On the Bias of Variational Resampling

Axel Finke
Newcastle University, UK

Oskar Kviman
KTH, Stockholm, Sweden

Nicola Branchini
University of Warwick, UK

Víctor Elvira
University of Edinburgh, UK

Abstract

Variational resampling (VR) is a method for deterministically resampling the N particles in sequential Monte Carlo (SMC) algorithms (also known as particle filters), by minimising the Kullback–Leibler divergence from the empirical measure of the N weighted original particles to the empirical measure of M unweighted resampled particles. The combination of VR with a weight transformation (called smoothing weights) has shown to often yield a smaller mean-square error (MSE) than standard resampling schemes in the literature. However, its bias has never been investigated. In this paper, we first show that VR incurs a weighting bias and a truncation bias. We then propose a mechanism to alleviate the weighting bias through an uneven weighting of the resampled particles. We also show that the truncation bias implies that the particle approximation of the target distribution is restricted to a region in which the unnormalised weights are larger than some threshold with high probability. We prove that this probability approaches 1 if $M = \mathrm{O}(N)$ as $N \to \infty$. Finally, we empirically illustrate that the smaller MSE of VR observed in the literature may be attributable to an underestimation of uncertainty caused by the use of the smoothing weights.

1 INTRODUCTION

Sequential Monte Carlo (SMC) methods (Chopin and Papaspiliopoulos, 2020) approximate probability distributions with a population of weighted samples ('particles') and have become indispensable across many domains since their popularisation in (Gordon et al., 1993), including epidemic tracking (Storvik et al., 2023), option pricing in finance (Creal, 2012), and Bayesian phylogenetic inference (Bouchard-Côté et al., 2012; Koptagel et al., 2022).

Unfortunately, weighted samples naturally suffer from degeneracy, with only a few carrying most of the probability mass. To mitigate this issue, SMC methods employ a resampling step which prunes low-weight particles and replicating high-weight ones. Classical resampling schemes, such as *multinomial* (Rubin, 1987), *stratified* (Kitagawa, 1996) or *systematic* (Carpenter et al., 1999), are designed to yield unbiased empirical approximations of the normalised weighted measure (Douc et al., 2005). However, drawbacks of existing resampling schemes have motivated alternative formulations (see, e.g., Li et al., 2015, for a review). Building on this, Kviman et al. (2024) recently introduced a *variational resampling (VR)* scheme. It casts the resampling problem as an optimisation task with some specific design choices, opening the door to other schemes. The analysis of VR – especially a characterisation of its bias – is the focus of this work.

Sampling–importance resampling. To alleviate some of the notational burden which SMC methods necessarily incur, we will postpone these to Section 3 and discuss VR in the context of approximating a single *target distribution*:

$$\pi(\mathrm{d}x) \coloneqq \frac{1}{\mathcal{Z}}\mu(\mathrm{d}x)G(x),$$

where μ is some probability measure on $\mathsf{X} = \mathbb{R}^D$, $G\colon \mathsf{X} \to (0,\infty)$ is some non-negative μ-integrable *potential* function, which can be evaluated pointwise, and $\mathcal{Z} \coloneqq \mu(G) \in (0,\infty)$ is the (often intractable) *normalising constant.* Here, for any probability measure η on X and suitable test function $f\colon \mathsf{X} \to \mathbb{R}$, we define $\eta(f) \coloneqq \int_{\mathsf{X}} f(x)\eta(\mathrm{d}x)$. We will assume that $\mu(G^2) < \infty$ as it is common in the literature. Throughout this work, we use the same symbols (e.g., π and μ) to de-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

note probability measures and their densities w.r.t. a suitable dominating measure $\mathrm{d}x$ (typically a suitable version of the Lebesgue measure). For some suitable set A, the symbol $\mathbb{I}_A$ denotes the indicator function, i.e., $\mathbb{I}_A(x) = 1$ if $x \in A$ and $\mathbb{I}_A(x) = 0$ if $x \notin A$; δ_x denotes the point measure at x, i.e., $\delta_x(A) \coloneqq \mathbb{I}_A(x)$; if $A = \{a\}$ is a singleton we write $\delta_x(a) \coloneqq \delta_x(\{a\})$.

As an example, π could be the posterior distribution in Bayesian statistics. In this case, $\mu(\mathrm{d}x) = \mathbb{P}(X \in \mathrm{d}x)$ is the prior distribution and the data Y are generated from a conditional distribution $\mathbb{P}(Y \in \mathrm{d}y|X = x) = G(x, y)\mathrm{d}y$. Treating a realisation of y of Y as fixed, we may then drop y from the notation to write instead $G(x) \coloneqq G(x, y)$. The normalising constant is then simply the likelihood: $\mathcal{Z} = \mathbb{E}[G(X, y)] = \mathbb{E}[G(X)]$.

Our goal is to approximate expectations $\pi(f)$ for some π-integrable *test function* $f\colon \mathsf{X} \to \mathbb{R}$. We use *sampling–importance resampling (SIR)* (Rubin, 1987, 1988). The method is summarised in Algorithm 1, where $M, N \in \mathbb{N}$ are specified by the user.

Algorithm 1 (SIR – Rubin 1987).

1. Sample $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \mu$ and approximate π by
$$\hat{\pi}^N(\mathrm{d}x) \coloneqq \sum_{n=1}^{N} W^n \delta_{X^n}(\mathrm{d}x),$$
where, for $n \in [N]$, $W^n \coloneqq G(X^n)/\sum_{l=1}^{N} G(X^l)$;
2. sample $A^{1:M} \sim \rho(\,\cdot\,; W^{1:N})$ and approximate π by
$$\tilde{\pi}^M(\mathrm{d}x) \coloneqq \sum_{m=1}^{M} \widetilde{W}^m \delta_{\widetilde{X}^m}(\mathrm{d}x),$$
where $\widetilde{X}^m \coloneqq X^{A^m}$, for $m \in [M]$.

Step 1 of Algorithm 1 performs *self-normalised importance sampling* to obtain weighted sample, from which an approximation $\hat{\pi}^N$ of π can be formed. $(X^{1:N}, W^{1:N})$. Here, $X^{1:N} \in \mathsf{X}^N$ and $W^{1:N} \in \{V^{1:N} \in [0,1]^N \mid \sum_{n=1}^{N} V^n = 1\} =: \mathsf{S}_N$ are the Monte Carlo samples (particles) and self-normalised importance weights.

Step 2 of Algorithm 1 performs *resampling* to obtain a modified weighted sample $(\widetilde{X}^{1:M}, \widetilde{W}^{1:M})$. That is, resampling first generates *parent indices* $A^{1:M}$ from a *resampling distribution,* i.e., from a joint distribution $\rho(\,\cdot\,; W^{1:N})$ on $[N]^M$ which generally depends on $W^{1:N}$. The Monte Carlo sample $\widetilde{X}^m \coloneqq X^{A^m}$ is then the mth *resampled particle* and $\widetilde{W}^m \geq 0$ is the corresponding *resampled weight* whose specification depends on the choice of resampling distribution ρ. For instance, multinomial, stratified and systematic resampling for all set $\widetilde{W}^m \coloneqq 1/M$, for $m \in [M]$. However, we stress that such homogeneous resampled weights are not necessary. For instance, the *optimal finite-state* resampling scheme Fearnhead (1998); Fearnhead and Clifford (2003) and the related *partial* resampling scheme (Martino et al., 2016) lead to non-uniform resampled weights.

Variational resampling. Recently, Kviman et al. (2024) introduced a deterministic resampling scheme in which $\rho(\,\cdot\,; W^{1:N})$ is a point mass implicitly specified through Algorithm 2, where, for $u > 0$, the function $C_u\colon [0, \infty) \to (0, u]$ is given by

$$k \mapsto C_u(k) = \begin{cases} \dfrac{k^k u}{(k+1)^{k+1}}, & \text{if } k > 0, \\ u, & \text{if } k = 0. \end{cases} \tag{1}$$

Algorithm 2 (VR – Kviman et al. 2024). Given a weighted sample $(X^{1:N}, W^{1:N})$, and $K^1 = \ldots = K^N \coloneqq 0$,

1. for $m = 1, \ldots, M$,
   (a) compute $A^m \in \arg\max_{n \in [N]} C_{W^n}(K^n)$,
   (b) set $K^{A^m} \leftarrow K^{A^m} + 1$;
2. set $\widetilde{X}^m \coloneqq X^{A^m}$ and $\widetilde{W}^m \coloneqq 1/M$, for $m \in [M]$.

In Algorithm 2, K^n is the total number of descendants of Particle n allocated by resampling. Thus, at the start of the algorithm, $K^1 = ... = K^N = 0$. Then, for each resampled particle $m = 1, ..., M$, Step 1a specifies an ancestor, say n (i.e., $A^m = n$); subsequently, Step 1b increments the offspring count K^n by 1.

Kviman et al. (2024) termed their method *variational resampling (VR)* because it minimises the Kullback–Leibler (KL) divergence from $\hat{\pi}^N$ to $\tilde{\pi}^M$ under the assumption that the latter sets set $\widetilde{W}^m \coloneqq 1/M$, for all $m \in [M]$. A proof is given in Kviman et al. (2024). For completeness, we also include a self-contained proof (Proposition 3) in Supplementary Appendix A.

VR (potentially combined with a modified set of weights – to be explained in Section 3.1 below) exhibited a lower mean-square error (MSE) and higher marginal-likelihood estimates than several popular low-variance resampling schemes in the numerical experiments in Kviman et al. (2024). The authors also noted that VR concentrates the distribution of the resampled particles "around the mode of the posterior density, in a typical ELBO-based VI fashion". However, no further analysis of VR was undertaken:

> "[The] characterization of the biasedness of [VR] (stemming from their deterministic

nature) was outside the scope of this work. Nonetheless, establishing theoretical guarantees is an important future-work direction." (Kviman et al., 2024)

Contributions. In this work, we analyse the bias of VR. Our contributions in this context are as follows.

1. We explain that VR suffers from a *weighting bias,* i.e., from a bias due to the fact that the expected number of offspring of a particle is not proportional to is weight. We then propose a novel 'weighted' version of VR (Algorithm 3) which removes this bias.

2. We prove that VR suffers from an additional *truncation bias.* Specifically, Proposition 2 proves that with high probability (that depends on N and M), VR results in an approximation $\tilde{\pi}^M$ of π which is truncated to a region in which the potential function exceeds some threshold $\beta > 0$. Corollary 1 further proves that this bias does not vanish asymptotically in N. That is, the probability of this truncation approaches 1 as $N \to \infty$ as long as $M = \mathrm{O}(N)$. Though numerical experiments hint that the truncation bias can be made to vanish for fixed N by letting $M \to \infty$.

