Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations

Citation: Conrad, Patrick R., Youssef M. Marzouk, Natesh S. Pillai, and Aaron Smith. "Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations." Journal of the American Statistical Association (October 21, 2015): 00–00.

As Published: http://dx.doi.org/10.1080/01621459.2015.1096787

Publisher: American Statistical Association

Persistent URL: http://hdl.handle.net/1721.1/99937

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations

Patrick R. Conrad[1], Youssef M. Marzouk[1], Natesh S. Pillai[2], and Aaron Smith[3]

[1]Department of Aeronautics and Astronautics, Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139, USA. {prconrad, ymarz}@mit.edu

[2]Department of Statistics, Harvard University
1 Oxford Street, Cambridge, MA 02138, USA. pillai@stat.harvard.edu

[3]Department of Mathematics and Statistics, University of Ottawa
585 King Edward Avenue, Ottawa, ON K1N 7N5, Canada. asmi28@uottawa.ca

September 16, 2015

Abstract

We construct a new framework for accelerating Markov chain Monte Carlo in posterior sampling problems where standard methods are limited by the computational cost of the likelihood, or of numerical models embedded therein. Our approach introduces local approximations of these models into the Metropolis-Hastings kernel, borrowing ideas from deterministic approximation theory, optimization, and experimental design. Previous efforts at integrating approximate models into inference typically sacrifice either the sampler's exactness or efficiency; our work seeks to address these limitations by exploiting useful convergence characteristics of local approximations. We prove the ergodicity of our approximate Markov chain, showing that it samples asymptotically from the *exact* posterior distribution of interest. We describe variations of the algorithm that employ either local polynomial approximations or local Gaussian process regressors. Our theoretical results reinforce the key observation underlying this paper: when the likelihood has some local regularity, the number of model evaluations per MCMC step can be greatly reduced without biasing the Monte Carlo average. Numerical experiments demonstrate multiple order-of-magnitude reductions in the number of forward model evaluations used in representative ODE and PDE inference problems, with both synthetic and real data.

Keywords: approximation theory, computer experiments, emulators, experimental design, local approximation, Markov chain Monte Carlo

1 Introduction

Bayesian inference for computationally intensive models is often limited by the computational cost of Markov chain Monte Carlo (MCMC) sampling. For example, scientific models in diverse fields such as geophysics, chemical kinetics, and biology often invoke ordinary or partial differential equations to describe the underlying physical or natural phenomena. These differential equations constitute the *forward model* which, combined with measurement or model error, yield a likelihood function. Given a numerical implementation of this physical model, standard MCMC techniques are in principle appropriate for sampling from the posterior distribution. However, the cost of running the forward model anew at each MCMC step can quickly become prohibitive if the forward model is computationally expensive.

An important strategy for mitigating this cost is to recognize that the forward model may exhibit regularity in its dependence on the parameters of interest, such that the model outputs may be approximated with fewer samples than are needed to characterize the posterior via MCMC. Replacing the forward model with an approximation or "surrogate" *decouples* the required number of forward model evaluations from the length of the MCMC chain, and thus can vastly reduce the overall cost of inference (Sacks et al., 1989; Kennedy and O'Hagan, 2001). Existing approaches typically create high-order global approximations for either the forward model outputs or the log-likelihood function using, for example, global polynomials (Marzouk et al., 2007; Marzouk and Xiu, 2009), radial basis functions (Bliznyuk et al., 2012; Joseph, 2012), or Gaussian processes (Sacks et al., 1989; Kennedy and O'Hagan, 2001; Rasmussen, 2003; Santner et al., 2003). As in most of these efforts, we will assume that the forward model is deterministic and available only as a black box, thus limiting ourselves to "non-intrusive" approximation methods that are based on evaluations of the forward model at selected input points.[1] Since we assume that the exact forward model is available and computable, but simply too expensive to be run a large number of times, the present setting is distinct from that of either pseudo-marginal MCMC or approximate Bayesian computation (ABC); these are important methods for intractable posteriors where the likelihood can only be estimated

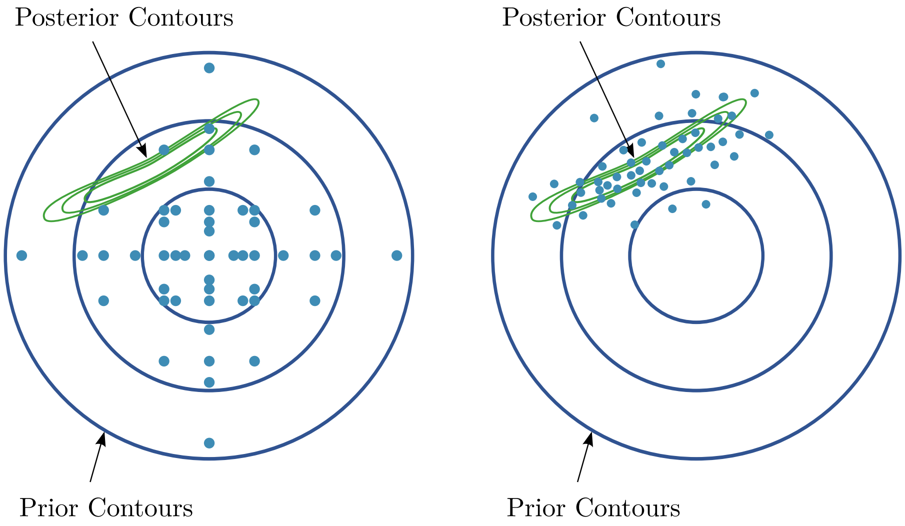
[1] Interesting examples of intrusive techniques exploit multiple spatial resolutions of the forward model (Higdon et al., 2003; Christen and Fox, 2005; Efendiev et al., 2006), models with tunable accuracy (Korattikara et al., 2013; Bal et al., 2013), or projection-based reduced order models (Frangos et al., 2010; Lieberman et al., 2010; Cui et al., 2014).

or simulated from, respectively (Andrieu and Roberts, 2009; Marin et al., 2011).[2]

Although current approximation methods can provide significant empirical performance improvements, they tend either to over- or under-utilize the surrogate, sacrificing exact sampling or potential speedup, respectively. In the first case, many methods produce some fixed approximation, inducing an approximate posterior. In principle, one might require only that the bias of a posterior expectation computed using samples from this approximate posterior be small relative to the variance introduced by the finite length of the MCMC chain, but current methods lack a rigorous approach to controlling this bias (Bliznyuk et al., 2008; Fielding et al., 2011); Cotter et al. (2010) show that bounding the bias is in principle possible, by proving that the rate of convergence of the forward model approximation can be transferred to the approximate posterior, but their bounds include unknown constants and hence do not suggest practical strategies for error control. Conversely, other methods limit potential performance improvement by failing to "trust" the surrogate even when it is accurate. Delayed-acceptance schemes, for example, eliminate the need for error analysis of the surrogate but require at least one full model evaluation for each accepted sample (Rasmussen, 2003; Christen and Fox, 2005; Cui et al., 2011), which remains a significant computational effort.

Also, analyzing the error of a forward model approximation can be quite challenging for the *global* approximation methods used in previous work—in particular for methods that use complex sequential experimental design heuristics to build surrogates over the posterior (Rasmussen, 2003; Bliznyuk et al., 2008; Fielding et al., 2011). Even when these design heuristics perform well, it is not clear how to establish rigorous error bounds for finite samples or even how to establish convergence for infinite samples, given relatively arbitrary point sets. Polynomial chaos expansions sidestep some of these issues by designing sample grids (Xiu and Hesthaven, 2005; Nobile et al., 2007; Constantine et al., 2012; Conrad and Marzouk, 2013) with respect to the prior distribution, which are known to induce a convergent approximation of the posterior density (Marzouk and Xiu, 2009). However, only using prior information is likely to be inefficient; whenever the data are informative, the posterior concentrates on a small fraction of the parameter space relative to the prior (Li and Marzouk, 2014). Figure 1 illustrates the contrast between a prior-based sparse grid (Conrad and Marzouk, 2013) and a posterior-adapted, unstructured, sample set. Overall, there is a need for efficient approaches with provable convergence properties—such that one can achieve exact sampling while making full use of the surrogate model.

[2] Typically the computational model itself is an approximation of some underlying governing equations. Though numerical discretization error can certainly affect the posterior (Kaipio and Somersalo, 2007), we do not address this issue here; we let a numerical implementation of the forward model, embedded appropriately in the likelihood



(a) Prior-based sparse grid samples.

(b) Posterior-adapted samples.

Figure 1: Schematic of an inference problem with a Gaussian prior and a posterior concentrated therein, with two experimental design approaches superimposed. Points are locations in the parameter space where the forward model is evaluated.

1.1 Our contribution

This work attempts to resolve the above-mentioned issues by proposing a new framework that integrates *local* approximations into Metropolis-Hastings kernels, producing a Markov chain that asymptotically (in the number of MCMC steps) samples from the *exact* posterior distribution. As examples of this approach, we will employ approximations of either the log-likelihood function or the forward model, using local linear, quadratic, or Gaussian process regression. To produce the sample sets used for these local approximations, we will introduce a sequential experimental design procedure that interleaves infinite refinement of the approximation with the Markov chain's exploration of the posterior. The overall experimental design reflects a combination of guidance from MCMC (so that samples are focused on the posterior) and local space filling heuristics (to ensure good quality sample sets for local approximation), triggered both by random refinement and by local error indicators of approximation quality. The result is a practical approach that also permits rigorous error analysis. This concept is inspired by the use of local approximations in *trust region methods* for derivative-free optimization (Conn et al., 2000, 2009), wherein local models similarly allow the reuse of model evaluations while enabling refinement until convergence. Local approximations also

function, define the exact posterior of interest.

have a long history in the statistics literature (Cleveland, 1979; Friedman, 1991) and have recently been reintroduced as an important strategy for scaling Gaussian processes to large data contexts (Gramacy and Apley, 2013).

Local approximations are convergent under relatively straightforward conditions (compared to global approximations), and we use this property to prove that the resulting MCMC algorithm converges asymptotically to the posterior distribution induced by the exact forward model and likelihood. Our proof involves demonstrating that the transition kernel converges quickly as the posterior distribution is explored and as the surrogate is refined; our theoretical analysis focuses on the specific case of a random-walk Metropolis algorithm coupled with local quadratic approximations of the log-posterior density. Our arguments are not limited to the random-walk Metropolis algorithm, however; they apply quite broadly and can be adapted to many other Metropolis-Hastings algorithms and local approximation schemes. Broadly, our theoretical results reinforce the notion that *it is possible to greatly reduce the number of evaluations of the forward model per MCMC step when the likelihood has some local regularity.* We complement the theory by demonstrating experimental performance improvements of up to several orders of magnitude on inference problems involving ordinary differential equation and partial differential equation forward models, with no discernable loss in accuracy, using several different MCMC algorithms and local approximation schemes.

We note that our theoretical results are asymptotic in nature; in this paper, we do not focus on finite-time error bounds. While we can comment on such bounds in a few specific settings, obtaining more general quantitative estimates for the finite-time bias of the algorithm is a significant challenge and will be tackled elsewhere. Nevertheless, we argue that asymptotic convergence is quite useful for practitioners, as it supports how the algorithm is actually applied. Since the aim of our approach is to reduce the use of the forward model, it is natural to ask how many model runs would be necessary to construct an MCMC chain that yields estimates with a certain error. We cannot *a priori* answer this question, just as we cannot (in general) say in advance how long it will take any other MCMC algorithm to reach stationarity. Yet asymptotic convergence makes our algorithm comparable to standard MCMC algorithms in practice: iterations continue until MCMC diagnostics suggest that the chain, and hence the underlying approximation, is sufficiently converged for the application. The cost of running the forward model is accumulated incrementally as the MCMC chain is extended, in a way that balances the error of the finite chain with the error introduced by the approximation. Moreover, this process may be interrupted at any time. This approach to posterior sampling stands in contrast with existing non-convergent methods, where the cost of constructing the approximation is incurred *before* performing inference, and where the user must carefully balance the error induced by the approximation with the MCMC sampling error, without any rigorous strategy for doing so.

The remainder of this paper is organized as follows. We describe the new MCMC approach in Section 2. Theoretical results on asymptotically exact sampling are provided in Section 3; proofs of these theorems are deferred to Appendix B. Section 4 then provides empirical assessments of performance in several examples. We emphasize that, while the examples demonstrate strong computational performance, the present implementation is merely a representative of a class of asymptotically exact MCMC algorithms. Therefore, Section 5 discusses several variations on the core algorithm that may be pursued in future work. A reusable implementation of the algorithm described is available as part of the MIT Uncertainty Quantification Library, `https://bitbucket.org/mituq/muq/`.

2 Metropolis-Hastings with local approximations

This section describes our framework for Metropolis-Hastings algorithms based on local approximations, which incrementally and infinitely refine an approximation of the forward model or likelihood as inference is performed.

2.1 Algorithm overview

Consider a Bayesian inference problem with posterior density

$$p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta|\mathbf{d}, \mathbf{f})p(\theta),$$

for inference parameters $\theta \in \Theta \subseteq \mathbb{R}^d$, data $\mathbf{d} \in \mathbb{R}^n$, forward model $\mathbf{f} : \Theta \to \mathbb{R}^n$, and probability densities specifying the prior $p(\theta)$ and likelihood function $\mathcal{L}$. The forward model may enter the likelihood function in various ways. For instance, if $\mathbf{d} = \mathbf{f}(\theta) + \eta$, where $\eta \sim p_\eta$ represents some measurement or model error, then $\mathcal{L}(\theta|\mathbf{d}, \mathbf{f}) = p_\eta(\mathbf{d} - \mathbf{f}(\theta))$.

A standard approach is to explore this posterior with a Metropolis-Hastings algorithm using a suitable proposal kernel L, yielding the Metropolis-Hastings transition kernel $K_\infty(X_t, \cdot)$; existing MCMC theory governs the correctness and performance of this approach (Roberts and Rosenthal, 2004). For simplicity, assume that the kernel L is translation-invariant and symmetric.[3] We assume

[3] Assuming symmetry simplifies our discussion, but the generalization to non-symmetric proposals is straightforward. Extensions to translation-dependent kernels, *e.g.,* the Metropolis-adjusted Langevin algorithm, are also possible (Conrad, 2014).

that the forward model evaluation is computationally expensive—requiring, for example, a high-resolution numerical solution of a partial differential equation (PDE). Also assume that drawing a proposal is inexpensive, and that given the proposed parameters and the forward model evaluation, the prior density and likelihood are similarly inexpensive to evaluate, *e.g.*, Gaussian. In such a setting, the computational cost of MCMC is dominated by the cost of forward model evaluations required by $K_\infty(X_t, \cdot)$.[4]

Previous work has explored strategies for replacing the forward model with some cheaper approximation, and a typical scheme works as follows (Rasmussen, 2003; Bliznyuk et al., 2012; Marzouk et al., 2007). Assume that one has a collection of model evaluations, $\mathcal{S} := \{(\theta, \mathbf{f}(\theta))\}$, and a method for constructing an approximation $\tilde{\mathbf{f}}$ of $\mathbf{f}$ based on those examples. This approximation can be substituted into the computation of the Metropolis-Hastings acceptance probability. However, $\mathcal{S}$ is difficult to design in advance, so the algorithm is allowed to refine the approximation, as needed, by computing new forward model evaluations near the sample path and adding them to the growing sample set $\mathcal{S}_t$.

Our approach, outlined in Algorithm 1, is in the same spirit as these previous efforts. Indeed, the sketch in Algorithm 1 is sufficiently general to encompass both the previous efforts mentioned above and the present work. We write K_t to describe the evolution of the sampling process at time t in order to suggest the connection of our process with a time-inhomogeneous Markov chain; this connection is made explicit in Section 3. Intuitively, one can argue that this algorithm will produce accurate samples if $\tilde{\mathbf{f}}$ is close to $\mathbf{f}$, and that the algorithm will be efficient if the size of $\mathcal{S}_t$ is small and $\tilde{\mathbf{f}}$ is cheap to construct.

Our implementation of this framework departs from previous work in two important ways. First, rather than using global approximations constructed from the entire sample set $\mathcal{S}_t$, we construct local approximations that use only a nearby subset of $\mathcal{S}_t$ for each evaluation of $\tilde{\mathbf{f}}$, as in LOESS (Cleveland, 1979) or derivative-free optimization (Conn et al., 2009). Second, previous efforts usually halt the growth of $\mathcal{S}_t$ after a fixed number of refinements;[5] instead, we allow an infinite number of refinements to occur as the MCMC chain proceeds. Figure 2 depicts how the sample set might

[4] Identifying the appropriate target for approximation is critical to the performance of our approach, and depends upon the relative dimensionality, regularity, and computational cost of the various components of the posterior model. In most settings, the forward model is a clear choice because it contributes most of the computational cost, while the prior and likelihood may be computed cheaply without further approximation. The algorithm presented here may be adjusted to accommodate other choices by merely relabeling the terms. For another discussion of this issue, see Bliznyuk et al. (2008).

[5] For example, Rasmussen (2003) and Bliznyuk et al. (2012) only allow refinements until some fixed time $T_{\text{ref}} < T$, and polynomial chaos expansions are typically constructed in advance, omitting refinement entirely (Marzouk et al., 2007).

