in \{1, \ldots, n\}$) with increasing data size with fixed choices for T and n ($T = 0.01$ and $n = 5$), then choose a robust scaling of T but with fixed n (as in Section 3.1), and then robustly scale both T and n (as in Section 3.2). This procedure is summarised in Remark 8, and the results are presented in Figures 2d–2f.

Remark 8 (SSH(γ) parameter setting)

We set the BF tuning parameters as follows:

1. Following the guidance of Remark 1 we begin by choosing ζ. Here, we conservatively chose $\zeta N = \exp(-2)N \approx 0.14N$. In this example, $b = 1$, $d = 1$, and $\gamma = \sigma_a^2 = 0.0625$ Consequently, we have $k_1 = (m\gamma/C + d/2)^{1/2}(-\log\zeta)^{-1/2}$ and $k_2 = bCk_1/m$.
2. Now, following Remark 2 and Corollary 1, we choose $T = \max\{C^{3/2}k_1/m, k_2C^{1/2}\}$ which is $T = \mathcal{O}(C/\sqrt{m})$ under the above choice of k_1, k_2.
3. Prior to choosing a temporal partition, following Remark 3 we again choose a lower bound on the CESS we would tolerate. Here, for simplicity we again choose $k_3 = k_4 = 1$, and so $\zeta' N = \exp(-2)N \approx 0.14N$.
4. We can now set n and $\mathcal{P}$ following Remark 5. As per Remark 6, we use a regular partition. As such, $\Delta_j = (k_4 b^2 C/2dm^2)^{1/2}$ (i.e., $n = \mathcal{O}((mC)^{1/2})$).

It is clear from the results for the SSH(γ) setting in Figures 2d–2f, and contrasting them with the SH(λ) setting of Figures 2a–2c, that the SSH(γ) setting is considerably more challenging. This is to be expected as the subposteriors become increasingly mismatched as data size increases. However, the effect of including scaling T and n does substantively improve Algorithm 1 as it did in Section 3.3. Considering the results of fixing T and n in Figure 2d, it is clear in this regime that Algorithm 1 is degenerate. Incorporating scaling of T in Figure 2e stabilises CESS_0 and leads to a slower degradation with data size of CESS_j. However, incorporating scaling of T and n following our guidance earlier in Section 3 retains the stabilised CESS_0 and substantively improves CESS_j to a level where it could lead to a practical algorithm.

3.5 Temporal regularity of partition

In Section 3.2 in order to simplify the guidance for selecting the partition $\mathcal{P}$, we imposed a regular mesh. This allowed us to use the minimal guidance for the temporal distance between points in the partition we developed in Theorem 5, which in conjunction with the guidance already established for choosing T in Section 3.1, allowed us to indirectly specify n and in turn $\mathcal{P}$. As discussed in

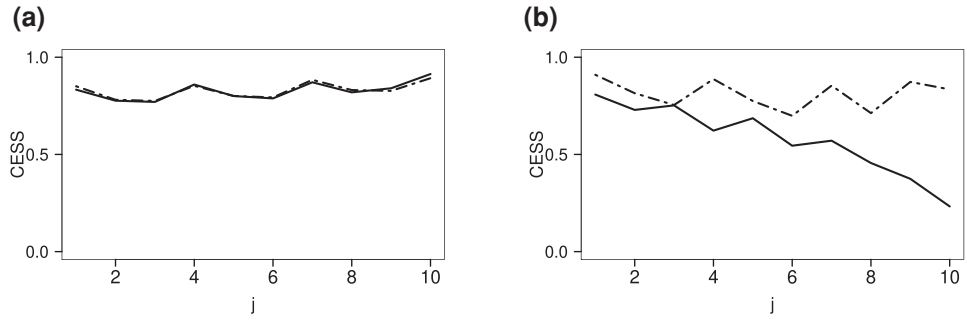

Figure 3. CESS at each iteration of Algorithm 1 ($j \in \{1, \ldots, 10\}$) under SH(λ) and SSH(γ) settings respectively. Solid lines denote results based upon selecting the unbiased estimator $\tilde{\rho}_j := \tilde{\rho}_j^{(a)}$. Dotted lines the unbiased estimator $\tilde{\rho}_j := \tilde{\rho}_j^{(b)}$. (a) SH($\lambda$) setting. $f_c(x) = \mathcal{N}(0, C\sigma^2)$. (b) SSH($\gamma$) setting. $f_c(x) = \mathcal{N}(\mu_c, C\sigma^2)$, $\mu_c = \pm 0.25$.

Section 3.2, there may be some advantage of using an irregular mesh (in which the temporal distance between points in the partition decreases as $T \uparrow n$). In this section, we investigate the impact of using a regular mesh on CESS_j ($j \in \{1, \ldots, n\}$) as a function of the iteration of Algorithm 1.

To investigate temporal regularity we revisit the idealised examples of the SH(λ) and SSH(γ) settings we introduced in Sections 3.3 and 3.4, respectively. For both settings we consider a data size of $m = 1,000$ distributed across $C = 2$ subposteriors, and specify a temporal horizon of T as Remark 2 and regular mesh of size $n = 10$. In implementing BF we use a particle set of size $N = 10,000$, and consider the use of two variants for the unbiased estimator in Step b(iii)B when implementing Algorithm 1—UE-a (Condition B.1) and UE-b (Condition B.2) of the online supplementary material, Appendix B—UE-a being a relatively straightforward construction, whereas UE-b requiring slightly more specification but in general leading to a more robust estimator as defined by the variance of the estimator. The results are presented in Figure 3.