3. We illustrate, empirically, that the potentially smaller MSE induced by VR observed in Kviman et al. (2024) may be attributable to an underestimation of uncertainty.

Beyond VR, our work provides the following insights into deterministic resampling and the evaluation of SMC algorithms more broadly:

4. A general two-step framework for analysing deterministic resampling schemes. Our analysis leads to a principled procedure that (i) removes weighting bias via Proposition 1 and (ii) isolates and characterises truncation effects. This framework is not specific to VR; it applies to any deterministic resampling scheme (all of which necessarily incur truncation effects). It offers concrete guidance for the design and theoretical assessment of deterministic resampling methods in general.

5. Our experiments highlight that lower empirical MSE can arise from underestimated uncertainty rather than improved approximation. Although this is an empirical observation rather than a formal theorem, it is not specific to VR and has broader implications for how MSE should be interpreted when evaluating SMC algorithms and other approximation techniques.

2 BIAS OF VARIATIONAL RESAMPLING

2.1 Motivating example

In this section, we analyse two sources of bias induced by VR in the context of the SIR algorithm. For concreteness, we use a Gaussian toy example as a running example. Here, the goal is to approximate a standard normal target distribution $\pi = \mathrm{N}(0,1)$ via the SIR algorithm with $\mu = \mathrm{N}(0, 1/(1-\lambda))$ and $G(x) = \exp(-\lambda x^2/2)$. We set $\lambda \coloneqq 8/10$ so that this example is equivalent (up to an irrelevant rescaling) to the example from Kviman et al. (2024, Section 5.1). Figure 1 illustrates our main findings (which will be made more rigorous over the next two sections).

1. *Weighting bias.* First, in Section 2.2, we prove that VR incurs a bias due to the fact that it sets all the resampled weights equal to $1/M$ even though the expected number of offspring of a particle is not proportional to its weight. This weighting bias can be seen in the approximation of the modal region of the target distribution in the fourth panel of Figure 1). We also propose a 'weighted' version of VR. As shown in the last panel of Figure 1, our proposed weighted VR avoids this weighting bias.

2. *Truncation bias.* Then, in Section 2.3, we prove that VR – with high probability – truncates the target distribution to a region in which the potential function exceeds a certain threshold; and that this truncation does not vanish as $N \to \infty$. This truncation bias can be seen in the underestimation of the tails of the target distribution in the fourth and fifth panel of Figure 1.

2.2 Weighting bias

VR incurs a bias due to the fact that it sets all the resampled weights equal to $1/M$ even though – in contrast to, e.g., multinomial, stratified or systematic resampling – the conditional expectation of the number of offspring of X^n is not exactly proportional to W^n.

This problem is incurred more generally by resampling schemes that are *deterministic*, i.e., whose resampling distribution can be written as

$$\rho(a^{1:M}; W^{1:N}) = \prod_{m=1}^{M} \delta_{h(m;W^{1:N})}(a^m),$$

for some mapping $h(\,\cdot\,; W^{1:N})\colon [M] \to [N]$ indexed by $W^{1:N} \in \mathsf{S}_N$. Proposition 1 below shows that the bias of deterministic resampling schemes can be removed

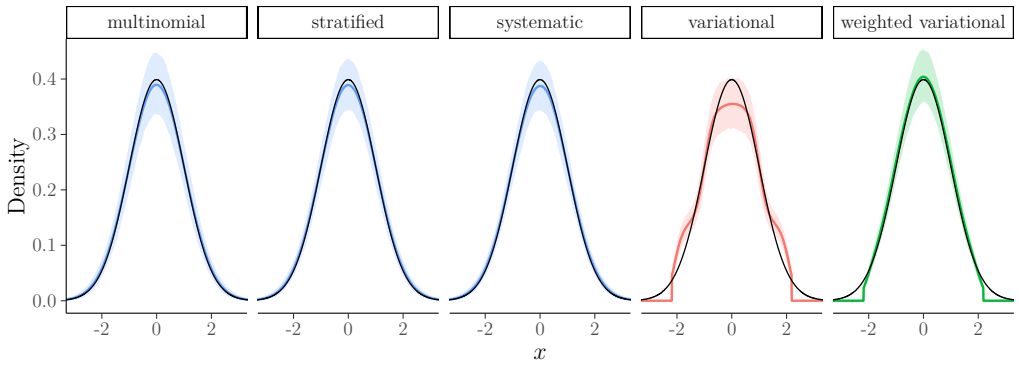

Figure 1: Average kernel-density estimate (coloured line) of SIR approximations for the Gaussian toy example with $\pi = \mathrm{N}(0, 1)$ (black line) and $N = M = 1000$. Based on 1000 independent repetitions of each algorithm; the shaded bands contain 90-% of the kernel-density estimates from each run. All kernel density estimates are limited to the range of values observed across all runs for a given resampling scheme. This figure illustrates both the weighting bias (in Panel 4) and the truncation bias (in Panels 4 and 5) of VR.

by adjusting the resampled weights. It uses the following assumption, where $\mathrm{im}(f)$ denotes the image (a.k.a. range) of some function f.

Assumption 1. *The deterministic resampling scheme is such that for any* $W^{1:N} \in \mathsf{S}_N$, $\{n \in [N] \mid W^n > 0\} \subseteq \mathrm{im}(h(\cdot\,; W^{1:N}))$.

Proposition 1. *Let* $M, N \in \mathbb{N}$. *Let* $\tilde{\pi}^M$ *be a SIR approximation based on some deterministic resampling scheme which satisfies Assumption 1; and that*

$$\widetilde{W}^m := \frac{W^{h(m;W^{1:N})}}{\#\{k \in [M] \mid h(k; W^{1:N}) = h(m; W^{1:N})\}}, \quad (2)$$

for all $m \in [M]$, *with convention* $\frac{0}{0} = 0$. *Then* $\mathbb{E}[\tilde{\pi}^M(f)] = \mathbb{E}[\hat{\pi}^N(f)]$, *for any* π*-integrable test function* $f \colon \mathsf{X} \to \mathbb{R}$.

Assumption 1 is an absolute-continuity condition which ensures that the support of the deterministic 'importance-sampling' proposal distribution $\rho(\cdot\,; W^{1:N})$ includes the support of a suitable target distribution. This assumption is, of course, very strong because it requires that any particle with positive weight is always guaranteed to have at least one offspring. Nonetheless, Proposition 1 allows us separate the bias into two components: (i) a removable "weighting" bias (eliminated under Assumption 1), and (ii) non-removable "truncation" bias (due to violating Assumption 1). The latter will be precisely characterised in Proposition 2 below. Specifically, we separate the bias through the following two-step procedure which applies to *any* deterministic resampling scheme:

1. *Truncation.* For any $n \in [N]$, replace the nth self-normalised particle weight W^n by
$$\frac{W^n}{\sum_{l \in \mathrm{im}(h(\cdot\,; W^{1:N}))} W^l} \mathbb{I}_{\mathrm{im}(h(\cdot\,; W^{1:N}))}(n). \quad (3)$$
2. *Deterministic resampling.* Perform deterministic resampling based on the truncated weights from (3) and using the resampled weights from (2). Note that conditional on the truncation, Assumption 1 is then satisfied, i.e., the *only source of bias comes from the truncation.*

Applying this idea to VR yields the novel 'weighted VR' scheme outlined in Algorithm 3, where $\widetilde{W}^m$ is obtained by combining (2) and (3) with the deterministic function $h(m; W^{1:M}) := A^m$ implicitly defined through Algorithm 2. In particular, we have simplified the resulting expression using that $\#\{k \in [M] \mid h(k; W^{1:N}) = h(m; W^{1:N})\} = K^{A^m}$ is the number of offspring of Particle A^m and $\mathrm{im}(h(\cdot\,; W^{1:N})) = \{l \in [N] \mid K^l > 0\}$ is the set of indices of particles that have at least one offspring.

Algorithm 3 (weighted VR). Proceed as in Algorithm 2 but with $\widetilde{W}^m$ in Step 2 replaced by
$$\widetilde{W}^m := \frac{W^{A^m}}{K^{A^m} \sum_{n \in \{l \in [N] \mid K^l > 0\}} W^n}.$$

As illustrated in the last two panels of Figure 1 our non-uniform weighting from Algorithm 3 can remove the weighting bias. However, as mentioned, it cannot remove the truncation bias. The latter is analysed in the next section.

2.3 Truncation bias

Our main result in this section is the following Proposition 2 (proved in Supplementary Appendix A) which provides a lower bound on the probability that the SIR approximation of π based on VR puts at least $\alpha \in (0,1]$ of its probability mass on the part of the state space in which the potential function exceeds some threshold $\beta > 0$, i.e., on

$$\mathbb{P}(\tilde{\pi}^M(\{G \geq \beta\}) \geq \alpha),$$

where $\{G \geq \beta\} \coloneqq \{x \in \mathsf{X} \mid G(x) \geq \beta\}$. Note that the randomness here is due to the randomness of the particles $X^{1:N}$; conditional on a specific realisation of these particles (and hence conditional on the weights $W^{1:N}$), VR is deterministic.

Proposition 2. *Let $N, M \in \mathbb{N}$, $\alpha \in (0,1]$, and let $X^1, \dots, X^N \overset{\text{IID}}{\sim} \mu$ conditional on which $A^1, \dots, A^M$ are generated by Algorithm 2. Additionally, assume that the random variables $G(X^1), \dots, G(X^N)$ are continuous. Then there exists $\beta_0 > 0$ such that for all $0 < \beta \leq \beta_0$,*

$$\nu(\beta) \coloneqq \mathbb{E}\left[\left\lfloor \frac{G(X^1)}{\beta \mathrm{e}} - 1 \right\rfloor \mathbb{I}_{\{G \geq 2\beta\mathrm{e}\}}(X^1)\right] > \frac{\alpha M}{N},$$

$$\sigma^2(\beta) \coloneqq \mathrm{var}\left[\left\lfloor \frac{G(X^1)}{\beta \mathrm{e}} - 1 \right\rfloor \mathbb{I}_{\{G \geq 2\beta\mathrm{e}\}}(X^1)\right] < \infty,$$

and

$$\mathbb{P}(\tilde{\pi}^M(\{G \geq \beta\}) \geq \alpha) \geq \frac{1}{1 + \frac{\sigma^2(\beta)}{N}\left(\nu(\beta) - \frac{\alpha M}{N}\right)^{-2}}.$$

To obtain some intuition for Proposition 2, take $\alpha = 1$. The following corollary then shows that the truncation bias induced by VR (seen, e.g., in the underestimation of the tails of the target distribution in Panels 4 and 5 of Figure 1) cannot be removed by increasing the number of *original* particles, N.

