Monte Carlo Filtering Objectives

Shuangshuang Chen[1,2*], **Sihao Ding**[2], **Yiannis Karayiannidis**[3] and **Mårten Björkman**[1]
[1] Royal Institute of Technology, Stockholm, Sweden
[2] AI Lab, Volvo Car Corporation
[3] Chalmers University of Technology, Gothenburg, Sweden
{shuche, celle}@kth.se, sihao.ding@volvocars.com, yiannis@chalmers.se

Abstract

Learning generative models and inferring latent trajectories have shown to be challenging for time series due to the intractable marginal likelihoods of flexible generative models. It can be addressed by surrogate objectives for optimization. We propose Monte Carlo filtering objectives (MCFOs), a family of variational objectives for jointly learning parametric generative models and amortized adaptive importance proposals of time series. MCFOs extend the choices of likelihood estimators beyond Sequential Monte Carlo in state-of-the-art objectives, possess important properties revealing the factors for the tightness of objectives, and allow for less biased and variant gradient estimates. We demonstrate that the proposed MCFOs and gradient estimations lead to efficient and stable model learning, and learned generative models well explain data and importance proposals are more sample efficient on various kinds of time series data.

1 Introduction

Learning a generative model with latent variables for time series is of interest in many applications. However, exact inference and marginalization are often intractable for flexible generative models, making it challenging to learn. There are a few popular approaches to circumvent these difficulties: implicit methods that learn generative models by comparing generated samples to data distributions like Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014]; and explicit methods that define surrogate objectives of the intractable marginal log-likelihood like Variational Autoencoder (VAEs) [Kingma and Welling, 2014], or tractable marginals by invertible transformations like Normalizing Flows (NFs) [Rezende and Mohamed, 2015]. Explicit methods are often preferable when latent/encoded information is of importance, e.g. filtering and smoothing problems for some subsequent tasks. In this work, we mainly focus on the second approach and propose a family of surrogate filtering objectives to learn generative models and adaptive importance proposal models for time series.

*Contact Author

Researchers have introduced various surrogate objectives using variational approximations of intractable posterior for time series, known as evidence lower bounds (ELBOs), such as STONE [Bayer and Osendorfer, 2014], VRNN [Chung *et al.*, 2015], SRNN [Fraccaro *et al.*, 2016], DKF [Krishnan *et al.*, 2017], KVAE [Fraccaro *et al.*, 2017]. However, they typically suffer from a general issue caused by the limited flexibility of the variational approximations, thus restricting the learning of generative models. To alleviate this constraint, IWAE [Burda *et al.*, 2016] proposes a tighter objective by averaging importance weights of multiple samples drawn from a variational approximation. Monte Carlo objectives (MCOs) [Mnih and Rezende, 2016] generalizes the IWAE objective and ELBOs for non-sequential data. AESMC [Le *et al.*, 2018], FIVO [Maddison *et al.*, 2017], and VSMC [Naesseth *et al.*, 2018] extend this idea for sequential data using the estimators by Sequential Monte Carlo (SMC) and propose closely related surrogates objectives for learning.

Inspired by MCOs and the sequential variants, we propose Monte Carlo filtering objectives (MCFOs), a new family of surrogate objectives for generative models of time series, that

- broadens previously limited choices of estimators for time series other than SMC,
- possesses unique properties such as monotonic convergence and asymptotic bias, revealing the factors that determine a tighter objective: the number of samples and importance proposals,
- reduces high variance in gradient estimates of proposal models common in state-of-the-art algorithms, without introducing bias, which allows for faster convergence and sample efficient proposal models.

The paper is organized as follows: we first review the definition of MCOs and discuss common limitations of existing filtering variants. In Section 3, we derive MCFOs, explain their relations to other objectives and important properties. We demonstrate two instances of MCFOs with SMC and Particle Independence Metropolis-Hasting (PIMH) to learn models on 1) Linear Gaussian State Space Models (LGSSMs), 2) nonlinear, non-Gaussian, high dimensional SSMs of video sequences, 3) non-Markovian music sequences[1].

[1] See [Chen *et al.*, 2021] for a complete version of this manuscript including appendices.

2 Background

2.1 Monte Carlo Objectives

For a generative model with observation $\mathbf{x}$ and latent state $\mathbf{z}$, a Monte Carlo objective (MCO) [Mnih and Rezende, 2016] is defined as an estimate of the marginal log-likelihood $\log p(\mathbf{x})$ by samples drawn from a proposal distribution q:

$$\mathbb{E}_{q(\mathbf{z})}[\log R] = \log p(\mathbf{x}) - \mathbb{E}_{q(\mathbf{z})}[\log \frac{p(\mathbf{x})}{R}] \leq \log p(\mathbf{x}), \quad (1)$$

where R is any unbiased estimator of $p(\mathbf{x})$ that $\mathbb{E}[R] = p(\mathbf{x})$. It is also a lower bound of $\log p(\mathbf{x})$ as can be shown using Jensen's inequality. When an estimator takes a single sample from q and $R = p(\mathbf{x}, \mathbf{z})/q(\mathbf{z})$, the MCO can be identified as ELBO in variational inference [Mnih and Rezende, 2016]. When an estimator averages importance weights from K samples, $R^K = K^{-1}\sum_{i=1}^K p(\mathbf{z}^i, \mathbf{x})/q(\mathbf{z}^i)$, it yields an importance weighted ELBO (IW-ELBO) [Burda *et al.*, 2016; Domke and Sheldon, 2018]. This bound is proven to be tighter with increasing K and asymptotically converges to $\log p(\mathbf{x})$, as $K \to \infty$.

