Submitted to the Annals of Statistics

CONDITIONAL SEQUENTIAL MONTE CARLO IN HIGH DIMENSIONS

BY AXEL FINKE[1,a] AND ALEXANDRE H. THIERY[2,b]

[1]*Department of Mathematical Sciences, Loughborough University, UK* [a]*a.finke@lboro.ac.uk*

[2]*Department of Statistics and Applied Probability, National University of Singapore, Singapore* [b]*a.h.thiery@nus.edu.sg*

The *iterated conditional sequential Monte Carlo (i-CSMC)* algorithm from Andrieu, Doucet and Holenstein (2010) is an MCMC approach for efficiently sampling from the joint posterior distribution of the T latent states in challenging time-series models, e.g. in non-linear or non-Gaussian state-space models. It is also the main ingredient in *particle Gibbs* samplers which infer unknown model parameters alongside the latent states. In this work, we first prove that the i-CSMC algorithm suffers from a curse of dimension in the dimension of the states, D: it breaks down unless the number of samples ('particles'), N, proposed by the algorithm grows exponentially with D. Then, we present a novel 'local' version of the algorithm which proposes particles using Gaussian random-walk moves that are suitably scaled with D. We prove that this *iterated random-walk conditional sequential Monte Carlo (i-RW-CSMC)* algorithm avoids the curse of dimension: for arbitrary N, its acceptance rates and expected squared jumping distance converge to non-trivial limits as $D \to \infty$. If $T = N = 1$, our proposed algorithm reduces to a Metropolis–Hastings or Barker's algorithm with Gaussian random-walk moves and we recover the well known scaling limits for such algorithms.

1. Introduction.

1.1. *Summary.* This work analyses Monte Carlo methods for approximating the joint smoothing distribution (i.e. the joint distribution of all latent states) in high-dimensional state-space models. Developing efficient Markov chain Monte Carlo (MCMC) algorithms for this task is challenging if the dimension of the latent states, the 'spatial' dimension D, or the number of observations, the 'time horizon' T, is large because of the difficulty of finding good 'global' proposal distributions on a large (DT-dimensional) space. For this reason, the acceptance rate of *independent Metropolis–Hastings (MH)* kernels for this problem is typically $\mathrm{O}(\mathrm{e}^{-DT})$ which means that the algorithm suffers from a *'curse of dimension',* i.e. its complexity grows exponentially in the size (TD) of the problem. Throughout this work, we define *complexity* as the number of full likelihood evaluations needed to control the approximation error of a fixed-dimensional marginal of the joint smoothing distribution.

For the moment, assume that D is fixed and sufficiently small. In this scenario, the *iterated conditional sequential Monte Carlo (i-CSMC)* algorithm (Andrieu, Doucet and Holenstein, 2010; Chopin and Singh, 2015; Andrieu, Lee and Vihola, 2018) has become a popular Monte Carlo method for approximating the joint smoothing distribution. The algorithm is based around a *conditional sequential Monte Carlo (CSMC)* algorithm which builds a proposal distribution sequentially in the 'time' direction by propagating $N + 1$ Monte Carlo samples termed 'particles' over the T time steps. One of these lineages is set equal to the current state of the Markov chain and termed the *reference path.* At each time step, some of the remaining N particle lineages are pruned out if they are unlikely represent good proposals ('selection').

MSC2020 subject classifications: Primary 65C05; secondary 60J05, 65C35, 65C40.

Keywords and phrases: high dimensions; curse of dimension; Markov chain Monte Carlo; particle filter; state-space model.

The remaining particle lineages are multiplied and extended to the next time by sampling from the model dynamics ('mutation'). The selection steps prevent the algorithm from wasting computational effort on extending samples which are unlikely to form good proposals. This ensures that the complexity of the algorithm remains linear (and hence avoids a curse of dimension) in T. Specifically, this linear complexity is due to the fact that the number of particles needs to be scaled as $N \sim T$ (Andrieu, Lee and Vihola, 2018; Lindsten, Douc and Moulines, 2015; Del Moral, Kohn and Patras, 2016; Brown et al., 2021). Recently, Lee, Singh and Vihola (2020) showed that the use of an extension known as *backward sampling* (Whiteley, 2010) removes this need so that the overall complexity of the algorithm can be further reduced to $\mathrm{O}(1)$ (for fixed D), recalling that 'complexity' is the number of likelihood evaluations needed to control approximation errors of fixed-dimensional marginals of the joint smoothing distribution. Empirically, this has also been found to hold for a related extension called *ancestor sampling* (Lindsten, Jordan and Schön, 2012).

Due to this favourable scaling in T, the i-CSMC algorithm has become a popular tool for Bayesian inference in low-dimensional state-space models (and beyond). For instance, it is the main ingredient within so-called *particle Gibbs* samplers (Andrieu, Doucet and Holenstein, 2010) which infer unknown model parameters alongside the latent states.

Unfortunately, as we show in this work, the i-CSMC algorithm suffers from a curse of dimension in the 'spatial' dimension D of the latent states. That is, for any time horizon T, the algorithm breaks down if $\log(N) = \mathrm{o}(D)$ – i.e. unless the number of particles grows exponentially in D – and this cannot be overcome through the use of backward sampling.

The main contribution of this work is to propose a novel CSMC algorithm, called *random-walk conditional sequential Monte Carlo (RW-CSMC)* algorithm. In contrast to the (standard) CSMC algorithm, it scatters the particles *locally* around the reference path using Gaussian random-walk proposals whose variance is suitably scaled with D. The algorithm is incorporated into a larger *iterated random-walk conditional sequential Monte Carlo (i-RW-CSMC)* algorithm which again induces a Markov kernel that leaves the joint smoothing distribution invariant. We prove that this strategy overcomes the curse of dimension in D, i.e. in the sense that the expected squared jumping distance associated with any D-dimensional time-marginal distribution is stable as $D \to \infty$. In other words, for any fixed T, the algorithm has complexity $\mathrm{O}(D)$ (and the number of particles does not need to grow with D).

We also discuss the complexity in the time horizon T. Specifically, if the model factorises over time, we are able to verify that our proposed i-RW-CSMC algorithm has the same scaling as the i-CSMC algorithm. That is, without backward sampling, we may grow the number of particles as $N = CT$, for some constant $C > 0$, to guarantee an overall complexity $\mathrm{O}(TD)$. The use of backward sampling again removes the need for growing N with T so that the overall complexity can be brought down to $\mathrm{O}(D)$. Admittedly, the 'factorisation-over-time' assumption is strong. However, we conjecture that the above-described scaling in T holds more generally, i.e. – just as in the i-CSMC algorithm – this assumption is not necessary. As evidence for this, we present a slight modification of the i-RW-CSMC algorithm based around the *embedded hidden Markov model (EHMM)* method from Neal (2003); Neal, Beal and Roweis (2004), which we term the *random-walk embedded hidden Markov model (RW-EHMM)* algorithm. Without making the 'factorisation-over-time' assumption, we prove that this modified algorithm does not require scaling N with T.

Table 1 summarises the complexity of the algorithms discussed in this work.

1.2. *Related work.* Our work can be viewed as an extension of high-dimensional scaling limits of classical MCMC algorithms (e.g., Roberts, Gelman and Gilks, 1997). This is because if $N = T = 1$, the i-RW-CSMC update reduces to a MH (or to Barker's) kernel (Metropolis et al., 1953; Hastings, 1970; Barker, 1965) with a *random-walk* proposal. In

TABLE 1

Complexity of the algorithms in this work. 'Complexity' is defined as the number of likelihood evaluations needed to control approximation errors of fixed-dimensional marginals. The *-symbol indicates that the complexity in T is only proved in for models that factorise over time in this work.

	With backward sampling?	
	No	Yes
i-CSMC	$\mathrm{O}(T\mathrm{e}^D)$	$\mathrm{O}(\mathrm{e}^D)$
i-RW-CSMC*	$\mathrm{O}(TD)$	$\mathrm{O}(D)$
RW-EHMM	$\mathrm{O}(D)$	

contrast, the i-CSMC algorithm reduces to a MH (or again to Barker's) algorithm with an *independent* proposal ('independent' refers to the fact that the proposed value does not depend on the current state of the Markov chain) which is known to break down in high dimensions.

If $T = 1$ and $N > 1$, these algorithms can be viewed as a MH (or Barker's) kernel with *multiple* proposals. Such methods were introduced in the seminal works of Tjelmeland (2004); Neal (2003). Classical optimal scaling results were extended to a closely related class of MCMC algorithms with multiple proposals in Bédard, Douc and Moulines (2012).

We limit our analysis to the i-RW-CSMC algorithm. However, alternative ways of constructing (iterated) CSMC algorithms with local moves are possible. Indeed, our work was motivated by Shestopaloff and Neal (2018) who proposed the first such algorithm – which, incidentally, reduces to a MH (or Barker's) kernel with delayed acceptance (Christen and Fox, 2005) if $N = T = 1$. A generic framework which admits the i-CSMC algorithm, the i-RW-CSMC algorithm, and the method from Shestopaloff and Neal (2018) as special cases can be found in Finke, Doucet and Johansen (2016, Section 6).

We have recently become aware of Malory (2021, Chapter 4) who independently analyse a related class of iterated CSMC algorithms with exchangeable particle proposals that is likewise a special case of Finke, Doucet and Johansen (2016, Section 6). Our work distinguishes itself from theirs by, among others, the following contributions.

1. We provide formal proof that the standard i-CSMC algorithm breaks down in high dimensions, even with backward sampling.
2. Our dimensional-stability guarantees for the i-RW-CSMC algorithm hold even if the state-space model is dependent over time – Malory (2021) assume that the target distribution factorises into a product of independent marginals over time.
3. Our methodology and analysis permits a backward-sampling extension which is vital for performing inference for long time series.

1.3. *Contributions and structure.* This work is structured as follows.

Section 2 reviews the i-CSMC algorithm and shows that it generalises classical MCMC kernels with independent proposals. Our main result in this section is the following.

- *Proposition 2.2* proves that the i-CSMC algorithm suffers from a curse of dimension in the spatial dimension D and that this cannot be overcome with backward sampling.

Section 3 introduces the novel RW-EHMM algorithm as a preliminary solution to the curse-of-dimensionality problem and as a precursor to our proposed i-RW-CSMC algorithm. It does not employ resampling and therefore requires $\mathrm{O}(N^2)$ operations to implement rather than $\mathrm{O}(N)$ iterations. However, we introduce this algorithm here for didactic reasons because it is simple to understand and shares many features with our main i-RW-CSMC algorithm. For instance, both algorithms scatter particles around the reference path using the same Gaussian random-walk type proposals which are scaled suitably with D. Our main results in this section are the following.

- *Proposition 3.2* and Proposition 3.3 prove that the RW-EHMM algorithm has stable acceptance rates in high dimensions.
- *Proposition 3.4* establishes a non-trivial limit for the expected squared jumping distance in high dimensions.
- *Corollary 3.5* verifies that the number of particles does not need to be scaled with T.

Section 4 introduces our novel i-RW-CSMC algorithm, shows that it generalises classical MCMC kernels with Gaussian random-walk proposals, and proves that it avoids the curse of dimension. Our main results in this section are the following.

- *Propositions 4.1 and 4.2* show that the i-RW-CSMC algorithm can be viewed as a 'perturbed' version of the RW-EHMM algorithm.
- *Proposition 4.5* and Corollary 4.6 prove that the i-RW-CSMC algorithm has stable acceptance rates in high dimensions.
- *Proposition 4.7* establishes a non-trivial limit for the expected squared jumping distance in high dimensions.
- *Proposition 4.8* verifies that, under the additional assumption that the model is independent over time, without backward sampling, N must grow at least linearly in the time horizon T; with backward sampling, N does not need to scale with T. We conjecture that this result holds more generally, i.e. that the 'factorisation-in-time' assumption is not necessary.

Additionally, *Remark 4.4* explains that whilst the (standard) CSMC algorithm that underlies the i-CSMC algorithm is inextricably linked to the justification of standard 'unconditional' sequential Monte Carlo (SMC) counterpart, no such 'unconditional' SMC counterpart exists for the RW-CSMC algorithm that underlies the i-RW-CSMC algorithm.

Section 5 provides numerical illustration of our theoretical results. Most of our proofs and further materials can be found in the appendix. In particular, Appendix F extends the proposed methodology deal with unknown 'static' model parameters – either via a particle-Gibbs type update or via another novel MCMC kernel that is loosely related to correlated pseudo-marginal methods.

1.4. *Notation and conventions.* Let $(\Omega, \mathcal{A}, \mathbb{P})$ be some probability space and denote expectation with respect to $\mathbb{P}$ by $\mathbb{E}$. The symbol $\mathrm{N}(\mu, \Sigma)$ denotes a normal distribution with mean vector μ and covariance matrix Σ; δ_x is the point mass (Dirac measure) at x. Unless otherwise indicated, all (transition) densities mentioned in this work are absolutely continuous w.r.t. a suitable version of the Lebesgue measure.

For $n \in \mathbb{N}$, we often write $[n] \coloneqq \{1, 2, \ldots, n\}$ and $[n]_0 \coloneqq \{0, 1, 2, \ldots, n\}$ and we let $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{0}_n \in \mathbb{R}^n$ be a column vectors of length n whose entries are all 1 and 0, respectively. The symbol $\mathrm{I}_n \in \mathbb{R}^{n \times n}$ denotes the identity matrix.

Finally, for any $N \in \mathbb{N}$, $n \in [N]_0$ and any $h^{1:N} \in \mathbb{R}^N$, and with convention $h^0 \coloneqq 0$, we define the *Boltzmann selection function*

$$\Psi^n(\{h^m\}_{m=1}^N) \coloneqq \frac{\exp(h^n)}{1 + \sum_{m=1}^N \exp(h^m)}, \tag{1}$$

as well as the *Rosenbluth–Teller selection function*

$$\Phi^n(\{h^m\}_{m=1}^N) \coloneqq \begin{cases} \dfrac{\exp(h^n)}{1 + \sum_{m=1}^N \exp(h^m) - 1 \wedge \exp(h^n)}, & \text{if } n \in [N], \\ 1 - \sum_{m=1}^N \Phi^m(\{h^l\}_{l=1}^N), & \text{if } n = 0. \end{cases} \tag{2}$$

To sample from (2), we can propose $n \in [N]$ with probability $\exp(h^n) / \sum_{m=1}^N \exp(h^m)$ and return n with probability $1 \wedge \sum_{m=1}^N \exp(h^m) / \sum_{l \in [N]_0 \setminus \{n\}} \exp(h^l)$; otherwise, we return 0.

If $N = 1$, these selection functions reduce to the well-known acceptance functions of Barker's algorithm (Barker, 1965): $\Psi^1 = \exp/(1+\exp)$ and of the MH algorithm (Metropolis et al., 1953; Hastings, 1970): $\Phi^1 = 1 \wedge \exp$. The following Peskun-ordering type result (Peskun, 1973) (which follows immediately from the definition) shows that the Rosenbluth–Teller selection function induces a smaller rejection probability than the Boltzmann selection function.

LEMMA 1.1. *For any $N \in \mathbb{N}$ and $h^{1:N} \in \mathbb{R}^N$, $\Psi^0(\{h^m\}_{m=1}^N) \geq \Phi^0(\{h^m\}_{m=1}^N)$.* ◁

2. Existing methodology: the i-CSMC algorithm.

2.1. *Feynman–Kac model.* For the measurable space $(\mathbf{E}, \boldsymbol{\mathcal{E}}) := (\mathbb{R}^D, \mathcal{B}(\mathbb{R})^{\otimes D})$, let $\mathbf{M}_1 \in \mathcal{P}(\mathbf{E})$ be a probability measure with density $\mathbf{m}_1 \colon \mathbf{E} \to [0, \infty)$. Furthermore, for $t \geq 2$, let $\mathbf{M}_t \colon \mathbf{E} \times \boldsymbol{\mathcal{E}} \to [0, 1]$ be some Markov kernel with transition density $\mathbf{m}_t \colon \mathbf{E} \times \mathbf{E} \to [0, \infty)$. Furthermore, for $t \geq 1$, let $\mathbf{G}_t \colon \mathbf{E} \to (0, \infty)$ be strictly positive measurable potential functions. The methods discussed in this work target the following probability measure on $(\mathbf{E}_{T,D}, \boldsymbol{\mathcal{E}}_{T,D}) := (\mathbf{E}^T, \boldsymbol{\mathcal{E}}^{\otimes T})$:

$$\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T}) \propto \mathbf{M}_1(\mathrm{d}\mathbf{x}_1)\mathbf{G}_1(\mathbf{x}_1) \prod_{t=2}^{T} \mathbf{M}_t(\mathbf{x}_{t-1}, \mathrm{d}\mathbf{x}_t)\mathbf{G}_t(\mathbf{x}_t).$$

EXAMPLE (state-space model). Let $(\mathbf{X}_t, \mathbf{Y}_t)_{t \geq 1}$ be a Markov chain on a space $\mathbf{E} \times \mathbf{F}$ with initial distribution $\mathbf{M}_1(\mathrm{d}\mathbf{x}_1)\mathbf{H}_1(\mathbf{x}_1, \mathrm{d}\mathbf{y}_1)$ and transition kernels $\mathbf{M}_t(\mathbf{x}_{t-1}, \mathrm{d}\mathbf{x}_t)\mathbf{H}_t(\mathbf{x}_t, \mathrm{d}\mathbf{y}_t)$. Assume that for each time $t \in [T]$, we observe a realisation $\mathbf{y}_t \in \mathbf{F}$ of $\mathbf{Y}_t$ but $\mathbf{X}_t$ is unobserved ('latent'). We are then typically interested in computing (at least approximately) the posterior distribution of the latent 'states' $\mathbf{X}_{1:T}$, often called the *joint smoothing distribution*:

$$\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T}) = \mathbb{P}(\mathbf{X}_{1:T} \in \mathrm{d}\mathbf{x}_{1:T} | \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}).$$

Such a model, called *state-space model* or *(general-state) hidden Markov model*, can be viewed as a Feynman–Kac model by considering the observed values $\mathbf{y}_{1:T}$ to be 'fixed' (so that they can be dropped from the notation) and assuming that $\mathbf{G}_t(\mathbf{x}_t) = \mathbf{h}_t(\mathbf{x}_t, \mathbf{y}_t)$, where $\mathbf{h}_t(\mathbf{x}_t, \cdot)$ is a density of $\mathbf{H}_t(\mathbf{x}_t, \cdot)$ w.r.t. to a suitable dominating measure. ◁

Such models are routinely used in a wide variety of fields (Cappé, Moulines and Rydén, 2005). Unfortunately, with the exception of a few special cases (e.g. state-space models that are both linear and Gaussian) the distribution $\boldsymbol{\pi}_{T,D}$ is typically intractable and must be approximated, e.g. using MCMC methods. It is therefore crucial to design $\boldsymbol{\pi}_{T,D}$-invariant MCMC kernels that can efficiently deal with long time horizons (large T) and large 'spatial' dimension (large D) both of which are nowadays often found in the models of interest to practitioners (see, e.g., Van Leeuwen, 2009; Cressie and Wikle, 2015).

2.2. *Description of the algorithm.*

2.2.1. *Basic algorithm.* In the remainder of this section, we review the iterated conditional sequential Monte Carlo (i-CSMC) algorithm. For the moment, assume that the 'spatial' dimension D is fixed (and not too large). For such scenarios, the i-CSMC algorithm (Andrieu, Doucet and Holenstein, 2010) has become a popular way of constructing an efficient $\boldsymbol{\pi}_{T,D}$-invariant Markov kernel. Specifically, the algorithm employs a collection of N particles to construct a proposal for the entire state sequence sequentially in the 'time' direction. Compared to updating the latent state sequence via an independent MH kernel, this

strategy brings down the computational complexity from $\mathrm{O}(\mathrm{e}^T)$ to at most $\mathrm{O}(T)$ and thus avoids a curse of dimension in the time horizon T.

For any $t \in [T]$ and any $\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}$, define

$$\mathbf{w}_t(\mathbf{x}_t) \coloneqq \log \mathbf{G}_t(\mathbf{x}_t).$$

The lth update of the i-CSMC scheme is then outlined in Algorithm 1, where we use the convention that any action described for the nth particle index is to be performed conditionally independently for all $n \in [N]$. We also use the convention that any quantity with time index $t < 1$ is to be ignored, e.g. so that $\mathbf{M}_1(\mathbf{z}_0^n, \cdot) \equiv \mathbf{M}_1(\cdot)$.

ALGORITHM 1 (i-CSMC). Given $\mathbf{x}_{1:T} \coloneqq \mathbf{x}_{1:T}[l] \in \mathbf{E}_{T,D}$.

1. For $t \in [T]$,
   a) if $t > 1$,
      i. set $A_{t-1}^0 = a_{t-1}^0 \coloneqq 0$,
      ii. sample $A_{t-1}^n = a_{t-1}^n \in [N]_0$ with probability
      $$\Psi^{a_{t-1}^n}(\{\mathbf{w}_{t-1}(\mathbf{z}_{t-1}^m) - \mathbf{w}_{t-1}(\mathbf{z}_{t-1}^0)\}_{m=1}^N) = \frac{\mathbf{G}_{t-1}(\mathbf{z}_{t-1}^{a_{t-1}^n})}{\sum_{m=0}^N \mathbf{G}_{t-1}(\mathbf{z}_{t-1}^m)},$$
   b) set $\mathbf{Z}_t^0 = \mathbf{z}_t^0 \coloneqq \mathbf{x}_t$ and sample $\mathbf{Z}_t^n = \mathbf{z}_t^n \sim \mathbf{M}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \cdot)$.
2. Sample $K_T = k_T \in [N]_0$ with probability
$$\Psi^{k_T}(\{\mathbf{w}_T(\mathbf{z}_T^m) - \mathbf{w}_T(\mathbf{z}_T^0)\}_{m=1}^N) = \frac{\mathbf{G}_T(\mathbf{z}_T^{k_T})}{\sum_{m=0}^N \mathbf{G}_T(\mathbf{z}_T^m)}.$$
3. Set $K_t = k_t \coloneqq a_t^{k_{t+1}}$, for $t = T-1, \ldots, 1$.
4. Set $\mathbf{X}'_{1:T} \coloneqq \mathbf{x}'_{1:T} \coloneqq (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T})$.
5. Return $\mathbf{x}_{1:T}[l+1] \coloneqq \mathbf{x}'_{1:T}$.

Step 1 of Algorithm 1 which a) performs (conditional) multinomial resampling by drawing the ancestor indices A_t^n; and b) generates the particles $\mathbf{Z}_t^n$, is known as the *conditional sequential Monte Carlo (CSMC)* algorithm.

The following running example illustrates how the algorithms discussed in this work reduce to versions of well known classical MCMC kernels if $N = T = 1$.

EXAMPLE (classical MCMC kernels). If $T = 1$ and $N = 1$, the target distribution is given by $\boldsymbol{\pi}_{1,D}(\mathrm{d}\mathbf{x}_1) \propto \mathbf{M}_1(\mathrm{d}\mathbf{x}_1)\mathbf{G}_1(\mathbf{x}_1)$ and Algorithm 1 proposes $\mathbf{Z}_1^1 = \mathbf{z}_1^1 \sim \mathbf{M}_1$ and accepts this proposal as the new state of the Markov chain with probability

$$\Psi^1(\mathbf{w}_1(\mathbf{z}_1^1) - \mathbf{w}_1(\mathbf{z}_1^0)) = \frac{\mathbf{G}_1(\mathbf{z}_1^1)}{\mathbf{G}_1(\mathbf{z}_1^0) + \mathbf{G}_1(\mathbf{z}_1^1)},$$

where $\mathbf{z}_1^0 = \mathbf{x}_1$. This can be recognised as a version of Barker's kernel (Barker, 1965) with independence proposal $\mathbf{M}_1$ (in the sense that the proposed state does not depend on the current state). ◁

EXAMPLE (multi-proposal MCMC kernels). If $T = 1$ but $N > 1$, Algorithm 1 (termed *conditional sampling–importance resampling* in Andrieu, Lee and Vihola 2018) reduces to an MCMC algorithm with multiple proposals (all being independent of each other and of the current state of the Markov chain). Multi-proposal MCMC algorithms were introduced in Neal (2003); Tjelmeland (2004); Frenkel (2004); Delmas and Jourdain (2009); Yang et al. (2018); Schwedes and Calderhead (2018) analyse Rao–Blackwellisation strategies for reusing all N proposed samples to estimate expectations of interest. ◁

2.2.2. *Extensions.*

Forced move. To reduce the probability of sampling $K_T = k_T = 0$ in Step 2 of Algorithm 1 (and hence improve the i-CSMC kernel in the Peskun order – see Lemma 1.1) Chopin and Singh (2015) proposed to replace the Boltzmann selection function in Step 2 by the Rosenbluth–Teller selection function, i.e. instead sample $K_T = k_T \neq 0$ with probability

$$\Phi^{k_T}(\{\mathbf{w}_T(\mathbf{z}_T^m) - \mathbf{w}_T(\mathbf{z}_T^0)\}_{m=1}^N) = \frac{\mathbf{G}_T(\mathbf{z}_T^{k_T})}{\sum_{m=1}^N \mathbf{G}_T(\mathbf{z}_T^m) - \mathbf{G}_T(\mathbf{z}_T^{k_T}) \wedge \mathbf{G}_T(\mathbf{z}_T^0)}.$$

This so called *forced move* approach can be recognised as an application of the *modified discrete-state Gibbs sampler* kernel from Liu (1996). See also Tjelmeland (2004) for an iterative algorithm for optimising the selection function.

EXAMPLE (classical MCMC kernels, continued). With the forced-move extension, Step 2 of Algorithm 1 accepts $\mathbf{Z}_1^1 = \mathbf{z}_1^1 \sim \mathbf{M}_1$ with probability

$$\Phi^1(\mathbf{w}_1(\mathbf{z}_1^1) - \mathbf{w}_1(\mathbf{z}_1^0)) = 1 \wedge \frac{\mathbf{G}_1(\mathbf{z}_1^1)}{\mathbf{G}_1(\mathbf{z}_1^0)},$$

where $\mathbf{z}_1^0 = \mathbf{x}_1$. This can be recognised as a version of an independent MH kernel (Metropolis et al., 1953; Hastings, 1970). ◁

Backward sampling. Steps 2 and 3 of Algorithm 1 sample a final-time particle index K_T and then trace back its ancestral lineage. This limits the new state $\mathbf{x}_{1:T}[l+1]$ to one of the $N+1$ particle lineages generated under the CSMC algorithm in Step 1 which often coalesce with the old reference path $\mathbf{x}_{1:T}[l]$. To ensure good mixing, we must therefore control the probability of such coalescence events by growing N linearly with T. This can be costly if T is large. To circumvent this problem, the *backward-sampling* extension (Whiteley, 2010) instead samples $K_t = k_t \in [N]_0$ in Step 3 with probability

$$\Psi^{k_t}(\{\mathbf{v}_t(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}}) - \mathbf{v}_t(\mathbf{z}_t^0, \mathbf{z}_{t+1}^{k_{t+1}})\}_{m=1}^N) = \frac{\mathbf{G}_t(\mathbf{z}_t^{k_t})\mathbf{m}_{t+1}(\mathbf{z}_t^{k_t}, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^N \mathbf{G}_t(\mathbf{z}_t^m)\mathbf{m}_{t+1}(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}})},$$

for $t = T-1, \ldots, 1$, where we have defined

$$\mathbf{v}_t(\mathbf{x}_{t:t+1}) \coloneqq \mathbf{w}_t(\mathbf{x}_t) + \log \mathbf{m}_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1}).$$

Lee, Singh and Vihola (2020) show that backward sampling allows us to keep N constant in T thus reducing the complexity of the algorithm from $\mathrm{O}(T)$ to $\mathrm{O}(1)$. A closely related method, *ancestor sampling,* was proposed in Lindsten, Jordan and Schön (2012).

Further extensions. Step 1 of Algorithm 1 proposes particles from the 'prior' $\mathbf{M}_t(\mathbf{x}_{t-1}, \cdot)$ and draws the parent indices A_{t-1}^n via (conditional) multinomial resampling. Other proposal kernels (Doucet, Godsill and Andrieu, 2000), other resampling schemes (Douc, Cappé and Moulines, 2005) or even auxiliary particle filter ideas (Pitt and Shephard, 1999; Shestopaloff and Doucet, 2019) could be employed. However, since none of these extensions overcome the curse of dimension proved below, we refrain from including them here to keep the presentation simple.

2.2.3. *Induced $\boldsymbol{\pi}_{T,D}$-invariant Markov kernel.* Given $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, let

$$\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^N(\mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T})$$

be the law of all the random variables $(\mathbf{Z}_{1:T}, A_{1:T-1}, K_{1:T}, \mathbf{X}'_{1:T})$ generated in Steps 1–4 of Algorithm 1 (with or without the forced-move extension and with or without backward sampling). Appendix B.1 gives a formal definition of this law.

Let $\mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}$ denote expectation w.r.t. $\mathbb{P}^N_{T,D,\mathbf{x}_{1:T}}$. Algorithm 1 induces a Markov kernel

$$\mathbf{P}^N_{T,D}(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) := \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}],$$

for $(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) \in \mathbf{E}_{T,D} \times \boldsymbol{\mathcal{E}}_{T,D}$. The following proposition shows that this Markov kernel leaves $\boldsymbol{\pi}_{T,D}$ invariant. It was proved Andrieu, Doucet and Holenstein (2010) for the basic algorithm and in Chopin and Singh (2015); Whiteley (2010) for the forced-move and backward-sampling extensions. For completeness, we provide an alternative, simple proof in Appendix B.2.

PROPOSITION 2.1. *For any $N, T, D \in \mathbb{N}$, $\boldsymbol{\pi}_{T,D}\mathbf{P}^N_{T,D} = \boldsymbol{\pi}_{T,D}$.* ◁

For any $t \in [T]$, we call

$$\alpha^N_{T,D,\mathbf{x}_{1:T}}(t) := \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_t \neq 0\}]$$

the *acceptance rate at time* t associated with Algorithm 1. This name is justified because $K_t = 0$ in Algorithm 1 implies $\mathbf{x}'_t = \mathbf{x}_t$, i.e. $\mathbf{x}_t[l+1] = \mathbf{x}_t[l]$.

2.3. *Curse of dimension.*

2.3.1. *High-dimensional regime.* We now prove that the i-CSMC algorithm suffers from a curse of dimension. This is established for a special case of the Feynman–Kac model from Section 2.1 which factorises into D independent and identically distributed (IID) 'spatial' components. Most other theoretical results in this work will be established under this regime. However, we stress that none of the algorithms discussed in this work are limited to this IID setting.