Corollary 1. *Assume the setting from Proposition 2. Then if $M = \mathrm{O}(N)$ as $N \to \infty$, there exists $\beta_0 > 0$ such that for all $0 < \beta \leq \beta_0$,*

$$\lim_{N \to \infty} \mathbb{P}(\tilde{\pi}^M(\{G \geq \beta\}) \geq \alpha) = 1.$$

Proof. Let (M_k) and (N_k) be sequences in $\mathbb{N}$ such that $N_k \to \infty$ and $M_k = \mathrm{O}(N_k)$ as $k \to \infty$. Then by Proposition 1, we can find $\beta_0 > 0$ such that $\nu(\beta) > \alpha M_1/N_1$ and consequently $\inf_{k \in \mathbb{N}} (\nu(\beta) - \alpha M_k/N_k) > 0$, as well as $\sigma^2(\beta) < \infty$, for all $0 < \beta \leq \beta_0$. □

Figure 3 illustrates Corollary 1 for $\alpha = 1$. Note that the truncation bias is most severe in the upper-triangular panels where M is much smaller than N. Conversely, the lower-triangular panels of Figure 3 suggest that the truncation bias vanishes if M is much larger than N.

3 NUMERICAL ILLUSTRATION

3.1 Setup

In this section, we compare the performance of the resampling schemes mentioned above in the context of SMC algorithms for four widely-used state-space models – including all those tested in Kviman et al. (2024).

State-space models. A *state-space model (SSM)* is a bivariate Markov chain $(X_t, Y_t)_{t \geq 1}$ on $\mathsf{X} \times \mathsf{Y}$, with initial distribution $\mathbb{P}((X_1, Y_1) \in \mathrm{d}x_1 \times \mathrm{d}y_1) = M_1(x_1)G_1(x_1, y_1)\mathrm{d}x_1\mathrm{d}y_1$ and transitions $\mathbb{P}((X_t, Y_t) \in \mathrm{d}x_t \times \mathrm{d}y_t | (X_{t-1}, Y_{t-1}) = (x_{t-1}, y_{t-1})) = M_t(x_{t-1}, x_t)G_t(x_t, y_t)\mathrm{d}x_t\mathrm{d}y_t$, in which only the second component is observable. For a given realised observation sequence $y_{1:T} \in \mathsf{Y}^T$ of length $T \in \mathbb{N}$, it is of interest to calculate (or, at least, approximate):

1. the *likelihood:* $\mathcal{Z}_T \coloneqq \mathbb{E}[\prod_{t=1}^T G_t(X_t, y_t)]$;
2. the *marginal filtering distribution* at time t: $\pi_{t|t}(\mathrm{d}x_t) = \mathbb{P}(X_t \in \mathrm{d}x_t | Y_{1:t} = y_{1:t})$;
3. the *marginal smoothing distribution* at time t: $\pi_{t|T}(\mathrm{d}x_t) = \mathbb{P}(X_t \in \mathrm{d}x_t | Y_{1:T} = y_{1:T})$.

We consider the following four models. In all of these, $\mathsf{X} = \mathbb{R}^D$, for some $D \in \mathbb{N}$ and $\mathsf{Y} = \mathbb{R}$.

1. a simple *linear-Gaussian* SSM: $D = 1$, $M_1(x_1) = \mathrm{N}(x_1; 0, 0.5^2/(1-0.95^2))$, where $\mathrm{N}(x; a, b)$ denotes the density of a normal distribution with mean a and variance b evaluated at x, and
$$M_{t+1}(x_t, x_{t+1}) \coloneqq \mathrm{N}(x_{t+1}; 0.95x_t, 0.5^2),$$
$$G_t(x_t, y_t) \coloneqq \mathrm{N}(y_t; x_t, 1);$$
2. a *stochastic volatility* model: $D = 1$, $M_1(x_1) = \mathrm{N}(x_1; 0, \sigma^2/(1-\phi^2))$ and
$$M_{t+1}(x_t, x_{t+1}) \coloneqq \mathrm{N}(x_{t+1}; \phi x_t, \sigma^2),$$
$$G_t(x_t, y_t) \coloneqq \mathrm{N}(y_t; 0, \beta^2 \exp(x_t));$$
as in Kviman et al. (2024), we set (ϕ, σ, β) to $(0.91, 1, 0.5)$ for generating simulated data and to $(0.8, 1, 0.01)$ for the real-data experiment;
3. the *nonlinear* SSM from Kitagawa (1998): $D = 1$, $M_1(x_1) = \mathrm{N}(x_1; 0, 5)$ and
$$M_{t+1}(x_t, x_{t+1}) \coloneqq \mathrm{N}(x_{t+1}; f_{t+1}(x_t), 1),$$
$$G_t(x_t, y_t) \coloneqq \mathrm{N}(y_t; x_t^2/20, 10),$$
with
$$f_{t+1}(x_t) \coloneqq \frac{x_t}{2} + \frac{25x_t}{1.0 + x_t^2} + 8\cos(1.2(t+1));$$

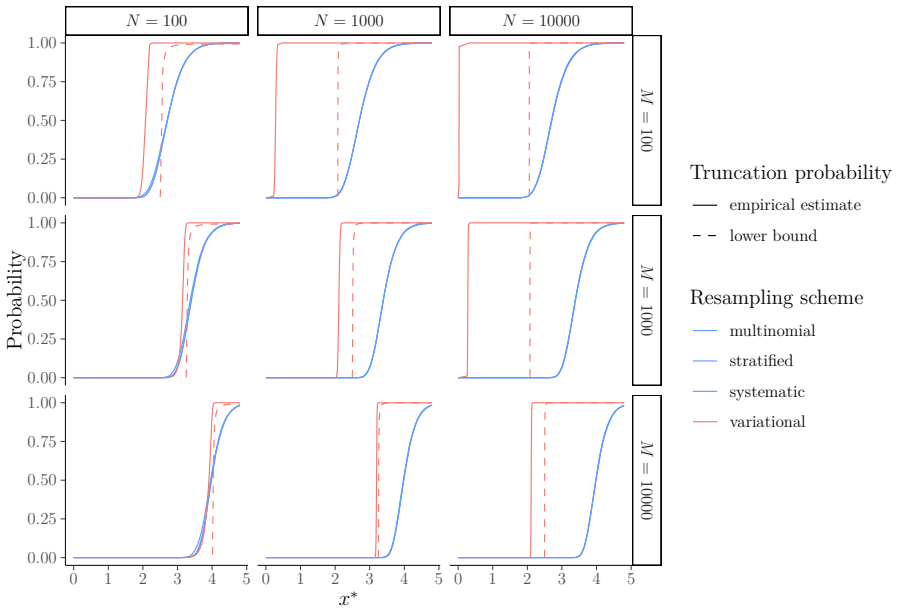

Figure 2: The probability that the support of the SIR approximation of $\pi = \mathrm{N}(0, 1)$ in the Gaussian toy example is fully contained within the interval $[-x^*, x^*]$. Empirical estimates are based on 10 000 independent repetitions of each algorithm for each combination of N and M; the lower bounds (for VR) are from Proposition 2. Note that the lower bound is only defined for $\beta \leq \beta_0$ for some $\beta_0 > 0$, or, equivalently, for sufficiently large values of x^*.

4. the *Lorenz-63* SSM: $D = 3$, $M_1 = \mathrm{N}(\mathbf{0}, \mathbf{I}_3)$, and

$$M_{t+1}(x_t, x_{t+1}) \coloneqq \mathrm{N}(x_{t+1}; f(x_t), \tfrac{1}{2}\delta),$$
$$G_t(x_t, y_t) \coloneqq \mathrm{N}(y_t; x_{t,1}, 1),$$

where $x_t = (x_{t,1}, x_{t,2}, x_{t,3})^{\mathrm{T}}$, $\delta \coloneqq 1/100$, and

$$f(x_t) \coloneqq \begin{bmatrix} x_{t,1} + 10\delta(x_{t,2} - x_{t,1}) \\ x_{t,2} + \delta(28x_{t,1} - x_{t,1}x_{t,3} - x_{t,2}) \\ x_{t,3} + \delta(x_{t,1}x_{t,2} - \frac{8}{3}x_{t,3}) \end{bmatrix}.$$

Sequential Monte Carlo. Algorithm 4 outlines the SMC algorithm used in our simulations. Here, $W_t^n \coloneqq w_t^n / \sum_{l=1}^N w_t^l$, and operations stated for n are to be carried out independently for *all* $n \in [N]$.

Algorithm 4 (SMC – e.g. Gordon et al. 1993).

1. Sample $X_1^n \sim M_1$; set $w_1^n \coloneqq G_1(X_1^n, y_1)/N$;
2. for $t = 2, \ldots, T$,
   (a) sample $A_{t-1}^{1:N} \sim \rho(\,\cdot\,; W_{t-1}^{1:N})$;
   (b) sample $X_t^n \sim M_t(X_{t-1}^{A_{t-1}^n}, \,\cdot\,)$;
   (c) set $w_t^n \coloneqq \widetilde{W}_{t-1}^n G_t(X_t^n, y_t)$.

Resampling schemes. We test the following resampling schemes the first four of which set $\widetilde{W}_{t-1}^n \coloneqq 1/N$ in Step 2c of Algorithm 4:

1. *'multinomial'*: standard multinomial resampling;
2. *'stratified'*: standard stratified resampling;
3. *'systematic'*: standard systematic resampling;
4. *'variational'*: the VR scheme from Kviman et al. (2024) (Algorithm 2);
5. *'weighted variational'*: the weighted VR scheme proposed in Section 2.2 (Algorithm 3); this sets

$$\widetilde{W}_{t-1}^n \coloneqq \frac{W_{t-1}^{A_{t-1}^n} / \#\{m \in [M] \mid A_{t-1}^m = A_{t-1}^n\}}{\sum_{a \in \{A_{t-1}^1, \ldots, A_{t-1}^N\}} W_{t-1}^a},$$

in Step 2c of Algorithm 4 (note that the summation in the denominator skips repeated elements).