2.2 Sequential Monte Carlo

For a sequential observation $\mathbf{x}_{1:T}$ with latent trajectory $\mathbf{z}_{1:T}$, the generative process can be factorized as $p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)\prod_{t=2}^T p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{x}_{1:t-1})$. Inferring latent trajectory, $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$, is of importance for marginalization and to learn generative models, however, usually intractable. Sequential Monte Carlo (SMC) approximates a target distribution, specifically $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$, using a set of weighted sample trajectories $\{\tilde{w}_T^i, \hat{\mathbf{z}}_{1:T}^i\}_{i=1:K}$. It combines Sequential Importance Sampling (SIS) with resampling, consisting of four main steps:

Sample K particles $\hat{\mathbf{z}}_t^i$ from proposal $q(\mathbf{z}_t|\bar{\mathbf{z}}_{1:t-1}^i, \mathbf{x}_{1:t})$ with previously resampled trajectories $\bar{\mathbf{z}}_{1:t-1}^i$;

Append to trajectory $\hat{\mathbf{z}}_{1:t}^i = (\hat{\mathbf{z}}_t^i, \bar{\mathbf{z}}_{1:t-1}^i)$;

Weight trajectories with $\tilde{w}_t^i = w_t^i / \sum_{j=1}^K w_t^j$, where $w_t^i = p(\mathbf{x}_{1:t}, \hat{\mathbf{z}}_{1:t}^i|\mathbf{x}_{1:t-1}, \bar{\mathbf{z}}_{1:t-1}^i)/q(\hat{\mathbf{z}}_t^i|\bar{\mathbf{z}}_{1:t-1}^i, \mathbf{x}_{1:t})$;

Resample from $\{\tilde{w}_t^i, \hat{\mathbf{z}}_{1:t}^i\}$ to obtain equally-weighted particles $\bar{\mathbf{z}}_{1:t}^i = \hat{\mathbf{z}}_{1:t}^{A_{t-1}^i}$, with ancestral indices A_{t-1}^i.

This iteration continues until time T. Besides being an approximate inference, SMC also gives an unbiased estimate of the marginal likelihood $p(\mathbf{x}_{1:T})$ by the importance weights:

$$\hat{p}(\mathbf{x}_{1:T}) = \prod_{t=1}^T \left(\frac{1}{K} \sum_{i=1}^K w_t^i \right). \quad (2)$$

The variance of this estimate, accessed by the so-called Effective Sample Size (ESS) for sample efficiency, is largely dependent on the proposal distributions q. We refer to [Doucet and Johansen, 2009] for a more in-depth discussion.

2.3 Variational Filtering Objectives

To learn a generative model for time series data, various ELBO-like surrogate objectives have been proposed using different factorizations of generative models and approximations $q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$. IW-ELBO can be extended to sequences by changing from Importance Sampling (IS) to SIS estimator $R^K = K^{-1}\sum_{i=1}^K p(\mathbf{z}_{1:T}^i, \mathbf{x}_{1:T})/q(\mathbf{z}_{1:T}^i)$. However, such an estimator suffers from exponential growth of variance with the length of sequences.To improve this, AESMC, FIVO and VSMC propose three closely related MCOs, exploiting the SMC estimators (2):

$$\begin{aligned} \text{ELBO}_{\text{SMC}} &= \mathbb{E}_{Q_{\text{SMC}}}\Big[\sum_{t=1}^T \log \underbrace{\left(\frac{1}{K} \sum_{i=1}^K w_t^i \right)}_{R_t^K} \Big] \\ Q_{\text{SMC}}(z_{1:T}^{1:K}) &= \int \left(\prod_{i=1}^K q(z_1^i) \right) \\ &\quad \prod_{t=2}^T \prod_{i=1}^K \left(q(z_t^i|z_{1:t-1}^{A_{t-1}^i}) \cdot \frac{w_{t-1}^{A_{t-1}^i}}{\sum_j w_{t-1}^j} \right) dA_{1:T-1}^{1:K}. \end{aligned} \quad (3)$$

It is found that the learning of generative models via the objective suffers high variance in gradient estimation, since importance weight w_t^i does not allow for smooth gradient computation. We show that simply ignoring some high variance term in the gradient estimate as the suggested solution in previous methods, introduces an extra bias and leads to non-optimality of proposal and generative parameters at convergence. To tackle these problems and extend variational filtering objectives, we propose MCFOs, and discuss their important properties in the following sections.

3 Monte Carlo Filtering Objectives

Instead of constructing variational lower bound from (2), we leverage the decomposition of joint marginal log-likelihood:

$$\log p(\mathbf{x}_{1:T}) = \log p(\mathbf{x}_1) + \sum_{t=2}^T \log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}). \quad (4)$$

