IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 January 2024, accepted 13 March 2024, date of publication 18 March 2024, date of current version 25 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3378752

RESEARCH ARTICLE

New Diagnostic Assessment of MCMC Algorithm Effectiveness, Efficiency, Reliability, and Controllability

HOSSEIN KAVIANIHAMEDANI[1], JULIANNE D. QUINN[1,2], AND JARED D. SMITH[3]
[1]Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA 22904, USA
[2]Department of Civil and Environmental Engineering, University of Virginia, Charlottesville, VA 22904, USA
[3]U.S. Geological Survey, Reston, VA 20192, USA

Corresponding author: Julianne D. Quinn (julianne.quinn@virginia.edu)

ABSTRACT Markov Chain Monte Carlo (MCMC) is a robust statistical approach for estimating posterior distributions. However, the significant computational cost associated with MCMC presents a considerable challenge, complicating the selection of an appropriate algorithm tailored to the specific problem at hand. This study introduces a novel and comprehensive framework for evaluating the performance of MCMC algorithms, drawing inspiration from diagnostics used for multi-objective evolutionary algorithms. We employ visualizations to evaluate key algorithmic characteristics: Effectiveness (the ability to accurately find representative posterior modes, quantified by the Kullback-Leibler Divergence (KLD) and Wasserstein Distance (WD)), Efficiency (the speed of posterior characterization), Reliability (consistency across different random seeds), and Controllability (insensitivity to hyperparameter variation). Evaluating three prominent MCMC algorithms—Metropolis-Hastings (MH), Adaptive Metropolis (AM), and Differential Evolution Adaptive Metropolis (DREAM)—on high-dimensional and bimodal test problems, our analysis uncovers several insights. First, across algorithms, the number of function evaluations most controls performance on the high-dimensional problem, while the number of chains most controls performance on the bimodal problem. While this suggests similar controllability across algorithms, differences emerge on the other algorithmic characteristics. For high numbers of functions evaluations, AM performs best on the high-dimensional problem, while for low (<5) and high (>15) chain counts, MH and AM perform best on the bimodal problem, as measured by KLD. However, outside these specific cases, DREAM consistently demonstrates superior efficiency and reliability, making it a robust choice for both high-dimensional and multimodal problems. These findings can inform MCMC algorithm selection for Bayesian inference applications, as well as hyperparameterization of the chosen algorithm. More importantly, the diagnostics represent a generalizable contribution to research on MCMC diagnostics that can be used to evaluate and inform the design of new algorithms.

INDEX TERMS Bayesian estimation, Markov chain Monte Carlo, high dimensionality, multi-modality, model diagnostics.

I. INTRODUCTION

Bayesian inference can be used to estimate the parameters, θ, of a model and their associated uncertainty, given the available data. This is useful for informing robust engineering designs that can tolerate this uncertainty; see example applications in [1], [2], [3], [4], [5], [6], and [7]. The approach relies on Bayes' theorem in which the modeler uses their knowledge of the system's physical behavior and mathematical constraints to develop a prior probability distribution for the parameters, $p(\theta)$, that is updated by the likelihood of observing the data x, $p(x|\theta)$. This allows the

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

estimation of the posterior probability of that parameter set given the observed data, $p(\theta|x)$:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}. \quad (1)$$

The posterior distribution represents the uncertainty of the model parameters to which we would like engineering designs to be robust.

Markov Chain Monte Carlo (MCMC) is a powerful statistical method used to estimate posterior distributions. MCMC uses a Monte Carlo simulation to sample from a Markov Chain whose transition probabilities from the current point in the chain to the next proposed point in the parameter space are determined by the relative posterior density of the current and proposed locations. In the long run, the distribution sampled by the Markov chain will become stationary and represent the true posterior distribution. However, using posterior estimates from MCMC prematurely before the algorithm has converged to its stationary distribution could result in a poor characterization of uncertainty, and consequently over- or under-designed systems. As such, a vast body of literature has focused on developing diagnostics to assess the convergence of MCMC search processes (see reviews of MCMC diagnostics by [8], [9]). Specifically, these diagnostics focus on determining 1) how many initial iterations should be discarded as ''burn-in'' because they are far from the posterior mode and not representative of the stationary distribution; and 2) how many iterations are sufficient to stop the algorithm, as the chain has converged to its stationary distribution [10].

In some cases, one can calculate these numbers theoretically. For example, if it can be shown through drift and minorization criteria [11] that the chain converges at an exponential rate, a bound on the number of burn-in iterations needed for the total variation distance between the estimated and true posterior to reach some tolerance can be calculated [12]. A similar estimation can be performed using Wasserstein distance [13]. With respect to the total number of iterations, if a Central Limit Theorem exists for some function g of the samples X (such as their mean or variance), one can estimate the sample size needed for $g(X)$ to converge to its true value at some confidence level [9].

Unfortunately, for black box-type parameter estimation problems that are common in engineering, no such theoretical bounds can be estimated [14]. Instead, MCMC users must rely on visual or quantitative metrics of convergence, many of which require running multiple chains [8]. Visual MCMC diagnostics monitor the progress of the search using graphical tools like trace plots, histograms, and correlograms. These visual aids cannot identify with certainty that a chain has converged, but they can identify problems that indicate it has not. For example, trace plots of the chain location vs. iterations illustrate if the chains are getting stuck, moving too slowly due to high auto-correlation, or trending and therefore not yet stationary; histograms indicate if the posterior estimates across chains are inconsistent with one another; and correlograms indicate if the autocorrelation in the chains is too high, reducing the effective sample size (ESS) [9]. Plots of more complex chain statistics have also been proposed [15], [16], [17], but these are problem or algorithm-specific.

Because graphical plots can only identify problems in convergence, these diagnostics are typically complemented by numerical convergence metrics that provide more objective stopping criteria. The most common metric, the Gelman-Rubin (GR) diagnostic, calculates the ratio of the variance across chains to the average within-chain variance, with a value < 1.1 recommended as a stopping criterion [18]. In multivariate settings, i.e. when calibrating multiple parameters, one can enforce this across all parameters or use the multivariate adaptation of the metric [19]. Another common stopping criterion is a threshold of the (multivariate) Effective Sample Size (mESS), which accounts for autocorrelation in the chain [20], [21]. Heidelberger and Welch [22] perform hypothesis tests for stationarity of the Markov chain at different points in the chain to determine how much to remove as burn-in because the null hypothesis that the chain is stationary is rejected when that portion is included. In a similar vein, others perform hypothesis tests comparing kernel density estimates across pairs of chains to determine if they have the same distribution (which is assumed to be the stationary distribution). Metrics for such tests include the L-1 distance [23], Hellinger distance [24] or Kullback-Leibler Divergence (KLD) [25]. If any of these graphical or quantitative diagnostics indicates non-convergence, adjustments to the search process can be made, such as ''thinning'' the chain by only retaining every k samples to reduce auto-correlation, modifying the probability of using different operators that are used to propose new chain locations, adapting the algorithm's hyperparameters (e.g. covariance matrix) to improve exploration, or simply extending the search duration [26].

While these metrics are useful for identifying if an individual search process has not converged, they provide limited insights into how to improve convergence. The conventional approach of manually tuning algorithmic hyperparameters to improve performance can be laborious, and recommended default ranges may not always perform well. Ideally, an algorithm should exhibit robustness to its hyperparameterization and be primarily controlled by the number of function evaluations (NFE) [27], [28]. However, existing diagnostics do not measure this controllability. Furthermore, simply diagnosing performance of an individual search process does not provide insights into which algorithms perform well on which class of problems, and which are robust across problems. To address these limitations, we propose new diagnostic tools to evaluate MCMC algorithms and inform the choice of suitable methods for specific types of inverse problems.

Drawing from diagnostics used to evaluate the performance of multi-objective evolutionary algorithms [27], [28], in this study, we present a novel and comprehensive framework for evaluating MCMC algorithm performance.

Our approach provides visualizations that show existing diagnostic metrics in a new way, illustrating the following algorithmic characteristics:

- *Effectiveness*: A measure of the ability of an MCMC algorithm to find a posterior mode (or multiple modes) that is (are) representative of the true uncertainty, and to characterize the full posterior distribution. Existing metrics include L-1 distance, Hellinger distance, and KLD.
- *Efficiency* The speed with which the posterior is able to be characterized. Existing metrics include the ESS and mESS.
- *Reliability*: How consistently the algorithm is able to characterize the posterior across different random seeds. This is typically quantified by the GR diagnostic.
- *Controllability*: The insensitivity of an algorithm's efficiency to its hyperparameterization, a desirable property so that the user does not have to fine-tune hyperparameters to achieve good performance. This is not typically quantified in MCMC diagnostics.

Diagnosing these features collectively across algorithmic hyperparameters and random seeds fills an important gap in the literature that only diagnoses convergence of a single search process, ignoring algorithmic controllability across hyperparameterizations. The visualizations we produce of these characteristics can inform the choice of a robust MCMC algorithm and corresponding hyperparameterization, whose convergence can then be assessed using existing diagnostics. As such, our new MCMC diagnostics play a complementary role to existing MCMC diagnostics.

Our paper is organized as follows. Our methods are described in Sections II-V. Section II briefly describes the MCMC algorithms we compare, Section III outlines the experimental design used for this comparison, Section IV lists the metrics used to quantify performance, and Section V introduces the test problems on which the algorithms are evaluated. We illustrate the results of this computational experiment and our new diagnostics in Section VI. Finally, we close with our conclusions about MCMC algorithm performance illustrated by our new diagnostics in Section VIII.

II. ALGORITHMS

In this section, we describe the three Bayesian estimation algorithms examined in our study: Metropolis-Hastings (MH), Adaptive Metropolis (AM), and Differential Evolution Adaptive Metropolis with a snooker update and sampling from an archive of past states (DREAM$_{(ZS)}$). These algorithms serve as powerful tools for exploring and sampling from complex parameter spaces in Bayesian analysis. While there are other algorithms for Bayesian estimation, we limit our exploration to these three for illustrative purposes of our new diagnostics. However, our diagnostics can be extended to other algorithms.

All algorithms were implemented using the BayesianTools package in R [29], which provides general-purpose MCMC samplers for Bayesian statistics. The BayesianTools package offers a wide range of functionalities for efficient implementation and analysis of Bayesian models, making it an accessible tool for conducting advanced Bayesian inference tasks, such as comparing alternative algorithms.

A. METROPOLIS HASTINGS (MH)

The MH algorithm [30], [31] is a widely used MCMC method that enables sampling from complex posterior distributions. First, an initial parameter set θ_0 is sampled from the prior distribution and then new parameters θ' are generated (proposed) from a proposal distribution that is centered about the current location. MH proposes new parameter sets by using a symmetric proposal distribution, typically a multivariate normal distribution (MVN), as is implemented in BayesianTools. This is referred to as Gaussian mutation. A proposed move is accepted with probability α, determined by equation 2:

$$\alpha = \min\left(1, \frac{p(\theta'|x)g(\theta_t|\theta')}{p(\theta_t|x)g(\theta'|\theta_t)}\right) \tag{2}$$

where $g(\theta'|\theta_t)$ is the probability of proposing parameters θ' given the current parameters are θ_t, and $g(\theta_t|\theta')$ is the reverse. Note that, $g(\theta'|\theta_t) = g(\theta_t|\theta')$ if the proposal distribution is symmetric. This is referred to as the Metropolis step, or accept-reject step.