For each resampling scheme, we compare two versions:

1. *'standard weights'*: resample at time t as described in Step 2a of Algorithm 4;
2. *'smoothing weights'*: as suggested in Kviman et al. (2024), resample at time t by sampling parent indices in Step 2a of Algorithm 4 from $\rho(\,\cdot\,; V_{t-1}^{1:N})$, where $V_t^n \coloneqq v_t^n / \sum_{l=1}^N v_t^l$ with $v_1^n \coloneqq M_1(X_1^n)G_1(X_1^n, y_1)$ and, for $t > 1$,

$$v_t^n \coloneqq v_{t-1}^{A_{t-1}^n} M_t(X_{t-1}^{A_{t-1}^n}, X_t^n) G_t(X_t^n, y_t).$$

Resampling according to modified weights $V_t^n \neq W_t^n$ induces further bias; this could be removed by accounting for the discrepancy between W_t^n and V_t^n in the resampled weights, e.g., by setting $\widetilde{W}_t^n \coloneqq W_t^{A_t^n} / (N V_t^{A_t^n})$ instead of $\widetilde{W}_t^n \coloneqq 1/N$, in the first four resampling schemes mentioned above. However, we do not perform such a modification here to allow for comparison with Kviman et al. (2024) and because preliminary simulations suggested that doings increases in the variance of the likelihood estimates.

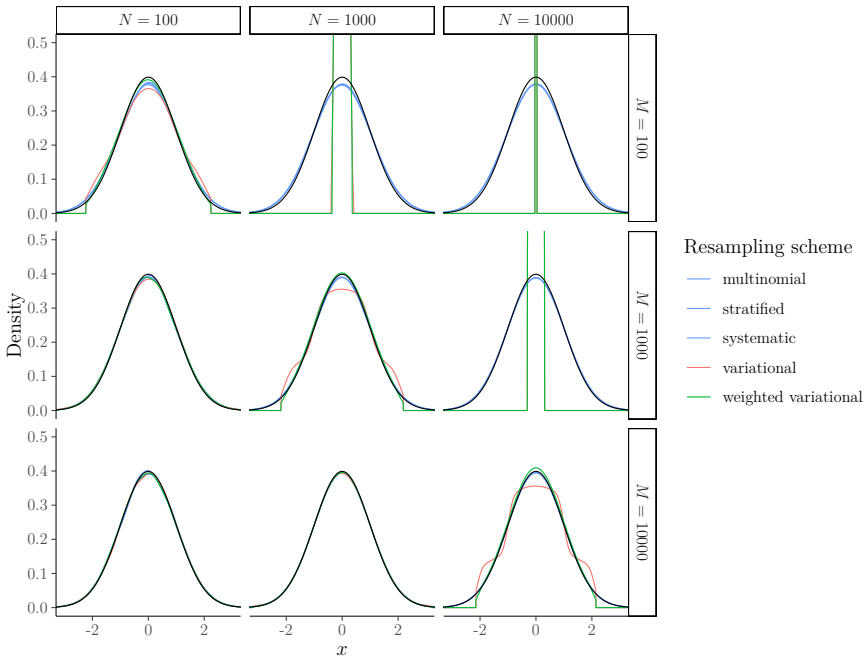

Figure 3: Average kernel density estimates of SIR approximations (coloured lines) for the same Gaussian toy example with $\pi = \mathrm{N}(0, 1)$ (black line). The results are based on 1000 independent repetitions of each algorithm. All kernel density estimates are limited to the range of values observed across all runs for a given resampling scheme and given choice of N and M. Note that these limits coincide for the original VR scheme from Kviman et al. (2024) (Algorithm 2) and the weighted VR scheme proposed above (Algorithm 3):

3.2 Results – simulated data

We run the SMC algorithm with $N = 1000$ particles for the different resampling schemes mentioned above. From each of the four SSMs, we simulate 20 observation sequences $y_{1:T}$ of length $T = 50$ and perform 20 independent repetitions of each algorithm for each of these observation sequences (i.e., in total, 400 independent repetitions of each algorithm per model). The ground truth is obtained via the Kalman filter for the linear-Gaussian SSM. For all others, it is approximated via an SMC algorithm with $N = 20\,000$ particles and stratified resampling which also generates 1000 trajectories from the joint smoothing distribution via backward sampling.

Likelihood estimates. Figure 4 illustrates the relative estimates of the likelihood: $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$ where $\widehat{\mathcal{Z}}_T := \prod_{t=1}^{T}\sum_{n=1}^{N} w_t^n$ is the SMC approximation of $\mathcal{Z}_T$. The results suggest that:

1. *VR does not improve likelihood estimation* compared to stratified or systematic resampling.
2. *Use of the 'smoothing weights' does not improve likelihood estimation.*

Filtering and smoothing errors. We now report an *average squared Mahalanobis distance* type metric:

$$\frac{1}{TD}\sum_{t=1}^{T}\mathbb{E}\big[(\hat{X}_t-\mu_t)^{\mathrm{T}}\Sigma_t^{-1}(\hat{X}_t-\mu_t)\big], \qquad (4)$$

to assess the error of SMC approximations $\hat{\pi}_t$ of the marginal filtering distributions ($\pi_t = \pi_{t|t}$) and marginal smoothing distributions ($\pi_t = \pi_{t|T}$). Here, μ_t and Σ_t are the ground-truth mean and covariance matrix of π_t and $\hat{X}_t \sim \hat{\pi}_t$.

Values of (4) approximately equal to 1 indicates the mean and variance are well approximated; values less than 1 can indicate an underestimation of uncertainty; values greater than 1 can indicate an overestimation of uncertainty.

For each model, we report three variants of (4) (Supplementary Appendix B gives further details):

1. *'filtering'* assesses the approximation of $\pi_{t|t}$ by $\hat{\pi}_t = \frac{1}{N}\sum_{n=1}^{N} W_t^n \delta_{X_t^n}$;
2. *'smoothing (ancestor tracing)'* assesses the approximation of $\pi_{t|T}$ by $\hat{\pi}_t = \frac{1}{N}\sum_{n=1}^{N} W_T^n \delta_{X_t^{(n)}}$, where $X_t^{(n)}$ is the time-t ancestor particle of the nth particle at time T (Kitagawa, 1993).
3. *'smoothing (backward sampling)'* assesses the approximation of $\pi_{t|T}$ by $\hat{\pi}_t = \frac{1}{\tilde{N}}\sum_{n=1}^{\tilde{N}} \delta_{\tilde{X}_t^n}$, where $\tilde{X}_t^n$ is the time-t particle from the nth smoothing trajectory (out of $\tilde{N} := N/2 = 500$) obtained via backward sampling (Godsill et al., 2004). In contrast to ancestor tracing, backward sampling does not suffer from path degeneracy and can therefore give more accurate approximations of $\pi_{t|T}$.

Figure 5 shows results for the linear-Gaussian SSM. Results for the other models are qualitatively similar (though for the Lorenz-63 model, (4) does not seem

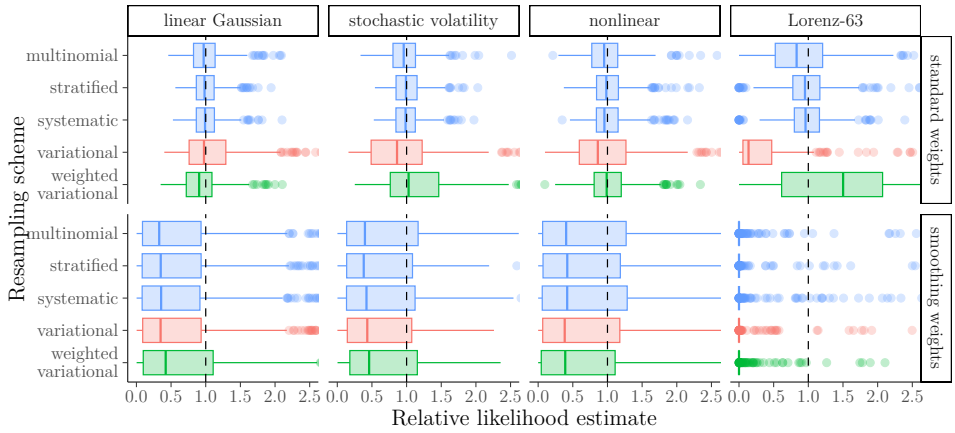

Figure 4: Relative estimates of the likelihood: $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$, obtained from the SMC algorithm with different resampling schemes. Values closer to 1 (dashed line) are better.

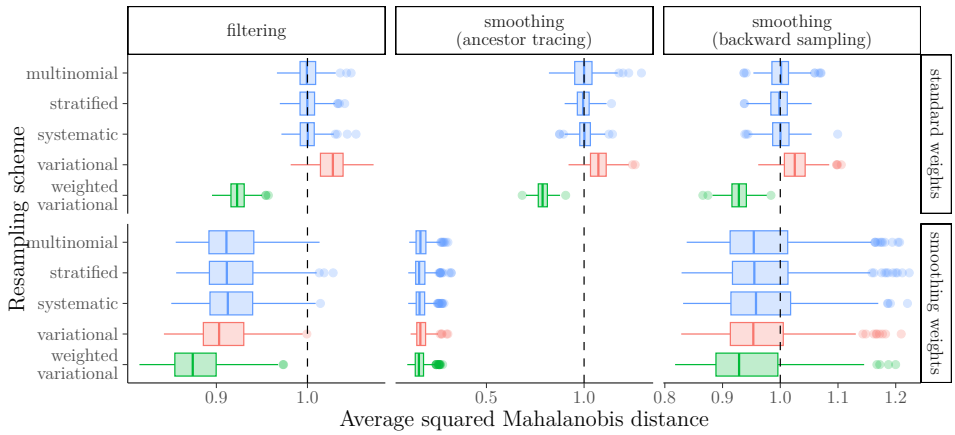

Figure 5: Average squared Mahalanobis distance of the approximations of the marginal filtering and smoothing distributions in the linear-Gaussian SSM. Values closer to 1 (dashed line) are better. Values below 1 indicate an underestimation of uncertainty.

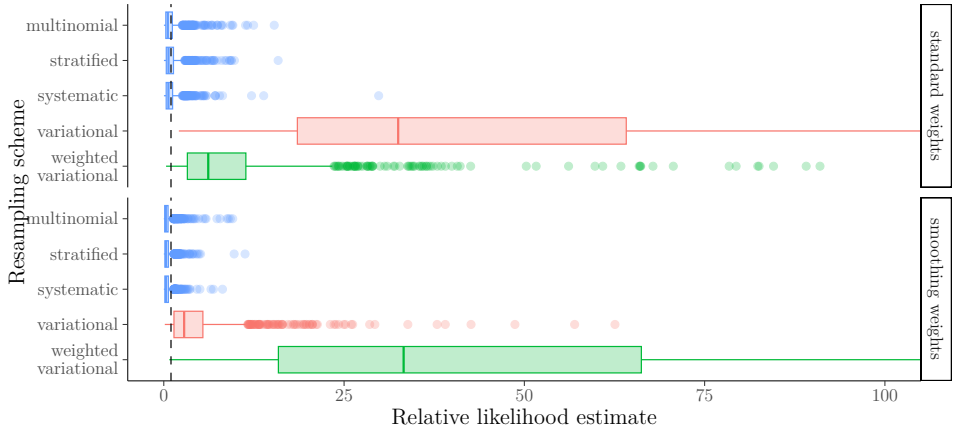

Figure 6: Relative estimates of the likelihood: $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$, in the stochastic volatility model using the same data set with $T = 2010$ observations and the same model parameters as in Kviman et al. (2024). Values closer to 1 (dashed line) are better.

very meaningful due to the likely poor performance of the bootstrap proposal mechanism) and are therefore deferred to Supplementary Appendix D. Therein we also show results for the average MSE of the SMC estimate of the mean of the filtering and smoothing distributions which was used in Kviman et al. (2024) and which can be obtained by taking Σ_t to be the identity matrix.