A1 The mutation kernels and potential functions factorise as

$$\mathbf{M}_t(\mathbf{x}_{t-1}, \mathrm{d}\mathbf{x}_t) = \prod_{d=1}^{D} M_t(x_{t-1,d}, \mathrm{d}x_{t,d}) \quad \text{and} \quad \mathbf{G}_t(\mathbf{x}_t) = \prod_{d=1}^{D} G_t(x_{t,d}),$$

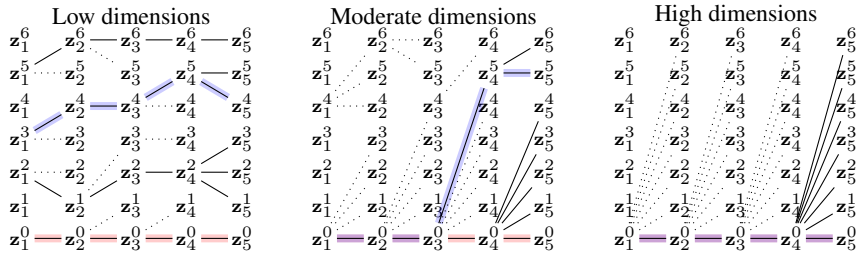
with the convention that any quantity with time index 0 is to be ignored and where

- $\mathbf{x}_t = x_{t,1:D} \in \mathbf{E}$, recalling that $(\mathbf{E}, \boldsymbol{\mathcal{E}}) = (\mathbb{R}^D, \mathcal{B}(\mathbb{R})^{\otimes D})$;
- $M_1 \in \mathcal{P}(\mathbb{R})$ is a probability measure with density $m_1 \colon \mathbb{R} \to [0, \infty)$ and, for $t \geq 2$, $M_t \colon \mathbb{R} \times \mathcal{B}(\mathbb{R}) \to [0, 1]$ is a Markov kernel with transition density $m_t \colon \mathbb{R}^2 \to [0, \infty)$;
- $G_t \colon \mathbb{R} \to (0, \infty)$ is a strictly positive and measurable potential function. ◁

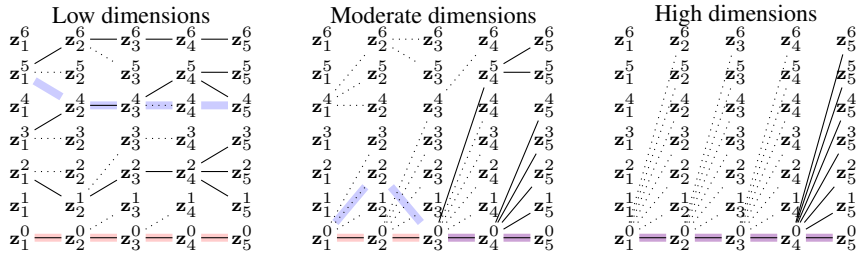
Thus, under **A1**, $\boldsymbol{\pi}_{T,D} = \pi_T^{\otimes D}$, with the following probability measure on $\mathbb{R}^T$:

$$\pi_T(\mathrm{d}x_{1:T}) \propto M_1(\mathrm{d}x_1)G_1(x_1) \prod_{t=2}^{T} M_t(x_{t-1}, \mathrm{d}x_t)G_t(x_t).$$

2.3.2. *Convergence to a degenerate limit.* We now show that in high ('spatial') dimensions, the law of genealogies under the i-CSMC algorithm converges to limit that is degenerate in the sense that all particle lineages immediately coalesce with the reference path so that all acceptance probabilities are zero. Once could naïvely hope that backward sampling circumvents this problem because it draws a new reference path that is not confined to one of the $N + 1$ surviving lineages. Unfortunately, our analysis shows that backward sampling, too, returns the old reference path in high dimensions. Typical behaviour of the genealogies is illustrated in Figure 1.



(a) Without backward sampling.



(b) With backward sampling.

FIG. 1. Breakdown of the i-CSMC algorithm in high dimensions. Black lines represent particle lineage induced by the algorithm, i.e. a line connects $\mathbf{z}_{t-1}^m$ and $\mathbf{z}_t^n$ iff $a_{t-1}^n = m$. Solid lines (——) represent the surviving lineages at time $T = 5$. Dotted lines (·····) represent lineages that have died out. The red line (▬) represents the old reference path $\mathbf{x}_{1:T}[l] = (\mathbf{z}_1^0, \ldots, \mathbf{z}_T^0)$. The blue line (▬) represents the new reference path $\mathbf{x}_{1:T}[l+1] = (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_t})$.

The degenerate limit of the law of the genealogies and the indices of the new reference path under the i-CSMC algorithm is the law $\mathbb{P}_T^N(\mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T})$ which deterministically sets all ancestor indices and all indices of the new reference path to 0. More formally,

$$\mathbb{P}_T^N(\mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T}) := \delta_0^{\otimes((N+1)(T-1)+T)}(\mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T}).$$

We let $\mathbb{E}_T^N$ denote expectation w.r.t. this law. The proof that the law of the genealogies and new reference path indices indeed converges to this trivial limit will be given below in Proposition 2.2 which relies on the following assumptions, where $\mathbb{E}$ denotes expectation w.r.t. $X_{1:T} \sim \pi_T$ and where

$$r_{t|T} := \mathbb{E}[\log G_t(X_t)] - \mathbb{E}[\log M_t(G_t)(X_{t-1})],$$

$$b_{t|T} := \mathbb{E}[\log G_t(X_t) + \log m_{t+1}(X_t, X_{t+1})] - \mathbb{E}[\log M_t(G_t m_{t+1}(\cdot, X_{t+1}))(X_{t-1})].$$

A2 $\inf_{t\in[T]} r_{t|T} =: \underline{r}_T > 0.$ ◁

A3 $\inf_{t\in[T-1]} b_{t|T} =: \underline{b}_T > 0.$ ◁

Assumptions **A2** and **A3** are not restrictive: a) they do not depend on multiplication of G_t by some positive constant; b) they automatically hold if the model factorises over time (see Assumption **A4** in Section 4.2.3) unless the potentials G_t are almost-everywhere constant; c) Appendix B.3 shows that they hold even in a simple linear-Gaussian state-space model.

We now state our first main result, Proposition 2.2 (proved in Appendix B.4), which shows that in high dimensions, all particle lineages coalesce immediately with the reference path unless in number of particles, $N + 1$, $N = N(D)$ grows exponentially in the spatial dimension D – even with the backward-sampling or forced-move extensions. Let $\|\cdot\|$ denote the total variation distance.

PROPOSITION 2.2 (curse of dimension). *Let $T \in \mathbb{N}$. Assume **A1** and **A2** (as well as **A3** if backward sampling is used) and write*

$$d^N_{T,D,\mathbf{x}_{1:T}} := \left\| \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{(A_{1:T-1}, K_{1:T}) \in \cdot\}] - \mathbb{E}^N_T[\mathbb{I}\{(A_{1:T-1}, K_{1:T}) \in \cdot\}] \right\|.$$

Then there exists a family $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ with $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$ and

$$\log N = \mathrm{o}(D) \quad \Longrightarrow \quad \lim_{D\to\infty} \sup_{\mathbf{x}_{1:T} \in \mathbf{F}_{T,D}} d^N_{T,D,\mathbf{x}_{1:T}} = 0. \qquad \triangleleft$$

An immediate consequence of Proposition 2.2 is the following corollary which shows that all acceptance rates vanish in high dimensions.

COROLLARY 2.3. *Under the assumptions of Proposition 2.2 and with the same sets $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$, for any $t \in [T]$:*

$$\log N = \mathrm{o}(D) \quad \Longrightarrow \quad \lim_{D\to\infty} \sup_{\mathbf{x}_{1:T} \in \mathbf{F}_{T,D}} \alpha^N_{T,D,\mathbf{x}_{1:T}}(t) = 0. \qquad \triangleleft$$

3. Simplified dimensionally stable methodology: the RW-EHMM algorithm.

3.1. *Description of the algorithm.* Our novel i-RW-CSMC algorithm will be presented in the next section. To ease the exposition, we first – in this section – introduce another novel algorithm, the *random-walk embedded hidden Markov model (RW-EHMM)* algorithm which is a simplified version of the i-RW-CSMC algorithm and is likewise stable in high dimensions. However, the implementation of a single RW-EHMM update requires $\mathrm{O}(N^2T)$ operations whilst a single i-RW-CSMC update only requires $\mathrm{O}(NT)$ operations.

3.1.1. *Basic algorithm.* The *RW-EHMM* algorithm proposed in this section also induces a $\boldsymbol{\pi}_{T,D}$-invariant Markov kernel. It can viewed as an instance of the *embedded hidden Markov model (EHMM)* methods from Neal (2003); Neal, Beal and Roweis (2004) which are closely related to iterated CSMC methods with backward sampling as explained in Finke, Doucet and Johansen (2016). The main difference between iterated CSMC and EHMM methods is that the former use resampling steps to permit implementation in $\mathrm{O}(NT)$ operations wheras the latter typically require $\mathrm{O}(N^2T)$ operations.

The lth update of the novel RW-EHMM scheme is outlined in Algorithm 2 where we use the convention that any action described for the nth particle index is to be performed conditionally independently for all $n \in [N]$ and any action described for the dth 'spatial' component is to be performed conditionally independently for all $d \in [D]$.

ALGORITHM 2 (RW-EHMM). Given $\mathbf{x}_{1:T} := \mathbf{x}_{1:T}[l] \in \mathbf{E}_{T,D}$.

1. For $t \in [T]$: set $\mathbf{Z}^0_t = \mathbf{z}^0_t := \mathbf{x}_t[l]$ and sample $\mathbf{Z}^{1:N}_t = \mathbf{z}^{1:N}_t$ as follows:
   a) sample $U^{1:N}_{t,d} \sim \mathrm{N}(0, \Sigma)$,
   b) set $z^n_{t,d} := z^0_{t,d} + \sqrt{\ell_t/D}\, U^n_{t,d}$,
   c) set $\mathbf{z}^n_t := z^n_{t,1:D}$.
2. Sample $K_{1:T} = k_{1:T} \in [N]^T_0$ with probability
$$\xi_T(\mathbf{z}_{1:T}, \{k_{1:T}\}) := \frac{\boldsymbol{\pi}_{T,D}(\mathbf{z}^{k_1}_1, \ldots, \mathbf{z}^{k_T}_T)}{\sum_{l_{1:T} \in [N]^T_0} \boldsymbol{\pi}_{T,D}(\mathbf{z}^{l_1}_1, \ldots, \mathbf{z}^{l_T}_T)}.$$
3. Set $\mathbf{X}'_{1:T} := \mathbf{x}'_{1:T} := (\mathbf{z}^{k_1}_1, \ldots, \mathbf{z}^{k_T}_T)$.
4. Return $\mathbf{x}_{1:T}[l+1] := \mathbf{x}'_{1:T}$.

Step 1 of Algorithm 2 scatters particles around the reference particle $\mathbf{z}^0_t = \mathbf{x}_t$ by adding correlated Gaussian noise independently in each dimension $d \in [D]$:

$$(Z^1_{t,d}, \ldots, Z^N_{t,d})^{\mathrm{T}} \sim \mathrm{N}\big(\mathbf{1}_N z^0_{t,d}, \tfrac{\ell_t}{D}\Sigma\big), \tag{3}$$

where $\mathbf{1}_N$ is a vector of 1s of length N and where

- $\Sigma := \frac{1}{2}(\mathbf{1}_N \mathbf{1}_N^{\mathrm{T}} + \mathrm{I}_N)$ is an $(N \times N)$ covariance matrix with 1 on the diagonal and $\frac{1}{2}$ everywhere else which governs the correlation between (univariate spatial components of) different particles,
- $\ell_t > 0$ is some scale factor that governs how far (on average) particles are scattered around the reference path.

This proposal was introduced by Tjelmeland (2004) who noted that

- the marginal distributions of individual particles are simply Gaussian random-walk moves with variance ℓ_t/D, i.e. $\mathbf{Z}_t^n \sim \mathrm{N}(\mathbf{z}_t^0, \frac{\ell_t}{D}\mathrm{I}_D)$, for $n \in [N]$;
- (3) can be viewed as first sampling a new 'centre' $Z'_{t,d} = z'_{t,d} \sim \mathrm{N}(z^0_{t,d}, \frac{\ell_t}{2D})$ and then sampling $Z^1_{t,d}, \dots, Z^N_{t,d} \overset{\text{IID}}{\sim} \mathrm{N}(z'_{t,d}, \frac{\ell_t}{2D})$:

$$\int_{-\infty}^{\infty} \mathrm{N}(z'; z^0_{t,d}, \tfrac{\ell_t}{2D}) \left[\prod_{n=1}^{N} \mathrm{N}(z^n_{t,d}, z', \tfrac{\ell_t}{2D})\right] \mathrm{d}z' = \mathrm{N}\big(z^{1:N}_{t,d}; \mathbf{1}_N z^0_{t,d}, \tfrac{\ell_t}{D}\Sigma\big).$$

Other types of 'local' proposals (i.e. not necessarily based on Gaussian random walks) could be used. However, we limit our analysis to this particular structure because it is *symmetric* in the sense that its density cancels out in the selection functions. More formally, letting λ denote a suitable version of the Lebesgue measure, $z^{-n}_{t,d} := (z^0_{t,d}, \dots, z^{n-1}_{t,d}, z^{n+1}_{t,d}, \dots, z^N_{t,d})$, and $\mathbf{z}_t^{-n} := (z^{-n}_{t,1}, \dots, z^{-n}_{t,D})$, the proposal $S^N_{t,D}(\mathbf{z}^n_t, \mathrm{d}\mathbf{z}_t^{-n}) := \prod_{d=1}^{D} \mathrm{N}(\mathrm{d}z^{-n}_{t,d}; \mathbf{1}_N z^n_{t,d}, \frac{\ell_t}{D}\Sigma)$ induced by Step 1 of Algorithm 2 satisfies:

$$\lambda(\mathrm{d}\mathbf{z}^j_t) S^N_{t,D}(\mathbf{z}^j_t, \mathrm{d}\mathbf{z}_t^{-j}) = \lambda(\mathrm{d}\mathbf{z}^k_t) S^N_{t,D}(\mathbf{z}^k_t, \mathrm{d}\mathbf{z}_t^{-k}), \qquad \text{for any } j, k \in [N]_0. \tag{4}$$

3.1.2. *Implementation in* $\mathrm{O}(N^2T)$ *operations.* Even though Step 2 of Algorithm 2 requires sampling from the distribution $\xi_T(\mathbf{z}_{1:T}, \cdot)$ whose support is $(N+1)^T$-dimensional, Neal (2003) recognised that this can be achieved in $\mathrm{O}(N^2T)$ operations as follows:

1. *Forward filtering.* For $t = 1, \dots, T$, compute (with convention $w^n_0 := 1$):

$$w^n_t := \sum_{m=0}^{N} \frac{w^m_{t-1}}{\sum_{l=0}^{N} w^l_{t-1}} \mathbf{m}_t(\mathbf{z}^m_{t-1}, \mathbf{z}^n_t) \mathbf{G}_t(\mathbf{z}^n_t). \tag{5}$$

2. *Backward sampling.* For $t = T, \dots, 1$ (with convention $\mathbf{m}_{T+1} \equiv 1$), sample $K_t = k_t \in [N]_0$ with probability

$$\frac{w^{k_t}_t \mathbf{m}_{t+1}(\mathbf{z}^{k_t}_t, \mathbf{z}^{k_{t+1}}_{t+1})}{\sum_{n=0}^{N} w^n_t \mathbf{m}_{t+1}(\mathbf{z}^n_t, \mathbf{z}^{k_{t+1}}_{t+1})}. \tag{6}$$

To provide additional intuition for these recursions, Appendix A shows that the particles $\mathbf{Z}_{1:T}$ "discretise" the model into an $(N+1)$-state HMM and that (5) and (6) can be viewed as the forward-filtering and backward-sampling recursions that sample from the joint posterior distribution of the states under this HMM.

3.1.3. *Induced* $\boldsymbol{\pi}_{T,D}$*-invariant Markov kernel.* Given $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, let

$$\widetilde{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T}),$$

be the law of all the random variables $(\mathbf{Z}_{1:T}, K_{1:T}, \mathbf{X}'_{1:T})$ generated in Steps 1–3 of Algorithm 2. Appendix C.1 gives a formal definition of this law.

Let $\widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}$ denote expectation w.r.t. $\widetilde{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}}$. Algorithm 2 induces a Markov kernel

$$\widetilde{\mathbf{P}}^N_{T,D}(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) := \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}],$$

for $(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) \in \mathbf{E}_{T,D} \times \boldsymbol{\mathcal{E}}_{T,D}$. The following proposition shows that this Markov kernel leaves $\boldsymbol{\pi}_{T,D}$ invariant.

PROPOSITION 3.1. *For any* $N, T, D \in \mathbb{N}$, $\boldsymbol{\pi}_{T,D}\widetilde{\mathbf{P}}^N_{T,D} = \boldsymbol{\pi}_{T,D}$. ◁

As in Neal (2003); Neal, Beal and Roweis (2004) this can be proved by noting that Algorithm 2 (in a slightly generalised form outlined at the beginning of Appendix C.1) targets the extended distribution:

$$\tilde{\boldsymbol{\pi}}_{T,D}(\mathrm{d}\mathbf{x}_{1:T} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{z}_{1:T}) = \boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T})\,\mathrm{Unif}_{[N]_0^T}(\mathrm{d}k_{1:T}) \prod_{t=1}^{T} \delta_{\mathbf{x}_t}(\mathbf{z}_t^{k_t}) S^N_{t,D}(\mathbf{z}_t^{k_t}; \mathrm{d}\mathbf{z}_t^{-k_t}),$$

where Unif_A is the uniform distribution on a set A. Step 1 of Algorithm 2 then samples from $\tilde{\boldsymbol{\pi}}_{T,D}(\mathrm{d}\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, k_{1:T})$ while Steps 2 and 3 jointly sample from $\tilde{\boldsymbol{\pi}}_{T,D}(\mathrm{d}\mathbf{x}_{1:T} \times \mathrm{d}k_{1:T}|\mathbf{z}_{1:T})$. For completeness, we give a more concise proof in Appendix C.2.

We stress that Proposition 3.1 does not require the high-dimensional regime from Assumption **A1**. That is, Algorithm 2 induces a valid (i.e. $\boldsymbol{\pi}_{T,D}$-invariant) Markov kernel even if the model does not factorise into D IID components.

3.2. *Stability in high dimensions.* We now show that the RW-EHMM algorithm is stable in high dimensions. For the analysis, we assume the regime from Assumption **A1**.

3.2.1. *Non-degenerate limiting law of the particle indices.* In the following, we show that as as $D \to \infty$, the law of the particle indices of the new reference path, $K_{1:T}$, under the RW-EHMM algorithm converges to a limit which is non-degenerate in the sense that the acceptance probabilities are strictly positive at each time step. Using the convention that ∂_t^i denotes the ith derivative w.r.t. x_t and with $\partial_t \coloneqq \partial_t^1$, as well as with $\pi_T(\varphi) \coloneqq \int_{\mathbb{R}^T} \varphi(x_{1:T})\pi_T(x_{1:T})\,\mathrm{d}x_{1:T}$, for any π_T-integrable function $\varphi : \mathbb{R}^T \to \mathbb{R}$, we make the following moment assumption.

B1 The density π_T is twice continuously differentiable and for any $s, t \in [T]$,
- $\partial_s\partial_t \log \pi_T$ is Lipschitz-continuous and bounded,
- $\pi_T(|\partial_t \log \pi_T|^4) < \infty$. ◁

Hereafter, we assume **B1**. The limiting law of $K_{1:T}$ (proved below) is then given by

$$\widetilde{\mathbb{P}}^N_T(\mathrm{d}v_{1:T} \times \mathrm{d}k_{1:T}) \coloneqq \left[\prod_{t=1}^{T} \mathrm{N}(\mathrm{d}v_t; \tilde{\mu}_{t|T}, \widetilde{\Sigma}_{t|T})\right] \prod_{t=1}^{T} \Psi^{k_t}(\{v_t^m\}_{m=1}^N).$$

Expectations w.r.t. $\widetilde{\mathbb{P}}^N_T$ are denoted by $\widetilde{\mathbb{E}}^N_T$. This law is defined through N-dimensional Gaussian random vectors $V_t \coloneqq V_t^{1:N}$ which are such that V_s and V_t are independent whenever $s \neq t$ and with mean vector and covariance matrix

$$\mathbb{E}[V_t] = -\tfrac{1}{2}\ell_t \mathcal{I}_{t|T}\mathbf{1}_N =: \tilde{\mu}_{t|T}, \quad \text{and} \quad \mathrm{var}[V_t] = \ell_t \mathcal{I}_{t|T}\Sigma =: \widetilde{\Sigma}_{t|T},$$

where by Lemma D.7 in Appendix D.6,

$$\mathcal{I}_{t|T} \coloneqq \pi_T([\partial_t \log \pi_T]^2) = -\pi_T(\partial_t^2 \log \pi_T).$$

3.2.2. *Convergence to the non-degenerate limit.* The following proposition (whose proof is a simpler version of the proof of Proposition 4.5 in Section 4 and is therefore omitted) shows that in high dimensions, the law of the indices $K_{1:T}$ under the RW-EHMM update specified in Algorithm 2 converges to a limit that is non-trivial in the sense that the events $\{K_t \neq 0\}$ have positive probability. Again, $\|\cdot\|$ is the total variation distance.

PROPOSITION 3.2 (convergence of the law of the particle indices). *Let $T, N \in \mathbb{N}$, assume **A1** as well as **B1**, and write*

$$\tilde{d}^N_{T,D,\mathbf{x}_{1:T}} := \left\| \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_{1:T} \in \cdot\}] - \widetilde{\mathbb{E}}^N_T[\mathbb{I}\{K_{1:T} \in \cdot\}] \right\|.$$

Then there exists a family $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ with $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$ and

$$\lim_{D\to\infty} \sup_{\mathbf{x}_{1:T}\in\mathbf{F}_{T,D}} \tilde{d}^N_{T,D,\mathbf{x}_{1:T}} = 0. \qquad \triangleleft$$

The following proposition (proved in Appendix C.3) shows that the *acceptance rate* at any time t associated with Algorithm 2,

$$\tilde{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t) := \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_t \neq 0\}],$$

converges to a strictly positive limit

$$\tilde{\alpha}^N_T(t) := \widetilde{\mathbb{E}}^N_T[\mathbb{I}\{K_t \neq 0\}].$$

Note that the acceptance rates and their limits depend on $\ell_{1:T}$ even though we do not make this explicit in our notation.

PROPOSITION 3.3 (dimensional stability of the acceptance rates). *Assume **A1** and **B1**. Then for $T, N \in \mathbb{N}$, $t \in [T]$ and $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ as in Proposition 3.2,*

$$\lim_{D\to\infty} \sup_{\mathbf{x}_{1:T}\in\mathbf{F}_{T,D}} |\tilde{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t) - \tilde{\alpha}^N_T(t)| = 0,$$

where

$$\tilde{\alpha}^N_T(t) \geq \left(1 + \frac{\exp(\ell_t \mathcal{I}_{t|T})}{N}\right)^{-1} > 0. \qquad \triangleleft$$

Of course, stabilising the acceptance rates in high dimensions is not sufficient for avoiding a breakdown. A widely used criterion for assessing the performance of MCMC algorithms is the *expected squared jumping distance (ESJD)* (Sherlock and Roberts, 2009), which (for the time-t component in Algorithm 2) is given by

$$\widetilde{\mathrm{ESJD}}^N_{T,D}(t) := \mathbb{E}[\|\mathbf{X}_t[l+1] - \mathbf{X}_t[l]\|_2^2],$$

where $\|\cdot\|_2$ denotes the Euclidean norm and where $\mathbf{X}_{1:T}[l]$ is the lth state of the Markov chain with transition kernel $\widetilde{\mathbf{P}}^N_{T,D}$ at stationarity. The following proposition (proved in Appendix C.3) shows that the ESJD is also stable in high dimensions.

PROPOSITION 3.4 (dimensional stability of the ESJD). *Assume **A1** and **B1** and let $T, N \in \mathbb{N}$. Then, for any $t \in [T]$,*

$$\lim_{D\to\infty} \left| \widetilde{\mathrm{ESJD}}^N_{T,D}(t) - \ell_t \tilde{\alpha}^N_T(t) \right| = 0. \qquad \triangleleft$$

3.2.3. *Stability as $T \to \infty$.* An immediate consequence of Proposition 3.3 is the following corollary which shows that the limiting acceptance rates $\tilde{\alpha}^N_T(t)$ are guaranteed to be bounded away from zero as $T \to \infty$.

COROLLARY 3.5 (time-horizon stability of the acceptance rates). *Assume **A1**, **B1** and that $\mathcal{I}(\ell) := \sup_{T\in\mathbb{N}} \sup_{t\in[T]} \ell_t \mathcal{I}_{t|T} < \infty$. Then, for any $N \in \mathbb{N}$,*

$$\inf_{T\in\mathbb{N}} \inf_{t\in[T]} \tilde{\alpha}^N_T(t) \geq \left(1 + \frac{\exp(\mathcal{I}(\ell))}{N}\right)^{-1} > 0. \qquad \triangleleft$$

REMARK 3.6. Proposition 3.4 makes it clear that the ESJD is also stable under the additional assumption that $\inf_{t\geq 1} \ell_t > 0$. $\triangleleft$

4. Proposed dimensionally stable methodology: the i-RW-CSMC algorithm.

4.1. *Description of the algorithm.*

4.1.1. *Basic algorithm.* In this section, we introduce our novel iterated *iterated random-walk conditional sequential Monte Carlo (i-RW-CSMC)* algorithm which induces an alternative $\boldsymbol{\pi}_{T,D}$-invariant Markov kernel. We also prove that the algorithm overcomes the curse of dimension suffered by the existing i-CSMC approach. The proposed algorithm scatters particles locally around the reference path in the same way as the RW-EHMM method introduced in the previous section. However, recall that this algorithm required $\mathrm{O}(N^2T)$ operations per iteration for fixed dimensions D. In contrast, the algorithm proposed in this section uses resampling steps to ensure that a single update can be implemented in $\mathrm{O}(NT)$ operations as in the standard i-CSMC approach.

For any $t \in [T]$ and any $\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}$, we define

$$\bar{\mathbf{w}}_t(\mathbf{x}_{t-1:t}) \coloneqq \log \mathbf{m}_t(\mathbf{x}_{t-1}, \mathbf{x}_t) + \log \mathbf{G}_t(\mathbf{x}_t),$$

with the convention that any quantity with time index $t < 1$ is to be ignored. The lth update of the novel i-RW-CSMC scheme is then outlined in Algorithm 3 where we use the convention that any action described for the nth particle index is to be performed conditionally independently for all $n \in [N]$ and any action described for the dth 'spatial' component is to be performed conditionally independently for all $d \in [D]$.

ALGORITHM 3 (i-RW-CSMC). Given $\mathbf{x}_{1:T} \coloneqq \mathbf{x}_{1:T}[l] \in \mathbf{E}_{T,D}$.

1. For $t \in [T]$:
   a) if $t > 1$,
      i. set $A_{t-1}^0 = a_{t-1}^0 \coloneqq 0$,
      ii. sample $A_{t-1}^n = a_{t-1}^n = l \in [N]_0$ with probability

$$\Psi^l(\{\bar{\mathbf{w}}_{t-1}(\mathbf{z}_{t-2}^{a_{t-2}^m}, \mathbf{z}_{t-1}^m) - \bar{\mathbf{w}}_{t-1}(\mathbf{z}_{t-2}^0, \mathbf{z}_{t-1}^0)\}_{m=1}^N) = \frac{\mathbf{m}_{t-1}(\mathbf{z}_{t-2}^{a_{t-2}^l}, \mathbf{z}_{t-1}^l)\mathbf{G}_{t-1}(\mathbf{z}_{t-1}^l)}{\sum_{m=0}^N \mathbf{m}_{t-1}(\mathbf{z}_{t-2}^{a_{t-2}^m}, \mathbf{z}_{t-1}^m)\mathbf{G}_{t-1}(\mathbf{z}_{t-1}^m)},$$

   b) set $\mathbf{Z}_t^0 = \mathbf{z}_t^0 \coloneqq \mathbf{x}_t$ and sample $\mathbf{Z}_t^{1:N} = \mathbf{z}_t^{1:N}$ as follows:
      i. sample $U_{t,d}^{1:N} \sim \mathrm{N}(0, \Sigma)$,
      ii. set $z_{t,d}^n \coloneqq z_{t,d}^0 + \sqrt{\ell_t / D} U_{t,d}^n$,
      iii. set $\mathbf{z}_t^n \coloneqq z_{t,1:D}^n$.
2. Sample $K_T = k_T \in [N]_0$ with probability

$$\Psi^{k_T}(\{\bar{\mathbf{w}}_T(\mathbf{z}_{T-1}^{a_{T-1}^m}, \mathbf{z}_T^m) - \bar{\mathbf{w}}_T(\mathbf{z}_{T-1}^0, \mathbf{z}_T^0)\}_{m=1}^N) = \frac{\mathbf{m}_T(\mathbf{z}_{T-1}^{a_{T-1}^{k_T}}, \mathbf{z}_T^{k_T})\mathbf{G}_T(\mathbf{z}_T^{k_T})}{\sum_{m=0}^N \mathbf{m}_T(\mathbf{z}_{T-1}^{a_{T-1}^m}, \mathbf{z}_T^m)\mathbf{G}_T(\mathbf{z}_T^m)}.$$

3. Set $K_t = k_t \coloneqq a_t^{k_{t+1}}$, for $t = T-1, \ldots, 1$.
4. Set $\mathbf{X}'_{1:T} \coloneqq \mathbf{x}_{1:T} \coloneqq (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T})$.
5. Return $\mathbf{x}_{1:T}[l+1] \coloneqq \mathbf{x}'_{1:T}$.

Step 1 of Algorithm 3 which a) performs (conditional) multinomial resampling by drawing the ancestor indices A_t^n; and b) generates the particles $\mathbf{Z}_t^n$ by scattering them around the reference particle $\mathbf{z}_t^0 = \mathbf{x}_t$ in the same way as Algorithm 2, will be referred to as the *random-walk conditional sequential Monte Carlo (RW-CSMC)* algorithm.

Step 2 samples a final-time particle index $K_T = k_T$ with probability proportional to the k_Tth particle weight at time T and Step 3 traces back the associated ancestral lineage.

The following proposition (proved in Appendix D.2) shows that the i-RW-CSMC algorithm can be viewed as a 'perturbed' version of the RW-EHMM algorithm. To our knowledge, this insight is novel.

PROPOSITION 4.1. *The combination of Steps 1a, 2 and 3 of Algorithm 3 induces a $\xi_T(\mathbf{z}_{1:T}, \cdot)$-invariant Markov kernel.* ◁

To provide additional intuition, Appendix A shows that the particles $\mathbf{Z}_{1:T}$ "discretise" the model into an $(N+1)$-state HMM and that the Markov kernel from Proposition 4.1 can be viewed running a slightly non-standard CSMC algorithm to target the joint posterior distribution of the states in this HMM.