Nonetheless, $\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ is usually intractable, which makes learning a generative model by maximizing (4) only possible in some limited cases. Instead, we define $\mathcal{L}_t^K$, an MCO for each $\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ with K samples:

$$\begin{aligned} &\mathcal{L}_t^K = \mathbb{E}_{Q_t^K}[\log R_t^K], \\ &Q_t^K(\mathbf{z}_{1:t}^{1:K}|\mathbf{x}_{1:t}) = p(\mathbf{z}_{1:t-1}^{1:K}|\mathbf{x}_{1:t-1}) \cdot \prod_{i=1}^K q(\mathbf{z}_t^i|\mathbf{z}_{1:t-1}^i, \mathbf{x}_{1:t}), \\ &R_t^K = \frac{1}{K} \sum_{i=1}^K \frac{p(\mathbf{z}_{1:t}^i, \mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}^i, \mathbf{x}_{1:t-1}) q(\mathbf{z}_t^i|\mathbf{z}_{1:t-1}^i, \mathbf{x}_{1:t})}, \end{aligned}$$

specifically $\mathcal{L}_1^K$ for $\log p(\mathbf{x}_1)$,

$$Q_1^K(\mathbf{z}_1^{1:K}|\mathbf{x}_1) = \prod_{i=1}^K q(\mathbf{z}_1^i|\mathbf{x}_1), \quad R_1^K = \frac{1}{K} \sum_{i=1}^K \frac{p(\mathbf{z}_1^i, \mathbf{x}_1)}{q(\mathbf{z}_1^i|\mathbf{x}_1)}, \quad (5)$$

where $q(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$ and $q(\mathbf{z}_1|\mathbf{x}_1)$ are the proposal distributions, R_t^K and R_1^K are the unbiased estimators of $p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ and $p(\mathbf{x}_1)$ with K samples; see [Chen *et al.*, 2021, Appendix B.1] for derivations. Summing up the series

of MCOs , a Monte Carlo filtering objective (MCFO), $\mathcal{L}_{\text{MCFO}}$:

$$\mathcal{L}^K_{\text{MCFO}}(\mathbf{x}_{1:T}, p, q) = \sum_{t=1}^{T} \mathcal{L}^K_t \leq \log p(\mathbf{x}_{1:T}),$$

is defined as the lower bound of $\log p(\mathbf{x}_{1:T})$. To avoid notation clutter, we leave out arguments when the context permits.

Considering the filtering problem for which future observations have no impact on the current posterior, replacing $p(\mathbf{z}^{1:K}_{1:t-1}|\mathbf{x}_{1:t-1})$ in Q^K_t with K sample approximations $\hat{p}(\mathbf{z}^{1:K}_{1:t-1}|\mathbf{x}_{1:t-1}) = \sum_{i=1}^{K} \tilde{w}^i_{t-1}\delta(\mathbf{z}^i_{1:t-1} - \hat{\mathbf{z}}^i_{1:t-1})$ retrieves the definition of ELBO_{SMC} in (3). The objective can be considered as an estimate of MCFOs by SMC and is consistent to MCFOs with the asymptotic bias of $\mathcal{O}(1/K)$; see [Chen *et al.*, 2021, Appendix B.2] for details. MCFOs can freely choose other estimator alternatives such as PIMH [Andrieu *et al.*, 2010] and unbiased MCMC with couplings [Jacob *et al.*, 2017] to further improve sampling efficiency of SMC.

3.1 Properties of MCFOs

Except for the general properties inherited from MCOs such as *bound* and *consistency*, the convergence of MCFOs is *monotonic* like IW-ELBO, but unique to earlier filtering objectives. Additionally, the asymptotic bias of MCFOs can be shown to relate to the total variances of estimators.

Proposition 1. (Properties of MCFOs). *Let $\mathcal{L}^K_{MCFO}$ be an MCFO of $\log p(\mathbf{x}_{1:T})$ by a series of unbiased estimators R_t of $p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ using K samples. Then,*

a) (Bound) $\log p(\mathbf{x}_{1:T}) \geq \mathcal{L}^K_{MCFO}$.

b) (Monotonic convergence) $\mathcal{L}^{K+1}_{MCFO} \geq \mathcal{L}^K_{MCFO} \geq \ldots \geq \mathcal{L}^1_{MCFO}$.

c) (Consistency) *If $p(\mathbf{z}_1, \mathbf{x}_1)/q(\mathbf{z}_1|\mathbf{x}_1)$ and $p(\mathbf{z}_{1:t}, \mathbf{x}_{1:t})/(p(\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})q(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}))$ for all $t \in [2, T]$ are bounded, then $\mathcal{L}^K_{MCFO} \to \log p(\mathbf{x}_{1:T})$ as $K \to \infty$.*

d) (Asymptotic Bias) *For a large K, the bias of bound is related to the variance of estimator R_t, $\mathbb{V}[R_t]$,*

$$\lim_{K\to\infty} K(\log p(\mathbf{x}_{1:T}) - \mathcal{L}^K_{MCFO}) = \sum_{t=1}^{T} \frac{\mathbb{V}[R_t]}{2p(\mathbf{x}_t|\mathbf{x}_{1:t-1})^2}.$$

Proof. See [Chen *et al.*, 2021, Appendix B.3].

Although increasing the number of samples K leads to a tighter MCFO, a large K is infeasible in terms of computational and memory. It has also been shown that larger K may deteriorate to learn proposals [Rainforth *et al.*, 2018]. An appropriate K is a critical hyperparameters that affects learning both generative and proposal models. On the other hand, *asymptotic bias* suggests another way for a tighter bound, i.e. using less variant estimators R_t, which has been overlooked in recent literature. It explains why the bounds defined by SMC are tighter than IW-ELBO by SIS. Thus, the proposal model that permits less variant R_t, either designed or learned, is another key instrument.

3.2 Optimal Importance Proposals

Considering proposals q as an argument of MCFOs, we can derive the optimal proposals when they maximize the bound.

Proposition 2. (Optimal importance proposals q^* for an MCFO). *The bound is maximized and exact to $\log p(\mathbf{x}_{1:T})$ when the importance proposals are*

$q^*(\mathbf{z}_1|\mathbf{x}_1) = p(\mathbf{z}_1|\mathbf{x}_1)$,

for all $t = 2 : T$

$$q^*(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}|\mathbf{x}_{1:t-1})} = p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}).$$

Proof. See [Chen *et al.*, 2021, Appendix B.4].

The optimal importance proposals always propagate samples from the previous filtering posterior $p(\mathbf{z}_{1:t-1}|\mathbf{x}_{1:t-1})$ to the new target $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$, thus leads MCFOs to be exact. For SSMs that assume Markovian latent variables and conditional independent observations, the optimal importance proposals are further simplified to $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t)$ [Doucet *et al.*, 2000, Proposition 2]. For common intractable problems, though the filtering posteriors and optimal proposals are not accessible, we can learn a parametric adaptive importance proposal model jointly with generative models by optimizing MCFOs.

3.3 Learning Generative and Proposal Models

To illustrate the learning of a flexible importance proposal and/or a generative model, we explicitly parameterize generative model $p_{\boldsymbol{\theta}}$ and proposal model $q_{\boldsymbol{\phi}}$ by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ respectively, and optimize them by gradient-based algorithms.

Earlier methods suffer from high variance in gradient estimate due to the second term in (6) in Table 1. This is mainly caused by 1) large magnitudes of $\log \hat{p}(\mathbf{x}_{1:T})$, especially at the beginning of training; and 2) high variance in the gradient for non-smooth categorical distribution of discrete ancestral indices A^i_{t-1} in SMC. FIVO and VSMC propose to ignore the high variance term to stabilize and accelerate convergence. However, it comes at the cost of an induced bias that cannot be eliminated by increasing the number of samplesand deteriorates the convergence to optimum [Roeder *et al.*, 2017].

MCFOs can circumvent the issue for $\nabla_{\boldsymbol{\phi}}\mathcal{L}^K_{\text{MCFO}}$ by *reparameterization trick* [Kingma and Welling, 2014] without an extra bias. Assuming the proposal distribution $q_{\boldsymbol{\phi}}$ is reparameterizable, $\nabla_{\boldsymbol{\phi}}\mathcal{L}^K_t$ is estimated less variantly by:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathcal{L}^K_t &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z}^i_{1:t-1}|\mathbf{x}_{1:t-1})}\Big[\nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}^i_t|\mathbf{z}^i_{1:t-1}, \mathbf{x}_{1:t})} \\ &\qquad \log \frac{1}{K}\sum_{i=1}^{K} \frac{p_{\boldsymbol{\theta}}(\mathbf{z}^i_{1:t}, \mathbf{x}_{1:t})}{p_{\boldsymbol{\theta}}(\mathbf{z}^i_{1:t-1}, \mathbf{x}_{1:t-1})q_{\boldsymbol{\phi}}(\mathbf{z}^i_t|\mathbf{z}^i_{1:t-1}, \mathbf{x}_{1:t})}\Big] \\ &\simeq \nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_{1:t}, \hat{\mathbf{z}}^{1:K}_{1:t-1}, \\ &\qquad \underbrace{g_{\boldsymbol{\phi}}(\hat{\mathbf{z}}^1_{1:t-1}, \boldsymbol{\epsilon}^i, \mathbf{x}_{1:t}), \ldots, g_{\boldsymbol{\phi}}(\hat{\mathbf{z}}^K_{1:t-1}, \boldsymbol{\epsilon}^K, \mathbf{x}_{1:t}))}_{\text{reparameterization trick, } \hat{\mathbf{z}}^i_t = g_{\boldsymbol{\phi}}(\hat{\mathbf{z}}^i_{1:t-1}, \boldsymbol{\epsilon}^i, \mathbf{x}_{1:t})},\end{aligned} \tag{7}$$

where $\{\hat{\mathbf{z}}^i_{1:t}\}_{i=1:K}$ are K sample trajectories, e.g. from SMC, and can be specified with ancestral indices A^i_{t-1} when resampling applies, $f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\cdot)$ is the logarithm average function $\log \frac{1}{K}\sum_{i=1}^{K}\left(\frac{\cdot}{\cdot}\right)$, and $\boldsymbol{\epsilon}^i$ is a sample from a base distribution $p(\boldsymbol{\epsilon})$. The same trick cannot directly apply to $\nabla_{\boldsymbol{\theta}}\mathcal{L}^K_t$, because of the existence of $\boldsymbol{\theta}$ in the expectation. Instead, we use the

Method	$\nabla_{\boldsymbol{\phi}}$	$\nabla_{\boldsymbol{\theta}}$
MCFO	See (7)	See (8)
AESMC/FIVO/VSMC	$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \log \hat{p}(\mathbf{x}_{1:T}) + \sum_{t=2}^{T} \log \frac{\hat{p}(\mathbf{x}_{1:t})}{\hat{p}(\mathbf{x}_{1:t-1})} \left(\sum_{i=1}^{K} \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \log(w_{t-1}^{A_{t-1}^i} / \sum_j w_{t-1}^j) \right)$ (6)	
IWAE	$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \tilde{g}_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \boldsymbol{\epsilon}_{1:T}^1), \ldots, \tilde{g}_{\boldsymbol{\phi}}(\mathbf{x}_{1:T}, \boldsymbol{\epsilon}_{1:T}^K))$, where $\tilde{g}$ is reparameterized function of $q_{\boldsymbol{\phi}}(\mathbf{z}_{1:T}\|\mathbf{x}_{1:T})$	
NASMC	$\sum_{t=1}^{T} \sum_{i=1}^{K} \tilde{w}_t^i \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}_t^i \| \mathbf{x}_{1:t}, \hat{\mathbf{z}}_{1:t-1}^{A_{t-1}^i})$	$\sum_{t=1}^{T} \sum_{i=1}^{K} \tilde{w}_t^i \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}_t^i, \mathbf{x}_t \| \mathbf{x}_{1:t-1}, \hat{\mathbf{z}}_{1:t-1}^{A_{t-1}^i})$