At a first glance, these results seem to support the finding in Kviman et al. (2024) that VR may lead to a smaller MSE than conventional resampling schemes when combined with the 'smoothing weights'. However, Figure 5 – along with the additional results in Supplementary Appendix D – suggest:

1. *VR does not generally reduce the MSE.* The smaller MSE observed in Kviman et al. (2024) seems to be almost entirely due to additional bias induced by the 'smoothing weights' rather than

due to the VR scheme itself (because multinomial, stratified or systematic resampling lead to a similarly reduced MSE when combined with these 'smoothing weights').

2. *A smaller MSE does not signify a 'better' approximation.* The smaller MSE is not due to a more accurate approximation of the filtering or smoothing distributions. It may be a sign of an underestimation of uncertainty (note that (4) is far below 1 when using the 'smoothing weights').

3.3 Results – real data

We end this section by reproducing the real-data experiment from Kviman et al. (2024), That is, we apply the stochastic volatility model with parameters $(\phi, \sigma, \beta) = (0.8, 1, 0.01)$ to the closing prices of the S&P500 stock index from 2006-04-03 to 2014-03-31. As in Kviman et al. (2024), we compare estimates of the likelihood using $N = 1000$ particles and for different resampling schemes with/without the 'smoothing weights'. Due to the length ($T = 2010$) of the observation sequence, we performed 1000 independent repeats of each algorithm; we obtained a ground-truth log-likelihood of 5473.36 using an SMC algorithm with 150 000 particles and stratified resampling. Supplementary Appendix C provides further details.

Figure 6 confirms the finding from Kviman et al. (2024) that in this particular setting, VR induces typically larger estimates of the likelihood than stratified resampling (reports for multinomial or systematic resampling were not reported). However, Figure 6 also makes it clear that such larger values do not necessarily signify a 'more accurate' approximation.

4 CONCLUSION

We have analysed the deterministic resampling scheme called *variational resampling (VR)* recently proposed in Kviman et al. (2024). We have identified two sources of bias incurred by VR: a *weighting* bias and a *truncation* bias; for the latter, we have provided a rigorous theoretical analysis; for the former, we have shown that it can be alleviated by an uneven weighting of the resampled particles. However, when used within SMC algorithms, the truncation bias appears to dominate and may even be made worse by the uneven post-resampling weighting. More generally, our numerical experiments do not find that VR outperforms standard resampling schemes such as multinomial, stratified or systematic resampling. Code is available at https://github.com/AxelFinke/bias-of-variational_resampling-aistats. Simulation outputs (250 MB) are hosted on Figshare: https://doi.org/10.6084/m9.figshare.30272443

Acknowledgments

NB acknowledges support from the ProbAI Hub.

References

Bouchard-Côté, A., Sankararaman, S., and Jordan, M. I. (2012). Phylogenetic inference via sequential Monte Carlo. *Systematic Biology*, 61(4):579–593.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). An improved particle filter for nonlinear problems. *IEE Proceedings – Radar, Sonar and Navigation*, 146(1):2–7.

Chopin, N. and Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*. Springer.

Creal, D. (2012). A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69. IEEE.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, Department of Statistics, University of Oxford, UK.

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F, Radar and Signal Processing*, 140(2):107–113.

Kitagawa, G. (1993). A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models. In *Proceedings of the 2nd US–Japan Joint Seminar on Statistical Time Series Analysis*, volume 110.

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.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215.

Koptagel, H., Kviman, O., Melin, H., Safinianaini, N., and Lagergren, J. (2022). VaiPhy: A variational inference based algorithm for phylogeny. *Advances in Neural Information Processing Systems*, 35:14758–14770.

Kviman, O., Branchini, N., Elvira, V., and Lagergren, J. (2024). Variational resampling. In *International Conference on Artificial Intelligence and Statistics*, pages 3286–3294. PMLR.

Li, T., Bolic, M., and Djuric, P. M. (2015). Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86.

Martino, L., Elvira, V., and Louzada, F. (2016). Weighting a resampled particle in sequential Monte Carlo. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5. IEEE.

Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):543–546.

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In *Bayesian Statistics 3. Proceedings of the third Valencia international meeting, 1–5 June 1987*, pages 395–402. Clarendon Press.

Storvik, G., Diz-Lois Palomares, A., Engebretsen, S., Rø, G. Ø. I., Engø-Monsen, K., Kristoffersen, A. B., de Blasio, B. F., and Frigessi, A. (2023). A sequential Monte Carlo approach to estimate a time-varying reproduction number in infectious disease models: the Covid-19 case. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 186(4):616–632.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A PROOFS

Proof (of Proposition 1). Again writing $h(\,\cdot\,) \coloneqq h(\,\cdot\,; W^{1:N})$ to simplify the notation:

$$\begin{aligned}
\mathbb{E}[\tilde{\pi}^M(f)] &= \mathbb{E}\Big[\sum_{m=1}^{M} \widetilde{W}^m f(\widetilde{X}^m)\Big] \\
&= \mathbb{E}\Big[\sum_{m=1}^{M} \frac{G(X^{h(m)})}{\#\{k \in [M] \mid h(k) = h(m)\} \sum_{l=1}^{N} G(X^l)} f(X^{h(m)})\Big] \\
&= \mathbb{E}\Big[\sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\mathbb{I}_{\{n\}}(h(m)) G(X^n)}{\#\{k \in [M] \mid h(k) = n\} \sum_{l=1}^{N} G(X^l)} f(X^n)\Big] \\
&= \mathbb{E}\Big[\sum_{n=1}^{N} \frac{G(X^n)}{\#\{k \in [M] \mid h(k) = n\} \sum_{l=1}^{N} G(X^l)} f(X^n) \sum_{m=1}^{M} \mathbb{I}_{\{n\}}(h(m))\Big] \\
&= \mathbb{E}\Big[\sum_{n=1}^{N} \frac{G(X^n)}{\sum_{l=1}^{N} G(X^l)} f(X^n)\Big] \\
&= \mathbb{E}[\hat{\pi}^N(f)].
\end{aligned}$$

This completes the proof. □

Lemma 1. *For any $u > 0$, the function C_u defined in (1) has an inverse $C_u^{-1}\colon (0, u] \to [0, \infty)$ which is strictly monotonically decreasing. Additionally, for any $0 < \beta \leq u$:*

$$\lfloor C_u^{-1}(\beta) \rfloor \geq \left\lfloor \frac{u}{\beta} \mathrm{e}^{-1} - 1 \right\rfloor \mathbb{I}_{(0, \mathrm{e}^{-1}u/2]}(\beta).$$

Proof. For any $u > 0$, the function C_u defined in (1) is strictly monotonically decreasing and hence bijective so that its inverse $C_u^{-1}\colon (0, u] \to [0, \infty)$ is well defined and also strictly monotonically decreasing.

For the lower bound, it suffices to show that $C_u^{-1}(\beta) \geq \mathrm{e}^{-1}u/\beta - 1$, for any $\beta \in (0, \mathrm{e}^{-1}u/2]$, or, equivalently, that for any $k \geq 0$:

$$C_u(k) \geq \frac{u}{k+1} \mathrm{e}^{-1}.$$

However, this inequality holds because

$$\begin{aligned}
C_u(k) &= \exp(k \log(k) - (k+1)\log(k+1) + \log(u)) \\
&= u \exp\Big(-k \log\Big(1 + \frac{1}{k}\Big) - \log(k+1)\Big) \\
&\geq \frac{u}{k+1} \mathrm{e}^{-1},
\end{aligned}$$

where we have used that $\log(1 + 1/k) \leq 1/k$ which follows from the well known inequality: $\log(x) \leq x - 1$, for any $x > 0$. □

Lemma 2 (Paley–Zygmund inequality). *Let Z be a non-negative real-valued random variable with finite variance and $0 \leq \theta \leq 1$. Then*

$$\mathbb{P}(Z \geq \theta\, \mathbb{E}[Z]) \geq \frac{(1-\theta)^2\, \mathbb{E}[Z]^2}{\mathrm{var}[Z] + (1-\theta)^2\, \mathbb{E}[Z]^2}.$$

Proof. By the Cauchy–Schwarz inequality:

$$\mathbb{E}[Z - \theta\,\mathbb{E}(Z)] \leq \mathbb{E}[(Z - \theta\,\mathbb{E}(Z))\,\mathbb{I}_{[\theta\,\mathbb{E}(Z),\infty)}(Z)] \leq \mathbb{E}[(Z - \theta\,\mathbb{E}(Z))^2]^{1/2}\,\mathbb{P}(Z \geq \theta\,\mathbb{E}[Z])^{1/2}.$$

This completes the proof. □

Proof (of Proposition 2). Write $U^n \coloneqq G(X^n)$, for any $n \in [N]$, to simplify the notation and assume that $U^1, \dots, U^N$ are (almost surely) mutually distinct. Let K^n denote the number of offspring of Particle n generated by Algorithm 2 with W^n replaced by U^n, for any $n \in [N]$ (note that the output of Algorithm 2 is invariant under a scaling of the weights $W^1, \dots, W^N$ by some common positive factor). Furthermore, define the random variable $\gamma \coloneqq \max\{\gamma' > 0 \mid \sum_{n=1}^N \lfloor C_{U^n}^{-1}(\gamma') \rfloor \geq M\}$. Then we have the inequality:

$$\begin{aligned}
\mathbb{P}(\tilde{\pi}^M(\{G \geq \beta\}) \geq \alpha) &\geq \mathbb{P}(\{\tilde{\pi}^M(\{G \geq \beta\}) \geq \alpha\} \cap \{\gamma \leq \beta\}) \\
&= \mathbb{P}\Bigg(\Bigg\{\sum_{n=1}^N K^n\,\mathbb{I}_{[\beta,\infty)}(U^n) \geq \alpha M\Bigg\} \cap \{\gamma \leq \beta\}\Bigg) \\
&= \mathbb{P}\Bigg(\Bigg\{\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\gamma) \rfloor\,\mathbb{I}_{[\beta,\infty)}(U^n) \geq \alpha M\Bigg\} \cap \{\gamma \leq \beta\}\Bigg) \\
&\geq \mathbb{P}\Bigg(\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\beta) \rfloor\,\mathbb{I}_{[\beta,\infty)}(U^n) \geq \alpha M\Bigg) \\
&= \mathbb{P}\Bigg(\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\beta) \rfloor \geq \alpha M\Bigg) \\
&= \mathbb{P}\Bigg(\frac{1}{N}\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\beta) \rfloor \geq \frac{\alpha M}{N}\Bigg). \qquad (5)
\end{aligned}$$

Here, the third line uses that $K^n = \lfloor C_{U^n}^{-1}(\gamma) \rfloor$, for $n \in [N]$; the fourth line follows from the fact that C_u^{-1} is monotonically decreasing for any $u > 0$ (Lemma 1); and the penultimate line is due to the fact that $U^n \leq \beta$ implies that $\lfloor C_{U^n}^{-1}(\beta) \rfloor = 0$, for $n \in [N]$.

