Online Variational Sequential Monte Carlo

Alessandro Mastrototaro [1] **Jimmy Olsson** [1]

Abstract

Being the most classical generative model for serial data, state-space models (SSM) are fundamental in AI and statistical machine learning. In SSM, any form of parameter learning or latent state inference typically involves the computation of complex latent-state posteriors. In this work, we build upon the variational sequential Monte Carlo (VSMC) method, which provides computationally efficient and accurate model parameter estimation and Bayesian latent-state inference by combining particle methods and variational inference. While standard VSMC operates in the offline mode, by re-processing repeatedly a given batch of data, we distribute the approximation of the gradient of the VSMC surrogate ELBO in time using stochastic approximation, allowing for online learning in the presence of streams of data. This results in an algorithm, online VSMC, that is capable of performing efficiently, entirely on-the-fly, both parameter estimation and particle proposal adaptation. In addition, we provide rigorous theoretical results describing the algorithm's convergence properties as the number of data tends to infinity as well as numerical illustrations of its excellent convergence properties and usefulness also in batch-processing settings.

1. Introduction

Being the most classical structured probabilistic generative model for serial data, *state-space models* (SSM), also known as general state-space *hidden Markov models*, are fundamental and ubiquitous in AI and statistical machine learning (Cappé et al., 2005; Bishop, 2016). In SSM, any form of parameter learning or state inference typically involves the computation of complex joint posterior distributions of latent-state variables—the so-called *joint-smoothing distributions*—given records of observations, which is a delicate problem whose analytical solution is intractable outside the limited cases of linear Gaussian models or models with finite state space. In a recent line of research (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018), this problem is addressed by combining *variational inference* (Blei et al., 2017; Kingma & Welling, 2014) and *sequential Monte Carlo* (SMC) *methods* (Gordon et al., 1993; Doucet et al., 2001; Chopin & Papaspiliopoulos, 2020) in order to design flexible families of variational joint-state distributions in the form of particle-trajectory laws. By optimizing the Kullback–Leibler divergence from (KLD) the law of the particle trajectories to the joint-smoothing distribution, this *variational SMC* (VSMC) approach is killing two birds with one stone by learning not only the unknown model parameters but also, in parallel, an optimal particle proposal kernel, the latter being a problem that has received a lot of attention in the literature (Doucet et al., 2000; Cornebise et al., 2008; Gu et al., 2015). The procedure can be viewed as a non-trivial extension of the *importance-weighted autoencoder* (IWAE) proposed by Burda et al. (2016), where standard self-normalized importance sampling has been replaced by sequential importance sampling with systematic resampling in order to obtain a tighter *evidence lower bound* (ELBO). The objective of VSMC is a similar ELBO whose gradient is approximated by the expectation, under the law of the random numbers generated by the SMC algorithm, of the gradient of the logarithm of the standard particle-based likelihood estimator, the latter being unbiased (Del Moral, 2004; Chopin & Papaspiliopoulos, 2020), is obtained as a by-product of the particle approximation of the marginal-state posterior—or *filter distribution*—flow; thus, whereas traditional particle-based inference in SSM typically relies on particle approximation of the joint-smoothing distributions (with aim of approximating, *e.g.*, the intermediate quantity of the expectation–maximization, EM, algorithm or the score function directly via the Fisher identity), which is cumbersome due to the so-called particle-path degeneracy phenomenon (Kitagawa, 1996; Cappé et al., 2005, Section 8.3), VSMC allows, using the reparameterization trick, the model parameters as well as an optimal particle proposal kernel to be simultaneously learned by processing repeatedly the given data batch using a standard particle filter.

[1]Department of Mathematics, KTH Royal Institute of Technology , Stockholm, Sweden. Correspondence to: Alessandro Mastrototaro <alemas@kth.se>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

In its basic form, VSMC is an offline inference technique, in the sense that it requires the full data batch to be processed between every update of the model and variational parameters. Still, in a wide range of AI and machine-learning contexts, observed data become available sequentially through a data stream, requiring learning to be performed in an online fashion. The increasing interest in online machine-learning technology also stems from the need of processing data batches of ever-increasing sizes. Thus, in this work we propose an online, stochastic approximation-based version of VSMC, *online variational SMC* (OVSMC), which can be used for simultaneous online model learning and proposal adaptation. On the contrary to traditional approaches to particle-based online parameter learning in SSM such as particle-based recursive maximum-likelihood (RML, Le Gland & Mevel, 1997; Del Moral et al., 2015) or online EM (Mongillo & Denève, 2008; Cappé, 2011), which rely on particle-based online approximation of the tangent filter (filter derivative) and the EM-intermediate quantity, respectively, our approach does not involve any particle smoothing, which, in order to avoid the problem of collapsing particle ancestral genealogies, typically calls for backward-sampling techniques which can be computationally very costly. In a recent work, Campbell et al. (2021) used variational approximations of the backward kernels for online state and parameter learning. Although this, interestingly, provides a non-particle-based methodology, it is very computationally intensive (see Section 5). In addition, our method allows for effective online adaptation of the particle proposal kernel in a way that differs from typical approaches in the literature; indeed, whereas traditional approaches aim to optimize the proposal time step by time step on the basis of local criteria (see, *e.g.*, Cornebise et al., 2008; 2014; Zhao et al., 2022), our approach is based on a global KLD-based criterion (described in detail in Section 3) which allows the particle cloud to be effectively guided toward state-space regions of high local likelihood, without running the risk of over-adapting locally the proposal to current (or temporarily neighboring) observations.

Although our proposed method has similarities with the *streaming variational Monte Carlo* methodology proposed by Zhao et al. (2022), it is essentially different from the same. An important difference is that the aforementioned work focuses on local optimization of model and variational parameters, such that these are assumed to vary with time and are therefore optimized time step by time step, while OVSMC estimate global, amortized parameters using stochastic approximation. In addition, in order to show that our approach is statistically well founded, we provide rigorous theoretical results describing its convergence (Theorem 4.7). Under certain strong mixing assumptions, the time-normalized gradient guiding the learning of batch VSMC can, using Birkhoff's ergodic theorem, be shown to converge as the observation batch size increases towards infinity to a deterministic function depending on the parameter as well as the particle sample size; we show that as the number of observations tends to infinity, OVSMC is solving the same problem as this ideal, 'asymptotic' VSMC, in the sense that the mean field targeted by OVSMC coincides with the time-normalized asymptotic gradient of VSMC.

Finally, we illustrate OVSMC numerically on a number of classical SSM and more complex generative models, for which the method exhibits fast parameter learning and efficient adaptation of the particle proposal kernel. In the same numerical study, we also show that OVSMC is a strong challenger of VSMC on batch problems.

The paper is structured as follows. In Section 2, we review particle filtering in the context of SSM, and their relation with variational inference in some recent works. In Section 3, we present our methodology for online learning of proposal distributions and parameters and Section 4 and Section 5 provide our theoretical results and numerical experiments, respectively.

2. Background

An SSM is a bivariate, time-homogeneous Markov chain $(X_t, Y_t)_{t\in\mathbb{N}}$ evolving on some general measurable product space $(\mathsf{X}\times\mathsf{Y}, \mathcal{X}\otimes\mathcal{Y})$. In most applications, $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$ are Euclidean and furnished with the corresponding Borel σ-fields. More specifically, the marginal process $(X_t)_{t\in\mathbb{N}}$, referred to as the *state process*, is itself assumed to be a time-homogeneous Markov with transition density $m_\theta(x_{t+1} \mid x_t)$ and initial-state density $m_0(x_0)$ (with respect to the same dominating measure dx, typically the Lebesgue measure) on X. The state process is latent but partially observed through the *observation process* $(Y_t)_{t\in\mathbb{N}}$, whose values are assumed conditionally independent given the state process such that the conditional distribution of each Y_t depends on the corresponding X_t only, and we denote by $g_\theta(y_t \mid x_t)$ the density (with respect to some dominating measure) on Y of the latter. Using this notation, the joint density of a given vector $X_{0:t} = (X_0, \ldots, X_t)$ (this is our generic notation for vectors) of states and corresponding observations $Y_{0:t}$ is given by

$$p_\theta(x_{0:t}, y_{0:t}) = m_0(x_0)g_\theta(y_0 \mid x_0) \times \prod_{s=0}^{t-1} m_\theta(x_{s+1} \mid x_s)g_\theta(y_{s+1} \mid x_{s+1}). \quad (1)$$

The model dynamics is governed by some parameter vector θ belonging to some parameter space Θ. In using these models, the focus is generally on inferring the hidden states given data, typically by determining the joint-smoothing distributions $p_\theta(x_{0:t} \mid y_{0:t}) = p_\theta(x_{0:t}, y_{0:t})/p_\theta(y_{0:t})$ or their

marginals, such as the filter distributions $p_\theta(x_t \mid y_{0:t})$. Here the joint density $p_\theta(x_{0:t}, y_{0:t})$ is given by (1), whereas the likelihood $p_\theta(y_{0:t})$ of the observations is the marginal of (1) with respect to $y_{0:t}$. As the calculation of the likelihood requires the computation of complex integrals in high dimensions, the joint-smoothing and filter distributions are generally—except in the cases where the model is linear Gaussian or the state space X is a finite set—intractable. The calculation of the joint-smoothing distributions is of critical importance also when the parameter θ is unknown and must be estimated using maximum-likelihood or Bayesian methods (see Cappé et al., 2005; Kantas et al., 2015, and the references therein). In the framework of SSMs we may distinguish between *batch* methods, where parameter and state inference is carried through given a fixed record of observations, and *online* methods, where inference is carried through in real time as new data becomes available through a data stream. In the online mode, SMC algorithms are particularly well suited for state inference; furthermore, these methods also provide a basis for online parameter estimation (Poyiadjis et al., 2011; Del Moral et al., 2015; Olsson & Westerborn Alenlöv, 2020). We next review briefly SMC and the principles of variational inference.

2.1. Sequential Monte Carlo Methods

In the context of SSM, SMC methods approximate the smoothing distribution flow $(p_\theta(x_{0:t} \mid y_{0:t}))_{t\in\mathbb{N}}$ by forming iteratively a sequence $(\xi_{0:t}^i, \omega_t^i)_{i=1}^N$, $t \in \mathbb{N}$, of random samples of particles (the $\xi_{0:t}^i$'s) with associated weights (the ω_t^i's). For $t \in \mathbb{N}$, let $\mathsf{X}^t := \mathsf{X} \times \cdots \times \mathsf{X}$ (t times); then for any real-valued measurable function h_t on X^{t+1} that is integrable with respect to $p_\theta(x_{0:t} \mid y_{0:t})$, it holds, letting $\Omega_t := \sum_{i=1}^N \omega_t^i$,

$$\sum_{i=1}^N \frac{\omega_t^i}{\Omega_t} h_t(\xi_{0:t}^i) \rightharpoonup \int h_t(x_{0:t}) p_\theta(x_{0:t} \mid y_{0:t})\, dx_{0:t}.$$

See Del Moral (2004) for a comprehensive treatment of the theory of SMC. The SMC procedure consists of two core operations performed alternately: a *selection step*, which resamples the particles with replacement according to their weights, and a *mutation step*, which propagates randomly the selected particles to new locations. After importance sampling-based initialization of $(\xi_0^i, \omega_0^i)_{i=1}^N$, the random sample is updated according to Algorithm 1, in which mutation (Line 4) is executed using some generic proposal Markov transition kernel, possibly depending on y_{t+1}, with density $r_\lambda(x_{t+1} \mid x_t, y_{t+1})$ parameterized by some vector λ belonging to some parameter space Λ. Line 3 corresponds to the selection step, where indices $(I_{t+1}^i)_{i=1}^N$ guiding the resampling are drawn from the categorical distribution on $\{1, \ldots, N\}$ induced by the particle weights.

Determining a good proposal distribution r_λ is crucial for the performance of the particle filter (see, *e.g.*, Cornebise et al., 2008; 2014; Gu et al., 2015; Zhao et al., 2022), and the particles should be guided towards state-space regions of non-vanishing likelihood in order to avoid computational waste. For instance, if the latent process is diffuse while the observations are highly informative, letting naively, as in the so-called *bootstrap particle filter* (Gordon et al., 1993), $r_\lambda \equiv m_\theta$ may result in many particles being assigned a negligible weight and thus significant sample depletion. A more appealing, data-driven option is to use the *locally optimal proposal* satisfying $r_\lambda(x_{t+1} \mid x_t, y_{t+1}) \propto m_\theta(x_{t+1} \mid x_t) g_\theta(y_{t+1} \mid x_{t+1})$ (and minimizing, *e.g.*, the Kullback–Leibler divergence (Cornebise et al., 2008)); however, this proposal is available in a closed form only in a few cases (Doucet et al., 2000; Cappé et al., 2005, Section 7.2.2.2). In Section 3 we will present a technique that allows to learn simultaneously, in an online fashion, both unknown model parameters and an efficient proposal. The proposed method is based on variational inference, which is briefly reviewed in the next section.

Algorithm 1 Particle filter

1: **Input:** $(\xi_{0:t}^i, \omega_t^i)_{i=1}^N, y_{t+1}$.
2: **for** $i = 1, \ldots, N$ **do**
3: draw $I_{t+1}^i \sim \mathsf{cat}((\omega_t^\ell)_{\ell=1}^N)$;
4: draw $\xi_{t+1}^i \sim r_\lambda(\cdot \mid \xi_t^{I_{t+1}^i}, y_{t+1})$;
5: set $\xi_{0:t+1}^i \leftarrow (\xi_{0:t}^{I_{t+1}^i}, \xi_{t+1}^i)$;
6: set $\omega_{t+1}^i \leftarrow \dfrac{m_\theta(\xi_{t+1}^i \mid \xi_t^{I_{t+1}^i}) g_\theta(y_{t+1} \mid \xi_{t+1}^i)}{r_\lambda(\xi_{t+1}^i \mid \xi_t^{I_{t+1}^i}, y_{t+1})}$;
7: **end for**
8: **return** $(\xi_{0:t+1}^i, \omega_{t+1}^i)_{i=1}^N$.

2.2. Variational Inference

Let $T \in \mathbb{N}$ be a fixed time horizon. To approximate $p_\theta(x_{0:T} \mid y_{0:T})$ by a variational-inference procedure, a family $\{q_\lambda(x_{0:T} \mid y_{0:T}) : \lambda \in \Lambda\}$ of *variational distributions* is designed, whereby the ELBO

$$\begin{aligned}\mathcal{L}^{\mathrm{VI}}(\lambda, \theta) &:= \mathbb{E}_{q_\lambda}\left[\log\left(\frac{p_\theta(X_{0:T}, y_{0:T})}{q_\lambda(X_{0:T} \mid y_{0:T})}\right)\right] \\ &\leq \log \mathbb{E}_{q_\lambda}\left[\frac{p_\theta(X_{0:T}, y_{0:T})}{q_\lambda(X_{0:T} \mid y_{0:T})}\right] = \log p_\theta(y_{0:T})\end{aligned}$$

is maximized with respect to (λ, θ). Here the bound follows from Jensen's inequality. We note that the equality is satisfied in the ideal case $q_\lambda(x_{0:T} \mid y_{0:T}) = p_\theta(x_{0:T} \mid y_{0:T})$. The optimization is performed by a combination of Monte Carlo sampling and stochastic gradient ascent.

