Online Particle Smoothing with Application to Map-matching

Samuel Duffield, Sumeetpal S. Singh

***Abstract*—We introduce a novel method for online smoothing in state-space models that utilises a fixed-lag approximation to overcome the well known issue of path degeneracy. Unlike classical fixed-lag techniques that only approximate certain marginals, we introduce an online resampling algorithm, called *particle stitching*, that converts these marginal samples into a full posterior approximation. We demonstrate the utility of our method in the context of map-matching, the task of inferring a vehicle's trajectory given a road network and noisy GPS observations. We develop a new state-space model for the difficult task of map-matching on dense, urban road networks.**

***Index Terms*—State-space models, sequential Monte Carlo, particle smoothing, backward simulation, map-matching, GPS.**

I. Introduction

Widely used in modelling dynamical systems and time series, a state-space model is fully defined by the following distributions for the hidden $\{x_t\}_{t=0}^{\infty}$ and observed process $\{y_t\}_{t=0}^{\infty}$

$$\begin{aligned} &p(x_0), && \\ &p(x_t|x_{t-1}), && t = 1, 2, 3, \dots \\ &p(y_t|x_t), && t = 0, 1, 2, \dots \end{aligned}$$

The statistical inference goal of *smoothing* is the task of approximating the joint smoothing distribution (or posterior distribution)

$$p(x_{0:T}|y_{0:T}) \propto p(x_0|y_0) \prod_{t=1}^{T} p(x_t|x_{t-1}) p(y_t|x_t), \qquad (1)$$

where $x_{0:T} = (x_0, \dots, x_T)$ are latent states to be inferred and $y_{0:T} = (y_0, \dots, y_T)$ are given observations.

For *online smoothing*, we have the additional requirement of being able to quickly and accurately update an approximation of $p(x_{0:T-1}|y_{0:T-1})$ to approximate $p(x_{0:T}|y_{0:T})$ in light of receiving a new observation y_T.

A popular and powerful approach to inference in generic state-space models is that of particle filtering/smoothing (or sequential Monte Carlo), see [1] for a recent, thorough review. Particle smoothers approximate the joint smoothing distribution with a collection of weighted (or unweighted) particles

$$p(x_{0:T}|y_{0:T}) \approx \sum_{i=1}^{N} w_T^{(i)} \delta_{x_{0:T}} \left(x_{0:T}^{(i)} \right),$$

as opposed to particle filters which are only asked to approximate the filtering marginals $p(x_T|y_{0:T})$. Existing online particle smoothing approximations either degenerate as the length of the state-space model increases [2], [3] or only target the smoothing marginals $p(x_t|y_{0:T})$ [4] rather than the joint smoothing distribution. More recently, efficient techniques have been developed to produce online approximations to expectations with respect to $p(x_{t-1}, x_t|y_{0:T})$ [5], [6]. However, these are integrand specific whereas a particle approximation to the joint smoothing distribution is significantly more useful as mathematical expectations can be calculated over a variety of functions defined over full trajectories, thus providing the user with complete flexibility.

In summary, our motivation is to develop an algorithm that simultaneously satisfies the following requirements

(R1) **Joint Smoothing**: The algorithm efficiently approximates $p(x_{0:T}|y_{0:T})$ rather than only marginals.
(R2) **Online**: Our approximation can be quickly updated as it receives new observations.
(R3) **Non-degenerate**: The algorithm avoids the path degeneracy of classical particle filters.

The application of particle smoothing to urban map-matching is particularly motivating. The combination of dense road networks (with frequent intersections) and noisy GPS observations leads to uncertainty over the location and route of vehicles. This is a compelling application of online joint smoothing where it is desirable to represent the route with a diverse collection of particles, where each particles describes a plausible vehicle trajectory.

The contribution of this work is both methodological and applied.

- We develop a new online resampling method called *particle stitching* that converts marginal fixed-lag samples, which are non-degenerate, into a full posterior approximation (1). In doing so we jointly satisfy (R1)-(R3) using particle smoothing.
- We present a new state-space model for map-matching in dense, urban road networks. We demonstrate the benefits of uncertainty quantification over popular optimisation-based approaches and show the performance of offline particle smoothing can be matched by the online particle smoothers introduced here.

The rest of the paper is structured as follows. In Section II we discuss related online and offline algorithms for particle smoothing, as summarised in TABLE I alongside those introduced in this paper. Section III describes how to combine blocked samples in a way that is invariant for a fixed-

S. Duffield and S.S. Singh are with the Department of Engineering, University of Cambridge

Contact: sddd2@cam.ac.uk, sss40@cam.ac.uk

S. Duffield is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) doctoral training award

lag joint smoothing distribution and Section IV introduces two efficient online methods for generating these blocked samples. Section V describes the problem of map-matching and Section VI demonstrates numerically the benefits of uncertainty quantification as well as the performance of the introduced online particle smoothers and their sensitivity to key parameters. In Section VII we conclude and discuss some potential extensions.

II. Background and Related Work

A. Particle Filtering and Path Degeneracy

A classical particle filter runs a single forward pass, updating particles at every observation. Each update consists of three steps: an optional *resample* step, a *propagation* step and a *weighting* step. These steps are described in Alg. 1.

The resampling operation converts a weighted sample into an unweighted sample that likely contains duplicates $\left\{x^{(i)}, w^{(i)}\right\}_{i=1}^N \to \left\{x^{(i)}, \frac{1}{N}\right\}_{i=1}^N$. The most common resampling method is multinomial sampling (with replacement)

$$\begin{aligned} &\text{Sample} \quad a^{(i)} \sim \text{Categorical}(\{w^{(i)}\}_{i=1}^N), && i = 1, \dots, N, \\ &\text{Set} \quad x^{(i)} \leftarrow x^{(a^{(i)})} \quad w^{(i)} \leftarrow 1/N, && i = 1, \dots, N, \end{aligned}$$

where $\text{Categorical}(\{w^{(i)}\}_{i=1}^N) = \sum_{i=1}^N w^{(i)} \delta(i)$ simply draws an index i from the index set $\{1, \dots, N\}$ with probabilities $w^{(i)}$.

Algorithm 1 Particle Filter

```
0: Input sample $\{x_{0:T-1}^{(i)}, w_{T-1}^{(i)}\}_{i=1}^N$ approximating $p(x_{0:T-1}|y_{0:T-1})$ and new observation $y_T$.
1: Resample (optional)
       $\left\{x_{0:T-1}^{(i)}, w_{T-1}^{(i)}\right\}_{i=1}^N \to \left\{x_{0:T-1}^{(i)}, \frac{1}{N}\right\}_{i=1}^N$
2: Propagate by sampling from some proposal distribution
       $x_T^{(i)} \sim q\left(x_T \,\middle|\, x_{T-1}^{(i)}, y_T\right), \qquad i = 1, \dots, N.$
   Append to particle $x_{0:T}^{(i)} = \left(x_{0:T-1}^{(i)}, x_T^{(i)}\right)$.
3: Weight and normalise
       $w_T^{(i)} \propto \frac{p\left(x_T^{(i)} \,\middle|\, x_{T-1}^{(i)}\right) p\left(y_T \,\middle|\, \tilde{x}_T^{(i)}\right)}{q\left(x_T^{(i)} \,\middle|\, x_{T-1}^{(i)}, y_T\right)} w_{T-1}^{(i)} \qquad i = 1, \dots, N.$
4: Output sample $\{x_{0:T}^{(i)}, w_T^{(i)}\}_{i=1}^N$ approximating $p(x_{0:T}|y_{0:T})$
```

Although most commonly used only to approximate the filtering marginals $p(x_T|y_{0:T})$, the particle filter update as described in Alg. 1 does provide an asymptotically unbiased approximation to the full smoothing distribution $p(x_{0:T}|y_{0:T})$.

The reason that a classical particle filter is almost never used to approximate the smoothing distribution is due to *path degeneracy*. Path degeneracy occurs in state-space models with large T. For repeated particle filter updates, the early coordinates of particles (e.g. $x_0^{(i)}$) will only be altered in the resampling step. Resampling can only decrease particle diversity (the number of distinct particles) and therefore as T increases, and resampling has occurred with suitable frequency, the particle approximation of $p(x_0|y_{0:T})$ will eventually collapse to just a single particle duplicated N times.

Due to the optional nature of the resampling step, adaptive schemes are often applied, with a popular choice to be to only resample if the *effective sample size*, $\text{ESS}(\{w^{(i)}\}_{i=1}^N) = 1/\sum_{i=1}^N w^{(i)\,2}$, falls below some threshold. Variants on the effective sample size criterion have been developed that may better quantify sample quality [11], [12], although not investigated here. In addition to such criteria that ensure resampling is only applied when necessary, a variety of techniques have been developed to mitigate the effects of path degeneracy. These include adaptively increasing the number of particles [13] as well as more sophisticated resampling schemes [14]–[16] which aim to only resample particles with negligible weights. These approaches can improve performance over multinomial resampling however still suffer the collapse in the approximation of $p(x_0|y_{0:T})$ for T sufficiently large.