In BayesianTools, the initial samples of the chain can be optimized at an estimate of the maximum of the posterior distribution, with the goal of reducing the amount of burn-in by starting in a high posterior density region. This is controlled by a binary hyperparameter Optimize = True or False. If true, BayesianTools utilizes the Brent algorithm [32] for single-parameter estimation problems, and the Nelder-Mead algorithm [33] for multi-dimensional problems, both of which are derivative-free. Nelder-Mead algorithm may converge to a non-stationary point [34], and it is a local optimizer, therefore it may not do well on multi-modal problems. The other hyperparameters of MH algorithm are the total number of function evaluations, the number of chains, and percent of function evaluations to remove as burn-in (see Table 1 for a list of the hyperparameters in each algorithm).

B. ADAPTIVE METROPOLIS (AM)

MH provides a foundational framework for Bayesian inference and has been successfully applied in various fields. However, one limitation of MH is the fixed proposal distribution, which may not effectively explore high-dimensional or multi-modal parameter spaces. To address this limitation and improve exploration efficiency, the AM algorithm [35] incorporates adaptive strategies for updating the covariance of the proposal distribution throughout the search. This adaptation is determined by the points sampled during the MCMC process. By adaptively updating the proposal covariance, AM strikes a balance between exploration and exploitation in the parameter space. It allows the algorithm to explore regions of high uncertainty by increasing the

variance when uncertainty across sampled points is high, leading to better mixing of the Markov chains. However, it also allows the algorithm to exploit regions of high probability density by decreasing variance when uncertainty across sampled points is low, thus improving convergence. While this may be unnecessary for low-dimensional problems for which MH may be faster, the adaptive nature of AM makes it more effective in high-dimensional parameter spaces and when dealing with complex posterior distributions. This adaptivity enhances the exploration capabilities of the algorithm, resulting in improved efficiency and convergence rates [36]. In BayesianTools, adaptation is controlled by two hyperparameters: AdaptStart, which indicates the percent of evaluations after burn-in at which adaptation begins, and AdaptInterval, which indicates the fraction of remaining evaluations after AdaptStart at which adaptation occurs.

We allow AM to be employed with delayed rejection, also called Delayed Rejection Adaptive Metropolis (DRAM) [37]. In DRAM, once a proposed point has been rejected, instead of proceeding to the next time step and remaining in the current state, a second-stage proposal is made that depends on *both* the current state and the state that was just proposed and rejected. The second-stage proposal is then accepted or rejected based on a modified acceptance probability that preserves reversibility of the Markov chain. This can be repeated multiple times, the number of which is controlled in our experiment by the parameter DRlevels (see Table 1). We also allow for optimization of initial starting points in the AM search, controlled by a binary Optimize hyperparameter, as in MH.

C. DIFFERENTIAL EVOLUTION ADAPTIVE METROPOLIS (DREAM)

While the ability to adapt the proposal distribution through AM can speed up convergence with respect to MH, it is still limited by using a single proposal operator (typically, Gaussian mutation). The DREAM$_{(ZS)}$ algorithm [38] is a population-based MCMC method that advances AM further by adding additional proposal operators to the AM algorithm: differential evolution (DE) and a snooker update (S). This can further enhance exploration on high-dimensional, multi-modal problems, but may come at the expense of deeper exploitation of high-posterior regions. For simplicity, we refer to this algorithm as simply "DREAM" throughout the remainder of the paper.

The population of DREAM refers to the states of multiple chains, as well as an archive of their past states. These are used jointly to propose new chain locations using operators beyond Gaussian mutation, including DE and a snooker update, which are accepted according to the Metropolis rule. DE is a vector translational operator originally developed for use in evolutionary optimization algorithms [39]. Mathematically, DE can be described by equations 3-4 [40]:

$$\theta_i' = \theta_{i,t} + \gamma(1+e)\left[\sum_{n=1}^{p}\theta_{j(n)} - \sum_{m=1}^{p}\theta_{k(m)}\right] + \epsilon N(0,1) \quad (3)$$

$$\gamma = \frac{2.38}{\sqrt{p * d}} \quad (4)$$

where $\theta_{i,t}$ and θ_i' are the current and proposed states of the i-th chain, respectively; $\theta_{j(n)}$ and $\theta_{k(m)}$ are the n-th and m-th of p samples from the archive of current or past states of the j-th and k-th chains, respectively; d is the problem dimension (i.e., number of model parameters); e is a constant chosen by the user to scale γ if desired (ter Braak and Vrugt [41] choose the default value of γ in equation 4 to yield acceptance rates close to 0.44 for $d = 1$ and 0.23 for large d, which have been shown numerically and theoretically to be optimal acceptance rates for random walk Metropolis [42], [43]); and ϵ is the variance of a Gaussian mutation after DE translation, whose value is also chosen by the user.

The DE translation in equation 3 is typically only applied to some of the dimensions. These are referred to as "crossover points" and the number of crossover points is determined by the nCr parameter. The value of this parameter can be updated throughout the search with frequency determined by the parameter UpdateInterval. Similarly, the archive of past states, $\mathbf{Z}$, is updated with frequency zUpdateFrequency.

A snooker update is another vector translational operator originally proposed by Gilks et al. [44] to adapt sampling in the direction of the highest density. DREAM$_{(ZS)}$ uses an updated snooker proposal operator developed by ter Braak & Vrugt [41], described mathematically by equation 5:

$$\theta_i' = \theta_{i,t} + \gamma_s(\theta_{j,t}^P - \theta_{k,t}^P) \quad (5)$$

where γ_s is another constant hyperparameter of the algorithm, while $\theta_{j,t}^P$ and $\theta_{k,t}^P$ are orthogonal projections of $\theta_{j,t}$ and $\theta_{k,t}$ onto the line $\theta_{i,t} - \theta_{n,t}$, where $\theta_{n,t}$ is the current state of another chain, n.

The additional operators of DREAM, as well as its use of an archive and interaction across chains, serve several beneficial purposes. The archive, which maintains a history of accepted samples from all chains, enables a more efficient exploration of the parameter space and improved mixing of the chains. By sampling from the past archive, the algorithm gains access to valuable information about the posterior distribution, enhancing its ability to explore diverse regions and locate multiple modes. The incorporation of DE and snooker moves within DREAM$_{(ZS)}$ further enhances exploration by introducing a stochastic perturbation mechanism. This mechanism helps to overcome local optima and encourages the chains to traverse the posterior distribution more effectively. Finally, the interaction across chains allows for greater exploration and facilitates convergence to the same posterior across chains [38], [45].

III. COMPUTATIONAL EXPERIMENT

In order to evaluate the performance of the MCMC algorithms used in this study, a comprehensive experimental setup was devised, representing the key contribution of this paper. The experimental design, which is inspired by [27] and [28], aims to assess the effectiveness, efficiency, reliability, and

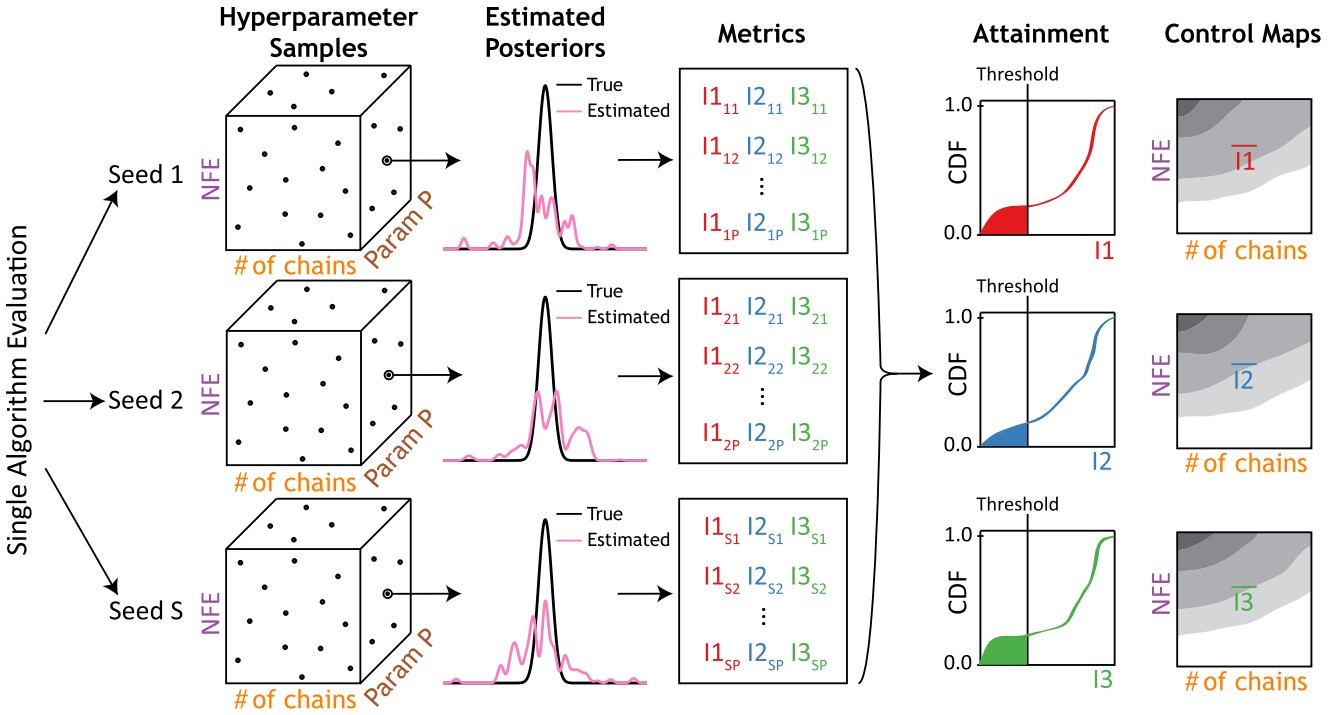

FIGURE 1. Experimental design of this study. For each algorithm and test problem, calibration is performed across a Latin hypercube sample of algorithm hyperparameters for multiple seeds. Performance metrics are computed based on the proximity of the estimated posterior to the true posterior. The reliability of the algorithm is illustrated by CDFs of the probability of attaining certain performance levels on the metrics, while the effectiveness, efficiency and controllability are illustrated in control maps of the average performance metric as a function of the number of function evaluations (NFE) and chains. This figure is adapted from Reed et al. [28].

controllability of the MCMC algorithms in converging to the true posterior distribution.

Figure 1 illustrates the experimental design that enables this assessment. The first step is to generate a Latin hypercube (LH) sample of algorithmic hyperparameters. Here we use a sample of 1000. Two of the hyperparameters are the number of function evaluations (NFE) and number of chains. For each of the LH samples, MCMC is performed with the corresponding hyperparameters and the posteriors are estimated empirically. The final posteriors consist of the elements from all chains, excluding the initial burn-in period. For instance, with 1000 iterations, 5 chains, and a 100-iteration burn-in, the resulting posterior consists of (1000 - 100) * 5 = 4500 chain locations. Since the Monte Carlo aspect of MCMC is random, this process is repeated for multiple random seeds, here 25.