The *importance weighted autoencoder* (`IWAE`, Burda et al., 2016) extends this idea further by optimizing a similar but

improved ELBO, where the expectation is taken with respect to the law of N independent and q_λ-distributed random variables, denoted by $q_\lambda^{\otimes N}$:

$$\begin{aligned}\mathcal{L}^{\texttt{IWAE}}(\lambda,\theta) &:= \mathbb{E}_{q_\lambda^{\otimes N}}\left[\log\left(\frac{1}{N}\sum_{i=1}^{N}\frac{p_\theta(X_{0:T}^i, y_{0:T})}{q_\lambda(X_{0:T}^i \mid y_{0:T})}\right)\right] \\ &\leq \log \mathbb{E}_{q_\lambda^{\otimes N}}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{p_\theta(X_{0:T}^i, y_{0:T})}{q_\lambda(X_{0:T}^i \mid y_{0:T})}\right] \\ &= \log \mathbb{E}_{q_\lambda}\left[\frac{p_\theta(X_{0:T}, y_{0:T})}{q_\lambda(X_{0:T} \mid y_{0:T})}\right] = \log p_\theta(y_{0:T}). \quad (2)\end{aligned}$$

Again, the bound follows from Jensen's inequality and the fact that the average that is logarithmised is an unbiased estimator of the likelihood. The IWAE provides an improvement on standard VI as $\mathcal{L}^{\texttt{IWAE}}$ provides a tighter lower bound on the likelihood $\log p(y_{0:T})$ and gets, assuming bounded weights, arbitrarily close to the same as N tends to infinity (Burda et al., 2016, Theorem 1). Still, once T is reasonably large and $q_\lambda(x_{0:T} \mid y_{0:T})$ is not sufficiently close to $p_\theta(x_{0:T} \mid y_{0:T})$, approximating $\mathcal{L}^{\texttt{IWAE}}$ by standard Monte Carlo implies generally high variance; indeed, in the extreme case, the highly skewed distribution of the terms of the likelihood estimator will effectively reduce IWAE to standard VI. This degeneracy problem can be counteracted by replacing standard Monte Carlo approximation with resampling-based SMC. For this reason, Le et al. (2018), Maddison et al. (2017) and Naesseth et al. (2018) all define an ELBO similar to (2), but where the expectation is taken with respect to the law $Q_{\lambda,\theta}^N$ of the random variables $(\xi_0^i)_{i=1}^N$ and $(\xi_t^i, I_t^i)_{i=1}^N$, $t \in \{1,\dots,T\}$, generated by the particle filter:

$$\mathcal{L}^{\texttt{SMC}}(\lambda,\theta) := \mathbb{E}_{Q_{\lambda,\theta}^N}\left[\log\left(\prod_{t=0}^{T}\frac{1}{N}\Omega_t\right)\right],$$

which is again a lower bound on $\log p_\theta(x_{0:T})$ as the argument of the logarithm is an unbiased estimator of the likelihood (see, *e.g.*, Chopin & Papaspiliopoulos, 2020, Proposition 16.3).

Like in standard VI, the maximization of $\mathcal{L}^{\texttt{SMC}}$ is carried out by alternately (1) processing the given data batch $y_{0:t}$ with the particle filter and (2) taking a stochastic gradient ascent step. The latter involves the differentiation of $\mathcal{L}^{\texttt{SMC}}$ with respect to (λ,θ), which can be carried through using the reparameterization trick (Kingma & Welling, 2014). More specifically, it is assumed that $r_\lambda(\cdot \mid x, y)$ is reparameterizable in the sense that there exists some auxiliary random variable ε, taking on values in some measurable space $(\mathsf{E}, \mathcal{E})$ and having distribution ν on $(\mathsf{E}, \mathcal{E})$ (the latter not depending on λ), and some function f_λ on $\mathsf{X} \times \mathsf{Y} \times \mathsf{E}$, parameterized by λ, such that for every $(x, y) \in \mathsf{X} \times \mathsf{Y}$, the pushforward distribution $\nu \circ f_\lambda^{-1}(x, y, \cdot)$ coincides with that governed by $r_\lambda(\cdot \mid x, y)$. In addition, importantly, for any given argument (x, y, ε), $f_\lambda(x, y, \varepsilon)$ is assumed to be differentiable with respect to λ. A similar reparameterization assumption is made for some initial proposal distribution denoted by $r_{0,\lambda}$. The previous assumptions allow us to reparameterize Algorithm 1 by splitting the procedure on Line 4 into two suboperations: first sampling $\varepsilon_{t+1}^i \sim \nu$ and then letting $\xi_{t+1}^i \leftarrow f_\lambda(\xi_t^{I_{t+1}^i}, y_{t+1}, \varepsilon_{t+1}^i)$. After this, the importance weights can be calculated as explicit functions of (λ,θ) according to

$$\begin{aligned}\omega_{t+1}^i(\lambda,\theta) &:= m_\theta(f_\lambda(\xi_t^{I_{t+1}^i}, y_{t+1}, \varepsilon_{t+1}^i) \mid \xi_t^{I_{t+1}^i}) \\ &\quad \times \frac{g_\theta(y_{t+1} \mid f_\lambda(\xi_t^{I_{t+1}^i}, y_{t+1}, \varepsilon_{t+1}^i))}{r_\lambda(f_\lambda(\xi_t^{I_{t+1}^i}, y_{t+1}, \varepsilon_{t+1}^i) \mid \xi_t^{I_{t+1}^i}, y_{n+1})} \quad (3)\end{aligned}$$

and, at initialization, $\xi_0^i \leftarrow f_\lambda(y_0, \varepsilon_0^i)$ and $\omega_0^i(\lambda,\theta) := m_0(f_\lambda(y_0, \varepsilon_0^i))g_\theta(y_0 \mid f_\lambda(y_0, \varepsilon_0^i))/r_{0,\lambda}(f_\lambda(y_0, \varepsilon_0^i) \mid y_0)$.

Le et al. (2018), Maddison et al. (2017) and Naesseth et al. (2018) have shown that the Monte Carlo approximation of $\nabla_{\lambda,\theta}\mathcal{L}^{\texttt{SMC}}(\lambda,\theta)$ can, in order to avoid unmanageable variance, be advantageously carried through by targeting the "surrogate" gradient

$$\mathcal{G}_T(\lambda,\theta) := \mathbb{E}_{Q_{\lambda,\theta}^N}\left[\nabla_{(\lambda,\theta)}\log\left(\prod_{t=0}^{T}\frac{1}{N}\Omega_t(\lambda,\theta)\right)\right], \quad (4)$$

where $Q_{\lambda,\theta}^N$ now corresponds to the law of the random variables $(\varepsilon_0^i)_{i=1}^N$ and $(\varepsilon_t^i, I_t^i)_{i=1}^N$, $t \in \{1,\dots,T\}$, generated by the particle filter using the reparameterization trick. The approximate gradient $\mathcal{G}_T(\lambda,\theta)$ is indeed different from $\nabla_{(\lambda,\theta)}\mathcal{L}^{\texttt{SMC}}(\lambda,\theta)$ in that the latter contains one additional term corresponding to the expectation of the product of $\nabla_{(\lambda,\theta)}\log Q_{\lambda,\theta}^N$ and the logarithm of the unbiased likelihood estimator (see Le et al., 2018, Appendix A, for details). Nonetheless, since, as demonstrated by Le et al. (2018), Maddison et al. (2017) and Naesseth et al. (2018), this term generally does not make a significant contribution to the gradient and is also very difficult to estimate with reasonable accuracy, we proceed as in the mentioned works and simply discard the same. Roeder et al. (2017), Tucker et al. (2018) and Finke & Thiery (2019) discussed biased gradient approximation in the framework of the IWAE, but we are not aware of any in-depth analyses in the context of VSMC.

As mentioned earlier, the variational SMC (VSMC) approach described above is an offline method in the sense that each parameter update requires $\mathcal{G}_T(\lambda,\theta)$ to be estimated by processing the entire data batch $y_{0:T}$ with the particle filter. However, in many applications it is of utmost importance to be able to update both parameter estimates and proposal distributions in real time, and in the next section we will therefore provide an online extension of VSMC to the setting where the data become available sequentially.

3. Online Implementation of VSMC

In this section our primary goal is to learn, for a given stream $(y_t)_{t\geq 0}$ of observations, the amortized proposal and model parameters as the particles evolve and new observations become available. By rewriting (4) as

$$\mathcal{G}_T(\lambda,\theta) = \mathbb{E}_{Q^N_{\lambda,\theta}}\left[\sum_{t=0}^{T} \nabla_{(\lambda,\theta)} \log \Omega_t(\lambda,\theta)\right]$$

we notice that the computation of the gradient is distributed over time, making it possible to adapt the method to the online setting. More precisely, in our scheme, the given parameters (θ_t, λ_t) are updated as

$$\lambda_{t+1} \leftarrow \lambda_t + \gamma^{\lambda}_{t+1} \nabla_\lambda \log \Omega_{t+1}(\lambda_t, \theta_t), \tag{5}$$

$$\theta_{t+1} \leftarrow \theta_t + \gamma^{\theta}_{t+1} \nabla_\theta \log \Omega_{t+1}(\lambda_{t+1}, \theta_t), \tag{6}$$

where $(\gamma^{\lambda}_t)_{t\in\mathbb{N}_{>0}}$ and $(\gamma^{\theta}_t)_{t\in\mathbb{N}_{>0}}$ are given step sizes (learning rates).

A pseudocode for our algorithm, which we refer to as *online variational SMC* (OVSMC), is displayed in Algorithm 2, from which it can be seen that the updates of λ and θ on Line 7 and Line 14 are based on two distinct sampling steps with different sample sizes L and N, respectively. Indeed, as pointed out by Le et al. (2018) and Zhao et al. (2022), the quantity $\log \Omega_{t+1}(\lambda, \theta)$ is a biased but consistent estimator of the log-predictive likelihood $\log p_\theta(y_{t+1} \mid y_{0:t})$ and will therefore, regardless of the proposal used, be arbitrarily close to this quantity as the number of particles increases. In contrast to the estimation of $\nabla_\theta \log \Omega_t(\lambda, \theta)$, this is generally problematic in the estimation of $\nabla_\lambda \log \Omega_t(\lambda, \theta)$. In fact, a large number of particles reduces the signal-to-noise ratio of the estimator of the latter gradient, up to a point where it reduces to pure noise. For this reason, we take, in the spirit of the *alternating ELBOs* strategy of Le et al. (2018, Section 4.1, in which the authors consider IWAE and VSMC ELBOs with alternating sample sizes) an approach where λ and θ are updated through two distinct optimization steps, the one for λ with a small number L of particles, typically less than ten, and the one for θ with a possibly large sample size N.

Appealingly, as clear from Algorithm 2, the method is based only on particle approximation of the filter distribution flow and therefore does not require saving the trajectories of the particles. This results in an online algorithm with memory requirements that remain uniformly limited in time and are, just like the computational complexity of the algorithm, linear in the number N of particles. In Section 4 we provide a rigorous theoretical justification of (a slightly modified version of) Algorithm 2. In particular, we show that (5)–(6) form a classical Robbins–Monro scheme (Robbins & Monro, 1951) with state-dependent Markov noise targeting a mean field corresponding to an 'asymptotic' (as the number t of data tends to infinity) VSMC.

Algorithm 2 Online Variational SMC (OVSMC)

1: **Input:** $(\xi^i_t, \omega^i_t)^N_{i=1}, y_{t+1}, \theta_t, \lambda_t$.
2: **for** $i \leftarrow 1, \dots, L$ **do**
3: draw $I^i_{t+1} \sim \mathsf{cat}((\omega^\ell_t)^N_{\ell=1})$;
4: draw $\varepsilon^i_{t+1} \sim \nu$;
5: compute $\omega^i_{t+1}(\lambda_t, \theta_t)$ according to (3);
6: **end for**
7: set $\lambda_{t+1} \leftarrow \lambda_t + \gamma^\lambda_{t+1} \nabla_\lambda \log \Omega_{t+1}(\lambda_t, \theta_t)$;
8: **for** $i \leftarrow 1, \dots, N$ **do**
9: draw $I^i_{t+1} \sim \mathsf{cat}((\omega^\ell_t)^N_{\ell=1})$;
10: draw $\varepsilon^i_{t+1} \sim \nu$;
11: set $\xi^i_{t+1} \leftarrow f_{\lambda_{t+1}}(\xi^{I^i_{t+1}}_t, y_{t+1}, \varepsilon^i_{t+1})$;
12: compute $\omega^i_{t+1}(\lambda_{t+1}, \theta_t)$ according to (3);
13: **end for**
14: set $\theta_{t+1} \leftarrow \theta_t + \gamma^\theta_{t+1} \nabla_\theta \log \Omega_{t+1}(\lambda_{t+1}, \theta_t)$;
15: **return** $(\xi^i_{t+1}, \omega^i_{t+1})^N_{i=1}, \theta_{t+1}, \lambda_{t+1}$.

4. Theoretical Results

In this section we study the limiting behavior of OVSMC and discuss its connection with batch VSMC. As explained in Section 3, the purpose of the double optimization steps in Algorithm 2 is to improve the performance of the algorithm in practical use; however, here, for simplicity, we move the parameters λ into θ, resulting in the latter also containing algorithmic parameters. Therefore, the algorithm that we theoretically analyze corresponds to Lines 8–14, with reparameterization function f_{θ_t} and ω^i_{t+1} depending only on θ_t. Furthermore, for technical reasons related to the ergodicity of an extended Markov chain that we will define below and the Lipschitz continuity (in θ) of its transition kernel, the analyzed algorithm repeats twice the generation of the auxiliary variable on Line 10, where one variable is used to estimate the gradient and the other to particle propagation. All details are provided in Appendix A, where the modified procedure is displayed in Algorithm 3 and we also provide all proofs.

Let, for $t \in \mathbb{N}$, $Z_t := (X_t, Y_t, (\xi^{I^i_t}_{t-1})^N_{i=1}, (\varepsilon^i_t)^N_{i=1})$, where $(\xi^{I^i_t}_{t-1})^N_{i=1}$ and $(\varepsilon^i_t)^N_{i=1}$ are generated according to the modified version of Algorithm 2. Then $(Z_t)_{t\in\mathbb{N}}$ is a state dependent Markov chain with transition kernel T_θ (described in the supplement), in the sense that given $Z_{0:t}$, Z_{t+1} is distributed according to $T_{\theta_t}(Z_t, \cdot)$ (note that θ_t is deterministic function of $Z_{0:t}$). Let $\mathbb{P}$ denote the law of $(Z_t)_{t\in\mathbb{N}}$ when initialized as described previously.

Assumption 4.1. The data $(y_t)_{t\in\mathbb{N}}$ is the output of an SSM $(X_t, Y_t)_{t\in\mathbb{N}}$ on $(\mathsf{X}\times\mathsf{Y}, \mathcal{X}\otimes\mathcal{Y})$ with state and observation transition densities $\bar{m}(x_{t+1} \mid x_t)$ and $\bar{g}(y_t \mid x_t)$, respectively.

Assumption 4.2. The transition densities $\bar{m}$, m_θ, g_θ and r_θ are uniformly bounded from above and below (in all their

arguments as well as in θ).

The strong mixing assumptions of Assumption 4.2 are standard in the literature and point to applications where the state space X is a compact set.