Considering the SH(λ) setting of Figure 3a we find that CESS_j is stable across iterations of Algorithm 1, which would suggest that there is little to be gained when heterogeneity is low in having a more flexible irregular mesh. The SSH(γ) setting of Figure 3b is slightly more complicated. The results here would suggest if using the UE-a in the SSH(γ) setting there may be some advantage to using an irregular mesh to balance CESS_j across the iterations of Algorithm 1. However, in both the SH(λ) and SSH(γ) settings when using the UE-b unbiased estimator we find that CESS_j is stable. This would suggest that there is little to be gained from specifying an irregular mesh over the regular one we have imposed in Section 3.2. Choosing a good estimator for a regular mesh is far simpler than optimising an irregular mesh for a poor estimator, and so the more critical consideration is to ensure a suitable unbiased estimator is chosen—a full discussion of which can be found in the online supplementary material, Appendix B.

3.6 Impact of using approximate subposteriors

In typical settings we will not be able to simulate i.i.d. realisations from each subposterior. Instead it is more realistic to assume we have access to realisations from an *approximation* of each subposterior: for instance, if we are splitting the data across C cores in order to implement a Monte Carlo algorithm in a more scalable fashion. In this subsection, we analyse the impact of using approximate subposteriors.

More formally, if we denote φ_c as the normalised cth subposterior (in particular, we have $\varphi_c(\boldsymbol{x}_c) := f_c(\boldsymbol{x}_c)/\|f_c\|_1$ where $\|f_c\|_1 = \int |f_c(\boldsymbol{x}^{(c)})| \, \mathrm{d}\boldsymbol{x}^{(c)}$ is the normalising constant of f_c), and let $\varphi_c^{(K)}$ be an approximation of φ_c obtained using a Monte Carlo sample of size K, then we want to show that substituting $\varphi_c^{(K)}$ for φ_c in Algorithm 1 (for large enough K) would result in output $\boldsymbol{y}$ being arbitrarily close to the target fusion density f. This is presented in Theorem 6.

Note that we naturally assume that $\|\varphi_c - \varphi_c^{(K)}\|_1 \to 0$ as $K \to \infty$, and our modification to Algorithm 1 results in instead proposing $\vec{\boldsymbol{x}}_0$ and $\boldsymbol{y}$ from the density

$$g^{(K)}(\vec{\boldsymbol{x}}_0, \boldsymbol{y}) : \propto \prod_{c=1}^{C} [\varphi_c^{(K)}(\boldsymbol{x}_0^{(c)})] \cdot \mathcal{N}(\boldsymbol{y}; \vec{\boldsymbol{M}}_1, \boldsymbol{V}_1) \cdot \prod_{j=0}^{1} \rho_j, \tag{23}$$

where for notational simplicity we take $n = 1$. For general values of n, the proof is similar.

Theorem 6. Suppose that for $\epsilon > 0$ there exists a K_0 such that for $K > K_0$

$$\|\varphi_c - \varphi_c^{(K)}\|_1 \leq \epsilon \tag{24}$$

for all $c = 1, \ldots, C$. Then for any $\varepsilon^* > 0$, we can find K' such that when $K > K'$ we have

$$\int \left| \int (g(\vec{\boldsymbol{x}}_0, \boldsymbol{y}) - g^{(K)}(\vec{\boldsymbol{x}}_0, \boldsymbol{y})) \, \mathrm{d}\vec{\boldsymbol{x}}_0 \right| \mathrm{d}\boldsymbol{y} \leq \epsilon^*. \tag{25}$$

Proof. See online supplementary material, Appendix F. □

Although Theorem 6 addresses the use of approximations to the subposteriors, in common with standard SMC literature (Kunsch 2005) we assume we have access to i.i.d. realisations from the approximate subposteriors to initialise the particle set. If the approximate $\varphi_c^{(K)}$ for $c \in \{1, \ldots, C\}$ are obtained by some Monte Carlo approaches (say, MCMC), then some care has to be taken if the approximate subposterior samples are serially correlated. Analysing theoretically such approximations is challenging, although a pragmatic solution would be to either thin the MCMC output for each subposterior, or randomly sample the MCMC trajectories.

3.7 Practical implementational considerations

As motivated in the introduction, the primary contribution of this paper is to develop a practical SMC approach for inference in the *fusion problem* (simulating from Equation (1)). The methodological development of Section 2, and the practical guidance of Sections 3.1 and 3.2, have been developed to this end. However, in some particular settings where this methodology is applied it is likely there will be a number of additional specific constraints that necessitate careful implementation, or some modification, of Algorithm 1. For instance, *latency* in communication between cores may be of particular concern, or in applications where there is a large amount of data on each individual subposterior the computational efficiency of some quantities in Algorithm 1 may need consideration. In this section, we highlight some aspects and minor (nonstandard) modifications of the methodology we have developed which may be useful for practitioners.

For the purposes of clarity for the primary contributions of this paper, the methodology and examples given elsewhere in the paper do not exploit the modifications we present below. We discuss other more substantial possible directions for the practical development of the BF methodology in the conclusions. We consider the possible modifications to BF grouped into the constituent elements of Algorithm 1: Initialisation; Propagation of the particle set; and, Computing importance weights; and, normalisation and resampling of the particle set. This is presented in Sections 3.7.1–3.7.3, respectively.

There is a growing literature on implementing SMC approaches in-parallel in distributed environments (see, e.g., Doucet & Lee 2018, Sec. 7.5.3). This includes distributed resampling methodologies (Lee & Whiteley 2016; Lee et al. 2010; Murray et al. 2016), and methodological adaptations such as *distributed particle filters* (Bolic et al. 2005; Heine & Whiteley 2017), and the *island particle filter* (Vergé et al. 2015). Note that typical SMC approaches in this area distribute the *particle set* across cores, and so fuller consideration of using such approaches for BF in parallel would need to be considered. Aspects of this subsection may be useful to developing BF in this direction, however full consideration of this is beyond the scope of this paper and is instead discussed in the conclusions.

