SIχO: Smoothing Inference with Twisted Objectives

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sequential Monte Carlo (SMC) is an algorithm for approximate posterior inference in probabilistic state space models. Its efficacy is largely determined by two design choices: the proposal distribution and the sequence of target distributions. Recent work showed that the model and proposal distribution can be learned with variational techniques, maximizing a lower bound on the marginal likelihood. However, these methods are predicated on targeting the sequence of filtering distributions, conditioned only on the previous and current observations. We introduce *SIXO*, a variational method that learns a sequence of target distributions that approximate the smoothing distributions, incorporating information from all observations, jointly with the model and proposal . The key idea is to learn a backwards message that warps the filtering distributions into the smoothing distributions. We develop an efficient approach to learn the required backward message using density ratio estimation. We interleave this update with conventional updates for learning the model and proposal distribution. SIXO has both theoretical and practical advantages. It leads to provably tighter lower bounds and offers more accurate posterior inferences and parameter estimates in a variety of domains.

1 Introduction

In this work we consider model learning and approximate posterior inference in probabilistic state space models. Sequential Monte Carlo (SMC) is a general-purpose method for addressing these problems [1, 2]. It produces samples of latent state trajectories (i.e. *particles*) that can be used to approximate posterior expectations, as well as an unbiased estimate of the marginal likelihood. SMC can facilitate model learning via expectation-maximization or direct maximization of the marginal likelihood estimate [3, 4]. It can also be cast in a variational framework as a rich family of approximate posterior distributions that can be optimized with modern automatic differentiation methods [5–7].

The quality of SMC's marginal likelihood and posterior estimates is driven by two factors: the choice of *proposal distributions* and the choice of *target distributions*. The proposal distribution specifies how particles are propagated from one time step to the next; the target specifies how those particles are weighted and which ones survive to future time steps. Past work linking variational inference and SMC developed a unified framework for learning model parameters and proposal distributions [5–7], but fixed the targets to the *filtering distributions*—the conditional distribution over latent states $\mathbf{x}_{1:t} = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ given observations $\mathbf{y}_{1:t} = (\mathbf{y}_1, \ldots \mathbf{y}_t)$—ignoring any future observations.

Figure 1 illustrates why setting the target distributions to the filtering distributions can be especially problematic. In this example, the latent states follow a simple random walk with Gaussian noise, but the only observation comes at the last time step. Thus, the filtering distributions reduce to the prior, a mean zero Gaussian as shown in Figure 1 (left). If the observation is far from this prior, the filtering distribution suddenly jumps at time $T = 10$. This is a recipe for disaster in SMC: the particles at

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

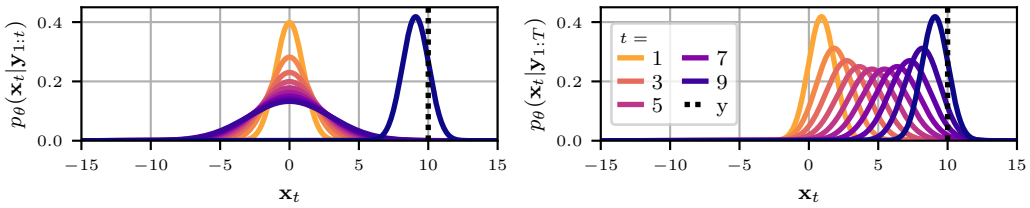


Figure 1: *The filtering (left) and smoothing (right) distributions for the Gaussian random walk with a single observation at time* $T = 10$. For $t = 1, \ldots, 9$ the filtering distributions reduce to the prior, a mean zero Gaussian distribution. At $t = 10$, the filtering distributions incorporate the single observation, resulting in a sudden shift and particle death. In contrast, the smoothing distributions change slowly over time, which leads to good particle approximations across all timesteps and low-variance SMC estimates. Efficiently approximating these smoothing distributions is the focus of this paper.

time $T - 1$ will be distributed according to a mean zero Gaussian, and very few will propagate to the next time step. When particles fail to propagate, the variance of the SMC estimator explodes.

Suppose instead that the target distributions were the *smoothing distributions*—the conditional distribution over the latents $\mathbf{x}_{1:t}$ given *all* observations $\mathbf{y}_{1:T}$. Figure 1 (right) shows the smoothing distributions for the simple Gaussian walk. Unlike the filtering distributions, the smoothing distributions shift slowly toward the observation over time. These slow, smooth changes are ideal for SMC: many particles will survive from one time step to the next, and the variance of the SMC estimator will be minimized.

In practice we do not have access to the smoothing distribution—if we did, there would be little need for SMC! Here, we introduce a new method called SIXO: Smoothing Inference for Twisted Objectives. SIXO provides a unified approach for learning model and proposal parameters, as well as a set of *twisting functions* that warp the filtering distributions into targets that better approximate the smoothing distributions. Like its predecessor FIVO [5–7], SIXO treats this problem using a variational approach, deriving a lower bound to the marginal likelihood. Unlike its predecessor, we prove that the SIXO bound can become tight, even with finitely many particles.

The key challenge with SIXO is learning the twisting functions. We consider multiple approaches and find that a simple density ratio estimation approach works best, and propose an algorithm which interleaves twist updates with updates to the model and proposal. Thus, SIXO offers a means of jointly learning the model parameters as well as SMC proposals and targets for accurate posterior inference.