Proposition 4.3. *Let Assumptions 4.1–4.2 hold. Then for every $\theta \in \Theta$, the canonical Markov chain $(Z_t^\theta)_{t\in\mathbb{N}}$ induced by T_θ is uniformly ergodic and admits a stationary distribution τ_θ.*

Now, letting $H_\theta(Z_t) := \nabla_\theta \log \Omega_t(\theta)$, we may define the mean field $h(\theta) := \int H_\theta(z)\,\tau_\theta(dz)$ (depending implicitly on the sample size N). Note that by the law of large numbers for ergodic Markov chains it holds, a.s., that $\lim_{t\to\infty} \sum_{s=0}^t H_\theta(Z_s^\theta)/t = h(\theta)$. Thus, since VSMC estimates the gradient $\mathcal{G}_t(\theta)$ by $\sum_{s=0}^t H_\theta(Z_s^\theta)$, finding a zero of the mean field $h(\theta)$ can be considered equivalent to applying VSMC to an infinitely large batch of observations from the model. From this perspective, it is interesting to note that our proposed OVSMC algorithm is a Robbins–Monro scheme targeting $h(\theta)$, with updates given by $\theta_{t+1} \leftarrow \theta_t + \gamma_{t+1}^\theta H_{\theta_t}(Z_{t+1})$ and $Z_{t+1} \sim T_{\theta_t}(Z_t, \cdot)$, $t \in \mathbb{N}$. Moreover, as established by our main result, Theorem 4.7, the scheme produces a parameter sequence $(\theta_t)_{t\in\mathbb{N}}$ making the gradient arbitrarily close to zero (in the L_2 sense). We preface this result by some further assumptions.

Assumption 4.4. The gradients $\nabla_\theta m_\theta$, $\nabla_\theta g_\theta$ and $\nabla_\theta r_\theta$, and their compositions with the reparameterization function f_θ, are all uniformly bounded and Lipschitz.

Assumption 4.5. There exists a bounded function V on Θ (the *Lyapunov* function) such that $h = \nabla_\theta V$.

Assumption 4.6. For every $t \in \mathbb{N}_{>0}$, $\gamma_{t+1}^\theta \le \gamma_t^\theta$. In addition, there exist constants $a > 0$, $a' > 0$ and $c \in (0, 1)$ such that for every t, $\gamma_t^\theta \le a\gamma_{t+1}^\theta$, $\gamma_t^\theta - \gamma_{t+1}^\theta \le a'(\gamma_{t+1}^\theta)^2$ and $\gamma_1 \le c$.

Theorem 4.7. *Let Assumptions 4.1, 4.2, 4.4, 4.5 and 4.6 hold. Then there exist constants $b > 0$ and $b' > 0$, possibly depending on V, such that for every $t \in \mathbb{N}$,*

$$\mathbb{E}[\|h(\theta_\tau)\|^2] \le \frac{b + b' \sum_{s=0}^t (\gamma_{s+1}^\theta)^2}{\sum_{s=0}^t \gamma_{s+1}^\theta},$$

where $\tau \sim \mathsf{cat}((\gamma_{s+1}^\theta)_{s=0}^t)$ is a random variable on $\{0, \dots, t\}$.

Corollary 4.8. *Let the assumptions of Theorem 4.7 hold and let, for every $t \in \mathbb{N}_{>0}$, $\gamma_t^\theta = c/\sqrt{t}$, where $c > 0$ is given in Assumption 4.6. Then $\mathbb{E}[\|h(\theta_\tau)\|^2] = \mathcal{O}(\log t/\sqrt{t})$.*

5. Experiments

In this section we illustrate numerically the performance of OVSMC. We illustrate the method's capability of learning online good amortized proposals and model parameters and more complex generative models, but also show that it can serve as a strong competitor to VSMC in batch scenarios. Stochastic gradients are passed to the ADAM optimizer (Kingma & Ba, 2015) in Tensorflow 2[1]. All the experiments are run on an Apple MacBook Pro M1 2020, memory 8GB.

5.1. Linear Gaussian SSM

Our first example is a standard linear Gaussian SSM with $\mathsf{X} = \mathbb{R}^{d_x}$ and $\mathsf{Y} = \mathbb{R}^{d_y}$ for some $(d_x, d_y) \in \mathbb{N}_{>0}^2$. More precisely, we let $m_\theta(\cdot \mid x)$ and $g_\theta(\cdot \mid x)$ be $N_{d_x}(Ax, S_u^\intercal S_u)$ and $N_{d_y}(Bx, S_v^\intercal S_v)$, respectively, where A and S_u are $d_x \times d_x$ matrices, $B \in \mathbb{R}^{d_y \times d_x}$ and $S_v \in \mathbb{R}^{d_y \times d_y}$. We start with the simplest case $d_x = d_y = 1$ and generate data under $A = 0.8$, $B = 1$ and $S_u = 0.5$. We consider two cases for $S_v \in \{0.2, 1.2\}$ corresponding to informative and more non-informative observations, respectively. The state process is initialized with its stationary distribution $N(0, S_u^2/(1-A^2))$. Our aim is to estimate A and S_u and, in parallel, optimizing the particle proposal. For this purpose, we let $r_\lambda(\cdot \mid x_t, y_{t+1})$ be $N(\mu_\lambda(x_t, y_{t+1}), \sigma_\lambda^2(x_t, y_{t+1}))$, where μ_λ and σ_λ^2 are two distinct neural networks with one dense hidden layer having three and two nodes, respectively, and relu activation functions. In Figure 1 we observe the convergence of the unknown parameters for a set of distinct starting values, noticing that more informative observations yields, as expected, faster convergence. Even though values close to the true parameters are obtained already after about 10000 to 20000 iterations, the algorithm is kept running until $t = 50000$ for the purpose of learning the proposal. We observe an improvement of the effective sample size (ESS) of the particle cloud as r_λ is converging towards the optimal proposal, as shown in Figure 8 in Appendix B. Figure 2 displays conditional proposal densities for some given (x_t, y_{t+1}) in a single run; we see that for both choices of S_v, the learned proposal is generally close to the locally optimal one, except in the presence of unlikely outliers in the data, in which case the learned proposal tend to move, as is desirable, its mode towards that of the prior (bootstrap) kernel.

Next, we show that our method can also be usefully applied in the batch-data mode. Following Naesseth et al. (2018) and Zhao et al. (2022), let $d_x = d_y = 10$ and let the matrix A have elements $A_{ij} = 0.42^{|i-j|+1}$. Furthermore, let $S_u = I$ and $S_v = 0.5I$. We consider two parameterizations of B, one sparse, $B = I$, and one dense where its elements are random, independent and standard normally distributed. With these parameterizations, a data record $y_{0:T}$, $T = 100$, was generated starting from an $N(0, I)$-distributed initial state. In this part, we view the model parameters θ as being known and focus on learning of the proposal. Instead

[1]The Python code may be found at https://bitbucket.org/amastrot/ovsmc.

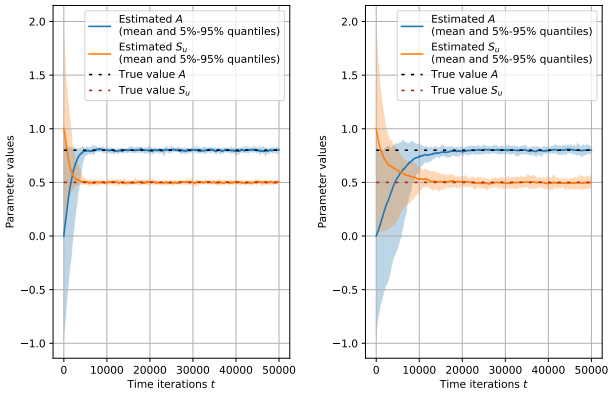

Figure 1: Parameter learning curves for the one-dimensional linear Gaussian SSM in Section 5.1, obtained using algorithm 2 with $L = 5$ and $N = 10000$ for $S_v = 0.2$ (left) and $S_v = 1.2$ (right). The means and the quantiles are calculated on the basis of 100 learning curves, each starting with a different initial value and based on independently generated observation data.

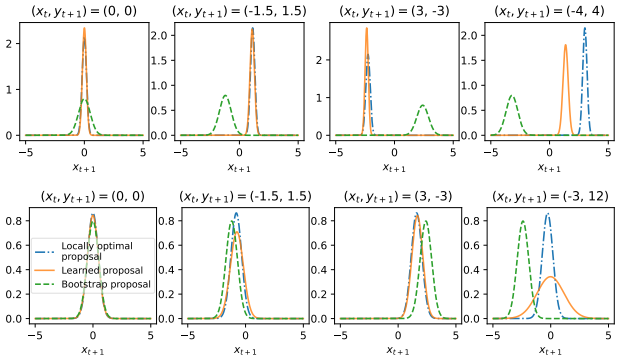

Figure 2: Comparisons of the bootstrap, locally optimal and learned proposals for different (x_t, y_{t+1}). Here OVSMC was run for 50000 iterations with $L = 5$ and $N = 10000$ for $S_v = 0.2$ (top) and $S_v = 1.2$ (bottom).

of learning, as done by Naesseth et al. (2018) and Zhao et al. (2022), different local proposals for each time step, we learn, in both batch and online mode, an amortized Gaussian proposal where the mean vector and diagonal covariance matrix are functions of x_t and y_{t+1}, modeled by neural networks. These neural networks, which are time invariant, have each a single hidden layer with 16 nodes each and the relu activation function. In this setting, we (1) processed the given observation batch M times using standard VSMC, with gradient-based parameter updates between every sweep of the data, and (2) compared the result with the output of OVSMC when executed on a data sequence of length $(T+1)M$ formed by repeating the given data record M times. The number of particles was equal to $L = 5$ for both methods. Figure 3 displays the resulting ELBO evolutions for the two methods and some different ADAM learning rates, and it is clear from this plot that OVSMC, with its appealing convergence properties and reduced noise, is indeed a challenger of VSMC in this batch context.

Comparison to Campbell et al. (2021)

As anticipated in Section 1, the *online variational filtering* (OVF) approach of Campbell et al. (2021) performs online parameter learning, although considering a variational family that is not particle-based and thus not directly comparable to the VSMC one. Even if it is algorithmically and computationally complex and includes a large number of hyperparameters to be tuned, OVF is a relevant recent work in the context of online variational learning. Thus, we used it as a benchmark for OVSMC, and let the latter solve the same problem as in Figure 1b in Campbell et al. (2021, Section 5.1). The result is displayed in Figure 4, where we demonstrate the comparability between the two methods for learning A and B; however, we note a significant difference in the computational time, since using the code provided by Campbell et al. (2021) OVF takes about 17 hours while OVSMC uses less than one hour for a single run. Even though we are aware that part of this difference may be due to the particular implementation used, the complexity gap between the two methods is definitely non-negligible.

5.2. Stochastic Volatility and RML

In this example we focus on parameter estimation and compare OVSMC with PaRIS-based RML (Olsson & Westerborn Alenlöv, 2020). We consider a univariate *stochastic volatility model* (Hull & White, 1987), where for $x \in \mathsf{X} = \mathbb{R}$, $m_\theta(\cdot \mid x)$ and $g_\theta(\cdot \mid x)$ are $N(\alpha x, \sigma^2)$ and $N(0, \beta^2 \exp(x))$, respectively, with $\alpha \in (0,1)$, $\sigma > 0$ and $\beta > 0$. The state process is initialized according to its $N(0, \sigma^2/(1-\alpha^2))$ stationary distribution. Figure 5 reports the learning curves of the estimated parameters $\theta = (\alpha, \sigma, \beta)$ obtained with OVSMC and particle-based RML, starting from different parameter vectors, for 27 independent data sequences generated under $\theta = (0.975, 0.165, 0.641)$. In both algorithms, the learning rate was 10^{-3}. The learning curves show similar behavior, with OVSMC converging faster than RML in several cases when estimating β, but fluctuating somewhat more when learning α and σ. This is to be expected, since the RML procedure is based on particle smoothing. However, we should remember that OVSMC does not use any backward sampling and in addition to learning θ also has the ability to optimize online the proposal kernel over the same family of deep Gaussian proposals as in Section 5.1. As the number of particles grows, the computational speed is also in favor of OVSMC, since the PaRIS-based RML has a computational bottle-

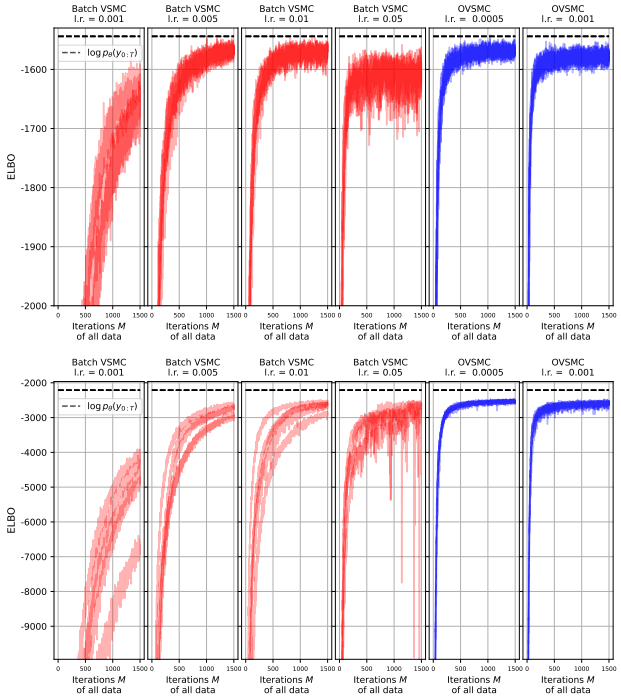

Figure 3: ELBO evolutions of VSMC and OVSMC (each running with $L = 5$) for the multivariate linear Gaussian SSM in Section 5.1 (top: B sparse; bottom: B dense). In each plot, which corrsponds to a particular learning rate, five independent runs och each algorithm are displayed on top of each other and compared with the target log-likelihood.

neck stemming from its inherent backward-sampling mechanism; with our implementation, OVSMC was three times faster than PaRIS-based RML in this specific case, and this ratio grows even more in favor of OVSMC as N increases.

5.3. Deep Generative Model of Moving Agent

In this section we study the applicability of OVSMC to more complex and high-dimensional models. Inspired by Le et al. (2018, Section 5.3), we produced a long and partially observable video sequence, with frames represented by 32×32 arrays, showing a moving agent. The motion is similar as the one described in the referenced paper, with the difference that the agent exhibits a stationary behavior by bouncing against the edges of the image. The agent is occluded from the image whenever it moves into the central region of the frame, covered by a 16×30 horizontal rectangle. The data generation is described in more detail in Appendix C.1. Following Le et al. (2018, Section C.1), we model the generative process and the proposal through the framework of *variational recurrent neural networks* (Chung et al., 2015), with a similar architecture described in more detail in Appendix C.2.

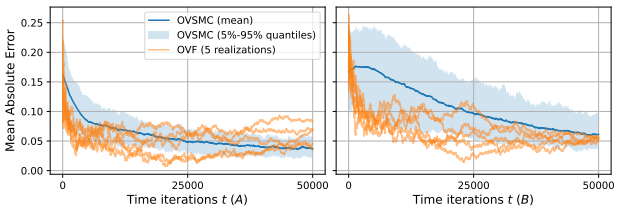

Figure 4: Mean absolute errors estimating A (left) and B (right) of five independent runs of OVF, along with the distribution of 40 independent runs of OVSMC, with $L = 5$ and $N = 10^4$, proposal kernel as described in Section 5.1, with 64 nodes in the hidden layer of each neural network, and ADAM learning rates 10^{-3}. Here $d_x = d_y = 10$, $S_u = 0.1I$, and $S_v = 0.25I$ and matrices A and B are diagonal with i.i.d. $\mathrm{Unif}(0.5, 1)$-distributed elements.

