Dear Enno Mammen,

Many thanks for the feedback regarding our paper titled “Conditional sequential Monte Carlo in high dimensions”. We thank the Associate Editor and both reviewers for the kind and helpful suggestions. We have adopted almost all of these in the revised version. Specifically, our manuscript now

1. demonstrates the performance of our method on a realistic numerical example;
2. provides additional intuition for the validity of our framework;
3. gives guidance on how to select the tuning parameters of our algorithm in practice.

If you have any further queries please do not hesitate to get in contact with us.

Yours sincerely,

Axel Finke, Alexandre H. Thiery
To the associate editor and reviewers: We greatly appreciate your invaluable comments for our paper. Below are our point-by-point responses to your comments. The colour coding is as follows:

- **magenta** indicates your comments;
- **black** indicates our responses;
- **green** indicates our main changes in the manuscript.

In addition to our revised manuscript, we are also attaching a “\TeX diff”-version which highlights all changes relative to our original submission.

**Associate Editor:**

This is a very nice paper, which brings fresh insight on a novel, elaborate PMCMC scheme; namely Particle Gibbs. However, like both referees, it took me time to understand properly its scientific message (which is quite subtle, actually). Thus, I would recommend the authors to pay particular attention to the recommendations of the referees, especially those regarding helping the readers get a better intuition of what some results mean and so on. (To be clear, all the recommendations of the referees look reasonable and worth being carefully addressed.)

Thank you for the kind words.

We have followed this advice and we have made two changes that should help the readers get better intuition for the validity of the methods.
1. As suggested by Reviewer 1, we have added a paragraph in Section 3 which explains how the RW-EHMM algorithm can be understood as the combination of two steps which each sample from full conditional distributions under a suitable extended target distribution that admits the desired target distribution $\pi_{T,D}$ as a marginal.

2. We have added a new appendix, Appendix A.1, which expands on the idea from Neal (2003); Neal et al. (2004) that the particles $(Z^{n}_t)_{t \leq T, n \leq N}$ “discretise” the space so that the problem of selecting a new reference path $X'_1:T = (Z^{K_1}_1, \ldots, Z^{K_T}_T)$ is nothing other than the problem of sampling from the joint posterior distribution of the “latent states” $K_{1:T}$ in a finite (more precisely: $(N + 1)$-state) hidden Markov model (HMM). With this in mind, conditional on the values of all the particles,

   a) the RW-EHMM algorithm performs the usual forward filtering–backward sampling recursions to sample from this joint posterior distribution (at a cost that is quadratic in $N$);

   b) the i-RW-CSMC algorithm reduces the cost to being linear in $N$ by instead running a (slightly non-standard) particle filter (potentially with backward sampling) to sample from this joint posterior distribution.

The latter interpretation, too, was kindly suggested by Reviewer 1.

We state the exact changes in the detailed response to Reviewer 1 below.

Also, I would strongly recommend adding a more realistic numerical example. As pointed by Ref 2, the current example is very toy. It is important to see the proposed method “in action”.

In Appendix E.3, we have now added simulation results for a multivariate stochastic volatility model which has served as a benchmark for SMC-type methods in high dimensions, e.g. in Guarniero et al. (2015). In particular, the joint smoothing distribution in this example is not analytically tractable, i.e. one really must resort to Monte Carlo methods such as the one we propose.

In particular, one potential concern is that it might be difficult to tune the $\ell_t$ coefficients to get good performance (relative to particle Gibbs).

We agree that the original manuscript should have included more details to guide practitioners. As suggested by Reviewer 1, we have added Section 6 to the manuscript which discusses various aspects of the practical implementation of the RW-EHMM and i-RW-CSMC algorithms, e.g.
1. how to initialise the MCMC chain;
2. how to choose the number of particles;
3. how to set the scale factors $\ell_t$.

In particular, we are now suggesting a simple scheme which adapts the scale factors $\ell_t$ so that a pre-specified target acceptance rate is maintained at each time step. We also provide new simulations in Appendix E.2 and E.3 which show this scheme “in action” and give some guidance on how to specify the target acceptance rate as a function of $N$. We state the exact changes in the detailed response to Reviewers 1 and 2 below.

1 Reviewer 1:

1.1 General comments

The proposed methods are interesting as they open up new ways of making proposals to approximate the smoothing distribution. The accompanying mathematical analysis is relevant as it explores the scenario in which the proposed algorithms would be most beneficial, namely in models with high dimensional hidden states.

Thank you for the kind comments.

1.2 On the exposition of Proposition 3.1 and Proposition 4.1

I personally find that the exposition of Proposition 3.1 and Proposition 4.1 lacks some intuition. While proofs are given in the Appendix, they are unnatural and, as the authors admitted, rather included for “completeness”. I do understand that the true motivations are already explained in Finke et al. (2016), among others. Still, I think that it is important to succinctly summarise at least some of their arguments in the current manuscript. As the authors put it, their Section 3 should be written for “didactic reasons because it is simple to understand”.

The formal proofs in the Appendix can be left as they are, but the reader would greatly appreciate some short phrases to explain the rationale behind the correctness of Markov kernels in Proposition 3.1 and Proposition 4.1.
4.1. For instance, Algorithm 2 is actually derived from a Gibbs sampler for the mixture distribution defined by

\[
\hat{\pi}(z_{1:T}^0) := \frac{1}{(N+1)^T} \sum_{k_1, \ldots, k_T} \pi(z_{1:T}^{k_1}, \ldots, z_{1:T}^{k_T}) \prod_t q_t(z_{-k} | z_{kt})
\]

where \(q_t(z_{-k} | z_{kt})\) is any proposal kernel such that \(q_t(z_{-k} | z_{kt}) = q_t(z_{-\ell} | z_{\ell t})\), for \(k \neq \ell\). For Proposition 4.1, some intuition (albeit not the whole proof) can be gained by thinking of sampling from \(\xi_T(z_{1:T}, d_{k_1:T})\) via an \((N+1)\)-particle i-CSMC algorithm where at each time step \(t\) the Markov proposal is simply the uniform law on \(\{0, 1, 2, \ldots, N\}\).

Thank you for pointing out that readers could use more intuition as to why the proposed Markov kernels leave the joint smoothing distribution invariant. We have been working with these extended state-space constructions for such a long time that we took some concepts for granted when we shouldn’t have.

To provide the missing intuition for the validity of the RW-EHMM algorithm, we have now added the following text below Proposition 3.1, where

\[
S_{t,D}^N(z_t^n, dz_t^{-n}) := \prod_{d=1}^D N(dz_{t,d}^{-n}; 1_N z_{t,d}^{-n}, \frac{\ell_t}{2} \Sigma),
\]

is the particle proposal kernel induced by Step 1 of Algorithm 2 (this is now defined in the text on Page 11 whereas it was previously only defined in the appendix):

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

\[
\hat{\pi}_{T,D}(dx_{1:T} \times dk_{1:T} \times dz_{1:T}) = \pi_{T,D}(dx_{1:T}) \text{Unif}_{[N]}(dk_{1:T}) \prod_{t=1}^T \delta_{x_t}(z_t^{k_t}) S_{t,D}^N(z_t^{k_t}; dz_t^{-k_t}),
\]

where Unif\(_A\) is the uniform distribution on a set \(A\). Step 1 of Algorithm 2 then samples from \(\hat{\pi}_{T,D}(dz_{1:T}|x_{1:T}, k_{1:T})\) while Steps 2 and 3 jointly sample from \(\hat{\pi}_{T,D}(dx_{1:T} \times dk_{1:T}|z_{1:T})\).

Thank you also for pointing out that the Markov kernels for sampling \(K_{1:T} = k_{1:T}\) as part of the i-RW-CSMC algorithm discussed in Propositions 4.1 and 4.2 can be viewed as
(slightly non-standard\textsuperscript{1}) conditional SMC kernels (with or without backward sampling) targeting a finite-state HMM.

We have now incorporated this idea into a new appendix, A.1, which provides further intuition for the recursions used for sampling the new set of particle indices $K_{1:T} = k_{1:T}$ within both the RW-EHMM and the i-RW-CSMC algorithm. We have also included pointers to this appendix in Section 3.1.2 (i.e. just before Proposition 3.1) and below Propositions 4.1 and 4.2.

\section*{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 $X'_{1:T} = (Z_{K_1}^1, \ldots, Z_{K_T}^T)$ within the RW-EHMM and i-RW-CSMC algorithms (Algorithms 2 and 3). Our discussion is based around ideas from Neal (2003); Neal et al. (2004); Finke et al. (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:

$$\xi_T(z_{1:T}, \{k_{1:T}\}) = \frac{\pi_{T,D}(z_{1:T}, \cdot)}{\sum_{k_{1:T} \in [N]_0^T} \pi_{T,D}(z_{1:T}, \cdot)}.$$  

Recall that conditional on the set of particles $Z_{1:T} = z_{1:T}$:

1. the RW-EHMM Algorithm draws $K_{1:T} \sim \xi_T(z_{1:T}, \cdot)$;
2. the i-RW-CSMC Algorithm draws $K_{1:T}$ according to a $\xi_T(z_{1:T}, \cdot)$-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 $\xi_T(z_{1:T}, \cdot)$ 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 $O(N^2T)$ 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

---

\textsuperscript{1}Because the particles are not conditionally independent given the history of the particle system.
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 same value of the set of particles $Z_{1:T} = z_{1:T}$. Hence, we will sometimes drop the dependence on $z_{1:T}$ from the notation. For any $k_{1:T} \in [N]_0^T$, and any $t \in [T]$, set

$$f_t(k_t | k_{t-1}) := \frac{m_t(z_{t-1}^{k_{t-1}}, z_t^{k_t}) G_t(z_t^{k_t})}{\sum_{n=0}^N m_t(z_{t-1}^{k_{t-1}}, z_t^{n}) G_t(z_t^{n})},$$

$$g_t(\tilde{y}_t | k_t) := \begin{cases} \sum_{n=0}^N m_t(z_t^{n}) G_t(z_t^{n}) & \text{if } t = 1, \\
\sum_{n=0}^N m_{t+1}(z_t^{k_t}, z_{t+1}^n) G_{t+1}(z_{t+1}^{n}), & \text{if } t > 1,
\end{cases}$$

with our usual convention that any quantity with time subscript 0 is to be ignored, i.e. $f_t(\cdot | k_0) \equiv f_t(\cdot)$. 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

$$p(k_{1:T}, \tilde{y}_{1:T-1}) = \prod_{t=1}^T f_t(k_t | k_{t-1}) g_t(\tilde{y}_t | k_t)$$

$$= \prod_{t=1}^T m_t(z_{t-1}^{k_{t-1}}, z_t^{k_t}) G_t(z_t^{k_t}) \propto \pi_{T,D}(z_1^{k_1}, \ldots, z_T^{k_T}),$$

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

$$p(k_{1:T} | \tilde{y}_{1:T-1}) = \frac{\pi_{T,D}(z_1^{k_1}, \ldots, z_T^{k_T})}{\sum_{l_{1:T} \in [N]_0^T} \pi_{T,D}(z_1^{l_1}, \ldots, z_T^{l_T})} = \xi_T(z_{1:T}, \{k_{1:T}\}).$$

### 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 $O(N^2T)$ 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, (3) is then nothing more than the forward-filtering recursion which propagates the one-step ahead predictive distributions $W_k^t = p(k_t|\tilde{y}_{1:t-1})$. For $t = 1, \ldots, T$ (and with convention $w^n_0 = 1$):

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

$$w^n_t = p(k_t = n, \tilde{y}_{t-1}|\tilde{y}_{1:t-2}) = \sum_{m=0}^N p(k_{t-1} = m|\tilde{y}_{t-2}) g_{t-1}(\tilde{y}_{t-1}|m) f_t(n|m)$$

$$= \sum_{k_{t-1} \in [N]} \frac{w^{m-1}_{t-1} m_t(z^n_{t-1}, z^n_t) G_t(z^n_t)}{\sum_{l=0}^N w^n_l}.$$

A.2.2 Backward sampling

Likewise, (4) 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

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

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

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

$$= \frac{w^n_t m_{t+1}(z^n_t, z^{k_{t+1}})}{\sum_{n=0}^N w^n_t m_{t+1}(z^n_t, z^{k_{t+1}})}.$$

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 $O(NT)$ operations) can be viewed as first running a CSMC algorithm targeting 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 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 $\delta_{(0,1,...,N)}$ 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 $n$th unnormalised particle weight of the CSMC algorithm targeting the finite-state HMM at time $t$ can then be shown to be given by

$$w_n^t := g_{t-1}(\tilde{y}_{t-1} | a_{t-1}^n) f_t(n | a_{t-1}^n) = m_t(z_{t-1}^n, z_t^n) G_t(z_t^n).$$

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 targeting the finite-state HMM, then this gives the backward-sampling probabilities from (5):

$$\frac{w_{t+1}^k g_t(\tilde{y}_t | k_t) f_{t+1}(k_{t+1} | k_t)}{\sum_{m=0}^N w_{t+1}^m g_t(m | k_{t+1}) f_{t+1}(k_{t+1} | m)} = \frac{m_t(z_{t-1}^n, z_t^n) G_t(z_t^n) m_{t+1}(z_t^n, z_{t+1}^n)}{\sum_{m=0}^N m_t(z_{t-1}^m, z_t^m) G_t(z_t^m) m_{t+1}(z_t^m, z_{t+1}^m)}.$$

1.3 On Corollary 3.5 and Proposition 4.8

Corollary 3.5 is supposed to show the performance stability as $T \to \infty$, but the result is formulated in terms of $\tilde{\alpha}_T^N$ (i.e. the acceptance rate as $D \to \infty$). This double limit makes the statement a little difficult to interpret. Is it possible to derive some stability in the $T \to \infty$ regime alone?

We agree that it would be appealing to obtain results time-horizon stability results such as Corollary 3.5 in finite dimensions. We attempted the following two approaches without success:
1. Our first attempt was to adapt the proof techniques from our manuscript to the finite-dimensional setting. Unfortunately, this is non-trivial because the relevant calculations are enabled by the relatively simple form of the limiting law that we derived and thus cannot be exploited in finite dimensions.

2. Our second attempt was to use proof techniques for the standard i-CSMC algorithm in finite dimensions (Andrieu et al., 2018; Lindsten et al., 2015; Del Moral et al., 2016). However, extending their proof techniques to our setting is non-trivial as these are geared towards proving uniform ergodicity; and since our algorithms reduce to a random-walk Metropolis–Hastings (MH) algorithm if $N = T = 1$, we can at best hope to verify geometric ergodicity (unless we truncate the space, including the proposal distributions).\(^2\)

---

Interestingly, for the i-RW-CSMC algorithm without backward sampling, one might still be able to prove the linear scaling of $N$ with $T$ using a version of the “doubly-conditional” CSMC construction from Andrieu et al. (2018). That is, we can bound the acceptance probability at any time $t$ from below by the following quantity which involves an expectation of a “local” normalising-constant estimate where, noting that the proposal kernels can be factorised into a marginal proposal for the particle and the conditional proposal kernel for the remaining particles as:

$$
\tilde{\alpha}_{T,D,x_1,T}^N(s) \geq \tilde{\alpha}_{T,D,x_1,T}^N(1) = \int S_{\gamma,D}(x_1:T, dx') \mathbb{E}_{x_1:T,x',0,1}^N \left[ \frac{(1 - 1/(N + 1))^T S_{T,D}(x_1:T, \gamma_{T,D})}{\prod_{t=1}^T \sum_{n=0}^N m_t(z_{t-1}^{N-1}, z_t) G_t(z_t^n)} \right],
$$

where, noting that the proposal kernels can be factorised into a marginal proposal for the $n$th particle and the conditional proposal kernel for the remaining particles as:

$$
S_{\gamma,D}(z_t^n, dz_t^n) = S_{T,D}(z_t^n, dz_t^n) S_{T,D}(z_t^n, dz_t^n),
$$

for every $n, m \in [N]$ with $m \neq n$, and writing $z_t^{(m,n)} := (z_t^1)_{t \in [N] \setminus \{m,n\}}$, we have defined the following four quantities which can be understood as the marginal proposal kernel for a single particle path (1); the unnormalised target density (2); the “local” normalising constant in a neighbourhood of $x_1:T$ defined by the marginal proposal kernel (4).

$$
S_{T,D}(x_1:T, dx') := \prod_{t=1}^T S_{T,D}(x_t, dx_t),
$$

$$
\gamma_{T,D}(x_1:T) := \prod_{t=1}^T m_t(x_{t-1}, x_t) G_t(x_t),
$$

$$
S_{\gamma,D}(x_1:T, dx') := \frac{S_{T,D}(x_1:T, dx') \gamma_{T,D}(x_1:T)}{S_{T,D}(x_1:T, \gamma_{T,D})},
$$

$$
S_{T,D}(x_1:T, \gamma_{T,D}) := \int S_{T,D}(x_1:T, dx') \gamma_{T,D}(x_1:T).
$$

We have also let $\mathbb{E}_{x_1:T,x',0,1}^N[\cdot]$ denote expectation w.r.t. the following “doubly-conditional RW-CSMC algorithm” which fixes $z_t^0 := x_t$ and $z_t^1 := x_t'$ for all $t \in [T]$:

$$
\left[ \prod_{t=1}^T \delta_{x_t}(dz_t^0) \delta_{x_t'}(dz_t^1) S_{T,D}(x_t^0, z_t^1, dz_t^2) \right] \times \left[ \prod_{t=2}^T \delta_0(da_{t-1}) \delta_1(da_{t-1}^1) \prod_{n=2}^N \tilde{R}_{t,D}^N((z_{t-1}^1, a_{t-1}), da_{t-1}^n) \right].
$$
We suspect that more classical tools from the MCMC literature such as drift conditions (Mengersen and Tweedie, 1996) may be needed. But again, the extension of these to our more general ($N > 1$ and $T > 1$) setting is highly non-trivial.

Like most algorithms for state-space models, RW-EHMM and i-RW-CSMC have their stability conditioned on some mixing properties (e.g. the Markov kernels $M_t$ must forget the past in some way). It seems to me that the condition $I(\ell) := \sup_{t,T} \ell_t I_{t||T} < \infty$ plays that role here; but the inequality looks much weaker than standard mixing assumptions for other smoothing algorithms (e.g. those found in Andrieu et al. (2018) for i-CSMC or Douc et al. (2011) for FFBS). If this is true, then it could be another advantage of the proposed method?

We agree that it would be interesting to relate our moment assumptions to the strong mixing assumptions (or the weaker assumptions from (Andrieu et al., 2018)). Unfortunately, beyond the fact that both sets of assumptions rule out clearly problematic cases such as $M_t(x_{t-1}, \cdot) = \delta_{x_{t-1}}$, we have not managed to establish further connections between our moment assumptions. It is possible that no meaningful relationship exists because these two sets of assumptions deal with different settings: the former are geared towards “local” proposals whereas the latter are geared towards “independent” proposals and then give uniform ergodicity which is not something one can hope for in our setting.

It might be useful to mention some of the practical implications of these results. First, Corollary 3.5 leads to the following recommendation: one should choose a moderate value of $N$, as pushing $N$ higher than a certain threshold does not appear to further improve mixing. The $O(N^2)$ complexity of RW-EHMM may thus be justifiable in some cases.

We agree and have now added these recommendations to the manuscript. Please see Section 1.5 below for details.

Second, for the i-RW-CSMC algorithm, even with $N$ fixed and $T \to \infty$, the version without backward sampling still gives satisfactory performance for estimating the marginal smoothing distribution $X_t$ if $t$ is at most $O(N)$ away from $T$. This can be seen via a trivial modification of the proof for the first point of Proposition 4.8, as well as in numerical
experiments performed by the authors. Thus if only marginals close to $T$ are of interest, one can use i-RW-CSMC algorithm without backward sampling and without scaling $N$ with $T$. Some connections could be made with the $O(N)$ order of coalescent time in particle filters as shown in, e.g. Koskela et al. (2020).

This is an interesting suggestion. When only the final few marginals of the joint smoothing distribution are of interest, we could use an iterated CSMC type algorithm (such as the i-RW-CSMC algorithm) – in the absence of backward sampling – without scaling the number of particles with $T$. We decided not to include it into the manuscript because of two caveats:

1. this suggestion would equally apply in the “standard” i-CSMC algorithm;

2. if one is really only interested in the final few marginals, it would be computationally cheaper to only apply the algorithm to a modified version of the Feynman–Kac model restricted to time steps $t, t+1, \ldots, T$ (with some arbitrary initial distribution at Time $t$).