Next, several metrics of the algorithm's *effectiveness* are computed (described in Section IV) for each random seed of each LH sample. The *reliability* in achieving these metrics is visualized by a Cumulative Distribution Function (CDF) across random seeds, also called an "attainment map" as it illustrates the probability of attaining different metric values. The *efficiency* is visualized by a contour map of the average metric across random seeds of each LH sample, shown on a 2D projection of the LH samples' NFE and number of chains. The sooner effective values are reached vs. NFE, the more efficient the algorithm. This plot, also called a "control map", illustrates how *controllable* the algorithm is; the noisier the contour map the less its performance is controlled by the NFE and number of chains and more by its other hyperparameters.

We also measure controllability quantitatively by performing variance-based sensitivity analysis, decomposing how much variance in the performance metric is explained by each hyperparameter. The more variance explained by NFE (and subsequently, the number of chains), the more controllable the algorithm, as these are the easiest hyperparameters for the user to set. The fraction of the variance in the performance metric Y explained by the i-th hyperparameter X_i individually is denoted its first-order sensitivity index, S_i:

$$S_i = V_i / Var(Y) \tag{6}$$

$$V_i = Var(E[Y|X_i]). \tag{7}$$

Any remaining variability is assumed to be explained by interactions across hyperparameters. Sensitivity indices were estimated using the method of Plischke et al. [46] using the Python SALib package [47].

The ranges of the hyperparameters for the LH samples are detailed in Table 1. These ranges were informed by values from the literature and were carefully selected to cover a broad spectrum of possible configurations, ensuring a thorough exploration of the algorithm's behavior [29], [38]. By varying the hyperparameters, we not only are able to assess the algorithm's controllability, but also to identify

TABLE 1. Ranges of algorithmic hyperparameters sampled uniformly by Latin hypercube sampling.

Hyperparameters across algorithms		
Hyperparameter	Description	Range
NFE	Number of function evaluations	10,000-200,000
nChains	Number of chains	2-20
Burn-in	Percent of function evaluations to discard	1-20
Additional MH and AM hyperparameters		
Optimize	Binary variable indicating whether to optimize the starting locations	0 (no) or 1 (yes)
Additional AM hyperparameters		
AdaptStart	Percent of Evaluations after Burn-in at which adaptation begins	0.5-5
AdaptInterval	Fraction of remaining evaluations after AdaptStart at which adaptation occurs	0.1-1
DRlevels	Number of levels for a delayed rejection sampler	1-2 (integer)
Additional DREAM(ZS) hyperparameters		
Adapt	Portion of iterations used in adaptation	0-1
P(snooker)	Probability of a snooker update at each iteration	0-1
nCr	Number of crossover points, i.e. dimensions of the current state changed in the proposal	1-5
p	Number of state pairs used to generate proposal with DE	1-3
ϵ	Variance of Gaussian mutation of DE translation (equation 3)	0-0.00005
e	Constant in equation 3 for DE proposal computation	-0.1-0.1
UpdateInterval	Interval number of iterations at which P(crossover) is updated	1-20
zUpdateFrequency	Interval number of iterations of evaluations after burn-in at which the archive is updated	1-20

the settings that yield optimal results for different types of problems (e.g. high-dimensional or multi-modal). Sensitivity to additional hyperparameters could be explored in future work, such as the initial covariance matrix of the Gaussian proposal distribution, or the interval of samples that should be dropped via thinning. Sensitivity to thinning could be further investigated to determine the extent that autocorrelation decreases the effective sample size which also increases the standard error estimates of the posterior mean. Investigating the sensitivity on an MCMC algorithm's performance to the thinning interval could inform the choice of effective ranges to reduce the impact of autocorrelation on the reliability of MCMC simulations.

The experimental framework was implemented on the Rivanna high-performance computing cluster at the University of Virginia. The insights derived from this experiment can provide guidance for selecting appropriate algorithms and corresponding configurations for inverse problems with the tested characteristics, as well as inform how to develop new algorithms with improved controllability by adapting more sensitive hyperparameters throughout the search. Finally, it illustrates a new framework for evaluating MCMC algorithms developed in the future.

IV. METRICS

To evaluate the effectiveness of the MCMC algorithms, we use three performance metrics that quantify different aspects of convergence: the GR diagnostic [18], KLD [48], and WD [49]. These metrics provide valuable insights into the quality of the MCMC samples and the approximation of the target distribution.

A. GELMAN-RUBIN (GR) DIAGNOSTIC

The GR diagnostic is a widely used measure to assess convergence when multiple, independent MCMC chains are employed and the true posterior is unknown. It compares the within-chain variance to the between-chain variance:

$$\hat{R} = \sqrt{\frac{\hat{V}}{\hat{W}}} \tag{8}$$

where $\hat{V}$ is the estimated marginal posterior variance of the target parameter across all chains and $\hat{W}$ is the estimated average within-chain variance of the target parameter. A GR value close to 1 indicates convergence to the same variance across chains, making it an easy-to-interpret metric. The GR diagnostic is computed using BayesianTools.

We note that the GR diagnostic is meant to be used to ensure convergence to the same variance across *independent* chains, and is therefore not an appropriate measure of convergence for DREAM since the chains communicate. This communication will likely result in a low GR early in the search, even if the algorithm has not converged. However, consistent variance across chains may not be an appropriate measure of convergence even in the case of independent chains, as the chains could represent consistently poor approximations of the true posterior. Despite these limitations, GR is still the most commonly employed MCMC convergence metric when the posterior is unknown, including for the DREAM algorithm [38]. As such, we still compute the GR for all algorithms, but also compute additional performance metrics that allow us to assess the utility of GR as an MCMC performance metric.

B. KULLBACK-LEIBLER DIVERGENCE (KLD)

GR is a proxy measure of convergence used when the true posterior is unknown. However, as discussed above, it can prematurely indicate convergence, particularly when the true posterior is multi-modal. For test problems where the true posterior is known, we can assess convergence using the KLD. KLD measures the difference between two probability distributions, $D_{\mathrm{KL}}(P \parallel Q)$, as the integrated divergence in probability of one pdf $P(\theta)$ (here, the estimated posterior) to

another $Q(\theta)$ (here, the true posterior):

$$D_{\mathrm{KL}}(P \parallel Q) = \int P(\theta) \log\left(\frac{P(\theta)}{Q(\theta)}\right) d\theta \tag{9}$$

KLD is commonly used in Bayesian statistics to assess the approximation of the true posterior distribution obtained from an MCMC algorithm when the true posterior is known, as is the case on test problems. It provides a measure of the dissimilarity between the approximate and true posterior distributions, allowing for flexible comparison of distributions with different parametric forms. However, the choice of the reference distribution (P vs. Q) can influence the results, as the measure is not symmetric. While there is a symmetric measure of KLD (Jeffrey's divergence) [50], we simply set the reference distribution to the true posterior for consistency. We use the R function KL.divergence in the FNN library to computed KLD [51], [52].

C. WASSERSTEIN DISTANCE (WD)

The WD, or ''Earth mover's distance'' is another measure of the similarity between two distributions. It measures the minimum transport distance to transform one probability distribution into another [53]. It can be used in MCMC diagnostics to compare the true posterior distribution to the estimated posterior distribution obtained from the algorithm:

$$\mathrm{W(P, Q)} = \inf_{\gamma \in \Gamma(P,Q)} \int \int \|x - y\| \, d\gamma(x, y) \tag{10}$$

where $W(P, Q)$ represents the WD between distributions P and Q; inf denotes the infimum, which represents the minimum value over all possible transports γ that move mass from P to Q; $\Gamma(P, Q)$ is the set of all joint probability distributions $\gamma(x, y)$ with marginals P and Q; and $\|x - y\|$ represents a chosen distance metric between points $\overrightarrow{x}$ and $\overrightarrow{y}$ in the underlying space.

The WD provides a measure of the discrepancy between two distributions, considering their underlying structure [54]. It can handle distributions with different supports. However, it can be computationally demanding, especially for high-dimensional distributions. Additionally, the choice of the distance metric may influence the results. To estimate WD, we generate n points $\overrightarrow{y}$ from the true posterior Q where n is equal to the total number of samples $\overrightarrow{x}$ from the MCMC chains after removing burn-in, which represent the estimated posterior P. The WD between these sets of points is computed using the Sinkhorn approximation [55] from the Python geomloss library, with transport distance between two points quantified by Euclidean distance.

Comparing the KLD and WD, KLD quantifies how dissimilar the estimated posterior probability is at each point θ compared to the true posterior probability, while WD compares how far the distributions are from one another in parameter-space. As such, KLD may be a better approximation of how close the estimated posterior is from the truth, while WD may be a better approximation of how far the search is from finding the true posterior in parameter-space.

V. TEST PROBLEMS

Because MCMC is typically applied to estimate the parameters of complex physical models, it would be useful to apply our diagnostics to such models. However, the KLD and WD metrics require a known posterior, so one would have to set a synthetic true parameter set to apply our diagnostics to a physical model. The posterior would then be a dirac delta function at the synthetic truth, and the KLD would be infinite. However, the WD could still be computed as the average Euclidean distance between all chain elements and the synthetic truth. This would capture closeness of the estimated posterior to the truth in parameter space, but not probability space. Because of these challenges, we simply focus our diagnostics on two analytical test problems that address two prevalent challenges encountered in complex models: high dimensionality and multi-modality.

A. HIGH-DIMENSIONAL TEST PROBLEM

Physical and data driven systems often involve a large number of interconnected variables, leading to high-dimensional parameter spaces. To simulate such scenarios, we employ a 100-dimensional multivariate normal distribution with a mean of $[0]^d$ and covariance Σ where the off-diagonal elements $\sigma_{i,j} = \frac{1}{2}\sqrt{i * j}$ $\forall i \neq j$ and the diagonal elements $\sigma_{i,i} = i$. This test problem is commonly used to represent high-dimensional data [40].

The choice to target high dimensionality is motivated by the need to develop robust techniques capable of effectively exploring and optimizing parameter spaces in physical and statistical models. Relevant model applications span a wide range of fields including but not limited to machine learning [56], climate [57], and finance [58].

B. BIMODAL TEST PROBLEM

Multi-modal behavior, characterized by the simultaneous existence of distinct modes or regions of high probability in the parameter space, is a prevalent phenomenon observed in various domains, including machine learning and statistics [59], natural language processing [60], climate modeling [61], and economics [62], where data often exhibits multiple diverse patterns or states. To address this characteristic, we employ a 10-dimensional bimodal mixed Gaussian distribution as our multi-modal test problem. The bimodal mixed Gaussian distribution consists of two distinct modes, each following a Gaussian distribution with means of $[-5]^d$ and $[5]^d$ and a common covariance matrix $\Sigma = I$, the identity matrix. The mode with mean $[-5]^d$ occurs with probability 1/3 and the mode with mean $[5]^d$ with probability 2/3. By employing such a distribution, we can assess the ability of our proposed approach to effectively locate and characterize multiple optima within the parameter space, a key challenge encountered in physical modeling.