The merits of resampling are well known, in particular resampling is vital in ensuring a diverse particle approximation to the filtering marginals $p(x_T|y_{0:T})$ - often the particle filter's primary task. Indeed, a particle filter without resampling is simply importance sampling on an a space whose dimension increases with every new observation. As a result, the divergence between importance and target distribution increases and the number of particles must increase exponentially [17] - the addition of resampling means a stable approximation to the filtering marginals is maintained.

B. Marginal Fixed-Lag

An alternative approach in mitigating the path degeneracy induced by repeated resampling is to simply stop resampling the early coordinates of particles, as proposed in [4]. That is, replace the resample step (Alg. 1, step 1) with the scheme described in Alg. 2.

Algorithm 2 Marginal Fixed-lag Resampling (for $T > L$)

```
1: Fix $\{x_{0:T-L-1}^{(i)}\}_{i=1}^N$
2: Resample only recent coordinates
       $\left\{x_{T-L:T}^{(i)}, w_T^{(i)}\right\}_{i=1}^N \to \left\{x_{T-L:T}^{(i)}, \frac{1}{N}\right\}_{i=1}^N$
   Stitch arbitrarily $x_{0:T}^{(i)} = (x_{0:T-L-1}^{(i)}, x_{T-L:T}^{(i)})$.
```

The justification for freezing $x_{0:T-L-1}^{(i)}$ is that after a certain lag L the smoothing distribution of early coordinates become (approximately) independent of new observations

$$p(x_{0:t}|y_{0:T}) \approx p(x_{0:t}|y_{0:\min(t+L,T)}), \tag{2}$$

this represents a *fixed-lag approximation*.

The major issue with the fixed-lag resampling scheme described in Alg. 2 is that early and recent coordinates of particles are *arbitrarily stitched* together and therefore only

	Joint Smoothing	Online	Path Degeneracy	Fixed-lag Approx.	Complexity
Particle Filter	✓	✓	✓		N
Marginal Fixed-lag [4]		✓	For large L	✓	N
Forward Filtering-Backward Smoothing [7]					N^2
Forward Filtering-Backward Simulation [8]	✓				N^2 N with RS†
Forward Smoothing for Additive Functionals [5]		✓			N^2
PaRIS [6]		✓			N with RS†
Block Sampling [9]	✓	✓	✓		LN
Online Particle Smoother (Alg. 4)	✓	✓	For large L	✓	N^2 N with RS†
Online Particle Smoother with Backward Simulation (Alg. 5)	✓	✓		✓	LN^2 LN with RS†

TABLE I: Comparison of particle smoothing algorithms, for number of particles N and fixed-lag parameter L. †For these algorithms the rejection sampling technique of [10] can be applied to obtain linear complexity when a bound for the transition density $p(x_t|x_{t-1})$ is available.

provide a particle approximation to the *fixed-lag marginal* smoothing distribution:

$$p^L_{\text{marg}}(x_{0:T}|y_{0:T}) = \prod_{t=0}^{T-L-1} p(x_t|y_{0:t+L}) \\ p(x_{T-L:T}|x_{T-L-1}, y_{T-L:T}). \quad (3)$$

As such we lose the ability to take expectations over the joint distribution of early coordinates $x_{0:T-L-1}$.

A more useful particle approximation targets the *fixed-lag joint* smoothing distribution

$$p^L(x_{0:T}|y_{0:T}) = p(x_0|y_{0:L}) \\ \prod_{t=1}^{T-L-1} p(x_t|x_{t-1}, y_{t:t+L}) \\ p(x_{T-L:T}|x_{T-L-1}, y_{T-L:T}), \quad (4)$$

which permits expectations over full trajectories $x_{0:T}$.

C. Offline Smoothing

A popular method for approximating the smoothing distribution $p(x_{0:T}|y_{0:T})$ is that of *forward filtering-backward smoothing* [7]. This method stores the output of each particle filter update $\{x_t^{(i)}\}_{i=1}^N \sim p(x_t|y_{0:t})$ and runs a full backward pass that updates the weights based on the backward decomposition

$$p(x_{t-1}|x_t, y_{0:t-1}) = \frac{p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})}{p(x_t|y_{0:t-1})}, \\ t = T, \dots, 1. \quad (5)$$

Similarly, *forward filtering-backward simulation* (FFBSi) [8] runs a full backward pass based on the same decomposition, but samples a new ancestor from the particle cloud (according to (5)) rather than only updating the weights and thus targets the joint smoothing distribution. Additionally, [10] showed that rejection sampling can significantly reduce the complexity of this pass in the case that $p(x_t|x_{t-1})$ has a tractable upper bound. An implementation of backward simulation is described in, Alg. 6.

Both of these algorithms require a full backward pass in light of every new observation and therefore are not suitable for online smoothing. In addition, the popular PMCMC approach [18] is iterative in nature and thus not extensible to online smoothing.

D. Online Smoothing

It was noted in [5] that the forward filtering-backward smoothing algorithm can be implemented in a single forward pass in the case of *additive functionals*, i.e. expectations of the form

$$\mathbb{E}_{p(x_{0:T}|y_{0:T})}\left[\sum_{t=0}^{T-1} g_t(x_t, x_{t+1})\right]. \quad (6)$$

Computing such expectations is useful for calibrating the state-space model parameters, see [3], [19], [20]. The PaRIS algorithm [6] combines this technique with rejection sampling to implement an efficient and cheap version of forward filtering-backward simulation in a single forward pass (for additive functionals).

Although it permits the implementation of these online algorithms, the requirement of additive functionals is very restrictive. Additionally, the forward only technique is *function specific*, i.e. it directly updates an approximation to the expected value of the additive functional. Our approach induces a controllable bias through the fixed-lag approximation but can approximate any expectation $\mathbb{E}_{p(x_{0:T}|y_{0:T})}[f(x_{0:T})]$ over the joint smoothing distribution and is therefore significantly more general than the marginal or additive functional approaches.

Block sampling [9] is an online method that targets the joint smoothing distribution. Every time a new observation is received, the block sampling scheme discards the most recent coordinates (within lag L) and re-proposes from an enlarged proposal distribution based on many more recent observations. Although this scheme does make use of a fixed-lag parameter, the weights and resampling still act on the full joint smoothing distribution. By moving a larger proportion of the trajectories, block sampling can (when combined with adaptive resampling schemes) mitigate but not avoid path degeneracy as resampling still takes place over full trajectories. Our proposed method builds on the block sampling approach by proposing blocks with a single coordinate overlap and then resampling in a way that targets the fixed-lag joint smoothing distribution (4).

III. Fixed-lag Particle Stitching

In this section, we first detail the technique underlying backward simulation and then describe how the same technique can be applied in a forward implementation under a fixed-lag approximation.

A. Backward Simulation

In a single iteration of backward simulation, we have samples

$$\left\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\right\}_{i=1}^{N} \text{ approximating } p(x_{t-1}|y_{0:t-1}),$$

$$\left\{x_t^{(j)}\right\}_{j=1}^{N} \text{ approximating } p(x_t|y_{0:T}),$$

but desire samples from the joint

$$\left\{(x_{t-1}^{(j)}, x_t^{(j)})\right\}_{j=1}^{N} \text{ approximating } p(x_{t-1}, x_t|y_{0:T}).$$

Here, and in the next sections, we have used tildes to represent the particles from which we look to resample - particles without tildes remain fixed. Seeing as $\{x_t^{(j)}\}_{j=1}^N$ already have the correct marginals this amounts to sampling

$$x_{t-1}^{(j)} \sim p(x_{t-1}|x_t^{(j)}, y_{0:t-1}) \qquad \text{for } j = 1, \ldots, N.$$

We cannot sample from $p(x_{t-1}|x_t^{(j)}, y_{0:t-1})$ directly, so we consider the decomposition

$$p(x_{t-1}|x_t^{(j)}, y_{0:t-1}) = \frac{p(x_t^{(j)}|x_{t-1})p(x_{t-1}|y_{0:t-1})}{\int p(x_t^{(j)}|x_{t-1})p(x_{t-1}|y_{0:t-1})dx_{t-1}}.$$

and then use $p(x_{t-1}|y_{0:t-1})$ to form an importance sampling approximation. Specifically, we use the particle filter's empirical approximation in the numerator and denominator to obtain the self-normalised weights [5], [8]

$$w_{t-1}^{(i\leftarrow j)} = \frac{\tilde{w}_{t-1}^{(i)} p(x_t^{(j)}|\tilde{x}_{t-1}^{(i)})}{\sum_{k=1}^N \tilde{w}_{t-1}^{(k)} p(x_t^{(j)}|\tilde{x}_{t-1}^{(k)})}.$$

where $\sum_{k=1}^N \tilde{w}_{t-1}^{(k)} p(x_t^{(j)}|\tilde{x}_{t-1}^{(k)})$ is an asymptotically unbiased approximation of $p(x_t^{(j)}|y_{0:t-1})$ and now $\sum_{i=1}^N w_{t-1}^{(i\leftarrow j)} = 1$.