We also thank the reviewer for pointing out the reference Koskela et al. (2020). Right before Proposition 4.8, we have now included a reference to Brown et al. (2021) (which is from the same research group as Koskela et al. (2020) and derives similar results but fits our setting slightly because it covers “conditional” SMC rather than “unconditional’ SMC algorithms).

1.4 On the absence of a non-conditional version

The Remark 4.4 is interesting and implies in particular that there is no natural estimator of the normalising constant of $\pi$ associated with the algorithms proposed by the authors. Comparing i-CSMC and i-RW-CSMC reminds the relation between importance sampling (in its MCMC-ised form) and MCMC: the former breaks down in high dimension, but has the advantage of being associated with a natural estimator of the normalising constant.

The point I want to make here is that while RW-EHMM and i-RW-CSMC can break the curse of dimensionality to sample from the smoothing distribution, they cannot do the same to estimate the normalising constant, at least not in an obvious way. Some more involved strategies might be required. For instance, Chapter 17 of Chopin and Papaspiliopoulos (2020) gives a systematic review of techniques using intermediate distributions.
We believe that your are correct that “local” proposals (such as those used by the RW-EHMM and i-RW-CSMC algorithms)

- can circumvent the curse of dimensionality when sampling from the joint smoothing distribution in MCMC-type settings;

- may not work well when the goal is to approximate normalising constants or other expectations of interest in an importance-sampling type setting, e.g. when employed within an “unconditional” SMC counterpart of such algorithms (if it exists).

Intuitively, this is because in importance-sampling type settings, local proposals increase the positive correlation of the particles which in turn increases the variance of any particle-based approximation.

Of course, we cannot discuss this intuition in the context of the our RW-EHMM and i-RW-CSMC algorithms since these do not have an unconditional SMC-type counterpart (as our work shows).

However, it is possible to comment on this intuition in the context of the algorithm proposed in Shestopaloff and Neal (2018) which can be viewed as another iterated CSMC-type algorithm with local proposals. Specifically, their algorithm uses MCMC kernels for proposing local moves around the reference path. And simulation results in Shestopaloff and Neal (2018); Finke et al. (2016) suggest that this works well for sampling from the joint smoothing distribution in high dimensions. However, unlike our i-RW-CSMC algorithm, the algorithm from Shestopaloff and Neal (2018) does have an unconditional SMC counterpart. This unconditional SMC counterpart is known as the sequential MCMC algorithm (see, e.g., Septier and Peters, 2016). Incidentally, if \( T = 1 \), sequential MCMC reduces to MCMC-based importance sampling (see, e.g., Franks and Vihola, 2020, Algorithm 4) mentioned in the above reviewer comment. And in the context of sequential MCMC methods, it has been shown that the additional (positive) particle correlation induced by the local proposals increases the asymptotic variance of particle approximations so that sequential MCMC algorithms are only prefereable to a ‘standard’ unconditional SMC algorithm if they can target a more efficient sequence of intermediate distributions (Finke et al., 2020).

In summary, the empirical and theoretical results from Shestopaloff and Neal (2018); Finke et al. (2016, 2020) support your intuition that whilst local moves can be beneficial in MCMC-type algorithms, they are harmful in importance-sampling type settings such as in unconditional SMC algorithms. And employing them in the latter is only sensible if they permit more efficient distribution flows to be targetted.

1.5 On practical aspects of the algorithms

The paper investigates numerically i-RW-CSMC in a simple Gaussian setting with Markov chains started in stationary. While this is still helpful for illustration purpose, the absence of more realistic examples means
that important open questions remain about practical and numerical aspects of the proposed algorithms.

We agree that the Gaussian example is very “toy” and serves mostly to illustrate the theoretical results. We have now added simulation results for a multivariate stochastic volatility model which has served as a benchmark for SMC-type methods in high dimensions, e.g. in Guarniero et al. (2015). In particular, the joint smoothing distribution in this example is not analytically tractable, i.e. one really must resort to Monte Carlo methods such as the one we propose.

The results are shown in Appendix E.3 which we repeat further down, right after our response to the next comment (we believe that it will make more sense to read our changes in this order).

One important issue (to which the authors alluded) is the lack of a recipe for choosing the scaling parameters \( \ell_t \).

Another obstacle that only appears in high dimension is how to come up with a reasonable starting point for the chain. In low dimension, it is natural to first run a particle filter with moderate \( N \) and then pick one of the smoothed trajectories as the initial path for i-CSMC. In high dimension, it is hard to design such a filter. (The authors also remarked the breakdown of even auxiliary filters, at the end of Subsection 2.2.2.) Therefore, some other initialisation strategy has to be devised.

We have now added the following section to the manuscript which discusses various aspects about the practical implication. In Appendix E we have also included new simulations which provide some support for our suggestions regarding the choice of tuning parameters.

Our newly included simulations for the stochastic volatility model (in Appendix E.3) also illustrate that the i-RW-CSMC algorithm still works well even if the MCMC chain is not initialised from stationarity but using a bootstrap particle filter with relatively few particles.

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 \( \ell_t \). We propose a simple adaptation strategy to specify \( \ell_t[g] \), the value of \( \ell_t \) at the \( g \)th iteration of the algorithm. Let \( \alpha_t[g] \) denote the average acceptance rate at time \( t \) up to the \( g \)th iteration. Then we set

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

Here, \( \alpha \in (0, 1) \) is some target acceptance rate. Motivated by optimal-scaling results in a related multi-proposal MCMC setting (Bédard et al., 2012), and by further empirical results shown in Appendix E.2, we take \( \alpha \) to be an increasing function in \( N \). Specifically, we use \( \alpha := 1 - (N + 1)^{-\beta} \) for some \( \beta \in (0, 1) \), say \( \beta = 1/3 \). In Appendix E.3, we illustrate that this strategy is able to quickly improve poor initial choices of \( \ell_t \). 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.
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 et al. (2015). We stress that this model does not generally satisfy the independent and identically distributed (IID) assumption A1.

Let \( \phi_{m,C} \) denote a density (w.r.t. a suitable version of the Lebesgue measure, \( \lambda \)) of a \( D \)-dimensional normal distribution with mean vector \( m \in \mathbb{R}^D \) and covariance matrix \( C \in \mathbb{R}^{D \times D} \). Let \( y_t = (y_{t,d})_{d \in [D]} \in \mathbb{R}^D \) be a vector of \( D \) log-returns observed at time \( t \in [T] \). Then:

\[
G_t(x_t) := \phi_{0_D, \text{diag} \left( \exp(x_t) \right)}(y_t),
\]
\[
m_t(x_{t-1}, x_t) := \phi_{\mu + \Phi(x_{t-1} - \mu), U}(x_t),
\]

where \( \exp \) is applied element-wise, where \( \text{diag}(x) \) is a diagonal matrix with diagonal given by the vector \( x \) and where \( \mu, \phi \in \mathbb{R}^D \), \( \Phi := \text{diag}(\phi) \), \( U \in \mathbb{R}^{D \times D} \) is some covariance matrix. We also recall that \( 0_D, 1_D \in \mathbb{R}^D \) are column vectors filled with zeros and ones, respectively, and \( I_D \in \mathbb{R}^{D \times D} \) is an identity matrix. At time \( t = 1 \), \( m_1(x_1) = \phi_{\mu, U_*}(x_1) \), where \( U_* \) is the stationary covariance matrix of the latent autoregressive process of log-volatilities, i.e.

\[
\text{vec} U_* = (I_{D^2} - \Phi \otimes \Phi)^{-1} \text{vec} U.
\]

Finally, we assume \( \mu = \nu 1_D \), \( \Phi = \phi I_D \) 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 \( \ell_t \)

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 \( \ell_t \) 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 2 and 1 of this appendix 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. 1.** Averaged (over ‘spatial’ components) squared jumping distance (adjusted for the number of particles) in the multivariate stochastic volatility model.
Finally, we illustrate the adaptive rule for setting the scale factors $\ell_t$ suggested in Section 6. Figure 3 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. 2.** Averaged (over ‘spatial’ components) integrated autocorrelation time (adjusted for the number of particles) in the multivariate stochastic volatility model.

**Fig. 3.** Evolution of the scale factors $\ell_t$ under the adaptive scheme from Section 6.
The influence of the starting point can also be seen in some of the theoretical results in the paper. For instance, Proposition 3.2 implies reasonable performance of RW-EHMM on a family of events $F_{T,D}$ such that $\lim_{T \to \infty} (\pi)_{T,D}(F_{T,D}) = 1$. Despite this last equation, choosing some initial value $x \in F_{T,D}$ might still become much harder as $D$ increases, since we are measuring the volume of $F_{T,D}$ through $(\pi)_{T,D}$.

The scaling-limit result indeed does rely on the fact that the Markov chain is started at stationarity, as has been assumed in the vast majority of literature on the (optimal) scaling of MCMC methods (Roberts and Rosenthal, 2001). The sets $F_{T,D}$ are artefacts of the methods of proof used in the work and rely on the framework of Ethier and Kurtz (2009). The article by Roberts and Rosenthal (2016) discusses the relationship between the diffusion-limit approach and other notions of mixing efficiency. When the Markov Chain is started far in the “tail” of the stationary distribution, one cannot expect a diffusion process to describe the initial “transient” phase necessary for the Markov chain to reach the typical set. As a matter of fact, in high-dimensional settings, random-walk Metropolis-type methods are often described by an ODE-limit (i.e. fluid limit) in the transient phase (Christensen et al., 2005; Jourdain et al., 2015). Although it is indeed an interesting question, it is very much out of the scope of the current paper to study this regime.

Nonetheless, we stress that we have found the algorithm to work well if the initial value is drawn from an “unconditional” SMC algorithm with a moderate number of particles (see Appendix E.3).

1.6 Typos

Some typos in the manuscript:

- Page 10, first line of the second paragraph of Section 3.1.1: “[...] RW-EHMM scheme is outlined in Algorithm 3”. It should read “Algorithm 2”.
- Page 11, fourth bullet point. “[...] first sampling a new ‘centre’ [...] $N(z_{t,D}^0, \ell_t/2)$”. The variance should read $\frac{\ell_t}{2D}$.
- Page 13, statement of Proposition 3.3. $\mathcal{I}_{t,T}$ should replace $\mathcal{I}_t$.
- Corollary 3.5 and Proposition 4.8: the regularity condition should read $\sup_{T \in \mathbb{N}} \sup_{t \in [T]} \ell_t \mathcal{I}_{t,T} < \infty$. 

We are grateful to the reviewer for pointing these out. We have fixed all of these. Regarding the last bullet point, we additionally fixed a similar typo in Appendices B.1 and C.1.

2 Reviewer 2:

This paper by Finke and Theiry addresses a topical and important problem on inference for high-dimensional state-space problems by devising algorithmic variants of Particle Markov Chain Monte Carlo (MCMC) by Andrieu et al. (2010) that work in the high dimensional setting. The authors do this by embedding a random walk proposal within the Particle MCMC algorithm. In the reported numerical example, the authors demonstrate the new algorithm does appear to work well in higher dimensions.

Positive of the paper include:

- The authors new embedded random walk algorithm is quantified with a careful delineation of its respective limiting distribution (Prop 3.2 and Prop 4.5).
- The line of analysis, compared to previous relevant work in Particle MCMC, appears novel.
- The presentation of the paper is clear and methodical with essentially no typos (that I could see).

On balance, after weighing up the positives and ‘negatives,’ I think accepting this contribution would be a good decision.

Thank you for the kind comments.
The results on the failure of Particle MCMC in the high-dimensional setting, and the success of the authors’ methods, are slightly opaque and difficult to interpret. Namely they are phrased in terms of maximised distance over a set increasing probability (eg Prop 2.2; Prop 4.5 etc). Without a careful understanding of how these sets are defined, it is hard for the reader to appreciate how tight these results are, if at all. I think if possible some clarification and/or mitigation here is needed. (For example, are there similar results in the literature where this line of proof is routine and/or is known to be tight).

We agree that, for example, we do not describe the rate at which the probability \( \pi_{T,D}(F_{T,D}) \) converges to one. In other words, our statement only shows that with overwhelming probability (i.e. probability converging to one), the stated properties hold (e.g. uniform concentration around the mean on sets of increasing probability as \( D \to \infty \) in Propositions 3.2 & 4.5). The exact description of the set \( F_{T,D} \) is in the proof of Proposition 4.5, although we do not find this description particularly illuminating and have chosen to keep this description in Appendix D.5. In the optimal scaling literature, these types of arguments and results are relatively standard. For example, in the seminal article (Roberts et al., 1997) on the optimal scaling of the random-walk Metropolis algorithm in high-dimensional settings, the proof already relies on the existence of similar sets (see, e.g., the definition of the set \( F_n \) in the proof of Theorem 1.1 of Roberts et al. 1997). As a matter of fact, a large number of results in the optimal scaling literature (Roberts and Rosenthal, 2001) do rely on the existence of such sets because the methods of proof rely on Theorem 8.7 in Chapter 4 of Ethier and Kurtz (2009) which relies on similar constructions. Studying the size of the largest (i.e. optimal) sets \( F_{T,D} \) that imply the results stated in Proposition 3.2 and 4.5 is out of the scope of this paper.

The numerical results are not for any practical problem but a stylised one.

We agree that the Gaussian example is very “toy” and serves mostly to illustrate the theoretical results. We have now added simulation results for a multivariate stochastic volatility model which has served as a benchmark for SMC-type methods in high dimensions, e.g. in Guarniero et al. (2015). In particular, the joint smoothing distribution in this example is not analytically tractable, i.e. one really must resort to Monte Carlo methods such as the one we propose. In addition, in this example, the Markov chain cannot be initialised from stationarity. Due to the space restrictions, we had to place these results in the appendix (Appendix E.3). They can also be found in Section 1.5 within our response to Reviewer 1.
The authors implicitly/explicitly imply that random walk (RW) within the particle type setting (Alg 2) is better than random walk on the smoother directly (defined in sec 2.1 example 1). Has this been verified somewhere? What about the blocked sampling approach with RW proposals (of the type authors have devised) – might this be competitive?

We have now added simulation results (included in Appendix E.2) which demonstrate that our i-RW-CSMC algorithm indeed outperforms a Gaussian random-walk MH algorithm on the full $(T \times D)$-dimensional space – even if the latter is allowed to make multiple, i.e. $N$, proposals. This is not surprising because the former exploits the decorrelation in the “time”-direction whereas the latter does not.

Of course, as we already discussed in our “Conclusion” section, blocking (in the “time”-direction, i.e. with each block consisting of one $D$-dimensional latent state) is closely related to backward sampling and similarly ensures that a MCMC algorithm can exploit the decorrelation in the “time”-direction. And indeed, in our simulations, a blocked version of the above-mentioned multi-proposal MCMC algorithm displayed a similar performance as the i-RW-CSMC algorithm. However, it is well known that the performance of such blocked Metropolis-within-Gibbs type algorithms can be highly sensitive to the choice of block sizes. In contrast, no tuning of block sizes is needed within our algorithm.

We have now added Appendix E.2 to the manuscript which discusses these observations in more detail and which is also repeated below.

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-proposal” versions which make $N$ proposals. Specifically, we compare the following four methods, where we recall that $\Phi^n$ denotes the Rosenbluth–Teller selection function which was defined in (2) and which reduces to the usual MH acceptance probability $\Phi^1 = 1 \land \exp$ 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 1 in this appendix summarises the method, where we use the convention that \(z_{1:T}^n := (z^n_1, \ldots, z^n_T)\). 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-proposal RWMH algorithm. Each state \(x_t\) 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 \(\ell_t/D\), for scale factors \(\ell_1, \ldots, \ell_T > 0\). Algorithm 2 in this appendix summarises the method. For \(N = 1\), this algorithm reduces to a standard blocked Gaussian RWMH algorithm.

---

**Algorithm 1 (RWMH)** Given \(x_{1:T} := x_{1:T}[l] \in E_{T,D}\).

1. Sample all particles \(Z^n_t = z^n_t\) as in Step 1 of Algorithm 2 in the main manuscript, where \(\ell_1 = \ldots = \ell_T = \ell\), for some \(\ell > 0\).
2. Sample \(K = k \in [N]_0\) with probability \(\Phi^k(\{h^m\}_{m=1}^N)\), where
   \[
   h^m := \log \pi_{T,D}(z^n_{1:T}) - \log \pi_{T,D}(z^0_{1:T}).
   \]
3. Set \(X'_1 := x'_1 := z^k_{1:T}\).
4. Return \(x_{1:T}[l + 1] := x'_{1:T}\).
Algorithm 2 (Blocked RWMH) Given $x_{1:T} := x_{1:T}[l] \in E_{T,D}$.

1. Sample all particles $Z_t^n = z_t^n$ as in Step 1 of Algorithm 2 in the main manuscript.

2. For $t = 1, \ldots, T$,
   a) sample $K_t = k_t \in [N]_0$ with probability $\Phi_{k_t} \{ \{ h_m^m \}_{m=1}^N \}$, where
      \[ h_t^m := \log \pi_{T,D}(x_{1:t-1}^t, z_t^m, x_{t+1:T}) - \log \pi_{T,D}(x_{1:t-1}^t, x_{t:T}) , \]
   b) set $X_t' := x_t' := z_t^{k_t}$.

3. Return $x_{1:T}[l + 1] := x_{l:T}'$.

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 $\ell_t$ in the i-RW-CSMC and blocked RWMH algorithms and the scale factor $\ell$ in the RWMH algorithm. Recall that as discussed in Section 6, we adapt the scale factors so that the acceptance rate is around $\alpha$.

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 4 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 et al. (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 com-
plectity. 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.

\[ N = 1 \quad N = 2 \quad N = 5 \quad N = 10 \quad N = 20 \quad N = 50 \]

\[ D = 1 \quad D = 2 \quad D = 5 \quad D = 10 \quad D = 20 \quad D = 50 \]

**Fig. 4.** Performance of the algorithms discussed in this section in different dimensions and for various choices of the tuning parameters \( N \) and \( \alpha \).
Thank you for pointing this out. This is now fixed.

References


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 \( O(e^{-DT}) \) 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 $O(1)$ (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 $\log(N) = o(D)$ – 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 $O(D)$ (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 $O(TD)$. The use of backward sampling again removes the need for growing $N$ with $T$ so that the overall complexity can be brought down to $O(D)$. 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 Rowes (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
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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Without backward sampling</th>
<th>With backward sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-CSMC</td>
<td>$O(Te^D)$</td>
<td>$O(e^D)$</td>
</tr>
<tr>
<td>i-RW-CSMC*</td>
<td>$O(TD)$</td>
<td>$O(D)$</td>
</tr>
<tr>
<td>RW-EHMM</td>
<td>$O(D)$</td>
<td></td>
</tr>
</tbody>
</table>

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 $O(N^2)$ operations to implement rather than $O(N)$ 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.

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 \(\mathcal{N}(\mu, \Sigma)\) denotes a normal distribution with mean vector \(\mu\) and covariance matrix \(\Sigma\); \(\delta_x\) 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 \([n] := \{1, 2, \ldots, n\}\) and \([n]_0 := \{0, 1, 2, \ldots, n\}\) and 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 \(\mathbf{I}_n \in \mathbb{R}^{n \times n}\) 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 \(h^0 := 0\), we define the Boltzmann selection function

\[
\Psi^n(\{h^m\}_{m=1}^N) := \frac{\exp(h^n)}{1 + \sum_{m=1}^N \exp(h^m)},
\]

as well as the Rosenbluth–Teller selection function

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

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 \(1 - \sum_{m=1}^N \exp(h^m) / \sum_{l \in [N] \setminus \{n\}} \exp(h^l)\); 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 \((E, \mathcal{E}) := (\mathbb{R}^D, B(\mathbb{R})^{\otimes D})\), let \(M_1 \in \mathcal{P}(E)\) be a probability measure with density \(m_1 : E \to [0, \infty)\). Furthermore, for \(t \geq 2\), let \(M_t : E \times E \to [0, 1]\) be some Markov kernel with transition density \(m_t : E \times E \to [0, \infty)\). Furthermore, for \(t \geq 1\), let \(G_t : E \to (0, \infty)\) be strictly positive measurable potential functions. The methods discussed in this work target the following probability measure on \((\mathcal{E}_{T,D}, \mathcal{E}_{T,D}) := (\mathcal{E}^T, \mathcal{E}^{\otimes T})\):

\[
\pi_{T,D}(dx_{1:T}) \propto M_1(dx_1)G_1(x_1)\prod_{t=2}^T M_t(x_{t-1}, dx_t)G_t(x_t).
\]

**Example** (state-space model). Let \((X_t, Y_t)_{t \geq 1}\) be a Markov chain on a space \(E \times F\) with initial distribution \(M_1(dx_1)H_1(x_1, dy_1)\) and transition kernels \(M_t(x_{t-1}, dx_t)H_t(x_t, dy_t)\). Assume that for each time \(t \in [T]\), we observe a realisation \(y_t \in F\) of \(Y_t\) but \(X_t\) is unobserved (‘latent’). We are then typically interested in computing (at least approximately) the posterior distribution of the latent ‘states’ \(X_{1:T}\), often called the joint smoothing distribution:

\[
\pi_{T,D}(dx_{1:T}) = \mathbb{P}(X_{1:T} \in dx_{1:T} | Y_{1:T} = y_{1:T}).
\]

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 \(y_{1:T}\) to be ‘fixed’ (so that they can be dropped from the notation) and assuming that \(G_t(x_t) = h_t(x_t, y_t)\), where \(h_t(x_t, \cdot)\) is a density of \(H_t(x_t, \cdot)\) 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 \(\pi_{T,D}\) is typically intractable and must be approximated, e.g. using MCMC methods. It is therefore crucial to design \(\pi_{T,D}\)-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 \(\pi_{T,D}\)-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 $O(e^T)$ to at most $O(T)$ and thus avoids a curse of dimension in the time horizon $T$.