Finally, we give empirical evidence to support our theoretical claims. Across a range of experiments with synthetic data, a stochastic volatility model of currency exchange rates, and a Hodgkin-Huxley model of membrane potential in a neuron, SIXO consistently outperforms FIVO and related methods. We dissect these results to illustrate how learning better targets enables more effective posterior inference and learning.

2 Background

Consider modeling sequential data $\mathbf{y}_{1:T} \in \mathcal{Y}^T$ using a sequence of unobserved latent variables $\mathbf{x}_{1:T} \in \mathcal{X}^T$ with Markovian structure, and let the joint distribution factorize as

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_1)p_{\boldsymbol{\theta}}(\mathbf{y}_1 \mid \mathbf{x}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{t-1})p_{\boldsymbol{\theta}}(\mathbf{y}_t \mid \mathbf{x}_t) \tag{1}$$

with global parameters $\boldsymbol{\theta} \in \Theta$. We further assume that the conditional distributions $p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and $p_{\boldsymbol{\theta}}(\mathbf{y}_t \mid \mathbf{x}_t)$ may depend nonlinearly on $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, respectively.

In general, the marginal likelihood and posterior for this model class are not readily available from the joint distribution due to the intractable integral over the latents $\mathbf{x}_{1:T}$, i.e.

$$p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}) = \int_{\mathcal{X}^T} p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}, \mathbf{x}_{1:T}) \, \mathrm{d}\mathbf{x}_{1:T},$$

cannot easily be computed due to the form of the conditional distributions.

2.1 Sequential Monte Carlo

A standard method for inference in sequential latent variable models is sequential Monte Carlo (SMC), which works by sampling from a sequence of "target" distributions that only need be evaluable pointwise [1]. Let $\{\gamma_t(\mathbf{x}_{1:t})\}_{t=1}^T$ be the unnormalized target distributions and let $\{\pi_t(\mathbf{x}_{1:t})\}_{t=1}^T$ be their normalized counterparts, with $Z_t = \int \gamma_t(\mathbf{x}_{1:t}) \, \mathrm{d}x_{1:t}$ and $\pi_t(\mathbf{x}_{1:t}) = \gamma_t(\mathbf{x}_{1:t})/Z_t$. As long as mild technical conditions are met [1, 2], and $\gamma_T(\mathbf{x}_{1:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})$, SMC returns an unbiased estimate of the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ and a set of weighted particles approximating the posterior $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T})$ [1]. For more details see the appendix.

A key to designing a successful SMC algorithm is ensuring the sequence of target distributions bridge smoothly from $\pi_1(\mathbf{x}_1)$ to $\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T})$. When $\pi_t(\mathbf{x}_{1:t})$ is close to the true posterior $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}|\mathbf{y}_{1:T})$, SMC's marginal likelihood estimate will be low variance [1]. If the distributions are mismatched, however, then trajectories likely under the posterior but unlikely under the target distributions will be discarded by SMC's resampling operation, increasing variance.

2.2 Filtering SMC and Model Learning

The most commonly-used SMC algorithm is filtering SMC, which sets the normalized targets to the *filtering* distributions, i.e. $\gamma_t(\mathbf{x}_{1:t}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ and $\pi_t(\mathbf{x}_{1:t}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$. Let $\widehat{Z}_{\mathrm{FSMC}}(\boldsymbol{\theta}, \mathbf{y}_{1:T})$ be the marginal likelihood estimator returned from running filtering SMC with proposal distributions $\{q_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})\}_{t=1}^T$ which may share parameters with $p_{\boldsymbol{\theta}}$.

Previous work used filtering SMC to fit model parameters by ascending a lower bound on the log marginal likelihood called a *filtering variational objective* (FIVO) [5–8]. The FIVO bound is derived using the unbiasedness of $\widehat{Z}_{\mathrm{FSMC}}$ and Jensen's inequality,

$$\mathcal{L}_{\mathrm{FIVO}}(\boldsymbol{\theta}, \mathbf{y}_{1:T}) \triangleq \mathbb{E}[\log \widehat{Z}_{\mathrm{FSMC}}(\boldsymbol{\theta}, \mathbf{y}_{1:T})] \leq \log \mathbb{E}[\widehat{Z}_{\mathrm{FSMC}}(\boldsymbol{\theta}, \mathbf{y}_{1:T})] = \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}),$$

and is optimized using stochastic gradient ascent in $\boldsymbol{\theta}$ [5–8].

2.3 Smoothing SMC via Twisting Functions

One disadvantage of filtering SMC is that when the posterior and filtering distributions are mismatched, i.e. $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$ is not a good approximation of $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:T})$, the marginal likelihood estimator $\widehat{Z}_{\mathrm{FSMC}}$ can be high variance [9]. This variance leads to loose lower bounds and poor model learning [5]. Intuitively, this is because filtering SMC iteratively samples from the filtering distributions which do not incorporate information from future observations. In many models of interest the future observations $\mathbf{y}_{t+1:T}$ contain the most information about the current latent state $\mathbf{x}_t$. Therefore, failing to take future observations into account can drastically increase the variance of the marginal likelihood estimate and decrease model learning performance.

Performing smoothing SMC would resolve this issue by choosing the targets to be the *smoothing* distributions, i.e. $\pi_t(\mathbf{x}_{1:t}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:T})$ and $\gamma_t(\mathbf{x}_{1:t}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:T})$. Unfortunately, $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:T})$ is not readily available from the model.