By Lemma 1, we can further bound

$$\frac{1}{N}\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\beta) \rfloor \geq \frac{1}{N}\sum_{n=1}^N \left\lfloor \frac{U^n}{\beta}\mathrm{e}^{-1} - 1 \right\rfloor \mathbb{I}_{[2\beta\mathrm{e},\infty)}(U^n) =: Z^N(\beta).$$

Then $\mathbb{E}[Z^N(\beta)] = \nu(\beta)$ and $\mathrm{var}[Z^N(\beta)] = \sigma^2(\beta)/N$, where we note the following.

- The function $\beta \mapsto \nu(\beta)$ is decreasing with $\lim_{\beta \to 0} \nu(\beta) = \infty$. As a result, there exists $\beta_0 > 0$ such that $\theta(\beta) \coloneqq \alpha M/[N\nu(\beta)] < 1$ for any $\beta \leq \beta_0$.
- For any $\beta > 0$, $\sigma^2(\beta) \leq \beta^{-1}\mathrm{e}^{-1}\mu(G^2) + 1/4 < \infty$, by assumption on the model.

Consequently, Lemma 2 lets us lower-bound (5) for any $0 < \beta \leq \beta_0$ by

$$\begin{aligned}
\mathbb{P}\Bigg(\frac{1}{N}\sum_{n=1}^N \lfloor C_{U^n}^{-1}(\beta) \rfloor \geq \frac{\alpha M}{N}\Bigg) &\geq \mathbb{P}\big(Z^N(\beta) \geq \theta(\beta)\nu(\beta)\big) \\
&\geq \frac{(1 - \theta(\beta))^2\nu(\beta)^2}{\sigma^2(\beta)/N + (1 - \theta(\beta))^2\nu(\beta)^2} \\
&= \left[1 + \frac{\sigma^2(\beta)}{N\nu(\beta)^2}\Big(1 - \frac{\alpha M}{N\nu(\beta)}\Big)^{-2}\right]^{-1}.
\end{aligned}$$

This completes the proof. □

Proposition 3 (VR minimises the KL divergence). *For a given weighted sample $(X^{1:N}, W^{1:N})$ (and hence a given weighted empirical measure $\hat{\pi}^N = \sum_{n=1}^N W^n \delta_{X^n}$), the collection of ancestor indices $A^{1:M} \in [N]^M$ generated by Algorithm 2 satisfies*

$$A^{1:M} \in \arg\min_{\tilde{A}^{1:M} \in [N]^M} \mathrm{KL}(\tilde{\pi}^M_{\tilde{A}^{1:M}} \| \hat{\pi}^N),$$

where

$$\tilde{\pi}^M_{\tilde{A}^{1:M}}(\mathrm{d}x) \coloneqq \sum_{m=1}^M \frac{1}{M} \delta_{X^{\tilde{A}^m}}(\mathrm{d}x)$$

is the resampled empirical measure of the particles based on ancestor indices $\tilde{A}^{1:M} \in [N]^M$ and where $\mathrm{KL}(\mu\|\nu)$ is the KL-divergence from ν to μ.

Proof. For a given collection of ancestor indices $A^{1:M} \in [N]^M$, let $K^n \coloneqq \#\{m \in [M] | A^m = n\}$ be the number of offspring of the nth particle, for $n \in [N]$. Then

$$\mathrm{KL}(\tilde{\pi}^M_{A^{1:M}} \| \hat{\pi}^N) = \frac{1}{M} \sum_{n=1}^N K^n \log\Big(\frac{K^n/M}{W^n}\Big) = -\frac{1}{M} \sum_{n=1}^N \sum_{k=0}^{K^n-1} \log C_{W^n}(k) - \frac{N}{M} \log(M),$$

with convention $\sum_{k=a}^b = 0$ if $b < a$. Note that $k \mapsto C_u(k)$ (defined in Equation 1 of the main text) is strictly decreasing. Thus, to minimise the KL-divergence, additional descendants must always be allocated to the ancestor particle n for which $C_{W^n}(K^n - 1)$ is maximal. More formally, the proof is completed by induction on the total number of offspring M.

- For $M = 1$, the KL divergence is minimised by taking $A^1 := \max_{n\in[N]} C_{W^n}(1)$ and hence $K^{A^1} = 1$.
- Assume now that the offspring counts $\tilde{K}^{1:N}$ with $\sum_{n=1}^N \tilde{K}^n = M - 1$ (equivalently, up to an irrelevant ordering, the ancestor indices $\tilde{A}^{1:(M-1)}$) minimise the KL-divergence for some $M > 1$. Then the offspring counts $K^{1:N}$ (equivalently, up to an irrelevant ordering, the ancestor indices $A^{1:M}$) minimise the KL-divergence if we take

$$A^{1:(M-1)} \coloneqq \tilde{A}^{1:(M-1)},$$
$$A^M \in \arg\max_{n\in[N]} C_{W^n}(\tilde{K}^n),$$

  and $K^{A^M} \coloneqq \tilde{K}^n + 1$, if $n = A^M$, and $K^{A^M} \coloneqq \tilde{K}^n$, otherwise. This is exactly what Algorithm 2 does.

□

B FURTHER DETAILS ON THE AVERAGE SQUARED MAHALANOBIS-TYPE METRIC

Here, we give more details on how we compute the estimates of the average squared Mahalanobis type metric from (4). Recall that D is the dimension of the latent state x_t:

1. *'filtering'* approximates (4) as

$$\frac{1}{TD} \sum_{t=1}^T \sum_{n=1}^N W_t^n (X_t^n - \mu_{t|t})^{\mathrm{T}} \Sigma_{t|t}^{-1} (X_t^n - \mu_{t|t}),$$

   where $\mu_{t|t}$ and $\Sigma_{t|t}$ are the ground-truth mean and covariance matrix of the marginal filtering distribution at time t.

2. *'smoothing (ancestor tracing)'* approximates (4) as

$$\frac{1}{TD} \sum_{t=1}^T \sum_{n=1}^N W_T^n (X_t^{B^n_{t|T}} - \mu_{t|T})^{\mathrm{T}} \Sigma_{t|T}^{-1} (X_t^{B^n_{t|T}} - \mu_{t|T}),$$

where $\mu_{t|T}$ and $\Sigma_{t|T}$ are the ground-truth mean and covariance matrix of the marginal smoothing distribution at time t and where $B^n_{t|T}$ is the time-t particle index t of the nth surviving particle lineage at time T, given by $B^n_{T|T} \coloneqq n$, and, for $t = T-1, \ldots, 1$:

$$B^n_{t|T} \coloneqq A_t^{B^n_{t+1|T}}.$$

Note that ancestor tracing suffers from path degeneracy if T is large relative to N. That is, for t much smaller than T, we likely have $B^1_{t|T} = \ldots = B^N_{t|T}$.

3. *'smoothing (backward sampling)'* approximates (4) as

$$\frac{1}{TD\tilde{N}} \sum_{t=1}^{T} \sum_{n=1}^{\tilde{N}} (X_t^{\tilde{B}^n_{t|T}} - \mu_{t|T})^{\mathrm{T}} \Sigma^{-1}_{t|T} (X_t^{\tilde{B}^n_{t|T}} - \mu_{t|T}),$$

where $\tilde{B}^1_{T|T}, \ldots, \tilde{B}^{\tilde{N}}_{T|T} \overset{\text{IID}}{\sim} \mathrm{Cat}(W_T^{1:N})$ and for $n \in [\tilde{N}]$ and $t = T-1, \ldots, 1$, we independently sample

$$\tilde{B}^n_{t|T} \sim \mathrm{Cat}\left(\left(\frac{W_t^k M_{t+1}(X_t^k, X_{t+1}^{\tilde{B}^n_{t|T}})}{\sum_{l=1}^N W_t^l M_{t+1}(X_t^l, X_{t+1}^{\tilde{B}^n_{t|T}})} \right)_{k=1}^{N} \right).$$

Throughout, we take $\tilde{N} \coloneqq N/2 = 500$. In contrast to ancestor tracing, backward sampling does not suffer from path degeneracy (in suitably regular models like the ones considered in this work).

C FURTHER DETAILS ABOUT THE REAL-DATA EXPERIMENT

In this section, we provide more details on the real-data experiment from Section 3.3. Our setup follows Kviman et al. (2024). That is, we apply the stochastic volatility model with parameters $(\phi, \sigma, \beta) = (0.8, 1, 0.01)$ to the closing prices of the S&P500 stock index from 2006-04-03 to 2014-03-31. The data are available from `https://finance.yahoo.com/quote/%5EGSPC/history?ltr=1`. However, as the functionality for directly downloading historical data from `finance.yahoo.com` is now paywalled, we downloaded the data from `https://www.kaggle.com/datasets/henryhan117/sp-500-historical-data`.

The stochastic volatility model assumes that the data are log-returns with mean zero. Let S_t denote the closing price on the tth trading day in the data set. Then the tth log-return is given by $r_t \coloneqq \log(S_{t+1}/S_t)$. The observations are then typically specified by de-meaning these log-returns, i.e.,

$$y_t = r_t - \bar{r},$$

for $t = 1, \ldots, T$, with $T = 2011$, where $\bar{r}$ is the mean of all the log-returns in the data set.