Sampling from the empirical distribution $\sum_{i=1}^N w_{t-1}^{(i\leftarrow j)} \delta_{x_{t-1}}(\tilde{x}_{t-1}^{(i)})$ directly for each j is the $O(N^2)$ backward simulation technique [8], which when iterated for $t = T, \ldots, 1$ is asymptotically unbiased for the joint smoothing distribution $p(x_{0:T}|y_{0:T})$.

B. Fixed-lag Forward Simulation - Intractable

In this section we aim to approximate the fixed-lag joint smoothing distribution defined in (4) during the forward pass of particle filtering thus obtaining an online algorithm. In the setting of fixed-lag forward simulation we have

$$\left\{x_{t-1}^{(i)}\right\}_{i=1}^{N} \text{ approximating } p(x_{t-1}|y_{0:T-1}),$$

$$\left\{\tilde{x}_t^{(j)}, \tilde{w}_t^{(j)}\right\}_{j=1}^{N} \text{ approximating } p(x_t|y_{0:T}),$$

where $t = T - L$. Recall the fixed-lag approximation uses all past, present and at most L future observations to infer x_{t-1}, which implies $p(x_{t-1}|y_{0:T-1}) \approx p(x_{t-1}|y_{0:T})$.

We desire unweighted samples from the joint

$$\left\{\left(x_{t-1}^{(i)}, x_t^{(i)}\right)\right\}_{i=1}^{N} \text{ approximating } p^L(x_{t-1}, x_t|y_{0:T}),$$

where $p^L(x_{t-1}, x_t|y_{0:T}) = p(x_{t-1}|y_{0:T-1})p(x_t|x_{t-1}, y_{t:T})$. As $\left\{x_{t-1}^{(i)}\right\}_{i=1}^{N}$ have the correct marginal, obtaining the desired joint samples amounts to sampling

$$x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)}, y_{t:T}) \qquad \text{for } i = 1, \ldots, N.$$

We cannot sample this conditional density directly and (similarly to backward simulation) employ importance sampling to approximate it.

In order to use the empirical approximation of $p(x_t|y_{0:T})$ as the sampling density (for an online implementation) we need to write

$$p(x_t|x_{t-1}^{(i)}, y_{t:T}) = \frac{h_t(x_{t-1}^{(i)}, x_t)p(x_t|y_{0:T})}{\int h_t(x_{t-1}^{(i)}, x_t)p(x_t|y_{0:T})dx_t},$$

for some suitably defined non-negative and integrable function $h_t(x_{t-1}^{(i)}, x_t)$. We then use the set of samples $\left\{\tilde{x}_t^{(j)}, \tilde{w}_t^{(j)}\right\}_{j=1}^{N}$ in the numerator and also to approximate the denominator to obtain a set of normalised weights

$$w_t^{(i\rightarrow j)} = \frac{\tilde{w}_t^{(j)} h_t(x_{t-1}^{(i)}, \tilde{x}_t^{(j)})}{\sum_{k=1}^N \tilde{w}_t^{(k)} h_t(x_{t-1}^{(i)}, \tilde{x}_t^{(k)})}.$$

To find h_t,

$$\begin{aligned} p(x_t|x_{t-1}^{(i)}, y_{t:T}) &= p(x_t|x_{t-1}^{(i)}, y_{0:T}) \\ &= \frac{p(y_{t:T}|x_t)p(x_t|x_{t-1}^{(i)})}{p(y_{t:T}|x_{t-1}^{(i)})}, \\ &= \frac{p(y_{t:T}|x_t)p(x_t|x_{t-1}^{(i)})}{p(y_{t:T}|x_{t-1}^{(i)})p(x_t|y_{0:T})} p(x_t|y_{0:T}). \end{aligned}$$

where in the last line we have multiplied and divided by $p(x_t|y_{0:T})$. A further application of Bayes' theorem to $p(x_t|y_{0:T})$ gives us

$$p(x_t|x_{t-1}^{(i)}, y_{t:T}) = \frac{p(y_{t:T}|y_{0:t-1})p(x_t|x_{t-1}^{(i)})}{p(y_{t:T}|x_{t-1}^{(i)})p(x_t|y_{0:t-1})} p(x_t|y_{0:T}).$$

Since $p(x_t|x_{t-1}^{(i)}, y_{t:T})$ integrates to 1, h_t is clearly

$$h_t(x_{t-1}^{(i)}, x_t) = \frac{p(x_t|x_{t-1}^{(i)})}{p(x_t|y_{0:t-1})}$$

which is where we stop as the density $p(x_t|y_{0:t-1})$ is not tractable (note that we could replace $p(x_t|y_{0:t-1})$ with an empirical approximation using marginal filtering particles but the resulting algorithm would have a prohibitive $O(N^3)$ complexity).

C. Fixed-lag Forward Simulation - Tractable

In the previous section, the $h_t(x_{t-1}^{(i)}, x_t)$ was intractable. We address this problem by defining a new conditional density

$$\pi(x_t, x_{t-1}|x_{t-1}^{(i)}, y_{0:T}) = p(x_t|x_{t-1}^{(i)}, y_{t:T})\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t)$$

which trivially admits $p(x_t|x_{t-1}^{(i)}, y_{t:T})$ as the marginal. The potential dependency of λ on $y_{0:T}$ is implicit. The conditional density λ is to be chosen and we show how to make this choice so that when using $p(x_{t-1}, x_t|y_{0:T})$ within an importance sampling approximation of π, the weight is now tractable. In keeping with the notation of the previous section,

$$\left\{(\tilde{x}_{t-1}^{(j)}, \tilde{x}_t^{(j)}), \tilde{w}_t^{(j)}\right\}_{j=1}^{N} \text{ approximates } p(x_{t-1}, x_t|y_{0:T}),$$

where we have used tildes to represent the particles from which we look to resample - non-tilde particles are fixed. We remark that this idea of resolving the intractability of weights via sampling a higher dimensional density is inspired by the work of [9] for blocked resampling of the path-space particle approximations.

Proposition 1. *Let* $\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t) = p(x_{t-1}|y_{0:t-1})$ *and* $H(x_{t-1}^{(i)}, x_{t-1}, x_t) = p(x_t|x_{t-1}^{(i)})/p(x_t|x_{t-1})$ *then*

$$\begin{aligned}\pi(x_t, x_{t-1}&|x_{t-1}^{(i)}, y_{0:T}) \\ &= \frac{H(x_{t-1}^{(i)}, x_{t-1}, x_t)p(x_t, x_{t-1}|y_{0:T})}{\int H(x_{t-1}^{(i)}, x_{t-1}, x_t)p(x_t, x_{t-1}|y_{0:T})dx_{t-1:t}}.\end{aligned}$$

The self-normalised approximation of $p(x_t|x_{t-1}^{(i)}, y_{0:T})$ *is* $\left\{\tilde{x}_t^{(j)}, \tilde{w}_t^{(i\to j)}\right\}$ *where*

$$w_t^{(i\to j)} = \tilde{w}_t^{(j)}\frac{p(\tilde{x}_t^{(j)}|x_{t-1}^{(i)})}{p(\tilde{x}_t^{(j)}|\tilde{x}_{t-1}^{(j)})}\left(\sum_{k=1}^{N}\tilde{w}_t^{(k)}\frac{p(\tilde{x}_t^{(k)}|x_{t-1}^{(i)})}{p(\tilde{x}_t^{(k)}|\tilde{x}_{t-1}^{(k)})}\right)^{-1}. \tag{7}$$

The interpretation of this proposition for forward smoothing is as follows: we can now sample from this approximation of π directly. Discarding the sampled x_{t-1} leaves samples $(x_{t-1}^{(i)}, x_t^{(i)})$ from the desired joint $p^L(x_{t-1}, x_t|y_{0:T})$. Repeating this for each i results in an $O(N^2)$ algorithm that is asymptotically unbiased for the fixed-lag joint distribution.

Proof. Expanding the density π yields

$$\begin{aligned}\pi(x_t, x_{t-1}&|x_{t-1}^{(i)}, y_{0:T}) \\ &= p(x_t|x_{t-1}^{(i)}, y_{t:T})\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t), \\ &= \frac{p(y_{t:T}|x_t)p(x_t|x_{t-1}^{(i)})}{p(y_{t:T}|x_{t-1}^{(i)})}\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t).\end{aligned}$$

Dividing by the sampling distribution $p(x_{t-1}, x_t|y_{0:T}) = p(x_{t-1}|y_{0:T})p(x_t|x_{t-1}, y_{t:T})$ gives

$$\begin{aligned}\pi(x_t, x_{t-1}&|x_{t-1}^{(i)}, y_{0:T})/p(x_{t-1}, x_t|y_{0:T}) \\ &= \frac{p(y_{t:T}|x_t)p(x_t|x_{t-1}^{(i)})}{p(y_{t:T}|x_{t-1}^{(i)})p(x_{t-1}, x_t|y_{0:T})}\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t).\end{aligned}$$

Bayes' theorem on $p(x_{t-1}, x_t|y_{0:T})$ gives

$$\begin{aligned}\pi(x_t, x_{t-1}&|x_{t-1}^{(i)}, y_{0:T})/p(x_{t-1}, x_t|y_{0:T}) \\ = &\frac{p(x_t|x_{t-1}^{(i)})p(y_{t:T}|y_{0:t-1})}{p(x_t|x_{t-1})p(y_{t:T}|x_{t-1}^{(i)})p(x_{t-1}|y_{0:t-1})}\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t).\end{aligned}$$