For any $t \in [T]$ and any $x_{1:T} \in E_{T,D}$, define

$$w_t(x_t) := \log G_t(x_t).$$

The $l$th update of the i-CSMC scheme is then outlined in Algorithm 1, where we use the convention that any action described for the $n$th 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 $M_1(z_0^n, \cdot) \equiv M_1(\cdot)$.

**Algorithm 1** (i-CSMC). Given $x_{1:T} := x_{1:T}[l] \in E_{T,D}$.

1. For $t \in [T]$, 
   a) if $t > 1$,
      i. set $A_{t-1}^n = a_{t-1}^n := 0$,
      ii. sample $A_t^n = a_t^n \in [N]_0$ with probability
         $$\psi^{a_{t-1}^n}((w_{t-1}(z_{m-1}^n) - w_{t-1}(z_0^n))_{m=1}^N) = \frac{G_t \left( z_{t-1}^n \right)}{\sum_{m=0}^N G_t \left( z_{t-1}^m \right)},$$
   b) set $Z_t^0 = x_t^0 := x_t$ and sample $Z_t^n = z_t^n \sim M_t(z_{t-1}^{a_t^n}, \cdot)$.
2. Sample $K_t = k_t \in [N]_0$ with probability
   $$\psi^{k_t} \left( \{ w_T(z_T^n) - w_T(z_T^0) \}_{m=1}^N \right) = \frac{G_T(z_T^{k_t})}{\sum_{m=0}^N G_T(z_T^m)}.$$
3. Set $K_t^i = k_t := a_i^{k_t+1}$, for $t = T - 1, \ldots, 1$.
4. Set $X_{1:T}^l := x_{1:T}^l := (z_T^{K_t}, \ldots, z_T^{k_t})$.
5. Return $x_{1:T}[l + 1] := x_{1:T}^l$.

Step 1 of Algorithm 1 which a) performs (conditional) multinomial resampling by drawing the ancestor indices $A_t^n$; and b) generates the particles $Z_t^n$, 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 $\pi_{1,D}(dx_1) \propto M_1(dx_1) G_1(x_1)$ and Algorithm 1 proposes $Z_1^0 = z_1^0 \sim M_1$ and accepts this proposal as the new state of the Markov chain with probability

$$\psi^1 \left( w_1(z_1^0) - w_1(z_1^0) \right) = \frac{G_1(z_1^0)}{G_1(z_1^0) + G_1(z_1^0)},$$

where $z_1^0 = x_1$. This can be recognised as a version of Barker’s kernel (Barker, 1965) with independence proposal $M_1$ (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. (2017); 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

$$
\phi_k^x \left( \left( w_T(z_T^m) - w_T(z_T^0) \right)_{m=1}^N \right) = \frac{G_T(z_T^k)}{\sum_{m=1}^N G_T(z_T^m) - G_T(z_T^k) \land G_T(z_T^0)}.
$$

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 $Z_t^1 = z_t^1 \sim M_1$ with probability

$$
\phi^1 \left( w_1(z_1^1) - w_1(z_1^0) \right) = 1 \land \frac{G_1(z_1^1)}{G_1(z_1^0)},
$$

where $z_1^0 = x_1$. 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 $x_{1:T}[l + 1]$ to one of the $N + 1$ particle lineages generated under the CSMC algorithm in Step 1 which often coalesce with the old reference path $x_{1:T}[l]$. 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

$$
\psi_{k_t} \left( \left( v_t(z_t^m, z_{t+1}^{k_t+1}) - v_t(z_t^0, z_{t+1}^{k_t+1}) \right)_{m=1}^N \right) = \frac{G_t(z_t^k)m_{t+1}(z_t^k, z_{t+1}^{k_t+1})}{\sum_{m=0}^N G_t(z_t^m)m_{t+1}(z_t^m, z_{t+1}^{k_t+1})},
$$

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

$$
v_t(x_{t+1}) := w_t(x_t) + \log m_{t+1}(x_t, x_{t+1}).
$$

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 $O(T)$ to $O(1)$. 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’ $M_t(x_{t-1}, \cdot)$ and draws the parent indices $A_{t-1}^p$ 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 $\pi_{T,D}$-invariant Markov kernel

Given $X_{1:T} = x_{1:T} = x_{1:T}[l]$, let

$$
\P_{T,D;X_{1:T}}^N(dx_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx'_{1:T})
$$

be the law of all the random variables $(Z_{1:T}, A_{1:T-1}, K_{1:T}, X_{1:T}'$) 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 $\mathbb{E}_{T,D,x_1:T}^N$ denote expectation w.r.t. $\mathbb{P}_{T,D,x_1:T}^N$. Algorithm 1 induces a Markov kernel
$$\mathbb{P}_{T,D}^N(x_1:T, dx'_{1:T}) := \mathbb{E}_{T,D,x_1:T}^N[\{X'_{1:T} \in dx'_{1:T}\}],$$
for $(x_1:T, dx'_{1:T}) \in \mathbb{E}_{T,D} \times \mathbb{E}_{T,D}$. The following proposition shows that this Markov kernel leaves $\pi_{T,D}$ 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 $N,T,D \in \mathbb{N}$, $\pi_{T,D} \mathbb{P}_{T,D}^N = \pi_{T,D}$.  

For any $t \in [T]$, we call
$$\alpha_{T,D,x_1:T}^N(t) := \mathbb{E}_{T,D,x_1:T}^N[\{K_t \neq 0\}]$$
the acceptance rate at time $t$ associated with Algorithm 1. This name is justified because $K_t = 0$ in Algorithm 1 implies $x'_t = x_t$, i.e. $x_t[l + 1] = x_t[l]$.

### 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
$$M_t(x_{t-1}, dx_t) = \prod_{d=1}^D M_t(x_{t-1,d}, dx_{t,d}) \quad \text{and} \quad G_t(x_t) = \prod_{d=1}^D G_t(x_{t,d}),$$
with the convention that any quantity with time index 0 is to be ignored and where
- $x_t = x_{t,1:D} \in \mathbb{E}$, recalling that $(\mathbb{E}, \mathcal{E}) = (\mathbb{R}^D, \mathcal{B}(\mathbb{R}) \otimes \mathcal{D})$;
- $M_1 \in \mathcal{P}(\mathbb{R})$ is a probability measure with density $m_1: \mathbb{R} \to [0, \infty)$ and, for $t \geq 2$, $M_t: \mathbb{R}^2 \to [0, 1]$ is a Markov kernel with transition density $m_t: \mathbb{R}^2 \to [0, \infty)$;
- $G_t: \mathbb{R} \to (0, \infty)$ is a strictly positive and measurable potential function.

Thus, under **A1**, $\pi_{T,D} = \pi_T^{\otimes D}$, with the following probability measure on $\mathbb{R}^T$:
$$\pi_T(dx_{1:T}) \propto M_1(dx_1)G_1(x_1) \prod_{t=2}^T M_t(x_{t-1}, dx_t)G_t(x_t).$$

#### 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.
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 \( z_t^m \) and \( z_t^n \) iff \( a^m_{t-1} = m \). 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 \( x_{1:T}[l] = (z_0^0, \ldots, z_T^0) \). The blue line (-----) represents the new reference path \( x_{1:T}[l+1] = (z_1^1, \ldots, z_T^k) \).

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 \( \mathbb{P}^N_T(\phi = \pi_1 \times \pi_{k_1} \times \ldots \times \pi_{k_{T-1}} \times \pi_{k_T}) \) which deterministically sets all ancestor indices and all indices of the new reference path to 0. More formally,

\[
\mathbb{P}^N_T(\phi = \pi_1 \times \pi_{k_1} \times \ldots \times \pi_{k_{T-1}} \times \pi_{k_T}) := \delta_0 \otimes ((N+1)(T-1)+T)(\phi = \pi_1 \times \pi_{k_1} \times \ldots \times \pi_{k_{T-1}} \times \pi_{k_T}).
\]

We let \( \mathbb{E}^N_T \) 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

\[
r_1[\pi_T] := \mathbb{E}[\log G_t(X_t)] - \mathbb{E}[\log M_t(G_t)(X_{t-1})],
\]

\[
b_1[\pi_T] := \mathbb{E}[\log G_t(X_t) + \log m_{t+1}(X_t, X_{t+1})] - \mathbb{E}[\log M_t(G_t m_{t+1}(\cdot, X_{t+1}))(X_{t-1})].
\]

A2 \( \inf_{t \in [T]} r_1[\pi_T] := r_T > 0 \).

A3 \( \inf_{t \in [T-1]} b_1[\pi_T] := b_T > 0 \). \( \checkmark \)

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
\[
d_{T,D,x_1:T}^N := \|\mathbb{E}_T^N[\{ (A_{1:T-1}, K_{1:T}) \in \cdot \}] - \mathbb{E}_T^N[\{ (A_{1:T-1}, K_{1:T}) \in \cdot \}] \|.
\]
Then there exists a family $F_{T,D} \in \mathcal{E}_{T,D}$ with $\lim_{D \to \infty} \pi_{T,D}(F_{T,D}) = 1$ and
\[
\log N = o(D) \implies \lim_{D \to \infty} \sup_{x_1:T \in F_{T,D}} d_{T,D,x_1:T}^N = 0.
\]
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 $F_{T,D} \in \mathcal{E}_{T,D},$ for any $t \in [T]:$
\[
\log N = o(D) \implies \lim_{D \to \infty} \sup_{x_1:T \in F_{T,D}} \alpha_{T,D,x_1:T}^N(t) = 0.
\]


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 $O(N^2T)$ operations whilst a single i-RW-CSMC update only requires $O(NT)$ operations.

3.1.1. Basic algorithm. The RW-EHMM algorithm proposed in this section also induces a $\pi_{T,D}$-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 $O(NT)$ operations whereas the latter typically require $O(N^2T)$ operations.

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

**Algorithm 2 (RW-EHMM).** Given $x_{1:T} := x_{1:T}[l] \in \mathcal{E}_{T,D}$.

1. For $t \in [T]$: set $Z_t^l := z_{t}^l := x_t[l]$ and sample $Z_{1:T}^1 := z_{1:T}^1$ as follows:
   a) sample $U_{t,d}^1 \sim N(0, \Sigma)$,
   b) set $z_{t,d}^0 := z_{t,d}^0 + \sqrt{\xi_t/DU_{t,d}^1}$,
   c) set $z_{t,d}^n := z_{t,d}^n$.
2. Sample $K_{1:T} = k_{1:T} \in [N]_T$ with probability
   \[
   \xi_T(z_{1:T}, \{ k_{1:T} \}) := \frac{\pi_{T,D}(z_{1:T}^1, \ldots, z_{1:T}^k)}{\sum_{l_1:T \in [N]_T} \pi_{T,D}(z_{1:T}^1, \ldots, z_{1:T}^l)}.
   \]
3. Set $x_{1:T} := x_{1:T}^l := (z_{1:T}^1, \ldots, z_{1:T}^k)$.
4. Return $x_{1:T}[l + 1] := x_{1:T}^l$.

Step 1 of Algorithm 2 scatters particles around the reference particle $z_t^l = x_t$ by adding correlated Gaussian noise independently in each dimension $d \in [D]$: \[
(Z_{1:d}^1, \ldots, Z_{t-1:d}^k, Z_{t:d}^k) \sim N(1_{N}, z_{1:d}^0, \Sigma).
\]
where \( \mathbf{I}_N \) is a vector of 1s of length \( N \) and where

- \( \Sigma := \frac{1}{2}(1_N \mathbf{I}_N^T + \mathbf{I}_N) \) 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 \( \ell_t/D \), i.e. \( \mathbf{Z}_t^n \sim N(\mathbf{z}_t^n, \ell_t/D \mathbf{I}_D) \), for \( n \in [N] \);
- (3) can be viewed as first sampling a new 'centre' \( \mathbf{z}'_{t,t} = \mathbf{z}'_{t,d} \sim N(\mathbf{z}_t^0, \ell_t/D_D) \) and then sampling \( \mathbf{z}^0_{t,t}, \ldots, \mathbf{z}^N_{t,t} \sim N(\mathbf{z}_t, \ell_t/D_D) \):

\[
\int_{-\infty}^{\infty} N(\mathbf{z}'_{t,t}, \ell_t/D_D) \prod_{n=1}^{N} N(\mathbf{z}_t^n, \mathbf{z}'_{t,t}, \ell_t/D_D) \, d\mathbf{z}' = N(\mathbf{z}^0_{t,t}; \mathbf{1}_N, \ell_t/D_D \Sigma).
\]

Other types of ‘local’ proposals (i.e. not necessarily based on Gaussian random walks) could be used as well. 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 \( \lambda \) denote a suitable version of the Lebesgue measure, \( \mathbf{z}^{-n}_{t,t} := (\mathbf{z}_t^n, \ldots, \mathbf{z}_{n+1}^n, \ldots, \mathbf{z}_t^N) \), and \( \mathbf{z}^{-n}_t := (\mathbf{z}_t^n, \ldots, \mathbf{z}_t^N) \), the proposal \( \mathcal{S}^N_{t,t}(\mathbf{z}_t^n, d\mathbf{z}^{-n}_t) := \prod_{d=1}^{T} N(d\mathbf{z}_t^n; 1_N, \ell_t/D_D \Sigma) \) induced by Step 1 of Algorithm 2 satisfies:

\[
\lambda(d\mathbf{z}^k) \mathcal{S}^N_{t,t}(\mathbf{z}_t^n, d\mathbf{z}^k) = \lambda(d\mathbf{z}^k) \mathcal{S}^N_{t,t}(\mathbf{z}_t^n, d\mathbf{z}^k), \quad \text{for any } j, k \in [N].
\]

3.1.2. Implementation in \( O(N^2T) \) operations. Even though Step 2 of Algorithm 2 requires sampling from the distribution \( \pi_{t,T}(\mathbf{z}_1^T, \cdot) \) whose support is \((N + 1)^T\)-dimensional, Neal (2003) recognised that this can be achieved in \( O(N^2T) \) operations as follows:

1. **Forward filtering.** For \( t = 1, \ldots, T \), compute (with convention \( w_0^n := 1 \)):

\[
\begin{align*}
\mathbf{w}_t^n &:= \sum_{m=0}^{N} \frac{\mathbf{w}_t^{m-1}}{\sum_{l=0}^{N} \mathbf{w}_t^l} \mathbf{m}_t(\mathbf{z}_t^n; \mathbf{w}_t^n) \mathbf{G}_t(\mathbf{z}_t^n). \\
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{w}_t^{k_t} \mathbf{m}_{t+1}^{k_t} &:= \sum_{n=0}^{N} \frac{\mathbf{w}_t^{n}}{\mathbf{w}_t^{k_t}} \mathbf{m}_{t+1}^{k_t} \mathbf{z}_t^n; \mathbf{z}_t^{k_t+1}. \\
\end{align*}
\]

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 \( \pi_{T,D} \)-invariant Markov kernel. Given \( \mathbf{X}_{1:T} = \mathbf{x}_{1:T} = \mathbf{x}_{1:T}[l] \), let

\[
\mathbb{P}_{T,D,x_{1:T}}^N(\mathbf{dx}_{1:T} \times d\mathbf{x}_{1:T}) = \mathbb{P}_{T,D,x_{1:T}}^N(\mathbf{dx}_{1:T} \times d\mathbf{x}_{1:T}).
\]

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 \( \mathbb{E}_{T,D,x_{1:T}}^N \) denote expectation w.r.t. \( \mathbb{P}_{T,D,x_{1:T}}^N \). Algorithm 2 induces a Markov kernel

\[
\mathbb{P}_{T,D}^N(\mathbf{x}_{1:T}, d\mathbf{x}_{1:T}^t) := \mathbb{E}_{T,D,x_{1:T}}^N(\{\mathbf{X}_{1:T} \in d\mathbf{x}_{1:T}^t\}),
\]

for \((\mathbf{x}_{1:T}, d\mathbf{x}_{1:T}^t) \in \mathcal{E}_{T,D} \times \mathcal{E}_{T,D} \). The following proposition shows that this Markov kernel leaves \( \pi_{T,D} \) invariant.
PROPOSITION 3.1. For any $N, T, D \in \mathbb{N}$, $\pi_{T,D} \tilde{P}_N^{T,D} = \pi_{T,D}$.

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:

$$
\pi_{T,D}(d\mathbf{x}_{1:T} \times dk_{1:T} \times d\mathbf{z}_{1:T}) = \pi_{T,D}(d\mathbf{x}_{1:T}) \text{Unif}_{[N]^T}(dk_{1:T}) \prod_{t=1}^{T} \delta_{x_t}(z_t^k) S_{1,D}^{N}(x_t^k, dz_t^{-k_t}),
$$

where $\text{Unif}_A$ is the uniform distribution on a set $A$. Step 1 of Algorithm 2 then samples from $\pi_{T,D}(d\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, k_{1:T})$ while Steps 2 and 3 jointly sample from $\pi_{T,D}(d\mathbf{x}_{1:T} \times dk_{1:T}|\mathbf{z}_{1:T})$. 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. $\pi_{T,D}$-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 $\partial_t$ denotes the $i$th derivative w.r.t. $x_t$ and with $\partial_t := \partial_1^T$, as well as with $\pi_T(\varphi) := \int_{\mathbb{R}^T} \varphi(x_{1:T}) \pi_T(x_{1:T}) dx_{1:T}$, for any $\pi_T$-integrable function $\varphi : \mathbb{R}^T \to \mathbb{R}$, we make the following moment assumption.

B1 The density $\pi_T$ 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

$$
\tilde{\pi}^{N}_{T}(dv_{1:T} \times dk_{1:T}) := \left[ \prod_{t=1}^{T} \mathbb{N}(dv_{t}; \tilde{\mu}_{t|T}, \tilde{S}_{t|T}) \right] \prod_{t=1}^{T} \psi^k_t(\{v_{m_t}^m\}_{m=1}^N).
$$

Expectations w.r.t. $\tilde{\pi}^{N}_{T}$ are denoted by $\tilde{E}^N$. This law is defined through $N$-dimensional Gaussian random vectors $V_t := V_{t|T}^{1:N}$ which are such that $V_s$ and $V_t$ are independent whenever $s \neq t$ and with mean vector and covariance matrix

$$
E[V_t] = -\frac{1}{2} \ell_t \mathcal{I}_{t|T}, \quad \text{and} \quad \text{var}[V_t] = \ell_t \mathcal{I}_{t|T},
$$

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

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

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

$$
\tilde{d}^N_{T,D,X_{1:T}} := \parallel \mathbb{E}^N_{T,D,X_{1:T}}[\{K_{1:T} \in \cdot\}] - \mathbb{E}^N_T[\{K_{1:T} \in \cdot\}] \parallel.
$$

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

$$
\lim_{D \to \infty} \sup_{X_{1:T} \in F_{T,D}} \tilde{d}^N_{T,D,X_{1:T}} = 0.
$$

The following corollary which shows that the limiting acceptance rates $\tilde{\alpha}_T$ bounded away from zero as $T \to \infty$

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

converges to a strictly positive limit

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

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 $F_{T,D} \in \mathcal{E}_{T,D}$ as in Proposition 3.2,

$$
\lim_{D \to \infty} \sup_{X_{1:T} \in F_{T,D}} |\tilde{\alpha}^N_{T,D,X_{1:T}}(t) - \tilde{\alpha}_T^N(t)| = 0,
$$

where

$$
\tilde{\alpha}_T^N(t) \geq \left(1 + \frac{\exp(\ell_t I_{t|T})}{N}\right)^{-1} > 0.
$$

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 et al., 2009), which (for the time-$t$ component in Algorithm 2) is given by

$$
\text{ESJD}^N_{T,D}(t) := \mathbb{E}[\parallel X_l[l+1] - X_l[l] \parallel^2],
$$

where $\parallel \cdot \parallel_2$ denotes the Euclidean norm and where $X_{1:T}[l]$ is the $l$th state of the Markov chain with transition kernel $P^N_{T,D}$ 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 $t \in [T]$,

$$
\lim_{D \to \infty} |\text{ESJD}^N_{T,D}(t) - \ell_t \tilde{\alpha}_T^N(t)| = 0.
$$

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 $\tilde{\alpha}_T(t)$ 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 $I(\ell) := \sup_{T \in \mathbb{N}} \sup_{t \in [T]} \ell_l I_{t|T} < \infty$. Then, for any $N \in \mathbb{N}$,

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

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

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 \( \pi_{T,D} \)-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 \( O(N^2T) \) 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 \( O(NT) \) operations as in the standard i-CSMC approach.