3.7.1 Initialising the particle set

In the initialisation step of BF (Algorithm 1 Step a(ii)A) we propose $\vec{\boldsymbol{x}}_0 := \boldsymbol{x}_0^{(1:C)}$ where for $c \in \{1, \ldots, C\}$, $\boldsymbol{x}_0^{(c)} \sim f_c$. Composing $\vec{\boldsymbol{x}}_0$ requires communication between the cores, and $\vec{\boldsymbol{x}}_0$ requires further communication back to the cores for the computation of the proposal importance

weight, $\rho_0(\vec{x}_0)$. Although $\rho_0(\vec{x}_0)$ can be trivially decomposed into a product of C terms corresponding to the contribution from each core separately Equation (5), computing $\rho_0(\vec{x}_0)$ still requires a third communication between cores during initialisation. In settings where *latency* is an issue, this repeated communication is undesirable. In this setting, one could attempt to improve the quality of the proposals made on each core (while avoiding any additional communication), and reduce the level of communication.

If we choose some $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^d$ (e.g., by performing a single pre-processing step and choosing $\tilde{\boldsymbol{\theta}}$ to be the weighted average of the approximate modes of each subposterior), we can modify the proposal distribution for the initial draw from each core to be

$$\tilde{f}_c(\boldsymbol{x}_0^{(c)}) \propto \exp\left\{-\frac{\|\boldsymbol{x}_0^{(c)} - \tilde{\boldsymbol{\theta}}\|^2}{2T}\right\} \cdot f_c(\boldsymbol{x}_0^{(c)}), \tag{26}$$

compensating for this modification by replacing ρ_0 within Algorithm 1 with

$$\tilde{\varrho}_0(\vec{\boldsymbol{x}}_0) := \exp\left\{\frac{\|\bar{\boldsymbol{x}}_0 - \tilde{\boldsymbol{\theta}}\|^2}{2T/C}\right\}, \quad \text{where } \bar{\boldsymbol{x}}_0 = C^{-1}\sum_{c=1}^{C} \boldsymbol{x}_0^{(c)}. \tag{27}$$

The validity of these modifications can be established by noting that, $\tilde{\varrho}_0(\vec{\boldsymbol{x}}_0) \cdot \prod_{c=1}^{C} \tilde{f}_c(\boldsymbol{x}_0^{(c)}) \propto \rho_0(\vec{\boldsymbol{x}}_0) \cdot \prod_{c=1}^{C} f_c(\boldsymbol{x}_0^{(c)})$, and recalling that re-normalisation within Algorithm 1 removes the need to compute the constant of proportionality for $\tilde{\varrho}_0$.

Noting that it is possible to sample from Equation (26) on each core in isolation by rejection sampling (using f_c as a proposal), then this can be done by each core in parallel in advance of initialising the algorithm, and will lead to improved proposal quality. Furthermore, note that computation of the proposal importance weight, $\tilde{\varrho}_0(\vec{\boldsymbol{x}}_0)$ in Equation (27), *does not* require further communication by the cores. In particular, we have removed two of the three communications required in the original formulation of the initialisation of BF. This simple modification to the BF algorithm is presented in Algorithm 2.

Algorithm 2 Modified Initialisation (in place of Algorithm 1 Step (aii))

(aii) For i in 1 to N,

A. $\vec{\boldsymbol{x}}_{0,i}$: For c in 1 to C, simulate $\boldsymbol{x}_{0,i}^{(c)} \sim \tilde{f}_c$. Set $\vec{\boldsymbol{x}}_{0,i} := \boldsymbol{x}_{0,i}^{(1:C)}$.

B. $w'_{0,i}$: Compute un-normalised weight $w'_{0,i} = \tilde{\varrho}_0(\vec{\boldsymbol{x}}_{0,i})$, as per Equation (27).

3.7.2 *Propagation of the particle set*

Considering the iterative propagation of the particle set in Algorithm 1 Step b(iii)A, note that for each particle we need to compute $\vec{\boldsymbol{M}}_j$ and $\boldsymbol{V}_j$, from Equations (8) and (9). In particular, communication between the cores is required as the computation of $\vec{\boldsymbol{M}}_j$ and $\boldsymbol{V}_j$ requires the temporal position of every trajectory over all cores. Upon propagation further communication is required in order to compute the updated importance weight of the particle in Algorithm 1 Step b(iii)B. This is clearly inefficient: we would like to minimise the number and size of communications.

It would be preferable to propagate $\vec{\boldsymbol{x}}_{j-1}$ to $\vec{\boldsymbol{x}}_j$ by considering the separate propagation of each of the C parallel processes which compose $\vec{\boldsymbol{x}}_{j-1}$, namely $\boldsymbol{x}_{j-1}^{(c)}$ $c \in \{1, \ldots, C\}$. This can be achieved by means of exploiting Corollary 2:

Corollary 2. Simulating $\vec{\boldsymbol{x}}_j \sim \mathcal{N}(\vec{\boldsymbol{M}}_j, \boldsymbol{V}_j)$, the required transition from $\vec{\boldsymbol{x}}_{j-1}$ to $\vec{\boldsymbol{x}}_j$ in Algorithm 1, can be expressed as

$$\boldsymbol{x}_j^{(c)} = \left(\frac{\Delta_j^2}{C(T - t_{j-1})}\right)^{1/2} \boldsymbol{\xi}_j + \left(\frac{T - t_j}{T - t_{j-1}}\Delta_j\right)^{1/2} \boldsymbol{\eta}_j^{(c)} + Mjc, \tag{28}$$

where $\boldsymbol{\xi}_j$ and $\boldsymbol{\eta}_j^{(c)}$ are standard Gaussian vectors, and $\boldsymbol{M}_j^{(c)}$ is the subvector of $\vec{\boldsymbol{M}}_j$ corresponding to the cth component adopting the notation in Equation (12).