We now observe that the choice of $\lambda(x_{t-1}|x_{t-1}^{(i)}, x_t) = p(x_{t-1}|y_{0:t-1})$ will make all terms involving (x_{t-1}, x_t) tractable.

$$\begin{aligned}\pi(x_t, x_{t-1}&|x_{t-1}^{(i)}, y_{0:T}) \\ &= \frac{p(x_t|x_{t-1}^{(i)})p(y_{t:T}|y_{0:t-1})}{p(x_t|x_{t-1})p(y_{t:T}|x_{t-1}^{(i)})}p(x_{t-1}, x_t|y_{0:T}).\end{aligned}$$

□

Algorithm 3 Fixed-lag Particle Stitching

```
Input samples ${x_{0:T-L-1}^{(i)}}_{i=1}^N$ and ${\tilde{x}_{T-L-1:T}^{(j)}, \tilde{w}_T^{(j)}}_{j=1}^N$, bound $\rho \geq p(x_{T-L}|x_{T-L-1})$ and the maximum number of rejections to attempt $R$.
Calculate the non-interacting stitching weights and normalise in $j$
    $\hat{w}_T^{(j)} \propto \tilde{w}_T^{(j)}/p(\tilde{x}_{T-L}^{(j)}|\tilde{x}_{T-L-1}^{(j)})$    $j = 1, \ldots, N.$
for $i = 1, \ldots, N$ do
    for $r = 1, \ldots, R$ do
        Sample $c^* \sim$ Categorical(${\hat{w}_T^{(j)}}_{j=1}^N$)
        Sample $u \sim U(0, 1)$
        if $u < p(\tilde{x}_{T-L}^{(c^*)}|x_{T-L-1}^{(i)})/\rho$ then
            Accept $c^*$
            break
        end if
    end for
    if a sample $c^*$ was accepted then
        Set $x_{T-L:T}^{(i)} = \tilde{x}_{T-L:T}^{(c^*)}$
    else
        Calculate the stitching weights and normalise in $j$
            $w_T^{(i\to j)} \propto \tilde{w}_T^{(j)}\frac{p(\tilde{x}_{T-L}^{(j)}|x_{T-L-1}^{(i)})}{p(\tilde{x}_{T-L}^{(j)}|\tilde{x}_{T-L-1}^{(j)})}$    $j = 1, \ldots, N.$
        Sample $c_i \sim$ Categorical(${w_T^{(i\to j)}}_{j=1}^N$)
        Set $x_{T-L:T}^{(i)} = \tilde{x}_{T-L:T}^{(c_i)}$
    end if
end for
Output sample ${x_{0:T}^{(i)}}_{i=1}^N$ approximating $p^L(x_{0:T}|y_{0:T})$.
```

In (7), we have described above an empirical approximation that stitches together particles from $p(x_{T-L-1}|y_{0:T-1})$ with those from $p(x_{T-L-1:T-L}|y_{0:T})$. By the conditional independence structure of state-space models this is equivalent to stitching together blocks from $p^L(x_{0:T-L-1}|y_{0:T-1})$ with those from $p(x_{T-L-1:T}|y_{0:T})$ - assuming we can sample from $p(x_{T-L-1:T}|y_{0:T})$.

Sampling from $p(x_{T-L-1:T}|y_{0:T})$ is not directly possible for non-trivial state-space models. In Section IV we describe two efficient methods for block sampling by recycling previously generated particles (Alg. 4 and Alg. 5).

The fixed-lag particle stitching described provides a method to, in principal, approximate the full smoothing distribution $p(x_{0:T}|y_{0:T})$. It does so by using very separate tools to the forward-only techniques in [5], [6], these techniques do not require any stitching and directly update an expectation over additive functionals. Our method generates a collection of particles approximating the joint smoothing distribution and is therefore far more general. We have utilised the block sampling framework from [9], in particular through the balancing distribution λ. However, the particle stitching is novel and effectively uses a fixed-lag approximation to implement a forward implementation of the technique underlying backward simulation [8]. This is in contrast to [9], where particles are reweighted and resampled over full trajectories and therefore still suffer from path degeneracy.

D. Rejection Sampling

Sampling from (7) can be done directly at a computational complexity of $O(N^2)$. However, when a bound for the transition density is available

$$\rho \geq p(x_{T-L}|x_{T-L-1}) \quad \forall x_{T-L-1}, x_{T-L}, \tag{8}$$

we can utilise the rejection sampling approach of [10] to avoid calculating the normalisation constants and bring the computational complexity down to $O(N)$. In practice, the rejection sampling is not always faster than the direct one. A pragmatic approach is the hybrid described in [21] where for each particle, up to $R < N$ samples are proposed to the rejection sampler, if all are subsequently rejected the direct scheme is applied, thus setting $R = 0$ recovers the direct scheme. This hybrid algorithm is described in Alg. 3.

IV. Sampling from $p(x_{T-L-1:T}|y_{0:T})$

We now describe two methods for sampling coordinates $\tilde{x}_{T-L-1:T}$ in a way that is asymptotically unbiased for $p(x_{T-L-1:T}|y_{0:T})$, and can therefore be plugged into the fixed-lag particle stitching procedure.

A. Particle Filter

Recall the online setting where we have unweighted particles $\{x^{(i)}_{0:T-1}\}_{i=1}^N$ approximating $p^L(x_{0:T-1}|y_{0:T-1})$ and receive a new observation y_T.

Our first method is based on the fact that the particle approximation provided by a classical particle filter is asymptotically unbiased for the full joint smoothing distribution $p(x_{0:T}|y_{0:T})$. Although this approximation deteriorates due to path degeneracy it may still be sufficient for sampling the recent coordinates $x_{T-L-1:T}$.

Thus, we propose applying the classical particle filter proposal and weighting steps to $\{x^{(i)}_{0:T-1}\}_{i=1}^N$, generating weighted particles $\{x^{(i)}_{0:T}, \tilde{w}^{(i)}_T\}_{i=1}^N$, before splitting the trajectories

$$\left\{x^{(i)}_{0:T}, \tilde{w}^{(i)}_T\right\}_{i=1}^N \\ \rightarrow \left\{x^{(i)}_{0:T-L-1}\right\}_{i=1}^N \text{ and } \left\{\tilde{x}^{(i)}_{T-L-1:T}, \tilde{w}^{(i)}_T\right\}_{i=1}^N,$$

where $\tilde{x}^{(i)}_{T-L-1:T} = x^{(i)}_{T-L-1:T}$. Under the fixed-lag approximation, $x_{0:T-L-1}$ is conditionally independent of y_T and therefore the new weights need not apply to these earlier coordinates. Whereas the $\{\tilde{x}^{(i)}_{T-L-1:T}, \tilde{w}^{(i)}_T\}_{i=1}^N$ are asymptotically unbiased for the desired sampling distribution $p(x_{T-L-1:T}|y_{0:T})$ and can therefore be plugged into the stitching procedure, Alg. 3. The coordinates x_{T-L-1} are duplicated to provide the overlap required for stitching.

Algorithm 4 Online Particle Smoother (for $T > L$)

```
0: Input smoothing sample $\{x^{(i)}_{0:T-1}\}_{i=1}^N$ and new observation $y_T$.
1: Fix $\{x^{(i)}_{0:T-L-1}\}_{i=1}^N$
2: Execute particle filter propagate and weight steps, Alg. 1: steps 2:3
       Generate $\{\tilde{x}^{(j)}_{T-L-1:T}, \tilde{w}^{(j)}_T\}_{j=1}^N$
               from $\{x^{(j)}_{T-L-1:T-1}\}_{j=1}^N$ and $y_T$,
   forming a weighted sample approximating $p(x_{T-L-1:T}|y_{0:T})$.
3: Stitch together
   $\{x^{(i)}_{0:T-L-1}\}_{i=1}^N$ and $\{\tilde{x}^{(j)}_{T-L-1:T}, \tilde{w}^{(j)}_T\}_{j=1}^N \rightarrow \{x^{(i)}_{0:T}\}_{i=1}^N$.
   using Alg. 3.
4: Output sample $\{x^{(i)}_{0:T}\}_{i=1}^N$ approximating $p^L(x_{0:T}|y_{0:T})$
```

The algorithm is described in Alg. 4, and ends up being a relatively simple modification to a classical particle filter where the resampling step is compulsory and altered to include the stitching probabilities in the weights (7).

When the transition bound (8) is available the complexity of the update remains $O(N)$ or $O(N^2)$ when the bound is unavailable.

B. Partial Backward Simulation

If the lag parameter L is chosen to be too large, the above mechanism will still suffer from path degeneracy in the same way a particle filter does. To remedy this we propose a partial run of backward simulation (Alg. 6) at each time step to rejuvenate the trajectories $x_{T-L-1:T}$. This technique is considered in [22] for generating samples from $p(x_{T-L:T}|y_{0:T})$ without the subsequent stitching.