For any \( t \in [T] \) and any \( x_{t:T} \in E_{T,D} \), we define

\[
\mathbf{w}_t(x_{t-1:t}) := \log m_t(x_{t-1}, x_t) + \log G_t(x_t),
\]

with the convention that any quantity with time index \( t < 1 \) is to be ignored. The \( l \)th 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 \( n \)th particle index is to be performed conditionally independently for all \( n \in [N] \) and any action described for the \( d \)th ‘spatial’ component is to be performed conditionally independently for all \( d \in [D] \).

**Algorithm 3 (i-RW-CSMC).** Given \( x_{1:T} := x_{1:T}[1] \in E_{T,D} \).

1. For \( t \in [T] 
    a) if \( t > 1 \)
       i. set \( A_t^{0} = a_t^{0} := 0 \)
       ii. sample \( A_t^{n} = a_t^{n} = l \in [N] \) with probability

\[
\psi_l(\{\mathbf{w}_{t-1}(z_{t-2}, z_{t-1}), \mathbf{w}_{t-1}(z_{t-2}, z_{t-1})\})^{N}_{m=1} = \frac{m_{t-1}(z_{t-2}, z_{t-1})G_{t-1}(z_{t-1})}{\sum_{m=0}^{N} m_{t-1}(z_{t-2}, z_{t-1})G_{t-1}(z_{t-1})},
\]

b) set \( Z_t^{0} = z_t^{0} := x_t \) and sample \( Z_t^{1:N} = z_t^{1:N} \) as follows:

   i. sample \( U_{t,d}^{1:N} \sim N(0, \Sigma) \)
   ii. set \( z_t^{n} := z_t^{n} = \sqrt{U_{t,d}^{1:N}} + \sqrt{D}/Du_{t,d}^{1:D} \)
   iii. set \( z_t^{n} := z_t^{n} = \sqrt{U_{t,d}^{1:D}} \)

2. Sample \( K_T = k_T \in [N] \) with probability

\[
\psi_{k_T}(\{\mathbf{w}_T(z_{T-1}^{k_T}, z_T^{m}) - \mathbf{w}_T(z_{T-1}^{k_T}, z_T^{m})\})^{N}_{m=1} = \frac{m_T(z_{T-1}^{k_T}, z_T^{m})G_T(z_T^{m})}{\sum_{m=0}^{N} m_T(z_{T-1}^{k_T}, z_T^{m})G_T(z_T^{m})},
\]

3. Set \( K_t = k_t := a_t^{k_t+1} \), for \( t = T - 1, \ldots, 1 \).
4. Set \( x_{1:T} := x_{1:T} := (z_0^{1}, \ldots, z_T^{k_T}) \).
5. Return \( x_{1:T}[l + 1] := x_{1:T} \).

Step 1 of Algorithm 3 which a) performs (conditional) multinomial resampling by drawing the ancestor indices \( A_t^0 \); and b) generates the particles \( Z_t^n \) by scattering them around the reference particle \( z_t^0 = x_t \) 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 \( k_T \)th 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 \( \xi_T(z_{1:T}, \cdot) \)-invariant Markov kernel.

To provide additional intuition, Appendix A shows that the particles \( Z_{1:T} \) “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 proposes \( Z_t^1 = z_t^1 \sim N(z_t^0, \frac{1}{T} I_D), \) where \( z_t^0 = x_t[1] \), which is then accepted as the new state of the Markov chain with probability

\[
\psi^1(\tilde{w}_1(z_t^1) - w_1(z_t^0)) = \frac{m_1(z_t^1)G_1(z_t^1)}{m_1(z_t^0)G_1(z_t^0)}.
\]

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 \( Z_t^1 = z_t^1 \sim N(z_t^0, \frac{1}{T} I_D), \) where \( z_t^0 = x_t[1] \), which is then accepted as the new state of the Markov chain with probability

\[
\psi^1(\tilde{w}_1(z_t^1) - w_1(z_t^0)) = 1 \wedge \frac{m_1(z_t^1)G_1(z_t^1)}{m_1(z_t^0)G_1(z_t^0)}.
\]

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 \( K_t = k_t \in [N] \) in Step 3 of Algorithm 3 with probability

\[
\psi^{k_t}(\{\tilde{v}_t(z_{t-1}^{a_{t-1}^{k_t}}, z_t^{m}, z_{t+1}^{k_{t+1}}) - \tilde{v}_t(z_{t-1}^0, z_t^0, z_{t+1}^{k_{t+1}})\}_{m=1}^N) = \frac{m_t(z_t^{k_t}, z_{t+1}^{k_{t+1}})G_t(z_t^{k_t})m_{t+1}(z_t^{k_t}, z_{t+1}^{k_{t+1}})}{\sum_{m=0}^N m_t(z_t^{a_{t-1}^{k_t}}, z_t^{m})G_t(z_t^{m})m_{t+1}(z^{m}, z_{t+1}^{k_{t+1}})}.
\]

(7)

where

\[
\tilde{v}_t(x_{t-1:t+1}) := \tilde{w}_t(x_{t-1:t}) + \log m_{t+1}(x_t, x_{t+1}).
\]
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 $\xi_T(z_{1:T}, \cdot)$-invariant Markov kernel. \(\triangleright\)

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 $\left(N + 1\right)$-state HMM mentioned above.

**Proposition 4.3.** For any $N, T, D \in \mathbb{N}$, $\pi_{T,D} = \pi_{T,D}$. \(\triangleright\)

We stress that this proposition does not require the high-dimensional regime from Assumption A1. That is, Algorithm 3 induces a valid (i.e. $\pi_{T,D}$-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. \(\triangleright\)

**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 $D \to \infty$, 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 $\partial_i^j$ denotes the $i$th derivative w.r.t. $x_t$ and with $\partial_i := \partial_i^1$, and we write $\pi_T(\varphi) := \int_{\mathbb{R}^T} \varphi(x_{1:T}) \pi_T(x_{1:T}) \, dx_{1:T}$, for any $\pi_T$-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
- $\partial_i^2 \bar{w}_t$, $\partial_i^2 \bar{w}_{t+1}$, $\partial_i \partial_{t+1} \bar{w}_{t+1}$ are Lipschitz-continuous and bounded,
- $\pi_T((\partial_i \bar{w}_t)^4)$, $\pi_T((\partial_i \bar{w}_{t+1})^4) < \infty$.

Before stating the result, we define a law $\bar{\mathbb{P}}_T^N(dv_{1:T} \times dw_{1:T} \times da_{1:T-1} \times dk_{1:T})$. 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 $V_t := V_t^{1:N}$ and $W_t := W_t^{1:N}$ 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

$$E \left[ V_t \middle| W_t \right] = \frac{1}{2} \ell_t \left[ \pi_T(\partial_i^2 \bar{w}_t) \textbf{1}_N \pi_T(\partial_i^2 \bar{w}_{t+1}) \textbf{1}_N \right] := \bar{\mu}_t|T, \Sigma_{t|T},$$

and, recalling that $\Sigma = \frac{1}{2}(I_N + a_1 N_1^T)$,

$$\text{var} \left[ V_t \middle| W_t \right] = \ell_t \left[ \pi_T((\partial_i \bar{w}_t)^2) \Sigma \pi_T((\partial_i \bar{w}_t)(\partial_i \bar{w}_{t+1})) \Sigma \pi_T((\partial_i \bar{w}_{t+1})^2) \Sigma \right] := \bar{\Sigma}_{t|T}. $$

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

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

$$I_{t|T} := \pi_T((\partial_i \log \pi_t)^2) = -\pi_T(\partial_i^2 \log \pi_T) = \text{var}[V_t^n + W_t^n]/\ell_t = -2E[V_t^n + W_t^n]/\ell_t < \infty.$$ 

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

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

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

$$\bar{a}_{T,D,x_{1:T}}^N := \| \bar{E}_T^N[I\{ (A_{1:T-1}, K_{1:T}) \in \cdot \} ] - \bar{E}_T^N[I\{ (A_{1:T-1}, K_{1:T}) \in \cdot \}] \|.$$
Then there exists a family $F_{T,D} \in \mathcal{E}_{T,D}$ with $\lim_{D \to \infty} \pi_{T,D}(F_{T,D}) = 1$ and
$$\lim_{D \to \infty} \sup_{x_{1:T} \in F_{T,D}} d_{T,D,x_{1:T}}^N = 0.$$<ref>

The following corollary shows that the acceptance rate at any time $t$ associated with Algorithm 3,
$$\bar{\alpha}_{T,D,x_{1:T}}^N(t) := \mathbb{E}_{T,D,x_{1:T}}^N \mathbb{I}\{K_t \neq 0\},$$
converges to a strictly positive limit
$$\bar{\alpha}_T(t) := \mathbb{E}_T^N \mathbb{I}\{K_t \neq 0\}.$$ 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 $\bar{\alpha}_T(t) > 0$. Furthermore, with $F_{T,D} \in \mathcal{E}_{T,D}$ as in Proposition 4.5:
$$\lim_{D \to \infty} \sup_{x_{1:T} \in F_{T,D}} |\bar{\alpha}_{T,D,x_{1:T}}^N(t) - \bar{\alpha}_T^N(t)| = 0.$$<ref>

**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):
$$\bar{\alpha}_1^1(1) = \begin{cases} \mathbb{E}[\psi^1(V_1^1)] = \mathbb{E}\left[\frac{\exp\{V_1^1\}}{1 + \exp\{V_1^1\}}\right], & \text{without forced-move}, \\ \mathbb{E}[\phi^1(V_1^1)] = \mathbb{E}[1 \wedge \exp\{V_1^1\}] = 2\Phi\left(-\frac{\sqrt{\ell_{1:T}}}{2}\right), & \text{with forced-move}, \end{cases}$$
where $\Phi$ is standard-normal cumulative distribution function. □

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 et al., 2009), which (for the time-$t$ component in Algorithm 3) is given by
$$\text{ESJD}_{T,D}^N(t) := \mathbb{E}[\|X_t[l+1] - X_t[l]\|_2^2],$$
where $\| \cdot \|_2$ denotes the Euclidean norm and where $X_{1:T}[l]$ is the $l$th state of the Markov chain with transition kernel $\tilde{P}_{T,D}^N$ 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],
$$\lim_{D \to \infty} |\text{ESJD}_{T,D}^N(t) - \ell_t\bar{\alpha}_T(t)| = 0.$$<ref>
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, $\alpha_{T,D,x,t}^N(t)$, can be bounded away from zero as $T \to \infty$ by scaling the number of particles appropriately. To be more specific: Without backward sampling, $\alpha_{T,D,x,t}^N(t)$ 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, $\alpha_{T,D,x,t}^N(t)$ 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 $N \geq CT$:
\[
\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \alpha_T^N(t) \geq \exp\left(-\frac{\mathcal{I}(\ell)}{C}\right) > 0.
\]

2. With backward sampling, for any $N \in \mathbb{N}$:
\[
\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \alpha_T^N(t) \geq \left(1 + \frac{\mathcal{I}(\ell)}{N}\right)^{-1} > 0.
\]

**REMARK 4.9.** Proposition 4.7 shows that $\inf_{T \in \mathbb{N}} \inf_{t \in [T]} \mathcal{ESJD}_{T,D}^N(t) > 0$ (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, $\hat{\mathbf{P}}_{T,D}^N = \hat{\mathbf{P}}_{T,D}^N$. 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 $\varphi$ denote a Lebesgue density of the standard normal distribution. Let $y_t = (y_{t,d})_{d \in [D]} \in \mathbb{R}^D$ 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
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 $\pi_{T,D}$-averaged acceptance rates as a function of the time index $t$. More precisely, for $X_{1:T} \sim \pi_{T,D}$, it shows:

1. first column: $E[\alpha_{N,T,D}^{X_{1:T}}(t)]$;
2. second column: $E[\bar{\alpha}_{N,T,D}^{X_{1:T}}(t)]$.

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: $\text{ESJD}_{N,T,D}^{X_{1:T}}(t) := E[\|X_{t}[l + 1] - X_{t}[l]\|^2]$;
2. second column: $\bar{\text{ESJD}}_{N,T,D}^{X_{1:T}}(t) := E[\|\bar{X}_{t}[l + 1] - \bar{X}_{t}[l]\|^2]$,

where $X_{1:T}[l]$ is the $l$th state of a stationary Markov chain with transition kernels $P_{N,T,D}^{X_{1:T}}$ and $\bar{X}_{1:T}[l]$ is the $l$th state of a stationary Markov chain with transition kernels $P_{N,T,D}^{\bar{X}_{1:T}}$. The first column illustrates that in high dimensions, the ESJD of the i-CSMC algorithm vanishes in
high dimensions. In contrast, the ESJD of the i-RW-CSMC algorithm converges to \( \ell_t \alpha_t^N(t) > 0 \) (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 \( X_t[l] = X_{t,1:D}[l] \) and \( \bar{X}_t[l] = \bar{X}_{t,1:D}[l] \), it shows:

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

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 \( \ell_t \). We propose a simple adaptation strategy to specify \( \ell_t[g] \), the value of \( \ell_t \) at the \( g \)th iteration of the algorithm. Let \( \alpha_t[g] \) denote the average acceptance rate at time \( t \) up to the
\begin{align*}
el_t[g] := \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}
\end{align*}

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 $\alpha$ to be an increasing function in $N$. Specifically, we use $\alpha := 1 - (N + 1)^{-\beta}$ 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 $\ell_t$. 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 $O(T)$ 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 $O(e^T)$ complexity, i.e. a curse of dimension in $T$.

The i-CSMC algorithm can be combined with backward sampling to reduce the complexity to $O(1)$ (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 $O(e^D)$) 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 $O(D)$ 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 $O(D)$ to $O(1)$ 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 $\ell_t$ 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

---

\(^1\)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.
relaxed, e.g. by allowing for some of the $D$ components of the target distribution to be differently scaled or by allowing for dependence between some of the $D$ components (see, e.g., Sherlock et al., 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 $\ell^*_t$ 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


1. In Subsection A.1, we show that the distribution $Z_{1:T} = k_{1:T} \in [N]_0^T$ of the new reference path $X'_{1:T} = (Z_1^K, \ldots, Z_T^K)$ 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:

$$\xi_T(z_{1:T}, \{k_{1:T}\}) = \frac{\pi_{T,D}(z_{1:T}^1, \ldots, z_{1:T}^K)}{\sum_{l_{1:T} \in [N]_0^T} \pi_{T,D}(z_{1:T}^1, \ldots, z_{1:T}^K)}.$$  

Recall that conditional on the set of particles $Z_{1:T} = z_{1:T}$:

1. the RW-EHMM Algorithm draws $K_{1:T} \sim \xi_T(z_{1:T}, \cdot)$;
2. the i-RW-CSMC Algorithm draws $K_{1:T}$ according to a $\xi_T(z_{1:T}, \cdot)$-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 $\xi_T(z_{1:T}, \cdot)$ 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 $O(N^2T)$ 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 $O(NT)$ 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 same value of the set of particles $Z_{1:T} = z_{1:T}$. Hence, we will sometimes drop the dependence on $z_{1:T}$ from the notation. For any $k_{1:T} \in [N]_0^T$, and any $t \in [T]$, set

$$f_t(k_t \mid k_{t-1}) := \frac{m_t(z_{t-1}^{k_{t-1}}, z_t^{k_t})G_t(z_t^{k_t})}{\sum_{n=0}^N m_t(z_{t-1}^{k_{t-1}}, z_t^n)G_t(z_t^n)},$$

$$g_t(\tilde{y}_t \mid k_t) := \begin{cases} \left[ \sum_{n=0}^N m_1(z_0^n)G_1(z_0^n) \right] \sum_{n=0}^N m_2(z_1^n, z_2^n)G_2(z_2^n), & \text{if } t = 1, \\
\sum_{n=0}^N m_{t+1}(z_t^{k_t}, z_{t+1}^n)G_{t+1}(z_{t+1}^n), & \text{if } t > 1,
\end{cases}$$

with our usual convention that any quantity with time subscript 0 is to be ignored, i.e. $f_1(\cdot \mid k_0) \equiv f_1(\cdot)$. Then clearly, $f_t(\cdot \mid k_{t-1})$ 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

$$p(k_{1:T}, \tilde{y}_{1:T-1}) = \prod_{t=1}^{T} f_t(k_t|k_{t-1}) g_t(\tilde{y}_t|k_t)$$

$$= \prod_{t=1}^{T} m_t(z_{t-1}^{k_t-1}, z_t^{k_t}) G_t(z_t^{k_t}) \propto \pi_{T,D}(z_1^{k_1}, \ldots, z_T^{k_T})$$

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

$$p(k_{1:T} | \tilde{y}_{1:T-1}) = \frac{\pi_{T,D}(z_1^{k_1}, \ldots, z_T^{k_T})}{\sum_{l:T \in [N]_0} \pi_{T,D}(z_1^l, \ldots, z_T^l)} = \xi_T(z_{1:T}, \{k_{1:T}\}).$$

### 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 $O(N^2T)$ 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 $w_0^n = 1$):

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

$$w_t^n = p(k_t = n, \tilde{y}_{t-1}|\tilde{y}_{1:t-2})$$

$$= \sum_{m=0}^{N} p(k_{t-1} = m|\tilde{y}_{t-2}) g_{t-1}(\tilde{y}_{t-1}|m) f_t(n|m)$$

$$= \sum_{k_{t-1} \in [N]_0} \sum_{l=0}^{N} w_{t-1}^m m_t(z_{t-1}^m, z_t^n) G_t(z_t^n).$$

**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

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

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

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

$$= \frac{w_t^{k_t} m_{t+1}(z_{t+1}^{k_{t+1}}, z_t^{k_t})}{\sum_{n=0}^{N} w_t^n m_{t+1}(z_{t+1}^n, z_t^n)}.$$
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 $O(NT)$ operations) can be viewed as first running a CSMC algorithm targeting 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 $\delta_{(0,1,...,N)}$ 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 targeting the finite-state HMM at time $t$ can then be shown to be given by

$$w^n_t := g_t(\tilde{y}_t|a^n_{t−1}) f_t(n|a^n_{t−1}) = m_t(z^n_{t−1}, z^n_t) G_t(z^n_t).$$

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 targeting the finite-state HMM, then this gives the backward-sampling probabilities from (7):

$$\frac{w^k_t g_t(y_t|k_t)f_{t+1}(k_{t+1}|k_t)}{\sum_{m=0}^N w^m_t g_t(y_t|m)f_{t+1}(k_{t+1}|m)} = \frac{m_t(z^{a^n_{t−1}}_{t−1}, z^k_t) G_{t+1}(z^k_t) m_{t+1}(z^k_t, z^{k_{t+1}}_{t+1})}{\sum_{m=0}^N m_t(z^{a^n_{t−1}}_{t−1}, z^m_t) G_{t}(z^m_t) m_{t+1}(z^m_t, z^{k_{t+1}}_{t+1})}.$$ 

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 $Z^0_t := x_t$ as well as $A^0_{t−1} := 0$. 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 $A^j_{t−1} := a^j_{t−1} := j_{t−1}$ as well as $Z^j_t := z^j_t := x_t$.