However, $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:T})$ can be factored into the product of the filtering distributions, $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$, and the *lookahead* distributions, $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t)$. If the lookahead distributions can be well-approximated by a series of "twisting" functions, $\{r(\mathbf{y}_{t+1:T}, \mathbf{x}_t)\}_{t=1}^T$, then running an SMC algorithm with targets $\gamma_t(\mathbf{x}_{1:t}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) r(\mathbf{y}_{t+1:T}, \mathbf{x}_t)$ would approximate smoothing SMC. The function r is referred to as a twisting function because it can be interpreted as changing or twisting the standard filtering targets [10].

Different twisting functions yield different SMC methods such as auxiliary particle filters and twisted particle filters [11, 10, 12]. As long as the final target $\gamma_T(\mathbf{x}_{1:T})$ is equal to $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})$, SMC's

marginal likelihood estimate remains unbiased, regardless of the choice of twisting functions. Instead, the quality of the twisting functions affects the variance of SMC's marginal likelihood estimate [1].

3 SIXO: Model Learning with Smoothing SMC

Our goal is to perform model learning by optimizing a lower bound on the marginal likelihood constructed using smoothing SMC. To construct the lower bound, fix $r_{\boldsymbol{\psi}}(x_T) = 1$ and let $\widehat{Z}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T})$ be the marginal likelihood estimator returned from running SMC with target distributions $\{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:T}, \mathbf{x}_t)\}_{t=1}^{T}$ and proposal distributions $\{q_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:T})\}_{t=1}^{T}$. Because the target distribution at time T is $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})$, $\widehat{Z}_{\text{SIXO}}$ will be an unbiased estimator of the true marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ [1]. This implies via Jensen's inequality that

$$\mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T}) \triangleq \mathbb{E}\left[\log \widehat{Z}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T})\right] \leq \log \mathbb{E}\left[\widehat{Z}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T})\right] = \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}) \quad (2)$$

i.e. $\mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T})$ is a lower bound on the log marginal likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ [8]. Note that this holds for any setting of the parameters, and by extension any twisting functions $r_{\boldsymbol{\psi}}$.

3.1 The Functional Form of the Twists

The structure of the lookahead distributions $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t)$ suggests a functional form for $r_{\boldsymbol{\psi}}$ that accepts a single latent $\mathbf{x}_t$ and produces distributions over all future observations $\mathbf{y}_{t+1:T}$. Because the twists will be evaluated once per timestep in an SMC sweep, this functional form would lead to an algorithm with $O(T^2)$ complexity. To reduce the complexity, we consider two methods: fixed-lag twisting and backwards twisting [13–16].

Fixed-lag twisting approximates the full lookahead distribution $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t)$ using a fixed window of L observations, i.e. it models $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:t+L} \mid \mathbf{x}_t)$. Specifically, we define the fixed-lag twisting functions $\{r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:t+L}, x_t)\}_{t=1}^{T-1}$ as a sequence of functions which accept $\mathbf{x}_t \in \mathcal{X}$ and produce a distribution over $\mathbf{y}_{t+1:t+L} \in \mathcal{Y}^L$. This reduces the computational complexity to $O(TL)$ at the cost of only looking at L observations.

In our experiments we also use a fixed-lag quadrature twist that scores only the next observation by computing $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1} \mid \mathbf{x}_t) = \int p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1}, \mathbf{x}_{t+1} \mid \mathbf{x}_t) d\mathbf{x}_{t+1}$ using numerical quadrature.

Backwards twisting is motivated by rewriting the lookahead distributions using Bayes' rule,

$$p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{y}_{t+1:T})\, p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T})}{p_{\boldsymbol{\theta}}(\mathbf{x}_t)} \propto \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{y}_{t+1:T})}{p_{\boldsymbol{\theta}}(\mathbf{x}_t)},$$

noting that terms which do not depend on $\mathbf{x}$ are constant because the tilting functions will be used to score particles in SMC. Thus, we need only approximate $p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{y}_{t+1:T}) / p_{\boldsymbol{\theta}}(\mathbf{x}_t)$. The numerator in this functional form is the reverse of the lookahead distributions—it accepts all future observations and produces a distribution over a single latent. This makes it possible to parameterize the twists using a recurrent function approximator (e.g. a recurrent neural network or RNN) run backwards across the observations $\mathbf{y}_{1:T}$ to produce the tilt values for each timestep.

Specifically, we define the backwards twists $\{r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:T}, \mathbf{x}_t)\}_{t=1}^{T-1}$ as a sequence of scalar-valued functions $\mathcal{Y}^{T-t} \times \mathcal{X} \to \mathbb{R}$ with parameters $\boldsymbol{\psi}$. Parameterizing backward twists with a recurrent function approximator results in $O(T)$ time complexity and allows the twist to condition on all future observations, making backwards twisting preferable to fixed-lag twisting.

3.2 Learning Twists

Ascending the Unified Objective One way to fit the twists and model is to ascend $\mathcal{L}_{\text{SIXO}}$ in the parameters of $p_{\boldsymbol{\theta}}, q_{\boldsymbol{\theta}}$, and $r_{\boldsymbol{\psi}}$, similar to FIVO [5–7]. The gradients of this objective include score-function terms that arise from the discrete resampling steps in SMC. We refer to ascending $\mathcal{L}_{\text{SIXO}}$ with these unbiased gradients as SIXO-u. Because the resampling gradient terms have high variance, SIXO-u is impractical for complex settings [16, 5]. For a detailed discussion and derivation of the gradient, see the appendix.