Proof. See online supplementary material, Appendix D. □

In particular, note that the interaction with the other trajectories solely appears in the mean of the trajectories at the previous iteration ($\bar{\boldsymbol{x}}_{j-1}$). Computation of $\bar{\boldsymbol{x}}_{j-1}$ can be conducted at the previous iteration of Algorithm 1 at the same time as the trajectories are communicated for composition and use in computing the importance weight—thus removing an unnecessary communication. As we already compute $\bar{\boldsymbol{x}}_{0,i}$, as required in the computation of ρ_0 in Algorithm 1 Step a(ii)B (or alternatively as required by $\tilde{\varrho}_0$ in Section 3.7.1), incorporating this into BF requires only a minor modification. This is presented in Algorithm 3.

Algorithm 3 Modified Propagation (in place of Algorithm 1 Step b(iii)A).

b(ii)A.1. For c in 1 to C, simulate $\boldsymbol{x}_{j,i}^{(c)} \mid (\bar{\boldsymbol{x}}_{j-1,i}, \boldsymbol{x}_{j-1,i}^{(c)})$ as per Equation (28).

b(ii)A.2. Set $\vec{\boldsymbol{x}}_{j,i} := \boldsymbol{x}_{j,i}^{(1:C)}$, and compute $\bar{\boldsymbol{x}}_{j,i} := \sum_{c=1}^{C} \boldsymbol{x}_{j,i}^{(c)}/C$.

3.7.3 *Updating the particle set weights*

In some settings it may not be practical to compute functionals of each subposterior (f_c, $c \in \{1, \ldots, C\}$), and so rendering the evaluation of ϕ_c, and in turn $\tilde{\rho}_j$ in Algorithm 1 Step b(iii) B, unfeasible. This may be due to a form of intractability of the subposteriors, (such as the settings considered by Andrieu and Roberts (2009)), or simply that their evaluation is computationally too expensive (such as in the large data settings considered by Pollock et al. 2020). This issue can be circumvented by means of the following corollary:

Corollary 3. The estimator

$$\tilde{\varrho}_j := \prod_{c=1}^{C} \frac{\Delta_j^{\kappa_c} \cdot e^{-\bar{U}_j^{(c)}\Delta_j}}{\kappa_c! \cdot p(\kappa_c \mid R_c)} \prod_{k_c=1}^{\kappa_c} \left(\bar{U}_j^{(c)} - \hat{\phi}_c\left(\boldsymbol{x}_{\chi_{c,k_c}}^{(c)}\right) \right),$$

where $\hat{\phi}_c$ is an unbiased estimator of ϕ_c, and $\bar{U}_j^{(c)}$ is a constant such that $\hat{\phi}_c\left(\boldsymbol{x}_t^{(c)}\right) \leq \bar{U}_j^{(c)}$ for all $\boldsymbol{x}_t^{(c)} \sim \mathbb{W}_{j,c} \mid R_c$, is an unbiased estimator of $\tilde{\rho}_j$.

Proof. Follows directly from the proof of Theorem 3 in the online supplementary material, Appendix B. □

The estimator $\tilde{\varrho}_j$ in Corollary 3 can be used as a substitute for $\tilde{\rho}_j$ in Algorithm 1 Step b(iii)B, and simulated by direct modification of Algorithm 4. To take advantage of Corollary 3 one simply has to find a *suitable* unbiased estimator of ϕ_c, which in many settings will be straightforward to construct as ϕ_c is linear in terms of $\nabla \log f_c(\boldsymbol{x})$ and $\Delta \log f_c(\boldsymbol{x})$. To find a suitable unbiased estimator to use in place of $\tilde{\rho}_j$, it is important to recognise the penalty for its introduction. In particular, introducing the estimator $\tilde{\varrho}_j$ will (typically) increase the variance of the estimator, which will manifest itself in the variance of the particle set weights in Algorithm 1. To control this we will (typically) require a heavier tailed choice of discrete distribution p in Corollary 3. An extensive discussion on finding low variance estimators of the type can be found in the online supplementary material, Appendix B. A concrete application of Corollary 3 can be found in the online supplementary material, Appendix E.

4 Examples

In this section, we apply our BF approach of Algorithm 1 to data obtained from a population survey (Section 4.1) and road accident data (Section 4.2). In both examples we compare the performance BF with CMC (Scott et al. 2016), DPMC (Xue & Liang 2019), and the WRS (X. Wang & Dunson 2013). To construct a fair benchmark for each method we construct a *benchmark distribution* by using the BayesLogit R package (Polya-Gamma Gibbs sampler, Choi & Hobert 2013) to sample from the fusion density directly. We then define and compute the integrated absolute

distance (IAD) for each method with respect to the benchmark distribution as follows:

$$\text{IAD} := \frac{1}{d}\sum_{j=1}^{d} \int |\hat{f}(\theta_j) - f(\theta_j)| \, \mathrm{d}\theta_j \in [0, 2], \tag{29}$$

where f is the benchmark distribution and $\hat{f}$ is the distribution obtained from the methodology employed, both computed using a kernel density estimate.

To better understand the performance and scaling of BF and competitor approaches, in Appendix G of the online supplementary material we further consider a range of idealised settings. Online supplementary material, Appendix G.1 concentrates on a comparison with the *exact* MCF approach, whereas online supplementary material, Appendix G.2 focuses on the approximate CMC, DPMC, and WRS approaches.