Conditional on $X_{1:T} = x_{1:T} = x_{1:T}[\ell]$, the joint law of all random variables $(J_{1:T}, Z_{1:T}, A_{1:T−1}, K_{1:T}, X'_{1:T})$ generated by this slightly generalised version of Algorithm 1 may be written as

$$\pi_{T,D,X_{1:T}}^{N,*}(dj_{1:T} \times dz_{1:T} \times da_{1:T−1} \times dk_{1:T} \times dx_{1:T})$$

:= $\text{Unif}_{[N]_0^T}(dj_{1:T}) \delta_{x_{1:T}}(dz_1^j \times \cdots \times dz_T^j) \prod_{j=0}^N M_1(dz^n_1) \prod_{j=0}^T (R_{t−1,D}^{N}(z_{t−1}, da^n_{t−1}) \prod_{n=0}^N M_t(z^n_{t−1}, dz^n_t))$ 

$$\times \prod_{t=1}^T \delta_{j_{t−1}, (da^n_{t−1})} \prod_{n=0}^N R^{N}_{t,D}(z_{t−1}, da^n_{t−1}) M_t(z^n_{t−1}, dz^n_t) \times R^{N}_{T,D}(z_T, dk_T)$$
sometimes work with nor the forced-move extension) we can readily check that
\[ \pi := \frac{\prod_{m=0}^{N} G_t(z_t^m)}{\sum_{m=0}^{N} G_t(z_t^m)}. \]

When using the forced-move extension, replace \( R_{t,D}^N(z_t, \{n\}) \) at time \( t = T \) in (8) by
\[
\begin{align*}
&\frac{\prod_{m=0}^{N} G_T(z_T^m)}{\sum_{m=0}^{N} G_T(z_T^m)} - G_T(z_T^k) \land G_T(z_T^j), \\
&1 - \sum_{l=0}^{N} G_T(z_T^l) - G_T(z_T^k) \land G_T(z_T^j),
\end{align*}
\]

for \( k_T \neq j_T \),
\[
\begin{align*}
&\frac{\prod_{m=0}^{N} G_T(z_T^m)}{\sum_{m=0}^{N} G_T(z_T^m)} G_T(z_T^k), \\
&1 - \sum_{l=0}^{N} G_T(z_T^l) G_T(z_T^j),
\end{align*}
\]

for \( k_T = j_T \).

Here, we have defined the following quantities.

- **Resampling kernels.** For any \( n \in [N]_0 \) and any \( t \in [T] \),
\[
R_{t,D}^N(z_t, \{n\}) := \Psi^n(\{w_t(z_t^m) - w_t(z_t^0)\}) = \frac{G_t(z_t^n)}{\sum_{m=0}^{N} G_t(z_t^m)}.
\]

- **Backward kernels.** For any \( t \in [T-1] \),
\[
B_{t,D}^N((z_t, z_{t+1}^k), \{n\}) := \Psi^n(\{v_t(z_t^m, z_{t+1}^k) - v_t(z_t^0, z_{t+1}^0)\}) = \frac{G_t(z_t^n)}{\sum_{m=0}^{N} G_t(z_t^m)} m_{t+1}(z_t^n, z_{t+1}^k).
\]

From this definition, we can recover the joint law of all random variables \( (Z_{1:T}, A_{1:T-1}, K_{1:T}, X'_{1:T}) \) generated in Steps 1–4 of Algorithm 1 by conditioning on the event \( \{J_1 = 0, \ldots, J_T = 0\} \), i.e.
\[
\mathbb{P}_{T,D,X_{1:T}}^{N} := \mathbb{P}_{T,D,X_{1:T}}^{N,*, \pi}(\cdot | J_1 = 0, \ldots, J_T = 0).
\]

Let \( \mathbb{E}_{T,D,X_{1:T}}^{N,*} \) denote expectation w.r.t. \( \mathbb{P}_{T,D,X_{1:T}}^{N,*} \). In the remainder of this section, we will sometimes work with \( \mathbb{P}_{T,D,X_{1:T}}^{N,*} \) rather than with \( \mathbb{P}_{T,D,X_{1:T}}^{N} \). This is justified because both versions of the i-CSMC algorithm induce the same Markov kernel:
\[
\mathbb{E}_{T,D,X_{1:T}}^{N,*}[\{X'_{1:T} \in dX'_{1:T}\}] = \mathbb{P}_{T,D,X_{1:T}}^{N,*}[\{X'_{1:T} \in dX'_{1:T}\}] = \mathbb{P}_{T,D}^{N}[X_{1:T} \in dX'_{1:T}].
\]

**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
\[
\pi_{T,D}(dx_{1:T}) = \mathbb{P}_{T,D,X_{1:T}}^{N,*}[dk_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx_{1:T}] \\
= \pi_{T,D}(dx'_{1:T}) \mathbb{P}_{T,D,X_{1:T}}^{N,*}[dk_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dj_{1:T} \times dx_{1:T}],
\]
i.e. (9) admits \( \pi_{T,D}(dx'_{1:T}) \) as a marginal.
For the backward-sampling extension, let \( \tilde{P}_{T,D}^N \) be the same as \( P_{T,D}^N \) (without backward sampling) except that the terms \( \delta_{j_{i-1}}(d_{i-1}) \) in (8) are replaced by \( B_{T-1}^N((z_{t-1}, z_t^i), \{a_{i-1}^j\}) \). Then
\[
\pi_{T,D}(dx_{1:T}) \tilde{P}_{T,D}^N(dx_{1:T}) = \pi_{T,D}(dx_{1:T}) P_{T,D}^N(dx_{1:T})
\]
That is, (10) again admits \( \pi_{T,D}(dx_{1:T}) \) as a marginal. Incidentally, \( P_{T,D}^N \) 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 \( (X_{1:T}, J_{1:T}, Z_{1:T}, A_{1:T}) \) invariant. \( \square \)

**B.3. Verification of Assumptions A2 and A3 in a linear-Gaussian state-space model.**
Consider a state-space model with \( D \)-dimensional observations \( y_t = y_{t:1:D} \in \mathbb{R}^D \) and
\[
H_t(x_t, dy_t) := \prod_{d=1}^{D} N(dy_t, x_t, 1),
M_t(x_{t-1}, dx_t) := \prod_{d=1}^{D} N(dx_t, x_{t-1}, 1).
\]
Let \( \varphi \) be a density function of a univariate standard normal distribution. Then this model satisfies A1 with
\[
G_t(x_{t:d}) = \varphi(y_{t:d} - x_{t:d}),
m_t(x_{t-1:d}, x_{t:d}) = \varphi(x_{t:d} - x_{t-1:d}).
\]
To simplify the notation and calculations, we hereafter drop the subscript \( d \), take \( y_1 = \ldots, y_T = 0 \) and assume as initial density \( m_1(x_1) = \varphi(x_1/\sigma^2 + 1)/\sqrt{\sigma^2 + 1} \), where \( \sigma^2 := (\sqrt{5} - 1)/2 \). Standard Kalman-filtering and Kalman-smoothing recursions then give
\[
\mathbb{P}(X_t \in dx_t | Y_{1:t} = y_{1:t}) = N(dx_t; 0, \sigma^2) \text{ as well as } \pi_T(dx_{1:T}) = \mathbb{P}(X_{1:T} \in dx_{1:T} | Y_{1:T} = y_{1:T}) = N(dx_{1:T}; 0_t, C) \text{ with } [C]_{s,t} = u^{t-s} \sigma^2_{s:t}, \text{ where } u := \sigma^2/(\sigma^2 + 1) \text{ and }
\]
\[
\sigma^2_t := \frac{u - (u^2)^{T-t}}{1 - u^2} \mathbb{I}\{t < T\} + (u^2)^{T-t} \sigma^2.
\]
Tedious but simple algebra then shows that A2 and A3 hold because
\[
\mathbb{P}_{T} > \mathbb{E}_{T} = r_{T|T} = \begin{cases} \frac{1}{2} (\log(\sigma^2 + 2) - \sigma^2), & \text{if } T = 1, \\ \frac{1}{2} (\log(2) + [\sigma^2(u^2 - 2) + u]/2), & \text{if } T > 1 \end{cases} > 0.15.
\]

**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 \( 0 \) denote a vector of zeros of appropriate length.
\[
\tilde{d}_{T,D}^N_{T,D} \leq \sum_{t=1}^{T} \sum_{n=1}^{N} \| \mathbb{E}_{T,D}^N_{T,D} [\mathbb{I}\{A_t^0 \in \cdot\} | A_{1:t-1} = 0] - \delta_t(\cdot) \|
\]
\[
+ \| \mathbb{E}_{T,D}^N_{T,D} [\mathbb{I}\{K_T \in \cdot\} | A_{1:T-1} = 0] - \delta_t(\cdot) \|
\]
\[
+ \sum_{t=1}^{T-1} \| \mathbb{E}_{T,D}^N_{T,D} [\mathbb{I}\{K_t \in \cdot\} | (A_{1:T-1}, K_{t+1:T}) = 0] - \delta_t(\cdot) \|.
\]
First, we consider the case without backward sampling. We write
\[ R_{t,T}(x_{t-1:d}) := \frac{1}{D} \sum_{d=1}^{D} \log M_t(G_t(x_{t-1:d}) - \log G_t(x_{t,d})) + r_{t|T} \]
\[ = \frac{1}{D} \sum_{d=1}^{D} \log M_t(G_t(x_{t-1:d}) - \mathbb{E}[\log M_t(G_t(X_{t-1})]) + \frac{1}{D} \sum_{d=1}^{D} \mathbb{E}[\log G_t(X_t)] - \log G_t(x_{t,d}). \]

For some \( \eta \in (0, 1/2) \), we set
\[ F_{T,D} := \{ x_{1:T} \in E_{T,D} : \forall t \in [T] : |R_{t|T,D}(x_{t-1:d})| < D^{-\eta} \}. \]

Since \( \eta < 1/2 \) implies that \( D^{\eta} \sqrt{2D\log \log D} = O(D) \), the law of the iterated logarithm then implies that \( \lim_{D \to \infty} \pi_{T,D}(F_{T,D}) = 1. \)

We can now turn to the terms in the decomposition (11). Let \( (x_{1:T,D})_{D \geq 1} \) be some sequence in \( (E_{T,D})_{D \geq 1} \), i.e. \( x_{t,D} = x_{t,1:D,M} \in \mathbb{R}^D \), for any \( D \geq 1 \). For any \( n \in [N] \) and \( t \in [T] \), let \( Z_{t,D}^n \sim M_t(x_{t-1,D,M}) \) and \( Z_{t,D}^0 := x_{t,D} \). For any \( t \in [T] \) (with \( A^n_t \) replaced by \( K_t \) if \( t = T \)), we have that
\[ |\mathbb{E}_{T,D,x_{1:T,D}}^N \mathbb{I}\{A^n_t \in [N]\}|A_1:t-1 = 0| - \delta_0([N]) = \mathbb{E}_{T,D,x_{1:T,D}}^N \left[ \sum_{n=1}^{N} \exp\left\{ \mathbb{w}_t(Z_{t,D}^n) - \mathbb{w}_t(Z_{t,D}^0) \right\} \right]_{A_1:t-1 = 0}. \]

For any \( n \in [N] \) and \( t \in [T] \), let \( Z_{t,D}^n \sim M_t(x_{t-1,D,M}) \) and \( Z_{t,D}^0 := x_{t,D} \). To complete the proof, it then suffices to show that
\[ \sum_{n=1}^{N} \exp\left\{ \mathbb{w}_t(Z_{t,D}^n) - \mathbb{w}_t(Z_{t,D}^0) \right\} \to_p 0, \]
as \( D \to \infty \), whenever \( \log N = o(D) \). By Markov’s inequality, for any \( \varepsilon > 0 \),
\[ \mathbb{P}\{ \sum_{n=1}^{N} \exp\left\{ \mathbb{w}_t(Z_{t,D}^n) - \mathbb{w}_t(Z_{t,D}^0) \right\} > \varepsilon \} \]
\[ \leq N \varepsilon^{-1} \exp\left\{ -D r_{t|T} + D |R_{t|T,D}(x_{t-1:d})| \right\} \]
\[ \leq N \varepsilon^{-1} \exp\left\{ -D r_{t|T} + D^{1-\eta} \right\}, \]
and \( r_{t|T} \geq C_T > 0 \) by A2. This completes the proof in the case without backward sampling.

We now consider the case with backward sampling. We write
\[ B_{t|T,D}(x_{t-1:t+1}) := \frac{1}{D} \sum_{d=1}^{D} \log M_t(G_t(X_{t-1:d}, \cdot)) (x_{t-1:d}) - v_t(x_{t:t+1:d}) + b_{t|T} \]
\[ = \frac{1}{D} \sum_{d=1}^{D} \log M_t(G_t(X_{t-1:d}, \cdot)) (x_{t-1:d}) - \mathbb{E}[\log M_t(G_t(X_{t-1:d}, \cdot)) (X_{t-1})] + \frac{1}{D} \sum_{d=1}^{D} \mathbb{E}[\log G_t(X_t)] - \log G_t(x_{t,d}). \]

with convention \( B_{T|T,D} = \mathcal{R}_{T|T,D} \). For some \( \eta \in (0, 1/2) \), we set
\[ F_{T,D} := \{ x_{1:T} \in E_{T,D} : \forall t \in [T] : |R_{t|T,D}(x_{t-1:d})| \vee |B_{t|T,D}(x_{t-1:t+1})| < D^{-\eta} \}. \]

Since \( \eta < 1/2 \) implies that \( D^{\eta} \sqrt{2D\log \log D} = O(D) \), the law of the iterated logarithm then again implies that \( \lim_{D \to \infty} \pi_{T,D}(F_{T,D}) = 1. \)

Again we consider the decomposition (11). We then have that
\[ |\mathbb{E}_{T,D,x_{1:T,D}}^N \mathbb{I}\{K_t \in [N]\}|(A_1:t-1, K_{t+1} = 0) - \delta_0([N]) = \mathbb{E}_{T,D,x_{1:T,D}}^N \left[ \sum_{n=1}^{N} \exp\left\{ \mathbb{w}_t(Z_{t,D}^n, Z_{t+1,D}^0) - \mathbb{w}_t(Z_{t,D}^0, Z_{t+1,D}^0) \right\} \right]_{A_1:t-1 = 0}. \]
To complete the proof, it suffices to show that
\[ \sum_{n=1}^{N} \nu_t(Z_{i,D}^{n}, Z_{i+1,D}^{n}) - \nu_t(Z_{i,D}^{0}, Z_{i+1,D}^{0}) \to 0, \]
as \( D \to \infty \), whenever \( \log N = o(D) \). By Markov’s inequality, for any \( \varepsilon > 0 \),
\[ \mathbb{P} \left( \sum_{n=1}^{N} \exp \left\{ \nu_t(Z_{i,D}^{n}, Z_{i+1,D}^{n}) - \nu_t(Z_{i,D}^{0}, Z_{i+1,D}^{0}) > \varepsilon \right\} \right) \leq N \varepsilon^{-1} \exp \{ -DB |T| + D|B(t-1:t+1,D)| \} \leq N \varepsilon^{-1} \exp \{ -DB |T| + D^{1-\eta} \}, \]
and \( b_{t|T} \geq b_{tT} > 0 \) by A3. This completes the proof in the case with backward sampling. \( \square \)

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 \( Z_{i}^{0} := z_{i} \). 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 \([N]\) and then sets \( Z_{i}^{j_{t}} := z_{i} \).

Conditional on \( X_{1:T}^{1} = x_{1:T} = x_{T}^{1}[l] \), the joint law of all random variables \( (J_{1:T}, Z_{1:T}, K_{1:T}, X_{1:T}') \)
generated by this slightly generalised version of Algorithm 2 may be written as
\[ \tilde{P}_{N,*}^{T,D,X_{1:T}}(dJ_{1:T} \times dz_{1:T} \times dk_{1:T} \times dx_{1:T}') := \text{Unif}_{[N]}(dJ_{1:T}) \delta_{x_{1:T}}(dz_{1:T} \times \cdots \times dz_{T}') \prod_{t=1}^{T} S_{t,D}(z_{t}^{j_{t}}, dz_{t}^{-j_{t}}) \times \xi_{T}(z_{1:T}, dk_{1:T}) \delta_{k_{1:T}}(d_{k_{1:T}}). \]
Here, we have defined \( z_{t}^{-n} := (z_{t}^{0}, \ldots, z_{t}^{n-1}, z_{t}^{n+1}, \ldots, z_{t}^{N}) \) as well as the following quantities.

- **Proposal kernels.** For any \( t \in [T] \) and any \( n \in [N] \),
\[ S_{t,D}^{N}(z_{t}^{n}, dz_{t}^{-n}) := \prod_{d=1}^{D} \text{N}(dz_{t,d}^{-n}; z_{t,d}, \Sigma), \]
where \( z_{t,d}^{-n} := (z_{t,d}^{0}, \ldots, z_{t,d}^{n-1}, z_{t,d}^{n+1}, \ldots, z_{t,d}^{N}) \).

- **Selection probability.**
\[ \xi_{T}(z_{1:T}, \{k_{1:T}\}) := \frac{\pi_{T,D}(z_{1:T}, \ldots, z_{T}')}{\sum_{l_{1:T} \in [N]_{D}} \pi_{T,D}(z_{1:T}, \ldots, z_{T}')}. \]

From this definition, we can recover the joint law of all random variables \( (Z_{1:T}, K_{1:T}, X_{1:T}') \) generated in Steps 1–3 of Algorithm 2 by conditioning on the event \( \{J_{1} = \ldots, J_{T} = 0\} \), i.e.
\[ \tilde{P}_{N,*}^{T,D,X_{1:T}}(dJ_{1:T} \times \{k_{1:T}\}) := \tilde{P}_{N,*}^{T,D,X_{1:T}}(dJ_{1:T} \times \{k_{1:T}\}) \mathbb{I}(J_{1} = \ldots, J_{T} = 0). \]
Let \( \tilde{P}_{N,*}^{T,D,X_{1:T}} \) denote expectation w.r.t. \( \tilde{P}_{N,*}^{T,D,X_{1:T}} \). In the remainder of this section, we will sometimes work with \( \tilde{P}_{N,*}^{T,D,X_{1:T}} \) rather than with \( \tilde{P}_{N,*}^{T,D,X_{1:T}} \). This is justified because both versions of the RW-EHMM algorithm induce the same Markov kernel:
\[ \tilde{P}_{N,*}^{T,D,X_{1:T}}(dX_{1:T}^{'} \times dx_{1:T}) = \tilde{P}_{N,*}^{T,D,X_{1:T}}(dX_{1:T}^{'} \times dx_{1:T}) = \tilde{P}_{N,*}^{T,D}(x_{1:T}^{'} \times dx_{1:T}). \]

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

$$\lambda(\text{d}z_t^j)S^N_{t,D}(z_t^j, \text{d}z_t^j) = \lambda(\text{d}z_t^k)S^N_{t,D}(z_t^k, \text{d}z_t^k),$$

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

We can then readily check that

$$\pi_{T,D}(\text{d}x_{1:T}) \bar{\Psi}^N_{T,D, x_{1:T}}(dj_{1:T} \times dz_{1:T} \times dk_{1:T} \times dx'_{1:T})$$

(12) admits \( \pi_{T,D}(\text{d}x'_{1:T}) \) as a marginal. \( \square \)

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

$$\tilde{\alpha}_T^N(t) = \sum_{n \in [N]} \tilde{\mathbb{E}}_T \left[ \mathbb{P}^N_{\{V^N_t\}_{t=1}} \right] \geq \left(1 + \frac{\exp(\ell_t I(t_n|T))}{N}\right)^{-1},$$

where the penultimate inequality follows by Lemma D.6 in Appendix D.6. \( \square \)


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

$$\mathbb{E} \left[ \left| \mathbb{S}J^N_{T,D}(t) - \ell_t \hat{\alpha}_T^N(t) \right| \right]$$

$$= \left| \int_{\mathbf{E}_{T,D}} \pi_{T,D}(\text{d}x_{1:T}) \mathbb{E}_T \left[ \mathbb{E}_T^N \left[ \left| \mathbf{X}_t' - \mathbf{x}_t \right| \right] - \ell_t \hat{\alpha}_T^N(t) \right] \right|$$

$$\leq \ell_t \sup_{\mathbf{x}_{1:T} \in \mathbf{F}_{T,D}} \left| \mathbb{E}_T^N \left[ \left| \mathbf{X}_t' - \mathbf{x}_t \right| - \ell_t \hat{\alpha}_T^N(t) \right] \right|$$

$$+ \sup_{\mathbf{x}_{1:T} \in \mathbf{E}_{T,D}} \left| \mathbb{E}_T \left[ \mathbb{E}_T^N \left[ \left| \mathbf{X}_t' - \mathbf{x}_t \right| - \ell_t \hat{\alpha}_T^N(t) \right] \right] \right|$$

(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 \( U_{l,d} = D^{\frac{1}{2}} \ell_t^{-1/2} (Z_{l,d} - x_{l,d}) \) as in Algorithm 2, then \( \tilde{U}_{l,D} := D^{-1} \sum_{d=1}^D (U_{l,d}^n)^2 \to \mathbb{P} 1 \), as \( D \to \infty \), and hence \( (\tilde{U}_{l,D} - 1) \to \mathbb{P} 0 \) and \( |\tilde{U}_{l,D}| \to \mathbb{P} 1 \) by the continuous mapping theorem. Furthermore, \( (|\tilde{U}_{l,D} - 1|)_{D \geq 1} \) and \( (|\tilde{U}_{l,D}|)_{D \geq 1} \) are uniformly integrable. Thus, for the second term, for any \( \mathbf{x}_{1:T} \in \mathbf{E}_{T,D} \),

$$\left| \mathbb{E}_T^N \left[ \left| \mathbf{X}_t' - \mathbf{x}_t \right| - \ell_t \hat{\alpha}_T^N(t) \right] \right|$$

$$\leq \ell_t N \left| \mathbb{E} \left[ |\tilde{U}_{l,D} - 1| \right] \right| \to 0.$$

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

$$\left| \mathbb{E}_T^N \left[ \left| \mathbf{X}_t' - \mathbf{x}_t \right| \right] \right|$$

$$\leq \ell_t N \left| \mathbb{E} \left[ |\tilde{U}_{l,D} - 1| \right] \right| \to 1,$$

and the third term therefore converges to zero since \( \pi(\mathbf{E}_{l,D} \setminus \mathbf{F}_{T,D}) \to 0 \). \( \square \)
**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 $\mathbb{Z}^0_t := z^0_t := x_t$ as well as $A^0_{t-1} = a^0_{t-1} := 0$. 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 $A^j_{t-1} = a^j_{t-1} := j - 1$ as well as $Z^j_t := z^j_t := x_t$.