Algorithm 1 SIXO-DRE

procedure DRE($\boldsymbol{\theta}, \boldsymbol{\psi}, N$)
  $\boldsymbol{\psi}_0 = \boldsymbol{\psi}$
  for $i = 1, \ldots, N$ **do**
    $\tilde{\mathbf{x}}^j_{1:T} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}), \quad j = 1, \ldots, M$
    $\mathbf{x}^j_{1:T}, \mathbf{y}^j_{1:T} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}), \quad j = 1, \ldots, M$
    $\mathcal{L}_{\text{DRE}}(\boldsymbol{\psi}) = \frac{1}{M(T-1)} \sum_{j=1}^{M} \sum_{t=1}^{T-1} \log \sigma(r_{\boldsymbol{\psi}}(\mathbf{y}^j_{t+1:T}, \mathbf{x}^j_t)) + \log(1 - \sigma(r_{\boldsymbol{\psi}}(\mathbf{y}^j_{t+1:T}, \tilde{\mathbf{x}}^j_t)))$
    Compute $\boldsymbol{\psi}_i$ using the gradients of $\mathcal{L}_{\text{DRE}}$ evaluated at $\boldsymbol{\psi}_{i-1}$
  return $\boldsymbol{\psi}_N$

procedure FIVO($\boldsymbol{\theta}, \{\gamma_t(\mathbf{x}_{1:t})\}_{t=1}^T, \{q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:T})\}_{t=1}^T, K, N$)
  $\boldsymbol{\theta}_0 = \boldsymbol{\theta}$
  for $i = 1, \ldots, N$ **do**
    $\mathcal{L}(\boldsymbol{\theta})$ = SMC($\{\gamma_t(\mathbf{x}_{1:t})\}_{t=1}^T$, $\{q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:T})\}_{t=1}^T$, K).
    Compute $\boldsymbol{\theta}_i$ using the biased gradients of $\mathcal{L}$ evaluated at $\boldsymbol{\theta}_{i-1}$
  return $\boldsymbol{\theta}_N$

procedure SIXO-DRE($\mathbf{y}_{1:T}$, $\boldsymbol{\theta}_0$, $\boldsymbol{\psi}_0$, K, N, S)
  for $s = 1, \ldots, S$ **do**
    $\boldsymbol{\psi}_s$ =DRE($\boldsymbol{\theta}_{s-1}, \boldsymbol{\psi}_{s-1}, N$)
    $\boldsymbol{\theta}_s$ =FIVO($\boldsymbol{\theta}_{s-1}, \{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:T}, \mathbf{x}_t)\}_{t=1}^T, \{q_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:T})\}_{t=1}^T, K, N$)
  return $\boldsymbol{\theta}_S, \boldsymbol{\psi}_S$

Learning Twists via Density Ratio Estimation To cast twist learning as density ratio estimation, note that the optimal backwards twist is the ratio of a "backwards message" $p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{y}_{t:1:T})$ and the latent marginal $p_{\boldsymbol{\theta}}(\mathbf{x}_t)$. Thus, we can learn the backwards twist using density ratio estimation (DRE) [17, 18].

Let $\{r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:T}, \mathbf{x}_t)\}_{t=1}^{T-1}$ be a sequence of backwards twist functions as defined in Section 3.1. Density ratio estimation via classification suggests training $r_{\boldsymbol{\psi}}$ as a classifier to distinguish between samples from $p_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}_{t+1:T})$ and samples from $p_{\boldsymbol{\theta}}(\mathbf{x}_t) p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T})$ using the binary cross-entropy loss [18]. Samples from these distributions are available from the model. When trained in this way, the logit produced by $r_{\boldsymbol{\psi}}(\mathbf{y}_{t+1:T}, \mathbf{x}_t)$ will approximate $\log p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{y}_{t+1:T}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}_t)$ up to an additive constant which can be ignored. For details, see the appendix and [18].

We use the DRE-learned twisting functions in an alternating scheme that first holds $p_{\boldsymbol{\theta}}, q_{\boldsymbol{\theta}}$ fixed and updates $r_{\boldsymbol{\psi}}$ using density ratio estimation, and then holds $r_{\boldsymbol{\psi}}$ fixed and updates $p_{\boldsymbol{\theta}}$ and $q_{\boldsymbol{\theta}}$ by ascending a biased gradient estimator (no resampling terms) of $\mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi})$ in $\boldsymbol{\theta}$. We call the full alternating procedure for learning both $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ SIXO-DRE, for details see Algorithm 1.

3.3 The SIXO Bound Can Become Tight

Maddison et al. [5] show that the FIVO bound can only become tight in models with uncommon dependency structures. We show that the SIXO bound can become tight for any model in the class defined in Section 2.

Proposition 1. Sharpness of the SIXO bound. *Let $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})$ be a latent variable model with Markovian structure as defined in Section 2, let $\mathcal{Q}$ be the set of possible proposal functions indexed by parameters $\boldsymbol{\theta} \in \Theta$, and let $\mathcal{R}$ be the set of possible twist functions indexed by parameters $\boldsymbol{\psi} \in \Psi$. Assume that $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:T}) \in \mathcal{Q}$ and $p(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t) \in \mathcal{R}$. Finally, let $\boldsymbol{\theta}^*, \boldsymbol{\psi}^* = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\psi}} \mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{y}_{1:T})$.*

Then the following holds:

1. $q_{\boldsymbol{\theta}^*}(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:T}) = p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:T})$ *for* $t = 1, \ldots, T$,
2. $r_{\boldsymbol{\psi}^*}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t) = p(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t)$ *for* $t = 1, \ldots, T-1$,
3. $\mathcal{L}_{\text{SIXO}}(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*, \mathbf{y}_{1:T}) = \log p(\mathbf{y}_{1:T})$ *for any number of particles* $K \geq 1$.

