BLACK-BOX UNADJUSTED HAMILTONIAN MONTE CARLO

Anonymous authors
Paper under double-blind review

ABSTRACT

Hamiltonian Monte Carlo and underdamped Langevin Monte Carlo are state-of-the-art methods for taking samples from high-dimensional distributions with a differentiable density function. To generate samples, they numerically integrate Hamiltonian or Langevin dynamics. This numerical integration introduces an asymptotic bias in Monte Carlo estimators of expectation values, which can be eliminated by adjusting the dynamics with a Metropolis-Hastings (MH) proposal step. Alternatively, one can trade bias for variance by avoiding MH, and select an integration step size that ensures sufficiently small asymptotic bias, relative to the variance inherent in a finite set of samples. Such *unadjusted* methods often significantly outperform their adjusted counterparts in high-dimensional problems where sampling would otherwise be prohibitively expensive, yet are rarely used in statistical applications due to the absence of an automated way of choosing a step size. We propose just such an automatic tuning scheme that takes a user-provided asymptotic bias tolerance and selects a step size that ensures it. The key to the method is a relationship we establish between the energy change during integration and the asymptotic bias. We show that this procedure rigorously bounds the asymptotic bias for Gaussian target distributions. We then numerically show that the procedure works beyond Gaussians. To demonstrate the practicality of the proposed scheme, we provide a comprehensive comparison of adjusted and unadjusted samplers on Bayesian inference problems and on a statistical physics model in more than one million parameters. With our tuning scheme, the unadjusted methods achieve close to optimal performance, significantly and consistently outperforming their adjusted counterparts.

1 INTRODUCTION

Sampling offers a way to compute expectation values $\mathbb{E}_p[f] = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$, where $f(\boldsymbol{x})$ is some function of parameters $\boldsymbol{x} \in \mathbb{R}^d$ and $p(\boldsymbol{x}) = e^{-\mathcal{L}(\boldsymbol{x})}/Z$ is a given probability density, often with $Z = \int e^{-\mathcal{L}(\boldsymbol{x})} dx$ unknown. This is a key tool in many disciplines, from social science (Gelman et al., 2013), to high energy physics (Duane et al., 1987), computational chemistry Leimkuhler & Matthews (2015), statistical physics (Leimkuhler & Matthews, 2015), and machine learning (Neal, 2012). Often a gradient $\nabla \mathcal{L}(\boldsymbol{x})$ is available, either analytically, or via automatic differentiation (Griewank & Walther, 2008; