Nesting Particle Filters for Experimental Design in Dynamical Systems

Sahel Iqbal [1] **Adrien Corenflos** [1] **Simo Särkkä** [1] **Hany Abdulsamad** [1]

Abstract

In this paper, we propose a novel approach to Bayesian experimental design for non-exchangeable data that formulates it as risk-sensitive policy optimization. We develop the Inside-Out SMC^2 algorithm, a nested sequential Monte Carlo technique to infer optimal designs, and embed it into a particle Markov chain Monte Carlo framework to perform gradient-based policy amortization. Our approach is distinct from other amortized experimental design techniques, as it does not rely on contrastive estimators. Numerical validation on a set of dynamical systems showcases the efficacy of our method in comparison to other state-of-the-art strategies.

1. Introduction

Traditionally, Bayesian inference on the parameters of a statistical model is performed *after the fact* by employing a posterior elicitation routine to previously gathered data. However, in many scenarios, experimenters can proactively *design* experiments to acquire maximal information about the parameters of interest. Bayesian experimental design (BED, Lindley, 1956; Chaloner & Verdinelli, 1995) offers a principled framework to achieve this goal by maximizing the expected information gain over the design space. BED has found applications in fields as varied as active learning (Bickford Smith et al., 2023), neuroscience (Shababo et al., 2013), physics (McMichael et al., 2021), psychology (Myung et al., 2013), and robotics (Schultheis et al., 2020). A recent overview of the field of BED can be found in Rainforth et al. (2024).

In Bayesian experimental design, we are given a prior $p(\theta)$ and a likelihood $p(x \mid \xi, \theta)$, where $\theta \in \Theta$ is the set of unknown parameters of interest, $x \in \mathcal{X}$ is the experimental outcome, and $\xi \in \Xi$ is a controllable design. The information gain (IG, Lindley, 1956) in a parameter θ upon applying a design ξ and observing an outcome x is defined as

$$\mathcal{G}(x, \xi) \coloneqq \mathbb{H}[p(\theta)] - \mathbb{H}[p(\theta \mid x, \xi)],$$

with $\mathbb{H}[p(\cdot)] \coloneqq -\mathbb{E}_{p(\cdot)}[\log p(\cdot)]$ denoting the entropy of a random variable with probability density p. Since the outcomes x are themselves random variables for a fixed design ξ, the goal in BED is to choose a design ξ^* that maximizes the *expected* information gain (EIG), defined as

$$\mathcal{I}(\xi) \coloneqq \mathbb{E}_{p(x|\xi)}\big[\mathbb{H}[p(\theta)] - \mathbb{H}[p(\theta \mid x, \xi)]\big], \qquad (1)$$

where $p(x \mid \xi) = \mathbb{E}_{p(\theta)}[p(x \mid \theta, \xi)]$. The expected information gain thus quantifies the decrease in uncertainty in the unknown variable θ upon implementing a design ξ.

While mathematically elegant, the BED framework involves maximizing nested expectations over intractable quantities such as the marginal likelihood of x and the posterior of θ appearing in (1). This is a computationally intensive task (Kueck et al., 2009; Rainforth et al., 2018), which becomes even more challenging when optimizing designs for a series of experiments conducted sequentially, where the impact of each individual design needs to be accounted for across the entire sequence of experiments. This makes the deployment of sequential BED close to impossible on real-time systems with high-frequency data.

Huan & Marzouk (2016) addressed this limitation by introducing a parametric policy to predict designs as a function of the running parameter posterior, thereby eliminating the cost of the maximization step in each experiment. Foster et al. (2021) extended this idea to *amortize* the overall cost of sequential BED by conditioning the policy on the entire outcome-design history and avoiding explicit posterior computation. In that approach, called *Deep Adaptive Design* (DAD), there is an upfront cost to learning the policy, but experiments can be performed in real-time. While DAD is only applicable for exchangeable models, *implicit DAD* (iDAD, Ivanova et al., 2021) generalized the concept to accommodate non-exchangeable models that cover dynamical systems. These methods leverage a lower bound to the EIG known as the sequential Prior Contrastive Estimation (sPCE) bound. sPCE exhibits significant bias in low-sample regimes, thus requiring a large number of samples for accurate estimates of the EIG (Foster et al., 2021).

In this work, we introduce a novel amortization scheme that circumvents the drawbacks of sPCE. Our approach is rooted

[1]Aalto University, Espoo, Finland. Correspondence to: Sahel Iqbal <sahel.iqbal@aalto.fi>.

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

in understanding sequential Bayesian experimental design as an adaptive risk-sensitive decision-making process (Whittle, 1990). We demonstrate that risk-sensitive decision-making can be cast as an inference problem for an equivalent non-Markovian non-linear and non-Gaussian state-space model (Toussaint & Storkey, 2006; Rawlik, 2013). This insight motivates a novel nested sequential Monte Carlo (SMC) algorithm that jointly estimates the EIG and the corresponding optimal designs. We refer to this algorithm as *Inside-Out SMC*^{2}, due to its relation to the SMC^2 algorithm of Chopin et al. (2013). Finally, we embed our technique within a particle Markov chain Monte Carlo (pMCMC) algorithm to perform gradient-based optimization of the amortizing policy.

We validate our algorithm on a range of dynamical systems with long experiment sequences, highlighting the computational advantages of our proposed method compared to existing work. The code to reproduce our results is available at https://github.com/Sahel13/InsideOutSMC2.jl.

2. Problem Statement

We are interested in the sequential BED problem for non-exchangeable data, specifically dynamical systems. Accordingly, we assume a scenario of $T \in \mathbb{N}$ sequential experiments to infer a parameter vector θ, starting from a prior $p(\theta)$, a Markovian likelihood $f(x_{t+1} \mid x_t, \xi_t, \theta)$, and an initial distribution $p(x_0)$, where t indexes the experiment number. We further assume that the designs are sampled from a *stochastic* policy $\pi_\phi(\xi_t \mid z_{0:t})$ parameterized by ϕ, where we define $z_0 \coloneqq \{x_0\}$ and $z_t \coloneqq \{x_t, \xi_{t-1}\}$ for all $t \geq 1$, and denote the outcome-design history up to time t by $z_{0:t} \coloneqq \{x_{0:t}, \xi_{0:t-1}\}$. This yields the following factorization for the joint distribution of outcomes and designs:

$$\begin{aligned} p_\phi(z_{0:T} \mid \theta) &= p(z_0) \prod_{t=1}^{T} p_\phi(z_t \mid z_{0:t-1}, \theta) \\ &= p(x_0) \left\{ \prod_{t=1}^{T} f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \right\} \\ &\quad \times \left\{ \prod_{t=1}^{T} \pi_\phi(\xi_{t-1} \mid z_{0:t-1}) \right\}. \end{aligned} \tag{2}$$

In this setting, the expected information gain can be written analogously to that of the single experiment:

$$\mathcal{I}(\phi) \coloneqq \mathbb{E}_{p_\phi(z_{0:T})} \Big[\mathbb{H}\big[p(\theta)\big] - \mathbb{H}\big[p(\theta \mid z_{0:T})\big] \Big], \tag{3}$$

where $p_\phi(z_{0:T}) = \mathbb{E}_{p(\theta)}\big[p_\phi(z_{0:T} \mid \theta)\big]$. This definition corresponds to the *terminal reward* framework in literature (Foster, 2021, Section 1.8), as it compares the prior entropy of θ with the posterior entropy at the end of the experiment sequence. Note that the EIG in (3) is evaluated under the expectation of the marginal distribution $p_\phi(z_{0:T})$, including the stochastic design policy. The resulting experimental design objective corresponds to finding the optimal policy parameters $\phi^* \coloneqq \arg\max_\phi \mathcal{I}(\phi)$. The upcoming sections will present a novel interpretation of this objective in a sequential decision-making framework that leverages the duality with inference techniques to perform policy amortization.

3. Sequential Bayesian Experimental Design as Probabilistic Inference

To formulate sequential BED as an inference problem, we demonstrate a factorization of the EIG over time steps.

Proposition 1. *For models specified by the joint density in (2), the expected information gain factorizes to*

$$\mathcal{I}(\phi) = \mathbb{E}_{p_\phi(z_{0:T})} \left[\sum_{t=1}^{T} r_t(z_{0:t}) \right], \tag{4}$$

where $r_t(z_{0:t})$ is a stage reward defined as

$$r_t(z_{0:t}) = \alpha_t(z_{0:t}) + \beta_t(z_{0:t}), \tag{5}$$

with $\alpha_t(z_{0:t})$ and $\beta_t(z_{0:t})$ defined as

$$\alpha_t(z_{0:t}) = \int p(\theta \mid z_{0:t}) \log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \, \mathrm{d}\theta,$$

$$\beta_t(z_{0:t}) = -\log \int p(\theta \mid z_{0:t-1}) f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \, \mathrm{d}\theta.$$

Furthermore, for models with additive, constant noise in the dynamics, the EIG can be written as

$$\mathcal{I}(\phi) \equiv \mathbb{E}_{p_\phi(z_{0:T})} \left[\sum_{t=1}^{T} \beta_t(z_{0:t}) \right], \tag{6}$$

where '$\equiv$' denotes equality up to an additive constant.

The proof is given in Appendix A. Written in this form, the expected information gain resembles the expected total reward of a discrete-time, finite-horizon, non-Markovian decision-making problem (Puterman, 2014) with a stage reward $r_t(z_{0:t})$ that captures the information content regarding the unknown parameters θ. We will now use this factorization of $\mathcal{I}(\phi)$ to derive a risk-sensitive objective and a dual inference perspective, leading to a novel amortized BED learning scheme.

3.1. The Dual Inference Problem

To leverage the duality between risk-sensitive decision-making and inference, we follow the formulation of Toussaint & Storkey (2006) and Rawlik (2013), and introduce the potential function

$$g_t(z_{0:t}) \coloneqq \exp\Big\{ \eta \, r_t(z_{0:t}) \Big\}, \tag{7}$$

with $\eta \in \mathbb{R}_{>0}$. If we define the potential of an entire trajectory, $g_{1:T}(z_{0:T})$, to be the product of the potential functions over time steps, then

$$\log g_{1:T}(z_{0:T}) = \sum_{t=1}^{T} \log g_t(z_{0:t}) = \eta \left[\sum_{t=1}^{T} r_t(z_{0:t}) \right],$$

is the total reward of a trajectory scaled by η. In this context, the potentials $g_{1:T}$ play the role of an un-normalized pseudo-likelihood proportional to the probability of the trajectory $z_{0:T}$ being optimal (Dayan & Hinton, 1997). This perspective allows us to define a non-Markovian state-space model characterized, for $t = 0, \ldots, T$, by the following joint density

$$\Gamma_t(z_{0:t}; \phi) = \frac{1}{Z_t(\phi)} \, p(z_0) \prod_{s=1}^{t} p_\phi(z_s \mid z_{0:s-1}) \, g_s(z_{0:s}), \quad (8)$$

where $p_\phi(z_t \mid z_{0:t-1})$ are the marginal dynamics under the running filtered posterior $p(\theta \mid z_{0:t-1})$

$$p_\phi(z_t \mid z_{0:t-1}) = \int p_\phi(z_t \mid z_{0:t-1}, \theta) \, p(\theta \mid z_{0:t-1}) \, \mathrm{d}\theta, \quad (9)$$

and $Z_t(\phi)$ is the normalizing constant

$$Z_t(\phi) = \int g_{1:t}(z_{0:t}) \, p_\phi(z_{0:t}) \, \mathrm{d}z_{0:t}.$$

For ease of exposition, we will henceforth refer to $Z_T(\phi)$ as the marginal likelihood of *being optimal*, even though it may not represent any meaningful probability.