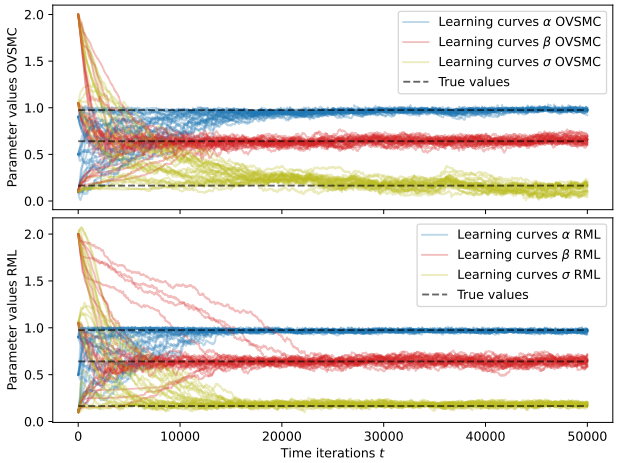

Figure 5: Parameter learning curves obtained with OVSMC (with $L = 5$ and $N = 1000$) and PaRIS-based RML (Olsson & Westerborn Alenlöv, 2020) (with $N = 1000$), for the stochastic volatility SSM in Section 5.2, with learning rate 10^{-3}.

In this setting, we generated a single video sequence with $T = 10^5$ frames, and processed the same by iterating Algorithm 2, with $N = 20$ and $L = 5$, T times. We remark that since we have to run the algorithm for a large number of iterations, it is clearly infeasible—and possibly not even useful—to input the whole history of frames and latent states to the recurrent neural network of the model; instead, we include only a fixed window (of length 40) of recent frames and states and discard all the previous information. This allows the average time and memory requirement per iteration to be kept at a constant value.

Figures 6–7 show a visual illustration of the method's ability to generate probable videos. The goal is to analyze the quality of the learned generative model by visualizing some

independently sampled videos and comparing them to a reference video generated from a different seed than the one used for training. We consider a 150 frames long reference video. For each sample we use the proposal distribution to generate the latent states for the first 60 iterations, *i.e.*, we access the true frame to sample and then decode the states, and after $t = 60$ we continue with the generative model only. The images show every third frame for the true video and five independent sample videos. We note that the first 60 frames of the sample videos look similar to the true video, as expected since the model has access to the true observations. After that, the sample videos continue similarly in an initial phase, but start to move at different speeds when the agent disappears behind the rectangle after $t = 75$. As the generated frame sequences show, the model predicts a smooth movement pattern of the agent, with some random perturbations, as in the second sample, where the agent leaves the rectangle after $t = 96$ from the top instead of below.

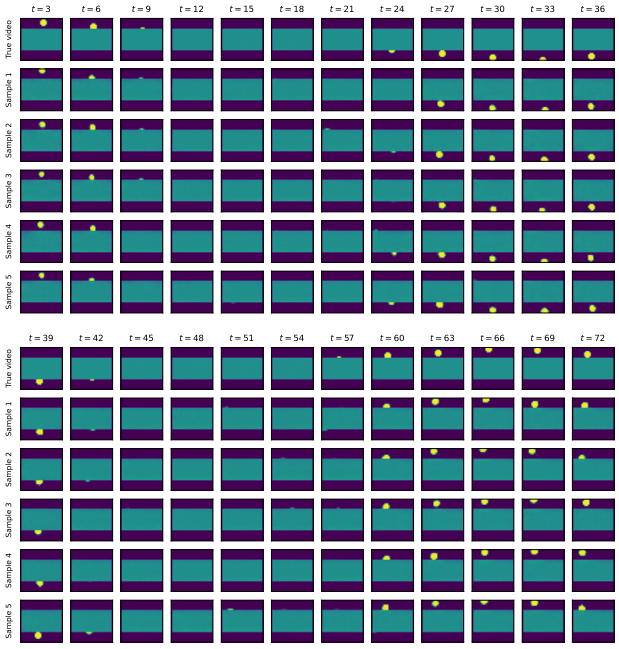

Figure 6: See caption in Figure 7.

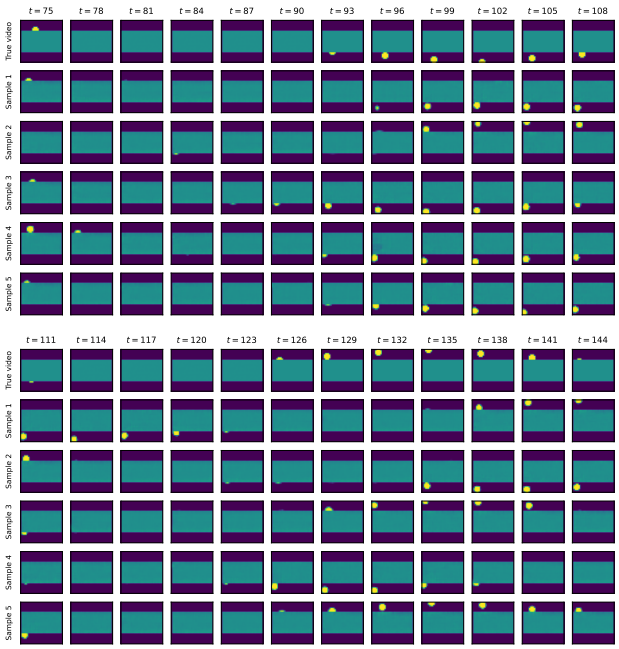

Figure 7: Comparison between original video and five samples under the learned model. Until $t = 60$, frames are generated using the learned proposal and the given data to encode the latent states; after this, the frames are produced using the generative model.

6. Conclusions

We have presented the OVSMC algorithm, a procedure that extends the batch VSMC to the context of streaming data. The proposed methodology allows us to learn, on-the-fly, unknown model parameters and optimal particle proposals in both standard SSM as well as and more complex generative models. Under strong mixing assumptions, which are standard in the literature, we provide theoretical support showing that the stochastic approximation scheme of OVSMC solves the same problem as VSMC for an infinite batch size. The interesting question of whether the estimates produced by OVSMC are still consistent for the true parameter in the case of correctly specified SSM despite the biased approximation of the ELBO gradient remains open for future work. Moreover, from an application point of view, it would be appealing to explore, empirically and theoretically, the case where the parameters vary through time (*i.e.* when dealing with non-stationary data), with the aim of identifying the conditions under which OVSMC would still work.

Acknowledgments

This work is supported by the Swedish Research Council, Grant 2018-05230.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2016.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Campbell, A., Shi, Y., Rainforth, T., and Doucet, A. Online variational filtering and parameter learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18633–18645. Curran Associates, Inc., 2021.

Cappé, O. Online EM algorithm for hidden Markov models. *J. Comput. Graph. Statist.*, 20(3):728–749, 2011.

Cappé, O., Moulines, E., and Rydén, T. *Inference in Hidden Markov Models*. Springer, New York, 2005.

Chopin, N. and Papaspiliopoulos, O. *An introduction to sequential Monte Carlo methods*. Springer, New York, 2020.

Chung, J., Kastner, K., Dinh, L., Goel, K.and Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2980 – 2988. Curran Associates, Inc., 2015.

Cornebise, J., Moulines, E., and Olsson, J. Adaptive methods for sequential importance sampling with application to state space models. *Stat. Comput.*, 18(4):461–480, 2008.

Cornebise, J., Moulines, E., and Olsson, J. Adaptive sequential Monte Carlo by means of mixture of experts. *Stat. Comput.*, 24(3):317–337, 2014.

Del Moral, P. *Feynman–Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.

Del Moral, P., Doucet, A., and Singh, S. S. Uniform stability of a particle approximation of the optimal filter derivative. *SIAM Journal on Control and Optimization*, 53(3):1278–1304, 2015.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. *Markov Chains*. Spinger, 2018.

Doucet, A., Godsill, S., and Andrieu, C. On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.*, 10:197–208, 2000.

Doucet, A., De Freitas, N., and Gordon, N. (eds.). *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

Finke, A. and Thiery, A. H. On importance-weighted autoencoders. *arXiv preprint arXiv:1907.10477*, 2019.

Gordon, N., Salmond, D., and Smith, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Radar Signal Process.*, 140:107–113, 1993.

Gu, S. S., Ghahramani, Z., and Turner, R. E. Neural adaptive sequential Monte Carlo. *Advances in neural information processing systems*, 28, 2015.

Hull, J. and White, A. The pricing of options on assets with stochastic volatilities. *J. Finance*, 42:281–300, 1987.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30 (3):328–351, 2015.

Karimi, B., Miasojedow, B., Moulines, E., and Wai, H.-T. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pp. 1944–1974. PMLR, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kitagawa, G. Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 1:1–25, 1996.

Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

Le Gland, F. and Mevel, L. Recursive estimation in HMMs. In *Proc. IEEE Conf. Decis. Control*, pp. 3468–3473, 1997.

Liu, J. S. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Stat. Comput.*, 6:113–119, 1996.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. Filtering variational objectives. *Advances in Neural Information Processing Systems*, 30, 2017.

Meyn, S. P. and Tweedie, R. L. *Markov Chains and Stochastic Stability*. Cambridge University Press, London, 2009.

Mongillo, G. and Denève, S. Online learning with hidden Markov models. *Neural Computation*, 20(7):1706–1716, 2008. doi: 10.1162/neco.2008.10-06-351.

Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. Variational sequential Monte Carlo. In *International conference on artificial intelligence and statistics*, pp. 968–977. PMLR, 2018.

Olsson, J. and Westerborn Alenlöv, J. Particle-based online estimation of tangent filters with application to parameter estimation in nonlinear state-space models. *Annals of the Institute of Statistical Mathematics*, 72:545–576, 2020.

Poyiadjis, G., Doucet, A., and Singh, S. S. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.

Robbins, H. and Monro, S. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951.

Roeder, G., Wu, Y., and Duvenaud, D. K. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tadic, V. Z. B. and Doucet, A. Asymptotic properties of recursive maximum likelihood estimation in non-linear state-space models. *arXiv preprint arXiv:1806.09571*, 2018.

Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. In *International Conference on Learning Representations*, 2018.

Zhao, Y., Nassar, J., Jordan, I., Bugallo, M., and Park, I. M. Streaming variational monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1150–1161, 2022.

A. Proof of Theorem 4.7

In the first part of this supplement we provide a detailed proof of Theorem 4.7, proceeding as follows.

- **Preliminaries (Section A.1).** For technical reasons, the proof of Theorem 4.7 to be presented calls for some more sophisticated notation, introduced in Section A.1.1. In addition, in Section A.1.2 we present the slight modification of Algorithm 2 that is the subject of our theoretical analysis and in which we optimize a unique parameter vector θ containing both model and proposal parameters.
- **Intermediate results (Section A.2).** In this part, we redefine the extended state-dependent Markov chain $(Z_t)_{t\in\mathbb{N}}$ (discussed in Section 4), including the original SSM as well as the random variables generated by Algorithm 3. Under strong mixing assumptions, we establish, in Proposition A.4 (corresponding to Proposition 4.3) uniform ergodicity of the Markov transition kernel T_θ governing $(Z_t)_{t\in\mathbb{N}}$ and the existence of a stationary distribution.

  After this, we consider the Robbins–Monroe scheme with state-dependent Markov noise targeting the mean field $h(\theta)$ defined (as in Section 4) as the expectation of the noisy measurement function H_θ under the stationary distribution of T_θ, and prove that it is bounded. Under the assumption that the state-process and emission transition densities as well as their gradients (with respect to θ) are bounded and Lipschitz continuous in θ, Proposition A.6 then establishes the existence of a solution to the Poisson equation associated with the Markov kernel T_θ.
- **Main proofs (Section A.3).** Next, we introduce further assumptions regarding the Lyapunov function associated with the mean field and the step-size sequence of the stochastic update, under which we restate and prove Theorem 4.7 and Corollary 4.8.
- **Auxiliary results (Section A.4).** Finally, we prove Proposition A.6 using Lemmas A.11 and A.12. The approach to these proofs is partially inspired by the work of Tadic & Doucet (2018).

We remark that our analysis is carried through for a fixed particle sample size $N \in \mathbb{N}_{>0}$ in the SMC algorithm, and it is beyond the scope of our paper to optimize the derived theoretical bounds with respect to N.

A.1. Preliminaries

A.1.1. Notation

We let $\mathbb{R}_+$ and $\mathbb{R}_+^*$ be the sets of nonnegative and positive real numbers, respectively, and denote vectors by $x_{m:n} := (x_m, x_{m+1}, \dots, x_{n-1}, x_n)$ or, alternatively, by $x^{m:n} := (x^m, x^{m+1}, \dots, x^{n-1}, x^n)$, depending on the situation. For some general state space $(\mathsf{S}, \mathcal{S})$, we let $\mathsf{M}(\mathcal{S})$ be the set of measures on $\mathcal{S}$, and $\mathsf{M}_1(\mathcal{S}) \subset \mathsf{M}(\mathcal{S})$ the probability measures. For a signed measure $\sigma(ds)$ on $(\mathsf{S}, \mathcal{S})$, we denote by $|\sigma|\,(ds)$ its total variation.

We now reintroduce the SSM $(X_t, Y_t)_{t\in\mathbb{N}}$, specified in Section 2, in a somewhat more rigorous way. More specifically, $(X_t, Y_t)_{t\in\mathbb{N}}$ is defined as a bivariate Markov chain evolving on $(\mathsf{X}\times\mathsf{Y}, \mathcal{X}\otimes\mathcal{Y})$ according to the Markov transition kernel

$$K_\theta : (\mathsf{X}\times\mathsf{Y})\times(\mathcal{X}\otimes\mathcal{Y}) \ni ((x,y),A) \mapsto \iint \mathbb{1}_A(x',y')\, M_\theta(x,dx')\, G_\theta(x',dy'),$$

where

$$M_\theta : \mathsf{X}\times\mathcal{X} \ni (x,A) \mapsto \int \mathbb{1}_A(x')\, m_\theta(x,x')\, \mu(dx'),$$
$$G_\theta : \mathsf{X}\times\mathcal{Y} \ni (x,B) \mapsto \int \mathbb{1}_B(y)\, g_\theta(x,y)\, \eta(dy),$$

with $m_\theta : \mathsf{X}\times\mathsf{X} \to \mathbb{R}_+$ and $g_\theta : \mathsf{X}\times\mathsf{Y} \to \mathbb{R}_+$ being the state and emission transition densities and $\mu \in \mathsf{M}(\mathcal{X})$ and $\eta \in \mathsf{M}(\mathcal{Y})$ reference measures. (Here we have slightly modified the notation in Section 2, by using the short-hand notation $m_\theta(x,x') = m_\theta(x' \mid x)$ and $g_\theta(x,y) = g_\theta(y \mid x)$.) The chain is initialized according to $\chi\otimes G_\theta : \mathcal{X}\otimes\mathcal{Y} \ni A \mapsto \int_A \chi(dx)\, G_\theta(x,dy)$, where χ is some probability measure on $(\mathsf{X},\mathcal{X})$ having density $m_0(x)$ with respect to μ. As specified in Section 2, only the process $(Y_t)_{t\in\mathbb{N}}$ is observed, whereas the state process $(X_t)_{t\in\mathbb{N}}$ is unobserved and hence referred to as *latent* or *hidden*. Moreover, θ is a parameter belonging to some vector space Θ and governing the dynamics of the model.