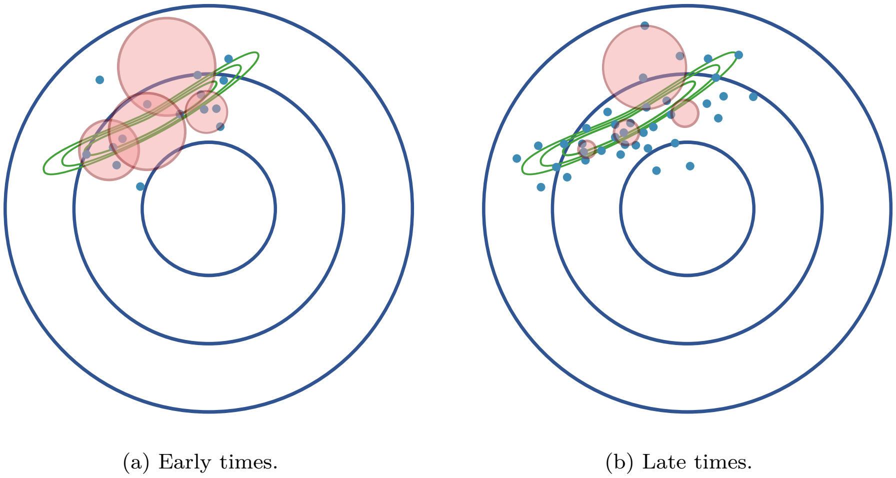
Algorithm 1 Sketch of approximate Metropolis-Hastings algorithm

```
 1: procedure RunChain($\theta_1, \mathcal{S}_1, \mathcal{L}, \mathbf{d}, p, \mathbf{f}, L, T$)
 2:     for $t = 1 \ldots T$ do
 3:         $(\theta_{t+1}, \mathcal{S}_{t+1}) \leftarrow K_t(\theta_t, \mathcal{S}_t, \mathcal{L}, \mathbf{d}, p, \mathbf{f}, L)$
 4:     end for
 5: end procedure

 6: procedure $K_t(\theta^-, \mathcal{S}, \mathcal{L}, \mathbf{d}, p, \mathbf{f}, L)$
 7:     Draw proposal $\theta^+ \sim L(\theta^-, \cdot)$
 8:     Compute approximate models $\tilde{\mathbf{f}}^+$ and $\tilde{\mathbf{f}}^-$, valid near $\theta^+$ and $\theta^-$
 9:     Compute acceptance probability $\alpha \leftarrow \min\left(1, \frac{\mathcal{L}(\theta|\mathbf{d},\tilde{\mathbf{f}}^+)p(\theta^+)}{\mathcal{L}(\theta|\mathbf{d},\tilde{\mathbf{f}}^-)p(\theta^-)}\right)$
10:     if approximation needs refinement near $\theta^-$ or $\theta^+$ then
11:         Select new point $\theta^*$ and grow $\mathcal{S} \leftarrow \mathcal{S} \cup (\theta^*, \mathbf{f}(\theta^*))$. Repeat from Line 8.
12:     else
13:         Draw $u \sim \text{Uniform}(0,1)$. If $u < \alpha$, return $(\theta^+, \mathcal{S})$, else return $(\theta^-, \mathcal{S})$.
14:     end if
15: end procedure
```

evolve as the algorithm is run, becoming denser in regions of higher posterior probability, allowing the corresponding local approximations to use ever-smaller neighborhoods and thus to become increasingly accurate. Together, these two changes allow us to construct an MCMC chain that, under appropriate conditions, asymptotically samples from the exact posterior. Roughly, our theoretical arguments (in Section 3 and Appendix B) will show that refinements of the sample set $\mathcal{S}_t$ produce a convergent approximation $\tilde{\mathbf{f}}$ and hence that K_t converges to the standard "full model" Metropolis kernel K_∞ in such a way that the chain behaves as desired. Obviously, we require that $\mathbf{f}$ be sufficiently regular for local approximations to converge. For example, when using local quadratic approximations, it is sufficient (but not necessary) for the Hessian of $\mathbf{f}$ to be Lipschitz continuous (Conn et al., 2009).

The remainder of this section expands this outline into a usable algorithm, detailing how to construct the local approximations, when to perform refinement, and how to select new points to refine the approximations. Section 2.2 describes how to construct local linear or quadratic models and outlines the convergence properties that make them useful. Section 2.3 explains when to trigger refinement, either randomly or based on a cross validation error indicator. Section 2.4 explains how to refine the approximations by evaluating the full model at a new point chosen using a space filling experimental design. Finally, Section 2.5 explains the changes required to substitute local Gaussian process approximations for polynomial approximations.


(a) Early times.

(b) Late times.

Figure 2: Schematic of the behavior of local approximations as the algorithm proceeds on the example from Figure 1. The balls are centered at locations where local models might be needed and the radius indicates the size of the sample set; the accuracy of local models generally increases as this ball size shrinks. At early times the sample set is sparse and the local approximations are built over relatively large balls, implying that their accuracy is limited. At later times refinements enrich the sample set near regions of high posterior density, allowing the local models to shrink and become more accurate.

2.2 Local polynomial approximation

This section describes how to construct local linear or quadratic models. We construct these models using samples from $\mathcal{S}$ drawn from a ball of radius R centered on θ, $\mathcal{B}(\theta, R) := \{(\theta_i, \mathbf{f}(\theta_i)) \in \mathcal{S} : \|\theta_i - \theta\|_2 \leq R\}$. If this set contains a sufficient number of samples, local polynomial models may easily be fit using least squares regression. We write the operators that produce such linear or quadratic approximations as $\mathcal{L}^{\sim j}_{\mathcal{B}(\theta,R)}$ or $\mathcal{Q}^{\sim j}_{\mathcal{B}(\theta,R)}$, respectively. The superscript $\sim j$, if non-empty, indicates that sample j should be omitted; this option is used to support cross-validation error indicators, described below.

It can be shown that the following error bounds hold independently for linear or quadratic approximations of each output component $i = 1 \ldots n$, for every point within the ball, $\theta' : \|\theta' - \theta\|_2 \leq R$ (Conn et al., 2009), assuming that the gradient or Hessian of $\mathbf{f}$ is Lipschitz continuous, respectively:

$$\left| f_i(\theta') - \left(\mathcal{L}^{\sim j}_{\mathcal{B}(\theta,R)}(\theta') \right)_i \right| \quad \leq \quad \kappa_l(\nu_1, \lambda, d) R^2, \tag{1a}$$

$$\left| f_i(\theta') - \left(\mathcal{Q}^{\sim j}_{\mathcal{B}(\theta,R)}(\theta') \right)_i \right| \quad \leq \quad \kappa_q(\nu_2, \lambda, d) R^3. \tag{1b}$$

where the constants κ are functions of the Lipschitz constants $\nu_1, \nu_2 < \infty$ of the gradient or Hessian of $\mathbf{f}$, respectively; a "poisedness" constant λ reflecting the geometry of the input sample set; and the parameter dimension d. Intuitively, λ is small if the points are well separated, fill the ball from which they are drawn, and do not lie near any linear or quadratic paths (for the linear and quadratic approximations, respectively). As long as λ is held below some fixed finite value, the model is said to be λ-*poised*, and these bounds show that the approximations converge as $R \to 0$.[6] These simple but rigorous local error bounds form the foundation of our theoretical analysis, and are the reason that we begin with local polynomial approximations. Usefully, they are representative of the general case, in that most reasonable local models converge in some sense as the ball size falls to zero.

It remains to precisely specify the choice of radius, R, and the weights used in the least squares regression. The radius R is selected to include a fixed number of points N. A linear model is fully defined by $N_{\text{def}} = d + 1$ points and a quadratic is defined by $N_{\text{def}} = (d + 1)(d + 2)/2$ points; hence, performing a least squares regression requires at least this many samples. Such models are interpolating, but the associated least squares system is often poorly conditioned unless the geometry of the sample set is carefully designed. Conn et al. (2009) show that adding additional samples can only stabilize the regression problem, so we select $N = \sqrt{d}N_{\text{def}}$, which seems to work well in practice.[7]

We depart from Conn *et al.* (2009) by performing a weighted regression using a variation of the tricube weight function often used with LOESS (Cleveland, 1979). If the radius that contains the inner N_{def} samples is R_{def}, then $R > R_{\text{def}}$ and the weight of each sample is:

$$w_i = \begin{cases} 1 & \|\theta_i - \theta\|_2 \leq R_{\text{def}}, \\ 0 & \|\theta_i - \theta\|_2 > R, \\ \left(1 - \left(\frac{\|\theta_i - \theta\|_2 - R_{\text{def}}}{R - R_{\text{def}}}\right)^3\right)^3 & \text{else.} \end{cases} \tag{2}$$

Setting the inner points to have unity weight ensures that the regression is full rank, while subsequently decreasing the weights to zero puts less emphasis on more distant samples. An interesting side effect of using this weight function is that the global approximation $\tilde{\mathbf{f}}$ has two continuous derivatives, even though it is constructed independently at each point (Atkeson et al., 1997).

[6] Although Conn et al. (2009) explicitly compute and control the value of λ, this step is not necessary in practice for our algorithm. The geometric quality of our sample sets is generally good because of the experimental design procedure we use to construct them. Also, we are less sensitive to poor geometry because we perform regression, rather than interpolation, and because the cross validation procedure described below considers geometric quality and can trigger refinement as needed.

[7] In very low dimensions, $\sqrt{d}$ provides very few extra samples and hence should be inflated. For $d = 6$, in the numerical experiments below, this exact form is used.

This process is described by the subroutine LocApprox in Algorithm 2, which produces an approximation at θ, using a fixed set of samples $\mathcal{S}$, optionally omitting sample j. The pseudocode uses $\mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$ to represent either polynomial fitting algorithm. Appendix A describes the regression procedure and the numerical approach to the corresponding least squares problems in more detail. Multiple outputs are handled by constructing a separate approximation for each one. Fortunately, the expensive step of the least squares problem is identical for all the outputs, so the cost of constructing the approximation scales well with the number of observations.

Algorithm 2 Construct local approximation

```
procedure LocApprox($\theta, \mathcal{S}, j$)
    Select $R$ so that $|\mathcal{B}(\theta, R)| = N$, where
    $\mathcal{B}(\theta, R) := \{(\theta_i, \mathbf{f}(\theta_i)) \in \mathcal{S} : \|\theta_i - \theta\|_2 \leq R\}$    ▷ Select ball of points
    $\tilde{\mathbf{f}} \leftarrow \mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$    ▷ Local approximation as defined in Section 2.2, possibly without sample $j$
    return $\tilde{\mathbf{f}}$
end procedure
```

2.3 Triggering model refinement

We separate the model refinement portion of the algorithm into two stages. This section discusses *when* refinement is needed, while Section 2.4 explains *how* the refinement is performed. The MCMC step uses local approximations at both θ^+ and θ^-, and either are candidates for refinement. We choose a refinement criteria that is symmetric, that is, which behaves identically if the labels of θ^+ and θ^- are reversed; by treating the two points equally, we aim to avoid adverse coupling with the decision of whether to accept a move.

Refinement is triggered by either of two criteria. The first is random: with probability β_t, the model refined at either the current point θ^- or the proposed point θ^+. This process fits naturally into MCMC and is essential to establishing the theoretical convergence results in the next section. The second criterion, based on a cross-validation error indicator, is intended to make the approximation algorithm efficient in practice. For a Metropolis-Hastings algorithm with a symmetric proposal, recall that the acceptance probability computed using the true forward model is

$$\alpha = \min\left(1, \frac{\mathcal{L}(\theta^+|\mathbf{d}, \mathbf{f})p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d}, \mathbf{f})p(\theta^-)}\right).$$

Since the acceptance probability is a scalar, and this equation is the only appearance of the forward model in the sampling algorithm, it is a natural target for an error indicator. We employ a leave-one-out cross validation strategy, computing the sensitivity of the acceptance probability to the omission of samples from each of the approximate models, producing scalar error indicators ϵ^+ and ϵ^-. Refinement is performed whenever one of these indicators exceed a threshold γ_t, at the point whose error indicator is larger.

To construct the indicators, begin by computing the ratio inside the acceptance probability, using the full sample sets and variations leaving out each sample, $j = 1, \ldots, N$.

$$\begin{aligned}
\zeta &:= \frac{\mathcal{L}(\theta^+|\mathbf{d}, \textsc{LocApprox}(\theta^+, \mathcal{S}, \emptyset))p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d}, \textsc{LocApprox}(\theta^-, \mathcal{S}, \emptyset))p(\theta^-)} \\
\zeta^{+,\sim j} &:= \frac{\mathcal{L}(\theta^+|\mathbf{d}, \textsc{LocApprox}(\theta^+, \mathcal{S}, j))p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d}, \textsc{LocApprox}(\theta^-, \mathcal{S}, \emptyset))p(\theta^-)} \\
\zeta^{-,\sim j} &:= \frac{\mathcal{L}(\theta^+|\mathbf{d}, \textsc{LocApprox}(\theta^+, \mathcal{S}, \emptyset))p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d}, \textsc{LocApprox}(\theta^-, \mathcal{S}, j))p(\theta^-)}
\end{aligned}$$

Next, find the maximum difference between the α computed using ζ and that computed using the leave-one-out variations $\zeta^{+,\sim j}$ and $\zeta^{-,\sim j}$. The error indicators consider the acceptance probability in both the forward and reverse directions, ensuring equivalent behavior under relabeling of θ^+ and θ^-; this prevents the cross validation process from having any impact on the reversibility of the transition kernel.

$$\epsilon^+ := \max_j \left(\left| \min\left(1, \zeta\right) - \min\left(1, \zeta^{+,\sim j}\right) \right| + \left| \min\left(1, \frac{1}{\zeta}\right) - \min\left(1, \frac{1}{\zeta^{+,\sim j}}\right) \right| \right) \tag{3}$$

$$\epsilon^- := \max_j \left(\left| \min\left(1, \zeta\right) - \min\left(1, \zeta^{-,\sim j}\right) \right| + \left| \min\left(1, \frac{1}{\zeta}\right) - \min\left(1, \frac{1}{\zeta^{-,\sim j}}\right) \right| \right) \tag{4}$$

We emphasize that the acceptance probability is a natural quantity of interest in this context; it captures the entire impact of the forward model and likelihood on the MH kernel. The cross-validation error indicator is easily computable, summarizes a variety of error sources, and is easily interpretable as an additive error in a probability. These features make it possible for the user to exercise a problem-independent understanding of the threshold to which it is compared, γ_t. In contrast, attempting to control the error in either the forward model outputs or log-likelihood at the current or proposed point is not generically feasible, as their scale and the sensitivity of the MH kernel to their perturbations cannot be known *a priori*.

Our two refinement criteria have different purposes, and both are useful to ensure a quick and accurate run. The cross validation criterion is a natural and efficient way to refine our estimates, and is the primary source of refinement during most runs. The random criterion is less efficient, but some random evaluations may be required for the algorithm to be asymptotically correct for all starting positions. Thus, we use both in combination. The two parameters β_t and γ_t are allowed to decrease over time, decreasing the rate of random refinement and increasing the stringency of the cross validation criterion; theory governing the rates at which they may decrease and guidance on choosing them in practice are discussed later.

2.4 Refining the local model

If refinement of the local model at a point θ is required, we perform refinement by selecting a single new nearby point θ^*, computing $\mathbf{f}(\theta^*)$, and inserting the new pair into $\mathcal{S}$. To be useful, this new model evaluation should improve the sample set for the local model $\mathcal{B}(\theta, R)$, either by allowing the radius R to decrease or by improving the local geometry of the sample set. Consider that MCMC will revisit much of the parameter space many times, hence our algorithm must ensure that local refinements maintain the global quality of the sample set, that is, the local quality at every nearby location.

Intuitively, local polynomial regression becomes ill-conditioned if the points do not fill the whole ball, or if some points are clustered much more tightly than others. The obvious strategy of simply adding θ to $\mathcal{S}$ is inadvisable because it often introduces tightly clustered points, inducing poorly conditioned regression problems. Instead, a straightforward and widely used type of experimental design is to choose points in a space-filling fashion; doing so near θ naturally fulfills our criteria. Specifically, we select the new point θ^* by finding a local maximizer of the problem:

$$\begin{aligned} \theta^* &= \operatorname*{arg\,max}_{\theta'} \min_{\theta_i \in \mathcal{S}} \|\theta' - \theta_i\|_2, \\ & \quad \text{subject to } \|\theta' - \theta\|_2 \leq R, \end{aligned}$$

where optimization iterations are initialized at $\theta' = \theta$. The constraint ensures that the new sample lies in the ball and thus can be used to improve the current model, and the inner minimization operator finds a point well separated from the entire set $\mathcal{S}$ in order to ensure the sample's global quality. Inspection of the constraints reveals that the inner minimization may be simplified to $\theta_i \in \mathcal{B}(\theta, 3R)$, as points outside a ball of radius $3R$ have no impact on the optimization. We seek a local optimum of the objective because it is both far easier to find than the global optimum, and is more likely to be useful: the global optimum will often be at radius R, meaning that the revised model cannot be built over a smaller ball. This strategy is summarized in Algorithm 3.

Algorithm 3 Refine a local approximation

```
procedure RefineNear($\theta, \mathcal{S}$)
    Select $R$ so that $|\mathcal{B}(\theta, R)| = N$                                    ▷ Select ball of points
    $\theta^* \leftarrow \arg\max_{\|\theta' - \theta\| \leq R} \min_{\theta_i \in \mathcal{S}} \|\theta' - \theta_i\|$    ▷ Optimize near $\theta$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\theta^*, \mathbf{f}(\theta^*)\}$                          ▷ Grow the sample set
    return $\mathcal{S}$
end procedure
```

Although there is a close relationship between the set of samples where the forward model is evaluated and the posterior samples that are produced by MCMC, they are distinct and in general the two sets do not overlap. A potential limitation of the space filling approach above is that it might select points outside the support of the prior. This is problematic only if the model is not feasible outside the prior, in which case additional constraints can easily be added.

2.5 Local Gaussian process surrogates

Gaussian process (GP) regression underlies an important and widely used class of computer model surrogates, so it is natural to consider its application in the present local approximation framework Sacks et al. (1989); Santner et al. (2003). Local Gaussian processes have been previously explored in (Vecchia, 1988; Cressie, 1991; Stein et al., 2004; Snelson and Ghahramani, 2007; Gramacy and Apley, 2013). This section explains how local Gaussian process approximations may be substituted for the polynomial approximations described above.

The adaptation is quite simple: we define a new approximation operator $\mathcal{G}_{\mathcal{S}}^{\sim j}$ that may be substituted for the abstract operator $\mathcal{A}_{\mathcal{B}(\theta,R)}^{\sim j}$ in Algorithm 2. The error indicators are computed much as before, except that we use the predictive distribution $\tilde{\mathbf{f}}(\theta) \sim \mathcal{N}(\mu(\theta), \sigma^2(\theta))$ instead of a leave-one-out procedure. We define $\mathcal{G}_{\mathcal{S}}^{\sim j}$ to be the mean of the local Gaussian process, $\mu(\theta)$, when $j = \emptyset$, and a draw from the Gaussian predictive distribution otherwise. This definition allows us to compute ϵ^+ and ϵ^- without further modification, using the posterior distribution naturally produced by GP regression.

Our implementation of GPs borrows heavily from Gramacy and Apley (2013), using a separable squared exponential covariance kernel (i.e., with a different correlation length ℓ_i for each input dimension) and an empirical Bayes approach to choosing the kernel hyperparameters, i.e., using optimization to find the mode of the appropriate posterior marginals. The variance is endowed with an inverse-gamma hyperprior and a MAP estimate is found analytically, while the correlation lengths and nugget are endowed with gamma hyperpriors whose product with the marginal likelihood is maximized numerically. Instead of constructing the GP only from nearest neighbors $\mathcal{B}(\theta, R)$, we use a subset of $\mathcal{S}$ that mostly lies near the point of interest but also includes a few samples further away. This combination is known to improve surrogate quality over a pure nearest-neighbor strategy (Gramacy and Apley, 2013). We perform a simple approximation of the strategy developed by Gramacy and Apley: beginning with a small number of the nearest points, we estimate the hyperparameters and then randomly select more neighbors to introduce into the set, where the existing samples are weighted by their distance under the norm induced by the current length scales. This process is repeated in several batches, until the desired number of samples is reached. We are relatively unconstrained in choosing the number of samples N; in the numerical examples to be shown later, we choose $N = d^{5/2}$, mimicking the choice for quadratic approximations. Multiple outputs are handled with separate predictive distributions, but the hyperparameters are jointly optimized.[8]

2.6 Algorithm summary

Our Metropolis-Hastings approach using local approximations is summarized in Algorithm 4. The algorithm proceeds in much the same way as the sketch provided in Algorithm 1. It is general enough to describe both local polynomial and Gaussian process approximations, and calls several routines developed in previous sections. The chain is constructed by repeatedly constructing a new state with K_t.[9] This function first draws a proposal and forms the approximate acceptance probability. Then error indicators are computed and refinement is performed as needed, until finally the proposal is accepted or rejected.