The duality principle becomes evident when we apply Jensen's inequality to show that the log marginal likelihood is an upper bound on the EIG scaled by η:

$$\begin{aligned} \log Z_T(\phi) &= \log \mathbb{E}_{p_\phi(z_{0:T})} \left[g_{1:T}(z_{0:T}) \right] \\ &\geq \mathbb{E}_{p_\phi(z_{0:T})} \left[\log g_{1:T}(z_{0:T}) \right] \\ &= \eta \, \mathcal{I}(\phi). \end{aligned}$$

Hence, maximizing the marginal likelihood is equivalent to maximizing a risk-sensitive EIG objective, which we denote as $\mathcal{I}_\eta(\phi)$ (Marcus et al., 1997; Rawlik, 2013),

$$\mathcal{I}_\eta(\phi) = \frac{1}{\eta} \log \mathbb{E}_{p_\phi(z_{0:T})} \left[\exp \left\{ \eta \sum_{t=1}^{T} r_t(z_{0:t}) \right\} \right]. \quad (11)$$

Note that η modulates the bias-variance trade-off of this objective. This aspect is revealed by considering a first-order expansion of the objective around $\eta = 0$

$$\mathcal{I}_\eta(\phi) \approx \mathbb{E} \left[\sum_{t=1}^{T} r_t(z_{0:t}) \right] + \frac{\eta}{2} \mathbb{V} \left[\sum_{t=1}^{T} r_t(z_{0:t}) \right],$$

where the expectation and variance operators are in relation to $p_\phi(z_{0:T})$. Note that in the limit $\eta \to 0$, we recover the risk-neutral EIG objective from (4).

The choice of a positive tempering parameter $\eta \in \mathbb{R}_{>0}$ leads to a risk-seeking objective that incentivizes exploration during policy amortization. This is compatible with the heuristic of *optimism in the face of uncertainty*, widely adopted in stochastic optimization settings (Neu & Pike-Burke, 2020).

In this section, we formalized the connection between a risk-sensitive sequential BED objective and inference in an equivalent non-Markovian state-space model. In the following, we leverage this insight to formulate a gradient-based policy optimization technique within a particle MCMC framework.

3.2. Amortization as Likelihood Maximization

The duality principle demonstrated in Section 3 enables us to view amortized BED from an inference-centric perspective and to frame policy optimization in terms of maximum likelihood estimation (MLE) in a non-Markovian, nonlinear, and non-Gaussian state-space model, as specified by (8). Following the literature on particle methods for MLE (Kantas et al., 2015), we employ a stochastic gradient ascent algorithm (Robbins & Monro, 1951).

To obtain the derivative of the log marginal likelihood, $\mathcal{S}(\phi) \coloneqq \nabla_\phi \log Z_T(\phi)$, also known as the score function, we make use of Fisher's identity (Cappé et al., 2005),

$$\begin{aligned} \mathcal{S}(\phi) &= \int \nabla_\phi \log \tilde{\Gamma}_T(z_{0:T}; \phi) \, \Gamma_T(z_{0:T}; \phi) \, \mathrm{d}z_{0:T} \\ &= \int \nabla_\phi \log p_\phi(z_{0:T}) \, \Gamma_T(z_{0:T}; \phi) \, \mathrm{d}z_{0:T}. \end{aligned}$$

where we define $\tilde{\Gamma}_T(z_{0:T}; \phi) \coloneqq p_\phi(z_{0:T}) \, g_{1:T}(z_{0:T})$ to be the un-normalized density from (8). This identity provides a Monte Carlo estimate of the score $\hat{\mathcal{S}}(\phi)$ given samples from $\Gamma_T(\cdot; \phi)$. It is well-known that computing this expectation naively by first sampling from $p_\phi(z_{0:T})$ and then weighting the samples by g results in very high variance estimates (see, e.g., in Doucet et al., 2009, Section 3.3). Alternatively, a lower variance estimate can be achieved by drawing approximate samples from $\Gamma_T(\cdot; \phi)$ via particle smoothing, yielding consistent, albeit biased, estimates of expectations under the smoothing distribution for a finite sample size (Chopin & Papaspiliopoulos, 2020, Chapter 12).

Another alternative is to use Markovian score climbing (MSC, Gu & Kong, 1998; Naesseth et al., 2020), see Algorithm 1. MSC uses a $\Gamma_T(\cdot; \phi)$-ergodic Markov chain Monte Carlo (MCMC, see, e.g., Brooks et al., 2011, for a review and definition) kernel, $\mathcal{K}_\phi(\cdot \mid z_{0:T})$, to compute a Monte Carlo estimate of the score. Contrary to simply using particle smoother approximations within a gradient ascent procedure (see Kantas et al., 2015, Section 5), Algorithm 1 is guaranteed to converge to a local optimum of the marginal likelihood (Naesseth et al., 2020, Proposition 1).

In this work, we construct the MCMC kernel $\mathcal{K}_\phi$ as a variant of the conditional sequential Monte Carlo (CSMC) ker-

Algorithm 1 Markovian score climbing

input Initial trajectory $z_{0:T}^0$, initial parameters ϕ_0, step size sequence $\{\gamma_i\}_{i\in\mathbb{N}}$, Markov kernel $\mathcal{K}$.
output Local optimum ϕ^* of the marginal likelihood.

$k \leftarrow 1$
while not converged **do**
  Sample $z_{0:T}^k \sim \mathcal{K}_{\phi_{k-1}}(\cdot \mid z_{0:T}^{k-1})$
  Compute $\hat{\mathcal{S}}(\phi_{k-1}) \leftarrow \nabla_\phi \log p_\phi(z_{0:T}^k)|_{\phi=\phi_{k-1}}$
  Update $\phi_k \leftarrow \phi_{k-1} + \gamma_k \hat{\mathcal{S}}(\phi_{k-1})$
  $k \leftarrow k+1$
end while
return ϕ_k

nel (Andrieu et al., 2010), namely the Rao–Blackwellized CSMC kernel (Olsson & Westerborn, 2017; Cardoso et al., 2023; Abdulsamad et al., 2023). In practice, CSMC (as well as its Rao–Blackwellized modifications aforementioned) can be implemented as a simple modification to a particle filter representation of the smoothing distribution (Kitagawa, 1996, Section 4.1). In the next section, we describe how such a particle filter can be implemented for (8), and, for the sake of brevity, we defer the full description of the conditional version to Appendix C.4.

4. Inside-Out SMC^2

4.1. Approximating the Filtered Posterior

A *bootstrap* particle filter samples particles from the transition density and weights them using the potential function (Chopin & Papaspiliopoulos, 2020, Section 10.3). In our non-Markovian model, this would imply sampling from the marginal dynamics $p_\phi(z_t \mid z_{0:t-1})$ in (9), and evaluating the potential function $g_t(z_{0:t})$ in (7). Both steps require computing the filtered posterior $p(\theta \mid z_{0:t-1})$. Fortunately, for models of the form

$$\theta \sim p(\theta), \quad z_t \sim p_\phi(z_t \mid z_{0:t-1}, \theta), \quad t \geq 1,$$

the iterated batch importance sampling (IBIS) algorithm of Chopin (2002) can be used to generate weighted Monte Carlo samples $\{\theta_t^m\}_{m=1}^M =: \theta_t^{1:M}$ that are approximately distributed according to $p(\theta \mid z_{0:t})$ at each time step using a specialized particle filtering procedure. We summarize a single step of the method in Algorithm 2.

In Algorithm 2, $\mathcal{M}(W^{1:M})$ denotes multinomial sampling using the normalized weights $W^{1:M}$, and Q_t is a $p(\theta \mid z_{0:t})$-ergodic Markov chain Monte Carlo kernel. If a degeneracy criterion is met, IBIS employs a resample-move step (lines 4–6 in Algorithm 2, see Gilks & Berzuini, 2001) to rejuvenate particles using the Markov kernel Q_t. A standard degeneracy measure, which we use in this work, is given by the effective sample size (ESS) (Liu & Chen, 1995) of the particle representation computed as $\text{ESS} = 1/\sum_{m=1}^M (W^m)^2$.

Algorithm 2 Single step of IBIS

notation Any operation with superscript m is to be understood as performed for all $m = 1, \dots, M$.
function IBIS_STEP($z_{0:t}, \theta^{1:M}, W^{1:M}$)
1: Compute $v_t(\theta^m) = p_\phi(z_t \mid z_{0:t-1}, \theta^m)$.
2: Reweight: $W^m \propto W^m v_t(\theta^m)$.
3: **if** some degeneracy criterion is fulfilled **then**
4: Resample: $a_t^m \sim \mathcal{M}(W^{1:M})$.
5: Move: $\tilde{\theta}^m \sim Q_t(\theta^{a_t^m}, \cdot)$.
6: Replace the current set of weighted particles with $(\theta^m, W^m) \leftarrow (\tilde{\theta}^m, 1/M)$.
7: **end if**
8: **return** $\{\theta^m, W^m\}_{m=1}^M$
end function

The ESS roughly corresponds to the number of equivalent independent samples one would need to compute integrals with the same precision. The resample-move step is then triggered if the ESS falls below a chosen fraction of the total particles M, taken to be 75% in this work. Details on the choice of the Markov kernel Q_t are given in Appendix C.2.

4.2. The Inside-Out SMC^2 Algorithm

Chopin (2004) showed that, for integrable functions ψ, $\sum_{m=1}^M W^m \psi(\theta^m)$ is a consistent and asymptotically (as $M \to \infty$) normal estimator of the integral $\int \psi(\theta)\, p(\theta \mid z_{0:t})\, \mathrm{d}z_{0:t}$. Therefore, a natural solution to perform inference in (8) is to use IBIS within a standard particle filter. This idea is similar to the SMC^2 algorithm of Chopin et al. (2013) which can be seen as a particle filter within IBIS targeting the distribution $p(\theta \mid y_{1:t})$ for a state-space model with noisy observations $y_{1:t}$. We thus call our algorithm *Inside-Out SMC^2*, which we reproduce in Algorithm 3, with N and M denoting the numbers of samples of z and θ respectively. Any operation therein with superscripts m or n is to be understood as performed for every $m = 1, \dots, M$ and $n = 1, \dots, N$, and we use an upper script dot $u^{\bullet n}$ to denote the collection $\{u^{mn}\}_{m=1}^M$.

At a time step t and given a trajectory $z_{0:t}^n$, IBIS approximates the θ-posterior with weighted samples $\{\theta_t^{\bullet n}, W_{t,\theta}^{\bullet n}\}$

$$p(\theta \mid z_{0:t}^n) \approx \hat{p}(\theta \mid z_{0:t}^n) \coloneqq \sum_{m=1}^M W_{t,\theta}^{mn}\, \delta_{\theta_t^{mn}}(\theta),$$

where δ is the Dirac delta function. We can then form an approximation to the marginal dynamics as follows

$$\begin{aligned}\hat{p}(x_{t+1} \mid z_{0:t}^n, \xi_t) &= \int f(x_{t+1} \mid x_t^n, \xi_t, \theta)\, \hat{p}(\theta \mid z_{0:t}^n)\, \mathrm{d}\theta \\ &= \sum_{m=1}^M W_{t,\theta}^{mn}\, f(x_{t+1} \mid x_t^n, \xi_t, \theta_t^{mn}),\end{aligned}$$

and consequently, for the augmented state as

$$\hat{p}_\phi(z_{t+1} \mid z_{0:t}^n) = \hat{p}(x_{t+1} \mid z_{0:t}^n, \xi_t)\, \pi_\phi(\xi_t \mid z_{0:t}^n).$$

Algorithm 3 Inside-Out SMC^2

1: Sample $z_0^n \sim p(z_0)$, $\theta_0^{mn} \sim p(\theta)$, set $W_{0,\theta}^{mn} \leftarrow 1/M$.
2: Sample $z_1^n \sim \hat{p}_\phi(\cdot \mid z_0^n)$ initialize the state history $z_{0:1}^n \leftarrow (z_0^n, z_1^n)$.
3: Compute and normalize the weights

$$W_z^n \propto \exp\Big\{ -\eta \log \hat{p}(x_1^n \mid x_0^n, \xi_0^n)\Big\}.$$

4: **for** $t \leftarrow 1, \dots, T-1$ **do**
5: Sample $b_t^n \sim \mathcal{M}(W_z^{1:N})$.
6: $\theta_t^{\bullet n}, W_{t,\theta}^{\bullet n} \leftarrow$ IBIS_STEP$(z_{0:t}^{b_t^n}, \theta_{t-1}^{\bullet b_t^n}, W_{t-1,\theta}^{\bullet b_t^n})$
7: Sample $z_{t+1}^n \sim \hat{p}_\phi(\cdot \mid z_{0:t}^{b_t^n})$, and append to state history $z_{0:t+1}^n \leftarrow [z_{0:t}^{b_t^n}, z_{t+1}^n]$.
8: Compute and normalize the weights

$$W_z^n \propto \exp\left\{-\eta \log \hat{p}(x_{t+1}^n \mid z_{0:t}^{b_t^n}, \xi_t^n)\right\}.$$

9: **end for**
10: **return** $\{z_{0:T}^n, W_z^n\}_{n=1}^N$.