We continue the running example from Section 2 which shows that the algorithms analysed in this work reduce to versions of well known classical MCMC kernels if $N = T = 1$.

EXAMPLE (classical MCMC kernels, continued). Algorithm 3 propoes $\mathbf{Z}_1^1 = \mathbf{z}_1^1 \sim \mathrm{N}(\mathbf{z}_1^0, \frac{\ell_1}{D}\mathrm{I}_D)$, where $\mathbf{z}_1^0 = \mathbf{x}_1[l]$, which is then accepted as the new state of the Markov chain with probability

$$\Psi^1(\bar{\mathbf{w}}_1(\mathbf{z}_1^1) - \bar{\mathbf{w}}_1(\mathbf{z}_1^0)) = \frac{\mathbf{m}_1(\mathbf{z}_1^1)\mathbf{G}_1(\mathbf{z}_1^1)}{\mathbf{m}_1(\mathbf{z}_1^0)\mathbf{G}_1(\mathbf{z}_1^0) + \mathbf{m}_1(\mathbf{z}_1^1)\mathbf{G}_1(\mathbf{z}_1^1)}.$$

This can be recognised as Barker's kernel (Barker, 1965) with a Gaussian random-walk proposal. Note that the symmetry property from (4) ensures that the proposal density cancels out in the acceptance probability. ◁

EXAMPLE (multi-proposal MCMC kernels, continued). For $N > 1$ and $T = 1$, Algorithm 3 is again a special case of a class of MCMC algorithms with multiple proposals (Tjelmeland, 2004). Related algorithms were analysed in Bédard, Douc and Moulines (2012); Bédard and Mireuta (2013) who also proved scaling limits in high dimensions. ◁

4.1.2. *Extensions.* The extensions discussed for the standard CSMC algorithm in Section 2.2.2 can be used for the i-RW-CSMC algorithms with only minor modifications.

Forced move. To use the forced-move approach, we simply replace the Boltzmann selection function by the Rosenbluth–Teller selection function in Step 2 of Algorithm 3.

EXAMPLE (classical MCMC kernels, continued). With the forced-move extension, Algorithm 3 proposes $\mathbf{Z}_1^1 = \mathbf{z}_1^1 \sim \mathrm{N}(\mathbf{z}_1^0, \frac{\ell_1}{D}\mathrm{I}_D)$, where $\mathbf{z}_1^0 = \mathbf{x}_1[l]$, which is then accepted as the new state of the Markov chain with probability

$$\Phi^1(\bar{\mathbf{w}}_1(\mathbf{z}_1^1) - \bar{\mathbf{w}}_1(\mathbf{z}_1^0)) = 1 \wedge \frac{\mathbf{m}_1(\mathbf{z}_1^1)\mathbf{G}_1(\mathbf{z}_1^1)}{\mathbf{m}_1(\mathbf{z}_1^0)\mathbf{G}_1(\mathbf{z}_1^0)}.$$

This can be recognised as a MH kernel (Metropolis et al., 1953; Hastings, 1970) with a Gaussian random-walk proposal. Again, the symmetry property from (4) ensures that the proposal density cancels out in the acceptance ratio. ◁

Backward sampling. To employ backward sampling, we sample each particle index $K_t = k_t \in [N]_0$ in Step 3 of Algorithm 3 with probability

$$\begin{aligned}\Psi^{k_t}(\{\bar{\mathbf{v}}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}}) - \bar{\mathbf{v}}_t(\mathbf{z}_{t-1}^0, \mathbf{z}_t^0, \mathbf{z}_{t+1}^{k_{t+1}})\}_{m=1}^N) \\ = \frac{\mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^{k_t}}, \mathbf{z}_t^{k_t})\mathbf{G}_t(\mathbf{z}_t^{k_t})\mathbf{m}_{t+1}(\mathbf{z}_t^{k_t}, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^N \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m)\mathbf{G}_t(\mathbf{z}_t^m)\mathbf{m}_{t+1}(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}})},\end{aligned} \tag{7}$$

where

$$\bar{\mathbf{v}}_t(\mathbf{x}_{t-1:t+1}) := \bar{\mathbf{w}}_t(\mathbf{x}_{t-1:t}) + \log \mathbf{m}_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1}).$$

The following proposition (proved in Appendix D.2) shows that when backward sampling is used, the i-RW-CSMC algorithm can again be viewed as a 'perturbed' version of the RW-EHMM algorithm. To our knowledge, this insight is novel.

PROPOSITION 4.2. *Proposition 4.1 remains valid if backward sampling is used, i.e. the combination of Steps 1a, 2 and 3 of Algorithm 3 again induces a $\xi_T(\mathbf{z}_{1:T}, \cdot)$-invariant Markov kernel.* ◁

Again, Appendix A shows that the Markov kernel from Proposition 4.2 be viewed running a slightly non-standard CSMC algorithm with backward sampling to target the $(N+1)$-state HMM mentioned above.

4.1.3. *Induced $\boldsymbol{\pi}_{T,D}$-invariant Markov kernel.* Given $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, let

$$\bar{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T})$$

be the law of all the random variables $(\mathbf{Z}_{1:T}, A_{1:T-1}, K_{1:T}, \mathbf{X}'_{1:T})$ generated in Steps 1–4 of Algorithm 3 (with or without the forced-move extension and with or without backward sampling). Appendix D.1 gives a more formal definition of this law.

Let $\bar{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}$ denote expectation w.r.t. $\bar{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}}$. Algorithm 3 induces a Markov kernel

$$\bar{\mathbf{P}}^N_{T,D}(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) := \bar{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}],$$

for $(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}) \in \mathbf{E}_{T,D} \times \boldsymbol{\mathcal{E}}_{T,D}$. The following proposition shows that this Markov kernel leaves $\boldsymbol{\pi}_{T,D}$ invariant. It can be proved by interpreting the i-RW-CSMC kernel as a special case of the generic iterated CSMC approach described in Finke, Doucet and Johansen (2016). For completeness, we provide a simple proof in Appendix D.2.

PROPOSITION 4.3. *For any $N, T, D \in \mathbb{N}$, $\boldsymbol{\pi}_{T,D}\bar{\mathbf{P}}^N_{T,D} = \boldsymbol{\pi}_{T,D}$.* ◁

We stress that this proposition does not require the high-dimensional regime from Assumption **A1**. That is, Algorithm 3 induces a valid (i.e. $\boldsymbol{\pi}_{T,D}$-invariant) Markov kernel even if the model does not factorise into D IID components.

REMARK 4.4. As explained in Andrieu, Doucet and Holenstein (2010), the (standard) CSMC algorithm is closely linked to the justification of a corresponding 'unconditional' SMC algorithm. However, for the RW-CSMC algorithm, no such 'unconditional' SMC counterpart exists. We expand on this in Appendix D.3. ◁

4.2. *Stability in high dimensions.* In this section, prove that the i-RW-CSMC algorithm is dimensionally stable. Throughout this section, we assume that the model follows the high-dimensional regime from Assumption **A1**. Such assumptions are common in the literature on optimal scaling (Roberts, Gelman and Gilks, 1997). However, we stress that the algorithm is agnostic to this structure, i.e. it does not exploit the fact that the target distribution factorises. We thus expect such results to hold more generally.

4.2.1. *Non-degenerate limiting law of the genealogies.* In the following, we show that as as $D \to \infty$, the law of the genealogies (and the indices of the new reference path) induced by the i-RW-CSMC algorithm converges to a limit that is non-degenerate in the sense that the particle lineages do not necessarily coalesce with the reference path (and, likewise, the indices of the new reference path do not necessarily coincide with those of the old reference path). Define

$$\bar{w}_t = \log G_t + \log m_t,$$

with the convention $\bar{w}_{T+1} \equiv 0$. We again use the convention that ∂_t^i denotes the ith derivative w.r.t. x_t and with $\partial_t \coloneqq \partial_t^1$, and we write $\pi_T(\varphi) \coloneqq \int_{\mathbb{R}^T} \varphi(x_{1:T})\pi_T(x_{1:T})\,\mathrm{d}x_{1:T}$, for any π_T-integrable function $\varphi : \mathbb{R}^T \to \mathbb{R}$. With these conventions, we make the following moment assumptions which are similar to the assumptions in Bédard, Douc and Moulines (2012), and which are assumed to hold for any $t \in [T]$.

C1 $\bar{w}_t$ is twice continuously differentiable and
- $\partial_t^2 \bar{w}_t$, $\partial_t^2 \bar{w}_{t+1}$, $\partial_t \partial_{t+1} \bar{w}_{t+1}$ are Lipschitz-continuous and bounded,
- $\pi_T(|\partial_t \bar{w}_t|^4), \pi_T(|\partial_t \bar{w}_{t+1}|^4) < \infty$. ◁

Before stating the result, we define a law $\bar{\mathbb{P}}_T^N(\mathrm{d}v_{1:T} \times \mathrm{d}w_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T})$. This is a joint distribution of random variables $(V_{1:T}, W_{1:T}, A_{1:T-1}, K_{1:T})$, where $A_{1:T-1}$ and $K_{1:T}$ are collections of ancestor and particle indices as in the fixed-dimensional case and where $V_t \coloneqq V_t^{1:N}$ and $W_t \coloneqq W_t^{1:N}$ are each N-dimensional Gaussian vectors which are such that (V_s, W_s) and (V_t, W_t) are independent whenever $s \neq t$ and such that

$$\mathbb{E}\begin{bmatrix} V_t \\ W_t \end{bmatrix} = \tfrac{1}{2}\ell_t \begin{bmatrix} \pi_T(\partial_t^2 \bar{w}_t)\mathbf{1}_N \\ \pi_T(\partial_t^2 \bar{w}_{t+1})\mathbf{1}_N \end{bmatrix} =: \bar{\mu}_{t|T},$$

and, recalling that $\Sigma = \frac{1}{2}(\mathrm{I}_N + \mathbf{1}_N \mathbf{1}_N^{\mathrm{T}})$,

$$\mathrm{var}\begin{bmatrix} V_t \\ W_t \end{bmatrix} = \ell_t \begin{bmatrix} \pi_T([\partial_t \bar{w}_t]^2)\Sigma & \pi_T([\partial_t \bar{w}_t][\partial_t \bar{w}_{t+1}])\Sigma \\ \pi_T([\partial_t \bar{w}_t][\partial_t \bar{w}_{t+1}])\Sigma & \pi_T([\partial_t \bar{w}_{t+1}]^2)\Sigma \end{bmatrix} =: \bar{\Sigma}_{t|T}.$$

We will also use the convention that $V_t^0 = W_t^0 \equiv 0$ for any $t \in [T]$ and $W_0^n \equiv 0$ for any $n \in [N]$.

Note that under Assumption **C1**, by Lemma D.7 in Appendix D.6,

$$\begin{aligned} \mathcal{I}_{t|T} &\coloneqq \pi_T([\partial_t \log \pi_t]^2) = -\pi_T(\partial_t^2 \log \pi_T) \\ &= \mathrm{var}[V_t^n + W_t^n]/\ell_t = -2\,\mathbb{E}[V_t^n + W_t^n]/\ell_t < \infty. \end{aligned}$$

We are now ready to state the limiting law of the genealogies. Throughout, we assume **C1**. Then we can define the law $\bar{\mathbb{P}}_T^N(\mathrm{d}v_{1:T} \times \mathrm{d}w_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T})$ by the following sampling procedure (a formal definition is given in Appendix D.4).

1. For $t \in [T]$, sample $(V_t, W_t) = (v_t, w_t) \sim \mathrm{N}(\bar{\mu}_{t|T}, \bar{\Sigma}_{t|T})$.
2. For $t \in [T-1]$,
   a) set $A_t^0 = a_t^0 \coloneqq 0$,
   b) sample $A_t^n = a_t^n = l \in [N]_0$ with probability $\Psi^l(\{v_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N)$, independently for $n \in [N]$.
3. Sample $K_T = k_T \in [N]_0$ with probability $\Psi^{k_T}(\{v_T^m + w_{T-1}^{a_{T-1}^m}\}_{m=1}^N)$.
4. For $t = T-1, \ldots, 1$, set $K_t = k_t \coloneqq a_t^{k_{t+1}}$.

As usual, if we use the forced-move extension, we must replace Ψ^n by Φ^n in Step 3. Likewise, if we use backward sampling, we must instead sample $K_t = k_t \in [N]_0$ with probability $\Psi^{k_t}(\{v_t^m + w_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N)$ in Step 4. Hereafter, $\bar{\mathbb{E}}_T^N$ denotes expectation w.r.t. $\bar{\mathbb{P}}_T^N$.

4.2.2. *Convergence to the non-degenerate limit.* Proposition 4.5, proved in Appendix D.5, shows that in high dimensions, the law of the genealogies and the indices of the new reference path under the i-RW-CSMC update specified in Algorithm 3 converges to the limiting law specified above. Here again, $\|\cdot\|$ denotes the total variation distance.

PROPOSITION 4.5 (convergence of the law of the genealogies). *Let $T, N \in \mathbb{N}$, assume **A1** as well as **C1**, and write*

$$\bar{d}_{T,D,\mathbf{x}_{1:T}}^N \coloneqq \big\|\bar{\mathbb{E}}_{T,D,\mathbf{x}_{1:T}}^N[\mathbb{I}\{(A_{1:T-1}, K_{1:T}) \in \cdot\}] - \bar{\mathbb{E}}_T^N[\mathbb{I}\{(A_{1:T-1}, K_{1:T}) \in \cdot\}]\big\|.$$

Then there exists a family $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ *with* $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$ *and*

$$\lim_{D\to\infty} \sup_{\mathbf{x}_{1:T}\in\mathbf{F}_{T,D}} \bar{d}^N_{T,D,\mathbf{x}_{1:T}} = 0. \qquad \triangleleft$$

The following corollary shows that the *acceptance rate* at any time t associated with Algorithm 3,

$$\bar{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t) := \bar{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_t \neq 0\}],$$

converges to a strictly positive limit

$$\bar{\alpha}^N_T(t) := \bar{\mathbb{E}}^N_T[\mathbb{I}\{K_t \neq 0\}].$$

Note that the acceptance rates and their limits depend on $\ell_{1:T}$ even though we do not make this explicit in our notation.

COROLLARY 4.6 (dimensional stability of the acceptance rates). *Assume* ***A1*** *and* ***C1****, $T, N \in \mathbb{N}$, and let $t \in [T]$. Then $\bar{\alpha}^N_T(t) > 0$. Furthermore, with $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ as in Proposition 4.5:*

$$\lim_{D\to\infty} \sup_{\mathbf{x}_{1:T}\in\mathbf{F}_{T,D}} |\bar{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t) - \bar{\alpha}^N_T(t)| = 0. \qquad \triangleleft$$

PROOF. The convergence follows immediately from Proposition 4.5. The strict positivity of the limit is due to the finite-moments assumption **C1**. □

EXAMPLE (classical MCMC kernels, continued). As mentioned above, Algorithm 3 reduces to a classical MCMC kernel with a suitably scaled Gaussian random-walk proposal if $T = N = 1$. In this case, the asymptotic acceptance rates for Barker's kernel and for the MH kernel derived in Roberts, Gelman and Gilks (1997); Bédard, Douc and Moulines (2012); Agrawal et al. (2021):

$$\bar{\alpha}^1_1(1) = \begin{cases} \mathbb{E}[\Psi^1(V^1_1)] = \mathbb{E}\big[\frac{\exp\{V^1_1\}}{1+\exp\{V^1_1\}}\big], & \text{without forced-move,} \\ \mathbb{E}[\Phi^1(V^1_1)] = \mathbb{E}[1 \wedge \exp\{V^1_1\}] = 2\Phi\big(-\frac{\sqrt{\ell_1 \mathcal{I}_{1|1}}}{2}\big), & \text{with forced-move,} \end{cases}$$

where Φ is standard-normal cumulative distribution function. $\triangleleft$

Of course, stabilising the acceptance rates in high dimensions is not sufficient for avoiding a breakdown. A widely used criterion for assessing the performance of MCMC algorithms is the *expected squared jumping distance (ESJD)* (Sherlock and Roberts, 2009), which (for the time-t component in Algorithm 3) is given by

$$\overline{\mathrm{ESJD}}^N_{T,D}(t) := \mathbb{E}[\|\mathbf{X}_t[l+1] - \mathbf{X}_t[l]\|_2^2],$$

where $\|\cdot\|_2$ denotes the Euclidean norm and where $\mathbf{X}_{1:T}[l]$ is the lth state of the Markov chain with transition kernel $\bar{\mathbf{P}}^N_{T,D}$ at stationarity. The following proposition (whose proof is the same as that of Proposition 3.4 in Appendix C.4 and is therefore omitted) shows that the ESJD is stable in high dimensions.

PROPOSITION 4.7 (dimensional stability of the ESJD). *Assume* ***A1*** *as well as* ***C1****, and let $T, N \in \mathbb{N}$. Then, for any $t \in [T]$,*

$$\lim_{D\to\infty} \big|\overline{\mathrm{ESJD}}^N_{T,D}(t) - \ell_t \bar{\alpha}^N_T(t)\big| = 0. \qquad \triangleleft$$

4.2.3. *Stability as $T \to \infty$.* In this section, we discuss scaling of the number of particles, $N+1$, in the time horizon, T, in high (spatial) dimensions. Throughout, we let $\ell := (\ell_t)_{t \geq 1}$ be a sequence of positive scaling factors.

Under stability assumptions on the Feynman–Kac model and for some fixed spatial dimension D, it is well known that the acceptance rates of the i-CSMC algorithm, $\alpha^N_{T,D,\mathbf{x}_{1:T}}(t)$, can be bounded away from zero as $T \to \infty$ by scaling the number of particles appropriately. To be more specific: Without backward sampling, $\alpha^N_{T,D,\mathbf{x}_{1:T}}(t)$ can be controlled by growing N linearly in T (Andrieu, Lee and Vihola, 2018; Lindsten, Douc and Moulines, 2015; Del Moral, Kohn and Patras, 2016). With backward sampling, $\alpha^N_{T,D,\mathbf{x}_{1:T}}(t)$ can be controlled without scaling N with T (Lee, Singh and Vihola, 2020). Proposition 4.8 (proved in Appendix D.6) verifies that – under the following 'factorisation-over-time' assumption – the i-RW-CSMC algorithm admits the same scaling of N with T as the i-CSMC algorithm.

A4 For any $t \in [T]$ and any $(x_{t-1}, x'_{t-1}) \in \mathbb{R}^2$, $M_t(x_{t-1}, \cdot) = M_t(x'_{t-1}, \cdot)$. ◁

PROPOSITION 4.8 (time-horizon stability of the acceptance rates). *Assume **A1**, **A4** as well as **C1** and that $\mathcal{I}(\ell) := \sup_{T \in \mathbb{N}} \sup_{t \in [T]} \ell_t \mathcal{I}_{t|T} < \infty$.*

1. *Without backward sampling, if there exists $C > 0$ such that $N \geq CT$:*

$$\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \bar{\alpha}^N_T(t) \geq \exp\left(-\frac{\exp(\mathcal{I}(\ell))}{C}\right) > 0.$$

2. *With backward sampling, for any $N \in \mathbb{N}$:*

$$\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \bar{\alpha}^N_T(t) \geq \left(1 + \frac{\exp(\mathcal{I}(\ell))}{N}\right)^{-1} > 0.$$ ◁

REMARK 4.9. Proposition 4.7 shows that $\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \overline{\mathrm{ESJD}}^N_{T,D}(t) > 0$ (i.e. the ESJD is also stable) under the additional assumption $\inf_{t \geq 1} \ell_t > 0$. ◁

Note that the lower bound for the case with backward sampling in Proposition 4.8 is the same as the one obtained for the RW-EHMM algorithm in Corollary 3.5. This is not a coincidence: under Assumption **A4**, Algorithm 3 with backward sampling induces the same Markov kernel as Algorithm 2, $\bar{\mathbf{P}}^N_{T,D} = \widetilde{\mathbf{P}}^N_{T,D}$. However, recall that in Corollary 3.5, we were able to prove stability of the acceptance rates in T without relying on Assumption **A4**. This, along with Propositions 4.1 and 4.2 (which show that the i-RW-CSMC algorithm can be viewed as a 'perturbed' version of the RW-EHMM algorithm) motivates the following conjecture.

CONJECTURE 4.10. Assumption **A4** is not necessary to guarantee stability of the acceptance rates in T with the scaling of $N = N(T)$ from Proposition 4.8. ◁

5. Numerical illustration. In this section, we illustrate the results on a simple state-space model specified as follows (additional simulations in a more realistic problem – a multivariate stochastic volatility model – are provided in Appendix E.3.2). Let φ denote a Lebesgue density of the standard normal distribution. Let $\mathbf{y}_t = (y_{t,d})_{d \in [D]} \in \mathbb{R}^D$ be some D-dimensional vector of observations collected at time $t \in [T]$. Then for any $d \in [D]$,

$$G_t(x_{t,d}) := \varphi(y_{t,d} - x_{t,d}),$$
$$m_t(x_{t-1,d}, x_{t,d}) := \varphi(x_{t,d} - x_{t-1,d}).$$

The results shown below are based on 100 independent runs of each algorithm for each value of D; each run uses $L = 25\,000$ iterations initialised from stationarity, and each uses

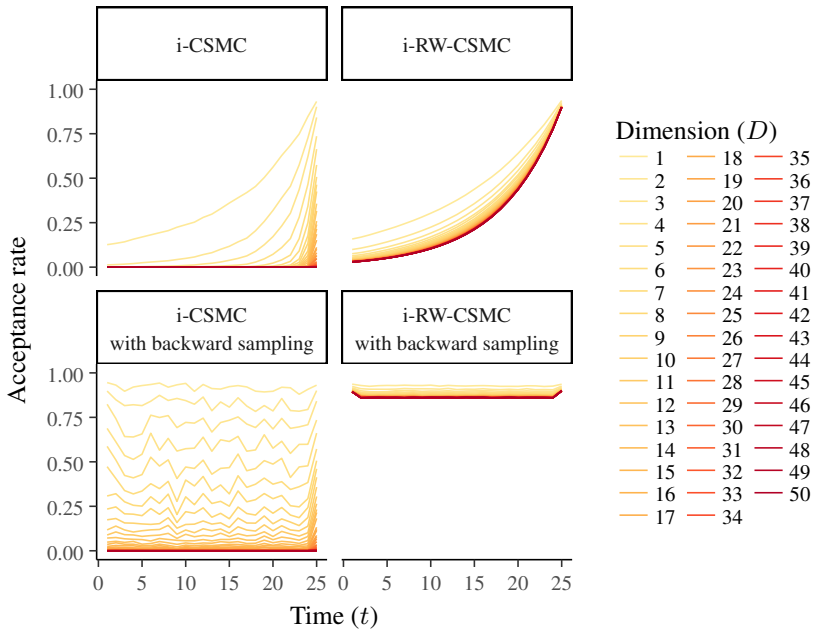


FIG. 2. The $\boldsymbol{\pi}_{T,D}$-averaged acceptance rates as a function of t.

a different observation sequence of length $T = 25$ sampled from the model. Throughout, we use $N + 1 = 32$ particles. In the i-RW-CSMC algorithm, $\ell_1 = \ldots = \ell_T = 1$.

Figure 2 displays the $\boldsymbol{\pi}_{T,D}$-averaged acceptance rates as a function of the time index t. More precisely, for $\mathbf{X}_{1:T} \sim \boldsymbol{\pi}_{T,D}$, it shows:

1. first column: $\mathbb{E}[\alpha^N_{T,D,\mathbf{X}_{1:T}}(t)]$;
2. second column: $\mathbb{E}[\bar{\alpha}^N_{T,D,\mathbf{X}_{1:T}}(t)]$.

The upper-left panel shows that for the i-CSMC algorithm, the acceptance rates vanish in high dimensions. In contrast, the upper-right panel shows that the acceptance rates converge to a non-trivial limit for the i-RW-CSMC algorithm. The first row also shows that, in both algorithms, the acceptance rates are an increasing function of the time index t due to the coalescence of the particle paths with the reference path. The second row shows that the backward-sampling extension removes this dependence on the time index and leads to acceptance rates which are stable over time. However, the lower-left panel illustrates that backward sampling does not save the i-CSMC algorithm in high dimensions – its acceptance rates still vanish. Additionally, in Appendix E, we illustrate that the effective sample size (ESS) (Kong, Liu and Wong, 1994) of the resampling and backward-sampling weights converges to a non-trivial limit > 1 under the i-RW-CSMC algorithm, whereas it collapses to 1 for the i-CSMC algorithm.

Figure 3 displays the expected squared jumping distance (ESJD) as a function of t. More specifically, it shows:

1. first column: $\mathrm{ESJD}^N_{T,D}(t) := \mathbb{E}[\|\mathbf{X}_t[l+1] - \mathbf{X}_t[l]\|_2^2]$;
2. second column: $\overline{\mathrm{ESJD}}^N_{T,D}(t) := \mathbb{E}[\|\bar{\mathbf{X}}_t[l+1] - \mathbf{X}_t[l]\|_2^2]$,

where $\mathbf{X}_{1:T}[l]$ is the lth state of a stationary Markov chain with transition kernels $\mathbf{P}^N_{T,D}$ and $\bar{\mathbf{X}}_{1:T}[l]$ is the lth state of a stationary Markov chain with transition kernels $\bar{\mathbf{P}}^N_{T,D}$. The first column illustrates that in high dimensions, the ESJD of the i-CSMC algorithm vanishes in

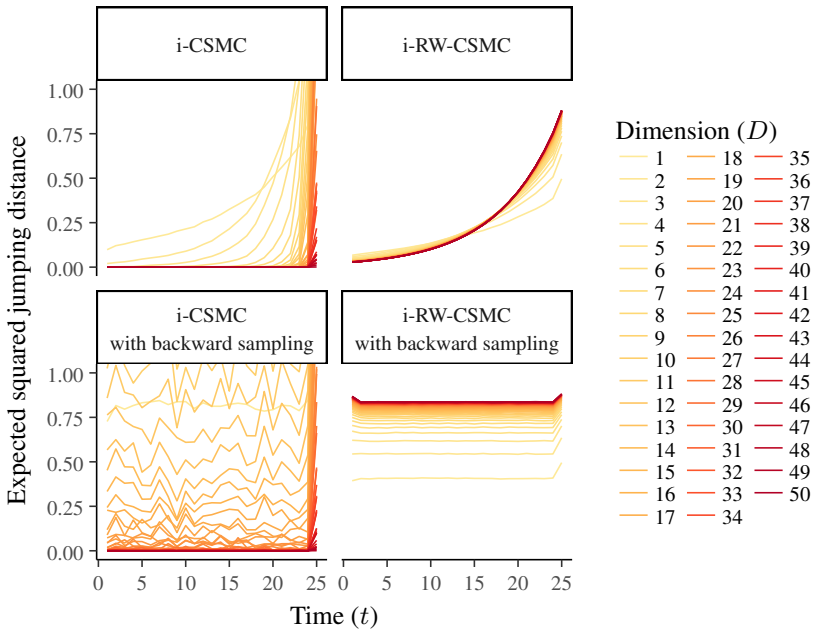


FIG. 3. Expected squared jumping distance as a function of t.

high dimensions. In contrast, the ESJD of the i-RW-CSMC algorithm converges to $\ell_t \bar{\alpha}_T^N(t) > 0$ (in accordance with Proposition 4.7).

Figure 4 displays the lag-D autocorrelation of the sample for the first 'spatial' component at each time t. More precisely, writing $\mathbf{X}_t[l] = X_{t,1:D}[l]$ and $\bar{\mathbf{X}}_t[l] = \bar{X}_{t,1:D}[l]$, it shows:

1. first column: $\mathrm{corr}(X_{t,1}[l+D], X_{t,1}[l])$;
2. second column: $\mathrm{corr}(\bar{X}_{t,1}[l+D], \bar{X}_{t,1}[l])$.

The fact that this leads to non-trivial limit in the second column illustrates that the i-RW-CSMC algorithm is stable in high dimensions as long as the number of iterations grows linearly with D. However, the first column shows that increasing the number of iterations in this manner does not save the i-CSMC algorithm from breaking down in high dimensions.

6. Practical implementation. In this section, we discuss how to implement the proposed algorithms in practice.

Initialisation. We suggest initialising the MCMC chain using a simple bootstrap particle filter, i.e. the "unconditional" counterpart of Algorithm 1, using a modest number of particles, say $N = 100$. Empirically, we have found this strategy to work well even in higher dimensions and it has the advantage that it requires no problem-specific tuning.

Choice of N. As we have shown, it is not needed to scale N with T or D. In addition, the acceptance rates are bounded above by 1 so that they can only increase sublinearly in N. Hence, we recommend to select N based on the available parallel computing architecture (and to a relatively small value on serial machines). Note that for small N, the RW-EHMM algorithm may even be a viable alternative.

Choice of ℓ_t. We propose a simple adaptation strategy to specify $\ell_t[g]$, the value of ℓ_t at the gth iteration of the algorithm. Let $\alpha_t[g]$ denote the average acceptance rate at time t up to the

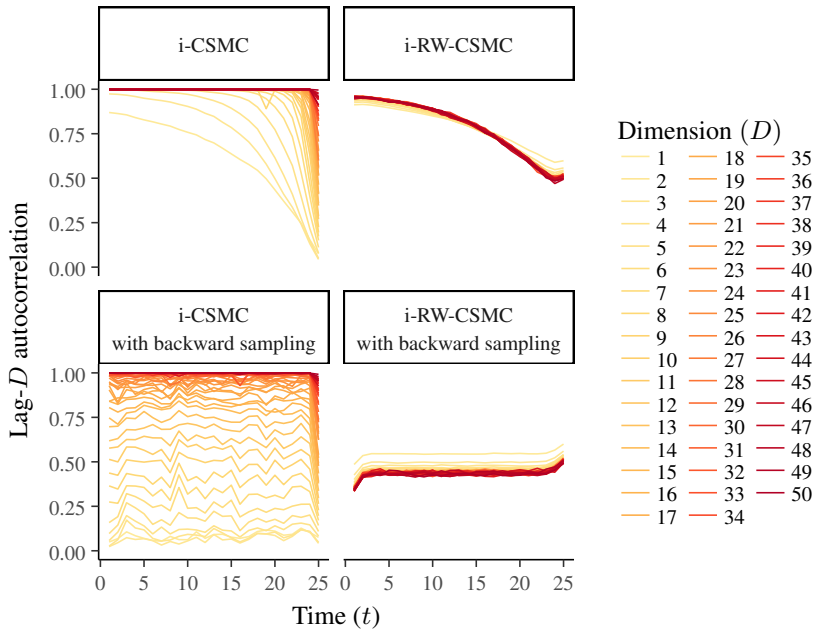


FIG. 4. Lag-D autocorrelations of the first marginal of the D-dimensional time-t state vector as a function of t.

gth iteration. Then we set

$$\ell_t[g] \coloneqq \begin{cases} 0.9 \cdot \ell_t[g-1], & \text{if } \alpha_t[g] < \max\{\alpha - 0.05, 0.05\}, \\ 1.1 \cdot \ell_t[g-1], & \text{if } \alpha_t[g] > \min\{\alpha + 0.05, 0.95\}, \\ \exp(\log(\ell_t[g-1]) + (\alpha_t[g] - \alpha)/g), & \text{otherwise.} \end{cases}$$

Here, $\alpha \in (0,1)$ is some target acceptance rate. Motivated by optimal-scaling results in a related multi-proposal MCMC setting (Bédard, Douc and Moulines, 2012), and by further empirical results shown in Appendix E.2, we take α to be an increasing function in N. Specifically, we use $\alpha \coloneqq 1 - (N+1)^{-\beta}$ for some $\beta \in (0,1)$, say $\beta = 1/3$. In Appendix E.3.2 we illustrate that this strategy is able to quickly improve poor initial choices of ℓ_t. It must be pointed out that such adaptation strategies break the guarantee that the algorithm leaves the desired target distribution invariant. Common practice in the literature on MCMC methods is therefore to stop the adaptation after some pre-specified burn-in phase.