However, according to `https://github.com/okviman/Variational-Resampling/blob/main/main_snp500.py`, Kviman et al. (2024) specified the observations as:

$$y_t \coloneqq r_{t+1} - r_t, \tag{6}$$

for $t = 1, \ldots, T$, with $T = 2010$. The fact that the observations used in Kviman et al. (2024) are thus the first differences of the log-returns rather than the (de-meaned) log-returns themselves may explain the unusually small values for the model parameters $\phi = 0.8$ and $\beta = 0.01$ which the authors obtained via a grid search based on these data. Nonetheless, to make our results comparable to those of Kviman et al. (2024), we similarly use (6).

D ADDITIONAL NUMERICAL RESULTS

D.1 Average squared Mahalanobis distance

Here, we report the average squared Mahalanobis distance of the SMC approximations of the marginal filtering and smoothing distributions for the three models whose results were not shown in Section 3.2 of the main manuscript.

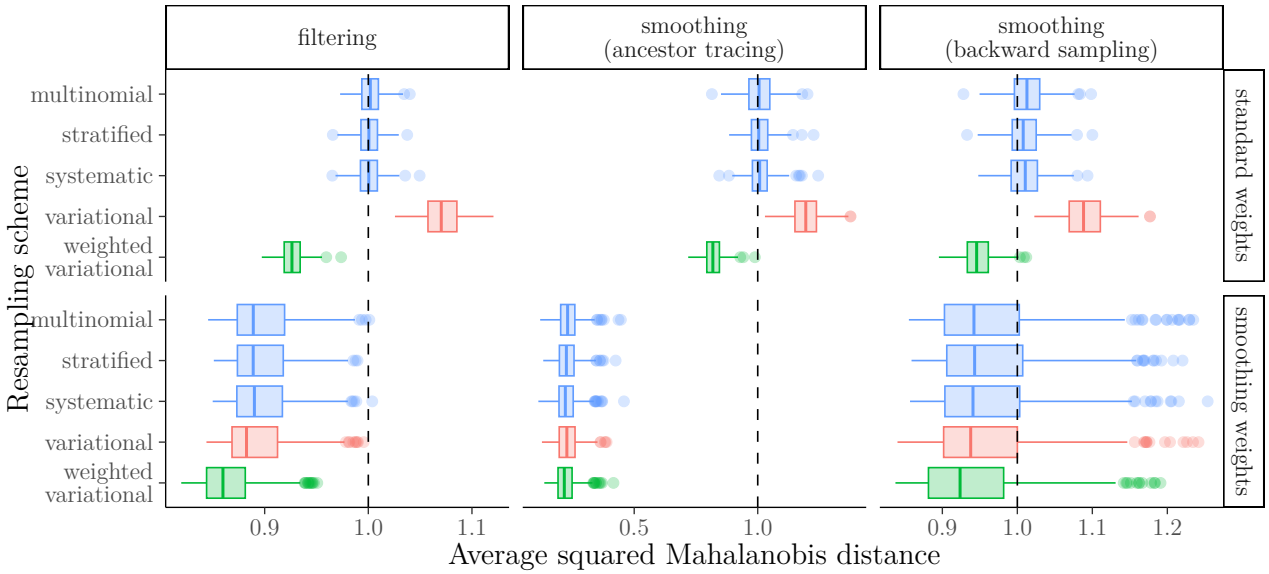

Figure 7: Average squared Mahalanobis distance of the approximations of the marginal filtering and smoothing distributions in the stochastic volatility model.

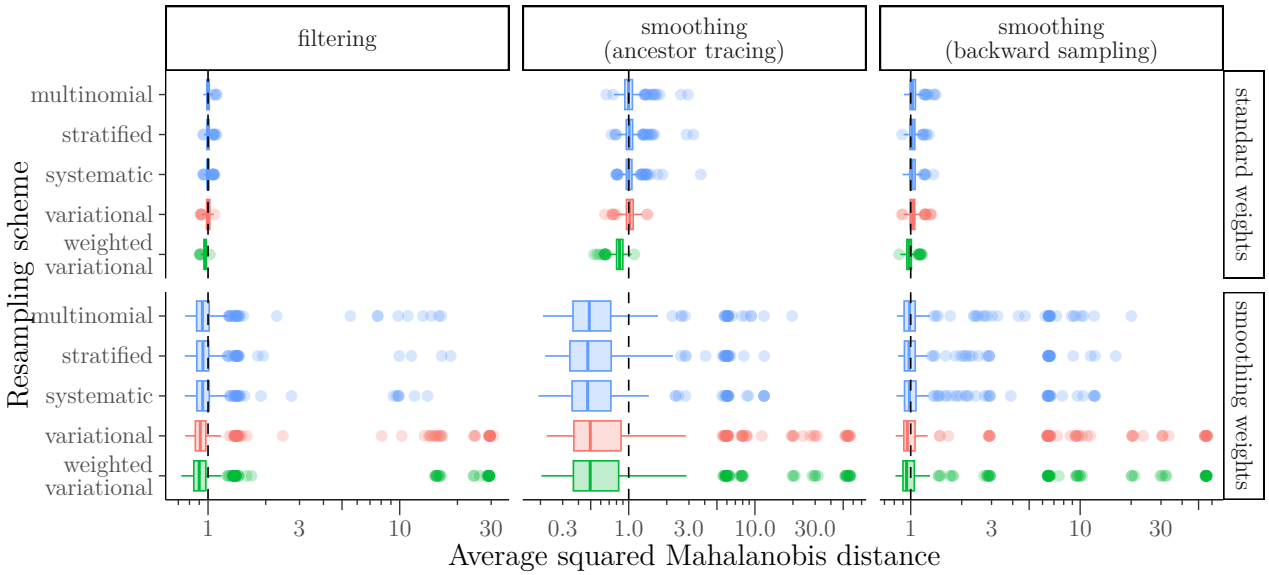

Figure 8: Average squared Mahalanobis distance of the approximations of the marginal filtering and smoothing distributions in the nonlinear state-space model. Note the log-scale on the first axis.

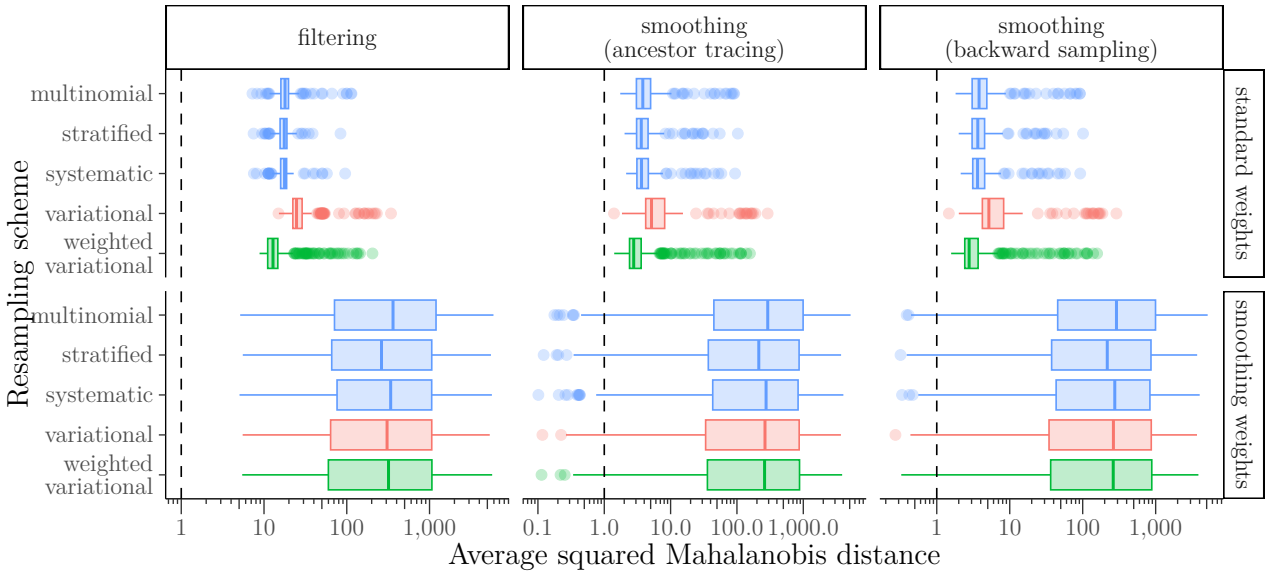

Figure 9: Average squared Mahalanobis distance of the approximations of the marginal filtering and smoothing distributions in the Lorenz-63 model. Note the log-scale on the first axis.

D.2 Average mean-square error

Here, we illustrate the average MSE of the SMC approximations of mean of the marginal filtering and smoothing distributions. Unsurprisingly, the MSE results are qualitatively very similar to those for the average squared Mahalanobis distance since the former can be calculated by simply taking the covariance matrices $\Sigma_{s|t}$ in the definition of the latter to be identity matrices of suitable sizes. However, the MSE is difficult to interpret: a small MSE may not indicate better performance because it could indicate an underestimation of posterior uncertainty. The MSE results are thus only included here for completeness and for comparison with Kviman et al. (2024).

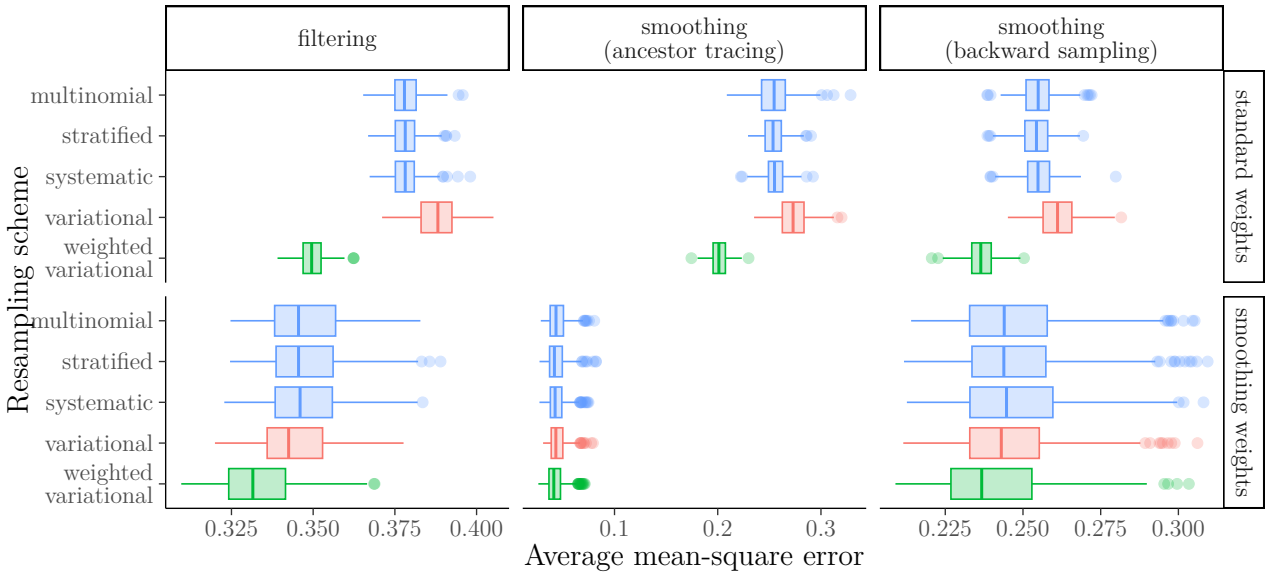

Figure 10: Average MSE of the estimates of the means of the marginal filtering and smoothing distributions in the linear-Gaussian state-space model.