VI. RESULTS AND DISCUSSION

A. DIAGNOSTICS ON 100D MVN TEST PROBLEM

In this section, we present our diagnostics on the 100D MVN test problem. Figure 2 displays two sets of maps to illustrate the controllability, reliability, and efficiency of the MCMC algorithms using the WD metric: Control and Attainment Maps. SI Figures S1-S2 show the same maps for the KLD and GR of the first dimension, respectively, which revealed similar findings for KLD as presented in Figure 2, while all algorithms did well on GR across hyperparameterizations. Because our sensitivity analysis revealed that MH and AM were most sensitive to whether or not optimization was used to initialize the starting locations of each chain, we present these maps separately for the cases where optimization = True vs. False, yielding five algorithms for the comparison: MH_{noOpt}, MH_{opt}, AM_{noOpt}, AM_{opt}, and DREAM.

The control maps in Figures 2a-2e illustrate the average WD between the estimated and true posterior across 25 random seeds as a function of the algorithm's chain count (x-axis) and number of function evaluations (NFE; y-axis). The sooner a low value in blue is reached along the y-axis, the more efficient the search is. Noise in the control maps indicates less controllability, i.e. greater sensitivity to other hyperparameters beyond NFE and number of chains. Some hyperparameter combinations failed to yield posterior distributions and are shown as gray.

When using optimization to initialize chain locations, MH and AM both perform very poorly (Figures 2d-2e). MH yields high WD across all combinations of chains and NFE (Figure 2d). AM's adaptation begins to improve performance at low chain counts (Figure 2e), but both algorithms perform much better when optimization is not used to initialize chain locations (Figures 2a-2b). We investigate the reason optimization performs poorly through additional visual diagnostics of the estimated marginal posterior distributions in Figure 3. Without optimization, MH and AM show slower convergence at higher chain counts (Figures 2a-2b), indicating that for high-dimensional but unimodal problems, it is better for these algorithms to maximize iterations of a few chains to increase exploitation than to spread them across chains to increase exploration. DREAM (Figure 2c) is less hampered by spreading its iterations across chains thanks to the interaction between them, whereby the states of multiple chains are used to propose new chain locations. This insensitivity to chain count results in more controllability and robustness in DREAM's performance. However, its robustness does come at the expense of optimality under certain configurations, as AM's adaptation initially slows convergence, but ultimately results in better posteriors. Consequently, AM with low chain counts is the best choice if one is not computationally limited and knows their problem is unimodal, but DREAM is the best choice if one is more computationally limited.

In addition to finding posteriors that match the true posterior across hyperparameters, we would also like algorithms that do this reliably across random seeds. We investigate this for the 100D MVN problem using attainment maps in Figure 2f, which illustrate the probability of attaining different WDs across random seeds. The more blue the attainment map, the higher the probability of attaining low WDs, i.e. the more reliably effective the algorithm is. MH and AM with optimization are shown to be not only inefficient, but unreliable, with a low probability of attaining low WDs. AM without optimization has the highest probability of achieving the lowest WDs (e.g. lowest WD achieved 50% of the time); however, this comes at the expense of increased variability as the probability of attaining near optimal WDs is much lower than both MH and DREAM. We see from the control maps that the higher WDs occur at higher chain counts and lower NFE. DREAM has the highest probability of attaining near-optimal WDs, proving to be not only the most robustly efficient across hyperparameters, but also the most reliably efficient across random seeds.

To verify the patterns seen in the control maps, we illustrate the posterior marginals for a random seed from the hyperparameter nearest each corner and the centroid of the control maps. Figure 3 illustrates these marginals for the 50th dimension of the 100D MVN, while SI Figures S3-S4 show them for the 1st and 100th dimensions, respectively, which yield similar conclusions. Across algorithms, as the NFE increases (higher plots), the posterior distributions tend to more closely approach the true posterior (black), with the exception of MH with optimization (light green), which performs poorly across hyperparameters. AM with optimization (light blue) also poorly matches the true posterior, but moves in the right direction as NFE increases for low chain counts. MH and AM without optimization (dark green and dark blue) ultimately come closest to the true posterior, but DREAM (red) performs better when the chain count is high but NFE is low (Figure 3e), illustrating its improved robustness at the expense of optimality. Further investigation is needed to understand why using optimization to initialize chains in MH and AM does not direct the search toward the true, single mode. It appears the Nelder-Mead optimization does not converge to the true mode, instead initializing the search in different regions of the space for different chains, and the algorithm takes a long time to explore beyond those estimated modes toward the truth.

Finally, we combine our illustration of reliability and controllability in Figure 4, which illustrates the CDF of WD for each hyperparameter. SI Figures S5-S6 show the same for the KLD and GR of the first dimension, respectively. In Figure 4, the color of each CDF represents the value of the hyperparameter that most explains variability in WD (yellow = low, purple = high). This hyperparameter is indicated by the variance decomposition shown in Figure 4f. The steeper the CDF, the more reliable the algorithm; the closer the CDFs are to 0, the more effective it is; and the more sensitive the algorithm is to NFE (blue), the more controllable it is. Fortunately, NFE is the most influential hyperparameter across all algorithms except for MH_{opt}, which is most sensitive to the number of chains (orange). AM_{opt} is also

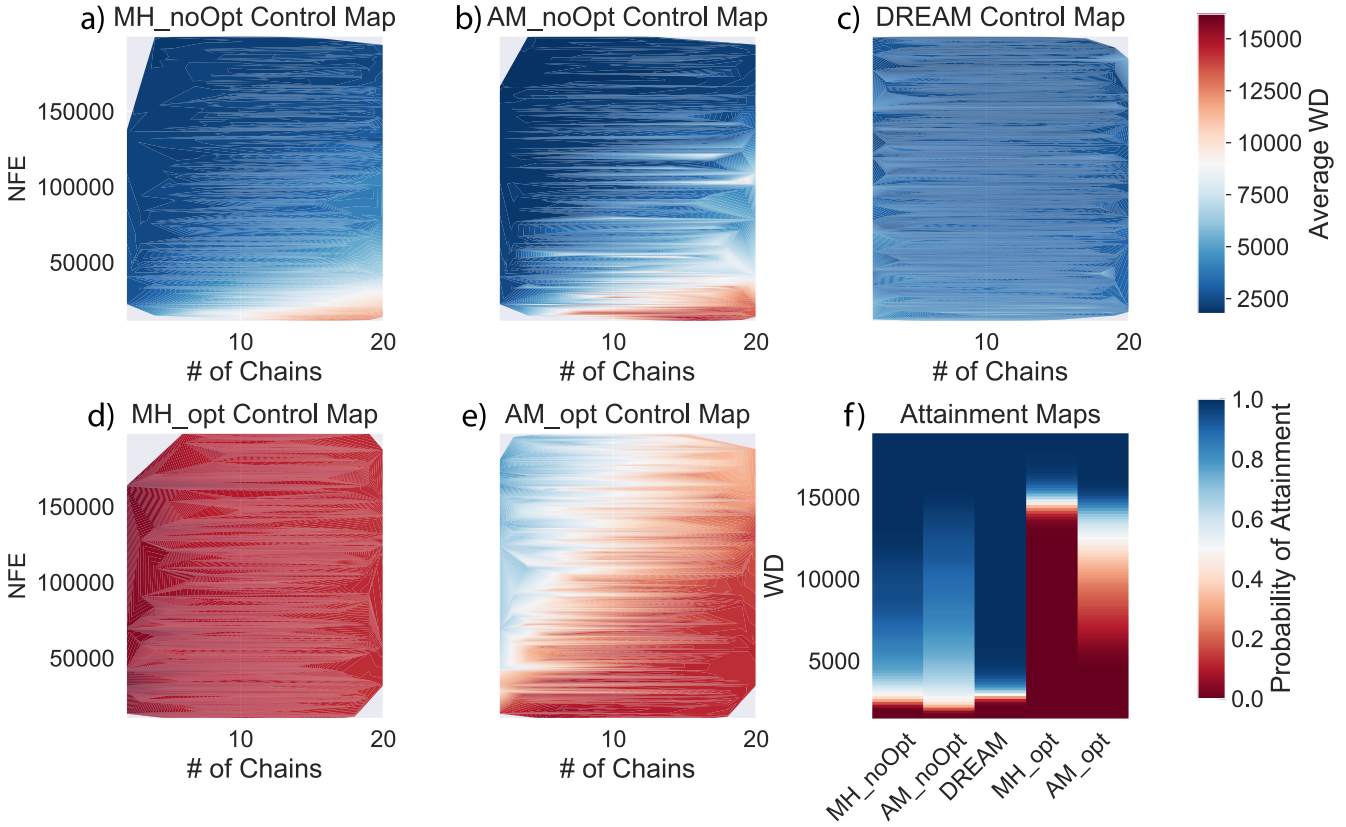

FIGURE 2. (a-e) Control maps for the 100D MVN test problem illustrating the average Wasserstein distance (WD) across random seeds as a function of the number of function evaluations (NFE) and number of chains for (a) MH without optimization, (b) AM without optimization, (c) DREAM, (d) MH with optimization, and (e) MH with optimization. (f) Attainment maps illustrating the probability of attaining different WDs (shown on the y axis) across all seeds and hyperparameters for each algorithm.

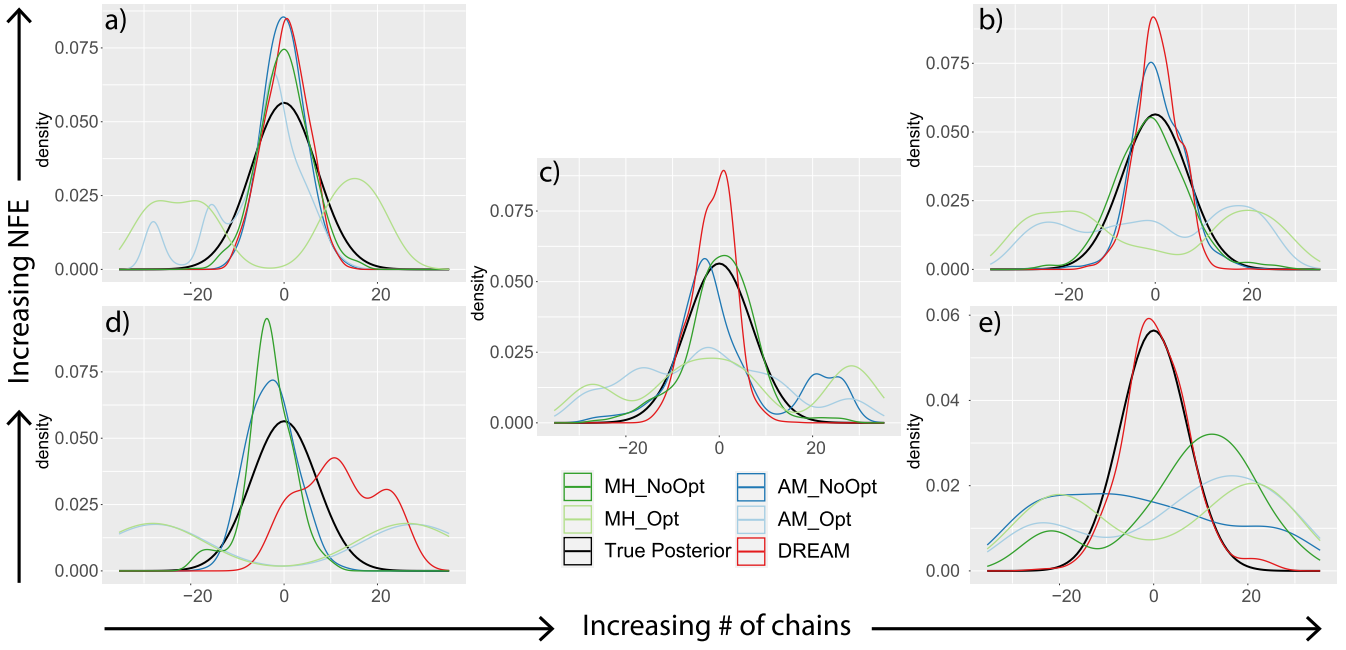

FIGURE 3. Posterior marginals of the 50th dimension of the 100D MVN test problem when using the hyperparameter closest to (a) the least number of chains and the most NFE, (b) the highest number of chains and the most NFE, (c) the median number of chains and the median NFE, (d) the least number of chains and the least NFE, and (e) the most number of chains and the least NFE.