3 Theoretical results

In this section we show that, under appropriate conditions, the following slightly modified version of Algorithm 4 converges to the target posterior $p(\theta|\mathbf{d})$ asymptotically:

1. The sequence of parameters $\{\beta_t\}_{t\in\mathbb{N}}$ used in that algorithm are of the form $\beta_t \equiv \beta > 0$. Our results hold with essentially the same proof if we use any sequence $\{\beta_t\}_{t\in\mathbb{N}}$ that satisfies

[8] Choosing an optimal number of samples is generally challenging, and we do not claim that this choice of N is the most efficient. Rather, it is the same scaling that we use for local quadratic approximations, and appears to work well for GP approximation in the range where we have applied it. For very low d, however, this N may need to be increased.

[9] Before MCMC begins, $\mathcal{S}_1$ needs to be seeded with a sufficient number of samples for the first run. Two simple strategies are to draw these samples from the prior, or else near the MCMC starting point, which is often the posterior mode as found by optimization.

Algorithm 4 Metropolis-Hastings with local approximations

```
 1: procedure RunChain($\mathbf{f}, L, \theta_1, \mathcal{S}_1, \mathcal{L}, \mathbf{d}, p, T, \{\beta_t\}_{t=1}^T, \{\gamma_t\}_{t=1}^T$)
 2:     for $t = 1 \ldots T$ do
 3:         $(\theta_{t+1}, \mathcal{S}_{t+1}) \leftarrow K_t(\theta_t, \mathcal{S}_t, \mathcal{L}, \mathbf{d}, p, \mathbf{f}, L, \beta_t, \gamma_t)$
 4:     end for
 5: end procedure

 6: procedure $K_t(\theta^-, \mathcal{S}, \mathcal{L}, \mathbf{d}, p, \mathbf{f}, L, \beta_t, \gamma_t)$
 7:     Draw proposal $\theta^+ \sim L(\theta^-, \cdot)$
 8:     $\tilde{\mathbf{f}}^+ \leftarrow$ LocApprox($\theta^+, \mathcal{S}, \emptyset$)                ▷ Compute nominal approximations
 9:     $\tilde{\mathbf{f}}^- \leftarrow$ LocApprox($\theta^-, \mathcal{S}, \emptyset$)
10:     $\alpha \leftarrow \min\left(1, \frac{\mathcal{L}(\theta|\mathbf{d}, \tilde{\mathbf{f}}^+)p(\theta^+)}{\mathcal{L}(\theta|\mathbf{d}, \tilde{\mathbf{f}}^-)p(\theta^-)}\right)$           ▷ Compute nominal acceptance ratio
11:     Compute $\epsilon^+$ and $\epsilon^-$ as in Equations 3-4.
12:     if $u \sim$ Uniform$(0,1) < \beta_t$ then                  ▷ Refine with probability $\beta_t$
13:         Randomly, $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$) or $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
14:     else if $\epsilon^+ \geq \epsilon^-$ and $\epsilon^+ \geq \gamma_t$ then          ▷ If needed, refine near the larger error
15:         $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$)
16:     else if $\epsilon^- > \epsilon^+$ and $\epsilon^- \geq \gamma_t$ then
17:         $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
18:     end if
19:     if refinement occured then repeat from Line 8.
20:     else                                                   ▷ Evolve chain using approximations
21:         Draw $u \sim$ Uniform$(0,1)$. If $u < \alpha$, return $(\theta^+, \mathcal{S})$, else return $(\theta^-, \mathcal{S})$.
22:     end if
23: end procedure
```

$\sum_t \beta_t = \infty$. Example B.13 in Appendix B shows that this is sharp: if $\sum_t \beta_t < \infty$, the algorithm can have a positive probability of failing to converge asymptotically, regardless of the sequence $\{\gamma_t\}_{t\in\mathbb{N}}$.

2. The approximation of $\log p(\theta|\mathbf{d})$ is made via quadratic interpolation on the $N = N_{\text{def}}$ nearest points. We believe this to be a representative instantiation of the algorithm; similar results can be proved for other approximations of the likelihood function.

3. The sub-algorithm RefineNear is replaced with:

$$\textsc{RefineNear}(\theta, \mathcal{S}) = \textbf{return}(\mathcal{S} \cup \{(\theta, f(\theta))\}).$$

This assumption substantially simplifies and shortens our argument, without substantially impacting the algorithm.

4. We fix a constant $0 < \lambda < 1$. In step 14, immediately before the word **then**, we add '**or**, for $\mathcal{B}(\theta^+, R)$ as defined in the subalgorithm LocApprox($\theta^+, \mathcal{S}, \emptyset$) used in step 8, the collection of points $\mathcal{B}(\theta^+, R) \cap \mathcal{S}$ is not λ-poised'. We add the same check, with θ^- replacing θ^+ and

'step 9' replacing 'step 8', in step 16. The concept of poisedness is defined in (Conn et al., 2009), but the details are not required to read this proof. This additional check is needed for our approximate algorithm to 'inherit' a one-step drift condition from the 'true' algorithm. Empirically, we have found that this check rarely triggers refinement for sensible values of λ.

3.1 Assumptions

We now make some general assumptions and fix notation that will hold throughout this section and in Appendix B. Denote by $\{X_t\}_{t \in \mathbb{N}}$ a version of the stochastic process on $\Theta \subset \mathbb{R}^d$ defined by this modified version of Algorithm 4. Let $L(x, \cdot)$ be the kernel on $\mathbb{R}^d$ used to generate new proposals in Algorithm 4, $\ell(x, y)$ denote its density, and L_t be the point proposed at time t in Algorithm 4. Let $K_\infty(x, \cdot)$ be the MH kernel associated with proposal kernel L and target distribution $p(\theta|\mathbf{d})$. Assume that, for all measurable $A \subset \Theta$, we can write $K_\infty(x, A) = r(x)\delta_x(A) + (1 - r(x)) \int_{y \in A} p(x, y) dy$ for some $0 \leq r(x) \leq 1$ and density $p(x, y)$. Also assume that $L(x, \cdot)$ satisfies

$$L(x, S) = L(x + y, S + y) \tag{5}$$

for all points $x, y \in \Theta$ and all measurable sets $S \subset \Theta$.

Denote by $\mathcal{S}_t$ the collection of points in $\mathcal{S}$ from Algorithm 4 at time t, denote by $R = R_t$ the value of R_{def} at time t, and denote by $q_t^1, \ldots, q_t^N$ the points in $\mathcal{S}_t$ within distance R_t of X_t.

We define the *Gaussian envelope condition*:

Assumption 3.1. *There exists some positive definite matrix $[a_{ij}]$ and constant $0 < G < \infty$ so that the distribution*

$$\log p_\infty(\theta_1, \theta_2, \ldots, \theta_d) = - \sum_{1 \leq i \leq j \leq d} a_{ij} \theta_i \theta_j$$

satisfies

$$\lim_{r \to \infty} \sup_{\|\theta\| \geq r} |\log p(\theta|\mathbf{d}) - \log p_\infty(\theta)| < G. \tag{6}$$

For $\theta \in \Theta$, define the *Lyapunov function*

$$V(\theta) = \frac{1}{\sqrt{p_\infty(\theta)}}. \tag{7}$$

Assumption 3.2. *The proposal kernel L and the density $p_\infty(\theta)$ satisfy the following:*

1. *For all compact sets $\mathcal{A}$, there exists $\epsilon = \epsilon(\mathcal{A})$ so that $\inf_{y \in \mathcal{A}} \ell(0, y) \geq \epsilon > 0$.*
2. *There exist constants $C, \epsilon_0, x_0 \geq 0$ so that $\ell(0, x) \leq C p_\infty(x)^{\frac{1}{1+\epsilon_0}}$ for all $\|x\| \geq x_0$.*
3. *The Metropolis-Hastings Markov chain Z_t with proposal kernel L and stationary density p_∞ satisfies the drift condition*

$$\mathbb{E}[V(Z_{t+1})|Z_t = x] \leq \alpha V(x) + b$$

   for some $0 \leq \alpha < 1$ and some $0 \leq b < \infty$.

Before giving the main result, we briefly discuss the assumptions above.

1. Assumption 3.1 is quite strong. It is chosen as a representative sufficient condition for convergence of our algorithm on unbounded state spaces primarily because it is quite easy to state and to check. The assumption is used only to guarantee that our approximation of the usual MH chain inherits a drift condition (*i.e.* so that Lemma B.9 of Appendix B holds), and may be replaced by other assumptions that provide such a guarantee. We give some relevant alternative assumptions at the end of Appendix B. In particular, instead of Assumption 3.1, if we assume that the posterior $p(\theta|\mathbf{d})$ has sub-Gaussian tails with bounded first and second derivatives, our methods can be reworked to show the ergodicity of a slight modification of Algorithm 4.

   Although Assumption 3.1 is very strong, it does hold for one important class of distributions: mixtures of Gaussians for which one mixture component has the largest variance. That is, the condition holds if $p(\theta|\mathbf{d})$ is of the form $\sum_{i=1}^k \alpha_i \mathcal{N}(\mu_i, \Sigma_i)$ for some weights $\sum_{i=1}^k \alpha_i = 1$, some means $\{\mu_i\}_{i=1}^k \in \mathbb{R}^d$, and some $d \times d$ covariance matrices $\{\Sigma_i\}_{i=1}^k$ that satisfy $v^\top \Sigma_1 v > v^\top \Sigma_i v$ for all $0 \neq v \in \mathbb{R}^d$ and all $i \neq 1$.

2. Assumption 3.2 holds for a very large class of commonly used Metropolis-Hastings algorithms (see, *e.g.*, Roberts and Tweedie (1996) for sufficient conditions for item 3 of Assumption 3.2.)

3.2 Ergodicity

Here we state our main theorems on the convergence of the version of Algorithm 4 introduced in this section. Proofs are given in Appendix B.

Theorem 3.3. *Suppose Assumption 3.2 holds. There exists some $G_0 = G_0(L, p_\infty, \lambda, N)$ so that if assumption 3.1 holds with $0 < G < G_0 < \infty$, then for any starting point $X_0 = x \in \Theta$, we have*

$$\lim_{t\to\infty} \|\mathcal{L}(X_t) - p(\theta|\mathbf{d})\|_{\mathrm{TV}} = 0.$$

If we assume that Θ is compact, the same conclusion holds under much weaker assumptions:

Theorem 3.4. *Suppose Θ is compact and that both $p(\theta|\mathbf{d})$ and $\ell(x, y)$ are bounded away from 0 and infinity. Then*

$$\lim_{t\to\infty} \|\mathcal{L}(X_t) - p(\theta|\mathbf{d})\|_{\mathrm{TV}} = 0.$$

Remark 3.5. *We focus only on ergodicity, and in particular, do not obtain rates of convergence, laws of large numbers, or central limit theorems. We believe that, using results from the adaptive MCMC literature (see Fort et al. (2012)), the law of large numbers and central limit theorem can be shown to hold for the Monte Carlo estimator from our algorithm. A significantly more challenging issue is to quantify the bias-variance tradeoff of our algorithm and its impact on computational effort. We plan to study this issue in a forthcoming paper.*

4 Numerical experiments

Although the results in Section 3 and further related results in Appendix B establish the asymptotic exactness of our MCMC framework, it remains to demonstrate that it performs well in practice. This section describes three examples in which local surrogates produce accurate posterior samples using dramatically fewer evaluations of the forward model than standard MCMC. Additionally, these examples explore parameter tuning issues and the performance of several algorithmic variations. Though certain aspects of these examples depart from the assumptions of Theorems 3.3 or 3.4, the discussion in Appendix B.6 suggests that the theory is extensible to these cases; the success of the numerical experiments below reinforces this notion.

For each of these examples, we consider the accuracy of the computed chains and the number of forward model evaluations used to construct them. In the absence of analytical characterizations of the posterior, the error in each chain is estimated by comparing the posterior covariance estimates computed from a reference MCMC chain—composed of multiple long chains computed *without* any approximation—to posterior covariance estimates computed from chains produced by Algorithm 4.

The forward models in our examples are chosen to be relatively inexpensive in order to allow the construction of such chains and hence a thorough comparison with standard samplers. Focusing on the number of forward model evaluations is a problem-independent proxy for the overall running time of the algorithm that is representative of the algorithm's scaling as the model cost becomes dominant.

The first example uses an exponential-quartic distribution to investigate and select tunings of the refinement parameters β_t and γ_t. The second and third examples investigate the performance of different types of local approximations (linear, quadratic, and Gaussian process) when inferring parameters for an ODE model of a genetic circuit and the diffusivity field in an elliptic PDE, respectively. We conclude with some brief remarks on the performance and scaling of our implementation.

4.1 Exponential-quartic distribution

To investigate tunings of the the refinement parameters β_t and γ_t, we consider a simple two dimensional target distribution, with log-density

$$\log p(\theta) = -\frac{1}{10}\theta_1^4 - \frac{1}{2}(2\theta_2 - \theta_1^2)^2,$$

illustrated in Figure 3. Performing MCMC directly on this model is of course very inexpensive, but we may still consider whether local quadratic approximations can reduce the number of times the model must be evaluated. For simplicity, we choose the proposal distribution to be a Gaussian random walk with variance tuned to $\sigma^2 = 4$.

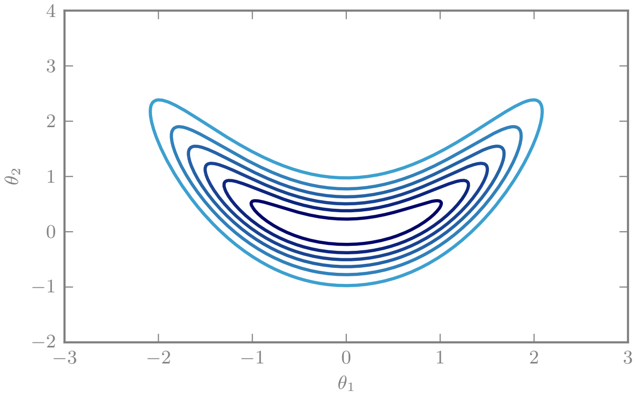
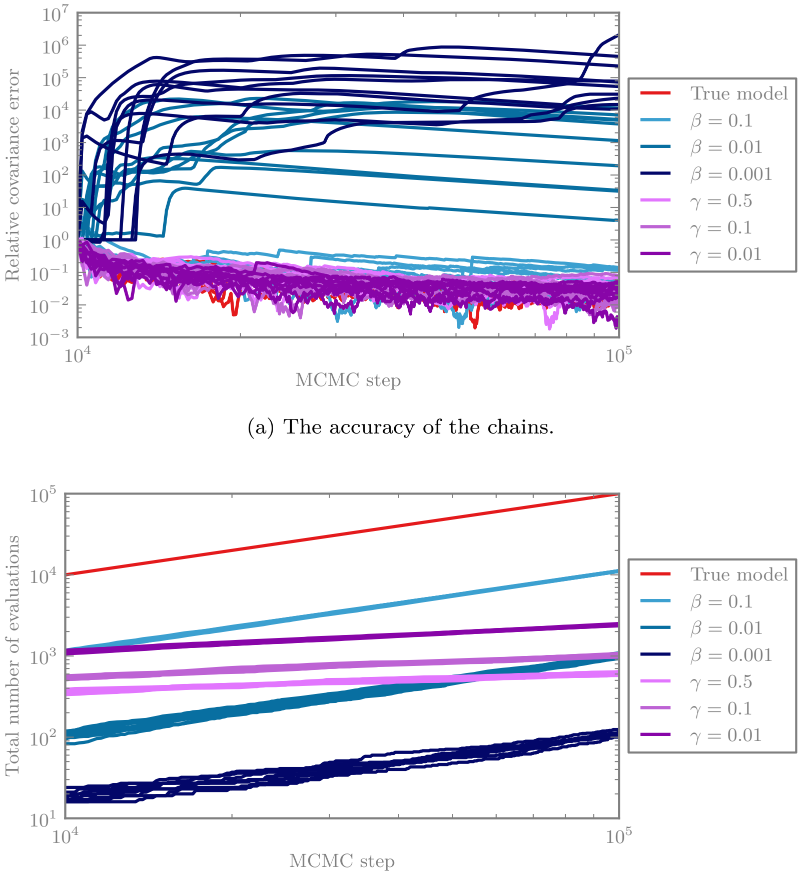


Figure 3: The logarithm of the target density in the exponential-quartic example.

As a first step towards understanding the response of our approach to β_t and γ_t, we test several constant values, setting only one of β_n or γ_n to be nonzero, choosing from $\beta_n \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ and $\gamma_n \in \{10^{-2}, 10^{-1}, 0.5\}$. With these settings, we run Algorithm 4, using local quadratic approximations of the log-target density.

The baseline configuration to which we compare Algorithm 4 comprises 30 chains, each run for 10^5 MCMC steps using the true forward model (*i.e.*, with no approximation). In all of the numerical experiments below, we discard the first 10% of a chain as burn-in. The reference runs are combined to produce a "truth" covariance, to which we compare the experiments. The chains are all initialized at the same point in the high target density region. Ten independent chains are run for each parameter setting, with each chain containing 10^5 MCMC steps. After discarding 10^4 burn-in samples for each chain, we consider the evolution of the error as the chain lengthens; we compute a relative error measure at each step, consisting of the Frobenius norm of the difference in covariance estimates, divided by the Frobenius norm of the reference covariance.



(a) The accuracy of the chains.

(b) The cost of the chains.

Figure 4: The accuracy and cost of sampling the exponential-quartic example using constant refinement parameters.

This accuracy comparison is summarized in Figure 4a, which shows the evolution of the error with the number of MCMC steps. The corresponding computational costs are summarized in Figure 4b, which shows the number of true model evaluations performed for any given number of MCMC steps. The distribution of errors obtained with the baseline chains, shown in red, reflects both the finite accuracy of the reference chain and the variance resulting from finite baseline chain lengths. As expected, the cost of a chain increases when β_t is larger or γ_t is smaller; these values trigger more frequent random refinements or more strictly constrain the acceptance probability error indicator, respectively. When β-refinement is set to occur at a very low rate, the resulting chain is inexpensive but of low accuracy, and in contrast, higher values of β show increased cost and reduced errors. The theory suggests that any constant $\beta_t > 0$ should yield eventual convergence, but this difference in finite time performance is not surprising. Even the $\beta_t = 0.01$ chains eventually show a steady improvement in accuracy over the interval of chain lengths considered here, which may reflect the predicted asymptotic behavior. Our experiments also show the efficacy of cross validation: all the chains using cross-validation refinement have accuracies comparable to the baseline runs while making significantly reduced use of the true model. These accuracies seem relatively insensitive to the value of γ.