4.1 US Census Bureau population surveys

In this example we applied BF to the 1994 and 1995 US Census Bureau population surveys, obtained from Bache and Lichman (2013), and of size $m = 199,523$, and investigated the effect of education on gross income. We took *gross income* as our observed data (y_i), treating it as a binary taking a value of one if income was greater than \$50,000. An income in excess of \$50,000 is moderately rare with only 12,382 individuals exceeding this threshold (which represents approximately 6% of the data). In addition to the intercept, we extracted three further *education* covariates indicating educational stages attained by the individual (each of which were binary). We then fitted the following logistic regression model with prior distribution $\mathcal{N}(0, 10\mathbf{I}_4)$:

$$y_i = \begin{cases} 1 & \text{with probability } \frac{\exp\{z_i^T\boldsymbol{\beta}\}}{1+\exp\{z_i^T\boldsymbol{\beta}\}}, \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

For this data set, we considered recovering the benchmark distribution by unifying subposteriors across an increasing number of cores $C \in \{10, 20, 40\}$. To construct subposteriors we distributed the data among the available C cores, and fit the logistic regression model of Equation (30) to each using a fractional prior. To emulate more realistic settings, we did not allocate the data randomly among the C cores. In our allocation there was an extremely unbalanced allocation of data, with three of the cores containing about 99% of the individuals earning in excess of \$50,000.

BF was implemented with a particle set of size $N = 30,000$, and following the guidance of Section 3. CMC, DPMC, and WRS were implemented following the guidance suggested by the authors. The marginal densities are presented in Figure 4 for the $C = 40$ setting, together with their IAD, and computational costs for the range of cores considered.

For this data set CMC and DPMC are poorly suited as they rely on the convergence of the posterior to a Gaussian distribution, and this is evidenced here with them capturing neither the marginals of the benchmark distribution, or showing any robustness with respect to the numbers of cores. Considering the marginals in Figure 4, the WRS substantially improves upon CMC (only struggling with β_2 and β_3). However, for slightly more computational expenditure (Figure 4f), BF substantially improves upon IAD over the WRS (Figure 4e), and also appears to show robustness with increasing C.

4.2 UK road accidents

In this example we considered the 'Road Safety Data' data set published by the Department for Transport of the UK government (gov.uk 2019). It comprises road accident data set from 2011 to 2018, and in total is of size $m = 1,111,320$. We treated our observation for each record to be binary taking a value of one if a *severe* accident was recorded. In total in the full data set there were 13,358 such severe accidents. We selected a number of covariates to investigate what effect they have on accident severity: in addition to an intercept, we considered *road speed limit*, *lighting condition* (which we treated as binary taking a value of one if lighting was *good*, and zero if lighting was *poor*), and *weather condition* (binary, taking one if *good* and zero if *poor*). The logistic

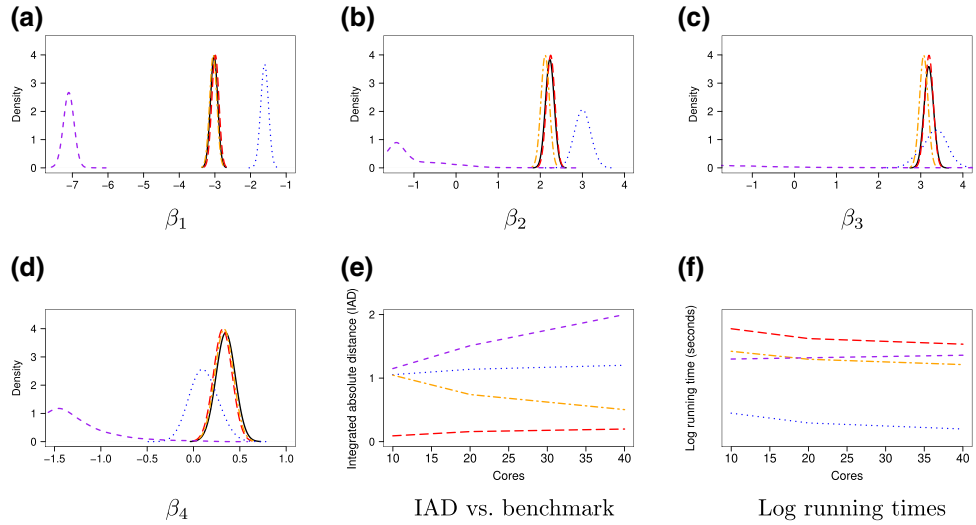

Figure 4. BF and competing algorithms applied to the US Census Bureau population survey data set of Section 4.1. (a–d) shows marginal density estimates for β_1–β_4 respectively, with the solid lines denoting the benchmark fitted target distribution. (e, f) shows the performance of BF in terms of IAD and computational cost with respect increasing numbers of cores. Long dashed lines denote BF. Dotted lines denote CMC. Dotted and dashed lines denote WRS. Short dashed lines denote DPMC.

regression model of Equation (30) was fit to the data set, again with a $\mathcal{N}(0, 10\mathbf{I}_4)$ prior distribution.

We again considered recovering the benchmark distribution by unifying subposteriors across an increasing number of cores $C \in \{10, 20, 40\}$. The subposteriors were obtained with the allocation of data to each core being in temporal order. We contrasted BF with a particle set of size $N = 30,000$, with fair implementations of CMC, DPMC, and the WRS. Marginal densities for the $C = 40$ setting are presented in Figure 5, together with IAD and overall computational cost for the range of cores considered. The results are in keeping with those of Section 4.1. CMC and DPMC perform extremely poorly, and for a modest increase in computational budget BF substantially improves upon WRS.

5 Conclusions

In this paper, we have developed a theoretical framework, and scalable SMC methodology, for unifying distributed statistical analyses on shared parameters from multiple sources (which we term *subposteriors*) into a single coherent inference. The work significantly extends the theoretical underpinning, and practical limitations, of the *exact* MCF approach of Dai et al. (2019). MCF is a rejection-sampling-based approach for sampling from Equation (1) without approximation. However, it lacks scalability with respect to the number of subposteriors to be unified, and robustness with subposterior dis-similarity. This is addressed by our BF approach (Algorithm 1), which both recovers the correct target distribution and is computationally competitive with leading approximate schemes. Fundamental to our BF approach is the construction of the fusion measure via an SMC procedure driven by the SDE in Equation (2).