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

\[
\mathbb{P}_{T,D,x_{1:T}}(\text{d}j_{1:T} \times \text{d}z_{1:T} \times \text{d}a_{1:T-1} \times \text{d}k_{1:T} \times \text{d}x'_{1:T}) := \text{Unif}_{[N]_0^T}(\text{d}j_{1:T}) \delta_{x_{1:T}}(\text{d}z_{1:T} \times \cdots \times \text{d}z_T^T) \left[ \prod_{t=1}^T S_{t,D}^N(z^t_j, \text{d}z^t_j) \right] \times \left[ \prod_{t=2}^T \delta_{j_{t-1}}(\text{d}a^j_{t-1}) \left( \prod_{n=0}^N \tilde{R}_{t,D}^N((z_{t-1:t}, a_{t-1}), \text{d}a^j_{t-1}) \right) \times \tilde{R}_{T,D}^N((z_{T-1:T}, a_{T-1}), \text{d}k_T) \times \left\{ \prod_{t=1}^{T-1} \tilde{B}_{t,D}^N((z_{t-1:t}, a_{t-1}, z^{k_{t+1}}_{t+1}), \text{d}k_t) \right\} \right] \quad \text{[without backward sampling]} \times \delta_{(z^1_t, \ldots, z^T_T)}(\text{d}x'_{1:T}).
\]

Here, we have defined $z_t^{-n} := (z^0_t, \ldots, z^{n-1}_t, z^{n+1}_t, \ldots, z^T_t)$ 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,

  \[
  S_{t,D}^N(z^n_t, \text{d}z^{-n}_t) := \prod_{d=1}^D \mathbb{N}(\text{d}z^{-n}_{t,d}, z^{n}_{t,d} 1_N, \Sigma),
  \]

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

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

  \[
  \tilde{R}_{t,D}^N((z_{t-1:t}, a_{t-1}), \{n\}) := \Psi^N(\{\tilde{\omega}_t(z^a_{t-1}, z^n_m) - \tilde{\omega}_t(z^a_{t-1}, z^0_m)\}_{m=1}^N) = \frac{m_t(z^a_{t-1}, z^n_0)G_t(z^n_0)}{\sum_{m=0}^N m_t(z^a_{t-1}, z^m_0)G_t(z^m_0)},
  \]

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

\[
\begin{cases}
\frac{\mathbf{h}^{k_T}}{\sum_{m=0}^N \mathbf{h}^{m} - \mathbf{h}^{k_T} \wedge \mathbf{h}^{j_T}}, & \text{if } k_T \neq j_T, \\
1 - \sum_{l=0}^{N} \frac{1}{l \neq j_T} \frac{\mathbf{h}^l}{\sum_{m=0}^N \mathbf{h}^{m} - \mathbf{h}^{l} \wedge \mathbf{h}^{j_T}}, & \text{if } k_T = j_T.
\end{cases}
\]
where we have defined the shorthand $h^n := m_T(z_{T-1}^{a_{T-1}^n}, z_T^n)G_T(z_T^n)$.

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

$$
\tilde{B}_{t,D}((z_{t-1}, a_{t-1}, z_{t+1}^k), \{n\}) := P_n((\tilde{v}(z_{t-1}^{a_{t-1}^m}, z_{t+1}^k) - \tilde{v}(z_{t-1}^{a_{t-1}^n}, z_{t+1}^k))_{m=1}^N
\sum_{m=0}^N m_t(z_{t-1}^{a_{t-1}^m}, z_{t+1}^k)G_t(z_{t+1}^m)_{m=1}^N.
$$

From this definition, we can recover the joint law of all random variables $(Z_{1:T}, A_{1:T-1}, K_{1:T}, X'_1:T)$ generated in Steps 1–4 of Algorithm 3 by conditioning on the event $\{J_1 = 0, \ldots, J_T = 0\}$, i.e.

$$
\mathbb{P}_{T,D,X_1:T} := \mathbb{P}_{T,D,X_1:T}^N(\cdot | J_1 = 0, \ldots, J_T = 0).
$$

Let $\mathbb{E}_{T,D,X_1:T}^N$ denote expectation w.r.t. $\mathbb{P}_{T,D,X_1:T}^N$. In the remainder of this section, we will sometimes work with $\mathbb{E}_{T,D,X_1:T}^N$ rather than with $\mathbb{P}_{T,D,X_1:T}^N$. This is justified because both versions of the i-RW-CSMC algorithm induce the same Markov kernel:

$$
\mathbb{E}_{T,D,X_1:T}^N(\mathbb{E}_{T,D,X_1:T}^N(1 \{X'_1:T \in dx'_1:T\})) = \mathbb{E}_{T,D,X_1:T}^N(\mathbb{E}_{T,D,X_1:T}^N(1 \{X'_1:T \in dx'_1:T\})) = \mathbb{P}_{T,D,X_1:T}^N(1 \{X'_1:T \in dx'_1:T\}).
$$

**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

$$
\lambda(dz_{t}^j)S_{t,D}^N(z_{t}^j, dz_{t}^j) = \lambda(dz_{t}^k)S_{t,D}^N(z_{t}^k, dz_{t}^{-k}),
$$

for any $j, k \in [N]$ where $z_t^{-n} := (z_t^0, \ldots, z_t^{n-1}, z_t^{n+1}, \ldots, z_t^N)$ and where $\lambda$ 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

$$
\pi_{T,D}(dx_{1:T})|P_{T,D,X_1:T}^N(\{dj_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx'_{1:T}\})
= \pi_{T,D}(dx'_{1:T})|P_{T,D,X_1:T}^N(\{dj_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx'_{1:T}\}),
$$

i.e. (15) admits $\pi_{T,D}(dx'_{1:T})$ as a marginal.

For the backward-sampling extension, let $\tilde{P}_{T,D,X_1:T}^N$ be the same as $\tilde{P}_{T,D,X_1:T}^N$ (without backward sampling) except that the terms $\delta_{j_{t-1}}(da_{t-1}^j)$ in (14) are replaced by $\tilde{B}_{t-1,D}((z_{t-2} \times a_{t-2} \times z_{t-1}^k), \{a_{t-1}^j\})$. Then

$$
\pi_{T,D}(dx_{1:T})|\tilde{P}_{T,D,X_1:T}^N(\{dj_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx'_{1:T}\})
= \pi_{T,D}(dx'_{1:T})|\tilde{P}_{T,D,X_1:T}^N(\{dj_{1:T} \times dz_{1:T} \times da_{1:T-1} \times dk_{1:T} \times dx'_{1:T}\})
$$

That is, (16) again admits $\pi_{T,D}(dx'_{1:T})$ as a marginal. Incidentally, $\tilde{P}_{T,D,X_1:T}^N$ 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 $(X_{1:T}, J_{1:T}, Z_{1:T}, A_{1:T})$ invariant.
PROOF (of Proposition 4.1). Let
\[
\Xi_T((z_{1:T}, j_{1:T}), da_{1:T-1} \times dk_{1:T})
\]
\[
:= \left[ \prod_{t=2}^{T} \delta_0(da_{t-1}^{j_t}) \prod_{n=0, n \neq j_t}^{N} \tilde{R}_{t-1,D}^n((z_{t-2:t-1}, a_{t-2}), da_{t-1}^n) \right] \times \tilde{R}_{T,D}^N((z_{1:T}, a_{1:T-1}), dk_T) \prod_{t=1}^{T-1} \delta_{a_{t+1}}(dk_t).
\]
Simple algebra then verifies that
\[
\xi_T(z_{1:T}, d_{j_{1:T}}) \Xi_T((z_{1:T}, j_{1:T}), da_{1:T-1} \times dk_{1:T})
\]
\[
= \xi_T(z_{1:T}, dk_{1:T}) \Xi_T((z_{1:T}, k_{1:T}), da_{1:T-1} \times d_{j_{1:T}}).
\]
This completes the proof. \qed

PROOF (of Proposition 4.2). Let
\[
\hat{\Xi}_T((z_{1:T}, j_{1:T}), da_{1:T-1} \times dk_{1:T})
\]
\[
:= \left[ \prod_{t=2}^{T} \delta_0(da_{t-1}^{j_t}) \prod_{n=0, n \neq j_t}^{N} \tilde{R}_{t-1,D}^n((z_{t-2:t-1}, a_{t-2}), \{a_{t-1}^{j_t}\}, \{z_{t-1}^{j_t}\}) \prod_{n=0}^{N} \tilde{R}_{t-1,D}^n((z_{t-2:t-1}, a_{t-2}), da_{t-1}^n) \right] \times \tilde{R}_{T,D}^N((z_{1:T}, a_{1:T-1}), dk_T) \prod_{t=1}^{T-1} \delta_{a_{t+1}}(dk_t).
\]
Simple algebra then verifies that
\[
\hat{\xi}_T(z_{1:T}, d_{j_{1:T}}) \hat{\Xi}_T((z_{1:T}, j_{1:T}), da_{1:T-1} \times dk_{1:T})
\]
\[
= \xi_T(z_{1:T}, dk_{1:T}) \hat{\Xi}_T((z_{1:T}, k_{1:T}), da_{1:T-1} \times d_{j_{1:T}}).
\]
This completes the proof. \qed

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, \(Q_{T,D}^{N,*}\), is:

\[
\mathbb{E}\left[ \mathbb{E}_{T,D,X_{1:T}}^{N,*}(\mathbb{I}\{(Z_{1:T}, A_{1:T-1}) \in dz_{1:T} \times da_{1:T-1}\}) \right] / \prod_{t=1}^{T} \frac{1}{N+1} \sum_{n=0}^{N} G_t(z_t^n)
\]
To see this, let we assume that $m$ where

$$
\pi_t = \prod_{n=0}^{N} M_1(dz^n_1) \prod_{t=2}^{N} R_{t-1, D}^N(z_{t-1}, da_{t-1}^n) M_t(z_{t-1}^{a_{t-1}^{n-1}}, dz_t^n)
$$

$=: \tilde{Q}_{T; D}^N(dz_{1:T} \times da_{1:T-1})$, where $X_{1:T} \sim \pi_{T; D}$, and where – to avoid complications arising from the division by zero – we assume that $m_t$ and $G_t$ are strictly positive.

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

$$
\mathbb{E}[\tilde{Q}_{T; D; X_{1:T}}^N(\{(Z_{1:T}, A_{1:T-1}) \in dz_{1:T} \times da_{1:T-1}\})]
\prod_{t=1}^{T} \frac{1}{N+1} \sum_{n=0}^{N} m_t(z_{t-1}^{a_{t-1}^{n-1}}, z_t^n) G_t(z_t^n)
\times \prod_{t=1}^{T} \lambda(dz_t^0) S_{E,D}^N(z_t^0, dz_t^{1:N})
\times \prod_{t=2}^{T} \prod_{n=0}^{N} \tilde{R}_{t-1, D}^N((z_{t-2:t-1}, a_{t-2}), da_{t-1}^n)
$$

$=: \tilde{Q}_{T; D}^N(dz_{1:T} \times da_{1:T-1})$, 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.

$$
\tilde{P}_T^N(dv_{1:T} \times dw_{1:T} \times da_{1:T-1} \times dk_{1:T})
:= \prod_{t=1}^{T} N(dv_t, w_t^T; \tilde{v}_t; \tilde{S}_t)
\times \prod_{t=1}^{T-1} \delta_0(da_t^0) \prod_{n=1}^{N} \tilde{R}_{t; T}^N((v_t, w_{t-1}, a_t), da_t^n)
\times \tilde{R}_{T; T}^N((v_T, w_{T-1}, a_{T-1}), dk_T)
\times \left[ \prod_{t=1}^{T-1} \delta_{a_t^{t+1}}(dk_t) \right] \text{ [without backward sampling]}
\times \left[ \prod_{t=1}^{T-1} \tilde{B}_{t; T}^N((v_t, w_{t-1:t}, a_t), dk_t) \right]. \text{ [with backward sampling]}
$$

Here, we have defined the following quantities.

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

$$
\tilde{R}_t^N((v_t, w_{t-1}, a_t), N) := \psi^N \left( \{v_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^{N} \right),
$$

where we recall the convention that $w_1^{a_{t-1}^m} \equiv 0$. As usual, when using the forced-move extension, we replace $\psi^N$ by $\phi^N$ in the definition above at time $t = T$.

- **Asymptotic backward kernels.** For any $n \in [N]_0$ and any $t \in [T-1]$,

$$
\tilde{B}_t^N((v_t, w_{t-1:t}, a_t), N) := \psi^N \left( \{v_t^m + w_t^m + w_{t-1}^{a_{t-1}^m}\}_{m=1}^{N} \right).
$$
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 $\Psi^n$ and Rosenbluth–Teller selection function $\Phi^n$ 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 $x_{1:T} \in E_{T,D}$ and any $t \in [T]$, define

$$\mathcal{V}_{t|T} := \pi_T([\partial_t \bar{w}_t]^2),$$

$$\mathcal{W}_{t|T} := \pi_T([\partial_t \bar{w}_{t+1}]^2),$$

$$S_{t|T} := \pi_T([\partial_t \bar{w}_t][\partial_t \bar{w}_{t+1}]),$$

$$\mathcal{V}_{t|T}(x_{t-1:t}) := D^{-1} \sum_{d=1}^{D} \{[\partial_t \bar{w}_t](x_{t-1:t,d})\}^2,$$

$$\mathcal{W}_{t|T}(x_{t:t+1}) := D^{-1} \sum_{d=1}^{D} \{[\partial_t \bar{w}_{t+1}](x_{t+1:t,d})\}^2,$$

$$S_{t|T}(x_{t-1:t+1}) := D^{-1} \sum_{d=1}^{D} \{[\partial_t \bar{w}_t](x_{t-1:t,d})\} \{[\partial_t \bar{w}_{t+1}](x_{t+1:t,d})\},$$

$$\bar{V}_{t|T}(x_{t-1:t}) := [D \mathcal{V}_{t|T}(x_{t-1:t})]^{-1/2} \sup_{d \in [D]} \{[\partial_t \bar{w}_t](x_{t-1:t,d})\},$$

$$\bar{W}_{t|T}(x_{t:t+1}) := [D \mathcal{W}_{t|T}(x_{t:t+1})]^{-1/2} \sup_{d \in [D]} \{[\partial_t \bar{w}_{t+1}](x_{t+1:t,d})\},$$

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:

$$\mathbf{F}_{T,D} := \left\{ x_{1:T} \in E_{T,D} \mid \sup_{t \in [T]} \left\{ \frac{|V_{t|T}(x_{t-1:t}) - V_{t|T}| \vee |W_{t|T}(x_{t:t+1}) - W_{t|T}| \vee |S_{t|T}(x_{t-1:t+1}) - S_{t|T}| \vee |\bar{V}_{t|T}(x_{t-1:t}) \vee \bar{W}_{t|T}(x_{t:t+1})|}{\leq D^{-\eta}} \right\} \right\}.$$

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

**Proof.** The results

$$\pi_{T,D}(\{x_{1:T} \in E_{T,D} \mid |V_{t|T}(x_{t-1:t}) - V_{t|T}| \leq D^{-\eta}\}) \to 1,$$

$$\pi_{T,D}(\{x_{1:T} \in E_{T,D} \mid |W_{t|T}(x_{t:t+1}) - W_{t|T}| \leq D^{-\eta}\}) \to 1,$$

$$\pi_{T,D}(\{x_{1:T} \in E_{T,D} \mid |S_{t|T}(x_{t-1:t+1}) - S_{t|T}| \leq D^{-\eta}\}) \to 1,$$

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

$$\pi_{T,D}(\{x_{1:T} \in E_{T,D} \mid \bar{V}_{t|T}(x_{t-1:t}) \leq D^{-\eta}\}) \to 1,$$

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

$$\pi_{T,D}(\{x_{1:T} \in E_{T,D} \mid \sup_{d \in [D]} [|\partial_t \bar{w}_t](x_{t-1:t,d})| \leq cD^{1/2-\eta}\})$$
where \((X_{1:T,d})_{d \geq 1}\) be IID samples from \(\pi_T\). But this holds since Markov’s inequality along with the fact that \(\eta < 1/4\) and \(C_1\) ensures that

\[
P((|\partial_t \tilde{w}_t(X_{t-1:t,1})| > cD^{1/2 - \eta})) \leq \pi_T(\{|\partial_t \tilde{w}_t|^4\}e^{-1}D^{4(\eta - 1/2)}) = o(D^{-1}).
\]

The result \(\pi_T(D(\{x_{1:T} \in \mathcal{E}_{T,D} | \tilde{W}_{t}(x_{t+1}) \leq D^{-\eta}\}) \rightarrow 1\) follows by the same arguments.