This has the additional requirement of storing the marginal approximations $\{\tilde{x}^{(k)}_t, \tilde{w}^{(k)}_t\}_{k=1}^N$ for $t = T-L-1, \ldots, T$ from

Algorithm 5 Online Particle Smoother with Backward Simulation (for $T > L$)

```
0: Input smoothing sample $\{x_{0:T-1}^{(i)}\}_{i=1}^N$, marginal filtering
   samples $\{\tilde{x}_t^{(k)}, \tilde{w}_t^{(k)}\}_{k=1}^N$ for $t = T-L-1, \dots, T-1$ and
   new observation $y_T$.
1: Fix $\{x_{0:T-L-1}^{(i)}\}_{i=1}^N$
2: Execute particle filter update, Alg. 1, to generate the new
   marginal filtering sample
       Generate $\{\tilde{x}_T^{(k)}, \tilde{w}_T^{(k)}\}_{k=1}^N$
                 from $\{\tilde{x}_{T-1}^{(k)}, \tilde{w}_{T-1}^{(k)}\}_{k=1}^N$ and $y_T$.
3: Run partial backward simulation, Alg. 6, on the weighted
   filtering samples
       $\{\tilde{x}_t^{(k)}, \tilde{w}_t^{(k)}\}_{k=1}^N$, for $t = T, \dots, T-L-1$
                 $\rightarrow \{\tilde{x}_{T-L-1:T}^{(j)}\}_{j=1}^N$.
   forming an unweighted sample approximating
   $p(x_{T-L-1:T} | y_{0:T})$.
4: Stitch together
       $\{x_{0:T-L-1}^{(i)}\}_{i=1}^N$ and $\{\tilde{x}_{T-L-1:T}^{(j)}, \frac{1}{N}\}_{j=1}^N \rightarrow \{x_{0:T}^{(i)}\}_{i=1}^N$.
   using Alg. 3.
5: Output sample $\{x_{0:T}^{(i)}\}_{i=1}^N$ approximating $p^L(x_{0:T}|y_{0:T})$
   and filtering samples $\{\tilde{x}_T^{(k)}, \tilde{w}_T^{(k)}\}_{k=1}^N$, $t = T-L, \dots, T$.
```

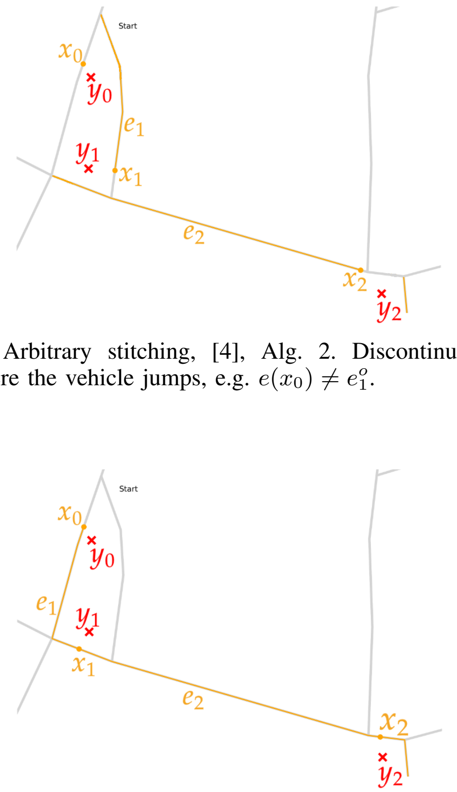
the particle filter, but permits the use of adaptive resampling and completely avoids path degeneracy.

The resulting algorithm, Alg. 5, has a complexity of $O(LN)$ per update if the transition bound (8) is available otherwise $O(LN^2)$.

Both techniques to sample from $p(x_{T-L-1:T}|y_{0:T})$ utilise the output of a particle filter. Indeed, they are also directly applicable to alternative filtering techniques such as auxiliary particle filters [23] and the backward simulation approach is also applicable to filters that discard historic trajectories such as the marginal particle filter [24]. In fact, Proposition 1 is very general and can be applied to any collection of particles that have the correct marginals - including more sophisticated approaches that resample using information from both the weights and the samples [15], [25].

V. Map-Matching

Map-matching is the task of inferring the true trajectory of a vehicle given noisy GPS observations and a map of the road network. A road network is defined as a graph within $\mathbb{R}^2$, where intersections are represented by nodes (vertices) and roads (assumed to be single lanes and one-way) are represented by edges with a two-way road being represented by two edges. Some collections of nodes and edges are depicted in Fig. 1, 3 and 5, with map-matched vehicle routes overlaid. The recorded GPS observations (see Fig. 1) may lie outside the road whereas valid vehicle positions/trajectories to be inferred must strictly lie on the road. Following [26], [27] for urban road networks we infer the vehicle's position along an edge but not it's width within the road.



(a) Arbitrary stitching, [4], Alg. 2. Discontinuities where the vehicle jumps, e.g. $e(x_0) \neq e_1^o$.

(b) Fixed-lag particle stitching, Alg. 3

Fig. 1: Comparison of fixed-lag techniques for map-matching. Each image displays a single trajectory from the resulting particle approximation. Both techniques produce plausible marginals but arbitrary stitching fails to produce a continuous trajectory.

Applications of map-matching are wide-ranging and thus a general purpose algorithm is highly desirable. Existing probabilistic approaches to map-matching have mostly adopted the approach of [26], where each observation is snapped to the nearest point on any and all edges that fall within a truncation distance. These points then form a discrete hidden Markov model, on which the Viterbi algorithm can produce a single route of high probability. In contrast, our approach provides uncertainty quantification through a collection of particles, with each particle representing a possible route. Previous applications of particle filtering to map-matching [28], [29] fail to tackle the problem of path degeneracy, with the exception of [27] who introduce the use of FFBSi for offline map-matching. Our formulation is similar to that of [27] but differs through the inclusion of a term in the transition density adapted from [26] that penalises non-direct routes (which is vital in dense urban road networks), as well as the use of the optimal proposal density (that takes into account the new observation) rather than simply the bootstrap proposal which will perform poorly for small GPS noise or dense road networks.

Traditional fixed-lag smoothing [4], Alg. 2 where particles are arbitrarily stitched together is not appropriate for map-matching, as seen in Figure 1. As these methods target the product of smoothing marginals in (3), arbitrary stitching

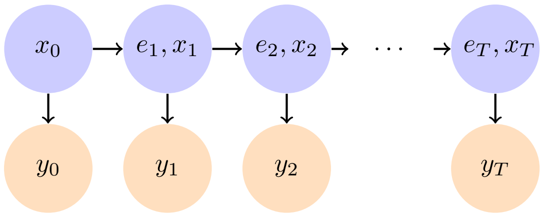


Fig. 2: Conditional independence structure of the state-space model for map-matching.

cannot produce a continuous trajectory and therefore does not permit expectations over multiple observation times.

In the rest of this section, we describe our state-space model for map-matching, the optimal proposal distribution and the induced weights in the context of fixed-lag particle stitching. In the next section we present some results on synthetic and real data. An open source python package providing easy offline and online map-matching is provided [1] alongside code for all the simulations to follow.

A. Model Variables

We define a state-space model for a single vehicle's trajectory with the variables

- $e_t \subset \mathbb{N}$, a finite ordered set of edge labels that define a connected path. Each edge label corresponds to a unique one-way section of road. The variable e_t details the edges traversed (and in which order) between observation times $t-1$ and t, including the choices made at encountered intersections (nodes).
- $x_t \in \mathbb{R}^2$ the position of the vehicle at observation time. The variable x_t defines a Cartesian coordinate restricted to lie on the road network, specifically x_t lies on the final edge of the finite ordered set of edge labels e_t, for $t > 0$.
- $y_t \in \mathbb{R}^2$ noisy observation of the vehicle's position x_t, not restricted to the road network.

Note here the change of notation from the previous section, now x_t refers only to vehicle position and the full latent states are $x_0, (e_1, x_1), (e_2, x_2), \dots, (e_T, x_T)$ as depicted in Fig. 2.

We denote e_t^o for the first edge in the ordered set e_t and e_t^* for the final edge, thus $e_t = (e_t^o, \dots, e_t^*)$. We use the notation $e(x_t)$ to denote the edge label on which x_t lies, so $e(x_{t-1}) = e_t^o$ and $e(x_t) = e_t^*$.

B. Model Distributions

Our transition density can be written as

$$\begin{aligned} p(e_t, x_t | x_{t-1}) = &\gamma(\|x_t - x_{t-1}\|_{e_t}) \\ &\exp(-\beta | \|x_t - x_{t-1}\|_{e_t} - \|x_t - x_{t-1}\| |) \\ &Z(x_{t-1})^{-1}, \end{aligned} \tag{9}$$

with normalising constant

$$\begin{aligned} Z(x_{t-1}) = &\sum_{e_t} \int_{x_t} \gamma(\|x_t - x_{t-1}\|_{e_t}) \\ &\exp(-\beta | \|x_t - x_{t-1}\|_{e_t} - \|x_t - x_{t-1}\| |) \mathrm{d}x_t. \end{aligned} \tag{10}$$

The summation in (10) is taken over all possible series of edges starting at $e(x_{t-1})$. $\|x_t - x_{t-1}\|_{e_t}$ is the distance travelled between x_{t-1} and x_t along the series e_t (restricted to the road network) whereas $\|x_t - x_{t-1}\|$ is the *great circle* distance (not restricted to the road network).