In addition to the theoretical and methodological development of BF presented in Section 2, in Section 3 we provide concrete theory and guidance on how to choose the free parameters of Algorithm 1 to ensure robustness with increasing numbers of subposteriors, and subposterior dissimilarity. In Section 4, we apply BF to the 'US Census Bureau population surveys' data set and 'UK road accidents' data set, contrasting it with competing approximate methodologies. Further extensive numerical studies in challenging idealised scenarios are given in the online Supplementary material, Appendix G to contrast the limitations of existing fusion approaches and our BF.

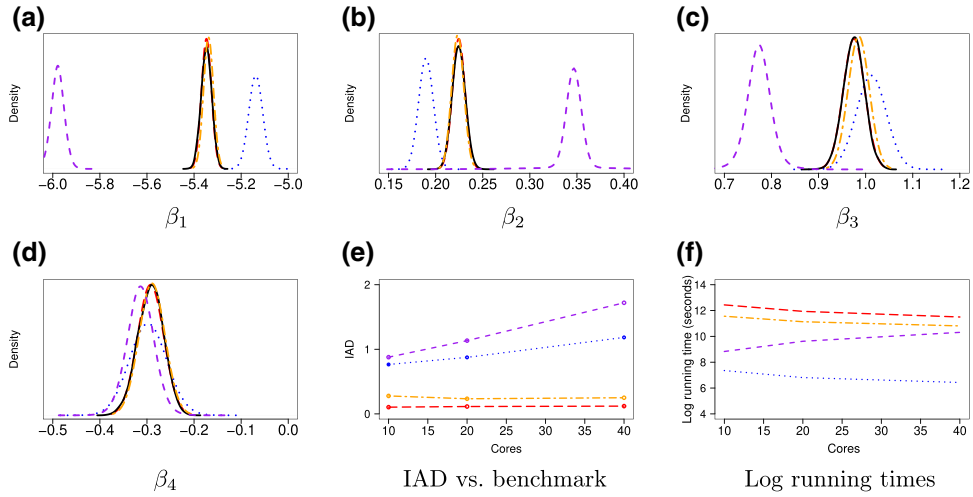

Figure 5. BF and competing algorithms applied to the UK road accident data set of Section 4.2. (a–d) shows marginal density estimates for β_1–β_4 respectively, with the solid lines denoting the benchmark fitted target distribution. (e, f) shows the performance of BF in terms of IAD and computational cost with respect increasing numbers of cores. Long dashed lines denote BF. Dotted lines denote CMC. Dotted and dashed lines denote the WRS. Short dashed lines denote DPMC results.

One of the key advantages of BF is that it is underpinned methodologically by SMC, which allows us to leverage many of the existing theoretical results and methodology found in that literature. As is typical within SMC it is desirable to attempt to minimise the discrepancy between the sequence of proposal and target distributions. In our setting this entails ensuring the propagated temporal marginal of g in Equation (13) (say g_{j-1}^{N}), is well-matched with the following temporal marginal of g (say g_j^{N}). Although not emphasised within the main text, there is clear scope to improve BF in this sense by modifying the diffusion theory presented in its development (see online supplementary material, Appendix A), to one which better incorporates information about each subposterior (e.g., this could be knowledge of the volume of data on each core). In the spirit of this, in the recent work of Chan et al. (2021) the covariance structure of each subposterior is estimated and the C spaces are transformed accordingly, leading to Brownian proposals being more attuned to the target distribution. This approach could in principle be modified to our BF setting, and would amount to modifying the *Fusion measure* in Equation (3) (in which the transition densities for each subposterior are that of a Langevin diffusion with unit volatility), to one with volatility which matches the covariance structure of its respective subposterior.

We have provided considerable practical guidance in Sections 3.6 and 3.7. In Section 3.6, we addressed the realistic scenario where we have only sample approximations of each subposterior. In Section 3.7, we rendered many aspects of BF which are nonstandard due to the particularities of the fusion problem into standard SMC structures. A truly parallel implementation of BF is a very attractive prospect for future development. As discussed in Section 3.7, although SMC is inherently well-suited to parallel implementation in distributed environments (Doucet & Lee 2018), in the fusion setting the natural direct interpretation of BF would be to consider the subposteriors (and associated data) as being distributed across cores, but the particle set to be shared across all cores. This is not the setting typically addressed by distributed SMC literature, and raises interesting challenges which require further innovation to be resolved.

A number of other methodological directions for BF are possible. As presented in Sections 2 and 3, the C subposteriors are unified together in a 'fork-and-join' manner. An alternative would be to unify the subposteriors in stages gradually by constructing a tree to perform the operation hierarchically, for instance by exploiting 'divide-and-conquer' SMC theory and methodologies such as that of Lindsten et al. (2017). This has been considered in the MCF setting in Chan et al. (2021). Another direction would be to consider how approximations could be used within the methodology. Many approximate approaches tackling the fusion problem are highly

computationally efficient, albeit at the expense of introducing an approximation error which can be difficult to quantify and on occasion significant. The work of A. Wang et al. (2019) constructs an explicit Monte Carlo scheme in which approximations can be readily used to develop exact Monte Carlo schemes. There is theory linking this paper with Pollock et al. (2020) and A. Wang et al. (2019), and so finding a similar approach to embedding approximations may be viable.