Proof. See Appendix. □

This is an important advantage of our work—the SIXO objective is the first to recover the true marginal likelihood with a finite number of particles while also being tailored to sequential tasks.

4 Related Work

Lin et al. [13] provide an extensive summary of look ahead techniques. These ideas were first introduced by Pitt and Shephard [11], with the *auxiliary particle filter* (APF), which constructs an estimate of the one-step backwards message $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1} \mid \mathbf{x}_t)$ using pilot simulations from model itself. Guarniero et al. [15] introduce the *iterated auxiliary particle filter* (iAPF). The iAPF builds on the APF by iteratively refining a parametric approximation of a recursive factorization of the full backwards message, $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+1:T} \mid \mathbf{x}_t)$, as a function of $\mathbf{x}_t$ using forward simulations from the (iA)PF on the previous step. Heng et al. [14] introduce *controlled sequential Monte Carlo* (cSMC), where at each iteration an additional twist is fitted to the residuals from the previous twisted SMC sweep using approximate dynamic programming. Park and Ionides [19] consider learning a parametric twist conditioned on a finite window of future data as an approximation to the full backwards message. However, windowed twists are disadvantageous as they have complexity per particle that scales in the length of the window. Importantly, these approaches do not consider model learning. We also propose learning the tilt using density ratio estimation by recasting the twist value as a classification logit, that does not require positing a particular distributional family for the look-ahead density over $\mathbf{y}_{1:T}$ or over latent state $\mathbf{x}_t$.

Moretti et al. [20] construct a "smoothed" FIVO bound by using the a forwards-filtering backwards-smoothing approach to sample particle lineages to reduce variance. However, this approach still suffers the same intrinsic pathology as FIVO, in that divergences between the filtering and smoothing distributions still result in few particles dominating in the backwards pass. Our work is similar in intent to Kim et al. [21], who develop a lower-variance IWAE gradient estimator by constructing a *baseline* from future likelihood estimates. We begin by considering the already-tighter FIVO bound, where constructing baselines is not tractable due to the interactions of particles in the resampling steps.

5 Experiments

Our experiments explore our claims that:

1. The SIXO bound can become tight while FIVO can not.
2. SIXO provides better posterior inference over latent states than FIVO.
3. Model learning with SIXO provides better parameter estimates than FIVO.

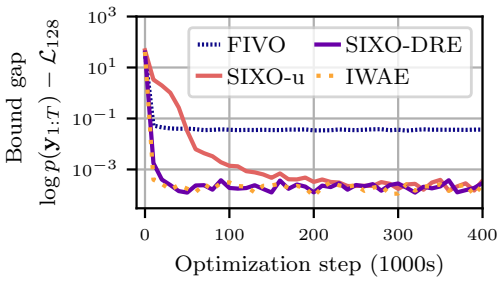
5.1 Gaussian Drift Diffusion

We first consider a one-dimensional Gaussian drift-diffusion (GDD) process with joint distribution

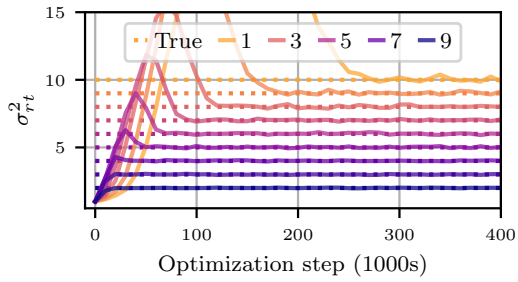
$$p_{\boldsymbol{\theta}}\left(\mathbf{x}_{1:T}, y\right) = \mathcal{N}\left(y \mid x_T + \alpha, \sigma_y^2\right) \mathcal{N}\left(x_1; \alpha, \sigma_x^2\right) \prod_{t=2}^{T} \mathcal{N}\left(x_t \mid x_{t-1} + \alpha, \sigma_x^2\right). \tag{3}$$

There is a single free model parameter α, the state is $x_t \in \mathbb{R}$, and the observed data are $y \in \mathbb{R}$. Figure 1, which is for drift $\alpha = 0$, shows that the filtering and smoothing distributions in this model quickly diverge, which can lead to poor inference for filtering methods.