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

\[
Z^n_{t,D} := \begin{cases} x_{t,D}, & \text{if } n = 0, \\ x_{t,D} + \sqrt{\tfrac{\ell}{D}} U^n_{t+1,D}, & \text{if } n \in [N], 
\end{cases}
\]

where \(U^n_{t,D} := U^n_{t:D} \in \mathbb{R}^{1:N}\), with \(U^n_{1:D} \sim N(0, \Sigma)\) for \(\Sigma := \tfrac{1}{2}(I_N + \mathbf{1}_N \mathbf{1}_N^T)\) and where \(U^n_{1:D}\) and \(U^n_{t,e,D}\) are independent whenever \(s \neq t\) or \(d \neq e\). 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

\[
\bar{w}_t(Z^n_{t-1:D}, Z^n_{t,D}) - \bar{w}_t(Z^n_{t-1:D}, Z^n_{t,D}) = V^n_{t,D} + W^n_{t-1,D} + \sum_{i=1}^4 (R^n_{t,D} + S^n_{t-1,D}) + \sum_{i=1}^2 t^{a^n_{t-1,n,i}},
\]

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

\[
\bar{v}_t(Z^n_{t-1:D}, Z^n_{t,D}, Z^n_{t+1,D}) - \bar{v}_t(Z^n_{t-1:D}, Z^n_{t,D}, Z^n_{t+1,D}) = V^n_{t,D} + W^n_{t-1,D} + \sum_{i=1}^4 (R^n_{t,D} + S^n_{t-1,D}) + \sum_{i=1}^2 t^{a^n_{t-1,n,i}} + W^n_{t,D} + \sum_{i=1}^4 s^n_{t,n,i},
\]

where

\[
V^n_{t,D} := \sqrt{\frac{\mathcal{V}_t(x_{t-1:t})}{\mathcal{V}_t(x_{t+1:t,D})}} \sqrt{\frac{\ell}{D}} \sum_{d=1}^D [\partial_t \tilde{w}_t](x_{t-1:t,d,D}) U^n_{t,d,D} + \frac{\ell}{2} \pi_T(\partial^2_T \tilde{w}_t),
\]

\[
W^n_{t,D} := \sqrt{\frac{\mathcal{W}_t(x_{t+1:t,D})}{\mathcal{W}_t(x_{t-1:t})}} \sqrt{\frac{\ell}{D}} \sum_{d=1}^D [\partial_t \tilde{w}_{t+1}](x_{t+1:t,d,D}) U^n_{t,d,D} + \frac{\ell}{2} \pi_T(\partial^2_T \tilde{w}_{t+1}),
\]

and with

\[
R^n_{t,D} := \left(1 - \sqrt{\frac{\mathcal{V}_t(x_{t-1:t})}{\mathcal{V}_t(x_{t-1:t,D})}} \right) \sqrt{\frac{\ell}{D}} \sum_{d=1}^D [\partial_t \tilde{w}_t](x_{t-1:t,d,D}) U^n_{t,d,D}
\]

\[
R^n_{t,D} := \frac{\ell}{2D} \sum_{d=1}^D \left[\partial^2_T \tilde{w}_t(x_{t-1:t,d,D}) - \frac{\ell}{2} \pi_T(\partial^2_T \tilde{w}_t)ight]
\]

\[
R^n_{t,D} := \frac{\ell}{2D} \sum_{d=1}^D \left[\partial^2 \tilde{w}_t(x_{t-1:t,d,D}) \{(U^n_{t,d,D})^2 - 1\}ight]
\]

\[
R^n_{t,D} := \frac{\ell}{2D} \sum_{d=1}^D \left[\partial^2 \tilde{w}_t(x_{t-1:t,d,D} + S^n_{t,d,D}) \sqrt{\ell D} U^n_{t,d,D}\right]
\]
\[ S^{m,1}_{t,D} := \left\{ 1 - \sqrt{\frac{W_{t,T}}{W_{t,T}(x_{t,1:t,d,D})}} \right\} \frac{\ell_{t}}{D} \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t:t+1,d,D}) U_{t,d,D}^{m} \]

\[ S^{m,2}_{t,D} := \frac{\ell_{t}}{2D} D \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t:t+1,d,D}) - \frac{\ell_{T} \pi}{2} \partial_{t} \left( \partial_{t} \bar{w}_{t+1} \right) \]

\[ S^{m,3}_{t,D} := \frac{\ell_{t}}{2D} D \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t:t+1,d,D}) \left\{ (U_{t,d,D}^{m})^2 - 1 \right\} \]

\[ S^{m,4}_{t,D} := \frac{\ell_{t}}{2D} D \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t,d,D} + \eta_{m,d,D} \sqrt{T_{t,d,D}} x_{t+1,d,D}) - \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t,t+1,d,D}) \left\{ (U_{t,d,D}^{m})^2 \right\} \]

\[ T^{m,n,1}_{t,t+1,D} := \sqrt{\frac{\ell_{t+1}}{D}} D \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t:t+1,d,D}) U_{t,d,D}^{m} \]

\[ T^{m,n,2}_{t,t+1,D} := \sqrt{\frac{\ell_{t+1}}{D}} D \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t+1} \right] (x_{t:t+1,d,D}) U_{t,d,D}^{m} \]

for some \( \eta_{t,d,D}, \varsigma_{t,d,D} \in [0, D^{-1/2}] \), and with the usual convention that \( V_{t,D}, W_{t,D}, R_{t,D}, \) and \( S_{t,D} \) are 0 if \( t = 0 \) or \( n = 0 \). Similarly \( T^{m,n,1}_{t,t+1,D} = 0 \) 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. \( \lim_{D \to \infty} \sup_{F_{T,D}} \mathbb{E} \left\| R_{i,j}^{n} \right\| = 0 \)
2. \( \lim_{D \to \infty} \sup_{F_{T,D}} \mathbb{E} \left\| S_{i,j}^{n} \right\| = 0 \)
3. \( \lim_{D \to \infty} \sup_{F_{T,D}} \mathbb{E} \left\| T_{i,j}^{m,n,1} \right\| = 0 \)

**Proof.** By definition of \( F_{T,D} \), using that \( x \mapsto \sqrt{x} \) is concave and increasing so that \( \sqrt{x} - \sqrt{y} \leq \sqrt{|x - y|} \), and since by Jensen’s inequality, \( \mathbb{E}[|X|] \leq \mathbb{E}[X^2]^{1/2} \),

\[ \mathbb{E}[|R_{i,j}^{n,1}|] \leq \sqrt{\frac{|V_{t,T}(x_{t,1:t,d,D}) - V_{t,T}|}{W_{t,T}(x_{t,1:t,d,D})}} \times \frac{\ell_{t}}{D} \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t} \right] (x_{t-1:t,d,D})^2 \mathbb{E}[(U_{t,d,D}^{m})^2] \]

\[ \leq D^{-\eta/2} \ell_{1}^{1/2} \to 0 \]

\[ \mathbb{E}[|S_{i,j}^{n,1}|] \leq \sqrt{\frac{|V_{t,T}(x_{t,1:t,d,D}) - V_{t,T}|}{W_{t,T}(x_{t,1:t,d,D})}} \times \frac{\ell_{t}}{D} \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t} \right] (x_{t:t+1,d,D})^2 \mathbb{E}[(U_{t,d,D}^{m})^2] \]

\[ \leq D^{-\eta/2} \ell_{1}^{1/2} \to 0 \]

From the definition of \( F_{T,D} \),

\[ \mathbb{E}[|R_{i,j}^{n,2}|] \leq \frac{\ell_{T}}{2} D^{-\eta} \to 0 \]

\[ \mathbb{E}[|S_{i,j}^{n,2}|] \leq \frac{\ell_{T}}{2} D^{-\eta} \to 0 \]

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

\[ \mathbb{E}[|R_{i,j}^{n,3}|] \leq \sqrt{\frac{\ell_{T}^{2}}{\pi D^{2}}} \sum_{d=1}^{D} \left[ \partial_{t} \bar{w}_{t} \right] (x_{t-1:t,d,D})^2 \mathbb{E}[(U_{t,d,D}^{m})^2] \]
By the Cramér–Wold theorem, it then suffices to prove that
\[ \mathbb{E}[\mathcal{L}_{m,n}^{t,n+1} \mid (x_{t:t+1},d,D)] \to 0. \]
Since \( \eta_{m,n}^{t,n} \) and \( \tau_{m,n}^{t,n} \) in \([0, D^{-1/2}]\) and since \( \partial_{t}^{2} \hat{w}_{t}, \partial_{t}^{2} \hat{w}_{t+1} \) and \( \partial_{t} \partial_{t+1} \hat{w}_{t+1} \) are Lipschitz-continuous
\[
\mathbb{E}[\mathcal{L}_{m,n}^{t,n+1} \mid (x_{t:t+1},d,D)] \to 0,
\]
\[
\mathbb{E}[\mathcal{L}_{m,n}^{t,n+1} \mid (x_{t:t+1},d,D)] \to 0,
\]
\[
\mathbb{E}[\mathcal{L}_{m,n}^{t,n+1} \mid (x_{t:t+1},d,D)] \to 0,
\]
where \([f]_{Lip}\) denotes the Lipschitz constant of \( f \) w.r.t. the 1-norm.

**Lemma D.4.** Assume A1 and C1 and let \((V_{n}^{t})_{t} \in [T], n \in [N] \) and \((W_{n}^{t})_{t} \in [T-1], n \in [N] \) be the families of Gaussian random variables defined in Section 4.2.1. Then, if \( x_{1:T,D} \in F_{T,D} \), as \( D \to \infty \),
\[ Y_{D} := (V_{1:T,D}^{1:N}, W_{1,D}^{1:N}, \ldots, W_{T-1,D}^{1:N})^{T}, \]
converges in distribution to
\[ Y := (V_{1}^{1:N}, \ldots, V_{T,D}^{1:N}, W_{1}^{1:N}, \ldots, W_{T-1}^{1:N})^{T}. \]

**Proof.** Since \( \mathbb{E}[\mathcal{L}_{m,n}^{t,n+1} \mid (x_{t:t+1},d,D)] \to 0 \), we only need to show convergence in distribution of the centred random vector
\[ \tilde{Y}_{D} := Y_{D} - \mathbb{E}[Y_{D}] = (\tilde{V}_{1:T,D}^{1:N}, \tilde{V}_{1,D}^{1:N}, \tilde{W}_{1,D}^{1:N}, \ldots, \tilde{W}_{T-1,D}^{1:N})^{T} \]
to
\[ \tilde{Y} := Y - \mathbb{E}[Y] = (\tilde{V}_{1:D}^{1:N}, \tilde{V}_{1:D}^{1:N}, \tilde{W}_{1:D}^{1:N}, \ldots, \tilde{W}_{T-1:D}^{1:N})^{T}. \]

By the Cramér–Wold theorem, it then suffices to prove that \( \lambda^{T} \tilde{Y}_{D} \to \lambda^{T} \tilde{Y} \) for any \( \lambda = (\lambda_{1}, \ldots, \lambda_{T}, \ldots, \lambda_{1}, \ldots, \lambda_{T}) \in \mathbb{R}^{N(2T-1)} \). Equivalently, we must show that \( \lambda^{T} \tilde{Y}_{D} \to \lambda^{T} \tilde{Y} \), where
\[ \tau^{2} := \text{var}[\lambda^{T} \tilde{Y}] \]
\[ = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_{n}^{m} \lambda_{n}^{m} \text{cov}[V_{n}^{t}, V_{m}^{t}] + \lambda_{t}^{m} \lambda_{t}^{m} \text{cov}[W_{t}^{n}, W_{m}^{t}] + 2 \lambda_{n}^{m} \lambda_{m}^{m} \text{cov}[V_{n}^{t}, W_{m}^{t}], \]

where we recall the convention that \( W_{m}^{0} = 0 \).

Let \( \mathcal{F}_{d,D} := \sigma(\{U_{t,e,D}^{n} \mid t \in [T], e \in [d], n \in [N]\}) \) as well as
\[ U_{d,D} := \sum_{t=1}^{T} \left[ \frac{V_{t|T}^{n} \lambda_{n}^{m} \lambda_{n}^{m} \text{cov}[V_{n}^{t}, V_{m}^{t}] + \lambda_{t}^{m} \lambda_{t}^{m} \text{cov}[W_{t}^{n}, W_{m}^{t}] + 2 \lambda_{n}^{m} \lambda_{m}^{m} \text{cov}[V_{n}^{t}, W_{m}^{t}]}{\mathcal{L}_{m,n}^{t,n+1}} \right] \sum_{n=1}^{N} \lambda_{n}^{m} U_{t,d,D}^{n} \]
\[ + \mathbb{I}\{t < T\} \left[ \frac{W_{t|T}^{n} \lambda_{n}^{m} \lambda_{n}^{m} \text{cov}[V_{n}^{t}, V_{m}^{t}] + \lambda_{t}^{m} \lambda_{t}^{m} \text{cov}[W_{t}^{n}, W_{m}^{t}] + 2 \lambda_{n}^{m} \lambda_{m}^{m} \text{cov}[V_{n}^{t}, W_{m}^{t}]}{\mathcal{L}_{m,n}^{t,n+1}} \right] \sum_{n=1}^{N} \lambda_{n}^{m} U_{t,d,D}^{n} \]
then $\mathcal{U}_{d,D}$ is $\mathcal{F}_{d,D}$-measurable. Therefore, for $x_{1:T,D} \in \mathcal{F}_{T,D}$, by the central limit theorem for triangular arrays (Dvoretzky, 1972), $\sum_{d=1}^{D} \mathcal{U}_{d,D} = \lambda^T Y_D$ converges in distribution to a zero-mean Gaussian random variable with variance $\tau^2$ if, as $D \to \infty$,

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

To verify the first condition, we note that

$$\sum_{d=1}^{D} \mathbb{E}[\mathcal{U}_{d,D}^2 | \mathcal{F}_{d-1,D}] - \mathbb{E}[\mathcal{U}_{d,D} | \mathcal{F}_{d-1,D}]^2 = \tau^2 + 2 \sum_{t=1}^{T-1} I_t H_{t|T,D} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_t^n \lambda_t^m \Sigma_{n,m},$$

where

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

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

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

$$a^{(i)} := \sup_{t \in [T]} \ell_t^2 \mathcal{V}_{t|T} < \infty, \quad a^{(2)} := \sup_{t \in [T-1]} \ell_t^2 \mathcal{W}_{t|T} < \infty, \quad a^{(3)} := 2 \sup_{t \in [T-1]} \ell_t \ell_{t+1} \sqrt{\mathcal{V}_{t|T} \mathcal{W}_{t|T}} < \infty,$$

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

$$b^{(i)}_{t,d,D} := [\mathcal{V}_{t|T}(x_{t-1:t,D})]^2, \quad b^{(2)}_{t,d,D} := [\mathcal{W}_{t|T}(x_{t:t+1,D})]^2, \quad b^{(3)}_{t,d,D} := \mathcal{V}_{t|T}(x_{t-1:t,D}) \mathcal{W}_{t|T}(x_{t:t+1,D}),$$

as well as

$$M^{(1)}_{t,d} := \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_t^n \lambda_t^m U_{t,d,D}^{m} U_{t,d,D}^{m}, \quad M^{(2)}_{t,d} := \mathbb{I}\{t < T\} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_t^n \lambda_t^m U_{t,d,D}^{m} U_{t,d,D}^{m}, \quad M^{(3)}_{t,d} := \mathbb{I}\{t < T\} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_t^n \lambda_t^m U_{t,d,D}^{m} U_{t,d,D}^{m}.$$

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

$$\{\{|\mathcal{U}_{d,D}| \geq \epsilon\} \subseteq \{a \sum_{i=1}^{3} \sum_{t=1}^{T} b^{(i)}_{t,d,D} | M^{(i)}_{t,d} \geq \epsilon^2\} \subseteq \{a \sum_{i=1}^{3} \sum_{t=1}^{T} |M^{(i)}_{t,d}| \geq D^{2n} \epsilon^2\},$$

and thus

$$\sum_{d=1}^{D} \mathbb{E}[\mathcal{U}_{d,D}^2 \mathbb{I}\{|\mathcal{U}_{d,D}| \geq \epsilon\} | \mathcal{F}_{d-1,D}] \leq \sum_{d=1}^{D} \mathbb{E} \left[ a \sum_{i=1}^{3} \sum_{t=1}^{T} |M^{(i)}_{t,d}| \mathbb{I}\{a \sum_{j=1}^{3} \sum_{s=1}^{T} |M^{(j)}_{s,d}| \geq D^{2n} \epsilon^2\} \right] \to 0.$$
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 $\lim_{D \to \infty} \pi_{T,D}(\mathbf{F}_{T,D}) = 1$. We also fix some $a_t^i \in [N]$ for all $(t,n) \in [T-1] \times [N]$ and some $k_t \in [N]$ for all $t \in [T]$.

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

$$
\sup_{\mathbf{F}_{T,D}} |\mathbb{E}[\mathcal{Y}_D(\mathbf{Z}_D)] - \mathbb{E}[\mathcal{Y}(Y)]|
\leq \sup_{\mathbf{F}_{T,D}} |\mathbb{E}[\mathcal{Y}_D(\mathbf{Z}_D)] - \mathbb{E}[\mathcal{Y}(Y_D)]| + \sup_{\mathbf{F}_{T,D}} |\mathbb{E}[\mathcal{Y}(Y_D)] - \mathbb{E}[\mathcal{Y}(Y)]|
\to 0,
$$

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

$$
\mathcal{Y}_D(\mathbf{Z}_D)
:= \prod_{t=1}^{T-1} \prod_{n=1}^{N} \Psi^{a_t^i} \left( \{ \tilde{w}_t(\mathbf{Z}_{t-1:D}^{a_{t-1}^i}, \mathbf{Z}_{t:D}^{m}) - \hat{w}_t(\mathbf{Z}_{t-1:D}^{0}, \mathbf{Z}_{t:D}^{0}) \}_{m=1}^{N} \right) 
\times \Psi^{k_t} \left( \{ \tilde{w}_T(\mathbf{Z}_{T-1:D}^{a_{T-1}^i}, \mathbf{Z}_{T:D}^{m}) - \hat{w}_T(\mathbf{Z}_{T-1:D}^{0}, \mathbf{Z}_{T:D}^{0}) \}_{m=1}^{N} \right)
\times \prod_{t=1}^{T-1} \Psi^{k_t} \left( \{ \tilde{v}_t(\mathbf{Z}_{t-1:D}^{a_{t-1}^i}, \mathbf{Z}_{t:D}^{m}), \mathbf{Z}_{t+1:D}^{k_{t+1}} - \hat{v}_t(\mathbf{Z}_{t-1:D}^{0}, \mathbf{Z}_{t:D}^{0}, \mathbf{Z}_{t+1:D}^{k_{t+1}}) \}_{m=1}^{N} \right),
$$

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

$$
\mathcal{Y}((v_1^{1:N}, \ldots, v_T^{1:N}, w_1^{1:N}, \ldots, w_{T-1}^{1:N})^T)
:= \prod_{t=1}^{T-1} \prod_{n=1}^{N} \Psi^{a_t^i} \left( \{ v_t^{m} + w_{t-1}^{a_{t-1}^i} \}_{m=1}^{N} \right)
\times \Psi^{k_t} \left( \{ v_T^{m} + w_{T-1}^{a_{T-1}^i} \}_{m=1}^{N} \right) \prod_{t=1}^{T-1} \Psi^{k_t} \left( \{ v_t^{m} + w_t^{m} + w_{t-1}^{a_{t-1}^i} \}_{m=1}^{N} \right).
$$

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

$$
\mathbb{E}[\mathcal{Y}_D(\mathbf{Z}_D)] - \mathbb{E}[\mathcal{Y}(Y_D)]
\leq \left[ \sup_{n \in [N]} \| \Psi^{m} \|_{\text{Lip}} \right]
\times \left[ \sum_{t=1}^{T-1} \sum_{n=1}^{N} \left( \sum_{i=1}^{4} \sum_{j=1}^{2} \left[ R_t^{a_{t-1},i} + R_t^{a_{t-1},i} + S_t^{a_{t-1},i} + S_t^{a_{t-1},i} \right] \right) \right]
$$

so that $\lim_{D \to \infty} \sup_{\mathbf{F}_{T,D}} |\mathbb{E}[\mathcal{Y}_D(\mathbf{Z}_D)] - \mathbb{E}[\mathcal{Y}(Y_D)]| \to 0$, 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 $\mathbb{E}[\mathcal{Y}(Y_D)] \to \mathbb{E}[\mathcal{Y}(Y)]$. Since $0 \leq \mathcal{Y} \leq 1$, this implies $\lim_{D \to \infty} \sup_{\mathbf{F}_{T,D}} |\mathbb{E}[\mathcal{Y}(Y_D)] - \mathbb{E}[\mathcal{Y}(Y)]| \to 0$.