fairly sensitive to the number of chains. This sensitivity to the number of chains, and the multimodal nature of the posteriors estimated by these algorithms in Figure 3, suggests that the optimization may be resulting in different chains converging

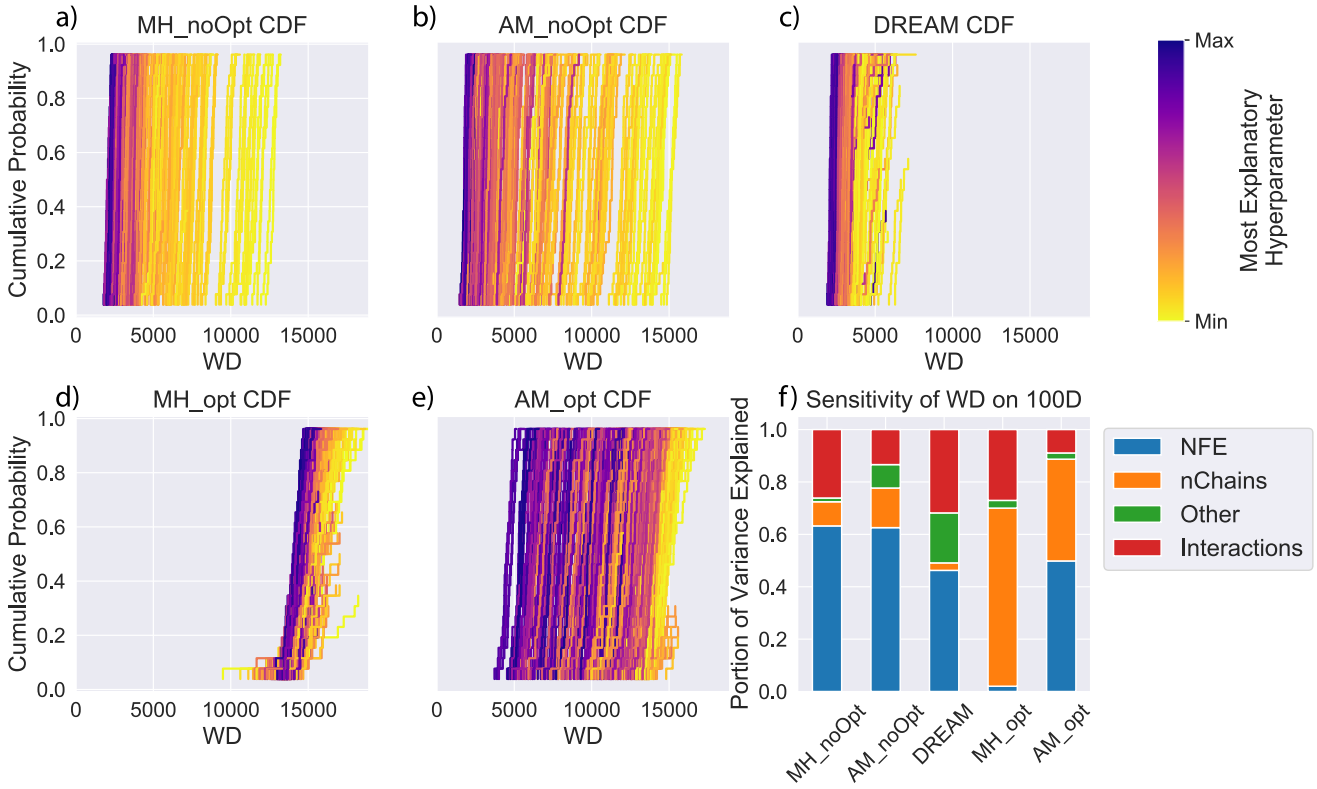

FIGURE 4. (a-e) CDFs of WD across random seeds for each hyperparameter on the 100D MVN test problem. The color of the hyperparameter indicates the value of the parameter to which that algorithm's WD was most sensitive. (f) Decomposition of how much variance in WD is explained by each hyperparameter and their interaction for each algorithm.

to different modes near their starting locations, which may not be near the true mode.

Analyzing the CDFs, they are steep for nearly all algorithms and hyperparameters, indicating reliability across random seeds. However, they are far more consistently close to 0 for DREAM, which aligns with the findings illustrated by the control maps. Across all algorithms, as the most explanatory hyperparameter increases, the CDFs tend to converge toward lower WD values. This is desirable for the algorithms that are most sensitive to NFE. Among these algorithms (all but MH_{opt}), DREAM exhibits the greatest sensitivity to other parameters beyond the number of chains (green) and interactions between hyperparameters (red). This suggests that although this algorithm is robust across hyperparameters, the additional operators do reduce controllability. This could perhaps be reduced by adapting their values and probabilities throughout the search as has proven successful in multi-objective evolutionary algorithms [28], something that could be explored in future work on algorithm development.

B. DIAGNOSTICS ON 10D BIMODAL MIXED-GAUSSIAN TEST PROBLEM

Here, we present our diagnostics assessing the performance of MH, AM, and DREAM on the 10D Bimodal Mixed-Gaussian test problem. Interestingly, unlike for the 100D MVN, the performance of MH and AM was not sensitive to whether optimization was used to initialize chain locations, so we include all hyperparameters together in our visualizations. We hypothesize that the different estimated modes across chains from the Nelder-Mead algorithm was less problematic than for the 100D MVN problem because there is in fact more than one mode on the bimodal problem.

Similar to the 100D MVN test problem, we display the control and attainment maps on the bimodal problem for the three MCMC algorithms in Figure 5. However, unlike for the 100D MVN problem, different metrics yielded different conclusions, so we show these maps for all three metrics: WD (Figures 5a-5d), GR of the first dimension (Figures 5e-5h), and KLD (Figures 5i-5l).

Examining the control maps, it's clear that DREAM exhibits better performance in achieving lower values of GR and WD (Figures 5d and 5h) compared to MH (Figures 5b and 5f) and AM (Figures 5c and 5g). DREAM also appears more controllable, with low GR values regardless of the NFE and number of chains, and WD improving for higher NFE. On the contrary, WD is poor for MH and AM regardless of the hyperparameterization, while GR is controlled primarily by NFE. DREAM is also shown to be more reliable on these metrics by the attainment maps (Figures 5a and 5e), as DREAM has a higher probability of achieving lower values of WD and GR across random seeds than MH and AM. However, we should note the comparison on GR is not fair since the DREAM chains are not independent,

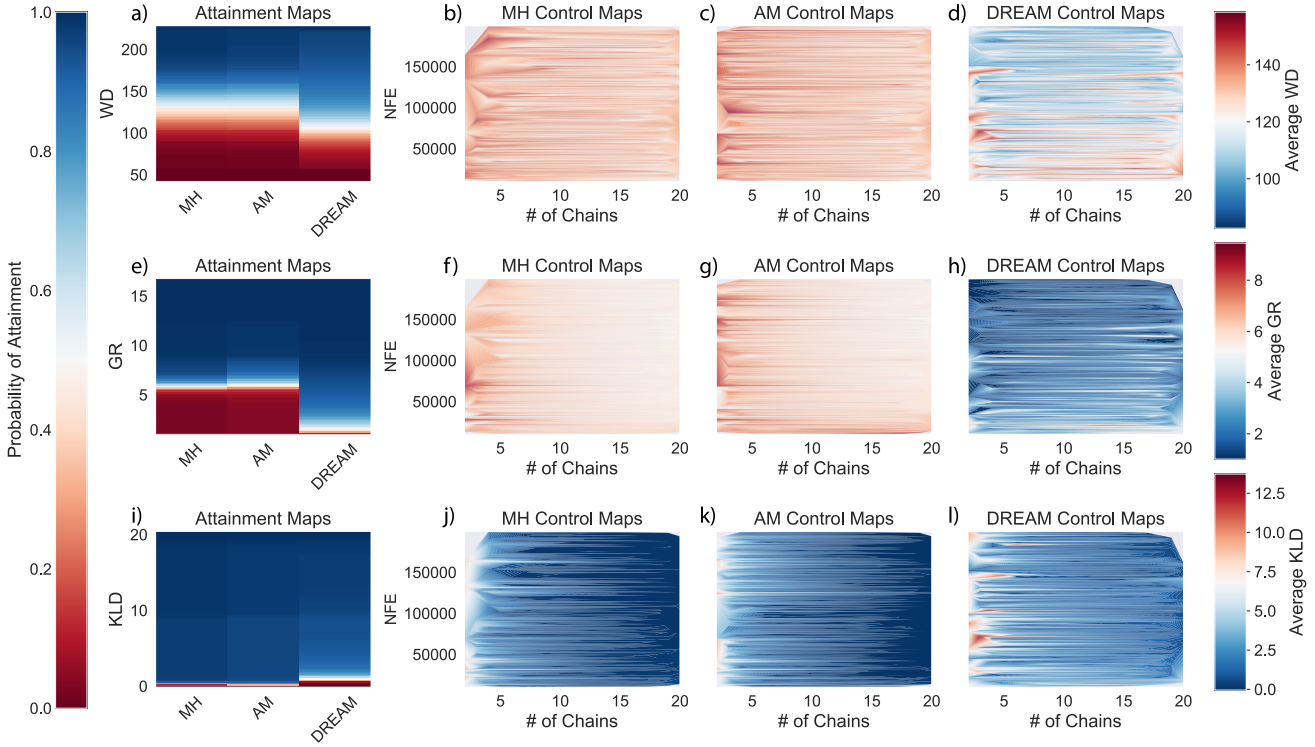

FIGURE 5. Attainment and control maps of each algorithm on the 10D Bimodal test problem based on (a-d) WD, (e-h) GR diagnostic of the first dimension, (i-l) KLD.