In practice, we use the two criteria jointly and set the parameters to decay with t. Allowing β_t to decay is a cost-saving measure, and is theoretically sound as long as $\sum_t \beta_t$ diverges; on the other hand, setting γ_t to decay increases the stringency of the cross validation criterion, improving robustness. Based upon our experimentation, we propose to use parameters $\beta_t = 0.01t^{-0.2}$ and $\gamma_t = 0.1t^{-0.1}$; this seems to be a robust choice, and we use it for the remainder of the experiments.

Figure 5 summarizes the accuracy and cost of these parameter settings, and also considers the impact of a faster decay for the cross validation criterion: $\gamma_t = 0.1t^{-0.6}$. The proposed parameters yield estimates that are comparable in accuracy to the standard algorithm, but cheaper (shifted to the left) by nearly two orders of magnitude. Observe that tightening γ_t more quickly does not improve accuracy, but does increase the cost of the chains.

Before concluding this example, we explore the behavior of the refinement scheme in more detail. Figure 6 shows that under the proposed settings, though most refinements are triggered by cross validation, a modest percentage are triggered randomly; we propose that this is a useful balance because it primarily relies on the apparent robustness of cross validation, but supplements it with the random refinements required for theoretical guarantees. Interestingly, even though the probability of random refinement is decreasing *and* the stringency of the cross-validation criterion is increasing,

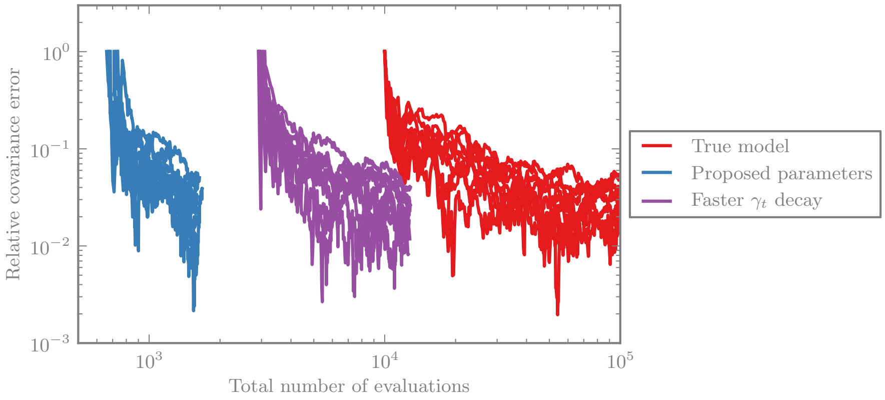


Figure 5: The accuracy of the inference as compared to the number of forward model evaluations required using the proposed parameters or a setting with faster γ_t decay. The plot depicts ten independent chains of each type, with the first 10% of each chain removed as burn-in.

the proportion of refinements triggered randomly is observed to increase. This behavior suggests that the local approximations are indeed becoming more accurate as the chains progress.

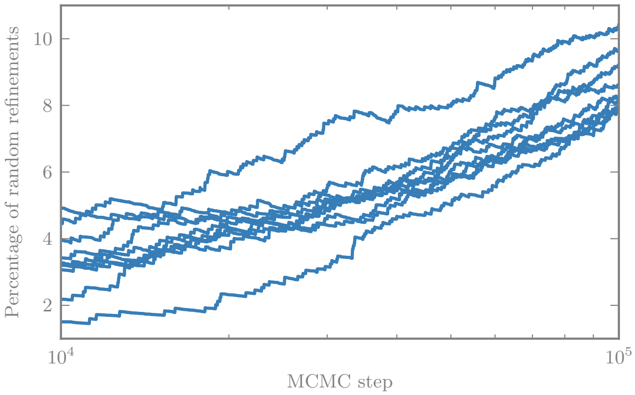
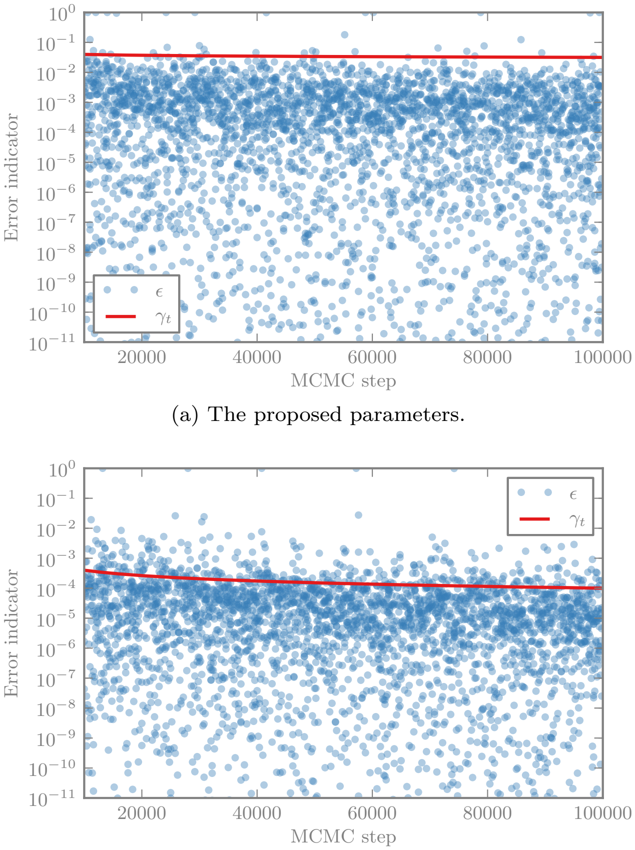


Figure 6: The percentage of refinements triggered by the random refinement criterion, for ten independent chains in the exponential-quartic example, using the proposed parameters.

Finally, it is instructive to directly plot the observed error indicators and compare them to the threshold used for refinement, as in Figure 7. Refinement occurs whenever the error indicators ϵ, denoted by circles, exceed the current γ_t. Comparing Figures 7a and 7b, we observe that many points lie just below the differing refinement thresholds, suggesting that choosing γ_t provides significant control over the behavior of the algorithm.



(a) The proposed parameters.

(b) Faster decay of γ_t.

Figure 7: The cross validation error indicator for the exponential-quartic example, using the proposed parameters or a faster γ_t decay. The indicator shown is $\epsilon = \max(\epsilon^+, \epsilon^-)$, computed before any refinement occurs. The error indicators are often much smaller than 10^{-11}—i.e., some proposals should obviously be accepted or rejected—but the plots are truncated to focus on behavior near the γ_t threshold.

4.2 Genetic toggle switch

Given the refinement parameters chosen in the previous example, we now consider the performance of several different types of local approximations in an ODE model with a compact parameter domain. We wish to infer the parameters of a genetic "toggle switch" synthesized in *E. coli* plasmids by Gardner et al. (2000), and previously used in an inference problem by Marzouk and Xiu (2009). Gardner et al. (2000) proposed a differential-algebraic model for the switch, with six unknown parameters $Z_\theta = \{\alpha_1, \alpha_2, \beta, \gamma, K, \eta\} \in \mathbb{R}^6$, while the data correspond to observations of the steady-state concentrations. As in Marzouk and Xiu (2009), the parameters are centered and scaled around their nominal values so that they can be endowed with uniform priors over the hypercube $[-1, 1]^6$. The measurement errors are independent and Gaussian, with zero mean and variances that depend on the experimental conditions. Further details on the problem setup are given in Appendix C. Figure 8 shows marginal posterior densities of the normalized parameters θ. These results broadly agree with Marzouk and Xiu (2009) and indicate that some directions are highly informed by the data while others are largely defined by the prior, with strong correlations among certain parameters.

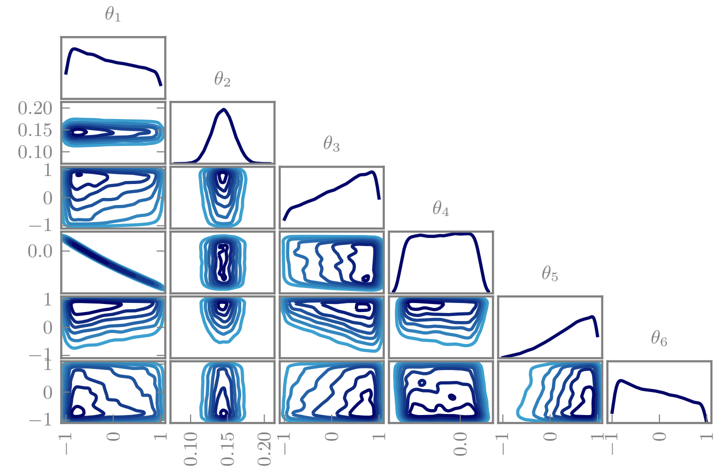


Figure 8: One- and two-dimensional posterior marginals of the six parameters in the genetic toggle switch.

We investigate the performance of three different local approximations of the forward model: linear, quadratic, and Gaussian process. The experiment proceeds as in the last section (Section 4.1), with two differences: first, we adapt the covariance of the Gaussian proposal using the adaptive Metropolis algorithm of Haario et al. (2001), a more practical choice than a fixed-size Gaussian random walk. Second, we limit our algorithm to perform at most two refinements per MCMC step, which is an *ad hoc* limit to the cost of any particular step. Figure 9 shows that the accuracy is nearly identical for all the cases, but the approximate chains use fewer evaluations of the true model, reducing costs by more than an order of magnitude for quadratic or Gaussian process approximations (Figure 10b). Local linear approximations show only modest improvements in the cost. Note that when proposals fall outside the support of the prior, the proposal is rejected without running either the true or approximate models; hence even the reference configuration runs the model less than once per MCMC step.

It is also instructive to plot the accuracy and cost as a function of the number of MCMC steps, as in Figure 10. All the accuracy trajectories in Figure 10a lie on top of each other, suggesting that the approximations do not have any discernable impact on the mixing time of the chain. Yet Figure 10b shows not only that the approximation strategies yield lower total cost at any given

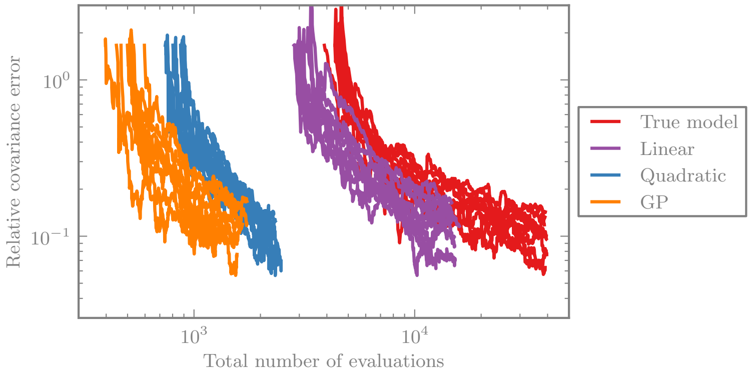


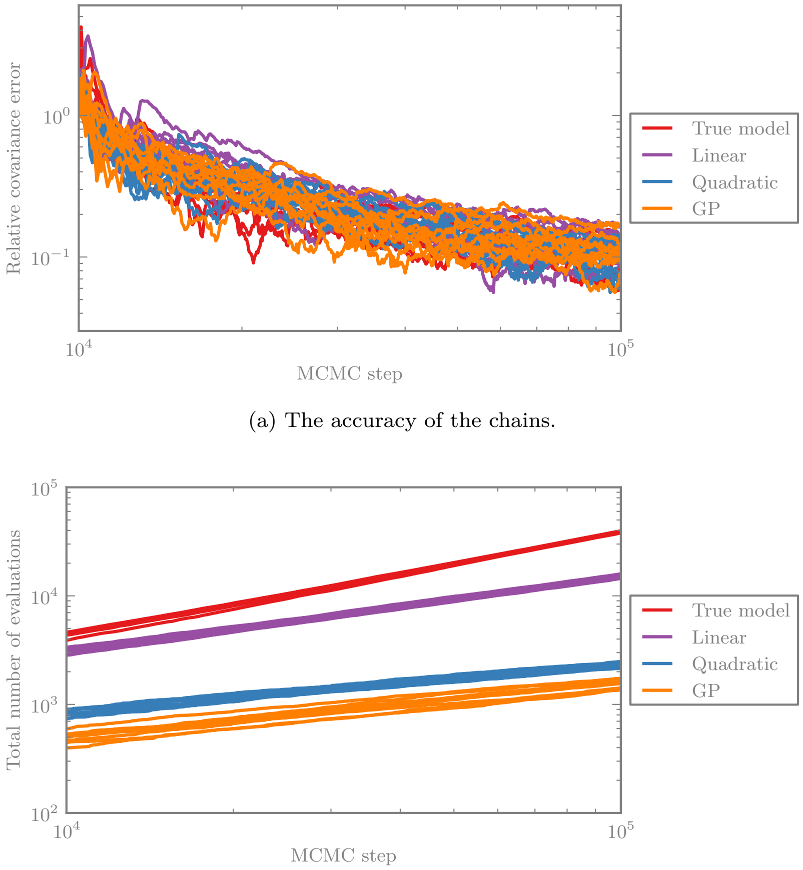
Figure 9: Approximate relative covariance errors in the MCMC chains versus their costs, for the genetic toggle switch problem, using several different local approximation strategies. The plot depicts ten independent chains of each type, with the first 10% of each chain removed as burn-in.

number of MCMC steps, but also that these costs accumulate at a *slower rate* than when the true model is used directly.

4.3 Elliptic PDE inverse problem

We now turn to a canonical inverse problem involving inference of the diffusion coefficient in an elliptic PDE (Dashti and Stuart, 2011). We leave the details of the PDE configuration to Appendix D; it suffices for our purposes that it is a linear elliptic PDE on a two-dimensional spatial domain, solved with a finite element algorithm at moderate resolution. The diffusion coefficient is defined by six parameters, each endowed with a standard normal prior. Noisy pointwise observations are taken from the solution field of the PDE and are relatively informative, and hence the posterior shifts and concentrates significantly with respect to the prior, as shown in Figure 11. We also emphasize that even though the PDE is linear, the forward model—*i.e.*, the map from the parameters to the observed field—is nonlinear and hence the posterior is not Gaussian. We also note that, while the design of effective posterior sampling strategies for functional inverse problems is an enormous and important endeavor (Cotter et al., 2013), our parameterization renders this problem relatively low-dimensional and the simple adaptive Metropolis sampler used to obtain our results mixes well.

Now we evaluate the performance of the various local approximation schemes, using the same experiments as in the previous section; results are summarized in Figure 12. As in the genetic toggle switch example, the accuracies of all the configurations are nearly indistinguishable, yet the approximate chains demonstrate significantly reduced use of the true forward model. Local linear approximations of the forward model decrease the cost by over an order of magnitude. Both the



(a) The accuracy of the chains.

(b) The cost of the chains.

Figure 10: Approximate relative covariance errors in the MCMC chains and their costs, shown over the length of the MCMC chain, for the genetic toggle switch problem, using several different local approximation strategies. The plot depicts ten independent chains of each type, with the first 10% of each chain removed as burn-in.

local quadratic and local GP regressors yield well *over two orders of magnitude reduction* in cost. We suggest that our schemes perform very well in this example both because of the regularity of the likelihood and because the concentration of the posterior limits the domain over which the approximation must be accurate.

4.4 Implementation and performance notes

We have now demonstrated how our approximate MCMC framework can dramatically reduce the use of the forward model, but we have not yet addressed the performance of our implementation in terms of running time or memory. Although in principle one might worry that the cost of storing the growing sample set $\mathcal{S}$ or of performing the nearest neighbor searches might become challenging, we find that neither is problematic in practice. Storing a few thousand samples, as required in our tests,

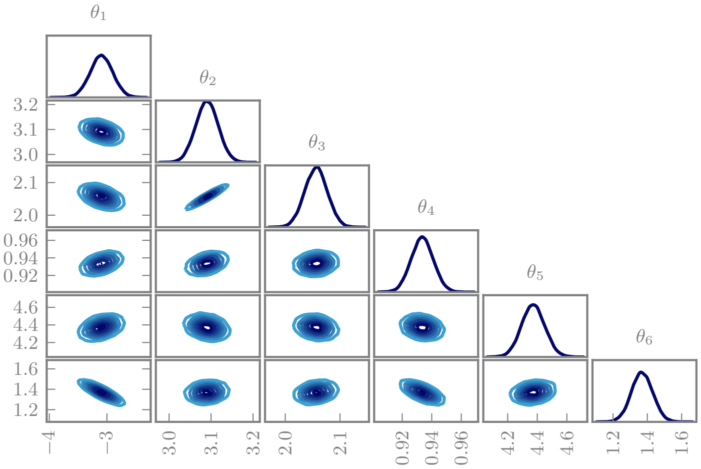


Figure 11: One- and two- dimensional posterior marginals of the parameters in the elliptic PDE inverse problem.

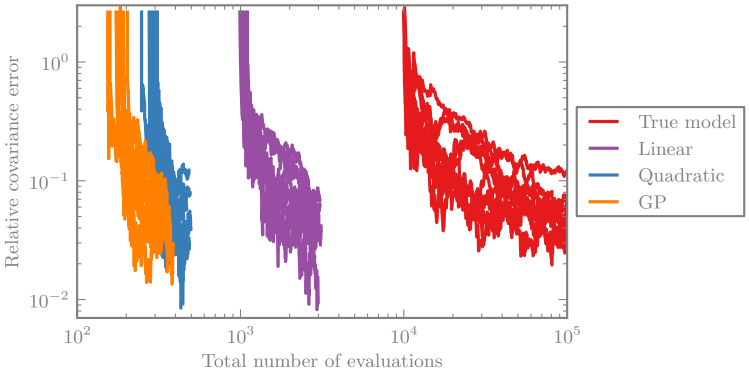


Figure 12: Approximate relative covariance errors in the MCMC chains versus their costs, for the elliptic PDE inverse problem, using several different local approximation strategies. The plot depicts ten independent chains of each type, with the first 10% of each chain removed as burn-in.

is trivial on modern machines. Finding nearest neighbors is a hard problem asymptotically with respect to the parameter dimension and size of the sample set, but our sample sets are neither high dimensional nor large. We use an efficient library to perform the nearest neighbor computations, which implements specialized algorithms that can vastly outperform the asymptotic complexity for low-dimensional nearest neighbors (Muja and Lowe, 2009), and we observe that its run time is an insignificant cost. Computing the error indicator is also relatively inexpensive in these settings: for polynomials, each cross-validation sample only requires a low-rank update of the least squares solution; and for Gaussian processes, drawing from the posterior predictive distribution is fast once the GP has been fit.