A.1.2. ALGORITHM

We let $R_\theta : \mathsf{X} \times \mathsf{Y} \times \mathcal{X} \to [0,1]$ be some Markov kernel, the so-called *proposal kernel*, parameterized by $\theta \in \Theta$ as well and having transition density $r_\theta : \mathsf{X} \times \mathsf{X} \times \mathsf{Y} \to \mathbb{R}_+$ with respect to μ, such that for every $(x, y, A) \in \mathsf{X} \times \mathsf{Y} \times \mathcal{X}$,

$$R_\theta((x,y),A) = 0 \Rightarrow \int \mathbb{1}_A(x') g_\theta(x',y)\, M_\theta(x,dx') = 0.$$

On the basis of the proposal kernel, define the weight function $w_\theta(x,x',y) := m_\theta(x,x') g_\theta(x',y)/r_\theta(x,x',y)$ for $(x,x',y) \in \mathsf{X} \times \mathsf{X} \times \mathsf{Y}$ such that $r_\theta(x,x',y) > 0$. In order to express the particles as explicit differentiable functions of θ, the proposal is assumed to be reparameterizable. More precisely, we assume that there exist some state-space $(\mathsf{E}, \mathcal{E})$, an easily sampleable probability measure $\nu \in \mathsf{M}_1(\mathcal{E})$ not depending on θ, and a function $f_\theta : \mathsf{X} \times \mathsf{Y} \times \mathsf{E} \to \mathsf{X}$ such that for all $(x,y) \in \mathsf{X} \times \mathsf{Y}$ and $\theta \in \Theta$, it holds that $\int h(f_\theta(x,y,v))\, \nu(dv) = \int h(x')\, R_\theta((x,y),dx')$ for all bounded real-valued measurable functions h on X; in other words, the pushforward distribution $\nu \circ f_\theta^{-1}(x,y,\cdot)$ coincides with $R_\theta((x,y),\cdot)$.

Algorithm 3 displays the procedure studied in our analysis. In this slightly modified version of Algorithm 2, the particle system will be represented by the *resampled* particles, denoted here as $(\tilde{\xi}_t^i)_{i=1}^N$, $t \geq 0$. In order to avoid introducing further notation, the resampled particles are conventionally all initialized at time -1 by some arbitrary value $u \in \mathsf{X}$ such that $\{f_\theta(u, y_0, \varepsilon_0^i)\}_{i=1}^N$, corresponding to $(\xi_0^i)_{i=1}^N$, are i.i.d. according to some initial proposal distribution $R_\theta((u,y_0),\cdot)$ depending on u, where $(\varepsilon_0^i)_{i=1}^N \sim \nu^{\otimes N}$. As we will see, the cloud of resampled particles will be included in the state-dependent Markov chain $(Z_t)_{t\in\mathbb{N}}$ governing the perturbations of the stochastic-approximation scheme under consideration; the initialization according to the constant u is described in (7).

In Algorithm 3, we operate with *two* samples of mutated particles, one used for the propagation of the particle cloud and one used for approximation of the gradient, in the sense that the noise variables $(\varepsilon_{t+1}^i)_{i=1}^N$ generated on Line 9 are not used to propagate the particles. Consequently, $f_{\theta_t}(\tilde{\xi}_t^i, y_{t+1}, \varepsilon_{t+1}^i)$ is generally different from ξ_{t+1}^i. Importantly, this decoupling makes the Markov transition kernel T_θ non-collapsed (free from Dirac components), which, as we will see, facilitates significantly the theoretical analysis of the algorithm.

Algorithm 3 OVSMC (modified version)

1: **Input:** $(\tilde{\xi}_{t-1}^i)_{i=1}^N$, $y_{t:t+1}$, θ_t
2: **for** $i = 1, \dots, N$ **do**
3: draw $\xi_t^i \sim R_{\theta_t}((\tilde{\xi}_{t-1}^i, y_t), \cdot)$;
4: set $\omega_t^i \leftarrow w_{\theta_t}(\tilde{\xi}_{t-1}^i, \xi_t^i, y_t)$;
5: **end for**
6: **for** $i = 1, \dots, N$ **do**
7: draw $I_{t+1}^i \sim \mathsf{cat}((\omega_t^\ell)_{\ell=1}^N)$;
8: set $\tilde{\xi}_t^i \leftarrow \xi_t^{I_{t+1}^i}$;
9: draw $\varepsilon_{t+1}^i \sim \nu$;
10: **end for**
11: set $\theta_{t+1} \leftarrow \theta_t + \gamma_{t+1} \nabla \log \left(\sum_{i=1}^N w_{\theta_t}(\tilde{\xi}_t^i, f_{\theta_t}(\tilde{\xi}_t^i, y_{t+1}, \varepsilon_{t+1}^i), y_{t+1}) \right)$;
12: **return** $(\tilde{\xi}_t^i)_{i=1}^N$, θ_{t+1}.

A.2. Intermediate results

A.2.1. CONSTRUCTION OF $(Z_t)_{t\in\mathbb{N}}$

We first provide a more detailed statement of Assumption 4.1, which assumes that the law of the data is governed by an unspecified SSM.

Assumption A.1. The observed data stream $(Y_t)_{t\in\mathbb{N}}$ is the output of an SSM $(X_t, Y_t)_{t\in\mathbb{N}}$ on $(\mathsf{X}\times\mathsf{Y}, \mathcal{X}\otimes\mathcal{Y})$ with some state and observation transition kernels $\bar{M}(x,dx')$ and $\bar{G}(x,dy)$, respectively. These kernels have transition densities $\bar{m}(x,x')$ and $\bar{g}(x,y)$ with respect to μ and η, respectively.

As explained in Section 4, the stochastic process $(Z_t)_{t\in\mathbb{N}}$, evolving on the product space $(\mathsf{Z}, \mathcal{Z}) := (\mathsf{X}\times\mathsf{Y}\times\mathsf{X}^N\times\mathsf{E}^N, \mathcal{X}\otimes\mathcal{Y}\otimes\mathcal{X}^{\otimes N}\otimes\mathcal{E}^{\otimes N})$, includes the data-generating SSM well as the random variables generated by Algorithm 3, *i.e.*, for $t\in\mathbb{N}$, $Z_t := (X_t, Y_t, \tilde{\xi}_{t-1}^{1:N}, \varepsilon_t^{1:N})$. Let $\mathbb{P}$ and $(\mathcal{F}_t)_{t\in\mathbb{N}}$ be the law and natural filtration of $(Z_t)_{t\in\mathbb{N}}$; then, as described in Section 4, $(Z_t)_{t\in\mathbb{N}}$ is a state-dependent Markov chain with transition kernel T_θ, in the sense that for any bounded measurable function h on Z, $\mathbb{P}$-a.s., $\mathbb{E}[h(Z_{t+1}) \mid \mathcal{F}_t] = T_{\theta_t} h(Z_t)$. The kernel T_θ is given by, with $z_t = (x_t, y_t, \tilde{x}_{t-1}^{1:N}, v_t^{1:N})$,

$$T_\theta(z_t, dz_{t+1}) := \bar{M}(x_t, dx_{t+1})\,\bar{G}(x_{t+1}, dy_{t+1}) \int_{x_t^{1:N}} \prod_{k=1}^N r_\theta(\tilde{x}_{t-1}^k, x_t^k, y_t) \times \prod_{j=1}^N \left(\sum_{i=1}^N \frac{w_\theta(\tilde{x}_{t-1}^i, x_t^i, y_t)}{\sum_{\ell=1}^N w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \delta_{x_t^i}(d\tilde{x}_t^j) \right) \nu^{\otimes N}(dv_{t+1}^{1:N})\,\mu^{\otimes N}(dx_t^{1:N}),$$

where we have written $\int_{x_t^{1:N}}$ to indicate that the integral is with respect to $(x_t^i)_{i=1}^N$. The initial state Z_0^θ is initialized according to the probability measure

$$\tau_0(dz_0) := \chi(dx_0)\,\bar{G}(x_0, dy_0)\,\delta_u^{\otimes N}(d\tilde{x}_{-1}^{1:N})\,\nu^{\otimes N}(dv_0^{1:N}), \tag{7}$$

where the dummy particles $\tilde{x}_{-1}^{1:N}$ we are initialized at an arbitrary point $u\in\mathsf{X}$. Under the following assumption, we will establish that the kernel T_θ is uniformly ergodic and has an invariant distribution.

Assumption A.2. There exists $\epsilon\in(0,1)$ such that for every $\theta\in\Theta$, $(x,x')\in\mathsf{X}^2$ and $y\in\mathsf{Y}$,

$$\epsilon \le \bar{m}(x,x') \le \frac{1}{\epsilon}, \qquad \epsilon \le m_\theta(x,x') \le \frac{1}{\epsilon}, \qquad \epsilon \le g_\theta(x,y) \le \frac{1}{\epsilon}, \qquad \epsilon \le r_\theta(x,x',y) \le \frac{1}{\epsilon}.$$

The strong mixing assumptions of Assumption A.2 (as well as Assumption A.5 introduced later) are standard in the literature and point to applications where the state and parameter space are compact sets. Note that from Assumption A.2 it follows directly that $\epsilon^3 \le w_\theta(x,x',y) \le \epsilon^{-3}$ for all $\theta\in\Theta$, $(x,x')\in\mathsf{X}^2$ and $y\in\mathsf{Y}$. Let $(Z_t^\theta)_{t\in\mathbb{N}}$ and $\mathbb{P}_\theta$ denote the canonical Markov chain and its law induced by the Markov transition kernel T_θ.

Remark A.3. Note that by Assumption A.2 it follows that the reference measure μ is a finite measure on $(\mathsf{X}, \mathcal{X})$. Thus, without loss of generality we assume in the following that μ is a probability measure.

We now rewrite and prove Proposition 4.3.

Proposition A.4 (Proposition 4.3)**.** *Let Assumptions A.1 and A.2 hold. Then for every $\theta\in\Theta$, $(Z_t^\theta)_{t\in\mathbb{N}}$ is geometrically uniformly ergodic and admits a unique stationary distribution $\tau_\theta\in\mathsf{M}_1(\mathcal{Z})$.*

Proof. In order to establish uniform ergodicity we show that T_θ allows the state space Z as a ν_1-small set for some $\nu_1\in\mathsf{M}(\mathcal{Z})$. Indeed for any $z_t = (x_t, y_t, \tilde{x}_{t-1}^{1:N}, v_t^{1:N})\in\mathsf{Z}$ and $A\in\mathcal{Z}$,

$$\begin{aligned}
T_\theta(z_t, A) &= \int\cdots\int \mathbb{1}_A(x_{t+1}, y_{t+1}, \tilde{x}_t^{1:N}, v_{t+1}^{1:N}) \bar{m}(x_t, x_{t+1}) \bar{g}(x_{t+1}, y_{t+1})\,\mu(dx_{t+1})\,\eta(dy_{t+1}) \\
&\quad \times \int_{x_t^{1:N}} \prod_{k=1}^N r_\theta(\tilde{x}_{t-1}^k, x_t^k, y_t) \prod_{j=1}^N \left(\sum_{i=1}^N \frac{w_\theta(\tilde{x}_{t-1}^i, x_t^i, y_t)}{\sum_{\ell=1}^N w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \delta_{x_t^i}(d\tilde{x}_t^j) \right) \times \mu^{\otimes N}(dx_t^{1:N})\,\nu^{\otimes N}(dv_{t+1}^{1:N}) \\
&\ge \epsilon^{7N+1} \int\cdots\int \mathbb{1}_A(x_{t+1}, y_{t+1}, \tilde{x}_t^{1:N}, v_{t+1}^{1:N}) \bar{g}(x_{t+1}, y_{t+1})\,\mu(dx_{t+1})\,\eta(dy_{t+1}) \\
&\quad \times \int_{x_t^{1:N}} \prod_{j=1}^N \left(\frac{1}{N} \sum_{i=1}^N \delta_{x_t^i}(d\tilde{x}_t^j) \right) \mu^{\otimes N}(dx_t^{1:N})\,\nu^{\otimes N}(dv_{t+1}^{1:N}).
\end{aligned}$$

Thus, we may conclude that T_θ allows Z as a ν_1-small set with

$$\nu_1(dz) = \epsilon^{7N+1} \bar{g}(x,y)\,\mu(dx)\,\eta(dy) \int_{x^{1:N}} \prod_{j=1}^N \left(\frac{1}{N} \sum_{i=1}^N \delta_{x^i}(d\tilde{x}^j) \right) \mu^{\otimes N}(dx^{1:N})\,\nu^{\otimes N}(dv^{1:N}). \tag{8}$$

Then, by Meyn & Tweedie (2009, Theorem 16.0.2(v)) or Douc et al. (2018, Theorem 15.3.1(iii)) it follows that $(Z_t^\theta)_{t\geq 0}$ is geometrically uniformly ergodic. Then the Dobrushin coefficient of T_θ is strictly less than one for all $\theta \in \Theta$, which implies that T_θ admits a unique stationary distribution τ_θ (see, *e.g.*, Douc et al., 2018, Theorems 18.2.4–5). □

A.2.2. Stochastic approximation update and mean field

Now, define, for $z = (x, y, \tilde{x}^{1:N}, v^{1:N}) \in \mathsf{Z}$,

$$H_\theta(z) := \nabla \ln \left(\sum_{i=1}^N w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y) \right) = \frac{\sum_{i=1}^N \nabla w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y)}{\sum_{i=1}^N w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y)},$$

where, here and everywhere in the following, gradients are with respect to θ, *i.e.*, $\nabla = \nabla_\theta$. Then Algorithm 3 is equivalent with the Robbins–Monro (Robbins & Monro, 1951) stochastic-approximation scheme

$$\theta_{t+1} \leftarrow \theta_t + \gamma_{t+1} H_{\theta_t}(Z_{t+1}), \quad t \in \mathbb{N},$$

where $Z_{t+1} \sim T_{\theta_t}(Z_t, \cdot)$, initialized by some starting guess θ_0. This procedure aims to find a zero of the mean field

$$h(\theta) := \int H_\theta(z)\, \tau_\theta(dz).$$

The next result, Proposition A.6, establishes the boundedness of the mean field $h(\theta)$, the convergence of the expectation of $H_\theta(Z_t^\theta)$ to the same and the existence of a solution to the Poisson equation associated with T_θ. This intermediate result will be instrumental in the proofs of Section A.3. Proposition A.6 will be proven under the assumption that the gradients of the model and proposal densities are bounded and Lipschitz in θ.

Assumption A.5. There exists $\kappa \in [1, \infty)$ such that for all $(\theta, \theta') \in \Theta^2$, $(x, x') \in \mathsf{X}^2$, $y \in \mathsf{Y}$ and $v \in \mathsf{E}$,

$$\begin{aligned}
&\max\{\|\nabla m_\theta(x, f_\theta(x, y, v))\|, \|\nabla g_\theta(f_\theta(x, y, v), y)\|, \|\nabla r_\theta(x, f_\theta(x, y, v), y)\|\} \leq \kappa, \\
&\max\{|m_\theta(x, x') - m_{\theta'}(x, x')|, |m_\theta(x, f_\theta(x, y, v)) - m_{\theta'}(x, f_{\theta'}(x, y, v))|, \\
&\qquad \|\nabla m_\theta(x, f_\theta(x, y, v)) - \nabla m_{\theta'}(x, f_{\theta'}(x, y, v))\|\} \leq \kappa \|\theta - \theta'\|, \\
&\max\{|g_\theta(x, y) - g_{\theta'}(x, y)|, |g_\theta(f_\theta(x, y, v), y) - g_{\theta'}(f_{\theta'}(x, y, v), y)|, \\
&\qquad \|\nabla g_\theta(f_\theta(x, y, v), y) - \nabla g_{\theta'}(f_{\theta'}(x, y, v), y)\|\} \leq \kappa \|\theta - \theta'\|, \\
&\max\{|r_\theta(x, x', y) - r_{\theta'}(x, x', y)|, |r_\theta(x, f_\theta(x, y, v), y) - r_{\theta'}(x, f_{\theta'}(x, y, v), y)|, \\
&\qquad \|\nabla r_\theta(x, f_\theta(x, y, v), y) - \nabla r_{\theta'}(x, f_{\theta'}(x, y, v), y)\|\} \leq \kappa \|\theta - \theta'\|.
\end{aligned}$$

Proposition A.6. *Under Assumptions A.2 and A.5 the following holds true.*

(i) $h(\theta)$ *is well-defined and bounded on* Θ.