Table 1: Comparison of gradient estimates by MCFO, AESMC/FIVO/VSMC, IWAE, NASMC.

score function of $p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ to estimate $\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^K$:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^K \simeq \sum_{i=1}^{K} \tilde{w}_t^i \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t, \hat{\mathbf{z}}_t^i | \mathbf{x}_{1:t-1}, \hat{\mathbf{z}}_{1:t-1}^i), \quad (8)$$

where $\tilde{w}_t^i$ are the normalized importance weights of $\hat{\mathbf{z}}_{1:t}^i$; see [Chen *et al.*, 2021, Appendix B.5, B.6] for detailed derivation of (7) and (8). Essentially, the score function estimate is equivalent as dropping the high variance term in (6) for $\nabla_{\boldsymbol{\theta}}$.

NASMC [Gu *et al.*, 2015] is closely related to MCFOs with SMC implementation in terms of $\nabla_{\boldsymbol{\theta}}$, and RWS [Bornschein and Bengio, 2015] is a special case of NASMC that replaces SMC by SIS. These two methods, however, construct a different surrogate objectives for optimizing $\boldsymbol{\phi}$. While NASMC and RWS minimize the approximated inclusive KL-divergence, $\mathrm{KL}(\hat{p}_{\boldsymbol{\theta}}(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) || q_{\boldsymbol{\phi}}(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}))$, MCFOs minimize the disclusive KL-divergence, $\mathrm{KL}(Q_t^K(\mathbf{z}_{1:t}^{1:K}|\mathbf{x}_{1:t}) || p_{\boldsymbol{\theta}}(\mathbf{z}_{1:t}^{1:K}|\mathbf{x}_{1:t}))$, on the extended latent space as the dual problem of maximizing surrogate objectives. When the family of proposals is adequately flexible to include simple true posteriors, both NASMC and MCFOs converge to the same optimum. To fit a potentially complex multi-modal posterior, the simple proposal learned by NASMC and RWS tends to have undesired low density everywhere in order to cover all modalities, thus impairs the sample efficiency of estimators and restricts the learning of generative models. For MCFOs, $Q_t^K(\mathbf{z}_{1:t}^{1:K}|\mathbf{x}_{1:t})$ is naturally a mixture of simple proposal distributions q_ϕ with importance weights $\tilde{w}_{t-1}^i$, which remains flexible to fit multi-modal posteriors, while sustaining sample efficiency.

Furthermore, when the latent and observation variables are assumed to be finite-order Markovian, both gradients of MCFOs can be updated incrementally which makes them well suited for arbitrarily long sequences and data streams.

4 Experiments

We seek to evaluate our method in experiments by answering: 1) what is the side-effect of ignoring the high variance term as proposed in earlier methods; 2) do the gradient estimates of MCFOs reduce the variance without the cost of additional bias; 3) how does the number of samples affect the learning of generative models and how sample efficient are the learned proposal models? We evaluate two instances of MCFOs, MCFO-SMC and MCFO-PIMH, using SMC and PIMH respectively to learn generative and proposal models on LGSSM, non-Gaussian, nonlinear, high dimensional SSMs of video sequences, and non-Markovian polyphonic music sequences[2]. We restrict the form of posteriors to $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t)$ for SSM cases to encode history into low dimensional representations, while using VRNN to accommodate long temporal dependencies for non-Markovian data. To be noted, all models are amortized over all time instances.

4.1 Gradient Estimation

Following [Rainforth *et al.*, 2018; Le *et al.*, 2018], we carry out experiments to examine gradient estimators on a tractable LGSSM, defined by θ_1 and θ_2, and importance proposal $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t)$, parameterized by $\boldsymbol{\phi}$:

$$\begin{aligned}
&p(z_1) = \mathcal{N}(z_1; \mu_0, \sigma_0^2), p_{\boldsymbol{\theta}}(z_t|z_{t-1}) = \mathcal{N}(z_t; \theta_1 z_{t-1}, \Sigma_Q), \\
&p_{\boldsymbol{\theta}}(x_t|z_t) = \mathcal{N}(x_t; \theta_2 z_t, \Sigma_R), \\
&q_{\boldsymbol{\phi}}(z_1|x_1) = \mathcal{N}(z_1; \phi_1 x_1 + \phi_2, \Sigma_{q,1}), \\
&q_{\boldsymbol{\phi}}(z_t|z_{t-1}, x_t) = \mathcal{N}(z_t; \phi_3 z_{t-1} + \phi_4 x_t + \phi_5, \Sigma_{q,t}).
\end{aligned} \quad (9)$$

The gradient estimates are computed by backwards automatic differentiationon the objectives defined by IWAE, AESMC and MCFO-SMC w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using sequences generated by the LGSSM. AESMC is implemented ignoring the high variance term in (6) as suggested. Figure 1 shows 1000 gradient samples by all three methods under different numbers of samples, K, when both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are at optima.