To investigate the run-time performance, we measured the average wall-clock time needed to construct each chain used in the genetic toggle switch and elliptic PDE examples on a typical desktop: true model (9 and 4 minutes, respectively), linear (4 and 5 minutes), quadratic (5 minutes and 1 hour), Gaussian process (2.4 and 8.5 hours).[10] For quadratic approximations, benchmarking suggests that around 70% of the run-time was spent computing QR factorizations needed to fit the quadratic surrogates and $< 2\%$ was spent performing nearest neighbor searches or running the full model. Even though the models take only a small fraction of a second to run, the linear approximation is already competitive in terms of run-time. For sufficiently expensive forward models, the fixed cost of constructing approximations will be offset by the cost of the model evaluations, and real run-times should reflect the strong performance we have demonstrated with problem-invariant metrics. Although Gaussian process approximations showed slightly superior performance in terms of model use, the computational effort required to construct them is much higher, suggesting that they will be most useful for extremely expensive models.

5 Discussion

We have proposed a new class of MCMC algorithms that construct local surrogates to reduce the cost of Bayesian inference in problems with computationally expensive forward models. These algorithms introduce local approximations of the forward model or log-likelihood into the Metropolis-Hastings kernel and refine these approximations incrementally and infinitely. The resulting Markov chain thus employs a sequence of approximate transition kernels, but asymptotically samples from the exact posterior distribution. We describe variations of the algorithm that employ either local polynomial or Gaussian process approximations, thus spanning two widely-used classes of surrogate models. Gaussian processes appear to provide somewhat superior performance in terms of reducing use of the forward model, but local quadratic models are cheaper to construct; therefore, both seem to be useful options, depending on cost of the true model. In either case, numerical experiments demonstrate significant reductions in the number of forward model evaluations used for posterior sampling in ODE and PDE model problems.

We do not claim that our algorithm provides minimal error in MCMC estimates given a particular budget of forward model runs; indeed, we expect that problem-specific methods could outperform our strategy in many cases. Instead, we argue that the convergence of the algorithm makes it

[10] The overhead in computing approximations for the elliptic PDE example is more expensive because the forward model has many more outputs than the genetic toggle switch example.

straightforward to apply to novel problems and to assess the quality of the results. The essential reason is that refinement of local approximations is directly tied to the progress of the MCMC chain. As MCMC expends more effort exploring the target distribution, the quality of the approximations increases automatically, via refinement criteria that target problem-independent quantities. The cost of constructing the approximations is incurred incrementally and is tuned to correspond to the MCMC sampling effort. Although it is not feasible to predict in advance how many MCMC steps or model runs will be needed, difficulty either in exploring the posterior or in approximating the model is typically revealed through non-stationary behavior of the chain. Hence, standard MCMC diagnostics can be used to monitor convergence of the chain *and* the underlying approximation. This argument is supported by our numerical results, which produce chains whose convergence is largely indistinguishable from that of regular MCMC. Moreover, after initial exploration of the refinement thresholds, numerical results in these examples are obtained without problem-specific tuning.

Our theoretical and numerical results underscore the notion that local regularity in the forward model or log-likelihood should be harnessed for computational efficiency, and that the number of model evaluations needed to approach exact sampling from the posterior can be much smaller than the number of MCMC samples. Although our convergence arguments can be made quantitative, we believe that doing so in a straightforward manner does not capture the greatest strength of our algorithm. Looking at the process described in Example B.14, we see that a reasonable start results in a bias bound that decays almost exponentially in the number of likelihood evaluations and that the number of likelihood evaluations will grow approximately logarithmically in the running time of the process. Our general bounds, however, only imply that the bias decays at some rate, which may potentially be quite slow. The discrepancy between these rates comes from the fact that our cross-validation approach attempts to evaluate the likelihood primarily in regions where refinement is important. In situations such as Example B.14, these well-chosen likelihood evaluations give a much better estimate than would be obtained from points chosen according to the posterior distribution; in other cases, they seem to be similar. A more general theory would need to avoid the problems that arise in Example B.13 and similar constructions.

There remains significant room to develop other algorithms within this framework. A wide variety of local approximations have theoretical convergence properties similar to those exploited here, offering the opportunity to explore other families of approximations, different weight functions and bandwidths, or variable model order, *cf.* (Cleveland and Loader, 1996; Gramacy and Apley, 2013). Other variations include constructing surrogates by sharing $\mathcal{S}$ across parallel MCMC chains; using any available derivative information from the forward model to help construct local approximations; or using local approximations as corrections to global surrogates, creating hybrid strategies that should combine the fast convergence of global approximations with the asymptotic exactness of our construction (Chakraborty et al., 2013). It should also be possible to extend our use of local approximations to other varieties of MCMC; of particular interest are derivative-based methods such as Metropolis-adjusted Langevin (MALA) or Hybrid Monte Carlo (HMC), where the easy availability of derivatives from our local approximations can dramatically impact their feasibility (Rasmussen, 2003). Several of these variations are explored in Conrad (2014). Finally, further work may reveal connections between the present strategy and other methods for intractable likelihoods, such as pseudo-marginal MCMC, or with data assimilation techniques for expensive models (Law et al., 2015).

Acknowledgments

P. Conrad and Y. Marzouk acknowledge support from the Scientific Discovery through Advanced Computing (SciDAC) program funded by the US Department of Energy, Office of Science, Advanced Scientific Computing Research under award number DE-SC0007099. N. Pillai is partially supported by the grant ONR 14-0001. He thanks Dr. Pedja Neskovic for his interest in this work. Aaron Smith was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

Adler, R. J. (1981). *The Geometry of Random Fields*. SIAM.

Andrieu, C. and G. O. Roberts (2009, April). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics 37*(2), 697–725.

Atkeson, C. G., A. W. Moore, and S. Schaal (1997). Locally Weighted Learning. *Artificial Intelligence Review 11*(1-5), 11–73.

Bal, G., I. Langmore, and Y. M. Marzouk (2013). Bayesian Inverse Problems with Monte Carlo Forward Models. *Inverse problems and imaging 7*(1), 81–105.

Bliznyuk, N., D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan (2008, June). Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation. *Journal of Computational and Graphical Statistics 17*(2), 270–294.

Bliznyuk, N., D. Ruppert, and C. A. Shoemaker (2012, April). Local Derivative-Free Approximation of Computationally Expensive Posterior Densities. *Journal of Computational and Graphical Statistics 21*(2), 476–495.

Chakraborty, A., B. K. Mallick, R. G. Mcclarren, C. C. Kuranz, D. Bingham, M. J. Grosskopf, E. M. Rutter, H. F. Stripling, and R. P. Drake (2013, June). Spline-Based Emulators for Radiative Shock Experiments With Measurement Error. *Journal of the American Statistical Association 108*(502), 411–428.

Christen, J. A. and C. Fox (2005, December). Markov chain Monte Carlo Using an Approximation. *Journal of Computational and Graphical Statistics 14*(4), 795–810.

Cleveland, W. S. (1979, April). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association 74*(368), 829–836.

Cleveland, W. S. and C. Loader (1996). Smoothing by local regression: Principles and methods. In W. Haerdle and M. G. Schimek (Eds.), *Statistical Theory and Computational Aspects of Smoothing*, Volume 1049, pp. 10–49. Springer, New York.

Conn, A. R., N. I. M. Gould, and P. L. Toint (2000). *Trust Region Methods.* SIAM.

Conn, A. R., K. Scheinberg, and L. N. Vicente (2009). *Introduction to Derivative-Free Optimization.* SIAM.

Conrad, P. R. (2014). *Accelerating Bayesian Inference in Computationally Expensive Computer Models Using Local and Global Approximations.* Phd dissertation, Massachusetts Institute of Technology.

Conrad, P. R. and Y. M. Marzouk (2013). Adaptive Smolyak Pseudospectral Approximations. *SIAM Journal of Scientific Computing 35*(6), A2643–2670.

Constantine, P. G., M. S. Eldred, and E. T. Phipps (2012). Sparse Pseudospectral Approximation Method. *Computer Methods in Applied Mechanics and Engineering 229-232*(1), 1–30.

Cotter, S. L., M. Dashti, and A. M. Stuart (2010, March). Approximation of Bayesian Inverse Problems. *SIAM Journal of Numerical Analysis 48*(1), 322–345.

Cotter, S. L., G. O. Roberts, A. M. Stuart, and D. White (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science 28*(3), 424–446.

Cressie, N. (1991). *Statistics for Spatial Data* (revised ed ed.). John Wiley and Sons, Inc.

Cui, T., C. Fox, and M. J. O'Sullivan (2011). Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research 47*(10), W10521.

Cui, T., Y. M. Marzouk, and K. E. Willcox (2014, March). Data-Driven Model Reduction for the Bayesian Solution of Inverse Problems. *International Journal for Numerical Methods in Engineering in press.*

Dashti, M. and A. Stuart (2011). Uncertainty Quantification and Weak Approximation of an Elliptic Inverse Problem. *SIAM Journal of Numerical Analysis 49*(6), 2524–2542.

Efendiev, Y., T. Hou, and W. Luo (2006). Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing 28*(2), 776–803.

Ferré, D., L. Hervé, and J. Ledoux (2013). Regular perturbation of V -geometrically ergodic Markov chains. *Journal of Applied Probability 50*(1), 184–194.

Fielding, M., D. J. Nott, and S.-Y. Liong (2011, February). Efficient MCMC Schemes for Computationally Expensive Posterior Distributions. *Technometrics 53*(1), 16–28.

Fort, G., E. Moulines, and P. Priouret (2012). Convergence of Adaptive and Interacting Markov chain Monte Carlo Algorithms. *Annals of Statistics 39*(6), 3262–3289.

Frangos, M., Y. Marzouk, K. Willcox, and B. van Bloemen Waanders (2010). *Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems, Biegler, Lorenz et al.* John Wiley and Sons.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics 19*(1), 1–141.

Gardner, T. S., C. R. Cantor, and J. J. Collins (2000, January). Construction of a genetic toggle switch in Escherichia coli. *Nature 403*(6767), 339–42.

Gramacy, R. B. and D. W. Apley (2013). Local Gaussian process approximation for large computer experiments. *arXiv preprint* (1), 1–27.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli 7*(2), 223–242.

Hammarling, S. and C. Lucas (2008). Updating the QR factorization and the least squares problem. Technical Report November, University of Manchester.

Higdon, D., H. Lee, and C. Holloman (2003). Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems. In *Bayesian Statistics 7*, pp. 181–197. Oxford University Press.

Joseph, V. R. (2012, August). Bayesian Computation Using Design of Experiments-Based Interpolation Technique. *Technometrics 54*(3), 209–225.

Kaipio, J. and E. Somersalo (2007, January). Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics 198*(2), 493–504.

Kennedy, M. and A. O'Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(3), 425–464.

Korattikara, A., Y. Chen, and M. Welling (2013, April). Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget. *arXiv preprint*, 1–13.

Law, K., A. Stuart, and K. Zygalakis (2015). *Data Assimilation: A Mathematical Introduction.* Texts in Applied Mathematics. Springer International Publishing.

Li, J. and Y. M. Marzouk (2014). Adaptive construction of surrogates for the Bayesian solution of inverse problems. *SIAM Journal on Scientific Computing 36*(3), A1163–A1186.

Lieberman, C., K. Willcox, and O. Ghattas (2010). Parameter and State Model Reduction for Large-Scale Statistical Inverse Problems. *SIAM Journal on Scientific Computing 32*(5), 2523–2542.

Marin, J.-M., P. Pudlo, C. P. Robert, and R. J. Ryder (2011, October). Approximate Bayesian computational methods. *Statistics and Computing 22*(6), 1167–1180.

Marzouk, Y. and D. Xiu (2009). A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics 6*(4), 826–847.

Marzouk, Y. M., H. N. Najm, and L. A. Rahn (2007, June). Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics 224*(2), 560–586.

Muja, M. and D. G. Lowe (2009). Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP 1*, 331–340.

Nobile, F., R. Tempone, and C. G. Webster (2007). A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis 46*(5), 2309.

Rasmussen, C. E. (2003). Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals. In *Bayesian Statistics 7*, pp. 651–659. Oxford University Press.

Roberts, G. and J. Rosenthal (2007). Coupling and Ergodicity of Adapative Markov Chain Monte Carlo Algorithms. *Journal of Applied Probability 44*, 458–475.

Roberts, G. O. and J. S. Rosenthal (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys 1*, 20–71.

Roberts, G. O. and R. L. Tweedie (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika 83*(1), 95–110.

Rosenthal, J. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association 90*, 558–566.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science 4*(4), 409–423.

Santner, T. J., B. J. Williams, and W. I. Notz (2003). *The Design and Analysis of Computer Experiments.* New York: Springer.

Snelson, E. and Z. Ghahramani (2007). Local and global sparse Gaussian process approximations. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07).*

Stein, M. L., Z. Chi, and L. J. Welty (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society. Series B (Methodological) 66*(2), 275–296.

Vecchia, A. V. (1988). Estimation and Model Identification for Continuous Spatial Processes. *Journal of the Royal Statistical Society. Series B (Methodological) 50*(2), 297–312.

Villani, C. (2009). *Optimal transport : old and new.* Grundlehren der mathematischen Wissenschaften. Berlin: Springer.

Xiu, D. and J. S. Hesthaven (2005). High-Order Collocation Methods for Differential Equations with Random Inputs. *SIAM Journal on Scientific Computing 27*(3), 1118.

A Local polynomial regression

Here we provide additional detail about the polynomial regression scheme described in Section 2.2. We consider the quadratic case, as the linear case is a simple restriction thereof. For each component f_j of $\mathbf{f}$, the quadratic regressor is of the form

$$\tilde{f}_j(\hat{\theta}) := a_j + b_j^T \hat{\theta} + \frac{1}{2}\hat{\theta}^T H_j \hat{\theta},$$

where $a_j \in \mathbb{R}$ is a constant term, $b_j \in \mathbb{R}^d$ is a linear term, and $H_j \in \mathbb{R}^{d \times d}$ is a symmetric Hessian matrix. Note that a_j, b_j, and H_j collectively contain $M = (d+2)(d+1)/2$ independent entries for each j. The coordinates $\hat{\theta} \in \mathbb{R}^d$ are obtained by shifting and scaling the original parameters θ as follows. Recall that the local regression scheme uses N samples $\{\theta^1, \dots, \theta^N\}$ drawn from the ball of radius R centered on the point of interest θ, along with the corresponding model evaluations $y_j^i = f_j(\theta^i)$.[11] We assume that the components of θ have already been scaled so that they are of comparable magnitudes, then define $\hat{\theta}^i = (\theta^i - \theta)/R$, so that the transformed samples are centered at zero and have maximum radius one. Writing the error bounds as in (1) requires this rescaling along with the 1/2 in the form of the regressor above (Conn et al., 2009).

Next, construct the diagonal weight matrix $W = \text{diag}(w^1, \dots, w^N)$ using the sample weights in (2), where we have $R = 1$ because of the rescaling. Then compute the N-by-M basis matrix Φ:

$$\Phi = \begin{pmatrix} 1 & \hat{\theta}_1^1 & \cdots & \hat{\theta}_d^1 & \frac{1}{2}\left(\hat{\theta}_1^1\right)^2 & \cdots & \frac{1}{2}\left(\hat{\theta}_d^1\right)^2 & \hat{\theta}_1^1\hat{\theta}_2^1 & \cdots & \hat{\theta}_{d-1}^1\hat{\theta}_d^1 \\ \vdots & & & & & & & & & \vdots \\ 1 & \hat{\theta}_1^N & \cdots & \hat{\theta}_d^N & \frac{1}{2}\left(\hat{\theta}_1^N\right)^2 & \cdots & \frac{1}{2}\left(\hat{\theta}_d^N\right)^2 & \hat{\theta}_1^N\hat{\theta}_2^N & \cdots & \hat{\theta}_{d-1}^N\hat{\theta}_d^N \end{pmatrix}$$

where we ensure that $N > M$. Finally, solve the n least squares problems,

$$\Phi^T W \Phi Z = \Phi^T W Y, \tag{8}$$

where each column of the N-by-n matrix Y contains the samples $\left(y_j^1, \dots, y_j^N\right)^T$, $j = 1, \dots, n$. Each column z_j of $Z \in \mathbb{R}^{M \times n}$ contains the desired regression coefficients for output j,

$$z_j^T = \begin{pmatrix} a_j & b_j^T & (H_j)_{1,1} & \cdots & (H_j)_{d,d} & (H_j)_{1,2} & \cdots & (H_j)_{d-1,d} \end{pmatrix}. \tag{9}$$

[11] To avoid any ambiguities, this appendix departs from the rest of the narrative by using a superscript to index samples and a subscript to index coordinates.

The least squares problem may be solved in a numerically stable fashion using a QR factorization of $W\Phi Z$, which may be computed once and reused for all n least squares problems. The cross-validation fit omitting sample i simply removes row i from both sides of (8). These least squares problems can be solved efficiently with a low-rank update of the QR factorization of the full least squares problem, rather than recomputing the QR factors from scratch (Hammarling and Lucas, 2008).

B Detailed theoretical results and proofs of theorems

B.1 Auxiliary notation

We now define some useful auxillary objects. For a fixed finite set $\mathcal{S} \subset \Theta$, we consider the stochastic process defined by Algorithm 4 with $\mathcal{S}_1 = \mathcal{S}$ and lines 11–20 and 22 removed. This process is essentially the original algorithm with all approximations based on a single set of points $\mathcal{S}$ and no refinements. Since there are no refinements, this process is in fact a Metropolis-Hastings Markov chain, and we write $K_{\mathcal{S}}$ for its transition kernel. For all measurable sets $U \subset \Theta$, this kernel can be written as $K_{\mathcal{S}}(x, U) = r_{\mathcal{S}}(x)\delta_x(U) + (1 - r_{\mathcal{S}}(x)) \int_{y \in U} p_{\mathcal{S}}(x, y)dy$ for some $0 \leq r_{\mathcal{S}}(x) \leq 1$ and density $p_{\mathcal{S}}(x, y)$. We denote by $\alpha_{\mathcal{S}}(x, y)$ the acceptance probability of $K_{\mathcal{S}}$.

We introduce another important piece of notation before giving our results. Let $\{Z_t\}_{t\in\mathbb{N}}$ be a (generally non-Markovian) stochastic process on some state space Ω. We say that a sequence of (generally random, dependent) kernels $\{Q_t\}_{t\in\mathbb{N}}$ is *adapted* to $\{Z_t\}_{t\in\mathbb{N}}$ if there exists an auxillary process $\{A_t\}_{t\in\mathbb{N}}$ so that:

- $\{(Z_t, A_t)\}_{t\in\mathbb{N}}$ is a Markov chain,
- Q_t is $\sigma(A_t)$-measurable, and
- $\mathbb{P}[Z_{t+1} \in \cdot | Z_t, A_t] = Q_t(Z_t, \cdot)$.