If the Markovian density $f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)$ is conditionally linear in the parameters and conjugate to the prior $p(\theta)$, we can compute the posterior in closed form, and therefore the marginal dynamics as well. In this case, we do not need IBIS, and this significantly reduces the computational complexity of Algorithm 3 (see Appendix C.3).

Note that, for the sake of clarity, in Algorithm 3 and in its analysis in Section 4.3, we consider the case when the dynamics have constant noise, corresponding to the stage reward in (6). The modification to Algorithm 3 for the more general case is straightforward, only requiring a modified weight function, as detailed in Section C.1.

4.3. Target Distribution of Inside-Out SMC^2

We now show that the nested particle filter introduced in the previous section asymptotically targets the correct distribution. Similarly to Chopin et al. (2013, Proposition 1), Algorithm 3 is a particle filter targeting a particle filter. Indeed, dropping the n indices and the explicit dependence on ϕ, let $\Gamma_t^M(z_{0:t}, \theta_{0:t-1}^{1:M}, a_{1:t-1}^{1:M})$ denote the distribution of all stochastic variables generated by an instance of the inner IBIS at line 6 in Algorithm 3. We first note that

$$\Gamma_0^M(z_0) = p(z_0), \quad \Gamma_0^M(z_0, \theta_0^{1:M}) = p(z_0) \prod_{m=1}^M p(\theta_0^m).$$

Let us break down the ratio of the distributions over successive iterations as

$$\frac{\Gamma_{t+1}^M(z_{0:t+1}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t-1}^{1:M}, a_{1:t-1}^{1:M})} = \frac{\Gamma_{t+1}^M(z_{0:t+1}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})} \times \frac{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t-1}^{1:M}, a_{1:t-1}^{1:M})}. \quad (12)$$

The second fraction in (12) the IBIS rejuvenation step:

$$\frac{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t-1}^{1:M}, a_{1:t-1}^{1:M})} = \prod_{m=1}^M W_{t,\theta}^{a_t^m} Q_t(\theta_{t-1}^{a_t^m}, \theta_t^m), \quad (13)$$

where the normalized weights (of the θ particles) $W_{t,\theta}^m$ are

$$W_{t,\theta}^m = \frac{v_t^m}{\sum_{m=1}^M v_t^m}, \quad v_t^m = p_\phi(z_t \mid z_{0:t-1}, \theta_{t-1}^m),$$

and we have assumed multinomial resampling at every time step. The first fraction in (12) is the trajectory update step:

$$\frac{\Gamma_{t+1}^M(z_{0:t+1}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})} \propto \frac{1}{M} \sum_{m=1}^M p_\phi(z_{t+1} \mid z_{0:t}, \theta_t^m) \times \exp\left\{-\eta \log \frac{1}{M} \sum_{m=1}^M f(x_{t+1} \mid x_t, \xi_t, \theta_t^m)\right\}. \quad (14)$$

The following proposition, akin to the law of large numbers and proven in Appendix B, ensures that Algorithm 3 asymptotically targets the correct distribution.

Proposition 2 (Consistency of the target distribution)**.** *Let* $\tilde{\Gamma}_t^M(z_{0:t}, \theta_t) = \mathbb{E}\left[\Gamma_t^M(z_{0:t}, \theta_t^{1:M}, a_{1:t}^{1:M})\right]$ *be the joint expected empirical distribution*[1] *over* $(z_{0:t}, \theta_t)$ *taken by integrating over* $a_{1:t}^{1:M}$ *and* $\theta_{0:t}^{1:M}$*. Under technical conditions listed in Appendix B, as* $M \to \infty$*, empirical expectations under* $\tilde{\Gamma}_t^M$ *converge to expectations under* $\Gamma_t(z_{0:t}, \theta_t) := \Gamma_t(z_{0:t})\, p(\theta_t \mid z_{0:t})$*. That is, for any bounded test function* $\psi(z_{0:t}, \theta_t)$*, we have*

$$\mathbb{E}_{\tilde{\Gamma}_t^M}\left[\psi(z_{0:t}, \theta_t)\right] \to \mathbb{E}_{\Gamma_t}\left[\psi(z_{0:t}, \theta_t)\right]$$

almost surely.

5. Related Work

Foster et al. (2021) propose optimizing the following *sequential Prior Contrastive Estimation* (sPCE) lower bound to the expected information gain

$$\mathcal{L}_T^{\text{sPCE}}(\phi, L) = \mathbb{E}_{p_\phi(\theta_0, z_{0:T})p(\theta_{1:L})}\big[g_L(\theta_{0:L}, z_{0:T})\big], \quad (15)$$

where

$$g_L(\theta_{0:L}, z_{0:T}) = \log \frac{p_\phi(z_{0:T} \mid \theta_0)}{\frac{1}{L+1}\sum_{\ell=0}^L p_\phi(z_{0:T} \mid \theta_\ell)}.$$

A naive nested Monte Carlo estimator would exclude θ_0 from the denominator, but such an estimator would have a large variance, particularly for long experiment sequences ($T \gg 1$). By including θ_0 in the estimate for $p_\phi(z_{0:T})$, the authors show that g_L is upper bounded by $\log(L+1)$, where L is the number of regularizing or *contrastive* samples, resulting in a low variance estimator amenable to optimization (Poole et al., 2019). However, this

[1] Note that $\Gamma_t^M(z_{0:t}, \theta_t^{1:M}, a_{1:t}^{1:M})$ is a random measure and hence this expectation is a measure.

bound also implies that one needs to take an exponentially large value of L to avoid being restricted by the upper bound which, depending on the true value of the EIG, may introduce significant bias. Our Inside-Out SMC^2 algorithm does not suffer from this particular drawback and only requires a small number of θ particles to provide an estimate of the EIG. This is achieved by leveraging the sequential structure of the experiments to maintain a running posterior over θ instead of relying on samples from the prior.

Drovandi et al. (2013), Drovandi et al. (2014), and Moffat et al. (2020) have previously proposed the use of IBIS for experimental design, albeit in the context of exchangeable data. In these approaches, IBIS is used solely to track the parameter posterior given the past experiments. Most importantly, their experiment design is myopic, only optimizing for the next experiment. In contrast, we embed IBIS into a general particle smoothing algorithm, where IBIS is used to approximate the marginalized dynamics. This formulation enables our technique to optimize across the entire horizon of experiments and lends itself to amortization.

Our work also shares similarities in formulation with that of Blau et al. (2022) which formulates the sequential BED problem as a hidden parameter Markov decision process (Doshi-Velez & Konidaris, 2016). Unlike our work, they adopt a reinforcement learning approach that applies only to exchangeable models and optimize the sPCE bound.

In the context of SMC, a related approach to ours is given by Wigren et al. (2019), who also propagate the filtering posterior for the parameter at hand to perform parameter-marginalized inference. However, they are not interested in experimental design, and in contrast to our Inside-Out SMC^2, they only need to compute the *marginal* posterior distribution, whereas we reuse the *pathwise* posterior distribution to compute our potential function g. Nonetheless, we believe the two approaches are related and could be combined in future work.

6. Empirical Evaluation

For the empirical evaluation of our method, we consider the setting of designing 50 sequential experiments to identify the parameters of input (design) dependent dynamical systems. The systems we examine are abstractions of robotic systems with dynamics that are described by stochastic differential equations and discretized using the Euler–Maruyama method (see, e.g., Särkkä & Svensson, 2023, Section 4.3). Hence, all scenarios involve likelihoods that are non-exchangeable conditionally Markovian densities. Moreover, these dynamical systems enforce input constraints that restrict the design space, making the sequential experimental design problem challenging.

We construct the stochastic policy π_ϕ using a mean function m_ϕ parameterized by a gated recurrent unit architecture (GRU, Cho et al., 2014), with an additional learnable parameter Σ_ϕ for the variance. Then the stochastic policy π_ϕ is constructed as the law of the random variable $\xi_t = a \cdot \tanh(s_t) + b$, where $s_t \sim \mathcal{N}(m_\phi(z_{0:t}), \Sigma_\phi)$. Here (a, b) are scale and shift parameters that reflect the design constraints. Exhaustive details of the network architecture and hyperparameters are given in Section D.

We compare our algorithm, Inside-Out SMC^2 (IO-SMC^2), to a few different types of design policies. We include two simple baselines - a random policy, which samples designs from a uniform distribution, and a pseudo-random binary signal (PRBS) policy, which randomly chooses between the upper and lower design limits. Additionally, we evaluate a myopic, non-amortized method, that corresponds to IO-SMC^2 with a one-step look-ahead. This leads to a greedy, sub-optimal design that optimizes only for the next experiment. This approach is comparable to Drovandi et al. (2013). Finally, we compare to iDAD (Ivanova et al., 2021) trained on the sPCE lower bound. iDAD, unlike DAD, can accommodate non-exchangeable sequential data. Although iDAD was originally developed for implicit models, we provide access to the conditional transition densities in our experiments to guarantee a fair comparison.

As evaluation metrics, we report both the sPCE bound and a nested Monte Carlo estimate of the *risk-neutral* EIG in (4). We choose to report this metric instead of the *risk-sensitive* EIG from (11) in order to maintain a consistent empirical comparison with the sPCE bound, which itself does not account for risk. We compute this Monte Carlo estimate of the EIG under equally weighted sample trajectories $\{z_{0:T}^n\}_{n=1}^N$ from the marginal $p_\phi(z_{0:T})$ as follows

$$\mathcal{I}(\phi) = \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} r_t(z_{0:t})\right] \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} \hat{r}_t(z_{0:t}^n). \quad (16)$$

Here, $\hat{r}_t(z_{0:t}^n)$ is itself a particle approximation of the true stage reward in (5) obtained using the filtering posterior provided by IBIS. The samples $\{z_{0:T}^n\}_{n=1}^N$ are drawn by running Algorithm 3, while setting $b_t^n = n$ in line 5. This leads to a routine that generates trajectories from the marginal distribution $p_\phi(z_{0:T})$.

For all experiments, the EIG estimate in (16) was computed at evaluation time for $T = 50$, $N = 16$ and $M = 1024$, while the sPCE bound in (15) was computed using 16 outer samples and $L = 10^6$ regularizing samples. The statistics of these estimates are computed for 25 seeds. Further experimental details can be found in Appendix D.

6.1. Stochastic Pendulum

We consider two different representations of the stochastic dynamics of a compound pendulum. The aim is to infer a

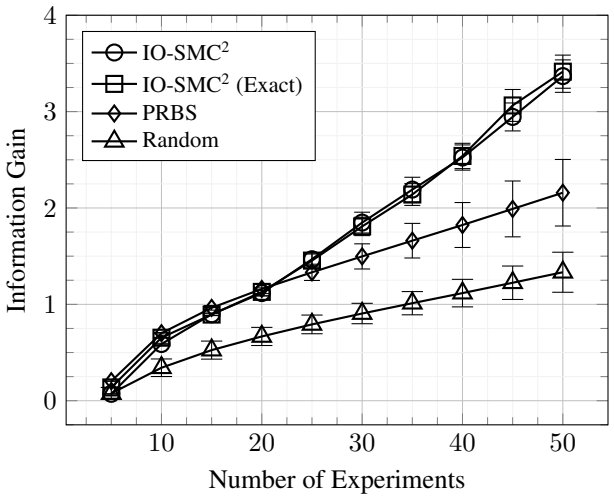

Figure 1. Accumulation of the information gain computed in closed form for different policies on the conditionally linear stochastic pendulum with a Gaussian prior. We report the mean and standard deviation over 512 realizations.