(ii) For every $\theta \in \Theta$, $h(\theta) = \lim_{t\to\infty} \mathbb{E}_\theta[H_\theta(Z_t^\theta)]$.

(iii) There exists a measurable function $\tilde{H}_\theta$ *on* Z *that satisfies the Poisson equation*

$$H_\theta(z) - h(\theta) = \tilde{H}_\theta(z) - T_\theta \tilde{H}_\theta(z),$$

for every $\theta \in \Theta$ *and* $z \in \mathsf{Z}$.

(iv) There exists a real number $\tilde{\alpha} \in [1, \infty)$ *such that for every* $(\theta, \theta') \in \Theta^2$ *and* $z \in \mathsf{Z}$,

$$\begin{aligned}
&\max\{\|H_\theta(z)\|, \|\tilde{H}_\theta(z)\|, \|T_\theta \tilde{H}_\theta(z)\|\} \leq \tilde{\alpha}, \\
&\max\{\|H_\theta(z) - H_{\theta'}(z)\|, \|T_\theta \tilde{H}_\theta(z) - T_{\theta'} \tilde{H}_{\theta'}(z)\|, \|h(\theta) - h(\theta')\|\} \leq \tilde{\alpha} \|\theta - \theta'\|.
\end{aligned}$$

The proof of Proposition A.6 is found in Section A.4.

A.3. Proof of the main results

We are now ready to prove Theorem 4.7 and Corollary 4.8, which are restated in some more detail below. Our proofs will be based on theory developed by Karimi et al. (2019), where the minimization of a non-convex smooth objective function is considered. Thus, we assume, in Assumption A.7, that the mean field $h(\theta)$ is indeed the gradient of some smooth function of θ, the so-called *Lyapunov* function (depending on N), maximized by the algorithm. However, since the 'surrogate' gradient considered in VSMC, whose time-normalized asymptotic limit is the target of OVSMC, is a biased approximation of the gradient of the ELBO $\mathcal{L}^{\text{SMC}}$ (see Section 2.2), the Lyapunov function does not have a straightforward interpretation in this case.

Assumption A.7. There exists a bounded function V on Θ (the *Lyapunov* function) such that $h = \nabla V$.

Assumption A.8. For every $t \in \mathbb{N}_{>0}$, $\gamma_{t+1} \leq \gamma_t$. In addition, there exist constants $a > 0$ and $a' > 0$ such that for every t,

$$\gamma_t \leq a\gamma_{t+1}, \quad \gamma_t - \gamma_{t+1} \leq a'\gamma_{t+1}^2, \quad \gamma_1 \leq 1/(2\tilde{\alpha} + 2c_h),$$

where the constant $\tilde{\alpha} \in [1, \infty)$ is given in Proposition A.6 and $c_h := \tilde{\alpha}(a+1)/2 + \tilde{\alpha}^2(2a+1) + \tilde{\alpha}a'$.

Theorem A.9 (Theorem 4.7)**.** *Let Assumptions A.1, A.2, A.5, A.7 and A.8 hold. Then for every $t \in \mathbb{N}$,*

$$\mathbb{E}[\|h(\theta_\tau)\|^2] \leq \frac{2(d_{0,t} + c_{0,t} + (4\tilde{\alpha}^3 + c_\gamma)\sum_{s=0}^t \gamma_{s+1}^2)}{\sum_{s=0}^t \gamma_{s+1}},$$

where $\tau \sim \mathsf{cat}((\gamma_{s+1})_{s=0}^t)$, $\tilde{\alpha}$ is defined in Proposition A.6 and

$$d_{0,t} := \mathbb{E}[V(\theta_{t+1}) - V(\theta_0)], \quad c_\gamma := \tilde{\alpha}^2(2 + \tilde{\alpha}), \quad c_{0,t} := \tilde{\alpha}(\gamma_1 - \gamma_{t+1} + 2).$$

Proof. The proof follows directly from Karimi et al. (2019, Theorem 2), with V, H_θ and h being multiplied by -1 as we deal with a maximization problem. We notice that (Karimi et al., 2019, Assumptions A1–A3) are satisfied by our Assumption A.7, with $c_0 = d_0 = 0$ and $c_1 = d_1 = 1$, and by Proposition A.6, letting $L = \tilde{\alpha}$. Moreover, (Karimi et al., 2019, Assumptions A5–A7) are satisfied by Proposition A.6, letting $L_{PH}^{(0)} = L_{PH}^{(1)} = \sigma = \tilde{\alpha}$. □

Corollary A.10 (Corollary 4.8)**.** *Let the assumptions of Theorem A.9 hold and let the step-size sequence $(\gamma_t)_{t\in\mathbb{N}}$ be given by $\gamma_t = 1/(\sqrt{t}(2\tilde{\alpha} + 2c_h))$, where $\tilde{\alpha}$ and c_h are provided in Assumption A.8. Then for every $t \in \mathbb{N}_{>0}$,*

$$\mathbb{E}[\|h(\theta_\tau)\|^2] = \mathcal{O}\left(\frac{\log t}{\sqrt{t}}\right),$$

where $\tau \sim \mathsf{cat}((\gamma_{s+1})_{s=0}^t)$.

Proof. Noticing that Assumption A.8 is satisfied with $a = 2$ and $a' = 2(\tilde{\alpha} + c_h)$, the result is a direct implication of Theorem A.9. □

A.4. Auxiliary results

In this section we establish Proposition A.6. The proof is prefaced by two lemmas.

Lemma A.11. *Let Assumptions A.2 and A.5 hold. Then there exists $\tilde{\kappa} \in [1, \infty)$ such that for all $(\theta, \theta') \in \Theta^2$, $(x, x') \in \mathsf{X}^2$, $y \in \mathsf{Y}$ and $v \in \mathsf{E}$,*

$$\|\nabla w_\theta(x, f_\theta(x, y, v), y)\| \leq \tilde{\kappa}$$

and

$$\max\{|w_\theta(x, x', y) - w_{\theta'}(x, x', y)|, |w_\theta(x, f_\theta(x, y, v), y) - w_{\theta'}(x, f_{\theta'}(x, y, v), y)|, \\ \|\nabla w_\theta(x, f_\theta(x, y, v), y) - \nabla w_{\theta'}(x, f_{\theta'}(x, y, v), y)\|\} \leq \tilde{\kappa}\|\theta - \theta'\|.$$

Proof. First, write

$$\nabla w_\theta(x, f_\theta(x,y,v), y) = \frac{1}{r_\theta(x, f_\theta(x,y,v), y)}\Big(g_\theta(f_\theta(x,y,v),y)\nabla m_\theta(x, f_\theta(x,y,v)) + m_\theta(x, f_\theta(x,y,v))\nabla g_\theta(f_\theta(x,y,v),y) - w_\theta(x, f_\theta(x,y,v),y)\nabla r_\theta(x, f_\theta(x,y,v),y)\Big),$$

implying, from Assumptions A.2 and A.5,

$$\|\nabla w_\theta(x, f_\theta(x,y,v), y)\| \le \frac{1}{\epsilon^2}(\|\nabla m_\theta(x, f_\theta(x,y,v))\| + \|\nabla g_\theta(f_\theta(x,y,v),y)\|) + \frac{1}{\epsilon^4}\|\nabla r_\theta(x, f_\theta(x,y,v),y)\| = \frac{\kappa(2\epsilon^2+1)}{\epsilon^4}.$$

In order to show that w_θ is Lipschitz, we apply Assumptions A.2 and A.5 according to

$$|w_\theta(x,x',y) - w_{\theta'}(x,x',y)| \le \frac{m_\theta(x,x')\,|g_\theta(x',y) - g_{\theta'}(x',y)| + g_{\theta'}(x',y)\,|m_\theta(x,x') - m_{\theta'}(x,x')|}{r_\theta(x,x',y)} + m_{\theta'}(x,x')g_{\theta'}(x',y)\frac{|r_{\theta'}(x,x',y) - r_\theta(x,x',y)|}{r_\theta(x,x',y)r_{\theta'}(x,x',y)} \le \frac{\kappa(2\epsilon^2+1)}{\epsilon^4}\|\theta-\theta'\|.$$

Proceeding similarly, we obtain

$$|w_\theta(x, f_\theta(x,y,v), y) - w_{\theta'}(x, f_{\theta'}(x,y,v), y)| \le \frac{(2\epsilon^2+1)\kappa}{\epsilon^4}\|\theta-\theta'\|. \tag{9}$$

Moreover, in order to show that also the gradient ∇w_θ is Lipschitz, consider the decomposition

$$\begin{aligned}
&\|\nabla w_\theta(x, f_\theta(x,y,v), y) - \nabla w_{\theta'}(x, f_{\theta'}(x,y,v), y)\| \\
&\quad\le \left\| \frac{g_\theta(f_\theta(x,y,v),y)\nabla m_\theta(x, f_\theta(x,y,v))}{r_\theta(x, f_\theta(x,y,v),y)} - \frac{g_{\theta'}(f_{\theta'}(x,y,v),y)\nabla m_{\theta'}(x, f_{\theta'}(x,y,v))}{r_{\theta'}(x, f_{\theta'}(x,y,v),y)} \right\| \\
&\qquad + \left\| \frac{m_\theta(x, f_\theta(x,y,v))\nabla g_\theta(f_\theta(x,y,v),y)}{r_\theta(x, f_\theta(x,y,v),y)} - \frac{m_{\theta'}(x, f_{\theta'}(x,y,v))\nabla g_{\theta'}(f_{\theta'}(x,y,v),y)}{r_{\theta'}(x, f_{\theta'}(x,y,v),y)} \right\| \\
&\qquad + \left\| \frac{w_{\theta'}(x, f_{\theta'}(x,y,v),y)\nabla r_{\theta'}(x, f_{\theta'}(x,y,v),y)}{r_{\theta'}(x, f_{\theta'}(x,y,v),y)} - \frac{w_\theta(x, f_\theta(x,y,v),y)\nabla r_\theta(x, f_\theta(x,y,v),y)}{r_\theta(x, f_\theta(x,y,v),y)} \right\|,
\end{aligned} \tag{10}$$

where, by Assumptions A.2 and A.5,

$$\begin{aligned}
&\left\| \frac{g_\theta(f_\theta(x,y,v),y)\nabla m_\theta(x, f_\theta(x,y,v))}{r_\theta(x, f_\theta(x,y,v),y)} - \frac{g_{\theta'}(f_{\theta'}(x,y,v),y)\nabla m_{\theta'}(x, f_{\theta'}(x,y,v))}{r_{\theta'}(x, f_{\theta'}(x,y,v),y)} \right\| \\
&\quad\le \frac{g_\theta(f_\theta(x,y,v),y)\|\nabla m_\theta(x, f_\theta(x,y,v)) - \nabla m_{\theta'}(x, f_{\theta'}(x,y,v))\|}{r_\theta(x, f_\theta(x,y,v),y)} \\
&\qquad + \frac{\|\nabla m_{\theta'}(x, f_{\theta'}(x,y,v))\|\,|g_\theta(f_\theta(x,y,v),y) - g_{\theta'}(f_{\theta'}(x,y,v),y)|}{r_\theta(x, f_\theta(x,y,v),y)} \\
&\qquad + \|\nabla m_{\theta'}(x, f_{\theta'}(x,y,v))\| g_{\theta'}(f_{\theta'}(x,y,v),y)\frac{|r_{\theta'}(x, f_{\theta'}(x,y,v),y) - r_\theta(x, f_\theta(x,y,v),y)|}{r_\theta(x, f_\theta(x,y,v),y)r_{\theta'}(x, f_{\theta'}(x,y,v),y)} \\
&\quad\le \frac{\kappa}{\epsilon^3}(\epsilon + \epsilon^2\kappa + \kappa)\|\theta-\theta'\|.
\end{aligned}$$

Using (9), the other terms of (10) may be treated similarly, yielding

$$\begin{aligned}
&\|\nabla w_\theta(x, f_\theta(x,y,v), y) - \nabla w_{\theta'}(x, f_{\theta'}(x,y,v), y)\| \\
&\quad\le \left(\frac{2\kappa}{\epsilon^3}(\epsilon + \epsilon^2\kappa + \kappa) + \frac{\kappa}{\epsilon^5}(\epsilon + 2\epsilon^2\kappa + 2\kappa)\right)\|\theta-\theta'\| \\
&\quad= (\epsilon + 2\epsilon^3 + 2\kappa + 2\epsilon^4\kappa + 4\epsilon^2\kappa)\frac{\kappa}{\epsilon^5}\|\theta-\theta'\|.
\end{aligned}$$

The proof is the concluded by letting

$$\tilde{\kappa} := \max\left\{\frac{(2\epsilon^2+1)\kappa}{\epsilon^4}, (\epsilon + 2\epsilon^3 + 2\kappa + 2\epsilon^4\kappa + 4\epsilon^2\kappa)\frac{\kappa}{\epsilon^5}\right\} = (\epsilon + 2\epsilon^3 + 2\kappa + 2\epsilon^4\kappa + 4\epsilon^2\kappa)\frac{\kappa}{\epsilon^5}.$$

□

Our second prefatory lemma establishes Lipschitz continuity and exponential contraction of the Markov dynamics underlying the state-dependent process $(Z_t)_{t\in\mathbb{N}}$. Recall that under Assumptions A.1 and A.2, Proposition A.4 provides the existence of a unique stationary distribution τ_θ of the canonical Markov chain $(Z_t^\theta)_{t\in\mathbb{N}}$ induced by T_θ. We may then define, for $t \in \mathbb{N}_{>0}$,

$$\tilde{T}_\theta^t : \mathsf{Z} \times \mathcal{Z} \ni (z, A) \mapsto T_\theta^t(z, A) - \tau_\theta(A),$$

where T_θ^t is the t-skeleton defined recursively as $T_\theta^1 = T_\theta$ and $T_\theta^{s+1}(z, A) = \int T_\theta^s(z, dz')\, T_\theta(z', A)$ for $(z, A) \in \mathsf{Z} \times \mathcal{Z}$. By convention, we let $T_\theta^0(z, A) := \delta_z(A)$.

Lemma A.12. *Let Assumptions A.2 and A.5 hold. Then there exists a constant $\varsigma \in [1, \infty)$ (possibly depending on N) such that for every $(\theta, \theta') \in \Theta^2$, $z \in \mathsf{Z}$, bounded measurable function h on Z and $t \in \mathbb{N}$,*

(i) $|\tilde{T}_\theta^t h(z)| \le \varrho^t \|h\|_\infty$,

(ii) $|\tilde{T}_\theta^t h(z) - \tilde{T}_{\theta'}^t h(z)| \le \varsigma \varrho^{t/2} \|h\|_\infty \|\theta - \theta'\|$,

(iii) $\max\{|\tau_\theta h - \tau_{\theta'} h|, |T_\theta h(z) - T_{\theta'} h(z)|\} \le \varsigma \|h\|_\infty \|\theta - \theta'\|$,

where $\varrho = 1 - \epsilon^{7N+1}$.