Let $\{X_t, \mathcal{S}_t\}_{t\in\mathbb{N}}$ be a sequence evolving according to the stochastic process defined by Algorithm 4 and define the following associated sequence of kernels:

$$\tilde{K}_t(x, A) \equiv \mathbb{P}[X_{t+1} \in A | \{X_s\}_{1\leq s<t}, X_t = x, \{\mathcal{S}_s\}_{1\leq s\leq t}].$$

The sequence of kernels $\{\tilde{K}_t\}_{t\in\mathbb{N}}$ is adapted to $\{X_t\}_{t\in\mathbb{N}}$, with $\{\mathcal{S}\}_{t\in\mathbb{N}}$ as the auxillary process. For any fixed t, one can sample from $\tilde{K}_t(x, \cdot)$ by first drawing a proposal y from $L(x, \cdot)$ and then accepting with probability

$$\tilde{\alpha}_t(x,y) \equiv c_1 \alpha_{\mathcal{S}_t}(x,y) + c_2 \alpha_{\mathcal{S}_t \cup \{(x,f(x))\}}(x,y) + c_3 \alpha_{\mathcal{S}_t \cup \{(y,f(y))\}}(x,y), \tag{10}$$

where c_1, c_2, c_3 are some positive constants that depend on x, y, β_t and γ_t and satisfy the identity $c_1 + c_2 + c_3 = 1$.

B.2 Book-keeping result

The following result will be used repeatedly in our ergodicity arguments.

Theorem B.1 (Approximate Ergodicity of Adaptive Chains)**.** *Fix a kernel K with stationary distribution π on state space $\mathcal{X}$ and let $\{Y_t\}_{t \geq 0}$ evolve according to K. Assume*

$$\|K^t(x,\cdot) - \pi\|_{\mathrm{TV}} \leq C_x (1-\alpha)^t \tag{11}$$

for some $0 < \alpha \leq 1$, $\{C_x\}_{x \in \mathcal{X}}$ and all $t \in \mathbb{N}$.

Let $\{K_t\}_{t \in \mathbb{N}}$ be a sequence of kernels adapted to some stochastic process $\{X_t\}_{t \in \mathbb{N}}$, with auxillary process $\{A_t\}_{t \in \mathbb{N}}$. Also fix a Lyapunov function V and constants $0 < a, \delta, \epsilon < 1$, $0 \leq b < \infty$ and $0 \leq B < \frac{2b}{a\epsilon}$. Assume that there exists a non-random time $\mathcal{T} = \mathcal{T}_{\epsilon,\delta}$ and a $\sigma\left(\{(X_s, A_s)\}_{s \in \mathbb{N}}^{\mathcal{T}}\right)$-measurable event $\mathcal{F}$ so that $\mathbb{P}[\mathcal{F}] > 1 - \epsilon$,

$$\mathbb{E}[V(X_{\mathcal{T}}) \boldsymbol{1}_{\mathcal{F}}] < \infty, \tag{12}$$

$$\sup_{t > \mathcal{T}} \sup_{x \,:\, V(x) < B} \|K_t(x,\cdot) - K(x,\cdot)\|_{\mathrm{TV}} < \delta + \boldsymbol{1}_{\mathcal{F}^c}, \tag{13}$$

and the following inequalities are satisfied for all $t > \mathcal{T}$:

$$\begin{aligned} \mathbb{E}[V(X_{t+1}) \boldsymbol{1}_{\mathcal{F}} | X_t = x, A_t] &\leq (1-a)V(x) + b \qquad (14) \\ \mathbb{E}[V(Y_{t+1}) | Y_t = y] &\leq (1-a)V(y) + b. \end{aligned}$$

Then

$$\limsup_{T \to \infty} \|\mathcal{L}(X_T) - \pi\|_{\mathrm{TV}} \leq 3\epsilon + \delta \frac{\log\left(\frac{e\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + \frac{4b}{aB} \lceil \frac{\log\left(\frac{\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + 1 \rceil,$$

where $\mathcal{C} = \mathcal{C}(\epsilon) \equiv \sup\{C_x : V(x) \leq \frac{2b}{\epsilon a}\}$.

Proof. Assume WLOG that $\mathcal{T} = 0$, fix $\gamma > 0$ and fix $\frac{\log \frac{b}{a(\max(\mathbb{E}[V(X_0)\mathbf{1}_{\mathcal{F}}],\pi(V))+1)}}{\log(1-a)} \leq S < T$. Let $\{Y_t\}_{t \geq S}$, $\{Z_t\}_{t \geq S}$ be Markov chains evolving according to the kernel K and starting at time S, with $Y_S = X_S$ and Z_S distributed according to π . By inequality (11), it is possible to couple $\{Y_t\}_{S \leq t \leq T}$, $\{Z_t\}_{S \leq t \leq T}$ so that

$$\mathbb{P}[Y_T \neq Z_T | X_S] \leq C_{X_S}(1-\alpha)^{T-S} + \gamma. \tag{15}$$

By inequality (13) and a union bound over $S \leq t < T$, it is possible to couple $\{X_t\}_{S \leq t \leq T}$, $\{Y_t\}_{S \leq t \leq T}$ so that

$$\mathbb{P}[X_T \neq Y_T] \leq \delta(T-S) + \mathbb{P}[\mathcal{F}^c] + \mathbb{P}[\max_{S \leq t \leq T}(\max(V(X_t), V(Y_t))) > B] + \gamma. \tag{16}$$

By inequalities (12) and (14),

$$\mathbb{E}[V(X_S)\mathbf{1}_{\mathcal{F}} | X_0, A_0] \leq \mathbb{E}[V(X_0)\mathbf{1}_{\mathcal{F}}](1-a)^S + \frac{b}{a} \leq \frac{2b}{a},$$

and so by Markov's inequality,

$$\mathbb{P}\big[\{V(X_S) > \frac{2b}{a\epsilon}\} \cap \mathcal{F}\big] \leq \epsilon. \tag{17}$$

By the same calculations,

$$\mathbb{P}[\{\max_{S \leq t \leq T}(\max(V(X_t), V(Y_t))) > B\} \cap \mathcal{F}] \leq (T-S+1)\frac{4b}{aB}. \tag{18}$$

Couple $\{Y_t\}_{S \leq t \leq T}$ to $\{X_t\}_{S \leq t \leq T}$ so as to satisfy inequality (16), and then couple $\{Z_t\}_{S \leq t \leq T}$ to $\{Y_t\}_{S \leq t \leq T}$ so as to satisfy inequality (15). It is possible to combine these two couplings of pairs of processes into a coupling of all three processes by the standard 'gluing lemma' (see *e.g.*, Chapter 1 of Villani (2009)). Combining inequalities (15), (16), (17), and (18), we have

$$\begin{aligned}
\|\mathcal{L}(X_T) - \pi\|_{\mathrm{TV}} &\leq \mathbb{P}[X_T \neq Y_T] + \mathbb{P}[Y_T \neq Z_T] \\
&\leq \mathbb{P}[X_T \neq Y_T] + \mathbb{E}[\mathbf{1}_{Y_T \neq Z_T}\mathbf{1}_{V(X_S) > B}\mathbf{1}_{\mathcal{F}}] + \mathbb{E}[\mathbf{1}_{Y_T \neq Z_T}\mathbf{1}_{V(X_S) \leq B}] + \mathbb{P}[\mathcal{F}^c] \\
&\leq \delta(T-S) + 3\epsilon + (T-S+1)\frac{4b}{aB} + 2\gamma + \mathcal{C}(1-\alpha)^{T-S}.
\end{aligned}$$

Approximately optimizing over $S < T$ by choosing $S' = T - \lceil \frac{\log\left(\frac{\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} \rceil$ for T large, we conclude

$$\begin{aligned}\limsup_{T\to\infty} \|\mathcal{L}(X_T) - \pi\|_{\mathrm{TV}} &\leq \limsup_{T\to\infty} \left(\delta(T-S') + 3\epsilon + (T-S+1)\frac{4b}{aB} + 2\gamma + \mathcal{C}(1-\alpha)^{T-S'} \right) \\ &\leq 3\epsilon + 2\gamma + \delta \frac{\log\left(\frac{\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + \frac{\delta}{\log(1-\alpha)} + \frac{4b}{aB}\lceil \frac{\log\left(\frac{\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + 1\rceil.\end{aligned}$$

Since this holds for all $\gamma > 0$, the proof is finished. □

Remark B.2. *In the adaptive MCMC literature, similar results are often stated in terms of a diminishing adaptation condition (this roughly corresponds to inequality* (13)*) and a containment condition (this roughly corresponds to inequalities* (12) *and* (14)*). These phrases were introduced in Roberts and Rosenthal (2007), and there is now a large literature with many sophisticated variants; see, e.g., Fort et al. (2012) for related results that also give LLNs and CLTs under similar conditions. We included our result because its proof is very short, and because checking these simple conditions is easier than checking the more general conditions in the existing literature.*

B.3 Good sets and monotonicity

We give some notation that will be used in the proofs of Theorems 3.4 and 3.3. Fix $0 \leq c, r, R \leq \infty$. For $0 < \ell < \infty$ and $x \in \mathbb{R}^d$, denote by $\mathcal{B}_\ell(x)$ the ball of radius ℓ around x. Say that a finite set $\mathcal{S} \subset \Theta \subset \mathbb{R}^d$ is (c, r, R)-*good* with respect to a set $\mathcal{A} \subset \Theta$ if it satisfies:

1. $\sup_{x \in \mathcal{A}, \|x\| \leq r} \min_{y \in \mathcal{S}} \|x - y\| \leq c$.

2. For all $x \in \mathcal{A}$ with $\|x\| > R$, we have that $|\mathcal{S} \cap \mathcal{B}_{\frac{1}{2}\|x\|}(x)| \geq N$.

We say that it is (c, r, R)-*good* if it is (c, r, R)-*good* with respect to Θ itself. The first condition will imply that the approximation $p_{\mathcal{S}}(x)$ is quite good for x close to the origin. The second condition gives an extremely weak notion of 'locality'; it implies the points we use to construct a 'local' polynomial approximation around x do not remain near the origin when $\|x\|$ itself is very far from the origin. We observe that our definition is monotone in various parameters:

- If $\mathcal{S}$ is (c, r, R)-good, then it is also (c', r', R')-good for all $c' \geq c$, $r' \leq r$ and $R' \geq R$.

- If $\mathcal{S}$ is (c, r, R)-good, then $\mathcal{S} \cup \mathcal{S}'$ is also (c, r, R)-good for any finite set $\mathcal{S}' \subset \Theta$.

- If $\mathcal{S}$ is $(\infty, 0, R)$-good and (c, r, ∞)-good, it is also (c, r, R)-good.

Our arguments will involve showing that, for any finite (c, r, R), the sets $\{\mathcal{S}_t\}_{t\geq 0}$ are eventually (c, r, R)-good.

B.4 Proof of Theorem 3.4, ergodicity in the compact case

In this section we give the proof of Theorem 3.4. Note that some statements are made in slightly greater generality than necessary, as they will be reused in the proof of Theorem 3.3.

Lemma B.3 (Convergence of Kernels). *Let the assumptions stated in the statement of Theorem 3.4 hold. For all $\delta > 0$, there exists a stopping time $\tau = \tau(\delta)$ with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$* [12] *so that*

$$\sup_{t>\tau} \sup_{x\in\Theta} \|K_\infty(x, \cdot) - \tilde{K}_t(x, \cdot)\|_{\mathrm{TV}} < \delta \tag{19}$$

and so that $\mathbb{P}[\tau < \infty] = 1$.

Proof. Fix $R \in \mathbb{R}$ so that $\Theta \subset \mathcal{B}_R(0)$. By results in (Conn et al., 2009),[13] for any $\lambda, \alpha > 0$, there exists a constant $c = c(\alpha, \lambda) > 0$ so that $\sup_{\theta\in\Theta} |p_{\mathcal{S}}(\theta) - p(\theta|\mathbf{d})| < \alpha$ if $\mathcal{S}$ is λ-poised and (c, R, R)-good. Set $c = c(\delta, \lambda)$ and define $\tau = \inf\{t : \mathcal{S}_t \text{ is } (c, R, R) - \text{good}\}$. By definition, this is a stopping time with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$ that satisfies inequality (19); we now check that $\mathbb{P}[\tau < \infty] = 1$.

By the assumption that $\ell(x, y)$ is bounded away from 0, there exist $\epsilon > 0$ and measures μ, $\{r_x\}_{x\in\Theta}$ so that

$$L(x, \cdot) = \epsilon\mu(\cdot) + (1 - \epsilon)r_x(\cdot). \tag{20}$$

Let $\{A_i\}_{i\in\mathbb{N}}$ and $\{B_i\}_{i\in\mathbb{N}}$ be two sequences of i.i.d. Bernoulli random variables, with success probabilities ϵ and β respectively. Let $\tau_0 = \inf\{t : X_t \in \Theta\}$ and define inductively $\tau_{i+1} = \inf\{t > \tau_i + 1 : X_t \in \Theta\}$. By equality (20), it is possible to couple the sequences $\{X_t\}_{t\in\mathbb{N}}, \{A_i\}_{i\in\mathbb{N}}$ so that

$$\begin{aligned} \mathbb{P}[L_{\tau_i} \in \cdot | \tau_i, X_{\tau_i}, A_i = 1] &= \mu(\cdot) \\ \mathbb{P}[L_{\tau_i} \in \cdot | \tau_i, X_{\tau_i}, A_i = 0] &= r_{X_{\tau_i}}(\cdot). \end{aligned} \tag{21}$$

We can further couple $\{B_i\}_{i\in\mathbb{N}}$ to these sequences by using B_i for the random variable in step 12 of Algorithm 4 at time τ_i. That is, when running Algorithm 4, we would run the subroutine

[12] Throughout the note, for any stochastic process $\{Z_t\}_{t\geq 0}$, we use the phrase "τ is a stopping time with respect to $\{Z_t\}_{t\geq 0}$" as shorthand for "τ is a stopping time with respect to the filtration $\mathcal{F}_t$ given by $\mathcal{F}_t = \sigma(\{Z_s\}_{0\leq s\leq t})$."

[13] The required result is a combination of Theorems 3.14 and 3.16, as discussed in the text after the proof of Theorem 3.16 of (Conn et al., 2009).

RefineNear in step 13 of the algorithm at time $t = \tau_i$ if $B_i = 1$, and we would not run that subroutine in that step at that time if $B_i = 0$. Define $I = \{i \in \mathbb{N} : A_i = B_i = 1\}$. Under this coupling of $\{A_i\}_{i\in\mathbb{N}}, \{B_i\}_{i\in\mathbb{N}}$, and $\{X_t\}_{t\in\mathbb{N}}$,

$$\{L_{\tau_i}\}_{i\in I,\, \tau_i<t} \subset \mathcal{S}_t.$$

Furthermore, $\{L_{\tau_i}\}_{i\in I,\, i\leq N}$ is an i.i.d sequence of N draws from μ and $\mathbb{P}[\tau_i < \infty] = 1$ for all i. Let $\mathcal{E}_j$ be the event that $\{L_{\tau_i}\}_{i\leq j}$ is (c, R, R)-good. We have $\tau \leq \tau_{\inf\{j\,:\,\mathcal{E}_j \text{ holds}\}}$. By independence of the sequence $\{L_{\tau_i}\}_{i\in\mathbb{N}}$, we obtain

$$\mathbb{P}[\tau < \infty] \geq \liminf_{j\to\infty} \mathbb{P}[\mathcal{E}_j] = 1.$$

This completes the proof of the Lemma. □

Remark B.4. *We mention briefly that this lemma can also be used to obtain a quantitative bound on the asymptotic rate of convergence of the bias of our algorithm.*

Observe that τ as defined in the proof of Lemma B.3 is stochastically dominated by an exponential distribution with mean $O(-dc^{-d}\log(c))$ as long as both $\ell(x,\cdot)$ and $p(\cdot|\mathbf{d})$ are bounded below. This gives a rather poor bound on the amount of time it takes for inequality (29) to hold. Inequality (29), together with standard 'perturbation' bounds relating the distance between transition kernels and the distance between their stationary distributions, imply a quantitative bound on the asymptotic rate of convergence of the bias of our algorithm. An example of such a perturbation bound may be found by applying Theorem 1 of (Korattikara et al., 2013), which does not in fact rely on time-homogeneity, to a subsequence of the stochastic process generated by our algorithm. Unfortunately, the resulting bound is rather poor, and does not seem to reflect our algorithm's actual performance.

We now prove Theorem 3.4:

Proof. It is sufficient to show that, for all $\epsilon, \delta > 0$ sufficiently small, the conditions of Theorem B.1 can be satisfied. We now set the constants and functions associated with Theorem B.1; we begin by choosing $C_x \equiv V(x) \equiv b = a = 1$, setting $\alpha = \frac{\inf_{x,y\in\Theta} \ell(x,y) \inf_{\theta\in\Theta} p(\theta|d)}{\sup_{\theta\in\Theta} p(\theta|d)}$, and setting $B = \infty$.

By the minorization condition, inequality (11) is satisfied for this value of α; by the assumption that $\ell(x,y), p(\theta|d)$ are bounded away from 0 and infinity, we also have $\alpha > 0$. Next, for all $\delta > 0$, Lemma B.3 implies that implies that $\sup_x \|K(x,\cdot) - \tilde{K}(x,\cdot)\|_{\text{TV}} < \delta$ for all times t greater than some a.s. finite random time $\tau = \tau(\delta)$ that is a stopping time with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$. Choosing $\mathcal{T} = \mathcal{T}_{\epsilon,\delta}$ to be the smallest integer so that $\mathbb{P}[\tau(\delta) > \mathcal{T}] \le 1-\epsilon$ and setting $\mathcal{F} = \{\tau \le \mathcal{T}\}$, this means that inequality (13) is satisfied. Inequalities (14) and (12) are trivially satisfied given our choice of V, a, b. Applying Theorem B.1 with this choice of $V, \alpha, a, b, \mathcal{T}$, we have for all $\epsilon, \delta > 0$ that

$$\limsup_{T\to\infty} \|\mathcal{L}(X_T) - \pi\|_{\mathrm{TV}} \le 3\epsilon + \delta \frac{\log\left(\frac{e\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)}.$$

Letting δ go to 0 and then ϵ go to 0 completes the proof. □

B.5 Proof of Theorem 3.3, ergodicity in the non-compact case

In this section, we prove Theorem 3.3. The argument is similar to that of Theorem 3.4, but we must show the following to ensure that the sampler does not behave too badly when it is far from the posterior mode:

1. $\mathcal{S}_t$ is $(\infty, 0, R)$-good after some almost-surely finite random time τ; see Lemma B.6.
2. The kernel $\tilde{K}_t$ satisfies a drift condition if $\mathcal{S}_t$ is $(\infty, 0, R)$-good; see Lemmas B.8 and B.9.
3. This drift condition implies that the chain X_t spends most of its time in a compact subset of Θ; see Lemma B.10.