7. Conclusion.

Comparison with classical MCMC algorithms. The iterated conditional sequential Monte Carlo (i-CSMC) algorithm (Andrieu, Doucet and Holenstein, 2010) is a powerful tool for inference about the joint smoothing distribution in state-space models and more generally. This is because the algorithm automatically exploits the decorrelation in the 'time' direction exhibited by the model. Let T denotes the time horizon and let D denote the 'spatial' dimension of each latent state.

For the moment assume that D is fixed and reasonably small.

- The i-CSMC algorithm has $\mathrm{O}(T)$ complexity[1] which is in contrast to a naïve independent Metropolis–Hastings (MH) algorithm which does not exploit the structure of the model and therefore suffers $\mathrm{O}(\mathrm{e}^T)$ complexity, i.e. a curse of dimension in T.
- The i-CSMC algorithm can be combined with backward sampling to reduce the complexity to $\mathrm{O}(1)$ (Lee, Singh and Vihola, 2020). This is similar to combining the independent MH updates with a 'blocking' (in the time direction) strategy. Indeed, Singh, Lindsten and Moulines (2017) reduced the complexity of the standard i-CSMC algorithm by using time-direction blocking instead of backward sampling.

Now, consider the case that the 'spatial' dimension D is large.

- Unfortunately, the i-CSMC algorithm (Andrieu, Doucet and Holenstein, 2010) (with or without backward sampling) then proposed propose 'global' moves in the 'space' direction and consequently suffers a curse of dimension in D (i.e. complexity $\mathrm{O}(\mathrm{e}^D)$) in the same way as an independent MH algorithm (with or without blocking in the time direction). Indeed, if $T = N = 1$ then the i-CSMC algorithm reduces to a standard independent MH algorithm for which this drawback is well known.
- It is also well known that proposing 'local' moves can overcome this curse of dimension (Roberts, Gelman and Gilks, 1997). That is, in the classical MCMC setting, we must use, e.g., suitably scaled random-walk rather than independence proposals. In this work, we have therefore proposed a novel iterated 'local' CSMC algorithm, termed iterated random-walk conditional sequential Monte Carlo (i-RW-CSMC) algorithm, which utilises Gaussian random-walk proposals whose variance is of order D^{-1}. The algorithm provably avoids the curse of dimension in D suffered by the original i-CSMC algorithm. To potentially remove the need for scaling the number of particles with T and thus achieve $\mathrm{O}(D)$ complexity, the i-RW-CSMC algorithm can be combined with backward sampling. This is again akin to blocking in the time direction in classical MCMC algorithms. In fact, Appendix E.2 illustrates that a multi-proposal Gaussian random-walk MH algorithm with blocking in the time direction can perform similar to the i-RW-CSMC algorithm with backward sampling. Our algorithm reduces to a standard Gaussian random-walk MH algorithm if $T = N = 1$.

Limitations. The i-RW-CSMC algorithm shares the well-known limitations of the random-walk MH algorithm. That is, it requires the state space to be continuous in order to allow for suitably scaled local proposals; and such 'local' proposals may not easily move between well-separated modes.

Extensions. Further reductions in the complexity from $\mathrm{O}(D)$ to $\mathrm{O}(1)$ may be feasible by employing blocking strategies in the 'space' direction (Rebeschini and van Handel, 2015; Finke and Singh, 2017). Indeed, Murphy and Godsill (2015) proposed a spatially-blocked i-CSMC algorithm. However, such strategies require a specific 'spatial' (de)correlation structure which can only be found in particular models.

The analysis of the i-RW-CSMC algorithm in high dimensions could be extended in a number of ways. First, we could easily consider models in which the potential functions G_t depend not only on x_t but also on x_{t-1} or allow scale factors ℓ_t differ across particles. Second, we could consider the case that ancestor sampling instead of backward sampling is used. Third, in the same way as this has been done in the literature on optimal scaling for classical MCMC algorithms, the assumptions on the high-dimensional regime could be relaxed, e.g. by allowing for some of the D components of the target distribution to be differently

[1] Recall that 'complexity' is measured as the number of full likelihood evaluations needed to control the approximation error of a fixed-dimensional marginal of the joint smoothing distribution.

scaled or by allowing for dependence between some of the D components (see, e.g., Sherlock and Roberts, 2009; Yang, Roberts and Rosenthal, 2020). Likewise, we could allow for non-Gaussian proposals (Neal and Roberts, 2011). Finally, we could investigate the optimal choice of the scaling factors $\ell_t^\star$ and the associated optimal acceptance rates as well as proving a suitable T-dimensional diffusion limit of a time-scaled T-dimensional spatial marginal of the Markov chain induced by the algorithm.

We stress that the i-RW-CSMC algorithm introduced in this work is by no means the only way of achieving 'local' CSMC updates. We have only focussed on this particular algorithm to keep the presentation simple. Alternative local CSMC algorithms are possible. For instance, the first such local algorithm proposed in the literature is the method from Shestopaloff and Neal (2018) which uses MCMC kernels for proposing local moves around the reference path. Their algorithm reduces to a delayed-acceptance MH algorithm (Christen and Fox, 2005) if $T = N = 1$. The algorithm showed promising performance in some high-dimensional settings in Finke, Doucet and Johansen (2016) who also provided a generic framework which admits this approach as well as the algorithms analysed in this work as special cases.

Acknowledgments. The first author would like to thank Arnaud Doucet for insightful discussions which led to this research.

Funding. The authors acknowledge support from the Singapore Ministry of Education Tier 2 (MOE2016-T2-2-135) and a Young Investigator Award Grant (NUSYIA FY16 P16; R-155-000-180-133).

REFERENCES

AGRAWAL, S., VATS, D., ŁATUSZYŃSKI, K. and ROBERTS, G. O. (2021). Optimal scaling of MCMC beyond Metropolis. *arXiv e-prints* arXiv:2104.02020. https://doi.org/10.48550/arXiv.2104.02020

ANDRIEU, C., DOUCET, A. and HOLENSTEIN, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72** 269–342. With discussion. https://doi.org/10.1111/j.1467-9868.2009.00736.x

ANDRIEU, C., LEE, A. and VIHOLA, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli* **24** 842–872. https://doi.org/10.3150/15-BEJ785

BARKER, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton–electron plasma. *Australian Journal of Physics* **18** 119–134. https://doi.org/10.1071/PH650119

BÉDARD, M., DOUC, R. and MOULINES, E. (2012). Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications* **122** 758–786. https://doi.org/10.1016/j.spa.2011.11.004

BÉDARD, M. and MIREUTA, M. (2013). On the empirical efficiency of local MCMC algorithms with pools of proposals. *Canadian Journal of Statistics* **41** 657–678. https://doi.org/10.1002/cjs.11196

BROWN, S., JENKINS, P. A., JOHANSEN, A. M. and KOSKELA, J. (2021). Simple conditions for convergence of sequential Monte Carlo genealogies with applications. *Electronic Journal of Probability* **26** 1–22. https://doi.org/10.1214/20-EJP56

CAPPÉ, O., MOULINES, E. and RYDÉN, T. (2005). *Inference in hidden Markov models*. *Springer Series in Statistics*. Springer.

CHOPIN, N. and SINGH, S. S. (2015). On particle Gibbs sampling. *Bernoulli* **21** 1855–1883. https://doi.org/10.3150/14-BEJ629

CHRISTEN, J. A. and FOX, C. (2005). Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics* **14** 795–810. https://doi.org/10.1198/106186005X76983

CRESSIE, N. and WIKLE, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

DEL MORAL, P., KOHN, R. and PATRAS, F. (2016). On particle Gibbs samplers. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* **52** 1687–1733. https://doi.org/10.1214/15-AIHP695

DELMAS, J.-F. and JOURDAIN, B. (2009). Does waste recycling really improve the multi-proposal Metropolis–Hastings algorithm? An analysis based on control variates. *Journal of Applied Probability* **46** 938–959. https://doi.org/10.1239/jap/1261670681

DHAENE, J., WANG, S., YOUNG, V. and GOOVAERTS, M. (2000). Comonotonicity and maximal stop-loss premiums. *Bulletin of the Swiss Association of Actuaries* **2** 99–113.

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* 64–69. IEEE. https://doi.org/10.1109/ISPA.2005.195385

DOUCET, A., GODSILL, S. J. and ANDRIEU, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* **10** 197–208. https://doi.org/10.1023/A:1008935410038

DVORETZKY, A. (1972). Asymptotic normality for sums of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. The Regents of the University of California.

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** 887–899. https://doi.org/10.1111/1467-9868.00421

FINKE, A., DOUCET, A. and JOHANSEN, A. M. (2016). On embedded hidden Markov models and particle Markov chain Monte Carlo methods. *arXiv e-prints* arXiv:1610.08962. https://doi.org/10.48550/arXiv.1610.08962

FINKE, A. and SINGH, S. S. (2017). Approximate smoothing and parameter estimation in high-dimensional state-space models. *IEEE Transactions on Signal Processing* **65** 5982–5994. https://doi.org/10.1109/TSP.2017.2733504

FRENKEL, D. (2004). Speed-up of Monte Carlo simulations by sampling of rejected states. *Proceedings of the National Academy of Sciences* **101** 17571–17575. https://doi.org/10.1073/pnas.0407950101

GUARNIERO, P., JOHANSEN, A. M. and LEE, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association* **112** 1636–1647. https://doi.org/10.1080/01621459.2016.1222291

HASTINGS, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109. https://doi.org/10.1093/biomet/57.1.97

KONG, A., LIU, J. S. and WONG, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* **89** 278–288. https://doi.org/10.1080/01621459.1994.10476469

LEE, A., SINGH, S. S. and VIHOLA, M. (2020). Coupled conditional backward sampling particle filter. *Annals of Statistics* **48** 3066–3089. https://doi.org/10.1214/19-AOS1922

LINDSTEN, F., DOUC, R. and MOULINES, E. (2015). Uniform ergodicity of the particle Gibbs sampler. *Scandinavian Journal of Statistics* **42** 775–797. https://doi.org/10.1111/sjos.12136

LINDSTEN, F., JORDAN, M. I. and SCHÖN, T. B. (2012). Ancestor sampling for particle Gibbs. In *Proceedings of the 2012 Conference on Neural Information Processing Systems*.

LIU, J. S. (1996). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika* **83** 681–682.

MALORY, S. (2021). Bayesian inference for stochastic processes, PhD thesis, Lancaster University. https://doi.org/10.17635/lancaster/thesis/1240

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21** 1087–1092. https://doi.org/10.1063/1.1699114

MURPHY, J. and GODSILL, S. J. (2015). Blocked particle Gibbs schemes for high dimensional interacting systems. *IEEE Journal of Selected Topics in Signal Processing* **10** 328–342. https://doi.org/10.1109/JSTSP.2015.2509940

NEAL, R. M. (2003). Markov Chain sampling for non-linear state space models using embedded hidden Markov models. *ArXiv Mathematics e-prints*. https://doi.org/10.48550/arXiv.math/0305039

NEAL, R. M., BEAL, M. J. and ROWEIS, S. T. (2004). Inferring state sequences for non-linear systems with embedded hidden Markov models. *Advances in Neural Information Processing Systems* **16** 401–408.

NEAL, P. and ROBERTS, G. (2011). Optimal scaling of random walk Metropolis algorithms with non-Gaussian proposals. *Methodology and Computing in Applied Probability* **13** 583–601. https://doi.org/10.1007/s11009-010-9176-9

PESKUN, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika* **60** 607–612. https://doi.org/10.1093/biomet/60.3.607

PITT, M. K. and SHEPHARD, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association* **94** 590–599. https://doi.org/10.1080/01621459.1999.10474153

REBESCHINI, P. and VAN HANDEL, R. (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability* **25** 2809–2866. https://doi.org/10.1214/14-AAP1061

ROBERTS, G. O., GELMAN, A. and GILKS, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability* **7** 110–120. https://doi.org/10.1214/aoap/1034625254

SCHWEDES, T. and CALDERHEAD, B. (2018). Quasi Markov chain Monte Carlo methods. *arXiv e-prints* arXiv:1807.00070. https://doi.org/10.48550/arXiv.1807.00070

SHERLOCK, C. and ROBERTS, G. (2009). Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli* **15** 774–798. https://doi.org/10.3150/08-BEJ176

SHESTOPALOFF, A. and DOUCET, A. (2019). Replica Conditional Sequential Monte Carlo. In *International Conference on Machine Learning* 5749–5757.

SHESTOPALOFF, A. Y. and NEAL, R. M. (2013). MCMC for non-linear state space models using ensembles of latent sequences. *ArXiv e-prints*. https://doi.org/10.48550/arXiv.1305.0320

SHESTOPALOFF, A. Y. and NEAL, R. M. (2018). Sampling latent states for high-dimensional non-linear state space models with the embedded HMM method. *Bayesian Analysis* **13** 797–822. https://doi.org/10.1214/17-BA1077

SINGH, S. S., LINDSTEN, F. and MOULINES, E. (2017). Blocking strategies and stability of particle Gibbs samplers. *Biometrika* **104** 953–969. https://doi.org/10.1093/biomet/asx051

TJELMELAND, H. (2004). Using all Metropolis–Hastings proposals to estimate mean values preprint No. 4/2004, Norwegian University of Science and Technology, Trondheim, Norway.

VAN LEEUWEN, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review* **137** 4089–4114. https://doi.org/10.1175/2009MWR2835.1

WHITELEY, N. (2010). Contribution to the discussion on 'Particle Markov chain Monte Carlo methods' by Andrieu, C., Doucet, A., and Holenstein, R. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72** 306–307.

YANG, J., ROBERTS, G. O. and ROSENTHAL, J. S. (2020). Optimal scaling of random-walk Metropolis algorithms on general target distributions. *Stochastic Processes and their Applications* **130** 6094–6132. https://doi.org/10.1016/j.spa.2020.05.004

YANG, S., CHEN, Y., BERNTON, E. and LIU, J. S. (2018). On parallelizable Markov chain Monte Carlo algorithms with waste-recycling. *Statistics and Computing* **28** 1073–1081. https://doi.org/10.1007/s11222-017-9780-4

APPENDIX A: FINITE-STATE HIDDEN MARKOV MODEL INTERPRETATION

In this section, we provide additional intuition for the validity for the steps that sample the particle indices $K_{1:T} = k_{1:T} \in [N]_0^T$ of the new reference path $\mathbf{X}'_{1:T} = (\mathbf{Z}_1^{K_1}, \ldots, \mathbf{Z}_T^{K_t})$ within the RW-EHMM and i-RW-CSMC algorithms (Algorithms 2 and 3). Our discussion is based around ideas from Neal (2003); Neal, Beal and Roweis (2004); Finke, Doucet and Johansen (2016). Central to it will be the following distribution over $k_{1:T} \in [N]_0^T$ which was introduced in the RW-EHMM algorithm:

$$\xi_T(\mathbf{z}_{1:T}, \{k_{1:T}\}) = \frac{\boldsymbol{\pi}_{T,D}(\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T})}{\sum_{l_{1:T} \in [N]_0^T} \boldsymbol{\pi}_{T,D}(\mathbf{z}_1^{l_1}, \ldots, \mathbf{z}_T^{l_T})}.$$

Recall that conditional on the set of particles $\mathbf{Z}_{1:T} = \mathbf{z}_{1:T}$:

1. the RW-EHMM Algorithm draws $K_{1:T} \sim \xi_T(\mathbf{z}_{1:T}, \cdot)$;
2. the i-RW-CSMC Algorithm draws $K_{1:T}$ according to a $\xi_T(\mathbf{z}_{1:T}, \cdot)$-invariant Markov kernel (see Propositions 4.1 and 4.2).

Our main results in this section are then the following:

1. In Subsection A.1, we show that the distribution $\xi_T(\mathbf{z}_{1:T}, \cdot)$ can be viewed as the joint posterior distribution of all T latent states in a finite-state hidden Markov model (HMM) with state space $[N]_0$.
2. In Subsection A.2, we then show that the recursions for sampling $K_{1:T} = k_{1:T}$ via Step 2 of the RW-EHMM Algorithm in $\mathrm{O}(N^2T)$ operations (see Subsection 3.1.2) are simply the forward filtering–backward sampling recursions for sampling from the joint posterior distributions of the latent states in the HMM from A.1.
3. In Subsection A.3, we then show that the recursions for sampling $K_{1:T} = k_{1:T}$ via Steps 1a, 2 and 3 of the i-RW-CSMC Algorithm (in $\mathrm{O}(NT)$ operations) can be viewed as a slightly non-standard CSMC algorithm (potentially with backward sampling) for sampling from the joint posterior distributions of the latent states in the HMM from A.1.

A.1. Finite-state hidden Markov model. All the developments that follow are conditional on the some value of the set of particles $\mathbf{Z}_{1:T} = \mathbf{z}_{1:T}$. Hence, we will sometimes drop the dependence on $\mathbf{z}_{1:T}$ from the notation. For any $k_{1:T} \in [N]_0^T$, and any $t \in [T]$, set

$$f_t(k_t|k_{t-1}) \coloneqq \frac{\mathbf{m}_t(\mathbf{z}_{t-1}^{k_{t-1}}, \mathbf{z}_t^{k_t})\mathbf{G}_t(\mathbf{z}_t^{k_t})}{\sum_{n=0}^N \mathbf{m}_t(\mathbf{z}_{t-1}^{k_{t-1}}, \mathbf{z}_t^{n})\mathbf{G}_t(\mathbf{z}_t^{n})},$$

$$g_t(\tilde{y}_t|k_t) \coloneqq \begin{cases} \left[\sum_{n=0}^N \mathbf{m}_1(\mathbf{z}_1^n)\mathbf{G}_1(\mathbf{z}_1^n)\right] \sum_{n=0}^N \mathbf{m}_2(\mathbf{z}_1^{k_t}, \mathbf{z}_2^n)\mathbf{G}_2(\mathbf{z}_2^n), & \text{if } t=1, \\ \sum_{n=0}^N \mathbf{m}_{t+1}(\mathbf{z}_t^{k_t}, \mathbf{z}_{t+1}^n)\mathbf{G}_{t+1}(\mathbf{z}_{t+1}^n), & \text{if } t>1, \end{cases}$$

with our usual convention that any quantity with time subscript 0 is to be ignored, i.e. $f_1(\cdot|k_0) \equiv f_1(\cdot)$. Then clearly, $f_t(\cdot|k_{t-1})$ is a probability mass function on $[N]_0$. With this notation, $f_t(k_t|k_{t-1})$ and $g_t(\tilde{y}_t|k_t)$ can be interpreted as the transition and emission probabilities of a finite-state HMM (where "observations" $\tilde{y}_t$ are added to the notation to make it more intuitive). In particular, the joint probability of the first T states and the first $T-1$ observations of this HMM is

$$\begin{aligned} p(k_{1:T}, \tilde{y}_{1:T-1}) &= \prod_{t=1}^T f_t(k_t|k_{t-1})g_t(\tilde{y}_t|k_t) \\ &= \prod_{t=1}^T \mathbf{m}_t(\mathbf{z}_{t-1}^{k_{t-1}}, \mathbf{z}_t^{k_t})\mathbf{G}_t(\mathbf{z}_t^{k_t}) \propto \boldsymbol{\pi}_{T,D}(\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T}), \end{aligned}$$

which implies that the joint posterior distribution of these states is:

$$p(k_{1:T}|\tilde{y}_{1:T-1}) = \frac{\boldsymbol{\pi}_{T,D}(\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T})}{\sum_{l_{1:T} \in [N]_0^T} \boldsymbol{\pi}_{T,D}(\mathbf{z}_1^{l_1}, \ldots, \mathbf{z}_T^{l_T})} = \xi_T(\mathbf{z}_{1:T}, \{k_{1:T}\}).$$

A.2. Sampling $K_{1:T}$ within the RW-EHMM algorithm. We now show that the recursions for sampling $K_{1:T} = k_{1:T}$ via Step 2 of the RW-EHMM Algorithm in $\mathrm{O}(N^2T)$ operations (see Subsection 3.1.2) are simply the forward filtering–backward sampling recursions for the HMM from A.1.

A.2.1. *Forward filtering.* Specifically, (5) is then nothing more than the forward-filtering recursion which propagates the one-step ahead predictive distributions $W_t^{k_t} = p(k_t|\tilde{y}_{1:t-1})$. For $t = 1, \ldots, T$ (and with convention $w_0^n = 1$):

$$W_t^n = p(k_t = n|\tilde{y}_{1:t-1}) = \frac{p(k_t = n, \tilde{y}_{t-1}|\tilde{y}_{1:t-2})}{\sum_{l=0}^N p(k_t = l, \tilde{y}_{t-1}|\tilde{y}_{1:t-2})} = \frac{w_t^n}{\sum_{l=0}^N w_t^l},$$

$$\begin{aligned} w_t^n &= p(k_t = n, \tilde{y}_{t-1}|\tilde{y}_{1:t-2}) \\ &= \sum_{m=0}^N p(k_{t-1} = m|\tilde{y}_{t-2})g_{t-1}(\tilde{y}_{t-1}|m)f_t(n|m) \\ &= \sum_{k_{t-1} \in [N]_0} \frac{w_{t-1}^m}{\sum_{l=0}^N w_{t-1}^l} \mathbf{m}_t(\mathbf{z}_{t-1}^m, \mathbf{z}_t^n)\mathbf{G}_t(\mathbf{z}_t^n). \end{aligned}$$

A.2.2. *Backward sampling.* Likewise, (6) is nothing more than the backward-sampling recursion. That is, for $t = T-1, \ldots, 1$, we sample $K_t = k_t \in [N]_0$ with probability

$$p(k_t|k_{t+1:T}, \tilde{y}_{1:T-1}) = p(k_t|k_{t+1}, \tilde{y}_{1:t})$$

$$= \frac{p(k_t, k_{t+1}, \tilde{y}_{t-1:t}|\tilde{y}_{1:t-2})}{\sum_{n=0}^{N} p(k_t = n, k_{t+1}, \tilde{y}_{t-1:t}|\tilde{y}_{1:t-2})}$$
$$= \frac{p(k_t, \tilde{y}_{t-1}|\tilde{y}_{1:t-2}) g_t(\tilde{y}_t|k_t) f_{t+1}(k_{t+1}|k_t)}{\sum_{n=0}^{N} p(k_t = n, \tilde{y}_{t-1}|\tilde{y}_{1:t-2}) g_t(\tilde{y}_t|n) f_{t+1}(k_{t+1}|n)}$$
$$= \frac{w_t^{k_t} \mathbf{m}_{t+1}(\mathbf{z}_t^{k_t}, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{n=0}^{N} w_t^n \mathbf{m}_{t+1}(\mathbf{z}_t^n, \mathbf{z}_{t+1}^{k_{t+1}})}.$$

A.3. Sampling $K_{1:T}$ within the i-RW-CSMC algorithm. We now show that the recursions for sampling $K_{1:T} = k_{1:T}$ via Steps 1a, 2 and 3 of the i-RW-CSMC Algorithm (in $\mathrm{O}(NT)$ operations) can be viewed as first running a CSMC algorithm targetting the HMM from A.1 and then selecting a single particle lineage via ancestral tracing or backward sampling.

A.3.1. *Conditional particle filter with ancestral tracing.* Without backward sampling, Steps 1a, 2 and 3 of the i-RW-CSMC algorithm can be interpreted as running a slightly non-standard CSMC algorithm (with ancestral tracing) which targets the one-step-ahead predictive distributions and which employs $N + 1$ particles which are jointly "proposed" from the distribution $\delta_{(0,1,\ldots,N)}$ at each time step. That is, the value of each particle is deterministically set equal to its particle index. We note that this is a slightly non-standard CSMC algorithm because the particles are not proposed independently given the history of the particle system as in, e.g., a bootstrap particle filter. However, such "stratified" proposals have long been used for particle filters tailored to finite state spaces such as the *discrete particle filter* from Fearnhead (1998); Fearnhead and Clifford (2003). The nth unnormalised particle weight of the CSMC algorithm targetting the finite-state HMM at time t can then be shown to be given by

$$w_t^n := g_{t-1}(\tilde{y}_{t-1}|a_{t-1}^n) f_t(n|a_{t-1}^n) = \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n) \mathbf{G}_t(\mathbf{z}_t^n).$$

A.3.2. *Conditional particle filter with Backward sampling.* If instead we employ backward sampling to sample a new particle path in the CSMC algorithm targetting the finite-state HMM, then this gives the backward-sampling probabilities from (7):

$$\frac{w_t^{k_t} g_t(\tilde{y}_t|k_t) f_{t+1}(k_{t+1}|k_t)}{\sum_{m=0}^{N} w_t^m g_t(\tilde{y}_t|m) f_{t+1}(k_{t+1}|m)} = \frac{\mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^{k_t}}, \mathbf{z}_t^{k_t}) \mathbf{G}_t(\mathbf{z}_t^{k_t}) \mathbf{m}_{t+1}(\mathbf{z}_t^{k_t}, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^{N} \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m) \mathbf{G}_t(\mathbf{z}_t^m) \mathbf{m}_{t+1}(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}})}.$$

APPENDIX B: DETAILS FOR SECTION 2

B.1. Joint law induced by Algorithm 1. We now formally define the joint law of all random variables generated in Algorithm 1. This will be used in some of the proofs below.

To simplify the presentation, we note that we fixed the reference path in Algorithm 1 to always have particle index 0, i.e. we always set $\mathbf{Z}_t^0 := \mathbf{z}_t^0 := \mathbf{x}_t$ as well as $A_{t-1}^0 = a_{t-1}^0 := 0$. However, in some of the proofs below, it is more convenient to work with a slightly more general version of the algorithm which, at the beginning each time step, draws a particle index $J_t = j_t$ from a uniform distribution on $[N]_0$ and then sets $A_{t-1}^{j_t} = a_{t-1}^{j_t} := j_{t-1}$ as well as $\mathbf{Z}_t^{j_t} := \mathbf{z}_t^{j_t} := \mathbf{x}_t$.

Conditional on $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, the joint law of all random variables $(J_{1:T}, \mathbf{Z}_{1:T}, A_{1:T-1}, K_{1:T}, \mathbf{X}'_{1:T})$ generated by this slightly generalised version of Algorithm 1 may be written as

$$\bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}(\mathrm{d}j_{1:T} \times \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T})$$

$$
\begin{aligned}
&\coloneqq \mathrm{Unif}_{[N]_0^T}(\mathrm{d}j_{1:T})\,\delta_{\mathbf{x}_{1:T}}(\mathrm{d}\mathbf{z}_1^{j_1}\times\cdots\times\mathrm{d}\mathbf{z}_T^{j_T})\left[\prod_{\substack{n=0\\ n\neq j_1}}^{N}\mathbf{M}_1(\mathrm{d}\mathbf{z}_1^n)\right]\\
&\quad\times\left[\prod_{t=2}^{T}\delta_{j_{t-1}}(\mathrm{d}a_{t-1}^{j_t})\prod_{\substack{n=0\\ n\neq j_t}}^{N}R_{t-1,D}^N(\mathbf{z}_{t-1},\mathrm{d}a_{t-1}^n)\mathbf{M}_t(\mathbf{z}_{t-1}^{a_{t-1}^n},\mathrm{d}\mathbf{z}_t^n)\right]\\
&\quad\times R_{T,D}^N(\mathbf{z}_T,\mathrm{d}k_T)\\
&\quad\times\begin{cases}\left[\prod_{t=1}^{T-1}\delta_{a_t^{k_{t+1}}}(\mathrm{d}k_t)\right] & \text{[without backward sampling]}\\ \left[\prod_{t=1}^{T-1}B_{t,D}^N((\mathbf{z}_t,\mathbf{z}_{t+1}^{k_{t+1}}),\mathrm{d}k_t)\right] & \text{[with backward sampling]}\end{cases}\\
&\quad\times\delta_{(\mathbf{z}_1^{k_1},\ldots,\mathbf{z}_T^{k_T})}(\mathrm{d}\mathbf{x}'_{1:T}).
\end{aligned}
\tag{8}
$$

Here, we have defined the following quantities.