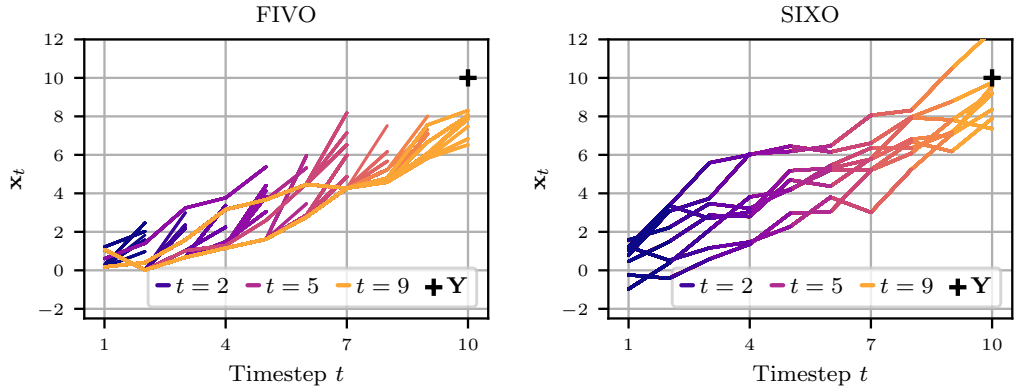
We compare joint model, proposal and twist learning using two variants of SIXO to VI using the IWAE bound [22] and FIVO with unbiased gradients [5–7]. All methods use an independent proposal at each time step parameterized as $q_t(x_t \mid x_{t-1}, y) = \mathcal{N}(x_t; f_t(x_{t-1}, y), \sigma_{qt}^2)$ where f_t is an affine function. This variational family contains the optimal proposal. SIXO-u optimizes the unified objective in (2) using unbiased gradients, with twists parameterized as $r_t(y, x_t) = \mathcal{N}(y; g_t(x_t), \sigma_{rt}^2)$, where g_t is an affine function. SIXO-DRE, in turn, uses the alternating updates in Algorithm 1. The twist, represented as a log density ratio, $\log r_t(y, x_t)$, is parameterized as a quadratic function of $\mathbf{x}_t$,



(a) Bound gap for different methods.



(b) Convergence for a twist parameter using SIXO-u.



(c) Smoothing particle lineages under FIVO (left) and SIXO (right). Although the proposal learned by FIVO is exploiting smoothing information (particles are proposed upwards towards the observed value), the filtering nature of SMC "resists" this, resampling particles towards the prior distribution. SIXO also leverages smoothing information, but proposed particles are preserved by using a learned sequence of target distributions.

Figure 2: Results for the one-sample Gaussian drift diffusion experiment presented in Section 5.1. Further figures, derivations and analysis are included in the supplement.

where the parameters of the quadratic function are generated by an MLP with inputs (y, t). Both of these parameterizations contain the true twist.

Figure 2a shows the convergence of the variational bound for each method. As expected IWAE recovers a tight variational bound, whereas FIVO does not. While SIXO-u does recover a tight variational bound, the high variance of the unbiased gradient estimator makes it impractical for non-toy problems. Conversely, SIXO-DRE achieves a tight bound but under biased, lower variance gradients. This motivates its use in more complex, non-linear settings where the unbiased FIVO gradients are not practical. Figure 2b shows that SIXO-u recovers the correct twist parameters. More figures illustrating the convergence of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are included in the supplement.

In Figure 2c we compare particle trajectories under FIVO and SIXO-u. We see that FIVO consistently proposes particles with high likelihood under the posterior, which are then discarded by the resampling steps in filtering SMC. In contrast, SIXO both proposes particles with high posterior likelihood and retains them through the resampling steps by properly scoring particles under the twisted target distributions. These results empirically verify the theoretical claims made in Section 3.3.

5.2 Stochastic Volatility Model

We now apply SIXO to a stochastic volatility model (SVM) of monthly foreign exchange rates for $N = 22$ currencies in the period from 9/2007 to 8/2017 [23]. The SVM generative model is

$$\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{x}_t = \boldsymbol{\mu} + \boldsymbol{\phi}(\mathbf{x}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\nu}_t, \quad \mathbf{y}_t = \boldsymbol{\beta} \exp\left(\frac{\mathbf{x}_t}{2}\right) \mathbf{e}_t, \tag{4}$$

with transition noise $\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, observation noise $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, I_{N \times N})$, states $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times N}$, and observations $\mathbf{y}_{1:T} \in \mathbb{R}^{T \times N}$. All multiplications are performed element-wise. The model has

Table 1: Performance of FIVO and SIXO on the SVM.

Method	Train Bound	Train Bound Value (as in [6])	Train $\mathcal{L}^{2048}_{\text{BPF}}$	Test $\mathcal{L}^{2048}_{\text{BPF}}$
FIVO	$\mathcal{L}^4_{\text{FIVO}}$	6923.66 ± 5.71	$\mathbf{58.98 \pm 0.0039}$	3352.51 ± 0.14
SIXO-q	$\mathcal{L}^4_{\text{SIXO-q}}$	6930.56 ± 4.76	$\mathbf{58.98 \pm 0.0052}$	3353.10 ± 0.81
SIXO-DRE	$\mathcal{L}^4_{\text{SIXO-DRE}}$	$\mathbf{6933.77 \pm 3.45}$	$\mathbf{58.98 \pm 0.0040}$	$\mathbf{3354.08 \pm 0.67}$