Remark B.5. *The Gaussian envelope condition (see Assumption 3.1) is used only to show the second step in the above proof strategy, which in turn is used to satisfy condition (14) of Theorem B.1. It can be replaced by any assumption on the target density for which $\mathcal{S}$ being $(\infty, 0, R)$-good for some $R < \infty$ implies that $\tilde{K}_{\mathcal{S}}$ satisfies a drift condition of the form given by inequality (14).*

We begin by showing, roughly, that for any $R > 0$, $\mathcal{S}_t$ is eventually $(\infty, 0, R)$-good:

Lemma B.6 (Approximations At Infinity Ignore Compact Sets)**.** *Fix any $\mathcal{X} > 0$ and any $k \ge 2$ and define*

$$\tau_{\mathcal{X}}^{(k)} = \sup\Big\{t \,:\, \|L_t\| > k\mathcal{X},\, \|L_t\| - R_t < \mathcal{X}\Big\}.$$

Then

$$\mathbb{P}[\{\text{There exists}\, k < \infty,\ s.t.\ \tau_{\mathcal{X}}^{(k)} < \infty\}] = 1.$$

Proof. Fix $N \in \mathbb{N}$, $\delta > 0$ and $0 < r_1 < r_2 < \infty$. For $0 < \ell < \infty$, denote by $\partial \mathcal{B}_\ell(0)$ the sphere of radius ℓ. Fix a finite covering $\{P_i\}$ of $\partial \mathcal{B}_{\frac{r_1+r_2}{2}}(0)$ with the property that, for any $x \in \partial \mathcal{B}_{\frac{r_1+r_2}{2}}(0)$, there exists at least one i so that $P_i \subset \mathcal{B}_\delta(x)$. For $k \in \mathbb{N}$, define a thickening of P_i by:

$$\mathcal{P}_i^{(k)} = \left\{ x \, : \, \frac{r_1 + r_2}{2} \frac{x}{\|x\|} \in P_i, \; \frac{r_1 + r_2}{2} + (k-1)\frac{r_2 - r_1}{2} \leq \|x\| \leq \frac{r_1 + r_2}{2} + k \frac{r_2 - r_1}{2} \right\}.$$

We will show that, almost surely, for every thickening $\mathcal{P}_i^{(k)}$ of an element P_i of the cover, either $|\mathcal{P}_i^{(k)} \cap \mathcal{S}_t|$ is eventually greater than N or $|\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}|$ is finite. Note that it is trivial that either $|\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}|$ is eventually greater than N or $|\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}|$ is finite; the goal is to check that if $\{L_t\}_{t \in \mathbb{N}}$ visits $\mathcal{P}_i$ infinitely often, $|\mathcal{P}_i^{(k)} \cap \mathcal{S}_t|$ must eventually be greater than N.

To see this, we introduce a representation of the random variables used in step 12 of Algorithm 4. Recall that in this step, L_t is added to $\mathcal{S}_t$ with probability β, independently of the rest of the history of the walk. We will split up the sequence B_t of Bernoulli(β) random variables according to the covering as follows: for each element $\mathcal{P}_i^{(k)}$ of the covering, let $\{B_t^{(i,k)}\}_{t \in \mathbb{N}}$ be an i.i.d. sequence of Bernoulli random variables with success probability β. At the mth time L_t is in $\mathcal{P}_i^{(k)}$, we use $B_m^{(i,k)}$ as the indicator function in step 12 of Algorithm 4. This does not affect the distribution of the steps that the algorithm takes.

By the Borel-Cantelli lemma, we have for each i, k that $\mathbb{P}[B_t^{(i,k)} = 1, \text{infinitely often}] = 1$. If $B_t^{(i,k)} = 1$ infinitely often, then $|\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}| = \infty$ implies that for all $M < \infty$, we have $|\mathcal{P}_i^{(k)} \cap \mathcal{S}_t| > M$ eventually. Let $\mathcal{C}_{i,k}$ be the event that $|\mathcal{P}_i^{(k)} \cap \mathcal{S}_t| > N$ eventually and let $\mathcal{D}_{i,k}$ be the event that $|\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}| = \infty$. Then this argument implies that

$$\mathbb{P}[\mathcal{C}_{i,k} | \mathcal{D}_{i,k}] = 1.$$

Since there are only countably many sets $\mathcal{P}_i^{(k)}$, we have

$$\mathbb{P}[\cap_{i,k} \left(\mathcal{C}_{i,k} \cup \mathcal{D}_{i,k}^c \right)] = 1. \tag{22}$$

Thus, conditioned on the almost sure event $\cap_{i,k} \left(\mathcal{C}_{i,k} \cup \mathcal{D}_{i,k}^c \right)$, all sets $\mathcal{P}_i^{(k)}$ that L_t visits infinitely often will also contribute points to $\mathcal{S}_t$ infinitely often.

Let $k(i) = \min\{k \, : \, |\mathcal{P}_i^{(k)} \cap \{L_t\}_{t \in \mathbb{N}}| = \infty\}$ when that set is non-empty, and set $k(i) = \infty$

otherwise. Let $I = \{i : k(i) < \infty\}$. Finally, set

$$\tau_{r_1,r_2} = \inf\{t : \forall i \in I, |\mathcal{P}_i^{(k(i))} \cap \mathcal{S}_t| \geq N\}. \tag{23}$$

Since $|I|$ is finite, we have shown that, for all $N, \delta > 0$ and $0 < r_2 < r_1 < \infty$, $\mathbb{P}[\tau_{r_1,r_2} < \infty] = 1$. Finally, we observe that for all $\delta = \delta(\mathcal{X}, d)$ sufficiently small, all $N \geq N_{\text{def}}$ and all $k \geq \max_{i \in I} k(i)$,

$$\tau_{\mathcal{X}}^{(k)} \leq \tau_{\frac{2}{3}\mathcal{X}, \frac{4}{3}\mathcal{X}}. \tag{24}$$

This completes the proof. □

Remark B.7. *We will eventually see that, in the notation of the proof of Lemma B.6, $k(i) = 1$ for all i.*

Next, we show that the approximation $p_{\mathcal{S}_t}(x)$ of the posterior used at time t is close to $p_\infty(X_t)$ when $\mathcal{S}_t$ is $(\infty, 0, R)$-good and $\|X_t\|$ is sufficiently large:

Lemma B.8 (Approximation at Infinity)**.** *For all $\epsilon > 0$ and $k \geq 2$, there exists a constant $\mathcal{X} = \mathcal{X}(\epsilon) > 0$ so that, if $R_t < (\|L_t\| - (k-1)\mathcal{X})\mathbf{1}_{\|L_t\| > k\mathcal{X}}$ and the set $\{q_t^{(1)}, \ldots, q_t^{(N)}\}$ is λ-poised, then*

$$|\log(p_{\mathcal{S}_t}(L_t)) - \log(p_\infty(L_t))| < \epsilon + \lambda(N+1)G.$$

Proof. Fix $\epsilon > 0$. By (6) in Assumption 3.1, there exists some $\mathcal{X} = \mathcal{X}(\epsilon)$ so that $\|x\| > \mathcal{X}$ implies

$$|\log(p(x|\mathbf{d})) - \log(p_\infty(x))| < G + \frac{\epsilon}{(N+1)\lambda}. \tag{25}$$

We fix this constant $\mathcal{X}$ in the remainder of the proof.

Denote by $\{f_i\}_{i=1}^{N+1}$ the Lagrange polynomials associated with the set $\{q_t^{(1)}, \ldots, q_t^{(N)}\}$. By Lemma 3.5 of (Conn et al., 2009),

$$\begin{aligned}
|\log(p_{\mathcal{S}_t}(L_t)) - \log(p_\infty(L_t))| &= |\sum_i f_i(L_t) \log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(L_t))| \\
&\leq |\sum_i \log(p_\infty(q_t^{(i)})) f_i(L_t) - \log(p_\infty(L_t))| \\
&\qquad + \sum_i |\log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(q_t^{(i)}))| \, |f_i(L_t)| \\
&\leq 0 + (N+1)\lambda \sup_i |\log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(q_t^{(i)}))|
\end{aligned}$$

where the last line follows from the definition of Lagrange polynomials and Definition 4.7 of (Conn et al., 2009). Under the assumption $\|q_t^{(i)}\| > \mathcal{X}$ for $\|L_t\| - R_t > (k-1)\mathcal{X}$, the conclusion follows from inequality (25). □

For $\epsilon > 0$, define $V_\epsilon(x) = V(x)^{\frac{1}{1+\epsilon}}$, where V is defined in Equation (7). Denote by $\alpha_\infty(x,y)$ the acceptance function of a Metropolis-Hastings chain with proposal kernel L and target distribution p_∞, and recall that $\tilde{\alpha}_t(x,y)$ as given in Equation (10) is the acceptance function for $\tilde{K}_t$. We show that $\tilde{K}_t$ inherits a drift condition from K_∞:

Lemma B.9 (Drift Condition)**.** *For $0 < \delta < \frac{1}{10}$ and $\mathcal{Y}, \mathcal{T} < \infty$, let $\mathcal{F}$ be the event that*

$$|\tilde{\alpha}_t(X_t, L_t) - \alpha_\infty(X_t, L_t)| < \delta + 2\mathbf{1}_{|X_t|<\mathcal{Y}} + 2\mathbf{1}_{|L_t|<\mathcal{Y}} \tag{26}$$

for all $t > \mathcal{T}$. Then, for $\epsilon = \epsilon_0$ as given in item 1 of Assumption 3.2, and all $\delta < \delta_0(\epsilon, a, b, V) < \frac{1}{10}$ sufficiently small and $\mathcal{Y}$ sufficiently large, X_t satisfies a drift condition of the form:

$$\mathbb{E}[V_\epsilon(X_{t+1})\mathbf{1}_{\mathcal{F}}|X_t, \mathcal{S}_t] \le a_1 V_\epsilon(X_t) + b_1 \tag{27}$$

for some $0 \le a_1 < 1$, $0 \le b_1 < \infty$ and for all $t > \mathcal{T}$.

Proof. Assume WLOG that $\mathcal{T} = 0$. Let Z_t be a Metropolis-Hastings Markov chain with proposal kernel L and target distribution p_∞. By Jensen's inequality and Assumption 3.2

$$\mathbb{E}[V_\epsilon(Z_{t+1})|Z_t = x] \le a_\epsilon V_\epsilon(x) + b_\epsilon$$

for some $0 < a_\epsilon < 1$ and some $0 \le b_\epsilon < \infty$.

Assume $X_t = x$ and fix δ so that $\delta < \delta_0$ and $(1+3\delta)a_\epsilon < a_\epsilon + \frac{1}{2}(1-\alpha_\epsilon)$. Then

$$\begin{aligned}
\mathbb{E}[V_\epsilon(X_{t+1})\mathbf{1}_{\mathcal{F}}|X_t = x, \mathcal{S}_t] &\le \int_{y\in\mathbb{R}^d} (\tilde{\alpha}_t(x,y)V_\epsilon(y) + (1-\tilde{\alpha}_t(x,y))V_\epsilon(x))\, \ell(x,y)dy \\
&\le \int_{\mathbb{R}^d\setminus[-\mathcal{Y},\mathcal{Y}]^d} \left(e^{2\delta}\alpha_\infty(x,y)V_\epsilon(y) + \left(1 - e^{-2\delta}\alpha_\infty(x,y)\right) V_\epsilon(x)\right) \ell(x,y)dy \\
&\qquad + \int_{y\in[-\mathcal{Y},\mathcal{Y}]^d} \left(V_\epsilon(x) + \sup_{\|z\|\le\mathcal{Y}} V_\epsilon(z)\right) \ell(x,y)dy \\
&\le (1+3\delta)\int_{\mathbb{R}^d} (\alpha_\infty(x,y)V_\epsilon(y) + (1-\alpha_\infty(x,y))\, V_\epsilon(x))\, \ell(x,y)dy \\
&\qquad + \left(V_\epsilon(x) + \sup_{\|z\|\le\mathcal{Y}} V_\epsilon(z)\right) L\left(x, [-\mathcal{Y},\mathcal{Y}]^d\right)
\end{aligned}$$

$$\leq (1+3\delta)a_\epsilon V_\epsilon(x) + (1+3\delta)b_\epsilon + \left(V_\epsilon(x) + \sup_{\|z\|\leq \mathcal{Y}} V_\epsilon(z)\right) L\left(x, [-\mathcal{Y},\mathcal{Y}]^d\right).$$

Since $\delta < \frac{1}{10}$ and $(1+3\delta)a_\epsilon < a_\epsilon + \frac{1}{2}(1-\alpha_\epsilon)$, we have

$$\mathbb{E}[V_\epsilon(X_{t+1})\mathbf{1}_{\mathcal{F}}|X_t = x, \mathcal{S}_t] \leq (a_\epsilon + \frac{1}{2}(1-\alpha_\epsilon))V(x) + (1+3\delta)b_\epsilon + \left(V_\epsilon(x) + \sup_{\|z\|\leq \mathcal{Y}} V_\epsilon(z)\right) L\left(x, [-\mathcal{Y},\mathcal{Y}]^d\right).$$

Since $V_\epsilon(x)L\left(x,[-\mathcal{Y},\mathcal{Y}]^d\right)$ is uniformly bounded in x for all fixed $\mathcal{Y}$ by item 2 of Assumption 3.2, the claim follows with

$$a_1 = a_\epsilon + \frac{1}{2}(1-\alpha_\epsilon) < 1,$$
$$b_1 = 2b_\epsilon + \sup_x V_\epsilon(x)L\left(x,[-\mathcal{Y},\mathcal{Y}]^d\right) + \sup_{\|z\|\leq \mathcal{Y}} V_\epsilon(z),$$

finishing the proof. □

We use these bounds to show that some compact set is returned to infinitely often:

Lemma B.10 (Infinitely Many Returns)**.** *For $G < G(L, p_\infty, \lambda, N)$ sufficiently small, there exists a compact set $\mathcal{A}$ that satisfies $\mathbb{P}[\sum_{t\in\mathbb{N}} \mathbf{1}_{X_t\in\mathcal{A}} = \infty] = 1$.*

Proof. Combining Lemmas B.6, B.8 and B.9, there exists some number $\mathcal{X} > 0$ and almost surely finite random time $\tau_{\mathcal{X}}$ so that X_t satisfies a drift condition of the form

$$\mathbb{E}[V(X_{t+1})\mathbf{1}_{t>\tau_{\mathcal{X}}}|X_t = x, \mathcal{S}_t] \leq aV(x) + b$$

for some function V and constants $0 \leq a < 1$, $b < \infty$. The existence of a recurrent compact set follows immediately from this drift condition and Lemma 4 of (Rosenthal, 1995). □

This allows us to slightly strengthen Lemma B.9:

Lemma B.11. *All times $\tau_{\mathcal{X},2\mathcal{X}}$ of the form given in Equation (23) satisfy $\mathbb{P}[\tau_{\mathcal{X},2\mathcal{X}} < \infty] = 1$ and are stopping times with respect to $\{\mathcal{S}_t\}$. Furthermore, for $G < G(L, p_\infty, \lambda, N)$ sufficiently small, there exists a random time τ of the form given in Equation (23) so that*

$$\mathbb{E}[V_\epsilon(X_{t+1})\mathbf{1}_{\tau<t}|X_t, \mathcal{S}_t] \leq a_1 V_\epsilon(X_t) + b_1 \tag{28}$$

for some $0 \leq a_1 < 1$, $0 \leq b_1 < \infty$.

Proof. By inequality (24), there exists a random time $\tau \equiv \tau_{\mathcal{X},2\mathcal{X}}$ of the form (23) that is at least as large as the random time $\tau_{\mathcal{X}}$ constructed in the proof of Lemma B.10 and that satisfies $\mathbb{P}[\tau < \infty] = 1$. As shown in Lemma B.10, an inequality of the form (28) holds for $\tau_{\mathcal{X}}$, and so the same inequality must also hold with $\tau_{\mathcal{X}}$ replaced by the larger time $\tau \geq \tau_{\mathcal{X}}$.

The only detail to check is that all random times $\tau_{\mathcal{X},2\mathcal{X}}$ of the form (23) are stopping times with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$. Let $\{P_i\}$ be the partition associated with $\tau_{\mathcal{X},2\mathcal{X}}$, as constructed in Lemma B.6. By Lemma B.10 and part 3 of Assumption 3.2, we have $\mathbb{P}[|\{L_t\}_{t\in\mathbb{N}} \cap \mathcal{P}_i^{(1)}| = \infty] = 1$ for all i. Thus, in the notation of Lemma B.6, $I^c = \emptyset$ and $k(i) = 1$ for all $i \in I$. Thus, we have shown that $\tau_{\mathcal{X},2\mathcal{X}} = \inf\{t : \forall i, \, |\mathcal{P}_i^{(1)} \cap \mathcal{S}_t| \geq N\}$, which is clearly a stopping time with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$, and the proof is finished. □

We now finish our proof of Theorem 3.3 analogously to our proof of Theorem 3.4.

The following bound is almost identical to Lemma B.3, but now proved under the Gaussian envelope assumption for the target density.

Lemma B.12 (Convergence of Kernels)**.** *Let the assumptions stated in the statement of Theorem 3.3 hold and fix a compact set $\mathcal{A} \subset \Theta$. For all $\delta > 0$, there exists a stopping time $\tau = \tau(\delta)$ with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$ so that*

$$\sup_{t>\tau} \sup_{x\in\mathcal{A}} \|K_\infty(x,\cdot) - \tilde{K}_t(x,\cdot)\|_{\mathrm{TV}} < \delta \tag{29}$$

and so that $\mathbb{P}[\tau < \infty] = 1$.