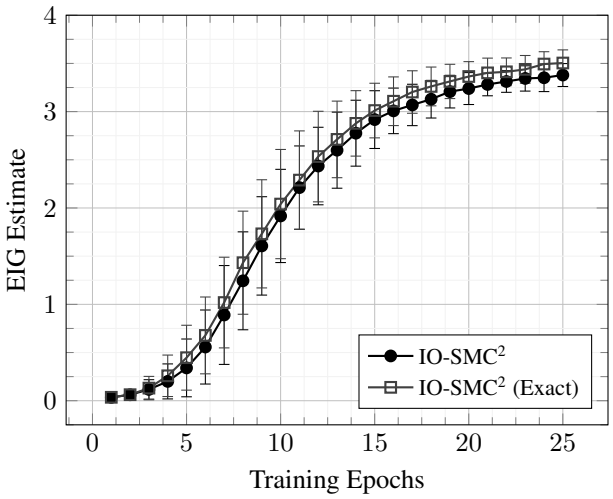

Figure 2. Training progression of the IO-SMC^2 policy and its exact variant on the conditionally linear stochastic pendulum. At every epoch, we evaluate the EIG estimate using the mean policy. We report the mean and standard deviation over 25 seeds.

vector of parameters that combines the mass and length of the pendulum by observing a sequence of states, comprised of its angular position and velocity. The design is the torque applied as input to the system.

6.1.1. Conditionally Linear formulation

First, we consider a conditionally linear formulation of the dynamics of the compound pendulum, see Appendix D.2.1. In conjunction with a Gaussian prior over the parameters, this setting allows us to compare IO-SMC^2 with exact posterior computation against the approximate posteriors delivered by IBIS. Details on exact posterior inference can be found in Appendix C.3.

In this conjugate setting, the information gain can be computed in closed form, given observations from sequential experiments. Figure 1 depicts the mean and standard deviation of the IG over experiments for different policies, obtained by simulating 512 sets of sequential experiments for different samples from the parameter prior. The amortized policies are superior to the random and PRBS policies. These results corroborate Table 1, which reports EIG estimates and sPCE bounds for all considered policies. The two variants of our algorithm outperform all considered baselines on both metrics.

To evaluate the stability of our algorithm during training, we trained 25 different policy networks using IO-SMC^2 and its exact variant. The means and standard deviations of EIG estimates obtained after each training epoch are depicted in Figure 2. Notice that the standard deviation shrinks over training epochs, reaching a final value comparable to that observed in EIG estimates for a single policy (Table 1), implying consistency across training runs.

Table 1. EIG estimates and sPCE lower bounds on the conditionally linear pendulum experiment for the considered methods. We report the mean $\pm$ standard deviation over 25 seeds.

Policy	EIG Estimate (16)	sPCE
Random	1.37 ± 0.08	1.44 ± 0.35
Myopic	1.45 ± 0.12	1.41 ± 0.32
PRBS	2.24 ± 0.19	2.33 ± 0.32
iDAD	2.58 ± 0.17	2.53 ± 0.35
IO-SMC^2	3.53 ± 0.15	3.66 ± 0.44
IO-SMC^2 (Exact)	3.63 ± 0.18	3.64 ± 0.41

Figure 1 and Figure 2 demonstrate that the two versions of IO-SMC^2 deliver comparable performance during learning and inference, empirically validating the use of IBIS in general non-conjugate settings.

6.1.2. Nonlinear formulation

We now consider the standard, nonlinear version of the stochastic pendulum. We choose a log-normal prior for the parameters, ensuring that the mass and length of the pendulum can only take positive values, see Appendix D.2.2. As a result, the exact version of our algorithm is no longer applicable, and we have to use IO-SMC^2 in its general form.

EIG estimates and sPCE lower bounds for different policies in this environment are reported in Table 2. The policy trained using IO-SMC^2 outperforms all baseline methods. An example of a trajectory generated by the mean of the trained policy is given in Figure 3. The policy has learned to swing the pendulum to achieve greater and greater angular velocities by alternating between maximum positive and negative designs, thus exploring more of the phase space.

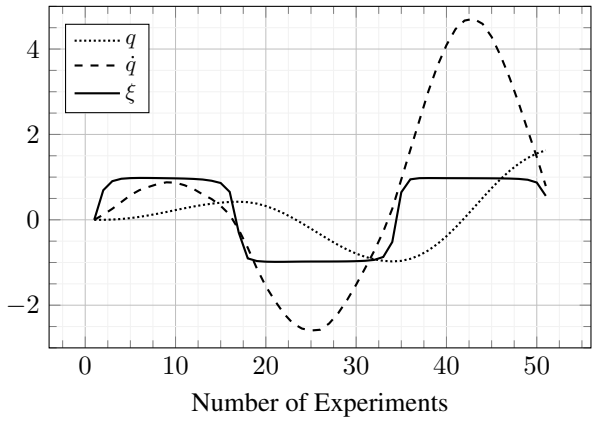

Figure 3. A sample experiment trajectory generated by the amortized policy during deployment on the nonlinear stochastic pendulum environment. q is the angle of the pendulum from the vertical, $\dot{q}$ is the angular velocity and ξ is the design.

Table 2. EIG estimates and sPCE lower bounds on the nonlinear pendulum experiment for the considered methods. We report the mean $\pm$ standard deviation over 25 seeds.

Policy	EIG Estimate (16)	sPCE
Random	2.12 ± 0.21	2.28 ± 0.25
Myopic	2.15 ± 0.18	2.27 ± 0.32
PRBS	3.00 ± 0.20	2.94 ± 0.33
iDAD	3.01 ± 0.29	3.18 ± 0.41
IO-SMC^2	3.72 ± 0.17	3.77 ± 0.38

6.2. Stochastic Cart-Pole

We now consider a cart-pole system with additive noise in the acceleration of the cart. The unknown parameters are the masses of the cart and pole and the length of the pole. The outcome of every experiment is an observation of the cart-pole state consisting of the position and velocity of the cart and the angular position and angular velocity of the pole. The design is the force applied to the cart at discrete intervals. The prior on the parameters is again a log-normal distribution. Further details on the experimental setup are given in Appendix D.3.

Table 3 reports the performance of each policy under consideration. The policy trained using IO-SMC^2 achieves the highest mean EIG estimate at 21.23. This experiment demonstrates the primary drawback of the sPCE bound; one needs approximately 1.3 billion regularizing samples to yield an sPCE bound of 21. To accommodate this number of samples, the implementation of Ivanova et al. (2021), which relies on parallel computation, would require about 660 GB of memory with double-precision floats. In contrast, IO-SMC^2 requires just 1024 inner samples to obtain our estimate in Table 3. Thus, for sequential experimental design problems with high EIG values, the advantage of IO-SMC^2 is clear. The sample efficiency of our EIG estimator

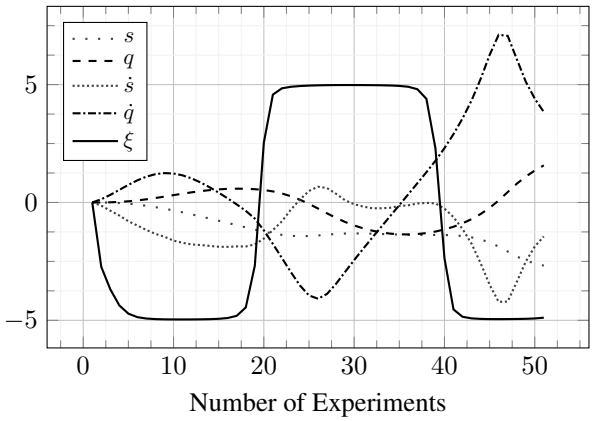

Figure 4. A sample experiment trajectory generated by the policy during deployment on the stochastic cart-pole environment. Here, s and $\dot{s}$ are the position and velocity of the cart respectively, q is the angle of the pole, $\dot{q}$ is its angular velocity and ξ is the design.

Table 3. EIG estimates and sPCE lower bounds on the stochastic cart-pole experiment for the considered methods. We report the mean $\pm$ standard deviation over 25 seeds. sPCE bounds for all policies hit the upper bound of $\log(10^6) \approx 13.82$, and precise estimates would need at least $\exp(21) \approx 1.3$ billion regularizing samples, far beyond our hardware limits.

Policy	EIG Estimate (16)	sPCE
Random	16.81 ± 0.77	13.72 ± 0.08
Myopic	16.53 ± 0.71	13.74 ± 0.10
PRBS	18.28 ± 0.50	13.80 ± 0.03
iDAD	18.99 ± 0.68	13.81 ± 0.01
IO-SMC^2	21.23 ± 0.62	13.82 ± 0.00

compared to sPCE is further demonstrated in Table 10 in Appendix D.3. Nevertheless, we note that the sPCE lower bound is still valuable for training despite its bias. Indeed, although iDAD underperforms compared to IO-SMC^2, it achieves a higher EIG at evaluation time than the expected upper bound of approximately 10, corresponding to the 2^{14} regularizing samples used at training time.

A sample trajectory generated by the trained policy mean is depicted in Figure 4. As in the case of the pendulum, the policy has learned to alternate between the design limits to explore the phase space efficiently.

6.3. Stochastic Double-Link

Our final experiment uses a stochastic double-link (double pendulum) system with the four unknowns being the masses and lengths of both links. The design is two-dimensional, corresponding to the torque applied at each of the two actuated joints. The dynamical equations and policy hyperparameters are given in Appendix D.4. Figure 5 plots a sample trajectory generated by a policy trained using our algorithm. The optimal policy implements two coordinated switches in

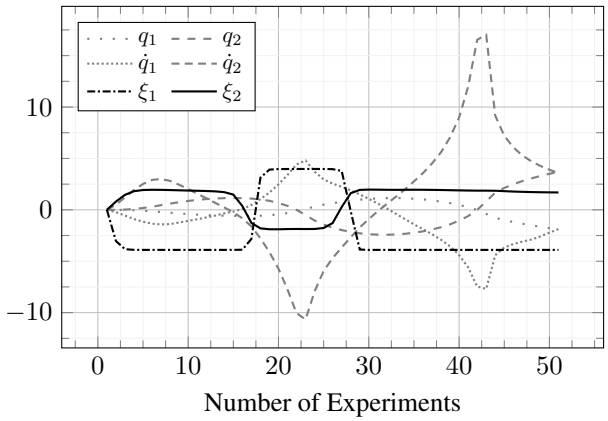

Figure 5. A sample experiment trajectory generated by the policy during deployment on the stochastic double-link environment. q_1 and q_2 are the angles from the vertical for the two links, $\dot{q}_1$ and $\dot{q}_2$ their respective angular velocities, and ξ_1 and ξ_2 are the designs.

Table 4. EIG estimates and sPCE lower bounds on the stochastic double-link experiment for the considered methods. We report the mean $\pm$ standard deviation over 25 seeds.

Policy	EIG Estimate (16)	sPCE
Random	7.81 ± 0.40	7.79 ± 0.43
Myopic	8.00 ± 0.44	8.13 ± 0.50
PRBS	5.25 ± 0.26	5.12 ± 0.42
iDAD	11.73 ± 0.45	11.52 ± 0.36
IO-SMC^2	11.53 ± 0.49	11.45 ± 0.42

the design dimensions to bring the system into informative states. In this experiment, iDAD also learns a similar policy, as reflected in the EIG estimates in Table 4.

7. Discussion and Limitations

We have introduced a novel method of amortized sequential Bayesian experimental design, taking inspiration from the control-as-inference framework. We cast the optimization of sequential designs as a smoothing problem in a non-Markovian state-space model. To perform inference in this model, we developed a novel nested particle filtering algorithm, which we call Inside-Out SMC^2. Our approach naturally lends itself to amortization via likelihood optimization in the form of Markovian score climbing. While we have used Inside-Out SMC^2 in the context of sequential BED, we believe it may find uses in other settings, where one wishes to obtain pathwise smoothing trajectories under parameter-marginalized models.