For $\nabla\boldsymbol{\phi}$, the induced bias of AESMC is distinct and does not disappear with increasing K, which makes parameters unable to converge to the exact optimum. Although increasing K decreases the variance for all methods, it is specially detrimental to AESMC for which gradient estimates are barely close to true gradients [Rainforth *et al.*, 2018], but beneficial to IWAE and MCFO. For $\nabla\boldsymbol{\theta}$, MCFO and AESMC have similar estimates close to the analytical gradients, while IWAE estimates are substantially deviated due to the high variance of SIS estimators. Training by MCFOs is expected to have a similar performance but with less computations, compared to the alternating strategy to optimize IWAE objective for $\boldsymbol{\phi}$ and AESMC for $\boldsymbol{\theta}$ [Le *et al.*, 2018]. See [Chen *et al.*, 2021, Appendix D.1] for gradient estimates at other locations.

4.2 Learning and Inference on LGSSMs

To examine the learning of generative and proposal parameters, we generate 5000 trajectories by LGSSM in (9) with $\theta_1 = 0.9$, $\theta_2 = 1.2$, of which 4000 are for training and rest for testing. Figure 2 illustrates 5 benchmarking methods including bootstrap filtering [Särkkä, 2013], IWAE, AESMC,

[2]The implementation of our algorithms and experiments are available at https://github.com/ssajj1212/MCFO.

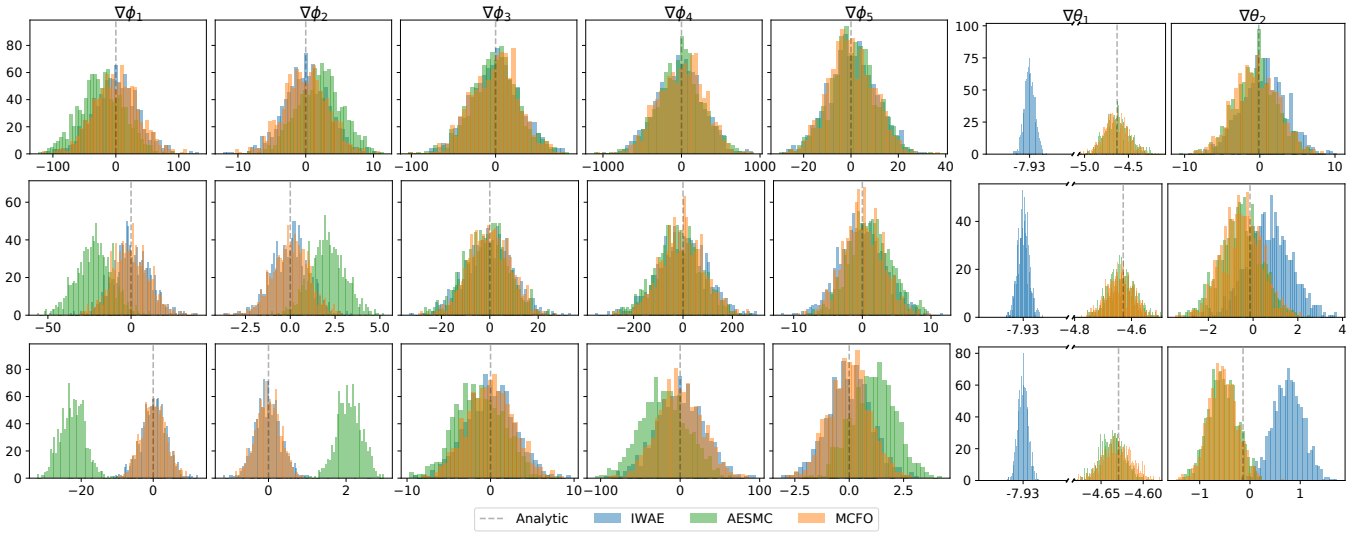

Figure 1: Gradient estimates of IWAE, AESMC, MCFO with respect to generative and proposal parameters at their optima with different numbers of samples K; *Top:* $K = 10$, *Middle:* $K = 100$, *Bottom:* $K = 1000$.

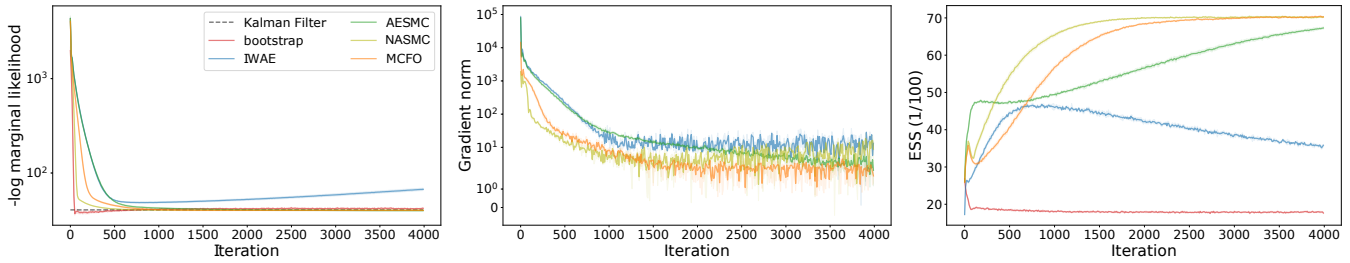

Figure 2: *Left*: Negative marginal log-likelihoods (NLLs) on test data. *Middle*: Gradient norms of parameters. *Right*: Effective sample size (ESS). Lines indicate the average of 3 random seed trainings and shaded areas for standard deviation.

NASMC and MCFO-SMC, using the same initialization and optimizer; see [Chen *et al.*, 2021, Appendix D.2] for experiment setups. Note that bootstrap uses prior as proposal, thus no proposal parameter needs to learn. To evaluate the performance of learned proposal models for sample efficiency and tightness of lower bound, we report the variance of estimators by ESS $= (\sum_i (\tilde{w}_t^i)^2)^{-1}$, and average over test sequences.