Proof. Fix a constant $0 < R < \infty$ so that $\mathcal{A} \subset \mathcal{B}_R(0)$. By results in (Conn et al., 2009), for any $\lambda, \alpha > 0$, there exists a constant $c = c(\alpha, \lambda) > 0$ so that $\sup_{\theta\in\mathcal{A}} |p_{\mathcal{S}}(\theta) - p(\theta|\mathbf{d})| < \alpha$ if $\mathcal{S}$ is λ-poised and (c, R, R)-good. Set $c = c(\epsilon, \lambda)$ and define $\tau' = \inf\{t : \mathcal{S}_t \text{ is } (c, R, R) - \text{good}\}$. By definition, τ' is a stopping time with respect to $\{\mathcal{S}_t\}_{t\in\mathbb{N}}$ that satisfies inequality (29). We now check that $\mathbb{P}[\tau' < \infty] = 1$. By the assumption that $\ell(x,y)$ is bounded away from 0, there exist $\epsilon > 0$ and measures μ, $\{r_x\}_{x\in\Theta}$ so that

$$L(x,\cdot) = \epsilon\mu(\cdot) + (1-\epsilon)r_x(\cdot). \tag{30}$$

Let $\{A_i\}_{i\in\mathbb{N}}$ and $\{B_i\}_{i\in\mathbb{N}}$ be two sequences of i.i.d. Bernoulli random variables, with success probabilities ϵ and β respectively. Let $\tau_0 = \inf\{t : X_t \in \mathcal{A}\}$ and define inductively $\tau_{i+1} = \inf\{t >$

$\tau_i + 1 : X_t \in \mathcal{A}\}$. By equality (30), it is possible to couple the sequences $\{X_t\}_{t \in \mathbb{N}}, \{A_i\}_{i \in \mathbb{N}}$ so that

$$\mathbb{P}[L_{\tau_i} \in \cdot | \tau_i, X_{\tau_i}, A_i = 1] = \mu(\cdot) \tag{31}$$
$$\mathbb{P}[L_{\tau_i} \in \cdot | \tau_i, X_{\tau_i}, A_i = 0] = r_{X_{\tau_i}}(\cdot).$$

We can further couple $\{B_i\}_{i \in \mathbb{N}}$ to these sequences by using B_i for the random variable in step 12 of Algorithm 4 at time τ_i. That is, when running Algorithm 4, we would run the subroutine RefineNear in step 13 of the algorithm at time $t = \tau_i$ if $B_i = 1$, and we would not run that subroutine in that step at that time if $B_i = 0$. Define $I = \{i \in \mathbb{N} : A_i = B_i = 1\}$. Under this coupling of $\{A_i\}_{i \in \mathbb{N}}, \{B_i\}_{i \in \mathbb{N}}$, and $\{X_t\}_{t \in \mathbb{N}}$,

$$\{L_{\tau_i}\}_{i \in I,\, \tau_i < t} \subset \mathcal{S}_t.$$

Furthermore, $\{L_{\tau_i}\}_{i \in I,\, i \leq N}$ is an i.i.d sequence of N draws from μ, and by Lemma B.10, $\mathbb{P}[\tau_i < \infty] = 1$ for all i. Let $\mathcal{E}_j$ be the event that $\{L_{\tau_i}\}_{i \leq j}$ is (c, R, R)-good. We have $\tau' \leq \inf\{\tau_j : \mathcal{E}_j \text{ holds}\}$. By independence of the sequence $\{L_{\tau_i}\}_{i \in \mathbb{N}}$, we obtain

$$\mathbb{P}[\tau' < \infty] \geq \liminf_{j \to \infty} \mathbb{P}[\mathcal{E}_j] = 1.$$

This argument shows that, for any compact set $\mathcal{A}$, there exists a stopping time τ' with respect to $\{\mathcal{S}_t\}_{t \in \mathbb{N}}$ so that $\mathbb{P}[\tau' < \infty] = 1$ and so that

$$\sup_{t > \tau'} \sup_{x \in \mathcal{A}} \|\tilde{K}_t(x, \cdot) - K_\infty(x, \cdot)\|_{\mathrm{TV}} < \delta. \tag{32}$$

This completes the proof of the Lemma.

□

We are finally ready to prove Theorem 3.3:

Proof of Theorem 3.3. As with the proof of Theorem 3.4, it is sufficient to show that, for all $\epsilon, \delta, G > 0$ sufficiently small and all $B \gg \epsilon^{-1}$ sufficiently large, the conditions of Theorem B.1 can be satisfied for some time $\mathcal{T} = \mathcal{T}_{\epsilon,\delta}$ with the same drift function V and constants α, a, b.

By Assumption 3.2 and Theorem 12 of Rosenthal (1995), inequality (11) holds for some $\alpha > 0$ and $\{C_x\}_{x \in \Theta}$. For any fixed $0 < B < \infty$ and all $0 < G, \delta$ sufficiently small, Lemma B.12 implies that there exists some almost surely finite stopping time $\tau_1 = \tau_1(\delta)$ so that inequality (13) holds for the set $\mathcal{F}_1 = \{\tau_1 > t\}$. Lemma B.11 implies that, for all $G > 0$ sufficiently small, there exists some almost surely finite stopping time τ_2 so that inequality (14) holds for the set $\mathcal{F}_2 = \{\tau_2 > t\}$. Choose $\mathcal{T}$ to be the smallest integer so that $\mathbb{P}[\max(\tau_1, \tau_2) > \mathcal{T}] < \epsilon$ and set $\mathcal{F} = \{\min(\tau_1, \tau_2) > \mathcal{T}\}$. We then have that inequalities (13) and (14) are satisfied. Finally, inequality (12) holds by part 2 of Assumption 3.2. We have shown that there exist fixed values of $\mathcal{C}$ and α so that the conditions of Theorem B.1 hold for all $\epsilon, \delta > 0$ sufficiently small. We conclude that, for all $\epsilon, \delta > 0$ sufficiently small,

$$\limsup_{T\to\infty} \|\mathcal{L}(X_T) - \pi\|_{\mathrm{TV}} \le 3\epsilon + \delta \frac{\log\left(\frac{e\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + \frac{4b}{aB}\lceil \frac{\log\left(\frac{\delta}{\mathcal{C}\log(1-\alpha)}\right)}{\log(1-\alpha)} + 1\rceil.$$

Letting B go to infinity, then δ go to 0 and finally ϵ go to 0 completes the proof. □

B.6 Alternative assumptions

In this section, we briefly give other sufficient conditions for ergodicity. We do not give detailed proofs but highlight the instances at which our current arguments should be modified.

The central difficulty in proving convergence of our algorithm is that, in general, the local polynomial fits we use may be very poor when R_t is large. This difficulty manifests in the fact that, for most target distributions, making the set $\mathcal{S}$ a (c, r, R)-good set does not guarantee that $\tilde{K}_{\mathcal{S}}$ inherits a drift condition of the form (14) from K_∞, for any value of c, r, R. Indeed, no property that is monotone in the set $\mathcal{S}$ can guarantee that $\tilde{K}_{\mathcal{S}}$ satisfies a drift condition. In a forthcoming project focused on theoretical issues, we plan to show convergence based on drift conditions that only hold 'on average' and over long time intervals. There are several other situations under which it is possible to guarantee the eventual existence of a drift condition, and thus ergodicity:

1. Fix a function $\delta_0 : \Theta \to \mathbb{R}^+$ and add the step "If $R_t > \delta_0(\theta^+)$, $\mathcal{S} \leftarrow \{(\theta^+, f(\theta^+)\} \cup \mathcal{S})$" between steps 7 and 8 of Algorithm 4. If $\lim_{r\to\infty} \sup_{\|x\|\ge r} \delta_0(x) = 0$ and

$$\lim_{r\to\infty} \sup_{\|x\|\ge r} \max(\|p'(\theta|\mathbf{d})\|, \|p''(\theta|\mathbf{d})\|) = 0,$$

then the main condition of Lemma B.9, inequality (26) (with α_∞ replaced by the acceptance function of K), holds by a combination of Theorems 3.14 and 3.16 of (Conn et al., 2009). If $p(\theta|\mathbf{d})$ has sub-Gaussian tails, the proof of Lemma B.9 can then continue largely as written if we replace $p_\infty(x)$ with $p(x|\mathbf{d})$ wherever it appears. Since the Gaussian envelope condition is only used to prove that the condition in Lemma B.9 holds, Theorem 3.3 holds with the Gaussian envelope condition replaced by these requirements.

2. Similar results sometimes hold if we only require that $\delta_0(x) \equiv \delta_0$ be a sufficiently small constant. Theorem 1 of Ferré et al. (2013), combined with Theorems 3.14 and 3.16 of (Conn et al., 2009), can be used to obtain weaker sufficient conditions under which the condition in Lemma B.9 holds.

3. If $d = 1$, $N_{\text{def}} = 2$, and the approximations in Algorithm 4 are made using linear rather than quadratic models, we state without proof that a drift condition at infinity proved in Lemma B.9 can be verified directly. For $d \geq 2$, more work needs to be done.

4. Finally, we discuss analogous results that hold for other forms of local approximation, such as Gaussian processes. When the target distribution is compact, we expect Theorem 3.4 to hold as stated whenever local approximations to a function based on (c, R, R)-good sets converge to the true function value as c goes to 0. In our proof of Theorem 3.4, we cite (Conn et al., 2009) for this fact. The proof of Theorem 3.4 will hold as stated for other local approximations if all references to (Conn et al., 2009) are replaced by references to appropriate analogous results. Such results typically hold for reasonably constructed local approximation strategies (Cleveland and Loader, 1996; Atkeson et al., 1997).

   When the target distribution is not compact, modifying our arguments can be more difficult, though we expect similar conclusions to often hold.

B.7 Examples for parameter choices

Example B.13 (Decay Rate for β)**.** *We note that if β_t decays too quickly, our sampler may not converge, even if $\gamma_t \to 0$ at any rate. Consider the proposal distribution L that draws i.i.d. uniform samples from $[0,1]^d$ and let $\lambda(\cdot)$ denote the Lebesgue measure. Consider a target distribution of the form $p(\theta|\mathbf{d}) \propto \mathbf{1}_{\theta \in G}$ for set G with Lebesgue measure $0 < \lambda(G) < 1$. If $\sum_t \beta_t < \infty$, then by Bo+rel-Cantelli, the probability $p = p\left(\{\beta_t\}_{t\in\mathbb{N}}\right)$ that no points are added to $\mathcal{S}$ except during the initial choice of reference points or failed cross-validation checks is strictly greater than 0. With probability $\lambda(G)^k > 0$, the first k reference points are all in G. But if both these events happen, all cross-validation checks are passed for any $\gamma > 0$, and so the walk never converges; it samples from the measure λ forever.*

Example B.14 (Decay Rate for γ)**.** *We note that we have not used the assumption that* $\gamma < \infty$ *anywhere. As pointed out in Example B.13, in a way this is justified—we can certainly find sequences* $\{\beta_t\}_{t \in \mathbb{N}}$ *and walks that are not ergodic for any sequence* $\gamma_t > 0$ *converging to zero at any rate.*

In the other direction, there exist examples for which having any reasonable fixed value of γ *gives convergence, even with* $\beta = 0$*. We point out that this depends on the initially selected points; one could be unlucky and choose points with log-likelihoods that happen to lie exactly on some quadratic that does not match the true distribution. Consider a target density* $\pi(x) \propto 1 + C\boldsymbol{1}_{x > \frac{1}{2}}$ *on* $[0, 1]$ *with independent proposal moves from the uniform measure on* $[0, 1]$*. To simplify the discussion, we assume that our approximation of the density at each point is linear and based exactly on the three nearest sampled points. Denote by* $\mathcal{S}_t$ *the points which have been evaluated by time* t*, and let* $\mathcal{S}_0 = \{\frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}\}$*. Write* $x_1, \ldots, x_{m(t)} = \mathcal{S}_t \cap [0, \frac{1}{2}]$ *and* $x_{m(t)+1}, \ldots, x_{n(t)} = \mathcal{S}_t \cap [\frac{1}{2}, 1]$*. It is easy to check that*

$$\|\mathcal{L}(X_{t+1}) - \pi\|_{\text{TV}} \leq x_{m(t)+3} - x_{m(t)-2}. \tag{33}$$

It is also easy to see that with probability one, for any $\gamma < \frac{1}{2}$*, there will always be a subinterval of* $[x_{m(t)-2}, x_{m(t)+3}]$ *with strictly positive measure for which a cross-validation check will fail. Combining this with inequality (33) implies that the algorithm will converge in this situation, even with* $\beta = 0$*. Furthermore, in this situation choosing* $\beta \equiv 0$ *results in a set* $\mathcal{S}_t$ *that grows extremely slowly in* t*, without substantially increasing bias.*

C Genetic toggle switch inference problem

Here we provide additional details about the setup of the genetic toggle switch inference problem from Section 4.2. This genetic circuit has a bistable response to the concentration of an input chemical, [IPTG]. Figure 13 illustrates these high and low responses, where the vertical axis corresponds to the expression level of a particular gene. (Gardner et al., 2000) proposed the following differential-algebraic model for the switch:

$$\begin{aligned}
\frac{du}{dt} &= \frac{\alpha_1}{1 + v^\beta} - u, \\
\frac{dv}{dt} &= \frac{\alpha_2}{1 + w^\gamma} - v, \\
w &= \frac{u}{(1 + [\text{IPTG}]/K)^\eta}.
\end{aligned} \tag{34}$$

The model contains six unknown parameters $Z_\theta = \{\alpha_1, \alpha_2, \beta, \gamma, K, \eta\} \in \mathbb{R}^6$, while the data correspond to observations of the steady-state values $v(t = \infty)$ for six different input concentrations of [IPTG], averaged over several trials each. As in (Marzouk and Xiu, 2009), the parameters are centered and scaled around their nominal values so that they can be endowed with uniform priors over the hypercube $[-1, 1]^6$. Specifically, the six parameters of interest are normalized around their nominal values to have the form

$$Z_i = \bar{\theta}_i(1 + \zeta_i \theta_i), \; i = 1, \dots, 6,$$

so that each θ_i has prior Uniform$[-1, 1]$. The values of $\bar{\theta}_i$ and ζ_i are given in Table 1. The data are observed at six different values of [IPTG]; the first corresponds to the "low" state of the switch while the rest are in the "high" state. Multiple experimental observations are averaged without affecting the posterior by correspondingly lowering the noise; hence, the data comprise one observation of v/v_{ref} at each concentration, where $v_{\text{ref}} = 15.5990$. The data are modeled as having independent Gaussian errors, *i.e.*, as draws from $\mathcal{N}(d_i, \sigma_i^2)$, where the high- and low-state observations have different standard deviations, specified in Table 2. The forward model may be computed by integrating the ODE system (35), or more simply by iterating until a fixed point for v is found.

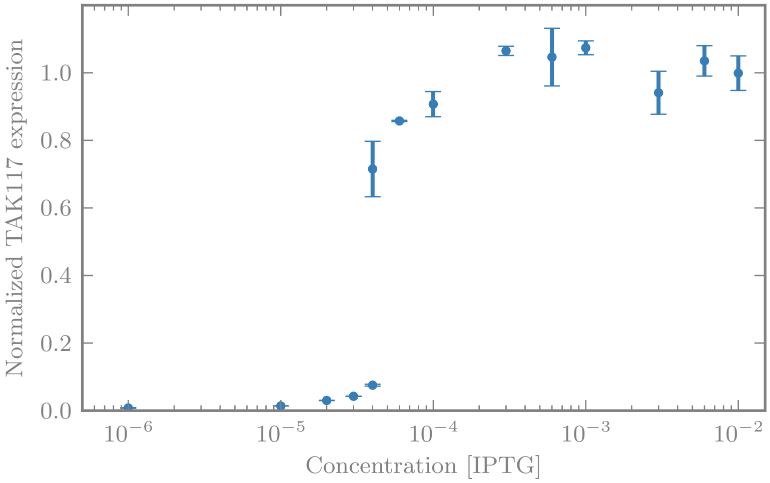


Figure 13: Response of the pTAK117 genetic toggle switch to the input concentration of IPTG (Gardner et al., 2000). The plot shows the mean and standard deviation of the experimentally-observed gene expression levels over a range of input concentrations. Expression levels are normalized by the mean response at the largest IPTG concentration.

Table 1: Normalization of the parameters in the genetic toggle switch example.

	α_1	α_2	β	γ	K	η
$\bar{\theta}_i$	156.25	15.6	2.5	1	2.0015	2.9618e-5
ζ_i	0.20	0.15	0.15	0.15	0.30	0.2

Table 2: Data and obervation error variances for the likelihood of the genetic toggle switch example.

[IPTG]	156.25	15.6	2.5	1	2.0015	2.9618e-5
d_i	0.00798491	1.07691684	1.05514201	0.95429837	1.02147051	1.0
σ_i	4.0e-5	0.005	0.005	0.005	0.005	0.005

D Elliptic PDE inverse problem

Here we provide details about the elliptic PDE inference problem. The forward model is given by the solution of an elliptic PDE in two spatial dimensions

$$\nabla_{\mathbf{s}} \cdot (k(\mathbf{s}, \theta) \nabla_{\mathbf{s}} u(\mathbf{s}, \theta)) = 0, \tag{35}$$

where $\mathbf{s} = (s_1, s_2) \in [0, 1]^2$ is the spatial coordinate. The boundary conditions are

$$\begin{aligned} u(\mathbf{s}, \theta)|_{s_2=0} &= s_1, \\ u(\mathbf{s}, \theta)|_{s_2=1} &= 1 - s_1, \\ \left.\frac{\partial u(\mathbf{s}, \theta)}{\partial s_1}\right|_{s_1=0} &= 0, \\ \left.\frac{\partial u(\mathbf{s}, \theta)}{\partial s_1}\right|_{s_1=1} &= 0. \end{aligned}$$

This PDE serves as a simple model of steady-state flow in aquifers and other subsurface systems; k can represent the permeability of a porous medium while u represents the hydraulic head. Our numerical solution of (35) uses the standard continuous Galerkin finite element method with bilinear basis functions on a uniform 30-by-30 quadrilateral mesh.

The log-diffusivity field $\log k(\mathbf{s})$ is endowed with a Gaussian process prior, with mean zero and an isotropic squared-exponential covariance kernel:

$$C(\mathbf{s}_1, \mathbf{s}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{s}_1 - \mathbf{s}_2\|^2}{2\ell^2} \right),$$

for which we choose variance $\sigma^2 = 1$ and a length scale $\ell = 0.2$. This prior allows the field to be easily parameterized with a Karhunen-Loève (K-L) expansion (Adler, 1981):

$$k(\mathbf{s}, \theta) \approx \exp\left(\sum_{i=1}^{d} \theta_i \sqrt{\lambda_i} k_i(\mathbf{s})\right),$$

where λ_i and $k_i(\mathbf{s})$ are the eigenvalues and eigenfunctions, respectively, of the integral operator on $[0, 1]^2$ defined by the kernel C, and the parameters θ_i are endowed with independent standard normal priors, $\theta_i \sim \mathcal{N}(0, 1)$. These parameters then become the targets of inference. In particular, we truncate the Karhunen-Loève expansion at $d = 6$ modes and condition the corresponding mode weights $(\theta_1, \ldots, \theta_6)$ on data. Data arise from observations of the solution field on a uniform 11×11 grid covering the unit square. The observational errors are taken to be additive and Gaussian:

$$d_j = u(\mathbf{s}_j, \theta) + \epsilon_j,$$

with $\epsilon_j \sim \mathcal{N}(0, 0.1^2)$.