Our experimental evaluation shows that our approach holds promise as a generic and efficient way to learn experimental design policies: it is amortized, non-myopic, widely applicable, and easy to train. Our learned policies outperform the main alternative (iDAD, Ivanova et al., 2021) while only requiring a fraction of the number of samples at both training and evaluation time. In particular, the sPCE bound, used therein, requires the number of samples to grow exponentially with the maximal EIG value. This can make it unsuitable as a learning objective in certain dynamical systems. On the contrary, IO-SMC^2 can compute approximations to the EIG, no matter its value, with a relatively small number of particles, making it a better-behaved and more viable alternative.

One limitation of our approach is the requirement to evaluate the conditional transition densities in closed form. Thus, our method is unsuitable for sequential experimental design problems in dynamical models with intractable densities; for instance, choosing optimal measurement times in a compartmental epidemic model modeled as a Markov jump process (Whitehouse et al., 2023).

Time complexity. IO-SMC^2 has a time complexity of $\mathcal{O}(NMT^2)$. CSMC requires $N \propto T$ samples to be stable (Lindsten et al., 2015, Proposition 5) for an increasing number of time steps, making our algorithm $\mathcal{O}(MT^3)$ when accounting for statistical stability. The outer SMC loop can be trivially parallelized, reducing it to $\mathcal{O}(MT^2)$.

Scalability. SMC samplers, which IBIS is an instance of, are known to scale well with the dimension (Chenguang Dai & Whiteley, 2022). Since the random walk Metropolis kernel, which we use within IBIS, scales reasonably well with the dimensionality of the problem (Gelman et al., 1997), we expect our algorithm to scale well in the parameter dimension. However, scaling in the number of state dimensions might be more problematic, as we are using a bootstrap proposal, known to degenerate when the dimension of the observations is large. This is related to the informativeness of the potential function, which in our case can be regulated by controlling the tempering parameter η.

Tempering. The choice of the tempering parameter η remains an open research question, that touches on the interaction between the optimism in the policy amortization step and the variance of the particle filtering weights within IO-SMC^2. Addressing this issue requires solving a bias-variance trade-off that is inherent to risk-sensitive objectives. In principle, the most favorable outcome would be to obtain an inference problem that can scale to any value of η.

Individual Contributions

The original idea for the article was conceived by HA and developed jointly with AC and SI. SI developed and implemented Inside-Out SMC^2 with AC's guidance, and the proof of its consistency is due to AC. The experiments are due equally to HA and SI. HA supervised the project. SI wrote the initial manuscript, and all authors contributed to revisions. SS reviewed and validated the technical details.

Acknowledgements

SI gratefully acknowledges funding from the Research Council of Finland. HA acknowledges funding by the Finnish Center for Artificial Intelligence (FCAI).

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 of which we feel must be specifically highlighted here.

References

Abdulsamad, H., Iqbal, S., Corenflos, A., and Särkkä, S. Risk-sensitive stochastic optimal control as Rao-Blackwellized Markovian score climbing. *arXiv preprint arXiv:2312.14000*, 2023.

Andrieu, C., Doucet, A., and Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

Belousov, B., Abdulsamad, H., Schultheis, M., and Peters, J. Belief space model predictive control for approximately optimal system identification. In *Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2019.

Bickford Smith, F., Kirsch, A., Farquhar, S., Gal, Y., Foster, A., and Rainforth, T. Prediction-oriented Bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Blau, T., Bonilla, E. V., Chades, I., and Dezfouli, A. Optimizing sequential experimental design with deep reinforcement learning. In *International Conference on Machine Learning*, 2022.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

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

Cardoso, G., Janati El Idrissi, Y., Le Corff, S., Moulines, E., and Olsson, J. State and parameter learning with PaRIS particle Gibbs. In *International Conference on Machine Learning*, volume 202, pp. 3625–3675. PMLR, 2023.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

Chenguang Dai, Jeremy Heng, P. E. J. and Whiteley, N. An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, 117(539):1587–1600, 2022.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chopin, N. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

Chopin, N. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

Chopin, N. and Papaspiliopoulos, O. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. SMC^2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(3):397–426, 2013.

Dayan, P. and Hinton, G. E. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Doshi-Velez, F. and Konidaris, G. Hidden parameter Markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *International Joint Conference on Artificial Intelligence*, 2016.

Doucet, A., Johansen, A. M., et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

Drovandi, C. C., McGree, J. M., and Pettitt, A. N. Sequential Monte Carlo for Bayesian sequentially designed experiments for discrete data. *Computational Statistics & Data Analysis*, 57(1):320–335, 2013.

Drovandi, C. C., McGree, J. M., and Pettitt, A. N. A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design. *Journal of Computational and Graphical Statistics*, 23(1):3–24, 2014.

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. Deep adaptive design: Amortizing sequential Bayesian experimental design. In *International Conference on Machine Learning*, 2021.

Foster, A. E. *Variational, Monte Carlo and Policy-Based Approaches to Bayesian Experimental Design*. PhD thesis, University of Oxford, 2021.

Gelman, A., Gilks, W. R., and Roberts, G. O. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110 – 120, 1997.

Gilks, W. R. and Berzuini, C. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146, 2001.

Gu, M. G. and Kong, F. H. A stochastic approximation algorithm with Markov chain Monte-Carlo method for incomplete data estimation problems. *Proceedings of the National Academy of Sciences*, 95(13):7270–7274, 1998.

Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1): 97–109, 1970.

Huan, X. and Marzouk, Y. M. Sequential Bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320*, 2016.

Ivanova, D. R., Foster, A., Kleinegesse, S., Gutmann, M., and Rainforth, T. Implicit deep adaptive design: Policy–based experimental design without likelihoods. In *Advances in Neural Information Processing Systems*, volume 34, pp. 25785–25798, 2021.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3), 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kitagawa, G. Monte Carlo filter and smoother for non-Gaussian nonlinear state-space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

Kueck, H., Hoffman, M., Doucet, A., and de Freitas, N. Inference and learning for active sensing, experimental design and control. In Araujo, H., Mendonça, A. M., Pinho, A. J., and Torres, M. I. (eds.), *Pattern Recognition and Image Analysis*, pp. 1–10, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

Lindley, D. V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27 (4):986–1005, 1956.

Lindsten, F., Douc, R., and Moulines, E. Uniform ergodicity of the particle Gibbs sampler. *Scandinavian Journal of Statistics*, 42(3):775–797, 2015.

Liu, J. S. and Chen, R. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.

Marcus, S. I., Fernández-Gaucherand, E., Hernández-Hernandez, D., Coraluppi, S., and Fard, P. Risk sensitive Markov decision processes. In *Systems and Control in the Twenty-First Century*, pp. 263–279, Boston, MA, 1997. Birkhäuser Boston.

McMichael, R. D., Dushenko, S., and Blakley, S. M. Sequential Bayesian experiment design for adaptive Ramsey sequence measurements. *Journal of Applied Physics*, 130 (14):144401, 2021.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Moffat, H., Hainy, M., Papanikolaou, N. E., and Drovandi, C. Sequential experimental design for predator–prey functional response experiments. *Journal of the Royal Society, Interface*, 17(166):20200156, 2020.

Myung, J. I., Cavagnaro, D. R., and Pitt, M. A. A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, 57(3):53–67, 2013.

Naesseth, C., Lindsten, F., and Blei, D. Markovian score climbing: Variational inference with KL(P||Q). In *Advances in Neural Information Processing Systems*, volume 33, pp. 15499–15510, 2020.

Neu, G. and Pike-Burke, C. A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1392–1403, 2020.

Olsson, J. and Westerborn, J. Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm. *Bernoulli*, 23(3):1951–1996, 2017.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. On nesting Monte Carlo estimators. In *International Conference on Machine Learning*, volume 80, pp. 4267–4276, 2018.

Rainforth, T., Foster, A., Ivanova, D. R., and Smith, F. B. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

Rawlik, K. C. *On Probabilistic Inference Approaches to Stochastic Optimal Control*. PhD thesis, University of Edinburgh, 2013.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 1951.

Särkkä, S. and Solin, A. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

Särkkä, S. and Svensson, L. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2nd edition, 2023.

Schultheis, M., Belousov, B., Abdulsamad, H., and Peters, J. Receding horizon curiosity. In *Conference on Robot Learning*, pp. 1278–1288. PMLR, 2020.

Shababo, B., Paige, B., Pakman, A., and Paninski, L. Bayesian inference and online experimental design for mapping neural microcircuits. In *International Conference on Neural Information Processing Systems*, 2013.

Tedrake, R. *Underactuated Robotics*. 2023. URL https://underactuated.csail.mit.edu. Course Notes for MIT 6.832, Accessed 21.12.2023.

Toussaint, M. and Storkey, A. Probabilistic inference for solving discrete and continuous state Markov decision processes. In *International Conference on Machine Learning*, pp. 945–952, 2006.

Whitehouse, M., Whiteley, N., and Rimella, L. Consistent and fast inference in compartmental models of epidemics using Poisson approximate likelihoods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(4):1173–1203, 2023.

Whittle, P. A risk-sensitive maximum principle. *Systems & Control Letters*, 15(3):183–192, 1990.

Wigren, A., Risuleo, R. S., Murray, L., and Lindsten, F. Parameter elimination in particle Gibbs sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Proof of Proposition 1

Proof. We start with the definition of the expected information gain in the terminal reward framework,

$$\begin{aligned}\mathcal{I}(\phi) &\coloneqq \mathbb{E}_{p_\phi(z_{0:T})}\Big[\mathbb{H}\big[p(\theta)\big] - \mathbb{H}\big[p(\theta \mid z_{0:T})\big]\Big] \\ &= \mathbb{E}_{p_\phi(z_{0:T},\theta)}\left[\log \frac{p(\theta \mid z_{0:T})}{p(\theta)}\right].\end{aligned}$$

With repeated applications of Bayes' rule, we can write this in an equivalent form as

$$\mathcal{I}(\phi) = \mathbb{E}_{p_\phi(z_{0:T},\theta)}\big[\log p_\phi(z_{0:T} \mid \theta)\big] - \mathbb{E}_{p_\phi(z_{0:T})}\big[\log p_\phi(z_{0:T})\big]. \tag{17}$$

Let us look at the first term in (17). From (2), we know that the conditional trajectory likelihood is

$$p_\phi(z_{0:T} \mid \theta) = p(x_0)\prod_{t=1}^{T} f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\, \pi_\phi(\xi_{t-1} \mid z_{0:t-1}).$$

We can then evaluate

$$\begin{aligned}T_1 &\coloneqq \mathbb{E}_{p_\phi(z_{0:T},\theta)}\big[\log p_\phi(z_{0:T} \mid \theta)\big] \\ &= \mathbb{E}_{p_\phi(z_{0:T},\theta)}\left[\log p(x_0) + \sum_{t=1}^{T}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\right] \\ &= -\mathbb{H}[p(x_0)] + \mathbb{E}_{p_\phi(z_{0:T},\theta)}\left[\sum_{t=1}^{T}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\right].\end{aligned} \tag{18}$$

For the second term in the above equation, we get

$$\begin{aligned}\mathbb{E}_{p_\phi(z_{0:T},\theta)}\left[\sum_{t=1}^{T}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] &= \sum_{t=1}^{T}\mathbb{E}_{p_\phi(z_{0:t},\theta)}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \\ &= \sum_{t=1}^{T}\mathbb{E}_{p_\phi(z_{0:t})\,p(\theta|z_{0:t})}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\mathbb{E}_{p(\theta|z_{0:t})}\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] \\ &=: \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\alpha_t(z_{0:t})\right],\end{aligned}$$

where we have made use of the fact that the conditional dynamics is Markovian in multiple places. We are then left with

$$T_1 = -\mathbb{H}[p(x_0)] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\alpha_t(z_{0:t})\right] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T}\log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\right].$$