- **Resampling kernels.** For any $n\in[N]_0$ and any $t\in[T]$,

$$
R_{t,D}^N(\mathbf{z}_t,\{n\})\coloneqq\Psi^n(\{\mathbf{w}_t(\mathbf{z}_t^m)-\mathbf{w}_t(\mathbf{z}_t^0)\}_{m=1}^N)=\frac{\mathbf{G}_t(\mathbf{z}_t^n)}{\sum_{m=0}^N\mathbf{G}_t(\mathbf{z}_t^m)}.
$$

When using the forced-move extension, replace $R_{T,D}^N(\mathbf{z}_T,\{k_T\})$ at time $t=T$ in (8) by

$$
\begin{cases}\dfrac{\mathbf{G}_T(\mathbf{z}_T^{k_T})}{\sum_{m=0}^N\mathbf{G}_T(\mathbf{z}_T^m)-\mathbf{G}_T(\mathbf{z}_T^{k_T})\wedge\mathbf{G}_T(\mathbf{z}_T^{j_T})}, & \text{if } k_T\neq j_T,\\ 1-\displaystyle\sum_{\substack{l=0\\ l\neq j_T}}^{N}\frac{\mathbf{G}_T(\mathbf{z}_T^l)}{\sum_{m=0}^N\mathbf{G}_T(\mathbf{z}_T^m)-\mathbf{G}_T(\mathbf{z}_T^l)\wedge\mathbf{G}_T(\mathbf{z}_T^{j_T})}, & \text{if } k_T=j_T.\end{cases}
$$

- **Backward kernels.** For $t\in[T-1]$,

$$
\begin{aligned}
B_{t,D}^N((\mathbf{z}_t,\mathbf{z}_{t+1}^{k_{t+1}}),\{n\})&\coloneqq\Psi^n(\{\mathbf{v}_t(\mathbf{z}_t^m,\mathbf{z}_{t+1}^{k_{t+1}})-\mathbf{v}_t(\mathbf{z}_t^0,\mathbf{z}_{t+1}^{k_{t+1}})\}_{m=1}^N)\\
&=\frac{\mathbf{G}_t(\mathbf{z}_t^n)\mathbf{m}_{t+1}(\mathbf{z}_t^n,\mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^N\mathbf{G}_t(\mathbf{z}_t^m)\mathbf{m}_{t+1}(\mathbf{z}_t^m,\mathbf{z}_{t+1}^{k_{t+1}})}.
\end{aligned}
$$

From this definition, we can recover the joint law of all random variables $(\mathbf{Z}_{1:T},A_{1:T-1},K_{1:T},\mathbf{X}'_{1:T})$ generated in Steps 1–4 of Algorithm 1 by conditioning on the event $\{J_1=0,\ldots,J_T=0\}$, i.e.

$$
\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^N\coloneqq\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^{N,\star}(\,\cdot\,|J_1=0,\ldots,J_T=0).
$$

Let $\mathbb{E}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$ denote expectation w.r.t. $\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$. In the remainder of this section, we will sometimes work with $\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$ rather than with $\mathbb{P}_{T,D,\mathbf{x}_{1:T}}^{N}$. This is justified because both versions of the i-CSMC algorithm induce the same Markov kernel:

$$
\begin{aligned}
\mathbb{E}_{T,D,\mathbf{x}_{1:T}}^{N,\star}[\mathbb{I}\{\mathbf{X}'_{1:T}\in\mathrm{d}\mathbf{x}'_{1:T}\}]&=\mathbb{E}_{T,D,\mathbf{x}_{1:T}}^{N}[\mathbb{I}\{\mathbf{X}'_{1:T}\in\mathrm{d}\mathbf{x}'_{1:T}\}]\\
&=\mathbf{P}_{T,D}^N(\mathbf{x}_{1:T},\mathrm{d}\mathbf{x}'_{1:T}).
\end{aligned}
$$

B.2. Proof of Proposition 2.1.

PROOF (of Proposition 2.1). For the plain algorithm (with neither the backward sampling nor the forced-move extension) we can readily check that

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T})\mathbb{P}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{x}'_{1:T})\\&=\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})\mathbb{P}^{N,\star}_{T,D,\mathbf{x}'_{1:T}}(\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{x}_{1:T}),\end{aligned}\tag{9}$$

i.e. (9) admits $\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})$ as a marginal.

For the backward-sampling extension, let $\mathring{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ be the same as $\mathbb{P}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ (without backward sampling) except that the terms $\delta_{j_{t-1}}(\mathrm{d}a^{j_t}_{t-1})$ in (8) are replaced by $B^N_{t-1,D}((\mathbf{z}_{t-1},\mathbf{z}^{j_t}_t),\{a^{j_t}_{t-1}\})$. Then

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T})\mathbb{P}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{x}'_{1:T})\\&=\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})\mathring{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}'_{1:T}}(\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{x}_{1:T}),\end{aligned}\tag{10}$$

That is, (10) again admits $\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})$ as a marginal. Incidentally, $\mathring{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ can be recognised as the law of all the random variables generated by the i-CSMC algorithm with ancestor sampling from Lindsten, Jordan and Schön (2012).

Finally, the algorithm with the forced-move extension can be justified as a partially collapsed Gibbs sampler because this extension leaves the marginal distribution of K_T (under (9) without backward sampling or under (10) with backward sampling) conditional on $(\mathbf{X}_{1:T},J_{1:T},\mathbf{Z}_{1:T},A_{1:T})$ invariant. □

B.3. Verification of Assumptions A2 and A3 in a linear-Gaussian state-space model.

Consider a state-space model with D-dimensional observations $\mathbf{y}_t=y_{t,1:D}\in\mathbb{R}^D$ and

$$\begin{aligned}\mathbf{H}_t(\mathbf{x}_t,\mathrm{d}\mathbf{y}_t)&\coloneqq\textstyle\prod_{d=1}^D\mathrm{N}(\mathrm{d}y_{t,d};x_{t,d},1),\\\mathbf{M}_t(\mathbf{x}_{t-1},\mathrm{d}\mathbf{x}_t)&\coloneqq\textstyle\prod_{d=1}^D\mathrm{N}(\mathrm{d}x_{t,d};x_{t-1,d},1).\end{aligned}$$

Let φ be a density function of a univariate standard normal distribution. Then this model satisfies A1 with

$$\begin{aligned}G_t(x_{t,d})&=\varphi(y_{t,d}-x_{t,d}),\\m_t(x_{t-1,d},x_{t,d})&=\varphi(x_{t,d}-x_{t-1,d}).\end{aligned}$$

To simplify the notation and calculations, we hereafter drop the subscript d, take $y_1=\ldots,y_T=0$ and assume as initial density $m_1(x_1)=\varphi(x_1/\sqrt{\sigma^2+1})/\sqrt{\sigma^2+1}$, where $\sigma^2\coloneqq(\sqrt{5}-1)/2$. Standard Kalman-filtering and Kalman-smoothing recursions then give $\mathbb{P}(X_t\in\mathrm{d}x_t|Y_{1:t}=y_{1:t})=\mathrm{N}(\mathrm{d}x_t;0,\sigma^2)$ as well as $\pi_T(\mathrm{d}x_{1:T})=\mathbb{P}(X_{1:T}\in\mathrm{d}x_{1:T}|Y_{1:T}=y_{1:T})=\mathrm{N}(\mathrm{d}x_{1:T};\mathbf{0}_T,C)$ with $[C]_{s,t}=u^{|t-s|}\sigma^2_{s\vee t}$, where $u\coloneqq\sigma^2/(\sigma^2+1)$ and

$$\sigma^2_t\coloneqq u\frac{1-(u^2)^{T-t}}{1-u^2}\mathbb{I}\{t<T\}+(u^2)^{T-t}\sigma^2.$$

Tedious but simple algebra then shows that A2 and A3 hold because

$$\underline{b}_T>\underline{r}_T=r_{T|T}=\left\{\begin{array}{ll}\frac{1}{2}(\log(\sigma^2+2)-\sigma^2), & \text{if } T=1,\\\frac{1}{2}(\log(2)+[\sigma^2(u^2-2)+u]/2), & \text{if } T>1,\end{array}\right\}>0.15.$$

B.4. Proof of Proposition 2.2.

PROOF (of Proposition 2.2). A telescoping-sum argument gives the following decomposition which will form the basis of the proof both in the case with and without resampling. Here, we let $\mathbf{0}$ denote a vector of zeros of appropriate length.

$$
\begin{aligned}
d^N_{T,D,\mathbf{x}_{1:T}} \leq & \sum_{t=1}^{T-1}\sum_{n=1}^{N} \big\| \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{A^n_t \in \cdot\} | A_{1:t-1} = \mathbf{0}] - \delta_0(\cdot) \big\| \\
& + \big\| \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_T \in \cdot\} | A_{1:T-1} = \mathbf{0}] - \delta_0(\cdot) \big\| \\
& + \sum_{t=1}^{T-1} \big\| \mathbb{E}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{K_t \in \cdot\} | (A_{1:T-1}, K_{t+1:T}) = \mathbf{0}] - \delta_0(\cdot) \big\|. \qquad (11)
\end{aligned}
$$

First, we consider the case *without* backward sampling. We write

$$
\begin{aligned}
\mathcal{R}_{t|T,D}(\mathbf{x}_{t-1:t}) &\coloneqq [\tfrac{1}{D}\textstyle\sum_{d=1}^{D} \log M_t(G_t)(x_{t-1,d}) - \log G_t(x_{t,d})] + r_{t|T} \\
&= \tfrac{1}{D}\textstyle\sum_{d=1}^{D} \log M_t(G_t)(x_{t-1,d}) - \mathbb{E}[\log M_t(G_t)(X_{t-1})] \\
&\quad + \tfrac{1}{D}\textstyle\sum_{d=1}^{D} \mathbb{E}[\log G_t(X_t)] - \log G_t(x_{t,d}).
\end{aligned}
$$

For some $\eta \in (0, 1/2)$, we set

$$
\mathbf{F}_{T,D} \coloneqq \{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid \forall\, t \in [T] : |\mathcal{R}_{t|T,D}(\mathbf{x}_{t-1:t})| < D^{-\eta}\}.
$$

Since $\eta < 1/2$ implies that $D^\eta \sqrt{2D \log\log D} = \mathrm{O}(D)$, the law of the iterated logarithm then implies that $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$.

We can now turn to the terms in the decomposition (11). Let $(\mathbf{x}_{1:T,D})_{D\geq 1}$ be some sequence in $(\mathbf{E}_{T,D})_{D\geq 1}$, i.e. $\mathbf{x}_{t,D} = x_{t,1:D,D} \in \mathbb{R}^D$, for any $D \geq 1$. For any $n \in [N]$ and $t \in [T]$, let $\mathbf{Z}^n_{t,D} \sim \mathbf{M}_t(\mathbf{x}_{t-1,D}, \cdot)$ and $\mathbf{Z}^0_{t,D} \coloneqq \mathbf{x}_{t,D}$. For any $t \in [T]$ (with A^n_t replaced by K_T if $t = T$), we have that

$$
\begin{aligned}
&|\mathbb{E}^N_{T,D,\mathbf{x}_{1:T,D}}[\mathbb{I}\{A^n_t \in [N]\} | A_{1:t-1} = \mathbf{0}] - \delta_0([N])| \\
&\quad = \mathbb{E}^N_{T,D,\mathbf{x}_{1:T,D}}\left[\frac{\sum_{n=1}^N \exp\{\mathbf{w}_t(\mathbf{Z}^n_{t,D}) - \mathbf{w}_t(\mathbf{Z}^0_{t,D})\}}{1 + \sum_{m=1}^N \exp\{\mathbf{w}_t(\mathbf{Z}^m_{t,D}) - \mathbf{w}_t(\mathbf{Z}^0_{t,D})\}} \,\middle|\, A_{1:t-1} = \mathbf{0} \right].
\end{aligned}
$$

For any $n \in [N]$ and $t \in [T]$, let $\mathbf{Z}^n_{t,D} \sim \mathbf{M}_t(\mathbf{x}_{t-1,D}, \cdot)$ and $\mathbf{Z}^0_{t,D} \coloneqq \mathbf{x}_{t,D}$. To complete the proof, it then suffices to show that

$$
\textstyle\sum_{n=1}^N \exp\{\mathbf{w}_t(\mathbf{Z}^n_{t,D}) - \mathbf{w}_t(\mathbf{Z}^0_{t,D})\} \to_{\mathbb{P}} 0,
$$

as $D \to \infty$, whenever $\log N = \mathrm{o}(D)$. By Markov's inequality, for any $\varepsilon > 0$,

$$
\begin{aligned}
&\mathbb{P}(\{\textstyle\sum_{n=1}^N \exp\{\mathbf{w}_t(\mathbf{Z}^n_{t,D}) - \mathbf{w}_t(\mathbf{Z}^0_{t,D})\} > \varepsilon\}) \\
&\quad \leq N\varepsilon^{-1} \exp\{-Dr_{t|T} + D|\mathcal{R}_{t|T,D}(\mathbf{x}_{t-1:t,D})|\} \\
&\quad \leq N\varepsilon^{-1} \exp\{-Dr_{t|T} + D^{1-\eta}\},
\end{aligned}
$$

and $r_{t|T} \geq \underline{r}_T > 0$ by A2. This completes the proof in the case without backward sampling.

We now consider the case *with* backward sampling. We write

$$
\begin{aligned}
&\mathcal{B}_{t|T,D}(\mathbf{x}_{t-1:t+1}) \\
&\quad \coloneqq [\tfrac{1}{D}\textstyle\sum_{d=1}^{D} \log M_t(G_t m_{t+1}(\cdot, x_{t+1,d}))(x_{t-1,d}) - v_t(x_{t:t+1,d})] + b_{t|T} \\
&\quad = \tfrac{1}{D}\textstyle\sum_{d=1}^{D} \log M_t(G_t m_{t+1}(\cdot, x_{t+1,d}))(x_{t-1,d}) - \mathbb{E}[\log M_t(G_t m_{t+1}(\cdot, X_{t+1}))(X_{t-1})] \\
&\qquad + \tfrac{1}{D}\textstyle\sum_{d=1}^{D} \mathbb{E}[\log\{G_t(X_t) m_{t+1}(X_t, X_{t+1})\}] - \log\{G_t(x_{t,d} m_{t+1}(x_{t,d}, x_{t+1,d})\}.
\end{aligned}
$$

with convention $\mathcal{B}_{T|T,D} = \mathcal{R}_{T|T,D}$. For some $\eta \in (0, 1/2)$, we set

$$\mathbf{F}_{T,D} \coloneqq \{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid \forall\, t \in [T] : |\mathcal{R}_{t|T,D}(\mathbf{x}_{t-1:t})| \vee |\mathcal{B}_{t|T,D}(\mathbf{x}_{t-1:t+1})| < D^{-\eta}\}.$$

Since $\eta < 1/2$ implies that $D^{\eta}\sqrt{2D \log\log D} = \mathrm{O}(D)$, the law of the iterated logarithm then again implies that $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$.

Again we consider the decomposition (11). We then have that

$$\begin{aligned}
&|\mathbb{E}^N_{T,D,\mathbf{x}_{1:T,D}}[\mathbb{I}\{K_t \in [N]\}|(A_{1:T-1}, K_{t+1}) = \mathbf{0}] - \delta_0([N])| \\
&= \mathbb{E}^N_{T,D,\mathbf{x}_{1:T,D}}\left[\frac{\sum_{n=1}^N \exp\{\mathbf{v}_t(\mathbf{Z}^n_{t,D}, \mathbf{Z}^0_{t+1,D}) - \mathbf{v}_t(\mathbf{Z}^0_{t,D}, \mathbf{Z}^0_{t+1,D})\}}{1 + \sum_{m=1}^N \exp\{\mathbf{v}_t(\mathbf{Z}^m_{t,D}, \mathbf{Z}^0_{t+1,D}) - \mathbf{v}_t(\mathbf{Z}^0_{t,D}, \mathbf{Z}^0_{t+1,D})\}} \,\middle|\, A_{1:t-1} = \mathbf{0}\right].
\end{aligned}$$

To complete the proof, it suffices to show that

$$\textstyle\sum_{n=1}^N \mathbf{v}_t(\mathbf{Z}^n_{t,D}, \mathbf{Z}^0_{t+1,D}) - \mathbf{v}_t(\mathbf{Z}^0_{t,D}, \mathbf{Z}^0_{t+1,D}) \to_{\mathbb{P}} 0,$$

as $D \to \infty$, whenever $\log N = \mathrm{o}(D)$. By Markov's inequality, for any $\varepsilon > 0$,

$$\begin{aligned}
&\mathbb{P}(\{\textstyle\sum_{n=1}^N \exp\{\mathbf{v}_t(\mathbf{Z}^n_{t,D}, \mathbf{Z}^0_{t+1,D}) - \mathbf{v}_t(\mathbf{Z}^0_{t,D}, \mathbf{Z}^0_{t+1,D}) > \varepsilon\}) \\
&\quad \le N\varepsilon^{-1} \exp\{-Db_{t|T} + D|\mathcal{B}_{t|T,D}(\mathbf{x}_{t-1:t+1,D})|\} \\
&\quad \le N\varepsilon^{-1} \exp\{-Db_{t|T} + D^{1-\eta}\},
\end{aligned}$$

and $b_{t|T} \ge \underline{b}_T > 0$ by **A3**. This completes the proof in the case with backward sampling. □

APPENDIX C: DETAILS FOR SECTION 3

C.1. Joint law induced by Algorithm 2. We now formally define the joint law of all random variables generated in Algorithm 2. This will be used in some of the proofs below.

To simplify the presentation, we note that we again fixed the reference path in Algorithm 2 to always have particle index 0, i.e. we always set $\mathbf{Z}^0_t \coloneqq \mathbf{z}^0_t \coloneqq \mathbf{x}_t$. However, in some of the proofs below, it is more convenient to work with a slightly more general version of the algorithm which, at each time step, draws a particle index $J_t = j_t$ from a uniform distribution on $[N]_0$ and then sets $\mathbf{Z}^{j_t}_t \coloneqq \mathbf{z}^{j_t}_t \coloneqq \mathbf{x}_t$.

Conditional on $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, the joint law of all random variables $(J_{1:T}, \mathbf{Z}_{1:T}, K_{1:T}, \mathbf{X}'_{1:T})$ generated by this slightly generalised version of Algorithm 2 may be written as

$$\begin{aligned}
&\widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T} \times \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T}) \\
&\quad \coloneqq \mathrm{Unif}_{[N]_0^T}(\mathrm{d}j_{1:T})\, \delta_{\mathbf{x}_{1:T}}(\mathrm{d}\mathbf{z}^{j_1}_1 \times \cdots \times \mathrm{d}\mathbf{z}^{j_T}_T) \left[\prod_{t=1}^T S^N_{t,D}(\mathbf{z}^{j_t}_t, \mathrm{d}\mathbf{z}^{-j_t}_t)\right] \\
&\qquad \times \xi_T(\mathbf{z}_{1:T}, \mathrm{d}k_{1:T})\, \delta_{(\mathbf{z}^{k_1}_1, \ldots, \mathbf{z}^{k_T}_T)}(\mathrm{d}\mathbf{x}'_{1:T}).
\end{aligned}$$

Here, we have defined $\mathbf{z}^{-n}_t \coloneqq (\mathbf{z}^0_t, \ldots, \mathbf{z}^{n-1}_t, \mathbf{z}^{n+1}_t, \ldots, \mathbf{z}^N_t)$ as well as the following quantities.

- **Proposal kernels.** For any $t \in [T]$ and any $n \in [N]_0$,

$$S^N_{t,D}(\mathbf{z}^n_t, \mathrm{d}\mathbf{z}^{-n}_t) \coloneqq \prod_{d=1}^D \mathrm{N}\big(\mathrm{d}z^{-n}_{t,d}; z^n_{t,d}\mathbf{1}_N, \tfrac{\ell_t}{D}\Sigma\big),$$

  where $z^{-n}_{t,d} \coloneqq (z^0_{t,d}, \ldots, z^{n-1}_{t,d}, z^{n+1}_{t,d}, \ldots, z^N_{t,d})$.
- **Selection probability.**

$$\xi_T(\mathbf{z}_{1:T}, \{k_{1:T}\}) \coloneqq \frac{\boldsymbol{\pi}_{T,D}(\mathbf{z}^{k_1}_1, \ldots, \mathbf{z}^{k_T}_T)}{\sum_{l_{1:T} \in [N]_0^T} \boldsymbol{\pi}_{T,D}(\mathbf{z}^{l_1}_1, \ldots, \mathbf{z}^{l_T}_T)}.$$

From this definition, we can recover the joint law of all random variables $(\mathbf{Z}_{1:T}, K_{1:T}, \mathbf{X}'_{1:T})$ generated in Steps 1–3 of Algorithm 2 by conditioning on the event $\{J_1 = 0, \dots, J_T = 0\}$, i.e.

$$\widetilde{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}} := \widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\,\cdot\,|J_1 = 0, \dots, J_T = 0).$$

Let $\widetilde{\mathbb{E}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ denote expectation w.r.t. $\widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$. In the remainder of this section, we will sometimes work with $\widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ rather than with $\widetilde{\mathbb{P}}^N_{T,D,\mathbf{x}_{1:T}}$. This is justified because both versions of the RW-EHMM algorithm induce the same Markov kernel:

$$\begin{aligned}\widetilde{\mathbb{E}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}] &= \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}] \\ &= \widetilde{\mathbf{P}}^N_{T,D}(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}).\end{aligned}$$

C.2. Proof of Proposition 3.1.

PROOF (of Proposition 3.1). Recall that by (4), the random-walk type proposal used to scatter the particles around the reference path is symmetric in the sense that

$$\lambda(\mathrm{d}\mathbf{z}^j_t) S^N_{t,D}(\mathbf{z}^j_t, \mathrm{d}\mathbf{z}^{-j}_t) = \lambda(\mathrm{d}\mathbf{z}^k_t) S^N_{t,D}(\mathbf{z}^k_t, \mathrm{d}\mathbf{z}^{-k}_t),$$

for any $j, k \in [N]_0$, where $\mathbf{z}^{-n}_t := (\mathbf{z}^0_t, \dots, \mathbf{z}^{n-1}_t, \mathbf{z}^{n+1}_t, \dots, \mathbf{z}^N_t)$ and where λ denotes a suitable version of the Lebesgue measure.

We can then readily check that

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T}) \widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T} \times \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T}) \\ &\quad = \boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T}) \widetilde{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}'_{1:T}}(\mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}j_{1:T} \times \mathrm{d}\mathbf{x}_{1:T}),\end{aligned} \tag{12}$$

i.e. (12) admits $\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})$ as a marginal. □

C.3. Proof of Proposition 3.3.

PROOF (of Proposition 3.3). The first part (the convergence statement) follows immediately from Proposition 3.2. For the second part (the lower bound), we note that

$$\bar{\alpha}^N_T(t) = \sum_{n \in [N]} \widetilde{\mathbb{E}}^N_T[\Psi^n(\{V^l_t\}_{l=1}^N)] \geq \left(1 + \frac{\exp(\ell_t \mathcal{I}_{t|T})}{N}\right)^{-1},$$

where the penultimate inequality follows by Lemma D.6 in Appendix D.6. □

C.4. Proof of Proposition 3.4.

PROOF (of Proposition 3.4). Let $\mathbf{F}_{T,D} \in \boldsymbol{\mathcal{E}}_{T,D}$ be as in Proposition 3.2. We then have

$$\begin{aligned}&\big|\widetilde{\mathrm{ESJD}}^N_{T,D}(t) - \ell_t \tilde{\alpha}^N_T(t)\big| \\ &\quad = \big|\textstyle\int_{\mathbf{E}_{T,D}} \boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T}) \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\|\mathbf{X}'_t - \mathbf{x}_t\|_2^2\big] - \ell_t \tilde{\alpha}^N_T(t)\big| \\ &\quad \leq \ell_t \sup_{\mathbf{x}_{1:T} \in \mathbf{F}_{T,D}} \big|\tilde{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t) - \tilde{\alpha}^N_T(t)\big| \\ &\qquad + \sup_{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}} \big|\widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\|\mathbf{X}'_t - \mathbf{x}_t\|_2^2\big] - \ell_t \tilde{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t)\big| \\ &\qquad + \boldsymbol{\pi}_{T,D}(\mathbf{E}_{T,D} \setminus \mathbf{F}_{T,D}) \sup_{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}} \big|\widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\|\mathbf{X}'_t - \mathbf{x}_t\|_2^2\big]\big|.\end{aligned} \tag{13}$$

We now consider the limit of each of the terms in the last line of (13) as $D \to \infty$. The first term converges to zero by Proposition 3.3. Take $U^n_{t,d} = D^{1/2}\ell_t^{-1/2}(Z^n_{t,d} - x_{t,d})$ as in Algorithm 2, then $\bar{U}^n_{t,D} \coloneqq D^{-1}\sum_{d=1}^D (U^n_{t,d})^2 \to_{\mathbb{P}} 1$, as $D \to \infty$, and hence $|\bar{U}^n_{t,D} - 1| \to_{\mathbb{P}} 0$ and $|\bar{U}^n_{t,D}| \to_{\mathbb{P}} 1$ by the continuous mapping theorem. Furthermore, $(|\bar{U}^n_{t,D} - 1|)_{D\geq 1}$ and $(|\bar{U}^n_{t,D}|)_{D\geq 1}$ are uniformly integrable. Thus, for the second term, for any $\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}$,

$$\begin{aligned}
&\big|\widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\|\mathbf{X}'_t - \mathbf{x}_t\|_2^2\big] - \ell_t \tilde{\alpha}^N_{T,D,\mathbf{x}_{1:T}}(t)\big| \\
&\qquad \leq \big|\ell_t \textstyle\sum_{n=1}^N \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\big(\bar{U}^n_{t,D} - 1\big)\,\mathbb{I}\{K_t = n\}\big]\big| \\
&\qquad \leq \ell_t N\,\mathbb{E}\big[\big|\bar{U}^1_{t,D} - 1\big|\big] \to 0.
\end{aligned}$$

Similarly, for any $\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}$,

$$\begin{aligned}
\big|\widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\|\mathbf{X}'_t - \mathbf{x}_t\|_2^2\big]\big| &\leq \big|\ell_t \textstyle\sum_{n=1}^N \widetilde{\mathbb{E}}^N_{T,D,\mathbf{x}_{1:T}}\big[\bar{U}^n_{t,D}\,\mathbb{I}\{K_t = n\}\big]\big| \\
&\leq \ell_t N\,\mathbb{E}\big[\big|\bar{U}^1_{t,D}\big|\big] \to 1,
\end{aligned}$$

and the third term therefore converges to zero since $\pi(\mathbf{E}_{t,D} \setminus \mathbf{F}_{T,D}) \to 0$. □

APPENDIX D: DETAILS FOR SECTION 4

D.1. Joint law induced by Algorithm 3. We now formally define the joint law of all random variables generated in Algorithm 3. This will be used in some of the proofs below.

To simplify the presentation, we note that we again fixed the reference path in Algorithm 3 to always have particle index 0, i.e. we always set $\mathbf{Z}^0_t \coloneqq \mathbf{z}^0_t \coloneqq \mathbf{x}_t$ as well as $A^0_{t-1} = a^0_{t-1} \coloneqq 0$. However, in some of the proofs below, it is more convenient to work with a slightly more general version of the algorithm which, at the beginning each time step, draws a particle index $J_t = j_t$ from a uniform distribution on $[N]_0$ and then sets $A^{j_t}_{t-1} = a^{j_t}_{t-1} \coloneqq j_{t-1}$ as well as $\mathbf{Z}^{j_t}_t \coloneqq \mathbf{z}^{j_t}_t \coloneqq \mathbf{x}_t$.

Conditional on $\mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l]$, the joint law of all random variables $(J_{1:T}, \mathbf{Z}_{1:T}, A_{1:T-1}, K_{1:T}, \mathbf{X}'_{1:T})$ generated by this slightly generalised version of Algorithm 3 may be written as

$$\begin{aligned}
&\bar{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T} \times \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T} \times \mathrm{d}\mathbf{x}'_{1:T}) \\
&\quad \coloneqq \mathrm{Unif}_{[N]_0^T}(\mathrm{d}j_{1:T})\,\delta_{\mathbf{x}_{1:T}}(\mathrm{d}\mathbf{z}^{j_1}_1 \times \cdots \times \mathrm{d}\mathbf{z}^{j_T}_T)\left[\prod_{t=1}^T S^N_{t,D}(\mathbf{z}^{j_t}_t, \mathrm{d}\mathbf{z}^{-j_t}_t)\right] \\
&\quad \times \left[\prod_{t=2}^T \delta_{j_{t-1}}(\mathrm{d}a^{j_t}_{t-1}) \prod_{\substack{n=0 \\ n\neq j_t}}^N \bar{R}^N_{t,D}((\mathbf{z}_{t-1:t}, a_{t-1}), \mathrm{d}a^n_{t-1})\right] \\
&\quad \times \bar{R}^N_{T,D}((\mathbf{z}_{T-1:T}, a_{T-1}), \mathrm{d}k_T) \\
&\quad \times \begin{cases} \left[\prod_{t=1}^{T-1} \delta_{a_t^{k_{t+1}}}(\mathrm{d}k_t)\right] & \text{[without backward sampling]} \\ \left[\prod_{t=1}^{T-1} \bar{B}^N_{t,D}((\mathbf{z}_{t-1:t}, a_{t-1}, \mathbf{z}^{k_{t+1}}_{t+1}), \mathrm{d}k_t)\right] & \text{[with backward sampling]} \end{cases} \\
&\quad \times \delta_{(\mathbf{z}^{k_1}_1, \ldots, \mathbf{z}^{k_T}_T)}(\mathrm{d}\mathbf{x}'_{1:T}). \qquad (14)
\end{aligned}$$

Here, we have defined $\mathbf{z}^{-n}_t \coloneqq (\mathbf{z}^0_t, \ldots, \mathbf{z}^{n-1}_t, \mathbf{z}^{n+1}_t, \ldots, \mathbf{z}^N_t)$ as well as the following quantities.

- **Proposal kernels.** For any $t \in [T]$ and any $n \in [N]_0$, as in the RW-EHMM algorithm,

$$S^N_{t,D}(\mathbf{z}^n_t, \mathrm{d}\mathbf{z}^{-n}_t) \coloneqq \prod_{d=1}^D \mathrm{N}\big(\mathrm{d}z^{-n}_{t,d}; z^n_{t,d}\mathbf{1}_N, \tfrac{\ell_t}{D}\Sigma\big),$$

where $z_{t,d}^{-n} := (z_{t,d}^0, \ldots, z_{t,d}^{n-1}, z_{t,d}^{n+1}, \ldots, z_{t,d}^N)$.

- **Resampling kernels.** For any $n \in [N]_0$ and any $t \in [T]$,

$$\bar{R}_{t,D}^N((\mathbf{z}_{t-1:t}, a_{t-1}), \{n\}) := \Psi^n(\{\bar{\mathbf{w}}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m) - \bar{\mathbf{w}}_t(\mathbf{z}_{t-1}^{a_{t-1}^0}, \mathbf{z}_t^0)\}_{m=1}^N)$$
$$= \frac{\mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n)\mathbf{G}_t(\mathbf{z}_t^n)}{\sum_{m=0}^N \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m)\mathbf{G}_t(\mathbf{z}_t^m)},$$

When using the forced-move extension, we replace $\bar{R}_{T,D}^N((\mathbf{z}_{T-1:T}, a_{T-1}), \{k_T\})$ at time $t = T$ in (8) by

$$\begin{cases} \dfrac{\mathbf{h}^{k_T}}{\sum_{m=0}^N \mathbf{h}^m - \mathbf{h}^{k_T} \wedge \mathbf{h}^{j_T}}, & \text{if } k_T \neq j_T, \\ 1 - \displaystyle\sum_{\substack{l=0 \\ l \neq j_T}}^N \frac{\mathbf{h}^l}{\sum_{m=0}^N \mathbf{h}^m - \mathbf{h}^l \wedge \mathbf{h}^{j_T}}, & \text{if } k_T = j_T, \end{cases}$$

where we have defined the shorthand $\mathbf{h}^n := \mathbf{m}_T(\mathbf{z}_{T-1}^{a_{T-1}^n}, \mathbf{z}_T^n)\mathbf{G}_T(\mathbf{z}_T^n)$.