Thus the following distributions fully define our state-space model:

- $\gamma(\|x_t - x_{t-1}\|_{e_t})$. Prior on distance travelled between observations - some simple analytical distribution on $\mathbb{R}^+$, penalising lengthy routes. We assume an exponential distribution with probability mass at 0 to represent the possibility of the vehicle remaining stationary (due to traffic lights, heavy traffic etc)
$$\gamma(d) = p^0 \mathbb{I}(d = 0) + (1 - p^0) \mathbb{I}(d > 0) \lambda e^{-\lambda d}. \tag{11}$$
- $\exp(-\beta | \|x_t - x_{t-1}\|_{e_t} - \|x_t - x_{t-1}\| |)$ adapted from [26], penalising non-direct (or winding) routes. Non-direct routes with lots of curvature will have a high discrepancy between the road distance travelled and great circle distance and thus will have a low probability under this term, reflecting a driver's preference to take short, direct routes where possible.
- $p(y_t | x_t) = \mathcal{N}(y_t | x_t, \sigma^2_{\text{GPS}} I_2)$. Isotropic Gaussian observation noise.
- We set $p(x_0)$ to be uniform on the road network. I.e. no prior information on the start of the trajectory other than constricting it to the road network (as with all inferred positions).

To make $p(x_0 | y_0)$ tractable we define the initial observation density to be a truncated Gaussian:

$$p(y_0 | x_0) \propto \mathcal{N}(y_0 | x_0, \sigma^2_{\text{GPS}} I_2) \, \mathbb{I}(||y_0 - x_0|| < r_{\text{GPS}}),$$

giving

$$p(x_0 | y_0) \propto \mathcal{N}(x_0 | y_0, \sigma^2_{\text{GPS}} I_2) \, \mathbb{I}(||y_0 - x_0|| < r_{\text{GPS}}),$$

where x_0 is restricted to the road network but y_0 is not. In our simulations we set $r_{\text{GPS}} = 5\sigma_{\text{GPS}}$.

C. Optimal Proposal

The (locally) optimal proposal [7] for particle filtering combines the transition density (9) and the newly received observation y_T:

$$q^{\text{opt}}(x_T, e_T | x_{T-1}, y_T) \propto p(e_T, x_T | x_{T-1}) p(y_T | x_T). \tag{12}$$

The standard reweighting step of the particle filter update (Alg. 1) then becomes

$$w_T^{\text{opt}\,(i)} \propto p(y_T | x_{T-1}^{(i)}),$$

where

$$p(y_T | x_{T-1}) = \sum_{e_T} \int_{x_T} p(e_T, x_T | x_{T-1}) p(y_T | x_T) \, \mathrm{d}x_T, \tag{13}$$

is the normalisation constant of (12).

Neither sampling from the optimal proposal (12), nor evaluating the subsequent weights (13), nor evaluating the normalising constant of the transition density (10) are immediately

[1] https://github.com/SamDuffield/bmm

tractable as we do not have closed form expressions for the edge geometries.

Instead, we opt to approximate the required integrals numerically by discretising the edges up to some maximal possible distance travelled d_{max}. This numerical integration can be implemented efficiently across particles by caching route searches and likelihood evaluations, as many particles will typically lie on the same or adjacent edges.

D. Fixed-lag Particle Stitching

In the context of the particle stitching described in Section III, we propose stitching together each

$$(x^{(i)}_{0:T-L-1}, e^{(i)}_{1:T-L-1}) \quad \text{with a sample from } \left\{(\tilde{x}^{(j)}_{T-L-1:T}, \tilde{e}^{(j)}_{T-L:T})\right\}_{j=1}^{N}.$$

Thus the adjusted weights become

$$w_T^{(i\to j)} \propto \frac{p(\tilde{e}^{(j)}_{T-L}, \tilde{x}^{(j)}_{T-L} | x^{(i)}_{T-L-1})}{p(\tilde{e}^{(j)}_{T-L}, \tilde{x}^{(j)}_{T-L} | \tilde{x}^{(j)}_{T-L-1})} w_T^{(j)} \mathbb{I}\left(e(x^{(i)}_{T-L-1}) = \tilde{e}^{o\,(j)}_{T-L}\right).$$

Similarly, for the rejection sampling we get non-interacting weights

$$\hat{w}_T^{(j)} \propto w_T^{(j)} / p(\tilde{e}^{(j)}_{T-L}, \tilde{x}^{(j)}_{T-L} | \tilde{x}^{(j)}_{T-L-1}),$$

and we accept a sample if $e(x^{(i)}_{T-L-1}) = \tilde{e}^{o\,(j)}_{T-L}$ and

$$\begin{aligned} u <& \gamma\left(\|\tilde{x}^{(j)}_{T-L} - x^{(i)}_{T-L-1}\|_{\tilde{e}^{(j)}_{T-L}}\right) \\ &\exp\left(-\beta\left|\|\tilde{x}^{(j)}_{T-L} - x^{(i)}_{T-L-1}\|_{\tilde{e}^{(j)}_{T-L}} - \|\tilde{x}^{(j)}_{T-L} - x^{(i)}_{T-L-1}\|\right|\right) \\ &\rho^{-1}, \end{aligned}$$

where $u \sim U(0,1)$ and we have a bound $\rho > \gamma(d)$ for any d. The availability of this bound depends on the choice of distribution for $\gamma(\cdot)$, in the case of (11) a bound is available: $\rho = \max((1-p^0)\lambda, p^0)$.

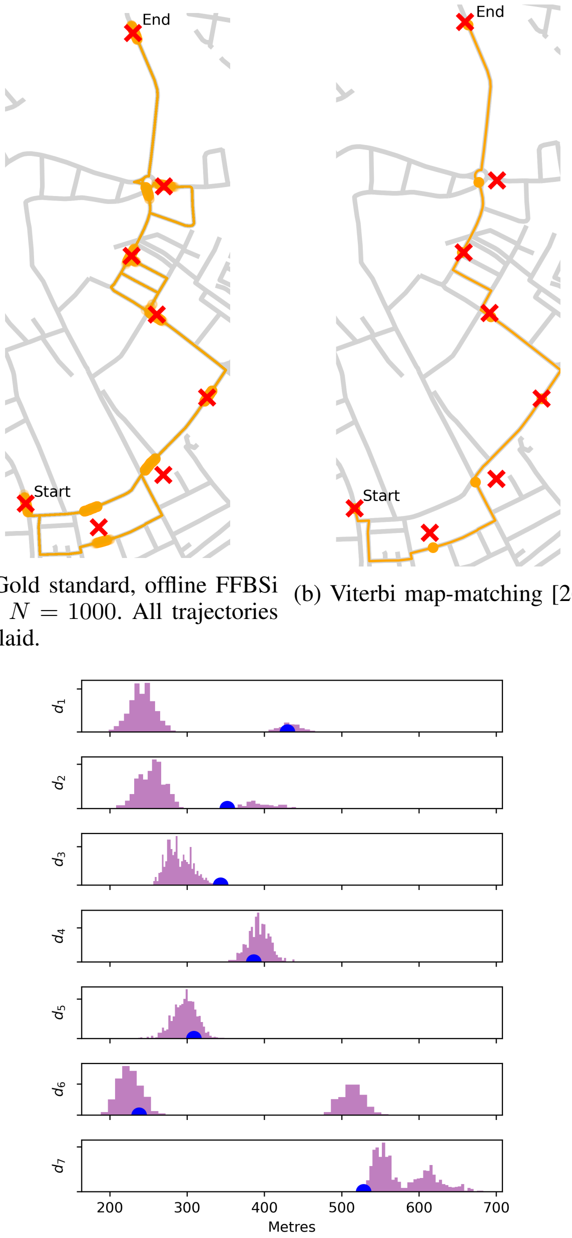
VI. Simulations

We tuned the model hyperparameters using offline gradient expectation-maximisation [19] (running offline FFBSi for the E-step) over 20 routes from the Porto taxi dataset [30] where observations are 15 seconds apart, resulting in values of $p^0 = 0.14$, $\lambda = 0.07/15$, $\beta = 0.05$ and $\sigma_{\text{GPS}} = 5.2$, all edges are discretised to a resolution of 1m.

The true map-matching posterior is analytically intractable and instead we use the approach of [27] as our *gold standard* for benchmarking: offline forward filtering-backward simulation with a large $N = 1000$, i.e. Alg. 1 to generate the filtering marginals and Alg. 6 for the smoothing particles.

A. Synthetic Data

In Fig. 3 we demonstrate the benefits of uncertainty quantification for offline map-matching by comparing our particle based gold standard (offline FFBSi) which is the approach of [27] against the popular optimisation approach of [26] on synthetic observations for a trip between the Cambridge Engineering department and the Fort St George pub.