Let's now look at the second term in (17). For that, we first need the following factorization of the marginal trajectory distribution

$$\begin{aligned}\log p_\phi(z_{0:T}) &= \log\left[p(z_0)\prod_{t=1}^{T}p_\phi(z_t \mid z_{0:t-1})\right] \\ &= \log\left[p(x_0)\prod_{t=1}^{T}p_\phi(x_t, \xi_{t-1} \mid z_{0:t-1})\right] \\ &= \log p(x_0) + \sum_{t=1}^{T}\Big[\log p(x_t \mid z_{0:t-1}, \xi_{t-1}) + \log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\Big],\end{aligned}$$

where

$$\begin{aligned} p(x_t \mid z_{0:t-1}, \xi_{t-1}) &= \int_\Theta p(x_t, \theta \mid z_{0:t-1}, \xi_{t-1}) \, \mathrm{d}\theta \\ &= \int_\Theta p(x_t \mid z_{0:t-1}, \xi_{t-1}, \theta) \, p(\theta \mid z_{0:t-1}, \xi_{t-1}) \, \mathrm{d}\theta \\ &= \int_\Theta f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \, p(\theta \mid z_{0:t-1}) \, \mathrm{d}\theta. \end{aligned}$$

In the last line, we have used the fact that θ is conditionally independent of ξ_{t-1} given $z_{0:t-1}$ (since the policy is independent of θ). We can now compute the second term of the EIG as

$$\begin{aligned} T_2 &\coloneqq \mathbb{E}_{p_\phi(z_{0:T})}\big[\log p_\phi(z_{0:T})\big] \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\log p(x_0) + \sum_{t=1}^{T}\Big[\log p(x_t \mid z_{0:t-1}, \xi_{t-1}) + \log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\Big]\right] \\ &= -\mathbb{H}[p(x_0)] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \log p(x_t \mid z_{0:t-1}, \xi_{t-1})\right] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\right]. \end{aligned}$$

The full expression for $\mathcal{I}(\phi)$ is hence

$$\begin{aligned} \mathcal{I}(\phi) &= T_1 - T_2 \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \alpha_t(z_{0:t})\right] - \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \log p(x_t \mid z_{0:t-1}, \xi_{t-1})\right] \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \Big\{\alpha_t(z_{0:t}) + \beta_t(z_{0:t})\Big\}\right] \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} r_t(z_{0:t})\right], \end{aligned}$$

where we have defined $\beta_t(z_{0:t}) = -\log p(x_t \mid z_{0:t-1}, \xi_{t-1})$ and $r_t(z_{0:t}) = \alpha_t(z_{0:t}) + \beta_t(z_{0:t})$. This concludes the proof for the first part of Proposition 1.

Let us now assume that our model has additive, constant noise in the dynamics (noise that is independent of the state, design and θ parameters). Under this assumption, the entropy $\mathbb{H}[f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)]$ is a constant. Let us now go back to the second term in (18),

$$\begin{aligned} \mathbb{E}_{p_\phi(z_{0:T}, \theta)}\left[\sum_{t=1}^{T} \log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] &= \sum_{t=1}^{T} \mathbb{E}_{p(\theta)\, p_\phi(z_{0:T}|\theta)}\left[\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{p(\theta)\, p_\phi(x_{0:t}, \xi_{0:t-1}|\theta)}\left[\log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{p(\theta)\, p_\phi(x_{0:t-1}, \xi_{0:t-1}|\theta)}\left[\int_{\mathcal{X}} \log f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)\, f(x_t \mid x_{t-1}, \xi_{t-1}, \theta) \, \mathrm{d}x_t\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{p(\theta)\, p_\phi(x_{0:t-1}, \xi_{0:t-1}|\theta)}\big[-\mathbb{H}[f(x_t \mid x_{t-1}, \xi_{t-1}, \theta)]\big] \\ &= \text{constant}. \end{aligned}$$

Using '$\equiv$' to denote equality up to an additive constant, we now have

$$T_1 \equiv -\mathbb{H}[p(x_0)] + \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \log \pi_\phi(\xi_{t-1} \mid z_{0:t-1})\right],$$

and hence,

$$\begin{aligned}\mathcal{I}(\phi) &= T_1 - T_2 \\ &\equiv \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} -\log p(x_t \mid z_{0:t-1}, \xi_{t-1})\right] \\ &= \mathbb{E}_{p_\phi(z_{0:T})}\left[\sum_{t=1}^{T} \beta_t(z_{0:t})\right]\end{aligned}$$

as required. □

B. Proof of Proposition 2

We prove this result by induction over t. We note that this result is not directly implied by the existing classical sequential Monte Carlo theory due to the dependency of the likelihood term on the filtering distribution over θ.

We will make use of the following assumptions.

Assumption 1. For all $z_{0:t+1}, t \geq 1$ there exists $\alpha > 0$ such that, for all θ, $0 < f(x_{t+1} \mid x_t, \xi_t, \theta) < \alpha$.

Assumption 2. For all $z_{0:t}, t \geq 1$ there exists an integrable function $\beta_{t+1}(z_{t+1})$ such that, for all θ, $0 < p(z_{t+1} \mid z_{0:t}, \theta) < \beta_{t+1}(z_{t+1})$.

Proof. For simplicity, we assume that resampling happens at each step of IBIS in Algorithm 2. The result is clear for $t = 0$ under the law of large numbers. Now assume it is true for a given t, then, applying Proposition 2 to the test function $\theta \mapsto f(x_{t+1} \mid x_t, \xi_t, \theta)$ we have, thanks to Assumption 1

$$\frac{1}{M}\sum_{m=1}^{M} f(x_{t+1} \mid x_t, \xi_t, \theta_t^m) \to \mathbb{E}_{p(\theta_t \mid z_{0:t})}\left[f(x_{t+1} \mid x_t, \xi_t, \theta_t)\right] = p(x_{t+1} \mid z_{0:t}, \xi_t)$$

so that, with probability 1,

$$\exp\left\{-\eta \log \frac{1}{M}\sum_{m=1}^{M} f(x_{t+1} \mid x_t, \xi_t, \theta_t^m)\right\} \to \exp\left\{-\eta \log p(x_{t+1} \mid z_{0:t}, \xi_t)\right\}.$$

Applying the induction hypothesis to $\theta \mapsto p(z_{t+1} \mid z_{0:t}, \theta)$, we have with probability 1,

$$\frac{1}{M}\sum_{m=1}^{M} p(z_{t+1} \mid z_{0:t}, \theta_t^m) \to p(z_{t+1} \mid z_{0:t}).$$

As a consequence,

$$\left[\frac{1}{M}\sum_{m=1}^{M} p(z_{t+1} \mid z_{0:t}, \theta_t^m)\right] \exp\left\{-\eta \log \frac{1}{M}\sum_{m=1}^{M} f(x_{t+1} \mid x_t, \xi_t, \theta_t^m)\right\} \to p(z_{t+1} \mid z_{0:t}) \exp\left\{-\eta \log p(x_{t+1} \mid z_{0:t}, \xi_t)\right\}$$

almost surely. Similarly, using the positivity of $p(z_{t+1} \mid z_{0:t}, \theta_t^m)$ and by Lebesgue's dominated convergence theorem, the normalizing constant of the right-hand side of (14)

$$\int \left[\frac{1}{M}\sum_{m=1}^{M} p(z_{t+1} \mid z_{0:t}, \theta_t^m)\right] \exp\left\{-\eta \log \frac{1}{M}\sum_{m=1}^{M} f(x_{t+1} \mid x_t, \xi_t, \theta_t^m)\right\} \mathrm{d}z_{t+1}$$

converges to

$$\int p(z_{t+1} \mid z_{0:t}) \exp\left\{-\eta \log p(x_{t+1} \mid z_{0:t}, \xi_t)\right\} \mathrm{d}z_{t+1}$$

and we have

$$\frac{\Gamma_{t+1}^M(z_{0:t+1}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})} \to \frac{p(z_{t+1} \mid z_{0:t}) \exp\left\{-\eta \log p(x_{t+1} \mid z_{0:t}, \xi_t)\right\}}{\int p(z_{t+1} \mid z_{0:t}) \exp\left\{-\eta \log p(x_{t+1} \mid z_{0:t}, \xi_t)\right\} \mathrm{d}z_{t+1}}.$$

The recursion is then obtained by noticing that the IBIS step (13) corresponds to a particle filter update targeting $p(\theta \mid z_{0:t+1})$, so that, under Assumption 2, we can follow Proposition 11.4 in Chopin & Papaspiliopoulos (2020) to obtain that, for any

bounded test function ψ, writing

$$\Gamma_t^{M,\mathrm{IBIS}}(\theta_t^{1:M}) = \frac{\Gamma_t^M(z_{0:t}, \theta_{0:t}^{1:M}, a_{1:t}^{1:M})}{\Gamma_t^M(z_{0:t}, \theta_{0:t-1}^{1:M}, a_{1:t-1}^{1:M})},$$

we have

$$\mathbb{E}_{\Gamma_t^{M,\mathrm{IBIS}}}[\psi(\theta_t)] \to \mathbb{E}_{p(\theta_t|z_{0:t})}[\psi(\theta_t)]$$

almost surely. Putting it all together, and noticing that $\Gamma_t(z_{0:t})p(\theta_t \mid z_{0:t}) = \Gamma_t(z_{0:t}, \theta_t)$, we obtain the result. □

C. Algorithmic Details

C.1. Reweighting in Inside-Out SMC^2 for General Potentials

Algorithm 4 Reweight function corresponding to (5)

notation Any operation with superscript m is to be understood as performed for all $m = 1, \ldots, M$.
function REWEIGHT(t)
1: $v_t^{mn} = f(x_t^n \mid x_{t-1}^n, \xi_{t-1}^n, \theta_{t-1}^{mn})$.
2: $W_{t,\theta}^{mn} \propto W_{t-1,\theta}^{mn} v_t^{mn}$.
3: $r_t^n = \sum_{m=1}^M W_{t,\theta}^{mn} \log v_t^{mn} - \log \sum_{m=1}^M W_{t-1,\theta}^{mn} v_t^{mn}$.
4: **return** $g_t^n = \exp\{\eta\, r_t^n\}$.
end function

Let us reproduce the general expression for the stage reward at time $t+1$ from (5).

$$\begin{aligned} r_t(z_{0:t+1}) &= \mathbb{E}_{p(\theta|z_{0:t+1})}\big[\log f(x_{t+1} \mid x_t, \xi_t, \theta)\big] - \log p(x_{t+1} \mid z_{0:t}, \xi_t) \\ &= \mathbb{E}_{p(\theta|z_{0:t+1})}\big[\log f(x_{t+1} \mid x_t, \xi_t, \theta)\big] - \log \mathbb{E}_{p(\theta|z_{0:t})}\left[f(x_{t+1} \mid x_t, \xi_t, \theta)\right]. \end{aligned} \tag{19}$$

We see that we have expectations with respect to the filtering posteriors of θ at times t and $t+1$. At line 8 of Algorithm 3, we have a trajectory $z_{0:t+1}^n$ and a particle representation $\sum_{m=1}^M W_{t,\theta}^{mn} \delta_{\theta_t^{mn}}(\theta) \approx p(\theta \mid z_{0:t}^n)$. The second term of the reward function can be estimated as

$$\log \mathbb{E}_{p(\theta|z_{0:t}^n)}\left[f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta)\right] \approx \log \sum_{m=1}^M W_{t,\theta}^{mn} f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta_t^{mn}).$$

Now, to compute the first term, we need to approximate $p(\theta \mid z_{0:t+1}^n)$. For this we perform the reweighting step of IBIS (line 2 from Algorithm 2) to get updated weights

$$\begin{aligned} W_{t+1,\theta}^{mn} &\propto W_{t,\theta}^{mn}\, p_\phi(z_{t+1}^n \mid z_{0:t}^n, \theta_t^{mn}) \\ &= W_{t,\theta}^{mn}\, f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta_t^{mn})\, \pi_\phi(\xi_t^n \mid z_{0:t}^n) \\ &\propto W_{t,\theta}^{mn}\, f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta_t^{mn}). \end{aligned}$$

The distribution $\sum_{m=1}^M W_{t+1,\theta}^{mn} \delta_{\theta_t^{mn}}(\theta)$ now approximates the posterior $p(\theta \mid z_{0:t+1}^n)$ as required. We choose not to perform the resample-move step here so as not to introduce additional variance. The first term of (19) can now be approximated as

$$\mathbb{E}_{p(\theta|z_{0:t+1}^n)}\big[\log f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta)\big] \approx \sum_{m=1}^M W_{t+1,\theta}^{mn} \log f(x_{t+1}^n \mid x_t^n, \xi_t^n, \theta_t^{mn}).$$

The entire reweighting procedure is outlined in Algorithm 4.