As discussed in Section 1, there is considerable scope for application of BF, as inference in the setting of Equation (1) arises directly and indirectly in many interesting practical settings. One interesting direction considers the use of fusion methodologies within the *Markov melding* framework of Goudie et al. (2019), in which a modular approach is taken to statistical inference where separate submodels are fit to data sources in isolation (often of varying dimensionality), and then joined. This type of application would necessitate theoretical developments to the Fusion methodologies to support subposteriors on mismatched dimensions. Another scenario where this methodological shortcoming arises would be in Bayesian hierarchical modelling for (generalised) regression models with missing covariates. Here, the missing variables could exist in different hierarchical layers (Daniels et al. 2013), and so may not be common to each subposterior. However, there appears to be some scope to addressing mismatched subposteriors within the SMC theory developed in Lindsten et al. (2017), and this may be interesting even in the case of matched subposteriors as it could plausibly make Fusion methodologies more robust to increasing dimensionality.

A number of future directions for the BF methodology are currently being pursued by the authors. One interesting avenue of research is to apply Fusion methodologies within statistical cryptography. In the simplest setting a number of trusted parties who wish to securely share their distributional information on a common parameter space and model, but would prefer not to reveal their individual level distributions, could do so by means of applying cryptography techniques and exploiting the exactness and linear contributions to computations of individual subposteriors within the Fusion approach. In a further example, the authors are investigating the application of BF for purely algorithmic reasons. One motivation for this (rather like the motivation for tempering MCMC approaches) is that the simulation of a multimodal target density could be prohibitively difficult, whereas the target density might be readily written as a product of densities with less pronounced multimodal behaviour, thus making it far more amenable to Monte Carlo sampling (see Chan et al. 2021).

Acknowledgments

We would like to thank Louis Aslett, Paul Jenkins, Yuxi Jiang, Adam Johansen, and especially Ryan Chan, for helpful discussions on aspects of the paper. We would like to thank the reviewers for their considered comments, and in particular the Associate Editor for highlighting the shortcomings of ESS. This work was supported by EPSRC grant numbers K014463, N031938, R018561, R034710, and the Alan Turing Institute. We also thank the Isaac Newton Institute for support during the programme 'Scalable inference; statistical, algorithmic, computational aspects'.

Supplementary material

Supplementary material is available online at *Journal of the Royal Statistical Society* (http://mtp.oxfordjournals.org/).

Data availability

The US Census Bureau survey dataset is available in the UCI Machine Learning Repository, at https://archive.ics.uci.edu/ml//datasets/Census-Income+%28KDD%29.

The UK Road Safety Dataset is available in the repository of the Department for Transport, UK Government, at https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data.

Conflict of interest: None declared.

References

Agarwal A., & Duchi J. (2011). Distributed delayed stochastic optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (pp. 873–881). Curran Associates Inc.

Andrieu C., & Roberts G. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, *37*, 697–725. http://doi.org/10.1214/07-AOS574

Bache K., & Lichman M. (2013). *UCI machine learning repository*. University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml/

Berger J. (1980). *Statistical decision theory and Bayesian analysis*. Springer.

Beskos A., Papaspiliopoulos O., & Roberts G. (2008). A factorisation of diffusion measure and finite sample path constructions. *Methodology and Computing in Applied Probability*, *10*, 85–104. http://doi.org/10.1007/s11009-007-9060-4

Beskos A., & Roberts G. (2005). An exact simulation of diffusions. *Annals of Applied Probability*, *15*(4), 2422–2444. http://doi.org/10.1214/105051605000000485

Bolic M., Djuric P., & Hong S. (2005). Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, *53*(7), 2442–2450. http://doi.org/10.1109/TSP.2005.849185

Buchholz A., Ahfock D., & Richardson S. (2019). Distributed computation for marginal likelihood based model choice. arXiv, arXiv:1910.04672, preprint: not peer reviewed.

Chan R., Pollock M., Johansen A., & Roberts G. (2021). Divide-and-conquer Monte Carlo Fusion. arXiv, arXiv:2110.07265, Preprint: not peer reviewed.

Choi H. M., & Hobert J. P. (2013). The Polya-Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. *Electronic Journal of Statistics*, *7*, 2054–2064. http://doi.org/10.1214/13-EJS837

Dai H., Pollock M., & Roberts G. (2019). Monte Carlo Fusion. *Journal of Applied Probability*, *56*, 174–191. http://doi.org/10.1017/jpr.2019.12

Daniels M. J., Wang C., & Marcus B. (2013). Fully Bayesian inference under ignorable missingness in the presence of auxiliary covariates. *Biometrics*, *70*, 62–72. http://doi.org/10.1111/biom.v70.1

Del Moral P. (2004). *Feynman-Kac formulae. Genealogical and interacting particle systems with applications*. Probability and Applications. Springer-Verlag New York LLC.

Doucet A., de Freitas N., & Gordon N. (2001). *Sequential Monte Carlo methods in practice* (1st ed.). Springer.

Doucet A., & Lee A. (2018). Sequential Monte Carlo methods. In M. Maathuis, M. Drton, S. Lauritzen, & M. Wainwright (Eds.), *Handbook of graphical models* (Chapter 7, pp. 165–189). CRC Press.

Elvira V., Martino L., & Robert C. (2022). Rethinking the effective sample size. *International Statistical Review, 90*, 525–550. https://doi.org/10.1111/insr.12500

Fleiss J. (1993). Review papers: The statistical basis of meta-analysis. *Statistical Methods in Medical Research*, *2*(2), 121–145. http://doi.org/10.1177/096228029300200202

Genest C., & Zidek J. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, *1*(1), 114–135.

Gordon N., Salmond J., & Smith A. (1993). A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, *140*, 107–113. http://doi.org/10.1049/ip-f-2.1993.0015

Goudie R., Presanis A., Lunn D., De Angelis D., & Wernisch L. (2019). Joining and splitting models with Markov melding. *Bayesian Analysis*, *14*(1), 81. http://doi.org/10.1214/18-BA1104

gov.uk (2019). 'Road Safety Data' dataset, Department for Transport, U.K. Government. Retrieved September 17, 2020, from https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data. Update Version: December 17, 2019.