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 \( (X_1, X_2) \sim N(0_2, I_2) \). Then
\[
e^{\sigma_1 X_1 - \sigma_1^2 / 2} \leq \text{ex} \ e^{\sigma_2 X_2 - \sigma_2^2 / 2}.
\]

**Proof.** Let \( \varphi : \mathbb{R} \rightarrow (0, \infty) \) denote the probability density function of a standard normal distribution, let \( \Phi : \mathbb{R} \rightarrow (0, 1) \) denote the associated cumulative distribution function and \( X \sim N(0, 1) \). Then for any \( a > 0, b \in \mathbb{R} \) and \( c \in \mathbb{R} \), writing \( l(a, c) := \log(c)/a + a/2 \),
\[
\mathbb{E}[(e^{a X + b} - c)_+] = \int_{l(a,c)}^{\infty} (e^{ax+b} - c)\varphi(x) \, dx = e^{a^2/2+b} \int_{l(a,c)}^{\infty} \varphi(x-a) \, dx - c \Phi(-l(a,c)) = e^{a^2/2+b} \Phi(-l(a,c) + a) \, dx - c \Phi(-l(a,c)).
\]
(20)

Let \( Y_i := e^{\sigma_i X_i - \sigma_i^2 / 2} \) for \( i \in \{1, 2\} \). Then by (20), for any \( d \in \mathbb{R} \),
\[
\mathbb{E}[(Y_2 - d)_+] - \mathbb{E}[(Y_1 - d)_+] = \Phi\left(\frac{\sigma_2}{2} - \frac{\log(d)}{\sigma_2}\right) - \Phi\left(\frac{\sigma_1}{2} - \frac{\log(d)}{\sigma_1}\right) \geq 0.
\]
Finally, \( \mathbb{E}[Y_1] = \mathbb{E}[Y_2] \) by the properties of the log-normal distribution. This completes the proof. \( \Box \)

**Lemma D.6.** Let \( X := (X_1, \ldots, X_N) \sim N(\mu, \sigma^2 \Sigma) \) and \( Y := (Y_1, \ldots, Y_N) \sim N(\nu, \tau^2 \Sigma) \), where \( |\Sigma|_{i,i} = 1 \) and \( |\Sigma|_{i,j} = 1/2 \) for \( i \neq j \) and where \( \mu := -a 1_N, \nu := -b 1_N \) for \( a \in \mathbb{R} \) and \( b \geq \tau^2/2 \). Assume also that \( X \) and \( Y \) are independent. Then for any binary vector \( \delta := \delta_{1:N} \in \{0, 1\}^N \),
\[
\mathbb{E}\left[\frac{\sum_{i=1}^{N} e^{X_i + \delta_i Y_i}}{1 + \sum_{i=1}^{N} e^{X_i + \delta_i Y_i}}\right] \geq \left(1 + \frac{e^{\sigma^2/2 + \sigma^2 + \tau^2/2 + b}}{N}\right)^{-1}.
\]

In particular, for \( \delta = (0, \ldots, 0) \) we have the tighter bound
\[
\mathbb{E}\left[\frac{\sum_{i=1}^{N} e^{X_i}}{1 + \sum_{i=1}^{N} e^{X_i}}\right] \geq \left(1 + \frac{e^{\sigma^2/2 + b}}{N}\right)^{-1}.
\]

**Proof.** We begin by proving the bound in the special case \( \delta = (0, \ldots, 0) \):
\[
\mathbb{E}\left[\frac{\sum_{i=1}^{N} e^{X_i}}{1 + \sum_{i=1}^{N} e^{X_i}}\right] \geq \mathbb{E}\left[\frac{1}{1 + e^{-X_i/N}}\right] \geq \frac{1}{1 + \mathbb{E}[e^{-X_i/N}]} = \left(1 + \frac{e^{\sigma^2/2 + b}}{N}\right)^{-1},
\]
where the first line follows since \( t \mapsto t/(1 + t) \) is concave and \( \sum_{i=1}^{N} e^{X_i} \leq \text{ex} \ N e^{X_i} \); 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 \( \tau^2/2 - b \leq 0 \),
\[
\frac{\sum_{i=1}^{N} e^{X_i + \delta_i Y_i}}{1 + \sum_{i=1}^{N} e^{X_i + \delta_i Y_i}} \geq \frac{\sum_{i=1}^{N} e^{X_i + \delta_i Y_i + (1-\delta_i)(\tau^2/2 - b)}}{1 + \sum_{i=1}^{N} e^{X_i + \delta_i Y_i + (1-\delta_i)(\tau^2/2 - b)}}.
\]
Furthermore, by Lemma D.5 and Dhaene et al. (2000, Theorem 5),
\[ \sum_{i=1}^{N} e^{X_i + \delta_i Y_i + (1 - \delta_i) (\tau^2/2 - b)} \leq_{\text{ex}} N e^{X_i + Y_i}, \]
and since $t \mapsto t/(1 + t)$ is concave,
\[
\mathbb{E}
\left[
\frac{\sum_{i=1}^{N} e^{X_i + \delta_i Y_i}}{1 + \sum_{i=1}^{N} e^{X_i + \delta_i Y_i}}
\right]
\geq
\mathbb{E}
\left[
\frac{\sum_{i=1}^{N} e^{X_i + Y_i}}{1 + \sum_{i=1}^{N} e^{X_i + Y_i}}
\right]
\geq
\mathbb{E}
\left[
\frac{N e^{X_1 + Y_1}}{1 + N e^{X_1 + Y_1}}
\right]
\geq
\frac{1}{1 + \mathbb{E}[e^{-X_1 - Y_1}]/N}
= \left(1 + \frac{e^{\sigma^2/2 + a + \tau^2/2 + b}}{N}\right)^{-1}.
\]
Here, the penultimate line again follows by Jensen’s inequality since $t \mapsto 1/(1 + t)$ is convex.

\[ \square \]

**Lemma D.7.** Let $\pi(x_{1:T})$ denote some twice differentiable probability density function on $\mathbb{R}^T$ and write $\partial_i f(x_{1:T})$ as shorthand for $\partial_i \int f(x_{1:T}) \, dx$ with $\partial_i^T := \partial_i$. Then
\[ \pi([\partial_i \log \pi])^2 = -\pi(\partial_i^T \log \pi). \]

**Proof.** If $T = 1$, using integration by parts,
\[ \pi([\partial_i \log \pi])^2 = \int \pi'(x)(\log \pi)'(x) \, dx \]
\[ = \pi'(x)|_{-\infty}^\infty - \pi((\log \pi)''\] \[= -\pi((\log \pi)''). \]
For $T > 1$, we let $\pi_t(x_{-t}) := \int_{x_{t+1:T}} \pi(x_{1:T}) \, dx_t$ denote the marginal density of $x_{-t} := (x_{1:t-1}, x_{t+1:T})$ and let $\pi_{t|t} = \pi(x_{1:T})/\pi_t(x_{-t})$. Since $\partial_i \log \pi(x_{1:T}) = \partial_i \log \pi_{t|t}(x_{t|t})$, \[ \pi([\partial_i \log \pi])^2 = \pi([\partial_i \log \pi_{t|t}])^2 \]
\[ = \int \left[ \int [\partial_i \log \pi_{t|t}(x_{t|t})]^2 \pi_{t|t}(x_{t|t}) \, dx_t \right] \pi_t(x_{-t}) \, dx_{-t} \]
\[ = -\pi(\partial_i^T \log \pi_{t|t}) \]
\[ = -\pi(\partial_i^T \log \pi), \]
where the third line follows by integration by parts in the same way as (21) with $\pi(x)$ replaced by the conditional distribution $\pi_{t|t}(x_{t|t})$. \[ \square \]

**Proof (of Proposition 4.8).** By Lemma 1.1, it suffices to consider the case without forced move. Under Assumption A4, we have $W_t^n = 0$ for any $t \in [T]$ and any $n \in [N]$, so that
\[ \tilde{F}_t^n(T) = F_t^n + \tilde{F}_t^n(T) = F_t^n + \int_{v_{t|t}(n)}^{v_t^n} \Psi^n((v_t^n)_{m=1}^{N}) \, dv_t^n, \]
does not depend on $a_{t-1}$ (nor on $w_{t-1}$). As a consequence,

$$\tilde{a}_t^n(t) = \prod_{s=t}^{T} \sum_{n \in [N]} \mathbb{E}_T^N[\psi^n(\{V^m_s\}_{m=1}^N)] \geq \prod_{s=t}^{T} \left(1 + \frac{\exp(\ell_s T_s [T])}{N}\right)^{-1},$$

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

$$\tilde{B}_t^N((v_t, w_{t:t-1}, a_{t-1}), \{n\}) = \psi^n(\{v^m_t\}_{m=1}^N),$$

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

$$\tilde{a}_t^n(t) = \sum_{n \in [N]} \mathbb{E}_T^N[\psi^n(\{V^m_t\}_{m=1}^N)] \geq \left(1 + \frac{\exp(\ell_t T_t [T])}{N}\right)^{-1},$$

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

\[\square\]

**APPENDIX E: ADDITIONAL SIMULATION RESULTS**

**E.1. Effective samples sizes.** Figure 5 displays the $\pi_{T,D}$-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

$$\text{ESS}(W_0^N) := \frac{1}{\sum_{n=0}^{N} (W^n)^2},$$

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

1. The first column shows $\mathbb{E} \{\mathbb{E}^N_{T,D,X_{1:T}}[\text{ESS}(W_0^N)]\}$, where,

$$W_t^n := \begin{cases} \psi^n(\{w_t(Z^{m}_{t-1}; Z^0_t) - w_t(Z^0_t)\}_{m=1}^N), & \text{without backward sampling}, \\ \psi^n(\{v_t(Z^m_t, Z^{K_{t+1}}_t) - v_t(Z^0_t, Z^{K_{t+1}}_t)\}_{m=1}^N), & \text{with backward sampling}. \end{cases}$$

2. The second column shows $\mathbb{E} \{\mathbb{E}^N_{T,D,X_{1:T}}[\text{ESS}(\tilde{W}_0^N)]\}$, where

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

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 $> 1$. 
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-proposal” versions which make $N$ proposals. Specifically, we compare the following four methods, where we recall that $\Phi^0$ denotes the Rosenbluth–Teller selection function which was defined in (2) and which reduces to the usual MH acceptance probability $\Phi^1 = 1 \land \exp$ 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 $z_{1:T}^n := (z_1^n, \ldots, z_T^n)$. 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-proposal RWMH algorithm. Each state $x_t$ 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 $\ell_t/D$, 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.
A LGORITHM 4 (RWMH). Given $x_{1:T} := x_{1:T}[l] \in E_{T,D}$.
1. Sample all particles $Z^u_{t} = z^u_{t}$ as in Step 1 of Algorithm 2, where $l_1 = \ldots = l_T = l$, for some $l > 0$.
2. Sample $K = k \in [N]_0$ with probability $\Phi^k(\{h^m\}_{m=1}^N)$, where
   \[ h^m := \log \pi_{T,D}(z^m_{1:T}) - \log \pi_{T,D}(z^0_{1:T}). \]
3. Set $X'_{1:T} := x'_{1:T} := z^k_{1:T}$.
4. Return $x_{1:T}[l+1] := x'_{1:T}$.

A LGORITHM 5 (Blocked RWMH). Given $x_{1:T} := x_{1:T}[l] \in E_{T,D}$.
1. Sample all particles $Z^u_{t} = z^u_{t}$ as in Step 1 of Algorithm 2.
2. For $t = 1, \ldots, T$,
   a) sample $K_t = k_t \in [N]_0$ with probability $\Phi^k_t(\{h^m_t\}_{m=1}^N)$, where
      \[ h^m_t := \log \pi_{T,D}(x'_{1:t-1}, z^m_{1:t}, x_{t+1:T}) - \log \pi_{T,D}(x'_{1:t-1}, x_{t:T}). \]
   b) set $X_t' := x_t' := z^k_t$.

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 $\ell_t$ in the i-RW-CSMC and blocked RWMH algorithms and the scale factor $\ell$ in the RWMH algorithm. Recall that as discussed in Section 6, we adapt the scale factors so that the acceptance rate is around $\alpha$.

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.

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 (2015). We stress that this model does not generally satisfy the IID assumption A1.

Let $q_{m,C}$ denote a density (w.r.t. a suitable version of the Lebesgue measure, $\lambda$) of a $D$-dimensional normal distribution with mean vector $m \in \mathbb{R}^D$ and covariance matrix $C \in \mathbb{R}^{D \times D}$. Let $y_t = (y_{t,d})_{d \in [D]} \in \mathbb{R}^D$ be a vector of $D$ log-returns observed at time $t \in [T]$. Then:

$$G_t(x_t) := q_{0_D, \text{diag}(\exp(x_t))}(y_t),$$

$$m_t(x_{t-1}, x_t) := q_{\mu + \Phi(x_{t-1} - \mu), U}(x_t),$$

where $\exp$ is applied element-wise, where $\text{diag}(x)$ is a diagonal matrix with diagonal given by the vector $x$ and where $\mu, \phi \in \mathbb{R}^D$, $\Phi := \text{diag}(\phi)$, $U \in \mathbb{R}^{D \times D}$ is some covariance matrix. We also recall that $0_D, 1_D \in \mathbb{R}^D$ are column vectors filled with zeros and ones, respectively, and $I_D \in \mathbb{R}^{D \times D}$ is an identity matrix. At time $t = 1$, $m_1(x_1) = q_{\mu, U_1}(x_1)$, where $U_1$ is the stationary covariance matrix of the latent autoregressive process of log-jectivilities, i.e.

$$\text{vec} U_* = (I_{D^2} - \Phi \otimes \Phi)^{-1} \text{vec} U.$$
Finally, we assume $\mu = \nu 1_D$, $\Phi = \phi 1_D$ 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 $\textbf{A1}$ is violated unless $\rho = 0$.

**E.3.2. Illustration of the algorithms and adaptation of $\ell_t$.** 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 $\ell_t$ 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.
Finally, we illustrate the adaptive rule for setting the scale factors $\ell_t$ 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 $\ell_1 = \ldots = \ell_T = 100$.

APPENDIX F: USE FOR PARAMETER ESTIMATION

The Feynman–Kac model is typically specified through a set of parameters $\theta \in \Theta$, i.e. $M_t = M_{\theta,t}$, $m_t = m_{\theta,t}$, $G_t = G_{\theta,t}$ and $\pi_{T,D} = \pi_{\theta,T,D}$. In this case, we likewise write
MCMC kernels induced by Algorithms 1, 2 and 3 as $P^N_{T,D} = P^N_{\theta,T,D}$, $\bar{P}^N_{T,D} = \bar{P}^N_{\theta,T,D}$, and $P^N_{\theta,T,D} = P^N_{\theta,T,D}$.

If $\theta$ 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 $\pi(d\theta \times dx_{1:T}) \propto \mu(d\theta) \pi_{\theta,T,D}(dx_{1:T})$, where the probability measure $\mu$ on $\Theta$ is the prior distribution for $\theta$.

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 $R_{X_1,T}(\theta,d\theta)$ denotes some $\pi(d\theta|x_{1:T})$-invariant MCMC kernel (e.g. often a convolution of multiple MH updates).

**Algorithm 6 (particle Gibbs sampler).** Given $(\theta[l], x_{1:T}[l]) \in \Theta \times E_{T,D}$,

1. sample $\theta[l+1] \sim R_{X_1,T}[\theta[l], \theta[l+1], x_{1:T}[l], \cdot)$,
2. sample $x_{1:T}[l+1] \sim P^N_{\theta[l+1],T,D}(x_{1:T}[l], \cdot)$.

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. $P^N_{\theta,T,D}$ or $\bar{P}^N_{\theta,T,D}$. To see this, note that these kernels leave $\pi_{\theta,T,D}(dx_{1:T}) = \pi(dx_{1:T}|\theta)$ invariant.

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

The RW-EHMM-based algorithm is outlined in Algorithm 7, where $q_{z_1:T}(\theta,d\theta')$ 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 $\theta$-direction compared to conditioning on a particular sequence of latent states.

**Algorithm 7 (alternative RW-EHMM-based parameter estimation).** Given $(\theta, x_{1:T}) := (\theta[l], x_{1:T}[l]) \in \Theta \times E_{T,D}$,

1. sample $Z_{1:T} = z_{1:T}$ via Step 1 of Algorithm 2,
2. sample $\theta' \sim q_{z_1:T}(\theta, \cdot)$ and set
   $r := \frac{q_{z_1:T}(\theta',\theta)\mu(\theta')}{q_{z_1:T}(\theta,\theta')\mu(\theta)} \sum_{n_1:T \in [N]_0^T} \pi_{\theta,T,D}(z_{1:T}^{n_1}, \ldots, z_{1:T}^{n_T})$
3. sample $U \sim \text{Unif}[0,1]$,
4. if $u \leq r$,
   • sample $K_{1:T} = k_{1:T} \sim \xi_{\theta,T}(z_{1:T}, \cdot)$,
   • return $(\theta[l+1], x_{1:T}[l+1]) := (\theta', z_{1:T}^{k_1}, \ldots, z_{1:T}^{k_T})$;
   else,
   • sample $K_{1:T} = k_{1:T} \sim \xi_{\theta,T}(z_{1:T}, \cdot)$,
   • return $(\theta[l+1], x_{1:T}[l+1]) := (\theta, z_{1:T}^{k_1}, \ldots, z_{1:T}^{k_T})$. 


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 $O(N)$ 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, $q_{z_{1:T}, a_{1:T-1}}(\theta, d\theta')$ 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 $n$th particle at time $t$ which was already introduced in Appendix D.1:

$$\tilde{R}_{\theta,t,D}^N((z_{t-1:t}, a_{t-1}), \{n\}) := \frac{m_{\theta,t}(z_{t-1}^{n_t}, z_t^{n_t}) G_{\theta,t}(z_t^{n_t})}{\sum_{m=0}^N m_{\theta,t}(z_{t-1}^{m_t}, z_t^{m_t}) G_{\theta,t}(z_t^{m_t})},$$

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

$$\tilde{B}_{\theta,t,D}^N((z_{t-1:t}, a_{t-1}, z_{t+1}^{k_{t+1}}), \{n\}) := \frac{m_{\theta,t}(z_{t-1}^{n_t}, z_t^{n_t}) G_{\theta,t}(z_t^{n_t}) m_{\theta,t+1}(z_t^{n_t}, z_{t+1}^{k_{t+1}})}{\sum_{m=0}^N m_{\theta,t}(z_{t-1}^{m_t}, z_t^{m_t}) G_{\theta,t}(z_t^{m_t}) m_{\theta,t+1}(z_t^{m_t}, z_{t+1}^{k_{t+1}})}.$$

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

\begin{algorithm}
\caption{Algorithm 8 (alternative i-RW-CSMC-based parameter estimation). \label{alg:8}}
Given $(\theta, x_{1:T}) := (\theta[l], x_{1:T}[l]) \in \Theta \times E_{T,D}$,
\begin{enumerate}
\item sample $(Z_{1:T}, A_{1:T-1}) = (z_{1:T}, a_{1:T-1})$ via Step 1 of Algorithm 3,
\item sample
\begin{itemize}
\item $\theta' = \theta' \sim q_{z_{1:T}, a_{1:T-1}}(\theta', \cdot)$;
\item $A_{1:T-1}' = a_{1:T-1}' \sim \prod_{t=1}^{T-1} \prod_{n=0}^N \tilde{R}_{\theta,t,D}^N((z_{t-1:t}, a_{t-1}'), \{a_t'\})$,
\end{itemize}
and set
$$r := \frac{q_{z_{1:T}, a_{1:T-1}}(\theta', \theta) \mu(\theta') \prod_{t=1}^T \sum_{n=0}^N m_{\theta,t}(z_{t-1}^{a_{t-1}'}^{n_t}, z_t^{n_t}) G_{\theta,t}(z_t^{n_t})}{\sum_{n=0}^N m_{\theta,t}(z_{t-1}^{a_{t-1}'}^{n_t}, z_t^{n_t}) G_{\theta,t}(z_t^{n_t})}$$
\item sample $U = u \sim \text{Unif}[0,1],$
\item if $u \leq r$,
\begin{itemize}
\item sample $k_T = k_{T} \sim \tilde{R}_{\theta,T,D}^N((z_{T-1:T}, a_{T-1}'), \cdot),$
\item for $t = T - 1, \ldots, 1$, sample $k_t = k_{t} \sim \tilde{B}_{\theta,t,D}^N((z_{t-1:t}, a_{t-1}', z_{t+1}^{k_{t+1}}), \cdot)$;
\item return $(\theta[l + 1], x_{1:T}[l + 1]) := (\theta'(z_1^{k_1}, \ldots, z_T^{k_T}));$
\end{itemize}
else,
\begin{itemize}
\item sample $k_T = k_{T} \sim \tilde{R}_{\theta,T,D}^N((z_{T-1:T}, a_{T-1}), \cdot),$
\item for $t = T - 1, \ldots, 1$, sample $k_t = k_{t} \sim \tilde{B}_{\theta,t,D}^N((z_{t-1:t}, a_{t-1}, z_{t+1}^{k_{t+1}}), \cdot)$;
\item return $(\theta[l + 1], x_{1:T}[l + 1]) := (\theta(z_1^{k_1}, \ldots, z_T^{k_T})).$
\end{itemize}
\end{enumerate}
\end{algorithm}

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.