MCFO-SMC and NASMC learn more sample efficient proposal models than bootstrap, AESMC and IWAE, and converge to the exact analytic optimum. Although AESMC does not differ significantly in terms of NLLs from MCFO and NASMC, the bias in gradient estimates shown previously, causes it slow to converge and cannot converge to the exact optimum. MCFO learns both generative and proposal models faster than AESMC. For this simple case, NASMC converges faster than MCFOs, since fitting a Gaussian proposal with NASMC to the uni-modal Gaussian posterior of LGSSM is easier than fitting a mixture of Gaussian proposals with MCFO. However, NASMC may fail to learn multi-modal posteriors for general intractable problems, as shown in the next section. Furthermore, increasing the number of samples and replacing SMC by PIMH with different number of sweeps only slightly improve the learning by MCFOs, see more results in [Chen *et al.*, 2021, Appendix D.2].

4.3 Video Sequences

To assess MCFOs in more general cases, we simulate 1000 video sequences of a single pendulum system in `gym`, out of which 500 are used for testing. Each sequence contains 20 32×32 pixel grayscale images representing factorized Bernoulli distributions of high-dimensional observations; see examples in Figure 3. The transition and proposal distributions, $p_{\boldsymbol{\theta}}(\mathbf{z}_t|\mathbf{z}_{t-1})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{t-1})$, are parameteric Gaussian MLPs, while observation models, $p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{z}_t)$, are parameteric Bernoulli MLPs. The latent dimension is set to 3, and optimizers and model definitions are the same for all methods; see [Chen *et al.*, 2021, Appendix E].

Figure 3 shows the commonly used one-step prediction errors in observations and ESSs on the test set, evaluated by SMC with 1000 particles on the models trained by AESMC, MCFO-SMC and MCFO-PIMH with $K = 10, 20, 50, 100$. Additionally, Table 2 reports both metrics averaged over the last 1000 iterations of trainings. Note that *NASMC fails to converge in this task regardless of* K. Compared to AESMC, both MCFO-SMC and MCFO-PIMH, show quicker convergences, lower prediction errors and higher ESSs that indicate more sample efficient proposal models, especially at smaller

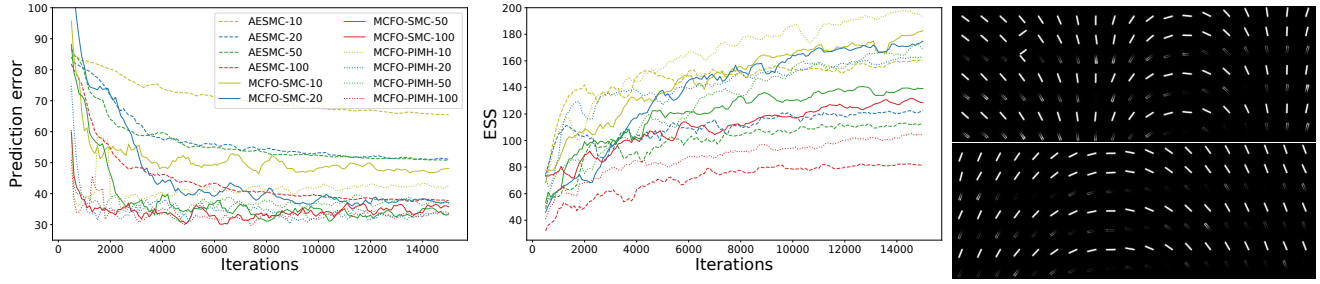

Figure 3: *Left, Middle*: One-step prediction errors and ESS on the test sets of generative and proposal models learned by AESMC, MCFO-SMC, MCFO-PIMH with $K = 10, 20, 50, 100$, evaluated by SMC with 1000 samples and moving average over 3 evaluation runs. *Right:* Two sequences with one-step predictions by AESMC, MCFO-SMC and MCFO-PIMH with $K = 100$. Each row is Bernoulli mean of observations, the one-step predictions and the absolute differences between predictions and observations by AESMC, MCFO-SMC and MCFO-PIMH.

	K	AESMC	MCFO-SMC	MCFO-PIMH	K	AESMC	MCFO-SMC	MCFO-PIMH
Prediction	10	65.53 ± 0.18	47.54 ± 0.94	$\mathbf{42.14 \pm 1.49}$	50	50.84 ± 0.16	$\mathbf{34.06 \pm 1.46}$	37.01 ± 1.47
ESS		160.04 ± 1.97	180.97 ± 3.16	$\mathbf{195.62 \pm 3.05}$		112.53 ± 2.42	139.24 ± 2.46	$\mathbf{168.78 \pm 4.51}$
Prediction	20	51.13 ± 0.37	37.18 ± 0.72	$\mathbf{33.82 \pm 2.06}$	100	37.87 ± 0.21	36.17 ± 1.82	$\mathbf{33.71 \pm 0.98}$
ESS		122.51 ± 1.89	$\mathbf{172.99 \pm 3.46}$	163.01 ± 2.10		81.93 ± 0.82	$\mathbf{130.09 \pm 2.12}$	104.53 ± 1.94

Table 2: One-step prediction errors and ESS on the test set of generative and proposal models learned by AESMC, MCFO-SMC, MCFO-PIMH with $K = 10, 20, 50, 100$, evaluated by SMC with 1000 samples averaged over last 1000 iterations.