(a) Gold standard, offline FFBSi with $N = 1000$. All trajectories overlaid.

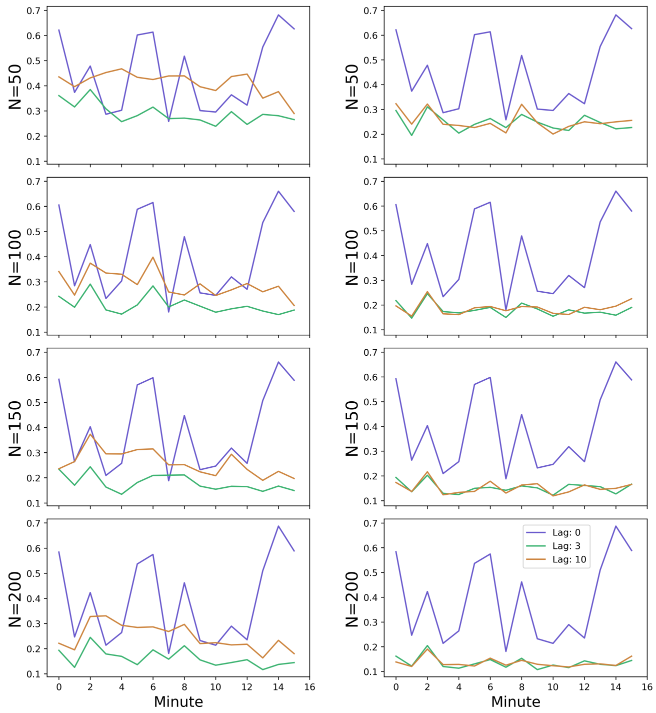
(b) Viterbi map-matching [26].

(c) Histograms represent $p(\|x_t - x_{t-1}\|_{e_t} | y_{0:T})$ from FFBSi. Spots represent distances inferred using Viterbi map-matching.

Fig. 3: Offline particle smoothing vs Viterbi map-matching for synthetic trip across Cambridge.

Although the optimisation based approach finds a plausible route (point estimate) it misses out on others that are equally plausible and thus valuable information is lost, this is particularly prevalent when inferring the distances the vehicle travelled, Fig. 3c. There is significant uncertainty in both the edges traversed and the distances travelled that is captured by FFBSi but not by the Viterbi algorithm.

All particles generated by FFBSi represent a plausible route - more direct routes are preferred but not overly so - this provides evidence to suggest the model is well suited to difficult dense urban road networks and that the optimal proposal is efficiently generating high probability particles.



(a) Online Particle Smoother

(b) with Backward Simulation.

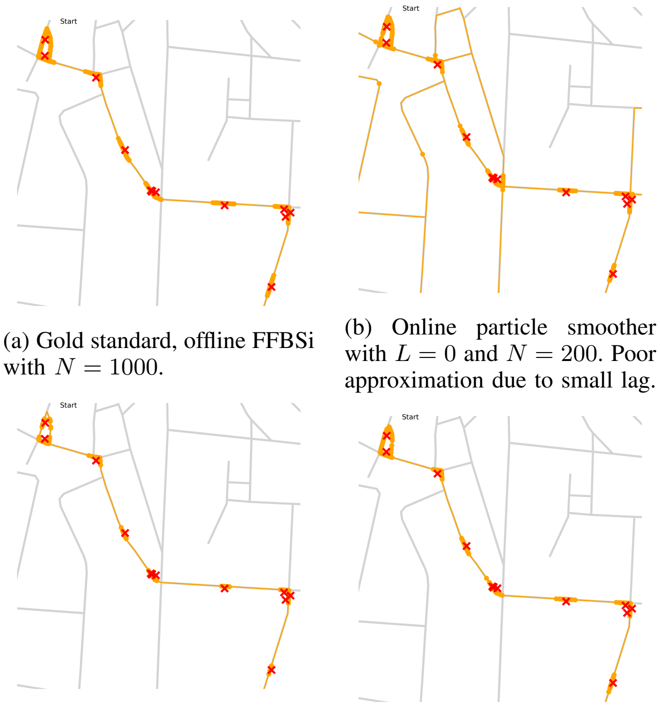
Fig. 4: Total variation distance from gold standard (offline FFBSi) for posterior over cumulative distance travelled in each minute of a 16 minute taxi route (observation every 15 seconds) [30]. Simulations averaged over 20 random seeds.

In the problem considered here and in [26], [27] there is no variability in the position of the vehicle along the width of the road - arguably more suitable for dense, urban networks. A less constrained inference problem in e.g. [31] would also infer the position along the width of the road. The model can then be formulated with Gaussian noise allowing Kalman filter or Rao-Blackwellisation techniques (that are not applicable to our state-space model) can be used to accelerate inference.

B. Real Data

We now explore the effects of the algorithmic parameters in the online particle smoothers (sample size, lag parameter and number of rejections) for map-matching a route from the Porto taxi dataset [30].

In Fig. 4, we analyse the posterior distribution for the cumulative (road) distance travelled by the taxi each minute. As observations are received every 15 seconds this amounts to expectations averaged over blocks of the full smoothing distribution and thus marginal approximations (3) would be insufficient. For each minute of the trip, we compare particle approximations by calculating the total variation distance over the empirical distributions. This total variation distance is calculated by binning the distance travelled variable, then the empirical distributions are defined over a discrete space and the total variation distance is tractable



(a) Gold standard, offline FFBSi with $N = 1000$.

(b) Online particle smoother with $L = 0$ and $N = 200$. Poor approximation due to small lag.

(c) Online particle smoother, $L = 10$ and $N = 200$. Evidence of path degeneracy due to large lag.

(d) Online particle smoother with backward simulation, $L = 10$ and $N = 200$.

Fig. 5: The start of a route from [30] with various particle approximations.

$$\mathrm{TV}\left(\{d_1^{(i)}\}_{i=1}^{N_1}, \{d_2^{(j)}\}_{j=1}^{N_2}\right)$$
$$= \frac{1}{2} \sum_{[a,b]\in\mathcal{D}} \left| \frac{1}{N_1} \sum_{i=1}^{N_1} \mathbb{I}[d_1^{(i)} \in [a,b]] - \frac{1}{N_2} \sum_{j=1}^{N_2} \mathbb{I}[d_2^{(j)} \in [a,b]] \right|,$$

where $\mathcal{D}$ represents $[0, \infty)$ discretised into 5 metre width bins and the empirical distances are assumed to be unweighted (although easily adjusted to include weights).

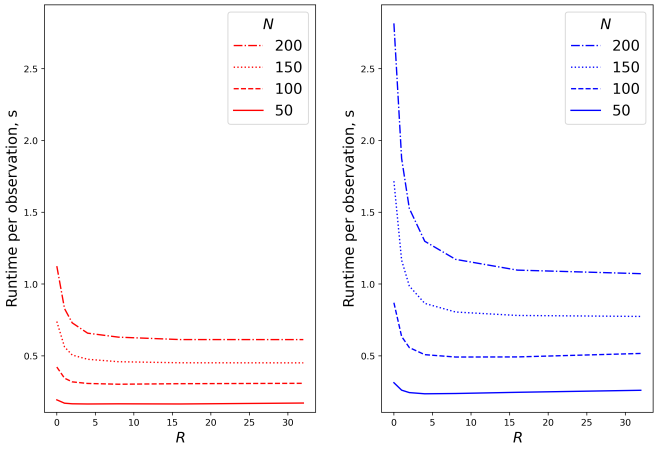
Fig. 5 depicts the varying algorithmic performance on the start of the route.

We initially observe that setting $L = 0$ performs very poorly for all sample sizes. The overly small lag parameter results in a large deviation between the true smoothing distribution (1) and the joint fixed-lag smoothing distribution (4), Fig. 5b.

The online particle smoother suffers from path degeneracy for the large lag $L = 10$ as observed by a loss of particle diversity (compared to the backward simulation techniques) in Fig. 5c. It does however perform well for $L = 3$.

The addition of backward simulation completely avoids the issue of path degeneracy, but induces a bias by targeting $p^L(x_{0:T}|y_{0:T})$ rather than $p(x_{0:T}|y_{0:T})$, this bias is controllable through the choice of the lag parameter, L. We observe that increasing the lag parameter from $L = 3$ to $L = 10$ does little to improve performance and as such can posit that the distributions (1) and (4) are suitably close for $L = 3$.

We have not compared numerically to the marginal or additive functional based smoothers described in section II



(a) Online Particle Smoother (b) with Backward Simulation.

Fig. 6: Algorithmic runtime vs number of rejections tolerated for $L = 3$. Runtimes averaged over 65 observations and 20 random seeds.

(such as PaRIS [6]) as by definition (6) they are not suitable to expectations over multiple observation times as analysed in Fig. 4 and depicted in Figure 1.

A classical particle filter can be recovered by setting $L = \infty$ in the online particle smoother and would suffer path degeneracy to an even greater extent than the online particle smoother with $L = 10$.

Finally, in Fig. 6 we compare the effect on algorithmic runtime from increasing the maximum number of rejections, R, attempted in the hybrid stitching scheme Alg. 3, as well as backward simulation if applicable. Recall that setting $R = 0$ recovers the full $O(N^2)$ scheme. For a suitably large number of rejections the runtimes of both algorithms can be seen to increase linearly in N.