Heine K., & Whiteley N. (2017). Fluctuations, stability and instability of a distributed particle filter with local exchange. *Stochastic Processes and their Applications*, *127*(8), 2508–2541. http://doi.org/10.1016/j.spa.2016.11.003

Jordan M., Lee J., & Yang Y. (2018). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, *114*(526), 668–681. http://doi.org/10.1080/01621459.2018.1429274

Kitagawa G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*(1), 1–25. https://doi.org/10.1080/10618600.1996.10474692

Kong A., Liu J., & Wong W. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, *89*(425), 278–288. http://doi.org/10.1080/01621459.1994.10476469

Kunsch H. R. (2005). Recursive monte carlo filters: Algorithms and theoretical analysis. *The Annals of Statistics*, *33*(5), 1983–2021. http://doi.org/10.1214/009053605000000426

Lee A., & Whiteley N. (2016). Forest resampling for distributed sequential Monte Carlo. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *9*(4), 230–248. http://doi.org/10.1002/sam.2016.9.issue-4

Lee A., Yau C., Giles M., Doucet A., & Holmes C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of Computational and Graphical Statistics*, *19*(4), 769–789. http://doi.org/10.1198/jcgs.2010.10039

Lindsten F., Johansen A., Naesseth C., Kirkpatrick B., Schön T., Aston J., & Bouchard-Côté A. (2017). Divide-and-conquer with sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, *26*(2), 445–458. http://doi.org/10.1080/10618600.2016.1237363

Minsker S., Srivastava S., Lin L., & Dunson D. (2014). Scalable and robust Bayesian inference via the median posterior. In E. Xing & T. Jebara (Eds.), Proceedings of the 31st international conference on *Machine Learning* (Vol. 32, pp. 1656–1664). PMLR.

Murray L., Lee A., & Jacob P. (2016). Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, *25*(3), 789–805. http://doi.org/10.1080/10618600.2015.1062015

Neiswanger W., Wang C., & Xing E. (2013). Asymptotically exact, embarrassingly parallel MCMC. arXiv, arXiv:1311.4780, preprint: not peer reviewed.

Pollock M., Fearnhead P., Johansen A., & Roberts G. (2020). Quasi-stationary Monte Carlo methods and the ScaLE algorithm (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *82*, 1–59. http://doi.org/10.1111/rssb.v82.5

Rendell L., Johansen A., Lee A., & Whiteley N. (2021). Global consensus Monte Carlo. *Journal of Computational and Graphical Statistics, 30*, 249–259. https://doi.org/10.1080/10618600.2020.1811105

Rogers L., & Williams D. (2000). *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*. Cambridge University Press.

Scott S. (2017). Comparing consensus Monte Carlo strategies for distributed Bayesian computation. *Brazilian Journal of Probability and Statistics*, *31*(4), 668–685. http://doi.org/10.1214/17-BJPS365

Scott S., Blocker A., Bonassi F., Chipman H., George E., & McCulloch R. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, *11*(2), 78–88. http://doi.org/10.1080/17509653.2016.1142191

Smith T., Spiegelhalter D., & Thomas A. (1995). Bayesian approaches to random-effects meta-analysis: A comparative study. *Statistics in Medicine*, *14*(24), 2685–2699. http://doi.org/10.1002/(ISSN)1097-0258

Srivastava S., Cevher V., Tan-Dinh Q., & Dunson D. (2015). Wasp: Scalable Bayes via barycenters of subset posteriors. In G. Lebanon, & S. V. N. Vishwanathan (Eds.), Proceedings of the eighteenth international conference on *Artificial Intelligence and Statistics, 38*, 912–920.

Stamatakis A., & Aberer A. (2013). Novel Parallelization Schemes for Large-Scale Likelihood-based Phylogenetic Inference. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing* (pp. 1195–1204).

Vergé C., Dubarry C., Moral P., & Moulines E., 2015). On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, *25*, 243–260. http://doi.org/10.1007/s11222-013-9429-x

Vono M., Dobigeon N., & Chainais P. (2019). Split-and-augmented Gibbs sampler-application to large-scale inference problems. *IEEE Transactions on Signal Processing*, *67*(6), 1648–1661. http://doi.org/10.1109/TSP.2019.2894825

Wang A., Pollock M., Roberts G., & Steinsaltz D. (2021). Regeneration-enriched Markov processes with application to Monte Carlo. *Annals of Applied Probabilty, 31*, 703–735. https://doi.org/10.1214/20-AAP1602

Wang X., & Dunson D. (2013). Parallelizing MCMC via Weierstrass sampler. arXiv, arXiv:1312.4605, preprint: not peer reviewed.

Wang X., Guo F., Heller K., & Dunson D. (2015). Parallelizing MCMC with random partition trees. In C. Cortes, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), Proceedings of the 28th international conference on *Neural Information Processing Systems* (pp. 451–459).

Xu M., Lakshminarayanan B., Teh Y., Zhu J., & Zhang B. (2014). Distributed Bayesian posterior sampling via moment sharing. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K.Q. Weinberger (Eds.), *Advances in neural information processing systems* (pp. 3356–3364).

Xue J., & Liang F. (2019). Double-parallel Monte Carlo for Bayesian analysis of big data. *Statistics and Computing*, *29*, 23–32. http://doi.org/10.1007/s11222-017-9791-1

Yıldırım S., & Ermiş B. (2019). Exact MCMC with differentially private moves. *Statistics and Computing*, *29*(5), 947–963. http://doi.org/10.1007/s11222-018-9847-x

Zhou Y., Johansen A., & Aston J. (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*, *25*(3), 701–726. http://doi.org/10.1080/10618600.2015.1060885