- **Backward kernels.** For $t \in [T-1]$,

$$\bar{B}_{t,D}^N((\mathbf{z}_{t-1:t}, a_{t-1}, \mathbf{z}_{t+1}^{k_{t+1}}), \{n\})$$
$$:= \Psi^n(\{\bar{\mathbf{v}}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}}) - \bar{\mathbf{v}}_t(\mathbf{z}_{t-1}^{a_{t-1}^0}, \mathbf{z}_t^0, \mathbf{z}_{t+1}^{k_{t+1}})\}_{m=1}^N)$$
$$= \frac{\mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n)\mathbf{G}_t(\mathbf{z}_t^n)\mathbf{m}_{t+1}(\mathbf{z}_t^n, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^N \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m)\mathbf{G}_t(\mathbf{z}_t^m)\mathbf{m}_{t+1}(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}})}.$$

From this definition, we can recover the joint law of all random variables $(\mathbf{Z}_{1:T}, A_{1:T-1}, K_{1:T}, \mathbf{X}'_{1:T})$ generated in Steps 1–4 of Algorithm 3 by conditioning on the event $\{J_1 = 0, \ldots, J_T = 0\}$, i.e.

$$\bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^N := \bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}(\,\cdot\,|J_1 = 0, \ldots, J_T = 0).$$

Let $\bar{\mathbb{E}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$ denote expectation w.r.t. $\bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$. In the remainder of this section, we will sometimes work with $\bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$ rather than with $\bar{\mathbb{P}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}$. This is justified because both versions of the i-RW-CSMC algorithm induce the same Markov kernel:

$$\bar{\mathbb{E}}_{T,D,\mathbf{x}_{1:T}}^{N,\star}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}] = \bar{\mathbb{E}}_{T,D,\mathbf{x}_{1:T}}^{N}[\mathbb{I}\{\mathbf{X}'_{1:T} \in \mathrm{d}\mathbf{x}'_{1:T}\}]$$
$$= \bar{\mathbf{P}}_{T,D}^N(\mathbf{x}_{1:T}, \mathrm{d}\mathbf{x}'_{1:T}).$$

D.2. Proof of Propositions 4.1. 4.2 and 4.3. In this section, we prove Propositions 4.1, 4.2 and 4.3 using the slightly generalised extended state-space construction defined above.

The proof of Proposition 4.3 proceeds along the same lines as the proof of Proposition 2.1.

PROOF (of Proposition 4.3). Recall that by (4), the random-walk type proposal used to scatter the particles around the reference path is symmetric in the sense that

$$\lambda(\mathrm{d}\mathbf{z}_t^j) S_{t,D}^N(\mathbf{z}_t^j, \mathrm{d}\mathbf{z}_t^{-j}) = \lambda(\mathrm{d}\mathbf{z}_t^k) S_{t,D}^N(\mathbf{z}_t^k, \mathrm{d}\mathbf{z}_t^{-k}),$$

for any $j, k \in [N]_0$, where $\mathbf{z}_t^{-n} := (\mathbf{z}_t^0, \ldots, \mathbf{z}_t^{n-1}, \mathbf{z}_t^{n+1}, \ldots, \mathbf{z}_t^N)$ and where λ denotes a suitable version of the Lebesgue measure.

For the plain algorithm (with neither the backward sampling nor the forced-move extension), we can then readily check that

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T})\bar{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{x}'_{1:T})\\ &\quad=\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})\bar{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}'_{1:T}}(\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{x}_{1:T}),\end{aligned}\tag{15}$$

i.e. (15) admits $\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})$ as a marginal.

For the backward-sampling extension, let $\widehat{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ be the same as $\bar{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ (without backward sampling) except that the terms $\delta_{j_{t-1}}(\mathrm{d}a^{j_t}_{t-1})$ in (14) are replaced by $\bar{B}^N_{t-1,D}((\mathbf{z}_{t-2:t-1},a_{t-2},\mathbf{z}^{k_t}_t),\{a^{j_t}_{t-1}\})$. Then

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}_{1:T})\bar{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}(\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{x}'_{1:T})\\ &\quad=\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})\widehat{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}'_{1:T}}(\mathrm{d}k_{1:T}\times\mathrm{d}\mathbf{z}_{1:T}\times\mathrm{d}a_{1:T-1}\times\mathrm{d}j_{1:T}\times\mathrm{d}\mathbf{x}_{1:T}).\end{aligned}\tag{16}$$

That is, (16) again admits $\boldsymbol{\pi}_{T,D}(\mathrm{d}\mathbf{x}'_{1:T})$ as a marginal. Incidentally, $\widehat{\mathbb{P}}^{N,\star}_{T,D,\mathbf{x}_{1:T}}$ can be recognised as the law of all the random variables generated by an i-RW-CSMC algorithm with ancestor sampling. This shows that ancestor sampling is a valid alternative to backward sampling in this algorithm.

Finally, use of the forced-move extension can again be justified as a partially collapsed Gibbs sampler because applying this extension in Step 2 of Algorithm 3 leaves the marginal distribution of K_T conditional on $(\mathbf{X}_{1:T},J_{1:T},\mathbf{Z}_{1:T},A_{1:T})$ invariant. □

PROOF (of Proposition 4.1). Let

$$\begin{aligned}&\Xi_T((\mathbf{z}_{1:T},j_{1:T}),\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T})\\ &\quad:=\left[\prod_{t=2}^{T}\delta_0(\mathrm{d}a^{j_t}_{t-1})\prod_{\substack{n=0\\n\neq j_t}}^{N}\bar{R}^N_{t-1,D}((\mathbf{z}_{t-2:t-1},a_{t-2}),\mathrm{d}a^n_{t-1})\right]\\ &\qquad\times\bar{R}^N_{T,D}((\mathbf{z}_{T-1:T},a_{T-1}),\mathrm{d}k_T)\prod_{t=1}^{T-1}\delta_{a^{k_{t+1}}_t}(\mathrm{d}k_t).\end{aligned}$$

Simple algebra then verifies that

$$\begin{aligned}&\xi_T(\mathbf{z}_{1:T},\mathrm{d}j_{1:T})\Xi_T((\mathbf{z}_{1:T},j_{1:T}),\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T})\\ &\quad=\xi_T(\mathbf{z}_{1:T},\mathrm{d}k_{1:T})\Xi_T((\mathbf{z}_{1:T},k_{1:T}),\mathrm{d}a_{1:T-1}\times\mathrm{d}j_{1:T}).\end{aligned}$$

This completes the proof. □

PROOF (of Proposition 4.2). Let

$$\begin{aligned}&\Xi_T((\mathbf{z}_{1:T},j_{1:T}),\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T})\\ &\quad:=\left[\prod_{t=2}^{T}\delta_0(\mathrm{d}a^{j_t}_{t-1})\prod_{\substack{n=0\\n\neq j_t}}^{N}\bar{R}^N_{t-1,D}((\mathbf{z}_{t-2:t-1},a_{t-2}),\mathrm{d}a^n_{t-1})\right]\\ &\qquad\times\bar{R}^N_{T,D}((\mathbf{z}_{T-1:T},a_{T-1}),\mathrm{d}k_T)\prod_{t=1}^{T-1}\bar{B}^N_{t,D}((\mathbf{z}_{t-1:t},a_{t-1},\mathbf{z}^{k_{t+1}}_{t+1}),\mathrm{d}k_t\}),\end{aligned}$$

and

$$\begin{aligned}&\widehat{\Xi}_T((\mathbf{z}_{1:T},j_{1:T}),\mathrm{d}a_{1:T-1}\times\mathrm{d}k_{1:T})\\ &\quad:=\left[\prod_{t=2}^{T}\bar{B}^N_{t-1,D}((\mathbf{z}_{t-2:t-1},a_{t-1},\mathbf{z}^{j_t}_t),\{a^{j_t}_{t-1}\})\prod_{\substack{n=0\\n\neq j_t}}^{N}\bar{R}^N_{t-1,D}((\mathbf{z}_{t-2:t-1},a_{t-2}),\mathrm{d}a^n_{t-1})\right]\\ &\qquad\times\bar{R}^N_{T,D}((\mathbf{z}_{T-1:T},a_{T-1}),\mathrm{d}k_T)\prod_{t=1}^{T-1}\delta_{a^{k_{t+1}}_t}(\mathrm{d}k_t).\end{aligned}$$

Simple algebra then verifies that

$$\begin{aligned}&\xi_T(\mathbf{z}_{1:T}, \mathrm{d}j_{1:T})\Xi_T((\mathbf{z}_{1:T}, j_{1:T}), \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T})\\ &\quad = \xi_T(\mathbf{z}_{1:T}, \mathrm{d}k_{1:T})\widehat{\Xi}_T((\mathbf{z}_{1:T}, k_{1:T}), \mathrm{d}a_{1:T-1} \times \mathrm{d}j_{1:T}).\end{aligned}$$

This completes the proof. □

D.3. Relationship with 'unconditional' SMC algorithms. In this section, we expand on the observation made in Remark 4.4 that whilst standard CSMC methods are closely related with a corresponding 'unconditional' SMC algorithm, no such 'unconditional' counterpart exists for the i-RW-CSMC algorithm.

As explained in Andrieu, Doucet and Holenstein (2010), standard CSMC methods are closely linked to the justification of a corresponding 'unconditional' SMC algorithm in the sense that the law of all the particles and parent indices generated by the latter, $\mathbb{Q}^{N,\star}_{T,D}$, is:

$$\begin{aligned}&\frac{\mathbb{E}[\mathbb{E}^{N,\star}_{T,D,\mathbf{X}_{1:T}}(\mathbb{I}\{(\mathbf{Z}_{1:T}, A_{1:T-1}) \in \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1}\})]}{\prod_{t=1}^{T} \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{G}_t(\mathbf{z}_t^n)}\\ &\quad \propto \prod_{n=0}^{N} \mathbf{M}_1(\mathrm{d}\mathbf{z}_1^n) \prod_{t=2}^{T}\left[\prod_{n=0}^{N} R^N_{t-1,D}(\mathbf{z}_{t-1}, \mathrm{d}a_{t-1}^n)\mathbf{M}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathrm{d}\mathbf{z}_t^n)\right]\\ &\quad =: \mathbb{Q}^{N,\star}_{T,D}(\mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1}),\end{aligned}$$

where $\mathbf{X}_{1:T} \sim \boldsymbol{\pi}_{T,D}$, and where – to avoid complications arising from the division by zero – we assume that $\mathbf{m}_t$ and $\mathbf{G}_t$ are strictly positive.

However, for the i-RW-CSMC algorithm, no such 'unconditional' SMC algorithm exists. To see this, let λ denotes a suitable version of the Lebesgue measure. Then by (4), the measure

$$\begin{aligned}&\frac{\mathbb{E}[\bar{\mathbb{E}}^{N,\star}_{T,D,\mathbf{X}_{1:T}}(\mathbb{I}\{(\mathbf{Z}_{1:T}, A_{1:T-1}) \in \mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1}\})]}{\prod_{t=1}^{T} \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{m}_t(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n)\mathbf{G}_t(\mathbf{z}_t^n)}\\ &\quad \propto \left[\prod_{t=1}^{T} \lambda(\mathrm{d}\mathbf{z}_t^0) S^N_{t,D}(\mathbf{z}_t^0, \mathrm{d}\mathbf{z}_t^{1:N})\right]\\ &\qquad \times \left[\prod_{t=2}^{T}\prod_{n=0}^{N} \bar{R}^N_{t-1,D}((\mathbf{z}_{t-2:t-1}, a_{t-2}), \mathrm{d}a_{t-1}^n)\right]\\ &\quad =: \bar{\mathbb{Q}}^{N,\star}_{T,D}(\mathrm{d}\mathbf{z}_{1:T} \times \mathrm{d}a_{1:T-1}),\end{aligned}$$

is not finite and hence there does not exist an algorithm that samples from it.

D.4. Formal definition of the limiting law. In this section, we give a more formal definition of the limiting law of the genealogies (i.e. of the ancestor indices, A_t^n) and of the particle indices of the new reference path, K_t, under the i-RW-CSMC algorithm that appears in Section 4.2.

$$\begin{aligned}&\bar{\mathbb{P}}^N_T(\mathrm{d}v_{1:T} \times \mathrm{d}w_{1:T} \times \mathrm{d}a_{1:T-1} \times \mathrm{d}k_{1:T})\\ &\quad := \left[\prod_{t=1}^{T} \mathrm{N}(\mathrm{d}[v_t, w_t]^{\mathrm{T}}; \bar{\mu}_{t|T}, \bar{\Sigma}_{t|T})\right]\\ &\qquad \times \left[\prod_{t=1}^{T-1} \delta_0(\mathrm{d}a_t^0) \prod_{n=1}^{N} \bar{R}^N_{t|T}((v_t, w_{t-1}, a_{t-1}), \mathrm{d}a_t^n)\right]\end{aligned}$$

$$\times \bar{R}^N_{T|T}((v_T, w_{T-1}, a_{T-1}), \mathrm{d}k_T)$$
$$\times \begin{cases} \left[\prod_{t=1}^{T-1} \delta_{a_t^{k_{t+1}}}(\mathrm{d}k_t)\right] & \text{[without backward sampling]} \\ \left[\prod_{t=1}^{T-1} \bar{B}^N_{t|T}((v_t, w_{t-1:t}, a_{t-1}), \mathrm{d}k_t)\right]. & \text{[with backward sampling]} \end{cases}$$

Here, we have defined the following quantities.

- **Asymptotic resampling kernels.** For any $n \in [N]_0$ and any $t \in [T]$,
$$\bar{R}^N_{t|T}((v_t, w_{t-1}, a_{t-1}), \{n\}) := \Psi^n(\{v_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N),$$
where we recall the convention that $w_{t-1}^0 \equiv 0$. As usual, when using the forced-move extension, we replace Ψ^n by Φ^n in the definition above at time $t = T$.
- **Asymptotic backward kernels.** For any $n \in [N]_0$ and any $t \in [T-1]$,
$$\bar{B}^N_{t|T}((v_t, w_{t-1:t}, a_{t-1}), \{n\}) := \Psi^n(\{v_t^m + w_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N).$$

D.5. Proof of Proposition 4.5. In this section, we prove Proposition 4.5. The proof can be viewed as an extension of the proof of Bédard, Douc and Moulines (2012, Lemma 10) to the case that $T > 1$. It relies on a Taylor-series expansion and a few technical lemmata which we state first.

LEMMA D.1. *For any $N \in \mathbb{N}$ and $n \in [N]_0$ Boltzmann selection function Ψ^n and Rosenbluth–Teller selection function Φ^n are Lipschitz-continuous.* ◁

PROOF. This can be verified by checking that the absolute value of the gradient is almost everywhere bounded. □

Main decomposition.. Throughout the remainder of this subsection, we fix some $N, T \in \mathbb{N}$. For any $\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}$ and any $t \in [T]$, define

$$\begin{aligned}
\mathcal{V}_{t|T} &:= \pi_T([\partial_t \bar{w}_t]^2), \\
\mathcal{W}_{t|T} &:= \pi_T([\partial_t \bar{w}_{t+1}]^2), \\
\mathcal{S}_{t|T} &:= \pi_T([\partial_t \bar{w}_t][\partial_t \bar{w}_{t+1}]), \\
\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t}) &:= D^{-1} \textstyle\sum_{d=1}^D \{[\partial_t \bar{w}_t](x_{t-1:t,d})\}^2, \\
\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1}) &:= D^{-1} \textstyle\sum_{d=1}^D \{[\partial_t \bar{w}_{t+1}](x_{t:t+1,d})\}^2, \\
\mathcal{S}_{t|T}(\mathbf{x}_{t-1:t+1}) &:= D^{-1} \textstyle\sum_{d=1}^D \{[\partial_t \bar{w}_t](x_{t-1:t,d})\}\{[\partial_t \bar{w}_{t+1}](x_{t:t+1,d})\}, \\
\widetilde{\mathcal{V}}_{t|T}(\mathbf{x}_{t-1:t}) &:= [D\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t})]^{-1/2} \sup_{d \in [D]} |[\partial_t \bar{w}_t](x_{t-1:t,d})|, \\
\widetilde{\mathcal{W}}_{t|T}(\mathbf{x}_{t:t+1}) &:= [D\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1})]^{-1/2} \sup_{d \in [D]} |[\partial_t \bar{w}_{t+1}](x_{t:t+1,d})|,
\end{aligned}$$

where we recall the convention that $\bar{w}_{T+1} \equiv 0$. With this notation, for some $0 \leq \eta < 1/4$, define the following family of Borel sets:

$$\mathbf{F}_{T,D} := \left\{ \mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \;\middle|\; \sup_{t \in [T]} \begin{bmatrix} |\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t}) - \mathcal{V}_{t|T}| \vee \\ |\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1}) - \mathcal{W}_{t|T}| \vee \\ |\mathcal{S}_{t|T}(\mathbf{x}_{t-1:t+1}) - \mathcal{S}_{t|T}| \vee \\ \widetilde{\mathcal{V}}_{t|T}(\mathbf{x}_{t-1:t}) \vee \widetilde{\mathcal{W}}_{t|T}(\mathbf{x}_{t:t+1}) \end{bmatrix} \leq D^{-\eta} \right\}. \tag{17}$$

LEMMA D.2. *For any* $T \in \mathbb{N}$, $\lim_{D\to\infty} \boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$. ◁

PROOF. The results

$$\begin{aligned}\boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid |\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t}) - \mathcal{V}_{t|T}| \le D^{-\eta}\}) \to 1,\\ \boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid |\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1}) - \mathcal{W}_{t|T}| \le D^{-\eta}\}) \to 1,\\ \boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid |\mathcal{S}_{t|T}(\mathbf{x}_{t-1:t+1}) - \mathcal{S}_{t|T}| \le D^{-\eta}\}) \to 1,\end{aligned} \tag{18}$$

follow from the law of the iterated logarithm since $\eta < 1/2$. To prove

$$\boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid \widetilde{\mathcal{V}}_{t|T}(\mathbf{x}_{t-1:t}) \le D^{-\eta}\}) \to 1,$$

we argue as in Bédard, Douc and Moulines (2012) that, by (18), it suffices to show that for any $c > 0$,

$$\begin{aligned}&\boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid \sup_{d\in[D]}|[\partial_t \bar{w}_t](x_{t-1:t,d})| \le cD^{1/2-\eta}\})\\ &\quad= \mathbb{P}(\{\sup_{d\in[D]}|[\partial_t \bar{w}_t](X_{t-1:t,d})| \le cD^{1/2-\eta}\})\\ &\quad= [1 - \mathbb{P}(\{|[\partial_t \bar{w}_t](X_{t-1:t,1})| > cD^{1/2-\eta}\})]^D\\ &\quad\to 1,\end{aligned}$$

where $(X_{1:T,d})_{d\ge1}$ be IID samples from π_T. But this holds since Markov's inequality along with the fact that $\eta < 1/4$ and **C1** ensures that

$$\mathbb{P}(\{|[\partial_t \bar{w}_t](X_{t-1:t,1})| > cD^{1/2-\eta}\}) \le \pi_T(|\partial_t \bar{w}_t|^4)c^{-1}D^{4(\eta-1/2)} = \mathrm{o}(D^{-1}).$$

The result $\boldsymbol{\pi}_{T,D}(\{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \mid \widetilde{\mathcal{W}}_{t|T}(\mathbf{x}_{t:t+1}) \le D^{-\eta}\}) \to 1$ follows by the same arguments. □

In the remainder of this subsection, we let $(\mathbf{x}_{1:T,D})_{D\ge1}$ be some sequence in $(\mathbf{E}_{T,D})_{D\ge1}$, i.e. $\mathbf{x}_{t,D} = x_{t,1:D,D} \in \mathbb{R}^D$, for any $D \ge 1$. We shall also often use the shorthand $\sup_{\mathbf{F}_{T,D}}$ for $\sup_{\mathbf{x}_{1:T,D}\in\mathbf{F}_{T,D}}$. We then set

$$\mathbf{Z}^n_{t,D} \coloneqq \begin{cases}\mathbf{x}_{t,D}, & \text{if } n = 0,\\ \mathbf{x}_{t,D} + \sqrt{\frac{\ell_t}{D}}\mathbf{U}^n_{t+1,D}, & \text{if } n \in [N],\end{cases}$$

where $\mathbf{U}^n_{t,D} \coloneqq U^n_{t,1:D,D}$, with $U^{1:N}_{t,d,D} \sim \mathrm{N}(\mathbf{0}_N, \Sigma)$ for $\Sigma \coloneqq \frac{1}{2}(\mathrm{I}_N + \mathbf{1}_N\mathbf{1}_N^{\mathrm{T}})$ and where $U^{1:N}_{t,d,D}$ and $U^{1:N}_{t,e,D}$ are independent whenever $s \ne t$ or $d \ne e$. We also fix some $a^n_t \in [N]_0$ for all $(t,n) \in [T-1]\times[N]$ and some $k_t \in [N]_0$ for all $t \in [T]$.

A second-order Taylor-series expansion then gives

$$\begin{aligned}&\bar{\mathbf{w}}_t(\mathbf{Z}^{a^n_{t-1}}_{t-1,D}, \mathbf{Z}^n_{t,D}) - \bar{\mathbf{w}}_t(\mathbf{Z}^0_{t-1,D}, \mathbf{Z}^0_{t,D})\\ &\quad= V^n_{t,D} + W^{a^n_{t-1}}_{t-1,D} + \textstyle\sum_{i=1}^4(R^{n,i}_{t,D} + S^{a^n_{t-1}}_{t-1,D}) + \sum_{i=1}^2 T^{a^n_{t-1},n,i}_{t-1,t,D},\end{aligned}$$

as well as (for $t < T$),

$$\begin{aligned}&\bar{\mathbf{v}}_t(\mathbf{Z}^{a^n_{t-1}}_{t-1,D}, \mathbf{Z}^n_{t,D}, \mathbf{Z}^{k_{t+1}}_{t+1,D}) - \bar{\mathbf{v}}_t(\mathbf{Z}^0_{t-1,D}, \mathbf{Z}^0_{t,D}, \mathbf{Z}^{k_{t+1}}_{t+1,D})\\ &\quad= V^n_{t,D} + W^{a^n_{t-1}}_{t-1,D} + \textstyle\sum_{i=1}^4(R^{n,i}_{t,D} + S^{a^n_{t-1}}_{t-1,D}) + \sum_{i=1}^2 T^{a^n_{t-1},n,i}_{t-1,t,D}\\ &\qquad+ W^n_{t,D} + \textstyle\sum_{i=1}^4 S^{n,i}_{t,D},\end{aligned}$$

where

$$V^n_{t,D} \coloneqq \sqrt{\frac{\mathcal{V}_{t|T}}{\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D})}}\sqrt{\frac{\ell_t}{D}}\sum_{d=1}^{D}[\partial_t \bar{w}_t](x_{t-1:t,d,D})U^n_{t,d,D} + \frac{\ell_t}{2}\pi_T(\partial_t^2 \bar{w}_t),$$

$$W^m_{t,D} \coloneqq \sqrt{\frac{\mathcal{W}_{t|T}}{\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D})}}\sqrt{\frac{\ell_t}{D}}\sum_{d=1}^{D}[\partial_t \bar{w}_{t+1}](x_{t:t+1,d,D})U^m_{t,d,D} + \frac{\ell_t}{2}\pi_T(\partial_t^2 \bar{w}_{t+1}),$$

and with

$$\begin{aligned}
R^{n,1}_{t,D} &\coloneqq \left\{1 - \sqrt{\frac{\mathcal{V}_{t|T}}{\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D})}}\right\}\sqrt{\tfrac{\ell_t}{D}}\textstyle\sum_{d=1}^{D}[\partial_t \bar{w}_t](x_{t-1:t,d,D})U^n_{t,d,D}\\
R^{n,2}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}[\partial_t^2 \bar{w}_t](x_{t-1:t,d,D}) - \tfrac{\ell_t}{2}\pi_T(\partial_t^2 \bar{w}_t)\\
R^{n,3}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}[\partial_t^2 \bar{w}_t](x_{t-1:t,d,D})\{(U^n_{t,d,D})^2 - 1\}\\
R^{n,4}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}\{[\partial_t^2 \bar{w}_t](x_{t-1:t,d,D} + \xi^n_{t,d,D}\sqrt{\ell_t}U^n_{t,d,D})\\
&\qquad\qquad - [\partial_t^2 \bar{w}_t](x_{t-1:t,d,D})\}(U^n_{t,d,D})^2\\
S^{m,1}_{t,D} &\coloneqq \left\{1 - \sqrt{\frac{\mathcal{W}_{t|T}}{\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D})}}\right\}\sqrt{\tfrac{\ell_t}{D}}\textstyle\sum_{d=1}^{D}[\partial_t \bar{w}_{t+1}](x_{t:t+1,d,D})U^m_{t,d,D}\\
S^{m,2}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}[\partial_t^2 \bar{w}_{t+1}](x_{t:t+1,d,D}) - \tfrac{\ell_t}{2}\pi_T(\partial_t^2 \bar{w}_{t+1})\\
S^{m,3}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}[\partial_t^2 \bar{w}_{t+1}](x_{t:t+1,d,D})\{(U^m_{t,d,D})^2 - 1\}\\
S^{m,4}_{t,D} &\coloneqq \tfrac{\ell_t}{2D}\textstyle\sum_{d=1}^{D}\{[\partial_t^2 \bar{w}_{t+1}](x_{t,d,D} + \eta^m_{t,d,D}\sqrt{\ell_t}U^m_{t,d,D}, x_{t+1,d,D})\\
&\qquad\qquad - [\partial_t^2 \bar{w}_{t+1}](x_{t:t+1,d,D})\}(U^m_{t,d,D})^2\\
T^{m,n,1}_{t,t+1,D} &\coloneqq \tfrac{\sqrt{\ell_t\ell_{t+1}}}{D}\textstyle\sum_{d=1}^{D}[\partial_t\partial_{t+1} \bar{w}_{t+1}](x_{t:t+1,d,D})U^m_{t,d,D}U^n_{t+1,d,D}\\
T^{m,n,2}_{t,t+1,D} &\coloneqq \tfrac{\sqrt{\ell_t\ell_{t+1}}}{D}\textstyle\sum_{d=1}^{D}\{[\partial_t\partial_{t+1} \bar{w}_{t+1}](x_{t:t+1,d,D} + \eta^m_{t,d,D}\sqrt{\ell_t}U^m_{t,d,D},\\
&\qquad\qquad x_{t+1,d,D} + \xi^n_{t+1,d,D}\sqrt{\ell_{t+1}}U^n_{t+1,d,D})\\
&\qquad - [\partial_t\partial_{t+1} \bar{w}_{t+1}](x_{t:t+1,d,D})\}U^m_{t,d,D}U^n_{t+1,d,D},
\end{aligned}$$

for some $\eta^m_{t,d,D}, \xi^n_{t,d,D} \in [0, D^{-1/2}]$, and with the usual convention that $V^n_{t,D}$, $W^n_{t,D}$, $R^{n,i}_{t,D}$, and $S^{n,i}_{t,D}$ are 0 if $t = 0$ or $n = 0$. Similarly $T^{m,n,i}_{t-1,t,D} = 0$ whenever $m = 0$, $n = 0$ or $t = 1$.