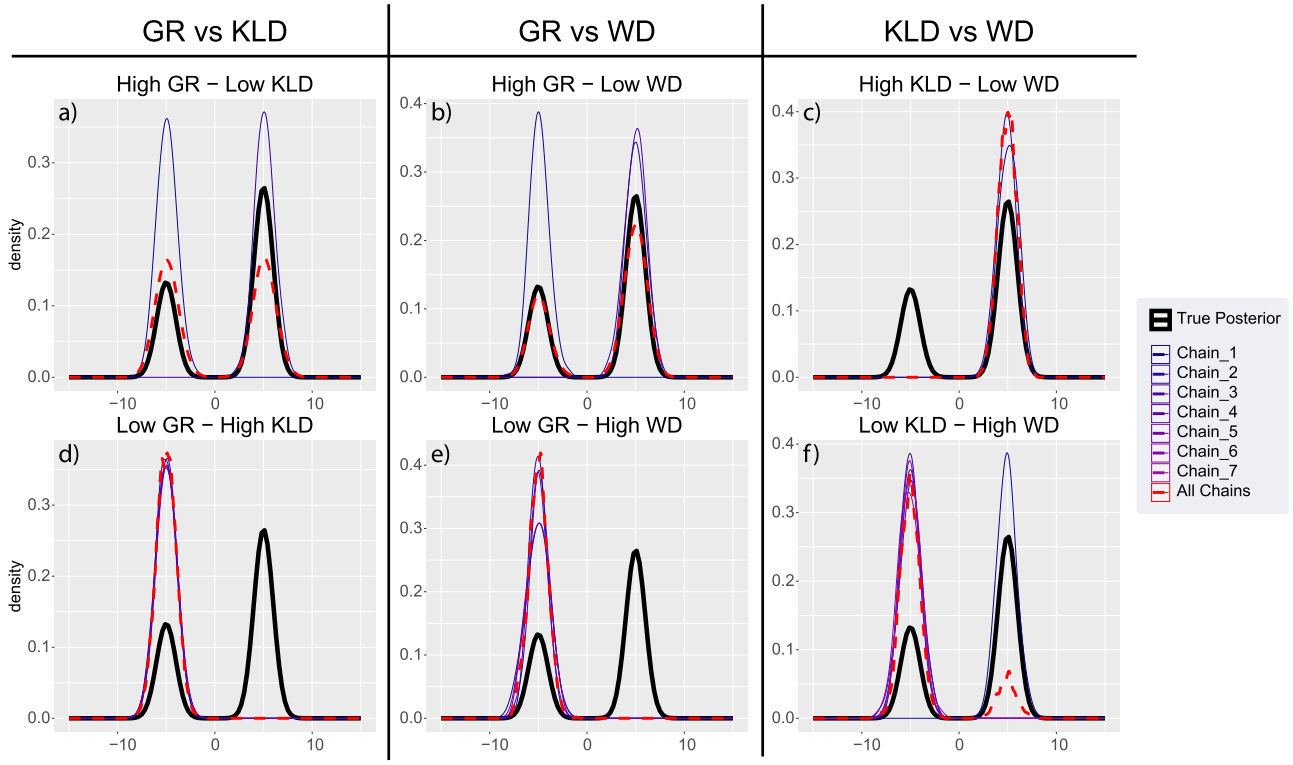

FIGURE 6. Comparison of the estimated MH marginal posterior of the first dimension of the 10D bimodal test problem from individual chains and across chains using select hyperparameter sets with (a) a high GR and low KLD and (d) the reverse; (b) a high GR and low WD and (e) the reverse; and (c) a high KLD and low WD and (f) the reverse.

thus potentially providing a false sense of improved convergence.

This false sense of improved convergence is confirmed by the control and attainment maps of KLD, which tell a

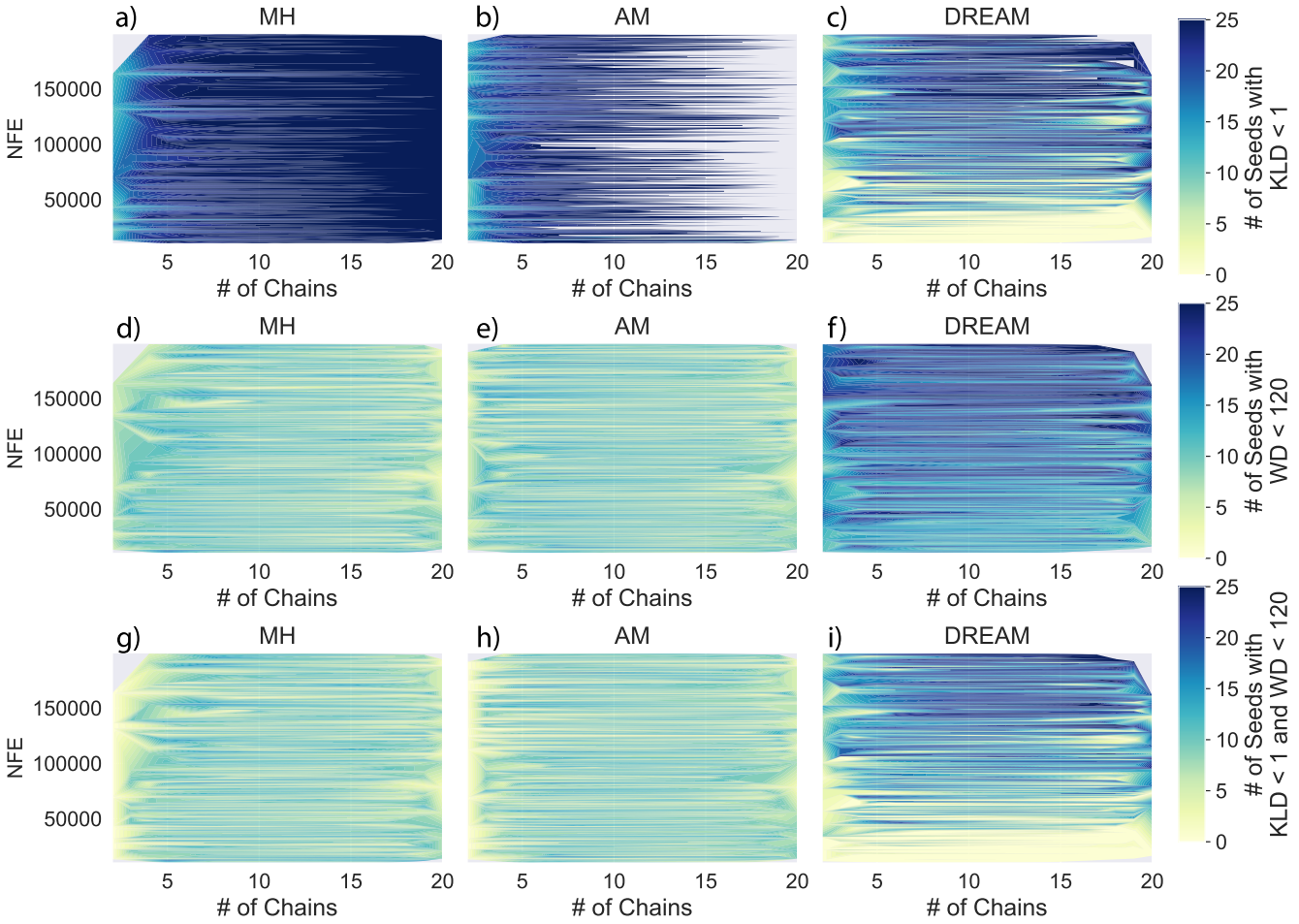

FIGURE 7. Control maps showing the number of seeds (out of 25) of each algorithm that achieved (a-d) KLD<1, (d-f) WD<120, and (g-i) both on the 10D Bimodal test problem.

different story. On this metric, all three algorithms show good performance in achieving low values of KLD, and in fact, MH and AM seem to outperform DREAM in achieving lower KLD values across hyperparameterizations and seeds. This is particularly true when employing a small (near 2) or high (near 20) number of chains, with all algorithms performing similarly at moderate numbers of chains (near 10). One can also see that the number of chains appears to be the controlling hyperparameter for this metric, similar to GR for MH and AM, but different from WD for DREAM.

To understand why conclusions about which algorithms perform best differ under these different metrics, we selected individual hyperparameterizations from the Latin hypercube sample of MH that yielded high values of one metric and low values of another. Figure 6 compares the true posterior marginal of the first dimension (black) to the estimated posterior marginals when using the elements of each individual chain of these hyperparameterizations (colored lines), as well when using the elements from all chains (red, dashed line). SI Figures S7-S8 show similar results for the 5th and 10th dimensions.

Analyzing the GR vs KLD plots (Figures 6a and 6d), we see that individual chains from the LH sample with a low KLD and high GR tend to find only one mode. Since these modes differ across chains, the GR diagnostic is high. However, the proportion of chains finding each mode is similar to those mode's likelihood, resulting in a close approximation to the true posterior across chains, i.e. a low KLD. Conversely, individual chains from the LH sample with a high KLD and low GR each converge to the same mode, resulting in low GR values. However, that mode is the less probable one, resulting in a high KLD. These results confirm what we would expect from theory [25]. Moving on to the GR vs WD plots (Figures 6b and 6e), we see similar phenomena. Individual chains from the LH sample with a high GR and low WD each identify different modes of the distribution, but in similar proportions to their likelihood, resulting in a close approximation to the true posterior across them. Conversely, individual chains from the LH sample with a high WD and low GR only detect the less likely mode. These findings again confirm theoretical understandings of these metrics, and illustrate that the GR diagnostic can be a poor

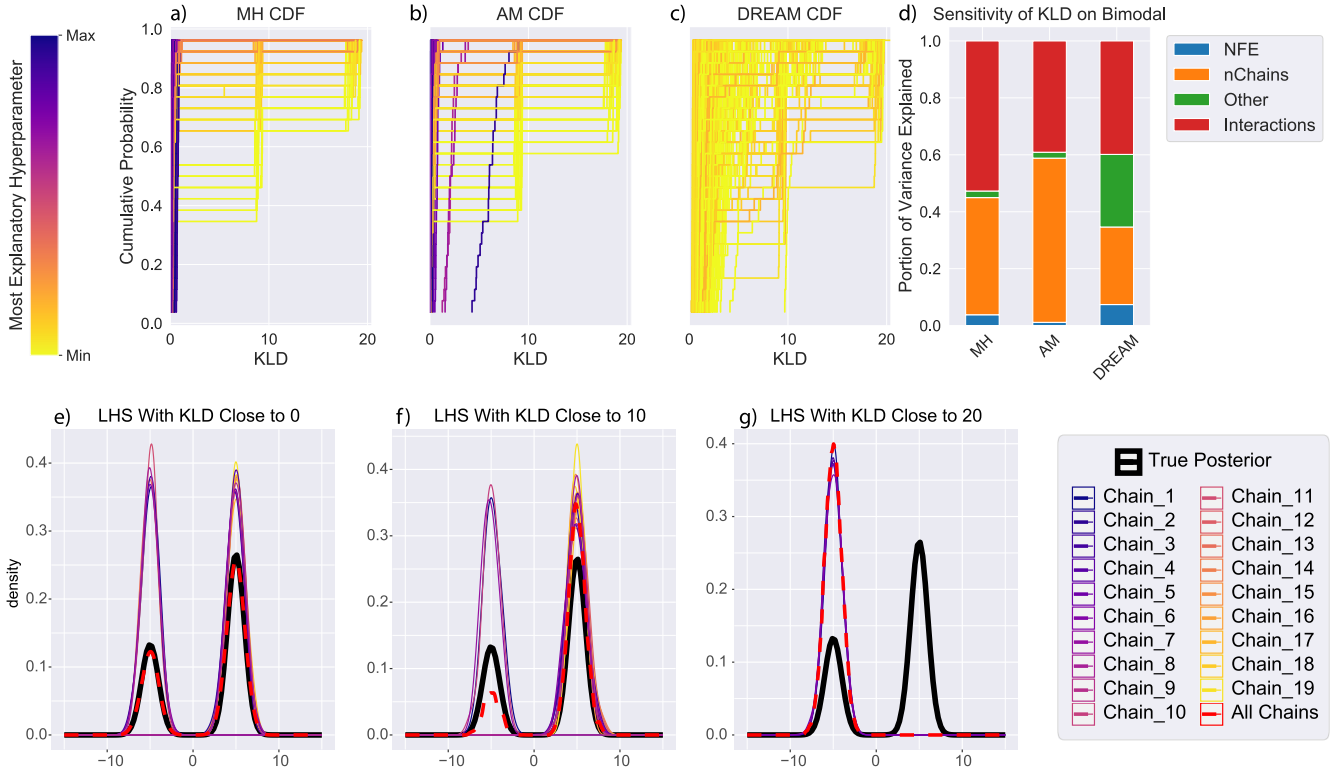

FIGURE 8. (a-c) CDFs of KLDs across random seeds for each hyperparameter. The color of the CDF indicates the value of the hyperparameter to which that algorithm's KLD is most sensitive. (d) Decomposition of how much variablity in KLD is explained by each hyperparameter and their interaction for each algorithm. (e-g) Comparison of the estimated marginal posterior of the first dimension of the 10D bimodal test problem from individual chains and across chains using select hyperparameter sets with (e) KLD near 0, (f) KLD near 10 and (g) KLD near 20.

metric of convergence on multi-modal problems, raising the question of how to best diagnose convergence on problems with unknown posteriors.