Proof. The first bound (i) follows by Proposition A.4, establishing that T_θ allows Z as a ν_1-small set, with ν_1 being defined in (8). Then by Meyn & Tweedie (2009, Theorem 16.2.4) it holds that for all $z \in \mathsf{Z}$ and bounded measurable functions h on Z,

$$|\tilde{T}_\theta^t h(z)| = |T_\theta^t h(z) - \tau_\theta h| \le \varrho^t \|h\|_\infty,$$

where $\varrho := 1 - \nu_1(\mathsf{Z}) = 1 - \epsilon^{7N+1}$.

We turn to (ii) and (iii) and introduce the short-hand notations

$$a_\theta^k := r_\theta(\tilde{x}_{t-1}^k, x_t^k, y_t),$$
$$\beta_\theta^N := \left(\sum_{i=1}^N \frac{w_\theta(\tilde{x}_{t-1}^i, x_t^i, y_t)}{\sum_{\ell=1}^N w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \delta_{x_t^i}\right)^{\otimes N},$$

which depend implicitly on $\tilde{x}_{t-1}^{1:N}$ and $x_t^{1:N}$. We may then write, for given $z_t \in \mathsf{Z}$ and bounded measurable function h on Z,

$$|T_\theta h(z_t) - T_{\theta'} h(z_t)| \le \int \cdots \int h(x_{t+1}, y_{t+1}, \tilde{x}_t^{1:N}, v_{t+1}^{1:N}) \bar{m}(x_t, x_{t+1}) \bar{g}(x_{t+1}, y_{t+1})\, \mu(dx_{t+1})\, \eta(dy_{t+1})$$
$$\times \int_{x_t^{1:N}} \left|\beta_\theta^N \prod_{k=1}^N a_\theta^k - \beta_{\theta'}^N \prod_{k=1}^N a_{\theta'}^k\right| (d\tilde{x}_t^{1:N})\, \mu^{\otimes N}(dx_t^{1:N})\, \nu^{\otimes N}(dv_{t+1}^{1:N}). \quad (11)$$

Here the total variation measure inside the integral can be bounded according to

$$\left|\beta_\theta^N \prod_{k=1}^N a_\theta^k - \beta_{\theta'}^N \prod_{k=1}^N a_{\theta'}^k\right| \le |\beta_\theta^N - \beta_{\theta'}^N| \prod_{k=1}^N a_\theta^k + \left|\prod_{k=1}^N a_\theta^k - \prod_{k=1}^N a_{\theta'}^k\right| \beta_{\theta'}^N$$
$$\le \frac{1}{\epsilon^N} |\beta_\theta^N - \beta_{\theta'}^N| + \left(\sum_{i'=1}^N |a_\theta^{i'} - a_{\theta'}^{i'}| \prod_{k=1}^{i'-1} a_\theta^k \prod_{k=i'+1}^N a_{\theta'}^k\right) \beta_{\theta'}^N$$
$$\le \frac{1}{\epsilon^N} |\beta_\theta^N - \beta_{\theta'}^N| + \frac{1}{\epsilon^{N-1}} \sum_{k=1}^N |a_\theta^k - a_{\theta'}^k|\, \beta_{\theta'}^N, \quad (12)$$

where we have applied Assumption A.2. To bound the second term in (12), we first note that by Assumption A.5,

$$\sum_{k=1}^{N} \left| a_\theta^k - a_{\theta'}^k \right| \leq N\kappa \|\theta - \theta'\|. \tag{13}$$

Moreover, by rewriting the measure β_θ^N as

$$\beta_\theta^N = \left(\sum_{i=1}^{N} \frac{w_\theta(\tilde{x}_{t-1}^i, x_t^i, y_t)}{\sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \delta_{x_t^i} \right)^{\otimes N} = \sum_{i_{1:N} \in \{1,\dots,N\}^N} \bar{w}_\theta^{i_{1:N}} \delta_{x_t^{i_{1:N}}},$$

where we have defined $\bar{w}_\theta^{i_{1:N}} := \prod_{j=1}^{N} (w_\theta(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t) / \sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t))$, we may bound the same as $\beta_\theta^N \leq \mu(\{x_t^1, \dots, x_t^N\}^N)/\epsilon^{6N}$, where we have defined the occupation measure

$$\mu(\{x_t^1, \dots, x_t^N\}^N) := \frac{1}{N^N} \sum_{i_{1:N} \in \{1,\dots,N\}^N} \delta_{x_t^{i_{1:N}}}.$$

Consequently, by combining this with (13) we obtain the bound

$$\frac{1}{\epsilon^{N-1}} \sum_{k=1}^{N} \left| a_\theta^k - a_{\theta'}^k \right| \beta_{\theta'}^N \leq \frac{\kappa N}{\epsilon^{7N-1}} \|\theta - \theta'\| \mu(\{x_t^1, \dots, x_t^N\}^N). \tag{14}$$

on the second term in (12). We now bound the first term in (12). For this purpose, write

$$\left| \beta_\theta^N - \beta_{\theta'}^N \right| = \sum_{i_{1:N} \in \{1,\dots,N\}^N} \left| \bar{w}_\theta^{i_{1:N}} - \bar{w}_{\theta'}^{i_{1:N}} \right| \delta_{x_t^{i_{1:N}}}, \tag{15}$$

where, by Lemma A.11,

$$\begin{aligned}
\left| \bar{w}_\theta^{i_{1:N}} - \bar{w}_{\theta'}^{i_{1:N}} \right| &= \left| \prod_{j=1}^{N} \frac{w_\theta(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t)}{\sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} - \prod_{j=1}^{N} \frac{w_{\theta'}(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t)}{\sum_{\ell=1}^{N} w_{\theta'}(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \right| \\
&\leq \sum_{j=1}^{N} \left(\prod_{j'=1}^{j-1} \frac{w_\theta(\tilde{x}_{t-1}^{i_{j'}}, x_t^{i_{j'}}, y_t)}{\sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \prod_{j'=j+1}^{N} \frac{w_{\theta'}(\tilde{x}_{t-1}^{i_{j'}}, x_t^{i_{j'}}, y_t)}{\sum_{\ell=1}^{N} w_{\theta'}(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \right. \\
&\qquad \left. \times \left| \frac{w_\theta(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t)}{\sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} - \frac{w_{\theta'}(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t)}{\sum_{\ell=1}^{N} w_{\theta'}(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \right| \right) \\
&\leq \frac{1}{(\epsilon^6 N)^{N-1}} \sum_{j=1}^{N} \left(\frac{|w_\theta(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t) - w_{\theta'}(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t)|}{\sum_{\ell=1}^{N} w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t)} \right. \\
&\qquad \left. + w_{\theta'}(\tilde{x}_{t-1}^{i_j}, x_t^{i_j}, y_t) \frac{\sum_{\ell=1}^{N} \left| w_{\theta'}(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t) - w_\theta(\tilde{x}_{t-1}^\ell, x_t^\ell, y_t) \right|}{\sum_{\ell'=1}^{N} w_\theta(\tilde{x}_{t-1}^{\ell'}, x_t^{\ell'}, y_t) \sum_{\ell''=1}^{N} w_{\theta'}(\tilde{x}_{t-1}^{\ell''}, x_t^{\ell''}, y_t)} \right) \\
&\leq \frac{1}{(\epsilon^6 N)^{N-1}} \sum_{j=1}^{N} \left(\frac{\tilde{\kappa}}{N\epsilon^3} \|\theta - \theta'\| + \frac{\tilde{\kappa}}{N\epsilon^9} \|\theta - \theta'\| \right) \\
&= \frac{\tilde{\kappa} N}{N^N \epsilon^{6N-3}} \left(1 + \frac{1}{\epsilon^6} \right) \|\theta - \theta'\|,
\end{aligned}$$

implying, via, (15), that the first term in (12) can be bounded as

$$\frac{1}{\epsilon^N} \left| \beta_\theta^N - \beta_{\theta'}^N \right| \leq \frac{\tilde{\kappa} N}{\epsilon^{7N-3}} \left(1 + \frac{1}{\epsilon^6} \right) \|\theta - \theta'\| \mu(\{x_t^1, \dots, x_t^N\}^N). \tag{16}$$

Thus, combining (12), (14) and (16) yields

$$\left|\beta_\theta^N \prod_{k=1}^N a_\theta^k - \beta_{\theta'}^N \prod_{k=1}^N a_{\theta'}^k\right| \leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}\|\theta - \theta'\|\mu(\{x_t^1, \ldots, x_t^N\}^N). \tag{17}$$

Now, by plugging the bound (17) into (11) we obtain

$$\begin{aligned}|T_\theta h(z_t) - T_{\theta'} h(z_t)| &\leq \|h\|_\infty \int \cdots \int \bar{m}(x_t, x_{t+1})\bar{g}(x_{t+1}, y_{t+1})\,\mu(dx_{t+1})\,\eta(dy_{t+1}) \\ &\quad \times (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}\|\theta - \theta'\| \int_{x_t^{1:N}} \mu(\{x_t^1, \ldots, x_t^N\}^N)(d\tilde{x}_t^{1:N})\,\mu^{\otimes N}(dx_t^{1:N})\,\nu^{\otimes N}(dv_{t+1}^{1:N}) \\ &= (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}\|h\|_\infty\|\theta - \theta'\|. \end{aligned} \tag{18}$$

Now, using the decomposition, for $t \in \mathbb{N}_{>0}$,

$$T_\theta^{t+1} - T_{\theta'}^{t+1} = \sum_{s=0}^t \left(T_{\theta'}^{t-s} T_\theta^{s+1} - T_{\theta'}^{t-s+1} T_\theta^s\right) = \sum_{s=0}^t \left(T_{\theta'}^{t-s} T_\theta \tilde{T}_\theta^s - T_{\theta'}^{t-s} T_{\theta'} \tilde{T}_\theta^s\right) = \sum_{s=0}^t T_{\theta'}^{t-s}(T_\theta - T_{\theta'})\tilde{T}_\theta^s$$

we obtain, using (18) and (i), the bound

$$\begin{aligned}\left|T_\theta^{t+1} h(z) - T_{\theta'}^{t+1} h(z)\right| &\leq \sum_{s=0}^t \int T_{\theta'}^{t-s}(z, dz')|T_\theta \tilde{T}_\theta^s h(z') - T_{\theta'} \tilde{T}_\theta^s h(z')| \\ &\leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}\|h\|_\infty\|\theta - \theta'\| \sum_{s=0}^t \varrho^s \\ &\leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}(1-\varrho)}\|h\|_\infty\|\theta - \theta'\|.\end{aligned}$$

Similarly,

$$\begin{aligned}|\tilde{T}_\theta^{t+1} h(z) - \tilde{T}_{\theta'}^{t+1} h(z)| &= \left|\sum_{s=0}^t \iint \tilde{T}_\theta^s h(z'')(T_\theta - T_{\theta'})(z', dz'')\tilde{T}_{\theta'}^{t-s}(z, dz')\right| \\ &\leq \sum_{s=0}^t \left|\int \tilde{T}_{\theta'}^{t-s}(z, dz')(T_\theta \tilde{T}_\theta^s h(z') - T_{\theta'} \tilde{T}_\theta^s h(z'))\right| \\ &\leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}\|\theta - \theta'\| \sum_{s=0}^t \varrho^{t-s}\|\tilde{T}_{\theta'}^s h(\cdot)\|_\infty \\ &\leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}}(t+1)\varrho^t\|h\|_\infty\|\theta - \theta'\|.\end{aligned}$$

Finally, we can write, for arbitrary $t \in \mathbb{N}_{>0}$ and $z \in \mathsf{Z}$,

$$\begin{aligned}|\tau_\theta h - \tau_{\theta'} h| &\leq |T_\theta^t h(z) - T_{\theta'}^t h(z)| + |\tilde{T}_\theta^t h(z)| + |\tilde{T}_{\theta'}^t h(z)| \\ &\leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}(1-\varrho)}\|h\|_\infty\|\theta - \theta'\| + 2\varrho^t\|h\|_\infty,\end{aligned}$$

implying that

$$|\tau_\theta h - \tau_{\theta'} h| \leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}(1-\varrho)}\|h\|_\infty\|\theta - \theta'\|.$$

The proof of (ii) and (iii) is now concluded by simply noting that

$$|\tilde{T}_\theta^t h(z) - \tilde{T}_{\theta'}^t h(z)| \leq (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}} t\varrho^{t/2-1}\varrho^{t/2}\|h\|_\infty\|\theta - \theta'\|,$$

and letting

$$\varsigma := (\epsilon^6\tilde{\kappa} + \tilde{\kappa} + \epsilon^4\kappa)\frac{N}{\epsilon^{7N+3}} \max\left\{\sup_{t\in\mathbb{N}} t\varrho^{t/2-1}, \frac{1}{1-\varrho}\right\}.$$

□

We are now ready to prove Proposition A.6.

Proof of Proposition A.6. Using Assumptions A.2 and A.5 and Lemma A.11 we conclude that for all $(\theta, \theta') \in \Theta^2$ and $z \in \mathsf{Z}$,

$$\|H_\theta(z)\| \leq \frac{\sum_{i=1}^N \|\nabla w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y)\|}{\sum_{\ell=1}^N w_\theta(\tilde{x}^\ell, f_\theta(\tilde{x}^\ell, y, v^\ell), y)} \leq \frac{\tilde{\kappa}}{\epsilon^3}, \tag{19}$$

from which (i) immediately follows.

We turn to (ii). Using Lemma A.12(i) and (19), for every $t \in \mathbb{N}$ and $z \in \mathsf{Z}$,

$$\|\mathbb{E}_\theta[H_\theta(Z_t^\theta) \mid Z_0^\theta = z] - h(\theta)\| = \|T_\theta^t H_\theta(z) - h(\theta)\| = \|\tilde{T}_\theta^t H_\theta(z)\| \leq \frac{\tilde{\kappa}}{\epsilon^3}\varrho^t.$$

Thus, for every $\theta \in \Theta$,

$$\begin{aligned} 0 \leq \liminf_{t\to\infty} \|\mathbb{E}_\theta[H_\theta(Z_t^\theta)] - h(\theta)\| \leq \limsup_{t\to\infty} \|\mathbb{E}_\theta[H_\theta(Z_t^\theta)] - h(\theta)\| \\ = \limsup_{t\to\infty} \|\mathbb{E}_\theta[\mathbb{E}_\theta[H_\theta(Z_t^\theta) \mid Z_0^\theta] - h(\theta)]\| \leq \limsup_{t\to\infty} \frac{\tilde{\kappa}}{\epsilon^3}\varrho^t = 0, \end{aligned}$$

which proves (ii).

In order to establish (iii), let

$$\tilde{H}_\theta(z) := \sum_{s=0}^\infty (T_\theta^s H_\theta(z) - h(\theta)), \quad z \in \mathsf{Z}.$$

Indeed, again by Lemma A.12(i), for every $z \in \mathsf{Z}$,

$$\|\tilde{H}_\theta(z)\| \leq \sum_{s=0}^\infty \|T_\theta^s H_\theta(z) - h(\theta)\| \leq \frac{\tilde{\kappa}}{\epsilon^3}\sum_{s=0}^\infty \varrho^s = \frac{\tilde{\kappa}}{\epsilon^3(1-\varrho)}.$$

which implies that $\tilde{H}_\theta$ and $T_\theta\tilde{H}_\theta$ are well defined and bounded. Moreover, by the dominated convergence theorem, for every $z \in \mathsf{Z}$,

$$T_\theta\tilde{H}_\theta(z) = \sum_{s=1}^\infty (T_\theta^s H_\theta(z) - h(\theta)),$$

implying (iii).