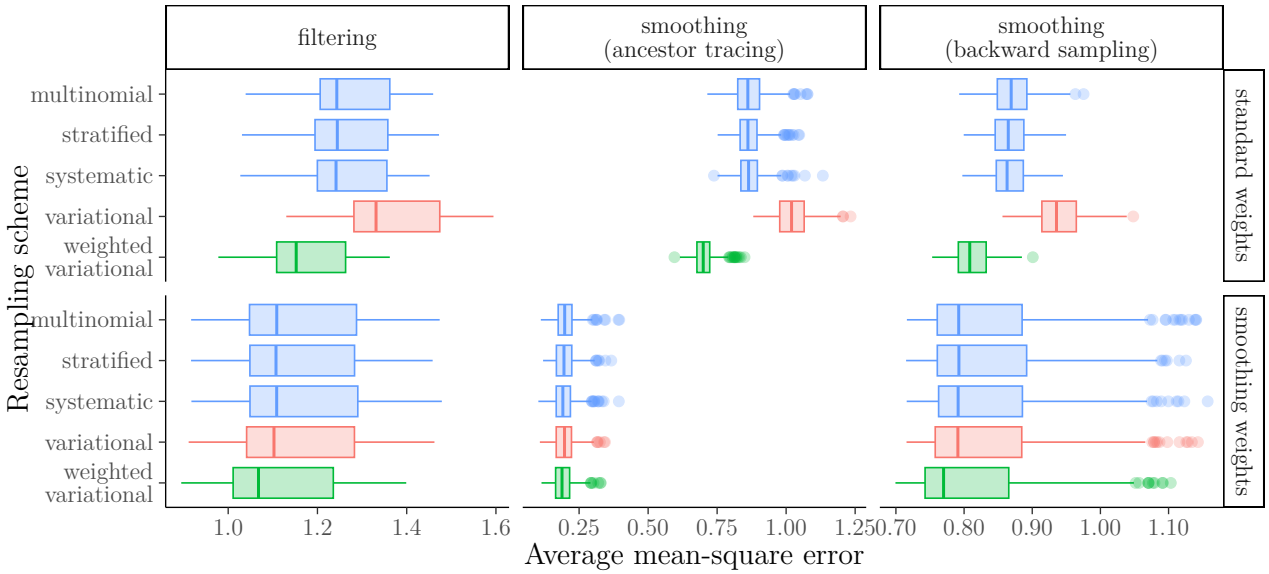

Figure 11: Average MSE of the estimates of the means of the marginal filtering and smoothing distributions in the stochastic volatility model.

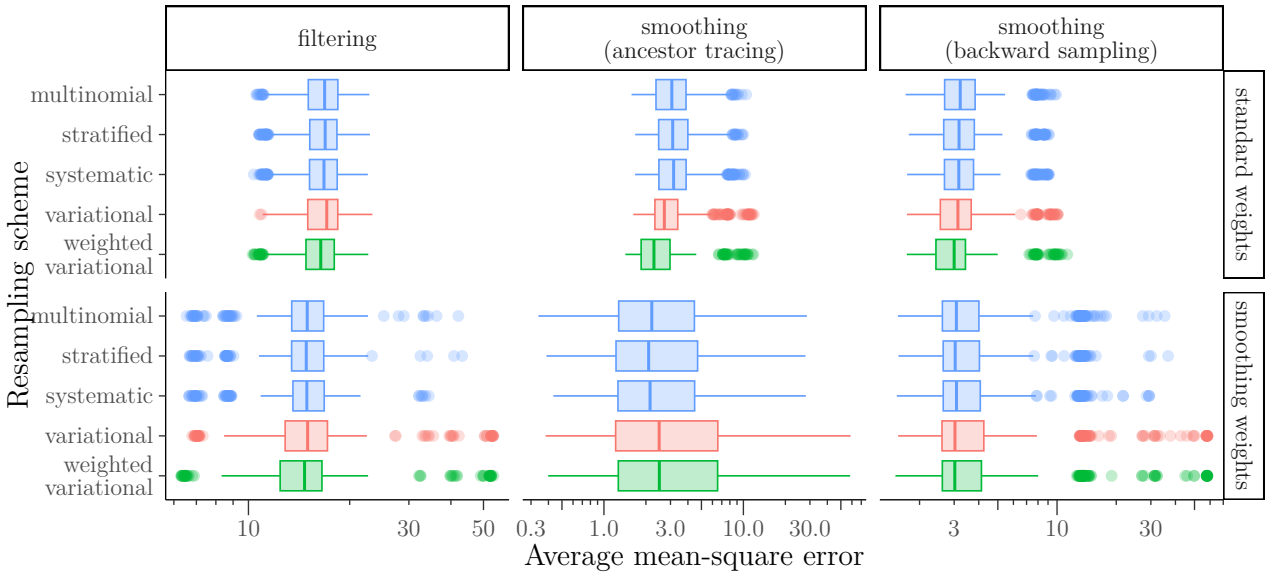

Figure 12: Average MSE of the estimates of the means of the marginal filtering and smoothing distributions in the nonlinear state-space model. Note the log-scale on the first axis.

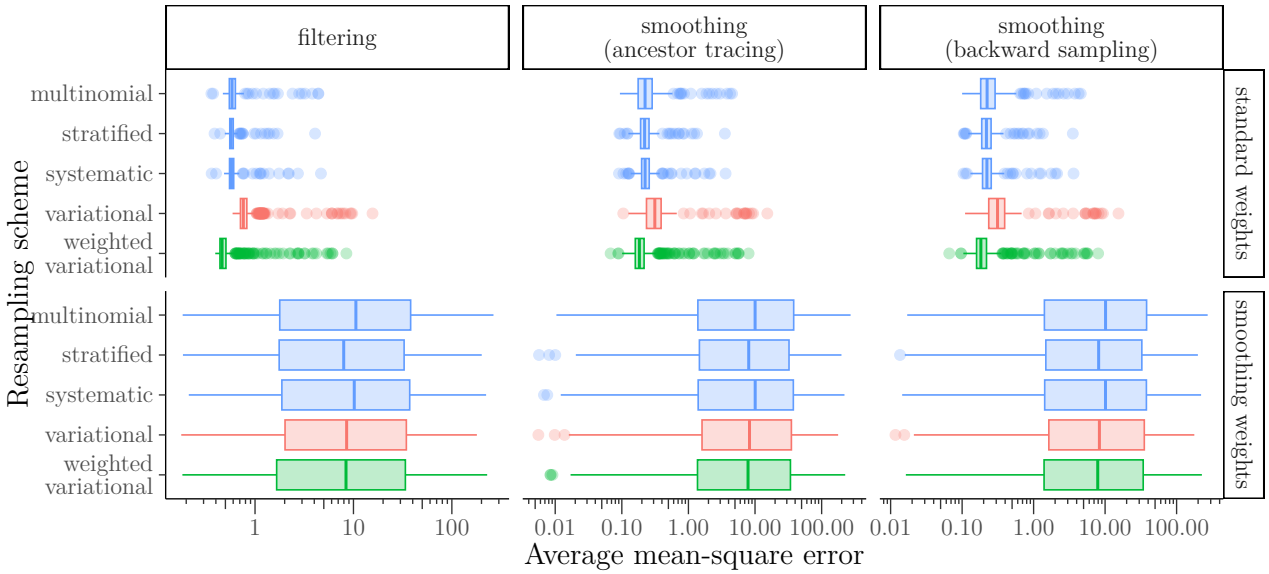

Figure 13: Average MSE of the estimates of the means of the marginal filtering and smoothing distributions in the Lorenz-63 model. Note the log-scale on the first axis.

D.3 Likelihood estimates for the real-data experiment

In this section, we give more detailed numerical results for the real-data example from Section 3.3 and illustrate why having a mean of $\log \widehat{\mathcal{Z}}_T$ that is below $\log \mathcal{Z}_T$ does not imply underestimation of $\mathcal{Z}_T$.

Due to the relatively small number of particles, the likelihood estimates seem to have a fairly heavy right-tail. For instance, as shown in Figure 6 (see also Tables 1 and 2, for multinomial, stratified and systematic resampling (without the 'smoothing weights'), the mean of $\log(\widehat{\mathcal{Z}}_T/\mathcal{Z}_T)$ is below 0 (and the median of $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$ is below 1). Yet, *the mean of* $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$ *is actually above* 1.

Table 1: Estimates of $\widehat{\mathcal{Z}}_T/\mathcal{Z}_T$ for the real-data experiment from Section 3.3. Note that according to these results, systematic and stratified resampling (with standard weights) perform best.

Resampling scheme	Resampling weights	Mean	Median	Std dev.
multinomial	standard weights	1.06	0.56	1.49
stratified		1.13	0.65	1.42
systematic		1.04	0.64	1.50
variational		52.03	32.50	58.58
weighted variational		10.18	6.16	13.55
multinomial	smoothing weights	0.55	0.28	0.93
stratified		0.55	0.31	0.79
systematic		0.52	0.30	0.67
variational		4.56	2.82	5.72
weighted variational		57.55	33.26	88.30

Table 2: Estimates of $\log(\widehat{\mathcal{Z}}_T/\mathcal{Z}_T)$ for the real-data experiment from Section 3.3.

Resampling scheme	Resampling weights	Mean	Median	Std dev.
multinomial	standard weights	-0.55	-0.59	1.11
stratified		-0.39	-0.43	1.00
systematic		-0.45	-0.44	0.98
variational		3.53	3.48	0.91
weighted variational		1.83	1.82	0.98
multinomial	smoothing weights	-1.27	-1.28	1.15
stratified		-1.17	-1.16	1.08
systematic		-1.18	-1.21	1.03
variational		1.01	1.04	1.01
weighted variational		3.49	3.50	1.06