Understanding the disagreement between WD and KLD values requires more investigation. Figure 6c reveals that the selected LH sample with a high KLD and low WD only detects the more likely mode of the distribution. This results in a high KLD because the posterior probabilities diverge significantly in the less likely mode. However, the WD is fairly low because the cost of transporting some of the density in the more probable mode to the less probable mode is small. Individual chains from the LH sample with a high WD and low KLD (Figure 6f) find different modes, but in near opposite proportions to their true likelihood. This results in a high WD because it is much more costly to transport excess density from the less likely mode to the more likely mode. However, the divergence between the estimated and true posterior is less significant since both modes are found, just not in the right proportions.

Figures 6c and 6f reveal the importance of considering multiple metrics to assess algorithm performance, as both WD and KLD are capturing important elements of distribution closeness, while failing to capture others. Consequently, in Figure 7, we show control maps combining KLD and WD to see which algorithms perform best on both. These figures illustrate the number of seeds yielding KLDs$<$1 (Figures 7a-7c), WDs$<$120 (Figures 7d-7f), and both (Figures 7g-7i), with darker blue indicating a higher number of seeds. Consequently, these maps illustrate all four diagnostic metrics: reliability is indicated by the number of random seeds meeting thresholds of acceptable effectiveness; controllability is illustrated by a lack of noise in reliability, with its value a function primarily NFE or chains; and efficiency is indicated by increased reliability at lower NFE. Consistent with Figure 6, we see that MH and AM meet the KLD threshold across more hyperparameterizations than DREAM, particularly at low NFE, while DREAM meets the WD threshold more often. Combining these, we see that MH and AM are able to meet both thresholds more often for low NFE (in about 5-10 seeds for $<$50,000 NFE compared to $<$5 seeds for DREAM), while DREAM meets both thresholds more often for $>$50,000 NFE. Thus, for multimodal problems, it may be best to use MH or AM if computationally limited, and DREAM otherwise.

Finally, to hone further in on algorithmic reliability, we show CDFs of the KLD metric for each algorithm in Figure 8. We choose the KLD metric since it better captures divergence in probability estimates from the true posterior. We also show CDFs of WD and GR of the first dimension in SI Figures S9-S10. As suspected from the control maps, the

sensitivity analysis in Figure 8d illustrates that the KLD of all algorithms is primarily controlled by the number of chains (orange). Consequently, for each hyperparameter, we color the CDF of its WD across random seeds by its associated number of chains, with yellow being low (2 chains) and purple being high (20 chains).

For MH and AM, we can see that low KLDs occur for high chain counts (Figures 8a-8c), while the trend vs. number of chains is less pronounced for DREAM. This is likely due to DREAM's higher sensitivity to other hyperparameters. We also observe three distinct clusters of KLD values at approximately 0, 10, and 20, particularly for MH and AM. Plotting the marginal posteriors of the first dimension from LH samples of MH with KLDs near these values in Figures 8e-8g, we see that a KLD near 0 indicates that the algorithm successfully captures both modes in close to perfect proportions, while a value near 10 suggests most chains captured only the more likely mode, and a value near 20 signifies detection of solely the less likely mode. These distinct KLD values make their measure of performance more intuitive than the WD values (see SI Figure S11), indicating it may be a clearer, although less precise performance measure for multimodal problems.

VII. CONCLUSION AND STATUS OF THE SCIENCE

This study introduced novel diagnostics for comparing MCMC algorithms in terms of their effectiveness, efficiency, reliability, and controllability via control and attainment maps. This fills an important gap in the MCMC literature, as existing diagnostics solely focus on diagnosing the effectiveness and efficiency of an individual search process, not on diagnosing its consistency (i.e. reliability and controllability) across multiple search processes with different random seeds and hyperparameter configurations. The findings from these new diagnostics have the potential to reduce the time required for hyperparameter tuning. While the diagnostics themselves require a non-trivial computational experiment, they can be performed on computationally cheap test problems with known posteriors, as done here. Users can then leverage the findings from these diagnostics to choose the most efficient algorithm and corresponding hyperparameter configuration to calibrate a more computationally expensive real-world problem with similar characteristics to the test problems. Existing MCMC diagnostics can then be applied to the single calibration run of the real-world problem to assess convergence of that individual search process. As such, our new diagnostics fill a complementary role to existing diagnostics: our diagnostics can inform the choice of algorithm, while existing diagnostics can then assess convergence using that algorithm.

We illustrate how our diagnostics can reveal which algorithm is most effective, efficient, controllable and reliable by applying them to three widely used MCMC algorithms – MH, AM, and DREAM – on test problems characterized by high dimensionality and bimodality, attributes commonly found in physical systems. The diagnostics offered valuable insights into the performance of these algorithms on different types of problems, as well as on which performance metrics should be used to evaluate algorithms in different contexts. In the context of the high-dimensional (100D) MVN test problem, our analysis revealed a notable sensitivity of MH and AM to the binary optimization hyperparameter, ironically resulting in sub-optimal performance when using optimization to initialize chains. While, MH and AM without optimization exhibited improved convergence and closer alignment with the true posterior distribution, these algorithms needed significant NFE to do so, especially when using multiple chains. In contrast, DREAM consistently demonstrated strong performance, as evidenced by both control and attainment maps. For the 10D Bimodal Mixed-Gaussian test problem, DREAM continued to perform well, achieving lower WD and GR values compared to MH and AM. However, when considering the KLD metric, MH and AM displayed competitive performance, particularly in scenarios involving a smaller number of chains.

These conflicting findings across performance metrics on the bimodal problem analysis revealed intricate trade-offs between WD, GR, and KLD values, shedding light on their strengths and weaknesses in assessing algorithm performance. Critically, it was highlighted that low GR values do not necessarily indicate convergence, just consistent variance across chains. This is particularly uninformative if the chains are consistent only because they are not independent, but communicate as in DREAM. In reality, the chains may represent consistently poor approximations of the true posterior. Consequently, multiple metrics could be used to assess MCMC convergence on problems with unknown posteriors, and further research is needed on developing alternative convergence metrics for such problems. For algorithm development, test problems with known posteriors could be used for performance assessment to avoid these biases. When the true posterior is known, WD and KLD represent better measures of performance, but capture different aspects of that performance. KLD is a better measure of how close the estimated probabilities of different parameter values are to their true probabilities, while WD is a better measure of how close those two distributions are in parameter space. For multi-modal problems, KLD may then be more appropriate.

Finally, the analysis in this paper also points to new avenues of research. First, an important area of future research is in applying these diagnostics to assess not only efficiency in the mean estimate of the posterior, but in the variance of that estimate. As discussed in the introduction, because of the Markovian property of MCMC algorithms, consecutive elements in the chain are not independent. This autocorrelation can reduce the effective sample size of the chain, thereby increasing the Monte Carlo error, and corresponding standard error of the posterior mean estimate. Future work could investigate how this uncertainty changes across hyperparameter configurations and random seeds by making control and attainment maps of the standard error of the posterior mean estimate across chains. Such analysis

could also include thinning of the chain as a hyperparameter, whereby only every k elements in the chain are retained, to see what impact that hyperparameter has on the standard error.

Second, the finding that while DREAM was robust, it did exhibit greater sensitivity to its additional hyperparameters suggests DREAM's controllability could be improved by adapting its probability of using its different proposal operators based on their success in proposing new chain locations that are accepted. This idea comes from the observation in the literature that performance of multi-objective evolutionary algorithms can be improved by adapting the probability of using different operators based on their success in generating non-dominated solutions [63]. In testing such proposed advancements for MCMC, performance metrics such as KLD and WD could be used to evaluate performance on known test problems. The visual diagnostics proposed here can then be used to evaluate and inform the design and hyperparameterization of such new MCMC algorithms.

VIII. CODE AND DATA

We provide the scripts written to generate synthetic data and do the analysis in this study in our Zenodo repository.[1] Code development history may also be found on our GitHub repository.[2]

ACKNOWLEDGMENT

The authors would like to thank the Research Computing with the University of Virginia for providing computational resources and technical support that have contributed to the results reported within this publication (https://rc.virginia.edu). Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by U.S. government.

REFERENCES

[1] A. E. Lim, J. G. Shanthikumar, and Z. M. Shen, "Model uncertainty, robust optimization, and learning," in *Models, Methods, and Applications for Innovative Decision Making*. Catonsville, MD, USA: INFORMS, 2006, pp. 66–94.

[2] J. Mandur and H. Budman, "A polynomial-chaos based algorithm for robust optimization in the presence of Bayesian uncertainty," *IFAC Proc. Volumes*, vol. 45, no. 15, pp. 549–554, 2012.

[3] J. Mandur and H. Budman, "Robust optimization of chemical processes using Bayesian description of parametric uncertainty," *J. Process Control*, vol. 24, no. 2, pp. 422–430, Feb. 2014.

[4] T. Campbell and J. P. How, "Bayesian nonparametric set construction for robust optimization," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 4216–4221.

[5] C. Liang and S. Mahadevan, "Bayesian sensitivity analysis and uncertainty integration for robust optimization," *J. Aerosp. Inf. Syst.*, vol. 12, no. 1, pp. 189–203, Jan. 2015.

[6] C. Ning and F. You, "Data-driven adaptive nested robust optimization: General modeling framework and efficient computational algorithm for decision making under uncertainty," *AIChE J.*, vol. 63, no. 9, pp. 3790–3817, Sep. 2017.

[1] https://zenodo.org/records/10433119

[2] https://github.com/hosseinkavianih/New-Diagnostics-Assessment-For-MCMC

[7] Y. Xie, K. Feng, M. Du, Y. Wang, and L. Li, "Robust optimization of stamping process based on Bayesian estimation," *J. Manuf. Processes*, vol. 101, pp. 245–258, Sep. 2023.