C.2. Choice of the Markov Kernel for IBIS

For the Markov kernel Q_t in IBIS (Algorithm 2), we follow the choice in Chopin et al. (2013) and use a Metropolis-Hastings kernel (Metropolis et al., 1953; Hastings, 1970). If the prior is Gaussian, we use a Gaussian random walk proposal

$$\tilde{\theta}^m \mid \theta^m \sim \mathcal{N}(\theta^m, c\,\hat{\Sigma}), \tag{20}$$

where

$$\hat{\Sigma} = \frac{1}{\sum_{m=1}^M w^m} \sum_{m=1}^M w^m (\theta^m - \hat{\mu})(\theta^m - \hat{\mu})^\top, \quad \hat{\mu} = \frac{1}{\sum_{m=1}^M w^m} \sum_{m=1}^M w^m \theta^m,$$

and $c \in \mathbb{R}_{>0}$ is a constant that can be tuned to achieve a desired acceptance ratio. For log-normal priors, we use a similar random walk proposal

$$\tilde{\theta}^m \mid \theta^m \sim \text{LogNormal}(\theta^m, c\,\hat{\Sigma}).$$

We found that the proposal in (20) worked better empirically compared to the proposal $\tilde{\theta}^m \mid \theta^m \sim N(\hat{\mu}, \hat{\Sigma})$ suggested in Chopin (2002), or a proposal which does not use the sample covariance, $\tilde{\theta}^m \mid \theta^m \sim N(\theta^m, cI)$. In our evaluation, we perform multiple move steps per IBIS step to get a richer representation of samples.

C.3. Inside-Out SMC^2 with Conjugate Prior-Likelihood Pairs

Algorithm 5 Inside-Out SMC^2 (Exact)

notation Any operation with superscript n is to be understood as performed for all $n = 1, \ldots, N$.
1: Sample $z_0^n \sim p(\cdot)$.
2: Sample $z_1^n \sim p(\cdot \mid z_0^n)$ and initialize the state history $z_{0:1}^n \leftarrow (z_0^n, z_1^n)$.
3: Compute and normalize the weights $W_z^n \propto g_1(z_{0:1}^n)$.
4: **for** $t \leftarrow 1, \ldots, T-1$ **do**
5: Sample $b_t^n \sim \mathcal{M}(W_z^{1:N})$.
6: Compute the θ posterior $p(\theta \mid z_{0:t}^{b_t^n})$.
7: Sample $z_{t+1}^n \sim p(\cdot \mid z_{0:t}^{b_t^n})$ and append to state history $z_{0:t+1}^n \leftarrow (z_{0:t}^{b_t^n}, z_{t+1}^n)$.
8: Compute and normalize the weights $W_z^n \propto g_t(z_{0:t}^n)$.
9: **end for**
10: **return** $\{z_{0:T}^n, W_z^n\}_{n=1}^N$.

Let $\theta \in \mathbb{R}^{d_\theta}$, $x_t \in \mathbb{R}^{d_x}$ and $\xi_t \in \mathbb{R}^{d_\xi}$. Let us consider conditionally linear, Gaussian transition dynamics for x,

$$f(x_{t+1} \mid x_t, \xi_t, \theta) = \mathcal{N}(x_{t+1} \mid H(x_t, \xi_t)\,\theta, \Sigma(x_t, \xi_t)),$$

where H is a map $\mathbb{R}^{d_x + d_\xi} \mapsto \mathbb{R}^{d_x \times d_\theta}$ and $\Sigma : \mathbb{R}^{d_x + d_\xi} \to \mathbb{R}^{d_x \times d_x}$ maps to positive definite matrices. Let us also assume that at time t, the filtered posterior of θ is Gaussian with mean m_t and covariance matrix P_t,

$$p(\theta \mid z_{0:t}) = \mathcal{N}(\theta \mid m_t, P_t).$$

Then, using basic identities of the multivariate normal distribution, the marginal density is

$$\begin{aligned} p(x_{t+1} \mid z_{0:t}, \xi_t) &= \int f(x_{t+1} \mid x_t, \xi_t, \theta)\, p(\theta \mid z_{0:t})\, \mathrm{d}\theta \\ &= \mathcal{N}(x_{t+1} \mid H m_t, H P_t H^T + \Sigma), \end{aligned}$$

where the functional dependence of H and Σ on (x_t, ξ_t) has been hidden for conciseness. Furthermore, upon observing the next augmented state $z_{t+1} = (x_{t+1}, \xi_t)$, the filtered posterior of θ can be updated using Bayes' rule:

$$\begin{aligned} p(\theta \mid z_{0:t+1}) &= \frac{p_\phi(x_{t+1}, \xi_t \mid z_{0:t}, \theta)\, p(\theta \mid z_{0:t})}{p_\phi(x_{t+1}, \xi_t \mid z_{0:t})} \\ &= \frac{f(x_{t+1} \mid x_t, \xi_t, \theta)\, \pi_\phi(\xi_t \mid z_{0:t})\, p(\theta \mid z_{0:t})}{p(x_{t+1} \mid z_{0:t}, \xi_t)\, \pi_\phi(\xi_t \mid z_{0:t})} \\ &= \frac{f(x_{t+1} \mid x_t, \xi_t, \theta)\, p(\theta \mid z_{0:t})}{p(x_{t+1} \mid z_{0:t}, \xi_t)} \\ &= \mathcal{N}(\theta \mid m_{t+1}, P_{t+1}), \end{aligned}$$

where

$$m_{t+1} = m_t + G(x_{t+1} - H m_t), \quad P_{t+1} = P_t - G H P_t, \quad G = P_t H^\top (H P_t H^\top + \Sigma)^{-1}.$$

Thus, we see that for a conditionally linear model with Gaussian priors and likelihoods, we can compute the marginal density and the θ posterior in closed form. The same holds for any conjugate prior-likelihood pair. Consequently, we can compute the stage reward in (5) and hence the potential function in closed form, and the resulting version of the IO-SMC^2 algorithm that does not use the inner particle filter is given in Algorithm 5.

Algorithm 6 Conditional Inside-Out SMC^2 kernel

input Reference trajectory $\{z_{0:T}, \{W^{\bullet}_{t,\theta}, \theta^{\bullet}_t\}_{t=0}^{T-1}\}$.
output New reference trajectory $\{z^*_{0:T}, \{W^{\bullet *}_{t,\theta}, \theta^{\bullet *}_t\}_{t=0}^{T-1}\}$.
Set $z_0^1 \leftarrow z_0$, $\theta_0^{\bullet 1} \leftarrow \theta_0^{\bullet}$ and $W_{0,\theta}^{\bullet 1} \leftarrow W_{0,\theta}^{\bullet}$.
for $n = 2, \dots, N$ **do**
  Sample $z_0^n \sim p(z_0)$, $\theta_0^{\bullet n} \sim p(\theta)$ and set $W_{0,\theta}^{\bullet n} \leftarrow 1/M$.
end for
Set $z_{0:1}^1 \leftarrow z_{0:1}$.
for $n = 2, \dots, N$ **do**
  Sample $z_1^n \sim \hat{p}_\phi(\cdot \mid z_0^n)$ and set $z_{0:1}^n \leftarrow (z_0^n, z_1^n)$.
end for
Compute and normalize the weights $W_z^n \propto g_1^n$ for all $n = 1, \dots, N$.
for $t \leftarrow 1, \dots, T-1$ **do**
  Set $z_{t+1}^1 \leftarrow z_{t+1}$, $\theta_t^{\bullet 1} \leftarrow \theta_t^{\bullet}$, $W_{t,\theta}^{\bullet 1} \leftarrow W_{t,\theta}^{\bullet}$, and $z_{0:t+1}^1 \leftarrow z_{0:t+1}$.
  for $n = 2, \dots, N$ **do**
    Sample $b_t^n \sim \mathcal{M}(W_z^{1:N})$.
    $\theta_t^{\bullet n}, W_{t,\theta}^{\bullet n}, \leftarrow$ IBIS_STEP$(z_{0:t}^{b_t^n}, \theta_{t-1}^{\bullet b_t^n}, W_{t-1,\theta}^{\bullet b_t^n})$
    Sample $z_{t+1}^n \sim \hat{p}_\phi(\cdot \mid z_{0:t}^{b_t^n})$, and append to state history $z_{0:t+1}^n \leftarrow [z_{0:t}^{b_t^n}, z_{t+1}^n]$.
  end for
  Compute and normalize the weights $W_z^n \propto g_t^n$ for all $n = 1, \dots, N$.
end for
Sample an index $j \sim \mathcal{M}(W_z^{1:N})$.
return $\{z_{0:T}^j, \{W_{t,\theta}^{\bullet j}, \theta_t^{\bullet j}\}_{t=0}^{T-1}\}$.

C.4. Conditional SMC

In Section 4.3, we saw that IO-SMC^2 is a nested particle filter that targets the distribution Γ_T^M. In this section, we construct a conditional version of our algorithm that keeps Γ_T^M invariant. The basic idea behind CSMC is that given a *reference trajectory* from the target distribution, at each time step in the forward pass, we sample $N-1$ samples conditionally on the reference particle surviving the resampling step (Andrieu et al., 2010). The CSMC kernel for Γ_t^M is outlined in Algorithm 6, where the potential function estimates g_t^n are computed as detailed in Algorithm 4.

While Algorithm 6 may look more complicated than "classical" CSMC algorithms (Andrieu et al., 2010), its complexity may be abstracted away by remembering that a 'particle' object is, in our case, an instance of the approximate inner distribution Γ_t^M, which is associated with its weights, particles and ancestors, noting that the ancestors do not appear in the computation of Algorithm 4 and are therefore omitted.

D. Experimental Details

D.1. Network Architectures and Hyperparameters

We use the same network architecture for all amortized policies in the evaluation. The architecture of our design policy network is similar to that in iDAD (Ivanova et al., 2021), with an encoder network transforming the augmented state sequences into a stacked representation $\{R(z_s)\}_{s=0}^t$ before passing it to the recurrent layers. The encoder networks for the augmented states are given in Table 5 and Table 6. For training, we used the Adam optimizer (Kingma & Ba, 2014). Our policies were trained on a single 9$^{\text{th}}$ Gen Intel Core i9 processor, while iDAD policies were trained on an Nvidia A100 GPU using the authors' implementation (Ivanova et al., 2021). The hyperparameters used to train iDAD are listed in Table 7, and are common to all experiments.

All IO-SMC^2 policies were trained with an additional slew rate penalty on the designs. We noticed that this detail promoted smoother design trajectories, that facilitated the amortization of the recurrent policy.

Table 5. The encoder architecture.

Layer	Description	Size	Activation
Input	Augmented state z	$\dim(z)$	-
Hidden layer 1	Dense	256	ReLU
Hidden layer 2	Dense	256	ReLU
Output	Dense	64	-

Table 6. The recurrent network architecture.

Layer	Description	Size	Activation
Input	$\{R(z_s)\}_{s=0}^{t}$	$64 \cdot (t+1)$	-
Hidden layer 1	LSTM / GRU	64	-
Hidden layer 2	LSTM / GRU	64	-
Hidden layer 3	Dense	256	ReLU
Hidden layer 4	Dense	256	ReLU
Output	Designs ξ	$\dim(\xi)$	-

Table 7. Hyperparameters for iDAD.

Hyperparameter	iDAD
Batch size	512
Number of contrastive samples	16383
Number of gradient steps	10000
Learning rate (LR)	5×10^{-4}
LR annealing parameter	0.96
LR annealing frequency (if no improvement)	400

D.2. Stochastic Pendulum Experiment

We consider two different dynamics for the compound pendulum, one conditionally linear in the parameters and another that is fully nonlinear. The following specifications are similar in both settings.