VII. Discussion

In this work, we have developed techniques to efficiently approximate the full joint smoothing distribution or rather the fixed-lag approximation to it in an online setting, this is highly desirable as it permits the online estimation of a range functions that are defined over full trajectories or any subset thereof. The online particle smoother (Alg. 4) comes at the same computational complexity as a classical particle filter, whereas the inclusion of backward simulation (Alg. 5) negates the issue of path degeneracy for more difficult models where a large lag parameter is required.

We formulated a state-space model that is specifically designed for urban map-matching and demonstrated the value of particle based uncertainty quantification versus established optimisation based approaches. We then showed that the performance of gold standard offline smoothing FFBSi can be obtained with the online particle smoothers.

The choice of the lag parameter L determines the distance between the distributions $p^L(x_{0:T}|y_{0:T})$ and $p(x_{0:T}|y_{0:T})$, naturally we desire this to be small and therefore L large. The question of how large is a difficult one as it is dependent on the mixing of the state-space model, as discussed in [6]. In practice, the lag parameter can be tuned through preliminary runs on offline or simulated data. This can be done by analysing the sensitivity of smoothing expectations of interest (model-specific) to the value of the lag parameter L and comparing against the true underlying values (in the case of simulated data) or the equivalent expectation from FFBSi. This is investigated for map-matching in Figure 4. The tuning of L is perhaps easier for the online particle smoother with backward simulation. The backward simulation completely avoids the issue of path degeneracy and thus error only arises from Monte Carlo variance and the difference between $p^L(x_{0:T}|y_{0:T})$ and $p(x_{0:T}|y_{0:T})$. An interesting extension would be to investigate the use of a variable lag parameter which is chosen dynamically, as achieved for a function specific version of the marginal fixed-lag particle filter in [32].

It would be desirable to obtain theoretical results bounding the error induced by the introduced online particle smoothers. We leave this for future work as we anticipate the analysis to be somewhat intricate - combining the work on central limit theorems for particle smoothing such as [5], [10] and the bias induced by a fixed-lag approximation [2]. As well as quantifying the impact of using the tractable weights with overlapping coordinate from III-C as opposed to the optimal but intractable weights in III-B.

The realisation of a low-probability transition or observation in the true underlying process can cause degeneracy either at stitching time or in the filtering weights. It is possible to reintroduce particle diversity by applying an MCMC kernel after stitching as in resample-move particle filters [33] or particle rejuvenation [34]. The MCMC kernel is applied independently to each particle and must be invariant for the full smoothing distribution $p(x_{0:T}|y_{0:T})$. As we are only concerned with increasing particle diversity rather than taking ergodic averages we need only propose moving a subset of the trajectories, whether that be at stitching time or the latest observation time.

References

[1] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*, ser. Springer Series in Statistics. Springer International Publishing, 2020.

[2] J. Olsson, O. Cappé, R. Douc, and Éric Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models," *Bernoulli*, vol. 14, no. 1, pp. 155 – 179, 2008.

[3] G. Poyiadjis, A. Doucet, and S. S. Singh, "Particle approximations of the score and observed information matrix in state space models with application to parameter estimation," *Biometrika*, vol. 98, no. 1, pp. 65–80, 02 2011.

[4] G. Kitagawa and S. Sato, *Monte Carlo Smoothing and Self-Organising State-Space Model*. New York, NY: Springer New York, 2001, pp. 177–195.

[5] P. Del Moral, A. Doucet, and S. S. Singh, "A backward particle interpretation of Feynman-Kac formulae," *ESAIM: M2AN*, vol. 44, no. 5, pp. 947–975, 2010.

[6] J. Olsson and J. Westerborn, "Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm," *Bernoulli*, vol. 23, no. 3, pp. 1951–1996, 08 2017.

[7] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul. 2000.

[8] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

[9] A. Doucet, M. Briers, and S. Senecal, "Efficient block sampling strategies for sequential Monte Carlo methods," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 693–711, 2006.

[10] R. Douc, A. Garivier, E. Moulines, and J. Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *The Annals of Applied Probability*, vol. 21, no. 6, pp. 2109–2145, 2011.

[11] L. Martino, V. Elvira, and F. Louzada, "Effective sample size for importance sampling based on discrepancy measures," *Signal Processing*, vol. 131, pp. 386–401, 2017.

[12] V. Elvira, L. Martino, and C. P. Robert, "Rethinking the effective sample size," *arXiv: Computation*, 2018.

[13] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, 2017.

[14] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, 2005, pp. 64–69.

[15] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, 2015.

[16] A. Koppel, A. S. Bedi, V. Elvira, and B. M. Sadler, "Nearly consistent finite particle estimates in streaming importance sampling," *IEEE Transactions on Signal Processing*, pp. 1–1, 2021.

[17] S. Chatterjee and P. Diaconis, "The sample size required in importance sampling," *Ann. Appl. Probab.*, vol. 28, no. 2, pp. 1099–1135, 04 2018.

[18] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[19] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statist. Sci.*, vol. 30, no. 3, pp. 328–351, 08 2015.

[20] P. Del Moral, A. Doucet, and S. S. Singh, "Uniform stability of a particle approximation of the optimal filter derivative," *SIAM Journal on Control and Optimization*, vol. 53, no. 3, pp. 1278–1304, 2015.

[21] E. Taghavi, F. Lindsten, L. Svensson, and T. B. Schön, "Adaptive stopping for fast particle smoothing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6293–6297.

[22] T. Clapp and S. Godsill, "Fixed-lag smoothing using sequential importance sampling," 1999.

[23] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[24] M. Klaas, N. d. Freitas, and A. Doucet, "Toward practical n^2 Monte Carlo: The marginal particle filter," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, ser. UAI'05. Arlington, Virginia, USA: AUAI Press, 2005, p. 308–315.

[25] L. Martino and V. Elvira, "Compressed monte carlo with application in particle filtering," *Information Sciences*, vol. 553, pp. 331–352, 2021.

[26] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proceedings of ACM SIGSPATIAL GIS 2009*, New York, NY, USA, 2009, p. 336–343.

[27] M. Roth, F. Gustafsson, and U. Orguner, "On-road trajectory generation from gps data: A particle filtering/smoothing application," in *2012 15th International Conference on Information Fusion*, 2012, pp. 779–786.

[28] P. Davidson, J. Collin, and J. Takala, "Application of particle filters to a map-matching algorithm," *Gyroscopy and Navigation*, vol. 2, no. 4, p. 285, Oct 2011.

[29] K. Kempinska, T. Davies, and J. Shawe-Taylor, "Probabilistic map-matching using particle filters," *arXiv e-prints*, p. arXiv:1611.09706, Nov. 2016.

[30] L. Moreira-Matias, J. Gama, M. Ferreira, J. Moreira, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 1393–1402, 09 2013.

[31] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P . Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[32] J. Alenlöv and J. Olsson, "Particle-based adaptive-lag online marginal smoothing in general state-space models," *IEEE Transactions on Signal Processing*, vol. 67, no. 21, pp. 5571–5582, 2019.

[33] W. R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[34] F. Lindsten, P. Bunch, S. S. Singh, and T. B. Schön, "Particle ancestor sampling for near-degenerate or intractable state transition models," 2015.

Algorithm 6 Backward Simulation

```
Input marginal filtering samples $\{\tilde{x}_t^{(k)}, \tilde{w}_t^{(k)}\}_{k=1}^N$ for $t = 0, \dots, T$, bound $\rho \geq p(x_t|x_{t-1})$ and the maximum number of rejections to attempt $R$.
Resample $\{\tilde{x}_T^{(k)}, \tilde{w}_T^{(k)}\}_{k=1}^N \to \{x_T^{(i)}\}_{i=1}^N$
for $t = T-1, \dots, 0$ do
  for $i = 1, \dots, N$ do
    for $r = 1, \dots, R$ do
      Sample $c^* \sim$ Categorical $\left(\{\tilde{w}_t^{(k)}\}_{k=1}^N\right)$
      Sample $u \sim U(0,1)$
      if $u < p(x_{t+1}^{(i)}|\tilde{x}_t^{(c^*)})/\rho$ then
        Accept $c^*$
        break
      end if
    end for
    if a sample $c^*$ was accepted then
      Set $x_t^{(i)} = \tilde{x}_t^{(c^*)}$
    else
      Calculate interacting weights and normalise in $k$
        $w_t^{(k \leftarrow i)} \propto p(x_{t+1}^{(i)}|\tilde{x}_t^{(k)})\tilde{w}_t^{(k)} \qquad k = 1, \dots, N.$
      Sample $c_i \sim$ Categorical $\left(\{w_t^{(k \leftarrow i)}\}_{k=1}^N\right)$
      Set $x_t^{(i)} = \tilde{x}_t^{(c_i)}$
    end if
  end for
end for
Output unweighted sample $\{x_{0:T}^{(i)}\}_{i=1}^N$ approximating $p(x_{0:T}|y_{0:T})$.
```