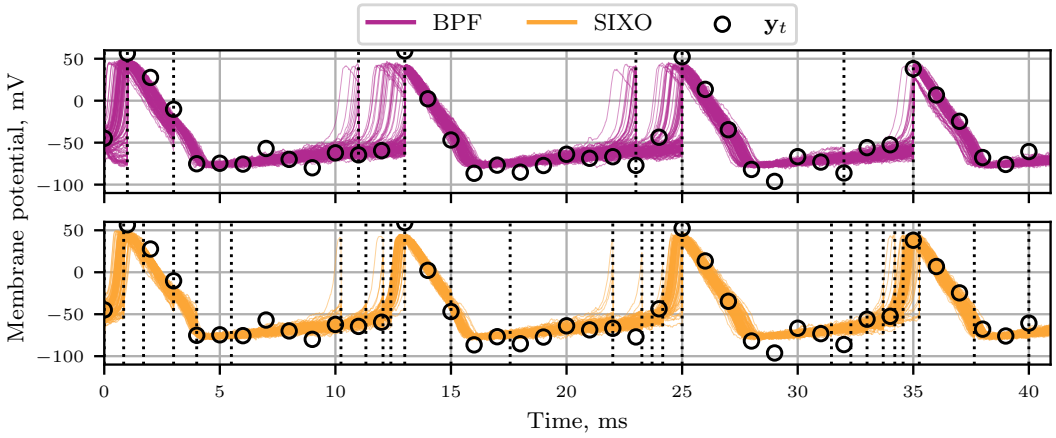


Figure 3: Comparison of the filtering distributions generated by a bootstrap particle filter (BPF) (top) and a SIXO sweep (bottom) on synthetic data from the Hodgkin-Huxley model. Dotted vertical lines are resampling events. Both sweeps use true model parameters and a bootstrap proposal, but SIXO uses DRE-trained twisting functions. We see that the twist has reduced the number of erroneous spikes generated under the BPF, and more particles accurately predict the initiation of a spike.

free parameters $\boldsymbol{\mu} \in \mathbb{R}^N, \boldsymbol{\phi} \in [0,1]^N, \boldsymbol{\beta} \in \mathbb{R}^N_+$, and $\mathbf{Q} \in \text{diag}(\mathbb{R}^N_+)$ such that there are $4N$ model parameters. The proposal, $q_{\boldsymbol{\theta}}$, is structured as $q_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}) \propto \prod_{t=1}^T \mathcal{N}(\mathbf{x}_t; \mu_t, \sigma_t^2) p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, such that there are $2N$ proposal parameters.

We compare three approaches: FIVO, SIXO with quadrature tilt (SIXO-q), and SIXO with density ratio tilt (SIXO-DRE). For more specifics and hyperparameters, see the supplement.

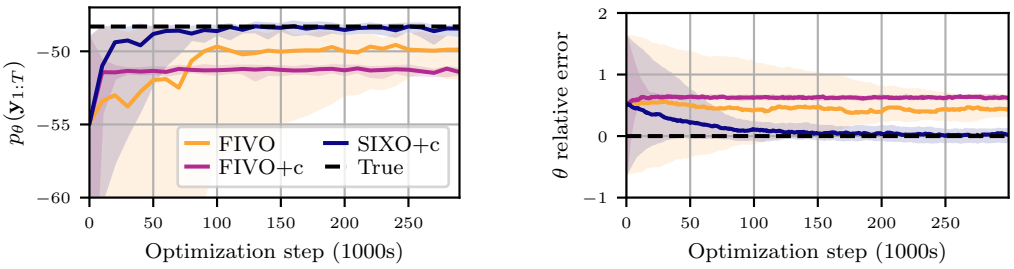
Train Bound Performance We first compare our methods in terms of log marginal likelihood lower bounds as in Naesseth et al. [6], see the left column of Table 1. Quadrature-based tilts outperform FIVO, and density ratio tilts outperform quadrature tilts. Note that as in Naesseth et al. [6], for each method we report its corresponding bound on the training set, so for FIVO we report the FIVO bound, for SIXO-q we report the SIXO-q bound, and for SIXO-DRE we report the SIXO-DRE bound.

Test Set Performance We also compare methods on a held-out test set to evaluate each method's influence on model learning. We construct this test set using the same data source as the training set, but use the period of time since Naesseth et al. [6] was published (an extra 55 months). We report the log marginal likelihood lower bound returned by a bootstrap particle filter ($q = p$) with 2048 particles, denoted $\mathcal{L}^{2048}_{\text{BPF}}$. We evaluate all checkpoints after the halfway point of training.

Interestingly, training set $\mathcal{L}^{2048}_{\text{BPF}}$ values for all models are the same, suggesting that training performance has saturated. It is clear, however, that SIXO performs better inference as it has higher train bound values (3rd column) than FIVO for roughly the same models.

5.3 Hodgkin-Huxley Model

We conclude by comparing FIVO and SIXO on the Hodgkin-Huxley (HH) model of neural action potentials [24, 25]. A single neuron is represented with a four-dimensional state-space: the instantaneous membrane potential and the relative conductivity of three ion gates. A noise-corrupted and subsampled membrane potential can be obtained using electrodes [26] or voltage imaging [27]. The



(a) $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ on a held-out validation set.

(b) Relative error in parameter estimate (0 = no error).

Figure 4: Results for learning the HH model. SIXO is the only method to recover the true parameter.

Table 2: Performance of FIVO and SIXO on the Hodgkin-Huxley model.

Method	Train Bound	Train Bound Value	Test $\mathcal{L}^{256}_{\text{BPF}}$	Relative Parameter Error
(True model)	(N/A)	(N/A)	(-48.30)	(0.0 ± 0.0)
FIVO	$\mathcal{L}^4_{\text{FIVO}}$	-49.80 ± 0.61	-50.06 ± 0.55	0.43 ± 0.12
FIVO (+ clip)	$\mathcal{L}^4_{\text{FIVO}}$	-51.27 ± 0.23	-51.30 ± 0.25	0.62 ± 0.02
SIXO	$\mathcal{L}^4_{\text{SIXO}}$	$\mathbf{-47.86 \pm 0.12}$	$\mathbf{-48.68 \pm 0.26}$	$\mathbf{0.011 \pm 0.09}$

state of the gates, however, is not observable, and must be inferred from the noisy potential recordings. The physiological parameters governing the time-evolution of the system are also of interest, such as the base conductance of each of the ion channels.