Let $x_t = [q_t, \dot{q}_t]^\top$ denote the state of the pendulum, with q_t being the angle from the vertical and $\dot{q}_t$ the angular velocity. The parameters of interest are (m, l), the mass and length of the pendulum, while $g = 9.81$ and $d = 0.1$ are the gravitational acceleration and damping constants. The design, $\xi_t \in [-1, 1]$, is the torque applied to the pendulum. We discretize the respective SDEs that describe the dynamics using Euler-Maruyama with a step size $\mathrm{d}t = 0.05$ and consider a horizon of $T = 50$ experiments. The initial state is fixed to $x_0 = [0, 0]^\top$.

D.2.1. Conditionally Linear formulation

In this setting, we transform the non-linear pendulum equations to obtain a conditional linear dependency on the parameters, similar to Belousov et al. (2019). The resulting parameter vector is $\theta = \left[\frac{3g}{2l}, \frac{3d}{ml^2}, \frac{3}{ml^2}\right]^\top$. The dynamics is described by the following Ito SDE (Särkkä & Solin, 2019, Chapter 3)

$$\mathrm{d}x_t = h(x_t, \xi_t)^\top \theta \, \mathrm{d}t + L \, \mathrm{d}\beta,$$

with a drift term $h(x_t, \xi_t) = [-\sin(q), -\dot{q}, \xi_t]^\top$, diffusion term $L = [0, 0.1]^\top$ and Brownian motion β.

To maintain conjugacy, we assume a Gaussian prior

$$p(\theta) = \text{Normal}\left(\begin{bmatrix}14.7\\0\\3.0\end{bmatrix}, \begin{bmatrix}0.1 & 0 & 0\\0 & 0.01 & 0\\0 & 0 & 0.1\end{bmatrix}\right).$$

The remaining hyperparameters are listed in Table 8.

Table 8. Hyperparameters for the conditionally linear pendulum experiment.

Hyperparameter	IO-SMC^2	IO-SMC^2 (Exact)
N	256	256
M	128	-
Tempering (η)	1.0	1.0
Slew rate penalty	0.1	0.1
IBIS moves	3	-
Learning rate	10^{-3}	10^{-3}
Training iterations	25	25

D.2.2. NONLINEAR FORMULATION

The unknown parameters are $\theta = (m, l)$. The dynamics is described by the SDE

$$\mathrm{d}x_t = h(x_t, \xi_t, \theta)^\top \,\mathrm{d}t + L\,\mathrm{d}\beta,$$

where $h(x_t, \xi_t, \theta) = [\dot{q}_t, \ddot{q}_t]^\top$ and

$$\ddot{q}_t = -\frac{3g}{2l}\sin q_t + \frac{(\xi_t - d\,\dot{q}_t)}{ml^2},$$

and $L = [0, 0.1]^\top$. The prior is a log-normal distribution

$$p(\theta) = \text{LogNormal}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0.01 & 0\\ 0 & 0.01\end{bmatrix}\right).$$

The remaining hyperparameters for IO-SMC^2 are listed in Table 9, and the training progression is depicted in Figure 6.

Table 9. Hyperparameters for the non-linear pendulum, stochastic cart-pole, and stochastic dual-link experiments.

Hyperparameter	Nonlinear pendulum	Stochastic cart-pole	Stochastic dual-link
N	256	256	256
M	128	1024	128
Tempering (η)	1.0	0.25	0.25
Slew rate penalty	0.2	0.1	0.1
IBIS moves	3	3	3
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}
Training iterations	25	15	25

D.3. Stochastic Cart-Pole Experiment

The cart-pole is described by a state $x_t = [s_t, q_t, \dot{s}_t, \dot{q}_t]^\top$, where s_t and $\dot{s}_t$ are the position and velocity of the cart, and q_t and $\dot{q}_t$ are the position and velocity of the pole. The unknowns are $\theta = (l, m_p, m_c)$, the length and mass of the pendulum and the mass of the cart, respectively. The design, $\xi_t \in [-5, 5]$, is the force applied to the cart. The corresponding SDE is

$$\mathrm{d}x_t = h(x_t, \xi_t, \theta)\,\mathrm{d}t + L\,\mathrm{d}\beta,$$

where $h(x_t, \xi_t, \theta) = [\dot{s}_t, \dot{q}_t, \ddot{s}_t, \ddot{q}_t]^\top$ and

$$\ddot{s}_t = \frac{1}{m_c + m_p \sin^2 q_t}\left[\xi_t + m_p \sin q_t (l\dot{q}_t^2 + g\cos q_t) - (k_1 s_t + d_1 \dot{s}_t) - (k_2 q_t + d_2 \dot{q}_t)\cos q_t / l\right],$$

$$\ddot{q}_t = \frac{1}{l(m_c + m_p \sin^2 q_t)}\big[-\xi_t \cos q_t - m_p l \dot{q}_t^2 \cos q_t \sin q_t - (m_c + m_p) g \sin q_t$$
$$- (k_1 s_t + d_1 \dot{s}_t)\cos q_t - (k_2 q_t + d_2 \dot{q}_t)\cos^2 q_t / l\big].$$

$(k_1, k_2, d_1, d_2) = 0.01$ are the linear and torsional stiffness and damping constants and $g = 9.81$. The diffusion term is $L = [0, 0, 0.1, 0]^\top$. We discretize the SDE with a step size $\mathrm{d}t = 0.05$ and consider a horizon of $T = 50$ experiments. The initial state is fixed at $x_0 = [0, 0, 0, 0]^\top$. The prior for θ is log-normal

$$p(\theta) = \text{LogNormal}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0.01 & 0 & 0\\0 & 0.01 & 0\\0 & 0 & 0.01\end{bmatrix}\right).$$

The remaining hyperparameters used for this experiment are given in Table 9. In Table 10, we compare the sample efficiency of our method against the sPCE bound as an estimator of the EIG. For a trained policy and a fixed number of trajectory samples, we vary the number of parameter particles (contrastive samples). Our method reaches the asymptotic value of approximately 21 at 1024 particles, while the sPCE does not and fully saturates its sample-size dependent upper bound (Foster et al., 2021, Appendix A) for all the experiments we ran. Finally, the training progression of the algorithm is depicted in Figure 7.

Table 10. EIG estimates and sPCE lower bounds for a trained policy on the stochastic cart-pole experiment for different numbers of θ particles. We report the mean and standard deviation over 25 random seeds.

No. of θ particles (M)	EIG Estimate (16)	sPCE	sPCE theoretical limit ($\log(M+1)$)
64	14.52 ± 0.71	4.17 ± 0.00	4.17
128	17.38 ± 0.93	4.86 ± 0.00	4.86
256	19.20 ± 0.68	5.55 ± 0.00	5.55
512	20.67 ± 0.82	6.24 ± 0.00	6.24
1024	21.29 ± 0.63	6.93 ± 0.00	6.93
2048	21.60 ± 0.37	7.62 ± 0.00	7.62
4096	21.87 ± 0.56	8.32 ± 0.00	8.32
8192	21.77 ± 0.56	9.01 ± 0.00	9.01

D.4. Stochastic Double-Link Experiment

The double-link system is described by a state vector $x = [q, \dot{q}]^\top = [q_1, q_2, \dot{q}_1, \dot{q}_2]^\top$, where q_1 and q_2 are the angles of the two joints (Tedrake, 2023, Appendix B). The system is doubly actuated, with $\xi = [\xi_1, \xi_2]^\top$, $\xi_1 \in [-4, 4]$ and $\xi_2 \in [-2, 2]$. The parameters of interest are $\theta = (m_1, m_2, l_1, l_2)$, the masses and lengths of the two joints, respectively. We define

$$M(q) = \begin{bmatrix}(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos q_2\\ m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 & m_2 l_2^2\end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix}0 & -m_2 l_1 l_2 (2\dot{q}_1 + \dot{q}_2)\sin q_2\\ \frac{1}{2} m_2 l_1 l_2 (2\dot{q}_1 + \dot{q}_2)\sin q_2 & -\frac{1}{2} m_2 l_1 l_2 \dot{q}_1 \sin q_2\end{bmatrix},$$

$$\tau_g(q) = -g \cdot \begin{bmatrix}(m_1+m_2) l_1 \sin(q_1) + m_2 l_2 \sin(q_1+q_2)\\ m_2 l_2 \sin(q_1+q_2)\end{bmatrix},$$

where $g = 9.81$. The system dynamics are then given by the equations

$$\begin{aligned}\mathrm{d}q &= \dot{q}\,\mathrm{d}t,\\ \mathrm{d}\dot{q} &= M^{-1}(q)[\tau_g(q) + \xi - C(q, \dot{q})\dot{q}]\,\mathrm{d}t + L\mathrm{d}\beta,\end{aligned}$$

where $L = \begin{bmatrix}0.1 & 0\\ 0 & 0.1\end{bmatrix}$ and $\beta = [\beta_1, \beta_2]^\top$, with β_1 and β_2 being two independent Brownian motions. We assume a log-normal prior of the form

$$p(\theta) = \text{LogNormal}\left(\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0.01 & 0 & 0 & 0\\0 & 0.01 & 0 & 0\\0 & 0 & 0.01 & 0\\0 & 0 & 0 & 0.01\end{bmatrix}\right).$$

The remaining hyperparameters for IO-SMC^2 are listed in Table 9, and the training progression is depicted in Figure 8.

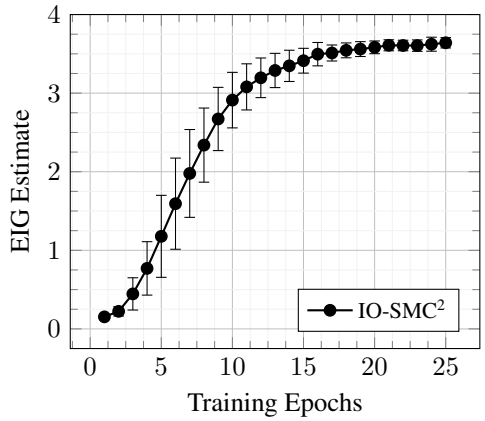

Figure 6. Training progression of the IO-SMC^2 policy on the non-linear stochastic pendulum experiment. At every epoch, we evaluate the EIG estimate using the mean policy. We report the mean and standard deviation of the EIG estimate over 25 unique training seeds.

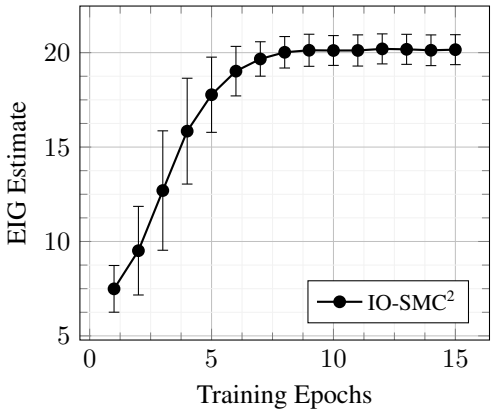

Figure 7. Training progression of the IO-SMC^2 policy for the stochastic cart-pole experiment. At every epoch, we evaluate the EIG estimate using the mean policy. We report the mean and standard deviation of the EIG estimate over 25 unique training seeds.

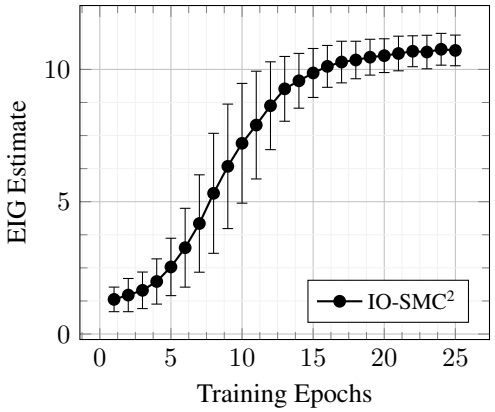

Figure 8. Training progression of the IO-SMC^2 policy for the stochastic double-link experiment. At every epoch, we evaluate the EIG estimate using the mean policy. We report the mean and standard deviation of the EIG estimate over 25 unique training seeds.