Methods	Nottingham	JSB chorales	MuseData	Piano-midi.de
MCFO-SMC-10	2.23 ± 0.16	3.87 ± 0.09	3.79 ± 0.10	6.24 ± 0.14
MCFO-SMC-20	2.14 ± 0.13	3.69 ± 0.12	3.65 ± 0.11	6.11 ± 0.15
MCFO-PIMH-10	2.12 ± 0.10	3.63 ± 0.07	3.59 ± 0.08	6.08 ± 0.09
MCFO-PIMH-20	$\mathbf{2.06 \pm 0.08}$	$\mathbf{3.54 \pm 0.08}$	$\mathbf{3.48 \pm 0.10}$	$\mathbf{6.03 \pm 0.12}$
FIVO	2.58 † (2.60 ± 0.18)	4.08 † (3.90 ± 0.14)	5.80 † (5.85 ± 0.15)	6.41 † (6.37 ± 0.19)
IWAE	2.52 † (2.50 ± 0.25)	5.77 † (5.43 ± 0.20)	6.54 † (6.28 ± 0.23)	6.74 † (6.54 ± 0.21)
NASMC [Gu *et al.*, 2015] †	2.72	3.99	6.89	7.61
SRNN [Fraccaro *et al.*, 2016] †	2.94	4.74	6.28	8.20
STONE [Bayer and Osendorfer, 2014] †	2.85	6.91	6.16	7.13

Table 3: Estimated NLL per time on polyphonic test sets by SMC with 500 particles. MCFOs, FIVO and IWAE are evaluated by 10 runs, and both FIVO and IWAE, trained the same as MCFO-SMC-10, are reported in parenthesis. † is originally reported.

K. Furthermore, MCFOs implicitly regularize to learn simpler generative models, see [Chen *et al.*, 2021, Appendix E]. Although MCFO-PIMH converges faster than MCFO-SMC and AESMC because of better Monte Carlo approximations, the improvement at convergence is marginally small considering that it requires more computations for each sweep in PIMH. Increasing K does improve generative model learning, but slightly impairs the sample efficiency of proposal models. No statistically significant gain is observed to increase K over 200. Therefore, the sweet spot of K needs to balance the performance of generation and inference.

4.4 Polyphonic Music

To demonstrate the performance of MCFOs for non-Markovian high dimensional data with complex temporal dependencies, we train VRNN models with MCFO-SMC and MCFO-PIMH on four polyphonic music datasets [Boulanger *et al.*, 2012]. We preprocess all musical notes to 88-dimensional binary sequences and configure generative and proposal models as [Maddison *et al.*, 2017]; see [Chen *et al.*, 2021, Appendix F] for experiment details. Table 3 reports the estimated NLLs using 500 samples, as with the other benchmarked methods, on the models trained by MCFO-SMC and MCFO-PIMH with 10, 20 samples. As can be seen for all four datasets, MCFO-SMC and MCFO-PIMH are either superior or comparable to the other state-of-the-art algorithms.

5 Conclusion

We introduce Monte Carlo filtering objectives (MCFOs), a new family of variational filtering objectives to learn generative and importance proposal models for time series. MCFOs extend the choices of estimators to use and accommodate some theoretical properties for tighter objectives. We show empirically that MCFOs and the proposed gradient estimators facilitate to learn parametric generative and proposal models more stably and efficiently, compared to state-of-the-art methods in various tasks. In future works, we would like to explore the equivalence of MCFOs for smoothing problems and explore tractable MCFOs by flow-based methods.

Acknowledgements

This work is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

References

[Andrieu *et al.*, 2010] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[Bayer and Osendorfer, 2014] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. In *NIPS 2014 Workshop on Advances in Variational Inference*, 2014.

[Bornschein and Bengio, 2015] Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[Boulanger *et al.*, 2012] Lewandowski Nicolas Boulanger, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, pages 1881–1888, 2012.

[Burda *et al.*, 2016] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[Chen *et al.*, 2021] Shuangshuang Chen, Sihao Ding, Yiannis Karayiannidis, and Mårten Björkman. Monte carlo filtering objectives: A new family of variational objectives to learn generative model and neural adaptive proposal for time series. *arXiv preprint arXiv:2105.09801*, 2021.

[Chung *et al.*, 2015] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[Domke and Sheldon, 2018] Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. In *Advances in neural information processing systems*, pages 4470–4479, 2018.

[Doucet and Johansen, 2009] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[Doucet *et al.*, 2000] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[Fraccaro *et al.*, 2016] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016.

[Fraccaro *et al.*, 2017] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, pages 3601–3610, 2017.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[Gu *et al.*, 2015] Shixiang Shane Gu, Zoubin Ghahramani, and Richard E Turner. Neural adaptive sequential monte carlo. In *Advances in neural information processing systems*, pages 2629–2637, 2015.

[Jacob *et al.*, 2017] Pierre E Jacob, John O'Leary, and Yves F Atchadé. Unbiased markov chain monte carlo with couplings. *arXiv preprint arXiv:1708.03625*, 2017.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[Krishnan *et al.*, 2017] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *31st AAAI Conference on Artificial Intelligence*, 2017.

[Le *et al.*, 2018] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *6th International Conference on Learning Representations, ICLR*, 2018.

[Maddison *et al.*, 2017] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583, 2017.

[Mnih and Rezende, 2016] Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196, 2016.

[Naesseth *et al.*, 2018] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977, 2018.

[Rainforth *et al.*, 2018] Tom Rainforth, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 4274–4282, 2018.

[Rezende and Mohamed, 2015] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[Roeder *et al.*, 2017] Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, pages 6925–6934, 2017.

[Särkkä, 2013] Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.