We implement the HH model as a four-dimensional nonlinear SLVM with Gaussian transition noise [28]. The observation is a single Gaussian-distributed value with mean equal to the instantaneous potential. We subsample observations by a factor of 50 to simulate an acquisition frequency of 1kHz. For more details, see the appendix.

In this model action potentials, or spikes, are rare events that happen quickly and invoke a rapid change in the state. Therefore, filtering-based inference is particularly disadvantageous, as noisy observations may artificially trigger spikes or "miss" spikes.

Inference We demonstrate this shortcoming in Figure 3, which shows a BPF generating spurious spikes and missing the initiation of other spikes. SIXO, despite using an unlearned bootstrap proposal, generates fewer spurious spikes and fewer particles miss spikes. SIXO also achieves a higher log marginal lower bound (-73.88 nats) than the bootstrap particle filter (-74.89 nats), showing that it performs more effective inference. These results hint at the broader potential for DRE twists as a flexible, general-purpose tool for improving inference in non-linear models, even under a fixed model and proposal.

Model Learning We conclude by comparing FIVO and SIXO for parameter recovery in Figure 4. The relative parameter error (rightmost column) is computed as $(\boldsymbol{\theta}^{\text{true}} - \boldsymbol{\theta}^*)/\boldsymbol{\theta}^{\text{true}}$. We found gradient clipping improves stability in SIXO, so we compare to FIVO with and without clipping. FIVO with clipping converges quickly to an incorrect parameter that achieves a low marginal likelihood, while FIVO without clipping converges more slowly to parameters that achieves a higher bound. SIXO is the only method to recover the correct model, and achieves the highest log marginal likelihood.

6 Conclusion

In this work we tackled the long-standing problem of constructing an efficient method to learn twisting functions for smoothing SMC, and then used smoothing SMC to learn better models. We accomplished this by casting twist function learning as density ratio estimation. The learned twists yielded a sequential model learning objective that can theoretically become tight, and we experimentally demonstrated that it improves inference and model learning.

References

[1] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering*, pages 656–704. Oxford University Press, 2011.

[2] Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

[3] Christophe Andrieu, Arnaud Doucet, Sumeetpal S Singh, and Vladislav B Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92 (3):423–438, 2004.

[4] Shixiang Shane Gu, Zoubin Ghahramani, and Richard E Turner. Neural adaptive sequential monte carlo. *Advances in neural information processing systems*, 28, 2015.

[5] Chris J Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. *arXiv preprint arXiv:1705.09279*, 2017.

[6] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential Monte Carlo. In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR, 2018.

[7] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential Monte Carlo. *arXiv preprint arXiv:1705.10306*, 2017.

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

[9] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010.

[10] Nick Whiteley and Anthony Lee. Twisted particle filters. *The Annals of Statistics*, 42(1): 115–141, 2014.

[11] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[12] Roland Hostettler and Thomas B Schön. Auxiliary-particle-filter-based two-filter smoothing for wiener state-space models. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.

[13] Ming Lin, Rong Chen, and Jun S Liu. Lookahead strategies for sequential monte carlo. *Statistical Science*, 28(1):69–94, 2013.

[14] Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

[15] Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.

[16] Dieterich Lawson, George Tucker, Christian A Naesseth, Chris Maddison, Ryan P Adams, and Yee Whye Teh. Twisted variational sequential Monte Carlo. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[17] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[18] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[19] Joonha Park and Edward L Ionides. Inference on high-dimensional implicit dynamic models using a guided intermediate resampling filter. *Statistics and Computing*, 30(5):1497–1522, 2020.

[20] Antonio Moretti, Zizhao Wang, Luhuan Wu, Iddo Drori, and Itsik Pe'er. Variational objectives for markovian dynamics with backward simulation. In *ECAI 2020*, pages 1371–1378. IOS Press, 2020.

[21] Geon-Hyeong Kim, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. Variational inference for sequential data with future likelihood estimates. In *International Conference on Machine Learning*, pages 5296–5305. PMLR, 2020.

[22] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[23] Siddhartha Chib, Yasuhiro Omori, and Manabu Asai. Multivariate stochastic volatility. In *Handbook of financial time series*, pages 365–400. Springer, 2009.

[24] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[25] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

[26] Justin M Kita and R Mark Wightman. Microelectrodes for studying neurobiology. *Current opinion in chemical biology*, 12(5):491–496, 2008.

[27] Darcy S Peterka, Hiroto Takahashi, and Rafael Yuste. Imaging voltage in neurons. *Neuron*, 69 (1):9–21, 2011.

[28] Quentin JM Huys and Liam Paninski. Smoothing of, and parameter estimation from, noisy biophysical recordings. *PLoS computational biology*, 5(5):e1000379, 2009.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We present a method build on FIVO [5] to include smoothing information.
   (b) Did you describe the limitations of your work? [Yes] We especially highlight the persistent problem of biased gradient estimators in non-trivial models, and the dependency on the variational family used for each set of function approximators.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We foresee no negative social impacts arising from this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Sketch proofs are included in the main paper, and expanded proofs and commentaries are included in the supplement.
   (b) Did you include complete proofs of all theoretical results? [Yes] See above.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release a public version of the code imminently, but do not include it here for risk of de-annonymization.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Extensive written configurations are included in the supplement. Weights and Biases logs, Git commits and scripts for exact reproduction of included results are contained in the GitHub repository.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We list the types of nodes and runtimes for the algorithms presented.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]