To prove (iv), write, using Lemma A.11,

$$\begin{aligned} \|H_\theta(z) - H_{\theta'}(z)\| &\leq \frac{\sum_{i=1}^N \|\nabla w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y) - \nabla w_{\theta'}(\tilde{x}^i, f_{\theta'}(\tilde{x}^i, y, v^i), y)\|}{\sum_{\ell=1}^N w_\theta(\tilde{x}^\ell, f_\theta(\tilde{x}^\ell, y, v^\ell), y)} \\ &\quad + \frac{\|H_{\theta'}(z)\| \sum_{i=1}^N \left|w_\theta(\tilde{x}^i, f_\theta(\tilde{x}^i, y, v^i), y) - w_{\theta'}(\tilde{x}^i, f_{\theta'}(\tilde{x}^i, y, v^i), y)\right|}{\sum_{\ell=1}^N w_\theta(\tilde{x}^\ell, f_\theta(\tilde{x}^\ell, y, v^\ell), y)} \\ &\leq \left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right)\|\theta - \theta'\|, \end{aligned} \tag{20}$$

and by (20) and Lemma A.12 it holds that for every $z \in \mathsf{Z}$,

$$\begin{aligned}\|\tilde{T}_\theta^t H_\theta(z) - \tilde{T}_{\theta'}^t H_{\theta'}(z)\| &\leq \|\tilde{T}_\theta^t H_\theta(z') - \tilde{T}_\theta^t H_{\theta'}(z')\| - \|\tilde{T}_\theta^t H_{\theta'}(z) - \tilde{T}_{\theta'}^t H_{\theta'}(z)\| \\ &\leq \left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right) \varrho^t \|\theta - \theta'\| + \frac{\tilde{\kappa}}{\epsilon^3} \varsigma \varrho^{t/2} \|\theta - \theta'\| \\ &\leq 2\left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right) \varsigma \varrho^{t/2} \|\theta - \theta'\|.\end{aligned}$$

Thus, we may write

$$\begin{aligned}\|T_\theta \tilde{H}_\theta(z) - T_{\theta'} \tilde{H}_{\theta'}(z)\| &= \left\| \sum_{s=1}^{\infty} (T_\theta^s H_\theta(z) - h(\theta)) - \sum_{s=1}^{\infty} (T_{\theta'}^s H_{\theta'}(z) - h(\theta')) \right\| \\ &\leq \sum_{s=0}^{\infty} \|\tilde{T}_\theta^s H_\theta(z) - \tilde{T}_{\theta'}^s H_{\theta'}(z)\| \\ &\leq \left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right) \frac{2\varsigma}{1 - \sqrt{\varrho}} \|\theta - \theta'\|.\end{aligned}$$

Finally, by Lemma A.12 and (20) again, we have

$$\|h(\theta) - h(\theta')\| \leq \int \|H_\theta(z) - H_{\theta'}(z)\| \, \tau_\theta(dz) + \|\tau_\theta H_{\theta'} - \tau_{\theta'} H_{\theta'}\| \leq \left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right) (1 + \varsigma) \|\theta - \theta'\|,$$

which allows us, by letting

$$\tilde{\alpha} := \left(\frac{\tilde{\kappa}}{\epsilon^3} + \frac{\tilde{\kappa}^2}{\epsilon^6}\right) \frac{2\varsigma}{1 - \sqrt{\varrho}},$$

to conclude the proof of (iv). □

B. ESS improvement for the model in Section 5.1

Figure 8 shows how the effective sample size (ESS, Liu, 1996) of the particle cloud improves in a single run of `OVSMC` while r_λ is being learned in univariate linear Gaussian model of Section 5.1, to finally reach the performance of the optimal proposal. This is evident for $S_v = 0.2$; on the other hand, when $S_v = 1.2$, the particles are well propagated into regions of non-negligible likelihood even with the bootstrap proposal, resulting in the normalized ESS being close to one regardless.

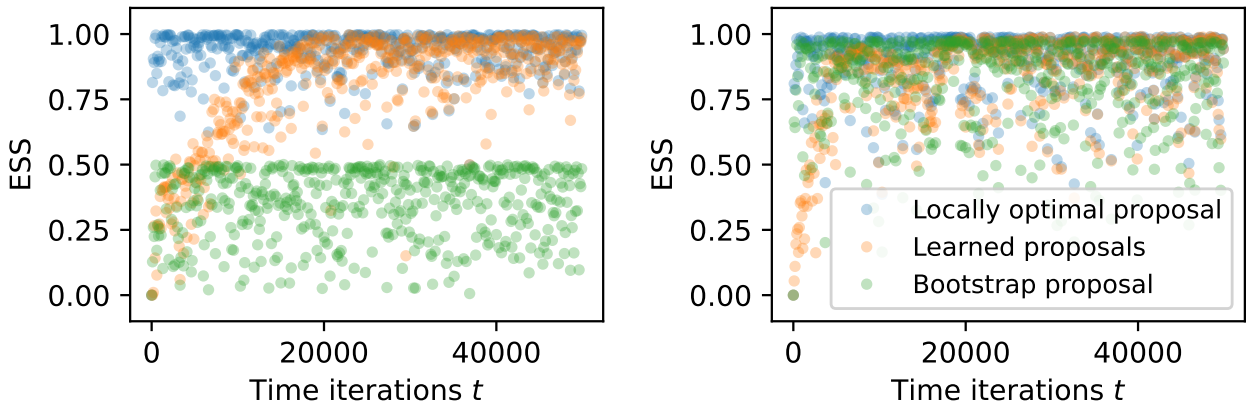

Figure 8: Evolution of (every 100^{th}) normalized ESS for the one-dimensional linear Gaussian SSM in Section 5.1, for $S_v = 0.2$ (left) and $S_v = 1.2$ (right).

C. Details on the deep generative model of Section 5.3

In this section we provide more details on the model presented in Section 5.3.

C.1. Data generation

First, we describe how the frames of the video are generated. The movement can shortly be described as a partially observed Gaussian random walk confined to a rectangle. More specifically, the agent moves on the rectangle $[0,1] \times [0,5] \subset \mathbb{R}^2$, starting from a uniformly sampled point on $[0,1] \times \{5\}$. It then moves according to a bivariate Gaussian random walk with covariance matrix $0.004I$ and vertical drift being initially -0.15 and changing sign every time that either bottom or top edges are hit. In practice, the agent bounces every time it hits any edge. Then each frame is created by projecting the rectangle into a 32×32 array and giving the agent an approximately round shape by overlapping, as a plus sign, two 3×5 rectangles, one vertical and one horizontal. In the created arrays, the background has value zero, while the pixels representing the agent are equal to one. All the frames have an horizontal 16×30 rectangle in the center, which (partially) occludes the view of the agent every time it (partially) falls in that area. This area is represented by 0.5-valued pixels.

C.2. Model architecture

Inspired by Le et al. (2018, Section C.1), the model is constructed on the basis of the *variational recurrent neural networks* (VRNN, Chung et al., 2015) framework. More precisely, the generative model is represented by the process $(X_{t+1}, H_t, Y_{t+1})_{t \in \mathbb{N}_{>0}}$, with joint density

$$p(x_{1:T}, h_{0:T}, y_{1:T}) = p_0(h_0) \prod_{t=0}^{T-1} m_\theta(x_{t+1} \mid h_t) g_\theta(y_{t+1} \mid h_t, x_{t+1}) p_{\lambda,\theta}(h_{t+1} \mid h_t, x_{t+1}, y_{t+1}),$$

where T is a fixed time horizon, $y_{1:T}$ represent the frames of the video, $x_{1:T}$ some lower-dimensional latent states and $h_{0:T}$ are the so-called *hidden states* of the *gated recurrent unit* (GRU), which is the specific *recurrent neural network* (RNN) used within the whole architecture. The initial and transition distributions of the generative model are

$$\begin{aligned} H_0 &\sim N(0, I), \\ X_{t+1} \mid H_t = h_t &\sim N(\mu_\theta^x(h_t), \sigma_\theta^x(h_t)^2), \\ Y_{t+1} \mid H_t = h_t, X_{t+1} = x_{t+1} &\sim \text{Bernoulli}(\mu_\theta^y(\varphi_\theta^x(x_{t+1}), h_t)), \\ H_{t+1} \mid H_t = h_t, X_{t+1} = x_{t+1}, Y_{t+1} = y_{t+1} &\sim \delta_{\text{GRU}_\lambda(h_t, \varphi_\theta^x(x_{t+1}), \varphi_\lambda^y(y_{t+1}))}, \end{aligned}$$

for $t \in \mathbb{N}$, while the proposal $r_\lambda(x_{t+1} \mid y_{t+1}, h_t)$ is given by

$$X_{t+1} \mid Y_{t+1} = y_{t+1}, H_t = h_t \sim N(\mu_\lambda^p(\varphi_\lambda^y(y_{t+1}), h_t), \sigma_\lambda^p(\varphi_\lambda^y(y_{t+1}), h_t)^2).$$

More in detail, the model comprises the following neural networks.

- μ_θ^x and σ_θ^x have two dense layers, the second one being the output, with each 128 nodes, corresponding then to the size of the latent states, whose first layer is shared. The activations of the first layers are ReLU functions, while the activations of the output layer are linear and softplus, respectively.
- φ_θ^x is a single dense output layer with 128 nodes and the ReLU activation function.
- φ_λ^y is the encoder of the frames and is represented by a sequential architecture with four convolutional layers, all with 4×4 filters, stride two and padding one, except the last one which has stride one and zero padding; the numbers of filters are, in order, 32, 128, 64 and 32. The activation functions of the convolutions are leaky ReLUs with slope 0.2 and we add batch normalization layers after each of them (except the last one which simply has linear activation). In the end the output is flattened to obtain a tensor in $\mathbb{R}^{32}$.
- μ_θ^y is the decoder, which is modeled by a sequential architecture with transposed convolutions. In particular, the first layer has 128 4×4 filters with stride one and zero padding. Then we have two layers with 64 and 32 4×4 filters, respectively, while the last one has a single 4×4 filter; all these have stride two and padding one. We use again leaky ReLUs as activations with slope 0.2 and batch normalization layers. The activation function of the output layer is instead a sigmoid, in order to obtain an output in $(0,1)^{32 \times 32}$.

- $\texttt{GRU}_\lambda$ is a *gated recurrent unit* RNN, which takes as input the concatenated outputs of φ_θ^x and φ_λ^y to produce deterministically the next hidden state h_{t+1} of the RNN. Here $\texttt{GRU}_\lambda$ takes h_t as an additional input to model the recurrence, but in practice, during training we need to input a time series of (functions of) latent states and frames. Since we deal with streaming data, it would be infeasible to input the whole history at every iteration, and we hence include only the 40 most recent latent states and frames when learning the $\texttt{GRU}_\lambda$.
- μ_λ^p and σ_λ^p have three dense layers of size 128 including the output and share the first two layers, with ReLUs activations except the last layers, which has linear and softplus functions, respectively.

We note that the subscripts to the neural networks defined above indicate which optimizer—the one for the generative model or the one for the proposal—that is used. Even if the GRU is supposed to be part of the generative model, we noticed a learning improvement by considering it part of the proposal, motivated by the fact that it is the first (deterministic) sampling operation in the propagation of the particles. In order to describe more clearly our procedure, we have displayed its pseudocode in Algorithm 4. In our notation, $(\xi_{t-39:t}^{x,i})_{i=1}^N$ are the particles representing the latent states of the process, while $(\xi_{t-40:t-1}^{h,i})_{i=1}^N$ refer to the hidden states of the GRU. Note that for most of the variables involved, direct dependencies on the parameters are omitted for a less cumbersome notation. We remark that for iterations $t < 40$, the starting time of the vectors of particles must be considered one for x and zero for h.

Algorithm 4 OVSMC for deep generative model of moving agent of Section 5.3.

Input: $(\xi_{t-39:t}^{x,i}, \xi_{t-40:t-1}^{h,i}, \omega_t^i)_{i=1}^N, y_{t-39:t+1}, \theta_t, \lambda_t$
for $i \leftarrow 1, \dots, L$ **do**
  draw $I_{t+1}^i \sim \mathsf{cat}((\omega_t^\ell)_{\ell=1}^N)$;
  set $\xi_{t-39:t}^{h,i} \leftarrow \texttt{GRU}_{\lambda_t}(\texttt{initial\_state} = \xi_{t-40}^{h,I_{t+1}^i}, \varphi_{\theta_t}^x(\xi_{t-39:t}^{x,I_{t+1}^i}), \varphi_{\lambda_t}^y(y_{t-39:t}))$;
  draw $\varepsilon_{t+1}^i \sim N(0, I_{128})$;
  set $\xi_{t+1}^{x,i} \leftarrow \mu_{\lambda_t}^x(\varphi_{\lambda_t}^y(y_{t+1}), \xi_t^{h,i}) + \sigma_{\lambda_t}^x(\varphi_{\lambda_t}^y(y_{t+1}), \xi_t^{h,i})\varepsilon_{t+1}^i$;
  set $\omega_{t+1}^i(\lambda_t, \theta_t) \leftarrow \dfrac{m_{\theta_t}(\xi_{t+1}^{x,i} \mid \xi_t^{h,i}) g_{\theta_t}(y_{t+1} \mid \xi_t^{h,i}, \xi_{t+1}^{x,i})}{r_{\lambda_t}(\xi_{t+1}^{x,i} \mid y_{t+1}, \xi_t^{h,i})}$;
end for
set $\lambda_{t+1} \leftarrow \lambda_t + \gamma_{t+1}^\lambda \nabla_\lambda \log\left(\sum_{i=1}^N \omega_{t+1}^i(\lambda_t, \theta_t)\right)$;
for $i \leftarrow 1, \dots, N$ **do**
  draw $I_{t+1}^i \sim \mathsf{cat}((\omega_t^\ell)_{\ell=1}^N)$;
  set $\xi_{t-39:t}^{h,i} \leftarrow \texttt{GRU}_{\lambda_{t+1}}(\texttt{initial\_state} = \xi_{t-40}^{h,I_{t+1}^i}, \varphi_{\theta_t}^x(\xi_{t-39:t}^{x,I_{t+1}^i}), \varphi_{\lambda_{t+1}}^y(y_{t-39:t}))$;
  draw $\varepsilon_{t+1}^i \sim N(0, I_{128})$;
  set $\xi_{t+1}^{x,i} \leftarrow \mu_{\lambda_{t+1}}^x(\varphi_{\lambda_{t+1}}^y(y_{t+1}), \xi_t^{h,i}) + \sigma_{\lambda_{t+1}}^x(\varphi_{\lambda_{t+1}}^y(y_{t+1}), \xi_t^{h,i})\varepsilon_{t+1}^i$;
  set $\omega_{t+1}^i(\lambda_{t+1}, \theta_t) \leftarrow \dfrac{m_{\theta_t}(\xi_{t+1}^{x,i} \mid \xi_t^{h,i}) g_{\theta_t}(y_{t+1} \mid \xi_t^{h,i}, \xi_{t+1}^{x,i})}{r_{\lambda_{t+1}}(\xi_{t+1}^{x,i} \mid y_{t+1}, \xi_t^{h,i})}$;
  set $\xi_{t-38:t+1}^{x,i} \leftarrow (\xi_{t-38:t}^{x,I_{t+1}^i}, \xi_{t+1}^{x,i})$;
end for
set $\theta_{t+1} \leftarrow \theta_t + \gamma_{t+1}^\theta \nabla_\theta \log\left(\sum_{i=1}^N \omega_{t+1}^i(\lambda_{t+1}, \theta_t)\right)$;
return $(\xi_{t-38:t+1}^{x,i}, \xi_{t-39:t}^{h,i}, \omega_{t+1}^i)_{i=1}^N, \theta_{t+1}, \lambda_{t+1}$.