LEMMA D.3. *Assume **A1** and **C1**. For any $t \in [T]$, $(m, n) \in [N]^2$, $i \in [4]$ and $j \in [2]$,*

1. $\lim_{D\to\infty} \sup_{\mathbf{F}_{T,D}} \mathbb{E}[|R^{n,i}_{t,D}|] = 0$,
2. $\lim_{D\to\infty} \sup_{\mathbf{F}_{T,D}} \mathbb{E}[|S^{m,i}_{t,D}|] = 0$,
3. $\lim_{D\to\infty} \sup_{\mathbf{F}_{T,D}} \mathbb{E}[|T^{m,n,j}_{t,t+1,D}|] = 0$. ◁

PROOF. By definition of $\mathbf{F}_{T,D}$, using that $x \mapsto \sqrt{x}$ is concave and increasing so that $|\sqrt{x} - \sqrt{y}| \le \sqrt{|x - y|}$, and since by Jensen's inequality, $\mathbb{E}[|X|] \le \mathbb{E}[X^2]^{1/2}$,

$$\begin{aligned}
\mathbb{E}[|R^{n,1}_{t,D}|] \le \sqrt{\tfrac{|\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D}) - \mathcal{V}_{t|T}|}{\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D})}}\\
\times \sqrt{\tfrac{\ell_t}{D}\textstyle\sum_{d=1}^{D}\{[\partial_t \bar{w}_t](x_{t-1:t,d,D})\}^2\,\mathbb{E}[(U^n_{t,d,D})^2]}
\end{aligned}$$

$$\leq D^{-\eta/2}\ell_l^{1/2} \to 0,$$

$$\mathbb{E}[|S_{t,D}^{m,1}|] \leq \sqrt{\tfrac{|\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D}) - \mathcal{V}_{t|T}|}{\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D})}} \times \sqrt{\tfrac{\ell_t}{D}\textstyle\sum_{d=1}^{D}\{[\partial_t \bar{w}_{t+1}](x_{t:t+1,d,D})\}^2\,\mathbb{E}[(U_{t,d,D}^m)^2]}$$

$$\leq D^{-\eta/2}\ell_l^{1/2} \to 0.$$

From the definition of $\mathbf{F}_{T,D}$,

$$\mathbb{E}[|R_{t,D}^{n,2}|] \leq \tfrac{\ell_t}{2} D^{-\eta} \to 0,$$

$$\mathbb{E}[|S_{t,D}^{m,2}|] \leq \tfrac{\ell_t}{2} D^{-\eta} \to 0.$$

By Jensen's inequality, $\mathbb{E}[|X|] \leq \mathbb{E}[X^2]^{1/2}$ and hence

$$\mathbb{E}[|R_{t,D}^{n,3}|] \leq \sqrt{\tfrac{\ell_t^2}{4D^2}\textstyle\sum_{d=1}^{D}\{[\partial_t^2 \bar{w}_t](x_{t-1:t,d,D})\}^2\,\mathrm{var}[(U_{t,d,D}^n)^2]}$$

$$\leq \tfrac{\ell_t}{2D^{1/2}}\|\partial_t^2 \bar{w}_t\|_\infty \to 0,$$

$$\mathbb{E}[|S_{t,D}^{m,3}|] \leq \sqrt{\tfrac{\ell_t^2}{4D^2}\textstyle\sum_{d=1}^{D}\{[\partial_t^2 \bar{w}_{t+1}](x_{t:t+1,d,D})\}^2\,\mathrm{var}[(U_{t,d,D}^m)^2]}$$

$$\leq \tfrac{\ell_t}{2D^{1/2}}\|\partial_t^2 \bar{w}_{t+1}\|_\infty \to 0,$$

$$\mathbb{E}[|T_{t,t+1,D}^{m,n,1}|] \leq \sqrt{\tfrac{\ell_t \ell_{t+1}}{D^2}\textstyle\sum_{d=1}^{D}\{[\partial_t \partial_{t+1} \bar{w}_{t+1}](x_{t:t+1,d,D})\}^2\,\mathbb{E}[U_{t,d,D}^m U_{t+1,d,D}^n]} = 0.$$

Since $\eta_{t,d,D}^m, \xi_{t,d,D}^n \in [0, D^{-1/2}]$ and since $\partial_t^2 \bar{w}_t$, $\partial_t^2 \bar{w}_{t+1}$ and $\partial_t \partial_{t+1} \bar{w}_{t+1}$ are Lipschitz-continuous

$$\mathbb{E}[|R_{t,D}^{n,4}|] \leq \tfrac{\ell_t^{3/2}}{2\sqrt{D}}[\partial_t^2 \bar{w}_t]_{\mathrm{LIP}}\,\mathbb{E}[|U_{t,d,D}^n|^3] \to 0,$$

$$\mathbb{E}[|S_{t,D}^{m,4}|] \leq \tfrac{\ell_t^{3/2}}{2\sqrt{D}}[\partial_t^2 \bar{w}_{t+1}]_{\mathrm{LIP}}\,\mathbb{E}[|U_{t,d,D}^m|^3] \to 0,$$

$$\mathbb{E}[|T_{t,t+1,D}^{m,n,2}|] \leq \tfrac{\sqrt{\ell_t \ell_{t+1}}}{\sqrt{D}}[\partial_t \partial_{t+1} \bar{w}_{t+1}]_{\mathrm{LIP}}\,\mathbb{E}[|(\ell_t^{1/2} U_{t,d,D}^m + \ell_{t+1}^{1/2} U_{t+1,d,D}^n) U_{t,d,D}^m U_{t+1,d,D}^n|] \to 0,$$

where $[f]_{\mathrm{LIP}}$ denotes the Lipschitz constant of f w.r.t. the 1-norm. □

LEMMA D.4. *Assume **A1** and **C1** and let $(V_t^n)_{t\in[T],n\in[N]}$ and $(W_t^n)_{t\in[T-1],n\in[N]}$ be the families of Gaussian random variables defined in Section 4.2.1. Then, if $\mathbf{x}_{1:T,D} \in \mathbf{F}_{T,D}$, as $D \to \infty$,*

$$Y_D \coloneqq (V_{1,D}^{1:N}, \ldots, V_{T,D}^{1:N}, W_{1,D}^{1:N}, \ldots, W_{T-1,D}^{1:N})^{\mathrm{T}},$$

converges in distribution to

$$Y \coloneqq (V_1^{1:N}, \ldots, V_T^{1:N}, W_1^{1:N}, \ldots, W_{T-1}^{1:N})^{\mathrm{T}}.$$ ◁

PROOF. Since $\mathbb{E}[V_{t,D}^n] = \mathbb{E}[V_t^n]$ and $\mathbb{E}[W_{t,D}^m] = \mathbb{E}[W_t^m]$, we only need to show convergence in distribution of the centred random vector

$$\widetilde{Y}_D \coloneqq Y_D - \mathbb{E}[Y_D] = (\widetilde{V}_{1,D}^{1:N}, \ldots, \widetilde{V}_{T,D}^{1:N}, \widetilde{W}_{1,D}^{1:N}, \ldots, \widetilde{W}_{T-1,D}^{1:N})^{\mathrm{T}}$$

to

$$\widetilde{Y} \coloneqq Y - \mathbb{E}[Y] = (\widetilde{V}_1^{1:N}, \ldots, \widetilde{V}_T^{1:N}, \widetilde{W}_1^{1:N}, \ldots, \widetilde{W}_{T-1}^{1:N})^{\mathrm{T}}.$$

By the Cramér–Wold theorem, it then suffices to prove that $\lambda^{\mathrm{T}}\widetilde{Y}_D \to_{\mathrm{d}} \lambda^{\mathrm{T}}\widetilde{Y}$ for any $\lambda = (\lambda_1^{1:N}, \dots, \lambda_T^{1:N}, \bar{\lambda}_1^{1:N}, \dots, \bar{\lambda}_{T-1}^{1:N}) \in \mathbb{R}^{N(2T-1)}$. Equivalently, we must show that $\lambda^{\mathrm{T}}\widetilde{Y}_D \to_{\mathrm{d}} \mathrm{N}(0, \tau^2)$, where

$$\begin{aligned}\tau^2 &\coloneqq \mathrm{var}[\lambda^{\mathrm{T}}\widetilde{Y}] \\ &= \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_t^n \lambda_t^m \,\mathrm{cov}[V_t^n, V_t^m] + \bar{\lambda}_t^n \bar{\lambda}_t^m \,\mathrm{cov}[W_t^n, W_t^m] + 2\lambda_t^n \bar{\lambda}_t^m \,\mathrm{cov}[V_t^n, W_t^m],\end{aligned}$$

where we recall the convention that $W_T^n \equiv 0$.

Let $\mathcal{F}_{d,D} \coloneqq \sigma(\{U_{t,e,D}^n \mid t \in [T], e \in [d], n \in [N]\})$ as well as

$$\begin{aligned}\mathcal{U}_{d,D} \coloneqq \sum_{t=1}^{T}\Bigg[&\sqrt{\frac{\mathcal{V}_{t|T}}{\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D})}}\sqrt{\tfrac{\ell_t}{D}}[\partial_t \bar{w}_t](x_{t-1:t,d,D})\sum_{n=1}^{N}\lambda_t^n U_{t,d,D}^n \\ &+ \mathbb{I}\{t<T\}\sqrt{\frac{\mathcal{W}_{t|T}}{\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D})}}\sqrt{\tfrac{\ell_t}{D}}[\partial_t \bar{w}_{t+1}](x_{t:t+1,d,D})\sum_{m=1}^{N}\bar{\lambda}_t^m U_{t,d,D}^m\Bigg],\end{aligned}$$

then $\mathcal{U}_{d,D}$ is $\mathcal{F}_{d,D}$-measurable. Therefore, for $\mathbf{x}_{1:T,D} \in \mathbf{F}_{T,D}$, by the central limit theorem for triangular arrays (Dvoretzky, 1972), $\sum_{d=1}^{D}\mathcal{U}_{d,D} = \lambda^{\mathrm{T}}\widetilde{Y}_D$ converges in distribution to a zero-mean Gaussian random variable with variance τ^2 if, as $D \to \infty$,

1. $\sum_{d=1}^{D}\mathbb{E}[\mathcal{U}_{d,D}^2|\mathcal{F}_{d-1,D}] - \mathbb{E}[\mathcal{U}_{d,D}|\mathcal{F}_{d-1,D}]^2 \to_{\mathbb{P}} \tau^2$,
2. $\sum_{d=1}^{D}\sum_{d=1}^{D}\mathbb{E}[\mathcal{U}_{d,D}^2\,\mathbb{I}\{|\mathcal{U}_{d,D}| \geq \varepsilon\}|\mathcal{F}_{d-1,D}] \to 0$, for any $\varepsilon > 0$.

To verify the first condition, we note that

$$\begin{aligned}&\sum_{d=1}^{D}\mathbb{E}[\mathcal{U}_{d,D}^2|\mathcal{F}_{d-1,D}] - \mathbb{E}[\mathcal{U}_{d,D}|\mathcal{F}_{d-1,D}]^2 \\ &\qquad = \tau^2 + 2\sum_{t=1}^{T-1}\ell_t H_{t|T,D}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_t^n \bar{\lambda}_t^m \Sigma_{n,m},\end{aligned}$$

where

$$H_{t|T,D} \coloneqq \sqrt{\frac{\mathcal{V}_{t|T}\mathcal{W}_{t|T}}{\mathcal{V}_{t|T}(\mathbf{x}_{t-1:t,D})\mathcal{W}_{t|T}(\mathbf{x}_{t:t+1,D})}}\mathcal{S}_{t|T}(\mathbf{x}_{t-1:t+1,D}) - \mathcal{S}_{t|T},$$

so that $\lim_{D\to\infty}\sup_{\mathbf{F}_{T,D}}|H_{t|T,D}| = 0$ by definition of $\mathbf{F}_{T,D}$.

It remains to check the second condition. Let $\varepsilon > 0$ and set

$$\begin{aligned}a^{(1)} &\coloneqq \sup_{t\in[T]}\ell_t^2\mathcal{V}_{t|T} < \infty, \\ a^{(2)} &\coloneqq \sup_{t\in[T-1]}\ell_t^2\mathcal{W}_{t|T} < \infty, \\ a^{(3)} &\coloneqq 2\sup_{t\in[T-1]}\ell_t\ell_{t+1}\sqrt{\mathcal{V}_{t|T}\mathcal{W}_{t|T}} < \infty,\end{aligned}$$

and $a \coloneqq \sup_{i\in[3]}a^{(i)}$,

$$\begin{aligned}b_{t,d,D}^{(i)} &\coloneqq [\widetilde{\mathcal{V}}_{t|T}(\mathbf{x}_{t-1:t,D})]^2, \\ b_{t,d,D}^{(2)} &\coloneqq [\widetilde{\mathcal{W}}_{t|T}(\mathbf{x}_{t:t+1,D})]^2, \\ b_{t,d,D}^{(3)} &\coloneqq \widetilde{\mathcal{V}}_{t|T}(\mathbf{x}_{t-1:t,D})\widetilde{\mathcal{W}}_{t|T}(\mathbf{x}_{t:t+1,D}),\end{aligned}$$

as well as

$$\begin{aligned}
\mathcal{M}_{t,d}^{(1)} &\coloneqq \textstyle\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_t^n\lambda_t^m U_{t,d,D}^n U_{t,d,D}^m,\\
\mathcal{M}_{t,d}^{(2)} &\coloneqq \mathbb{I}\{t<T\}\textstyle\sum_{n=1}^{N}\sum_{m=1}^{N}\bar{\lambda}_t^n\bar{\lambda}_t^m U_{t,d,D}^n U_{t,d,D}^m,\\
\mathcal{M}_{t,d}^{(3)} &\coloneqq \mathbb{I}\{t<T\}\textstyle\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_t^n\bar{\lambda}_t^m U_{t,d,D}^n U_{t,d,D}^m.
\end{aligned}$$

Then, since $b_{t,d,D}^{(i)} \leq D^{-2\eta}$ for all $i \in [3]$ by definition of $\mathbf{F}_{T,D}$,

$$\begin{aligned}
\{|\mathcal{U}_{d,D}| \geq \varepsilon\} &\subseteq \{a\textstyle\sum_{i=1}^{3}\sum_{t=1}^{T} b_{t,d,D}^{(i)}|\mathcal{M}_{t,d}^{(i)}| \geq \varepsilon^2\}\\
&\subseteq \{a\textstyle\sum_{i=1}^{3}\sum_{t=1}^{T}|\mathcal{M}_{t,d}^{(i)}| \geq D^{2\eta}\varepsilon^2\},
\end{aligned}$$

and thus

$$\begin{aligned}
&\sum_{d=1}^{D}\mathbb{E}[\mathcal{U}_{d,D}^2\,\mathbb{I}\{|\mathcal{U}_{d,D}| \geq \varepsilon\}|\mathcal{F}_{d-1,D}]\\
&\qquad\leq \sum_{d=1}^{D}\mathbb{E}\Bigg[a\sum_{i=1}^{3}\sum_{t=1}^{T}|\mathcal{M}_{t,d}^{(i)}|\,\mathbb{I}\Bigg\{a\sum_{j=1}^{3}\sum_{s=1}^{T}|\mathcal{M}_{s,d}^{(j)}| \geq D^{2\eta}\varepsilon^2\Bigg\}\Bigg] \to 0.
\end{aligned}$$

This completes the proof. □

PROOF (of Proposition 4.5). We present the proof for the general case *with* backward sampling. This immediately implies the proof for the case *without* backward sampling. Likewise, we omit the proof in the case of the forced-move extension.

We fix some $N, T \in \mathbb{N}$ and define $\mathbf{F}_{T,D}$ as in (17). By Lemma D.2, we then have $\lim_{D\to\infty}\boldsymbol{\pi}_{T,D}(\mathbf{F}_{T,D}) = 1$. We also fix some $a_t^n \in [N]_0$ for all $(t,n) \in [T-1]\times[N]$ and some $k_t \in [N]_0$ for all $t \in [T]$.

The proof of the statement is then complete upon verifying that, as $D \to \infty$,

$$\begin{aligned}
&\sup_{\mathbf{F}_{T,D}}|\mathbb{E}[\Upsilon_D(\mathbf{Z}_D)] - \mathbb{E}[\Upsilon(Y)]|\\
&\qquad\leq \sup_{\mathbf{F}_{T,D}}|\mathbb{E}[\Upsilon_D(\mathbf{Z}_D)] - \mathbb{E}[\Upsilon(Y_D)]| + \sup_{\mathbf{F}_{T,D}}|\mathbb{E}[\Upsilon(Y_D)] - \mathbb{E}[\Upsilon(Y)]|\\
&\qquad\to 0,
\end{aligned} \tag{19}$$

where $\mathbf{Z}_D \coloneqq \mathbf{Z}_{1,D}^{1:N}, \ldots, \mathbf{Z}_{T,D}^{1:N}$ and

$$\begin{aligned}
&\Upsilon_D(\mathbf{Z}_D)\\
&\quad\coloneqq \textstyle\prod_{t=1}^{T-1}\prod_{n=1}^{N}\Psi^{a_t^n}(\{\bar{\mathbf{w}}_t(\mathbf{Z}_{t-1,D}^{a_{t-1}^m}, \mathbf{Z}_{t,D}^m) - \bar{\mathbf{w}}_t(\mathbf{Z}_{t-1,D}^0, \mathbf{Z}_{t,D}^0)\}_{m=1}^N)\\
&\qquad\times \Psi^{k_T}(\{\bar{\mathbf{w}}_T(\mathbf{Z}_{T-1,D}^{a_{T-1}^m}, \mathbf{Z}_{T-1,D}^m) - \bar{\mathbf{w}}_T(\mathbf{Z}_{T-1,D}^0, \mathbf{Z}_{T,D}^0)\}_{m=1}^N)\\
&\qquad\times \textstyle\prod_{t=1}^{T-1}\Psi^{k_t}(\{\bar{\mathbf{v}}_t(\mathbf{Z}_{t-1,D}^{a_{t-1}^m}, \mathbf{Z}_{t,D}^m, \mathbf{Z}_{t+1,D}^{k_{t+1}}) - \bar{\mathbf{v}}_t(\mathbf{Z}_{t-1,D}^0, \mathbf{Z}_{t,D}^0, \mathbf{Z}_{t+1,D}^{k_{t+1}})\}_{m=1}^N),
\end{aligned}$$

and where Y and Y_D are as in Lemma D.4 as well as

$$\begin{aligned}
&\Upsilon((v_1^{1:N}, \ldots, v_T^{1:N}, w_1^{1:N}, \ldots, w_{T-1}^{1:N})^{\mathrm{T}})\\
&\quad\coloneqq \textstyle\prod_{t=1}^{T-1}\prod_{n=1}^{N}\Psi^{a_t^n}(\{v_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N)\\
&\qquad\times \Psi^{k_T}(\{v_T^m + w_{T-1}^{a_{T-1}^m}\}_{m=1}^N)\textstyle\prod_{t=1}^{T-1}\Psi^{k_t}(\{v_t^m + w_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^N).
\end{aligned}$$

We now consider the two terms on the r.h.s. of (19). For the first term, a standard telescoping-sum decomposition, and using the fact that the selection functions are Lipschitz (see Lemma D.1) and bounded above by 1, gives

$$\begin{aligned}
&\mathbb{E}[\Upsilon_D(\mathbf{Z}_D)] - \mathbb{E}[\Upsilon(Y_D)]| \\
&\quad \leq \left[\sup_{m\in[N_0]}[\Psi^m]_{\mathrm{LIP}}\right] \\
&\qquad \times \Big[N\textstyle\sum_{t=1}^{T-1}\sum_{n=1}^{N}[\{\sum_{i=1}^{4}|R_{t,D}^{n,i}| + |S_{t-1,D}^{a_{t-1}^n,i}|\} + \sum_{j=1}^{2}|T_{t-1,t,D}^{a_{t-1}^n,n,j}|] \\
&\qquad\quad + \textstyle\sum_{n=1}^{N}[\{\sum_{i=1}^{4}|R_{T,D}^{n,i}| + |S_{T-1,D}^{a_{T-1}^n,i}|\} + \sum_{j=1}^{2}|T_{T-1,T,D}^{a_{t-1}^n,n,j}|] \\
&\qquad\quad + \textstyle\sum_{t=1}^{T-1}\sum_{n=1}^{N}[\{\sum_{i=1}^{4}|R_{t,D}^{n,i}| + |S_{t,D}^{n,i}| + |S_{t-1,D}^{a_{t-1}^n,i}|\} + \sum_{j=1}^{2}|T_{t-1,t,D}^{a_{t-1}^n,n,j}|]\Big],
\end{aligned}$$

so that $\lim_{D\to\infty}\sup_{\mathbf{F}_{T,D}}|\mathbb{E}[\Upsilon_D(\mathbf{Z}_D)] - \mathbb{E}[\Upsilon(Y_D)]| \to 0$, by Lemma D.3.

For the second term on the r.h.s. of (19), Lemma D.4 and the continuous mapping theorem ensure that $\Upsilon(Y_D) \to_{\mathrm{d}} \Upsilon(Y)$. Since $0 \leq \Upsilon \leq 1$, this implies $\lim_{D\to\infty}\sup_{\mathbf{F}_{T,D}}|\mathbb{E}[\Upsilon(Y_D)] - \mathbb{E}[\Upsilon(Y)]| \to 0$. □

D.6. Proof of Proposition 4.8. In this section, we prove Proposition 4.8. The proof relies on a few lemmata which we state first.

LEMMA D.5. *Let $\sigma_2 \geq \sigma_1 > 0$ and $(X_1, X_2) \sim \mathrm{N}(0_2, \mathrm{I}_2)$. Then*

$$\mathrm{e}^{\sigma_1 X_1 - \sigma_1^2/2} \leq_{\mathrm{cx}} \mathrm{e}^{\sigma_2 X_2 - \sigma_2^2/2}. \qquad \triangleleft$$

PROOF. Let $\varphi\colon \mathbb{R} \to (0,\infty)$ denote the probability density function of a standard normal distribution, let $\Phi\colon \mathbb{R} \to (0,1)$ denote the associated cumulative distribution function and $X \sim \mathrm{N}(0,1)$. Then for any $a > 0$, $b \in \mathbb{R}$ and $c \in \mathbb{R}$, writing $l(a,c) \coloneqq \log(c)/a + a/2$,

$$\begin{aligned}
\mathbb{E}[(\mathrm{e}^{aX+b} - c)_+] &= \int_{l(a,c)}^{\infty} (\mathrm{e}^{ax+b} - c)\varphi(x)\,\mathrm{d}x \\
&= \mathrm{e}^{a^2/2+b}\int_{l(a,c)}^{\infty} \varphi(x-a)\,\mathrm{d}x - c\Phi(-l(a,c)) \\
&= \mathrm{e}^{a^2/2+b}\Phi(-l(a,c)+a)\,\mathrm{d}x - c\Phi(-l(a,c)). \qquad (20)
\end{aligned}$$

Let $Y_i \coloneqq \mathrm{e}^{\sigma_i X_i - \sigma_i^2/2}$ for $i \in \{1,2\}$. Then by (20), for any $d \in \mathbb{R}$,

$$\mathbb{E}[(Y_2 - d)_+] - \mathbb{E}[(Y_1 - d)_+] = \Phi\Big(\frac{\sigma_2}{2} - \frac{\log(d)}{\sigma_2}\Big) - \Phi\Big(\frac{\sigma_1}{2} - \frac{\log(d)}{\sigma_1}\Big) \geq 0.$$

Finally, $\mathbb{E}[Y_1] = \mathbb{E}[Y_2]$ by the properties of the log-normal distribution. This completes the proof. □

LEMMA D.6. *Let $X \coloneqq (X_1,\ldots,X_N) \sim \mathrm{N}(\mu, \sigma^2\Sigma)$ and $Y \coloneqq (Y_1,\ldots,Y_N) \sim \mathrm{N}(\nu, \tau^2\Sigma)$, where $[\Sigma]_{i,i} = 1$ and $[\Sigma]_{i,j} = 1/2$ for $i \neq j$ and where $\mu \coloneqq -a\mathbf{1}_N$, $\nu \coloneqq -b\mathbf{1}_N$ for $a \in \mathbb{R}$ and $b \geq \tau^2/2$. Assume also that X and Y are independent. Then for any binary vector $\delta \coloneqq \delta_{1:N} \in \{0,1\}^N$,*

$$\mathbb{E}\left[\frac{\sum_{i=1}^{N}\mathrm{e}^{X_i+\delta_i Y_i}}{1+\sum_{i=1}^{N}\mathrm{e}^{X_i+\delta_i Y_i}}\right] \geq \left(1 + \frac{\mathrm{e}^{\sigma^2/2+a+\tau^2/2+b}}{N}\right)^{-1}.$$

In particular, for $\delta = (0,\ldots,0)$ we have the tighter bound

$$\mathbb{E}\left[\frac{\sum_{i=1}^{N}\mathrm{e}^{X_i}}{1+\sum_{i=1}^{N}\mathrm{e}^{X_i}}\right] \geq \left(1 + \frac{\mathrm{e}^{\sigma^2/2+a}}{N}\right)^{-1}. \qquad \triangleleft$$

PROOF. We begin by proving the bound in the special case $\delta = (0, \dots, 0)$:

$$\mathbb{E}\left[\frac{\sum_{i=1}^N \mathrm{e}^{X_i}}{1+\sum_{i=1}^N \mathrm{e}^{X_i}}\right] \geq \mathbb{E}\left[\frac{1}{1+\mathrm{e}^{-X_1}/N}\right]$$
$$\geq \frac{1}{1+\mathbb{E}[\mathrm{e}^{-X_1}/N]} = \left(1+\frac{\mathrm{e}^{\sigma^2/2+a}}{N}\right)^{-1},$$

where the first line follows since $t \mapsto t/(1+t)$ is concave and $\sum_{i=1}^N \mathrm{e}^{X_i} \leq_{\mathrm{cx}} N\mathrm{e}^{X_1}$; the second step is due to Jensen's inequality and the fact that $t \mapsto 1/(1+t)$ is convex; the last step follows from the properties of the log-normal distribution.

We now extend the approach to arbitrary $\delta \in \{0,1\}^N$. Since $\tau^2/2 - b \leq 0$,

$$\frac{\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i}}{1+\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i}} \geq \frac{\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i+(1-\delta_i)(\tau^2/2-b)}}{1+\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i+(1-\delta_i)(\tau^2/2-b)}}.$$

Furthermore, by Lemma D.5 and Dhaene et al. (2000, Theorem 5),

$$\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i+(1-\delta_i)(\tau^2/2-b)} \leq_{\mathrm{cx}} N\mathrm{e}^{X_1+Y_1},$$

and since $t \mapsto t/(1+t)$ is concave,

$$\mathbb{E}\left[\frac{\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i}}{1+\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i}}\right] \geq \mathbb{E}\left[\frac{\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i+(1-\delta_i)(\tau^2/2-b)}}{1+\sum_{i=1}^N \mathrm{e}^{X_i+\delta_i Y_i+(1-\delta_i)(\tau^2/2-b)}}\right]$$
$$\geq \mathbb{E}\left[\frac{\sum_{i=1}^N \mathrm{e}^{X_i+Y_i}}{1+\sum_{i=1}^N \mathrm{e}^{X_i+Y_i}}\right]$$
$$\geq \mathbb{E}\left[\frac{N\mathrm{e}^{X_1+Y_1}}{1+N\mathrm{e}^{X_1+Y_1}}\right]$$
$$\geq \frac{1}{1+\mathbb{E}[\mathrm{e}^{-X_1-Y_1}]/N}$$
$$= \left(1+\frac{\mathrm{e}^{\sigma^2/2+a+\tau^2/2+b}}{N}\right)^{-1}.$$

Here, the penultimate line again follows by Jensen's inequality since $t \mapsto 1/(1+t)$ is convex. □ □

LEMMA D.7. *Let $\pi(x_{1:T})$ denote some twice differentiable probability density function on $\mathbb{R}^T$ and write $\partial_t^i f(x_{1:T})$ as shorthand for $\frac{\partial^i}{\partial x_t^i} f(x_{1:T})$ with $\partial_t^1 =: \partial_t$. Then*

$$\pi([\partial_t \log \pi]^2) = -\pi(\partial_t^2 \log \pi). \qquad \triangleleft$$

PROOF. If $T = 1$, using integration by parts,

$$\pi([(\log \pi)']^2) = \int \pi'(x)(\log \pi)'(x)\,\mathrm{d}x$$
$$= \pi'(x)|_{-\infty}^{\infty} - \pi((\log \pi)'') = -\pi((\log \pi)''). \tag{21}$$

For $T > 1$, we let $\pi_{-t}(x_{-t}) := \int_{-\infty}^{\infty} \pi(x_{1:T})\,\mathrm{d}x_t$ denote the marginal density of $x_{-t} := (x_{1:t-1}, x_{t+1:T})$ and let $\pi_{t|-t}(x_t|x_{-t}) := \pi(x_{1:T})/\pi_{-t}(x_{-t})$. Since $\partial_t \log \pi(x_{1:T}) = \partial_t \log \pi_{t|-t}(x_t|x_{-t})$,

$$\begin{aligned}
\pi([\partial_t \log \pi]^2) &= \pi([\partial_t \log \pi_{t|-t}]^2) \\
&= \int \left[\int [\partial_t \log \pi_{t|-t}(x_t|x_{-t})]^2 \pi_{t|-t}(x_t|x_{-t})\,\mathrm{d}x_t \right] \pi_{-t}(x_{-t})\,\mathrm{d}x_{-t} \\
&= -\pi(\partial_t^2 \log \pi_{t|-t}) \\
&= -\pi(\partial_t^2 \log \pi),
\end{aligned}$$

where the third line follows by integration by parts in the same way as (21) with $\pi(x)$ replaced by the conditional distribution $\pi_{t|-t}(x_t|x_{-t})$. □

PROOF (of Proposition 4.8). By Lemma 1.1, it suffices to consider the case without forced move. Under Assumption **A4**, we have $W_t^n \equiv 0$ for any $t \in [T]$ and any $n \in [N]$, so that

$$\bar{R}_{t|T}^N((v_t, w_{t-1}, a_{t-1}), \{n\}) = \Psi^n(\{v_t^m\}_{m=1}^N),$$

does not depend on a_{t-1} (nor on w_{t-1}). As a consequence,

$$\bar{\alpha}_T^N(t) = \prod_{s=t}^{T} \sum_{n\in[N]} \bar{\mathbb{E}}_T^N[\Psi^n(\{V_s^m\}_{m=1}^N)] \geq \prod_{s=t}^{T} \left(1 + \frac{\exp(\ell_s \mathcal{I}_{s|T})}{N}\right)^{-1},$$

where the last line follows by Lemma D.6. This completes the proof of the first part of the proposition.

We now prove the lower bound in the case that backward sampling is employed. Since $W_t^n \equiv 0$ for any $t \in [T]$ and any $n \in [N]$ due to Assumption **A4**, we additionally have that

$$\bar{B}_{t|T}^N((v_t, w_{t:t-1}, a_{t-1}), \{n\}) = \Psi^n(\{v_t^m\}_{m=1}^N),$$

does not depend on a_{t-1} (nor on $w_{t:t-1}$). As a consequence,

$$\bar{\alpha}_T^N(t) = \sum_{n\in[N]} \bar{\mathbb{E}}_T^N[\Psi^n(\{V_t^l\}_{l=1}^N)] \geq \left(1 + \frac{\exp(\ell_t \mathcal{I}_{t|T})}{N}\right)^{-1},$$

where the last line follows by Lemma D.6. This completes the proof of the second part of the proposition. □

APPENDIX E: ADDITIONAL SIMULATION RESULTS

E.1. Effective samples sizes. Figure 5 displays the $\boldsymbol{\pi}_{T,D}$-averaged *effective sample size (ESS)* of the 'resampling' and 'backward-sampling' weights at time t for each algorithm in the setting from Section 5. More specifically, let

$$ESS(W^{0:N}) := \frac{1}{\sum_{n=0}^N (W^n)^2},$$

denote the ESS for self-normalised importance sampling weights $W_t^{0:N}$ (Kong, Liu and Wong, 1994). Below, let $\mathbb{E}$ denote expectation w.r.t. $\mathbf{X}_{1:T} \sim \boldsymbol{\pi}_{T,D}$.

1. The first column shows $\mathbb{E}\{\mathbb{E}_{T,D,\mathbf{X}_{1:T}}^N[ESS(W_t^{0:N})]\}$, where,

$$W_t^n := \begin{cases} \Psi^n(\{\mathbf{w}_t(\mathbf{Z}_t^m) - \mathbf{w}_t(\mathbf{Z}_t^0)\}_{m=1}^N), & \text{without backward sampling,} \\ \Psi^n(\{\mathbf{v}_t(\mathbf{Z}_t^m, \mathbf{Z}_{t+1}^{K_{t+1}}) - \mathbf{v}_t(\mathbf{Z}_t^0, \mathbf{Z}_{t+1}^{K_{t+1}})\}_{m=1}^N), & \text{with backward sampling.} \end{cases}$$

2. The second column shows $\mathbb{E}\{\bar{\mathbb{E}}^N_{T,D,\mathbf{X}_{1:T}}[ESS(\overline{W}_t^{0:N})]\}$, where

$$\overline{W}_t^n := \begin{cases} \Psi^n(\{\bar{\mathbf{w}}_t(\mathbf{Z}_{t-1}^{A_{t-1}^m}, \mathbf{Z}_t^m) - \bar{\mathbf{w}}_t(\mathbf{Z}_{t-1}^0, \mathbf{Z}_t^0)\}_{m=1}^N), & \text{without backward sampling,} \\ \Psi^n(\{\bar{\mathbf{v}}_t(\mathbf{Z}_{t-1}^{A_{t-1}^m}, \mathbf{Z}_t^m, \mathbf{Z}_{t+1}^{K_{t+1}}) - \bar{\mathbf{v}}_t(\mathbf{Z}_{t-1}^0, \mathbf{Z}_t^0, \mathbf{Z}_{t+1}^{K_{t+1}})\}_{m=1}^N), & \text{with backward sampling.} \end{cases}$$

By construction, the ESS takes values in $[1, N+1]$. The first column shows that for the i-CSMC algorithm, the ESS degenerates to its smallest possible value, 1, in high dimensions. In contrast, for the i-RW-CSMC algorithm, the ESS converges to a non-trivial limit > 1.