[8] M. K. Cowles and B. P. Carlin, "Markov chain Monte Carlo convergence diagnostics: A comparative review," *J. Amer. Stat. Assoc.*, vol. 91, no. 434, p. 883, Jun. 1996.

[9] V. Roy, "Convergence diagnostics (don't short) for Markov chain Monte Carlo," *Annu. Rev. Statist. Appl.*, vol. 7, no. 1, pp. 387–412, Mar. 2020.

[10] J. P. Hobert and G. L. Jones, "Honest exploration of intractable probability distributions via Markov chain Monte Carlo," *Stat. Sci.*, vol. 16, no. 4, pp. 312–334, Nov. 2001.

[11] J. S. Rosenthal, "Minorization conditions and convergence rates for Markov chain Monte Carlo," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, p. 558, Jun. 1995.

[12] G. O. Roberts and R. L. Tweedie, "Bounds on regeneration times and convergence rates for Markov chains," *Stochastic Processes Their Appl.*, vol. 80, no. 2, pp. 211–229, 1999.

[13] A. Durmus and É. Moulines, "Quantitative bounds of convergence for geometrically ergodic Markov chain in the Wasserstein distance with application to the metropolis adjusted Langevin algorithm," *Statist. Comput.*, vol. 25, no. 1, pp. 5–19, Jan. 2015.

[14] C. J. Geyer, "Introduction to Markov chain Monte Carlo," in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. L. Jones, and X. L. Meng, Eds. Boca Raton, FL, USA: CRC Press, 2011, pp. 3–48.

[15] G. O. Roberts, "Convergence diagnostics of the Gibbs sampler," *Bayesian Statist.*, vol. 4, pp. 775–782, 1992.

[16] C. Ritter and M. A. Tanner, "Facilitating the Gibbs sampler: The Gibbs stopper and the Griddy–Gibbs sampler," *J. Amer. Stat. Assoc.*, vol. 87, no. 419, p. 861, Sep. 1992.

[17] P. Mykland, L. Tierney, and B. Yu, "Regeneration in Markov chain samplers," *J. Amer. Stat. Assoc.*, vol. 90, no. 429, p. 233, Mar. 1995.

[18] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Stat. Sci.*, vol. 7, no. 4, pp. 457–472, Nov. 1992.

[19] S. P. Brooks and A. Gelman, "General methods for monitoring convergence of iterative simulations," *J. Comput. Graph. Statist.*, vol. 7, no. 4, p. 434, Dec. 1998.

[20] C. P. Robert, G. Casella, C. P. Robert, and G. Casella, "The metropolis-hastings algorithm," *Monte Carlo Stat. methods*, pp. 267–320, 2004.

[21] D. Vats, J. M. Flegal, and G. L. Jones, "Multivariate output analysis for Markov chain Monte Carlo," *Biometrika*, vol. 106, no. 2, pp. 321–337, Jun. 2019.

[22] P. Heidelberger and P. D. Welch, "Simulation run length control in the presence of an initial transient," *Oper. Res.*, vol. 31, no. 6, pp. 1109–1144, Dec. 1983.

[23] B. Yu, "Estimating L1 error of kernel estimator: Monitoring convergence of Markov samplers," Cahier De Recherche, Dept. Statistics, Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep., 1995. [Online]. Available: https://digitalassets.lib.berkeley.edu/sdtr/ucb/text/409.pdf

[24] E. L. Boone, J. R. W. Merrick, and M. J. Krachey, "A Hellinger distance approach to MCMC diagnostics," *J. Stat. Comput. Simul.*, vol. 84, no. 4, pp. 833–849, Apr. 2014.

[25] A. Dixit and V. Roy, "MCMC diagnostics for higher dimensions using Kullback Leibler divergence," *J. Stat. Comput. Simul.*, vol. 87, no. 13, pp. 2622–2638, Sep. 2017.

[26] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*. Boca Raton, FL, USA: CRC Press, 2013.

[27] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization," *Evol. Comput.*, vol. 20, no. 3, pp. 423–452, Sep. 2012.

[28] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat, "Evolutionary multiobjective optimization in water resources: The past, present, and future," *Adv. Water Resour.*, vol. 51, pp. 438–456, Jan. 2013.

[29] F. Hartig, F. Minunno, and S. Paul. (2023). *BayesianTools: General-Purpose MCMC SMC Samplers Tools for Bayesian Statistics, 2023, R Package Version 0.1.8*. [Online]. Available: https://github.com/florianhartig/BayesianTools

[30] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[31] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, p. 97, Apr. 1970.

[32] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *Comput. J.*, vol. 14, no. 4, pp. 422–425, Apr. 1971.
[33] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.
[34] K. I. M. McKinnon, "Convergence of the Nelder-mead simplex method to a nonstationary point," *SIAM J. Optim.*, vol. 9, no. 1, pp. 148–158, Jan. 1998.
[35] H. Haario, E. Saksman, and J. Tamminen, "An adaptive metropolis algorithm," *Bernoulli*, vol. 7, no. 2, p. 223, Apr. 2001.
[36] G. O. Roberts and J. S. Rosenthal, "Examples of adaptive MCMC," *J. Comput. Graph. Statist.*, vol. 18, no. 2, pp. 349–367, Jan. 2009.
[37] H. Haario, M. Laine, A. Mira, and E. Saksman, "DRAM: Efficient adaptive MCMC," *Statist. Comput.*, vol. 16, no. 4, pp. 339–354, Dec. 2006.
[38] E. Laloy and J. A. Vrugt, "High-dimensional posterior exploration of hydrologic models using multiple-try DREAM(ZS) and high-performance computing," *Water Resour. Res.*, vol. 48, no. 1, Jan. 2012, Art. no. W01526.
[39] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, p. 341, 1997.
[40] J. A. Vrugt, C. J. F. ter Braak, C. G. H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon, "Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling," *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 10, no. 3, pp. 273–290, Jan. 2009.
[41] C. J. F. ter Braak and J. A. Vrugt, "Differential evolution Markov chain with snooker updater and fewer chains," *Statist. Comput.*, vol. 18, no. 4, pp. 435–446, Dec. 2008.
[42] A. Gelman, G. O. Roberts, and W. R. Gilks, "Efficient metropolis jumping rules," *Bayesian Statist.*, vol. 5, nos. 599–608, p. 42, 1996.
[43] G. O. Roberts and J. S. Rosenthal, "Optimal scaling for various metropolis-hastings algorithms," *Stat. Sci.*, vol. 16, no. 4, pp. 351–367, Nov. 2001.
[44] W. R. Gilks, G. O. Roberts, and E. I. George, "Adaptive direction sampling," *Statistician*, vol. 43, no. 1, p. 179, 1994.
[45] R. Nishihara, I. Murray, and R. P. Adams, "Parallel MCMC with generalized elliptical slice sampling," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2087–2112, 2014.
[46] E. Plischke, E. Borgonovo, and C. L. Smith, "Global sensitivity measures from given data," *Eur. J. Oper. Res.*, vol. 226, no. 3, pp. 536–550, May 2013.
[47] J. Herman and W. Usher, "SALib: An open-source Python library for sensitivity analysis," *J. Open Source Softw.*, vol. 2, no. 9, p. 97, Jan. 2017.
[48] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
[49] R. L. Dobrushin, "Prescribing a system of random variables by conditional distributions," *Theory Probab. Appl.*, vol. 15, no. 3, pp. 458–486, Jan. 1970.
[50] H. Jeffreys, *Theory of Probability*, 2nd ed. Oxford, U.K.: Clarendon, 1948.
[51] S. Boltz, E. Debreuve, and M. Barlaud, "KNN-based high-dimensional Kullback–Leibler distance for tracking," in *Proc. 8th Int. Workshop Image Anal. Multimedia Interact. Services (WIAMIS)*, Jun. 2007, p. 16.
[52] S. Boltz, É. Debreuve, and M. Barlaud, "High-dimensional statistical measure for region-of-interest tracking," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1266–1283, Jun. 2009.
[53] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Manage. Sci.*, vol. 6, no. 4, pp. 366–422, Jul. 1960.
[54] S. S. Vallender, "Calculation of the Wasserstein distance between probability distributions on the line," *Theory Probab. Appl.*, vol. 18, no. 4, pp. 784–786, Sep. 1974.
[55] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–8.
[56] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. Mach. Learn. Res.*, vol. 11, pp. 2487–2531, 2010.
[57] A. J. Cannon, "Multivariate quantile mapping bias correction: An N-dimensional probability density function transform for climate model simulations of multiple variables," *Climate Dyn.*, vol. 50, nos. 1–2, pp. 31–49, Jan. 2018.
[58] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance," 2016, *arXiv:1602.06561*.
[59] J.-H. Choi and J.-S. Lee, "EmbraceNet: A robust deep learning architecture for multimodal classification," *Inf. Fusion*, vol. 51, pp. 259–270, Nov. 2019.
[60] D. Wang, J. Su, and H. Yu, "Feature extraction and analysis of natural language processing for deep learning English language," *IEEE Access*, vol. 8, pp. 46335–46345, 2020.
[61] M. E. Mann and J. Park, "Global-scale modes of surface temperature variability on interannual to century timescales," *J. Geophys. Res., Atmos.*, vol. 99, no. D12, pp. 25819–25833, 1994.
[62] D. E. Giles, "Hermite regression analysis of multi-modal count data," *Econ. Bull.*, vol. 30, no. 4, pp. 2936–2945, 2010.
[63] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evol. Comput.*, vol. 21, no. 2, pp. 231–259, May 2013.

HOSSEIN KAVIANIHAMEDANI received the B.S. and M.S. degrees in civil engineering from the Sharif University of Technology, Tehran, Iran, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in systems and information engineering with the University of Virginia, Charlottesville, VA, USA. His research interests include Bayesian inference, machine learning, and robust optimization.

JULIANNE D. QUINN received the B.S. degree in earth and environmental engineering from Columbia University, New York City, NY, USA, in 2011, and the Ph.D. degree in civil and environmental engineering from Cornell University, Ithaca, NY, USA, in 2017. She was a Postdoctoral Associate with Cornell University, from 2017 to 2018. Since 2018, she has been an Assistant Professor with the Department of Civil and Environmental Engineering and the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. Her research interests include multi-objective control, uncertainty analysis, and risk analysis applied to environmental systems. She is a member of American Geophysical Union, European Geosciences Union, American Society of Civil Engineers, Institute for Operations Research and Management Sciences, and Society for Decision Making under Deep Uncertainty.

JARED D. SMITH received the B.S. degree in environmental engineering from Clarkson University, Potsdam, NY, USA, in 2013, and the M.S. and Ph.D. degrees in environmental and water resources systems engineering from Cornell University, Ithaca, NY, USA, in 2016 and 2019, respectively. He was a Postdoctoral Research Associate with the University of Virginia, Charlottesville, VA, USA, from 2019 to 2021. He was with U.S. Geological Survey, Reston, VA, USA, as a Machine Learning Specialist, from 2021 to 2023. He is currently a Physical Scientist. His research interests include predictive modeling, uncertainty propagation, and resource assessment applied to environmental systems. He is a member of American Geophysical Union.