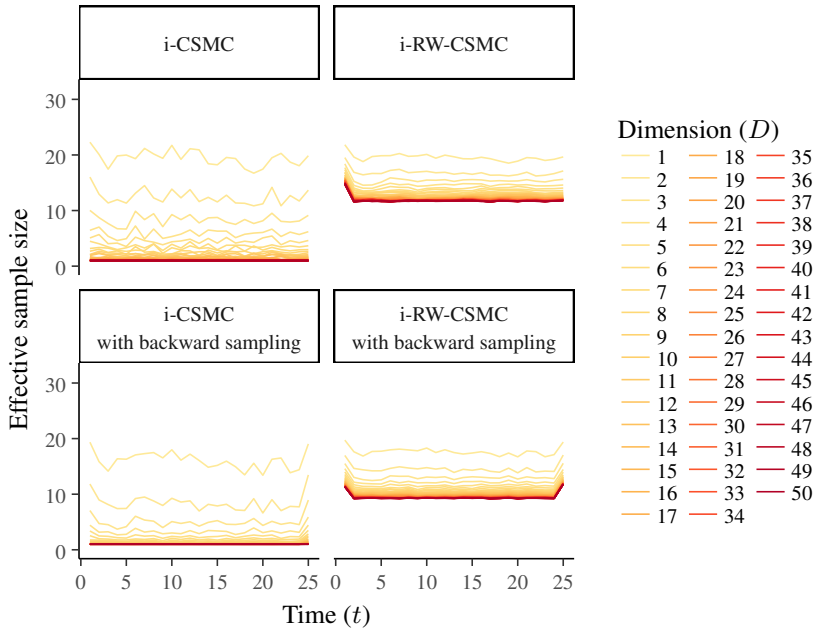


FIG. 5. The $\pi_{T,D}$-averaged effective sample sizes of the 'resampling weights' (top row) and 'backward-sampling weights' (bottom row) as a function of t.

E.2. Comparison with classical MCMC algorithms and choice of tuning parameters. Here, we compare the performance of our proposed methodology with classical MCMC algorithms that use the same Gaussian random-walk proposal kernel. For a fair comparison, the latter will be "multi-propsal" versions which make N proposals. Specifically, we compare the following four methods, where we recall that Φ^n denotes the Rosenbluth–Teller selection function which was defined in (2) and which reduces to the usual MH acceptance probability $\Phi^1 = 1 \wedge \exp$ if $N = 1$ proposals are used.

- **i-CSMC.** The standard i-CSMC algorithm, Algorithm 1 (with forced-move and backward-sampling extensions).
- **i-RW-CSMC.** The i-RW-CSMC algorithm from Algorithm 3 (with forced-move and backward-sampling extensions).
- **RWMH.** A random-walk Metropolis–Hastings (RWMH) algorithm on the full, i.e. $(T \times D)$-dimensional, space. For a fair comparison with the previous two algorithms, we implement a multi-proposal version of this method which proposes N new points – rather than just 1 – at each iteration. That is, the structure of the algorithm is that of Algorithm 3 using the forced-move extension in the case of $T = 1$. Algorithm 4 summarises the method,

where we use the convention that $\mathbf{z}_{1:T}^n \coloneqq (\mathbf{z}_1^n, \ldots, \mathbf{z}_T^n)$. Note that due to the $(T \times D)$-dimensional space, the variance of the proposal kernels is properly scaled as $\ell/(TD)$, for some $\ell > 0$. For $N = 1$, this algorithm reduces to a standard Gaussian random-walk algorithm (on the full space).

- **Blocked RWMH.** A blocked version of the above-mentioned multi-proposl RWMH algorithm. Each state $\mathbf{x}_t$ corresponds to a block. In this case, as in the i-RW-CSMC algorithm, the variance of the proposal kernels at time t is properly scaled as ℓ_t/D, for scale factors $\ell_1, \ldots, \ell_T > 0$. Algorithm 5 summarises the method. For $N = 1$, this algorithm reduces to a standard blocked Gaussian RWMH algorithm.

ALGORITHM 4 (RWMH). Given $\mathbf{x}_{1:T} \coloneqq \mathbf{x}_{1:T}[l] \in \mathbf{E}_{T,D}$.

1. Sample all particles $\mathbf{Z}_t^n = \mathbf{z}_t^n$ as in Step 1 of Algorithm 2, , where $\ell_1 = \ldots = \ell_T = \ell$, for some $\ell > 0$.
2. Sample $K = k \in [N]_0$ with probability $\Phi^k(\{\mathbf{h}^m\}_{m=1}^N)$, where
$$\mathbf{h}^m \coloneqq \log \boldsymbol{\pi}_{T,D}(\mathbf{z}_{1:T}^m) - \log \boldsymbol{\pi}_{T,D}(\mathbf{z}_{1:T}^0).$$
3. Set $\mathbf{X}'_{1:T} \coloneqq \mathbf{x}'_{1:T} \coloneqq \mathbf{z}_{1:T}^k$.
4. Return $\mathbf{x}_{1:T}[l+1] \coloneqq \mathbf{x}'_{1:T}$.

ALGORITHM 5 (Blocked RWMH). Given $\mathbf{x}_{1:T} \coloneqq \mathbf{x}_{1:T}[l] \in \mathbf{E}_{T,D}$.

1. Sample all particles $\mathbf{Z}_t^n = \mathbf{z}_t^n$ as in Step 1 of Algorithm 2.
2. For $t = 1, \ldots, T$,
   a) sample $K_t = k_t \in [N]_0$ with probability $\Phi^{k_t}(\{\mathbf{h}_t^m\}_{m=1}^N)$, where
$$\mathbf{h}_t^m \coloneqq \log \boldsymbol{\pi}_{T,D}(\mathbf{x}'_{1:t-1}, \mathbf{z}_t^m, \mathbf{x}_{t+1:T}) - \log \boldsymbol{\pi}_{T,D}(\mathbf{x}'_{1:t-1}, \mathbf{x}_{t:T}),$$
   b) set $\mathbf{X}'_t \coloneqq \mathbf{x}'_t \coloneqq \mathbf{z}_t^{k_t}$.
3. Return $\mathbf{x}_{1:T}[l+1] \coloneqq \mathbf{x}'_{1:T}$.

As part of the simulation study, we also investigate the choice of the tuning parameter N (used by all four above-mentioned algorithms) and the choice of the target acceptance rate $\alpha \in (0, 1)$ which is used to adaptively tune the scale factors ℓ_t in the i-RW-CSMC and blocked RWMH algorithms and the scale factor ℓ in the RWMH algorithm. Recall that as discussed in Section 6, we adapt the scale factors so that the acceptance rate is around α.

The model is the same as in Section 5. However, due to the substantial number of comparisons, we only consider $T = 5$ observations. The i-CSMC and i-RW-CSMC algorithms both employ the forced-move and backward-sampling extensions. We use 25 000 iterations for each algorithm in each configuration and results are averaged over four independent repetitions. Figure 6 compares the squared jumping distance (averaged over all components and time steps) for the different algorithms and configurations and illustrates the following.

1. The optimal acceptance rate appears to be around 0.2 if we use only $N = 1$ proposals but increases with N. This is in line with the results for a related multi-proposal MCMC algorithm (in case that $T = 1$) from Bédard, Douc and Moulines (2012).
2. The i-RW-CSMC algorithm performs better than the (multi-proposal) RWMH algorithm. This is not surprising because the former exploits the decorrelation in the "time"-direction whereas the latter does not.
3. As discussed the Section 7, backward sampling plays a similar rôle as blocking (in the "time"-direction). Hence, it is expected (and our simulations illustrate this) that both the

i-RW-CSMC algorithm and the blocked (multi-proposal) RWMH algorithm have a similar complexity. The blocked RWMH algorithm appears to even perform slightly better than the i-RW-CSMC algorithm. This may be due to the fact that the former uses the superior Rosenbluth–Teller selection function (2) at each time step whereas the latter only uses the Boltzmann selection function (1) (except in the final time step). However, note that the blocked RWMH algorithm requires manual selection of the block sizes (here: taken to be equal to a single state) whereas no such tuning is needed when using the i-RW-CSMC algorithm with backward sampling. Indeed backward sampling can be interpreted as automatically selecting suitable block sizes depending on the proposed set of particles.

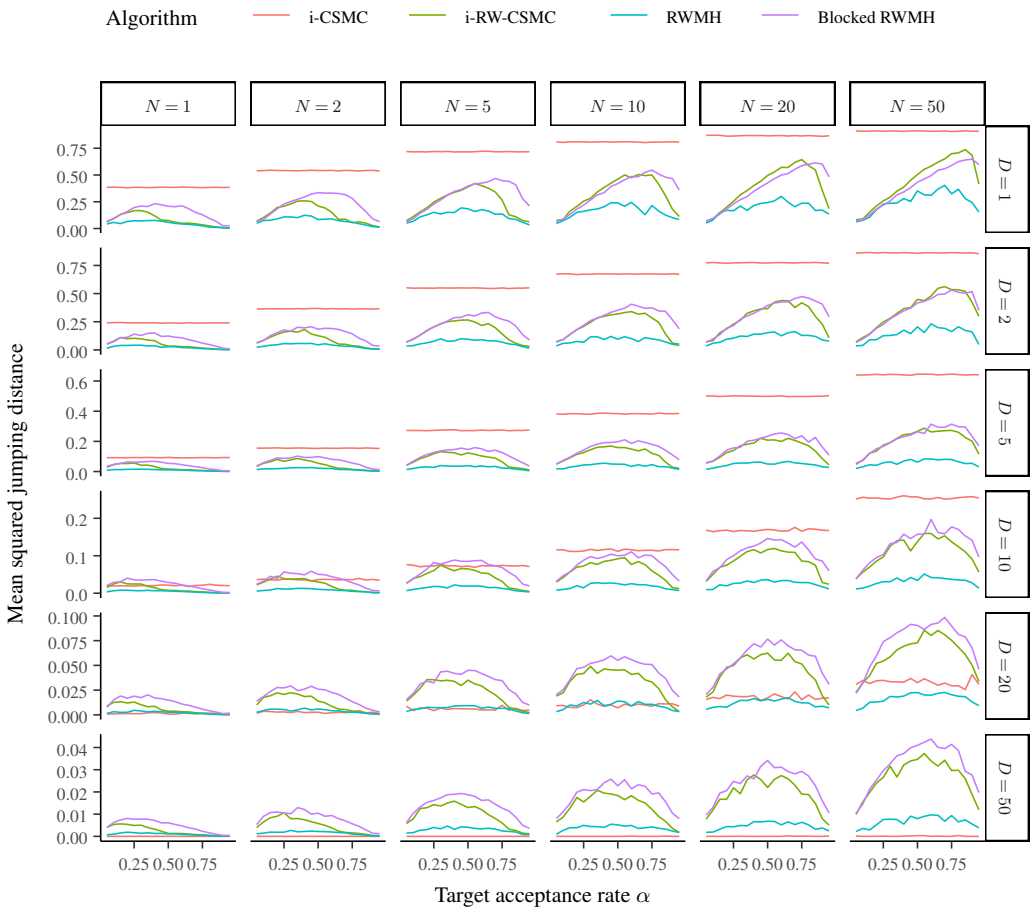


FIG. 6. Performance of the algorithms discussed in this section in different dimensions and for various choices of the tuning parameters N and α.

E.3. Multivariate stochastic volatility model.

E.3.1. *Model.* Our second example is a multivariate stochastic volatility model which was previously used as a benchmark in Guarniero, Johansen and Lee (2017). We stress that this model does not generally satisfy the IID assumption **A1**.

Let $\varphi_{m,C}$ denote a density (w.r.t. a suitable version of the Lebesgue measure, λ) of a D-dimensional normal distribution wit mean vector $m \in \mathbb{R}^D$ and covariance matrix $C \in \mathbb{R}^{D \times D}$.

Let $\mathbf{y}_t = (y_{t,d})_{d\in[D]} \in \mathbb{R}^D$ be a vector of D log-returns observed at time $t \in [T]$. Then:

$$\mathbf{G}_t(\mathbf{x}_t) \coloneqq \varphi_{\mathbf{0}_D,\mathrm{diag}(\exp(\mathbf{x}_t))}(\mathbf{y}_t),$$

$$\mathbf{m}_t(\mathbf{x}_{t-1}, \mathbf{x}_t) \coloneqq \varphi_{\mu+\Phi(\mathbf{x}_{t-1}-\mu),U}(\mathbf{x}_t),$$

where $\exp$ is applied element-wise, where $\mathrm{diag}(x)$ is a diagonal matrix with diagonal given by the vector x and where $\mu, \phi \in \mathbb{R}^D$, $\Phi \coloneqq \mathrm{diag}(\phi)$, $U \in \mathbb{R}^{D\times D}$ is some covariance matrix. We also recall that $\mathbf{0}_D, \mathbf{1}_D \in \mathbb{R}^D$ are column vectors filled with zeros and ones, respectively, and $\mathrm{I}_D \in \mathbb{R}^{D\times D}$ is an identity matrix. At time $t = 1$, $\mathbf{m}_1(\mathbf{x}_1) = \varphi_{\mu,U_*}(\mathbf{x}_1)$, where U_* is the stationary covariance matrix of the latent autoregressive process of log-volatilities, i.e.

$$\mathrm{vec}\, U_* = (\mathrm{I}_{D^2} - \Phi \otimes \Phi)^{-1} \mathrm{vec}\, U.$$

Finally, we assume $\mu = \nu \mathbf{1}_D$, $\Phi = \phi \mathrm{I}_D$ as well as $[U]_{i,i} = \tau$ and $[U]_{i,j} = \tau\rho$, for some $\nu \in \mathbb{R}$, $\tau > 0$, $\phi, \rho \in (-1, 1)$ and any $i, j \in [D]$ with $i \neq j$. Note that the IID assumption **A1** is violated unless $\rho = 0$.

E.3.2. *Illustration of the algorithms and adaptation of ℓ_t.* We compare the performance of the i-CSMC and i-RW-CSMC algorithms, with $N = 1000$ and $N = 50$ particles as well as $30\,000$ and $600\,000$ iterations, respectively, on a simulated data set generated using parameters $(\nu, \phi, \tau, \rho) = (0, 0.9, 2, 0.25)$ and for $T = 50$ and $D = 30$. Each algorithm is initialised by running a standard "unconditional" SMC algorithm (i.e. a so-called "bootstrap particle filter") with $N = 1000$ and $N = 50$ particles, respectively.

We use the adaptive rule for setting the scale factors ℓ_t suggested in Section 6 and with target acceptance rate as $\alpha = 1 - (N+1)^{-1/3} \approx 73\,\%$. To illustrate the utility of this adaptation rule, we initialise the scale factors to overly large values: $\ell_1 = \ldots = \ell_T = 100$

The results are shown in Figures 8 and 7 which illustrate that the i-RW-CSMC algorithm outperforms the i-CSMC algorithm in terms of average squared jumping distance and in terms of the average integrated autocorrelation time (where averages are taken over all "spatial" components), both of which are scaled to account for the fact that the i-CSMC algorithm uses a larger number of particles.

We stress that these metrics may overstate the performance of the i-CSMC algorithm because – in contrast to the i-RW-CSMC algorithm – it did not actually yield reliable estimates of any marginals under the joint smoothing distribution. For instance, at time $t = 1$, only 30 out of the $30\,000$ iterations resulted in acceptance.

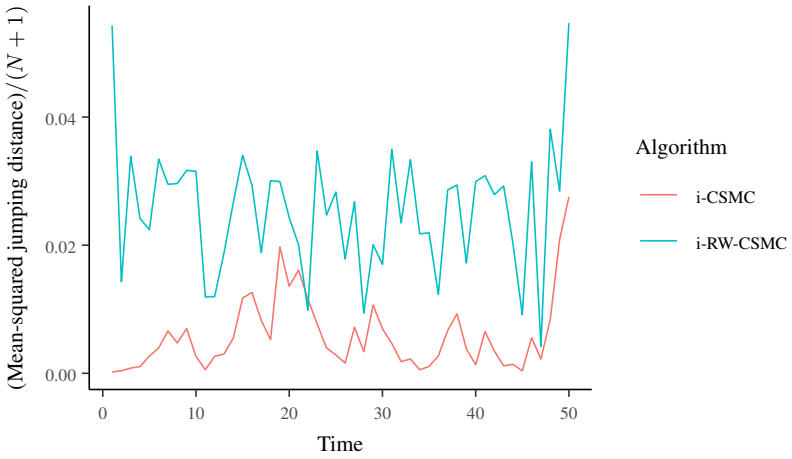


FIG. 7. Averaged (over 'spatial' components) squared jumping distance (adjusted for the number of particles) in the multivariate stochastic volatility model.

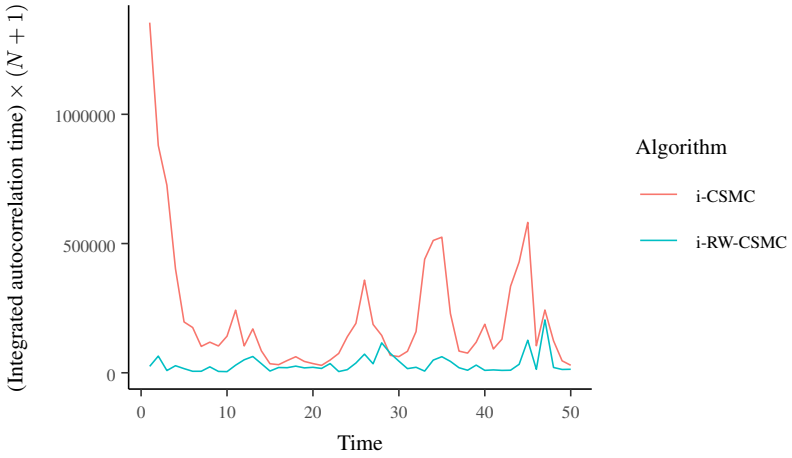


FIG. 8. Averaged (over 'spatial' components) integrated autocorrelation time (adjusted for the number of particles) in the multivariate stochastic volatility model.

Finally, we illustrate the adaptive rule for setting the scale factors ℓ_t suggested in Section 6. Figure 9 illustrates that the adaptive rule leads to a quick reduction in the scale factors down from the overly large initial values $\ell_1 = \ldots = \ell_T = 100$.

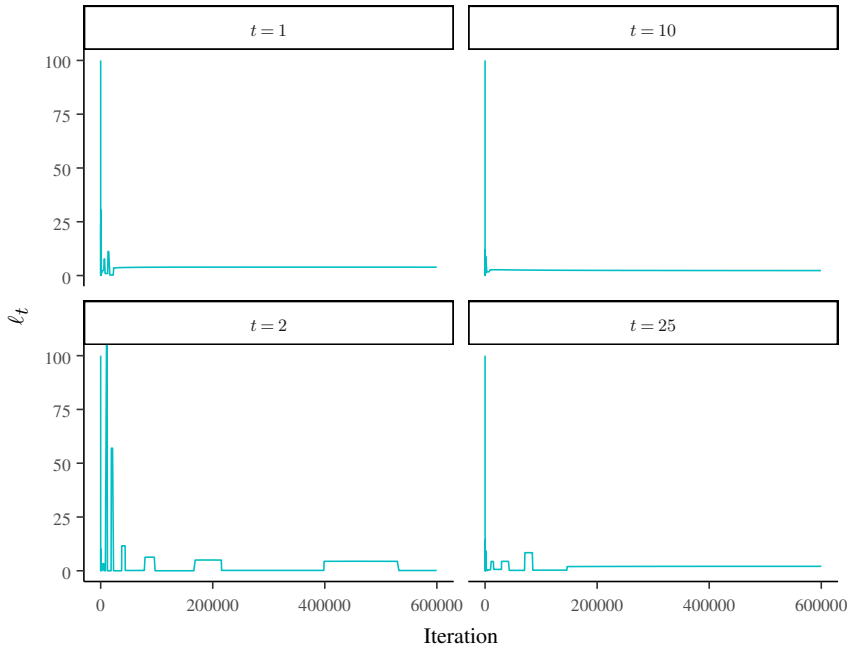


FIG. 9. Evolution of the scale factors ℓ_t under the adaptive scheme from Section 6.

APPENDIX F: USE FOR PARAMETER ESTIMATION

The Feynman–Kac model is typically specified through a set of parameters $\theta \in \Theta$, i.e. $\mathbf{M}_t = \mathbf{M}_{\theta,t}$, $\mathbf{m}_t = \mathbf{m}_{\theta,t}$, $\mathbf{G}_t = \mathbf{G}_{\theta,t}$ and $\boldsymbol{\pi}_{T,D} = \boldsymbol{\pi}_{\theta,T,D}$. In this case, we likewise write MCMC kernels induced by Algorithms 1, 2 and 3 as $\mathbf{P}^N_{T,D} = \mathbf{P}^N_{\theta,T,D}$, $\widetilde{\mathbf{P}}^N_{T,D} = \widetilde{\mathbf{P}}^N_{\theta,T,D}$, and $\bar{\mathbf{P}}^N_{T,D} = \bar{\mathbf{P}}^N_{\theta,T,D}$

If θ is unknown, then Bayesian inference in this model requires an MCMC algorithm that targets the *joint* posterior distribution of the parameters and the latent states which is proportional to $\varpi(\mathrm{d}\theta \times \mathrm{d}\mathbf{x}_{1:T}) \propto \mu(\mathrm{d}\theta)\boldsymbol{\pi}_{\theta,T,D}(\mathrm{d}\mathbf{x}_{1:T})$, where the probability measure μ on Θ is the prior distribution for θ.

In this section, we discuss two classes of MCMC algorithms which target this joint posterior distribution. The first includes the particle Gibbs sampler from Andrieu, Doucet and Holenstein (2010); the second includes a novel algorithm.

Particle Gibbs sampler. The first parameter-estimation algorithm is the *particle Gibbs sampler* proposed in Andrieu, Doucet and Holenstein (2010). Its $(l+1)$th iteration is given in Algorithm 6, where $R_{\mathbf{x}_{1:T}}(\theta, \mathrm{d}\vartheta)$ denotes some $\varpi(\mathrm{d}\theta|\mathbf{x}_{1:T})$-invariant MCMC kernel (e.g. often a convolution of multiple MH updates).

ALGORITHM 6 (particle Gibbs sampler). Given $(\theta[l], \mathbf{x}_{1:T}[l]) \in \Theta \times \mathbf{E}_{T,D}$,

1. sample $\theta[l+1] \sim R_{\mathbf{x}_{1:T}[l]}(\theta[l], \cdot)$,
2. sample $\mathbf{x}_{1:T}[l+1] \sim \mathbf{P}^N_{\theta[l+1],T,D}(\mathbf{x}_{1:T}[l], \cdot)$.

In Step 2 of the particle Gibbs sampler, it is straightforward to instead use the Markov kernel induced by Algorithm 2 or 3, i.e. $\widetilde{\mathbf{P}}^N_{\theta,T,D}$ or $\bar{\mathbf{P}}^N_{\theta,T,D}$. To see this, note that these kernels leave $\boldsymbol{\pi}_{\theta,T,D}(\mathrm{d}\mathbf{x}_{1:T}) = \varpi(\mathrm{d}\mathbf{x}_{1:T}|\theta)$ invariant.

Alternative algorithm. For the RW-EHMM and i-RW-CSMC algorithms, an alternative type of parameter-estimation method, in which the θ-updates make use of the information contained in *all* particles $\mathbf{Z}_t^n$, is possible.

The RW-EHMM-based algorithm is outlined in Algorithm 7, where $q_{\mathbf{z}_{1:T}}(\theta, \mathrm{d}\theta')$ is some proposal kernel for the parameters which may depend on the values of the particles. It can be viewed as a version of the parameter-estimation algorithms based around embedded HMM methods proposed in Shestopaloff and Neal (2013) who argued that averaging over multiple particles may allow for larger steps to be taken in the θ-direction compared to conditioning on a particular sequence of latent states.

ALGORITHM 7 (alternative RW-EHMM-based parameter estimation). Given $(\theta, \mathbf{x}_{1:T}) \coloneqq (\theta[l], \mathbf{x}_{1:T}[l]) \in \Theta \times \mathbf{E}_{T,D}$,

1. sample $\mathbf{Z}_{1:T} = \mathbf{z}_{1:T}$ via Step 1 of Algorithm 2,
2. sample $\Theta' = \theta' \sim q_{\mathbf{z}_{1:T}}(\theta, \cdot)$ and set
$$r \coloneqq \frac{q_{\mathbf{z}_{1:T}}(\theta', \theta)\mu(\theta') \sum_{n_{1:T} \in [N]_0^T} \boldsymbol{\pi}_{\theta',T,D}(\mathbf{z}_1^{n_1}, \ldots, \mathbf{z}_T^{n_T})}{q_{\mathbf{z}_{1:T}}(\theta, \theta')\mu(\theta) \sum_{n_{1:T} \in [N]_0^T} \boldsymbol{\pi}_{\theta,T,D}(\mathbf{z}_1^{n_1}, \ldots, \mathbf{z}_T^{n_T})},$$
3. sample $U = u \sim \mathrm{Unif}_{[0,1]}$,
4. if $u \leq r$,
   - sample $K_{1:T} = k_{1:T} \sim \xi_{\theta',T}(\mathbf{z}_{1:T}, \cdot)$,
   - return $(\theta[l+1], \mathbf{x}_{1:T}[l+1]) \coloneqq (\theta', (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T}))$;

   else,
   - sample $K_{1:T} = k_{1:T} \sim \xi_{\theta,T}(\mathbf{z}_{1:T}, \cdot)$,
   - return $(\theta[l+1], \mathbf{x}_{1:T}[l+1]) \coloneqq (\theta, (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T}))$.

Since Algorithm 7 relies on the RW-EHMM scheme, its computational cost again grows quadratically in N. This motivates us to propose Algorithm 8 which only requires $\mathrm{O}(N)$ operations. To our knowledge, Algorithm 8 is novel. For simplicity, we only state the version of the algorithm with the backward-sampling but without the forced-move extension. Here, $q_{\mathbf{z}_{1:T}, a_{1:T-1}}(\theta, \mathrm{d}\theta')$ is some proposal kernel for the parameters which may depend on the values of the particles and ancestor indices. Likewise, we have used the following notation for the probability of resampling the nth particle at time t which was already introduced in Appendix D.1:

$$\bar{R}_{\theta,t,D}^N((\mathbf{z}_{t-1:t}, a_{t-1}), \{n\}) \coloneqq \frac{\mathbf{m}_{\theta,t}(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n)\mathbf{G}_{\theta,t}(\mathbf{z}_t^n)}{\sum_{m=0}^N \mathbf{m}_{\theta,t}(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m)\mathbf{G}_{\theta,t}(\mathbf{z}_t^m)},$$

and for the probability of selecting the nth particle at time t via backward sampling which likewise was already introduced in Appendix D.1:

$$\bar{B}_{\theta,t,D}^N((\mathbf{z}_{t-1:t}, a_{t-1}, \mathbf{z}_{t+1}^{k_{t+1}}), \{n\}) \coloneqq \frac{\mathbf{m}_{\theta,t}(\mathbf{z}_{t-1}^{a_{t-1}^n}, \mathbf{z}_t^n)\mathbf{G}_{\theta,t}(\mathbf{z}_t^n)\mathbf{m}_{\theta,t+1}(\mathbf{z}_t^n, \mathbf{z}_{t+1}^{k_{t+1}})}{\sum_{m=0}^N \mathbf{m}_{\theta,t}(\mathbf{z}_{t-1}^{a_{t-1}^m}, \mathbf{z}_t^m)\mathbf{G}_{\theta,t}(\mathbf{z}_t^m)\mathbf{m}_{\theta,t+1}(\mathbf{z}_t^m, \mathbf{z}_{t+1}^{k_{t+1}})}.$$

In addition, $a_t' \coloneqq a_t'^{0:N} \in [N]_0^{N+1}$ denote values of a second set of proposed time-t ancestor indices $A_t' \coloneqq A_t'^{0:N}$.

ALGORITHM 8 (alternative i-RW-CSMC-based parameter estimation). Given $(\theta, \mathbf{x}_{1:T}) \coloneqq (\theta[l], \mathbf{x}_{1:T}[l]) \in \Theta \times \mathbf{E}_{T,D}$,

1. sample $(\mathbf{Z}_{1:T}, A_{1:T-1}) = (\mathbf{z}_{1:T}, a_{1:T-1})$ via Step 1 of Algorithm 3,
2. sample
   - $\Theta' = \theta' \sim q_{\mathbf{z}_{1:T}, a_{1:T-1}}(\theta, \cdot)$;
   - $A'_{1:T-1} = a'_{1:T-1} \sim \prod_{t=1}^{T-1} \prod_{n=0}^{N} \bar{R}^N_{\theta';t,D}((\mathbf{z}_{t-1:t}, a'_{t-1}), \{a'^n_t\})$,

   and set
$$r \coloneqq \frac{q_{\mathbf{z}_{1:T}, a'_{1:T-1}}(\theta', \theta)\mu(\theta') \prod_{t=1}^{T} \sum_{n=0}^{N} \mathbf{m}_{\theta';t}(\mathbf{z}_{t-1}^{a'^n_{t-1}}, \mathbf{z}_t^n)\mathbf{G}_{\theta';t}(\mathbf{z}_t^n)}{q_{\mathbf{z}_{1:T}, a_{1:T-1}}(\theta, \theta')\mu(\theta) \prod_{t=1}^{T} \sum_{n=0}^{N} \mathbf{m}_{\theta,t}(\mathbf{z}_{t-1}^{a^n_{t-1}}, \mathbf{z}_t^n)\mathbf{G}_{\theta,t}(\mathbf{z}_t^n)},$$
3. sample $U = u \sim \mathrm{Unif}_{[0,1]}$,
4. if $u \leq r$,
   - sample $K_T = k_T \sim \bar{R}^N_{\theta';T,D}((\mathbf{z}_{T-1:T}, a'_{T-1}), \cdot)$,
   - for $t = T-1, \ldots, 1$, sample $K_t = k_t \sim \bar{B}^N_{\theta';t,D}((\mathbf{z}_{t-1:t}, a'_{t-1}, \mathbf{z}_{t+1}^{k_{t+1}}), \cdot)$;
   - return $(\theta[l+1], \mathbf{x}_{1:T}[l+1]) \coloneqq (\theta', (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T}))$;

   else,
   - sample $K_T = k_T \sim \bar{R}^N_{\theta,T,D}((\mathbf{z}_{T-1:T}, a_{T-1}), \cdot)$,
   - for $t = T-1, \ldots, 1$, sample $K_t = k_t \sim \bar{B}^N_{\theta,t,D}((\mathbf{z}_{t-1:t}, a_{t-1}, \mathbf{z}_{t+1}^{k_{t+1}}), \cdot)$;
   - return $(\theta[l+1], \mathbf{x}_{1:T}[l+1]) \coloneqq (\theta, (\mathbf{z}_1^{k_1}, \ldots, \mathbf{z}_T^{k_T}))$.

Algorithm 8 could potentially be improved by employing (conditional) systematic rather than multinomial resampling. In this case, the ancestor indices in both the numerator and denominator can be drawn based on the same uniform random number at each time step.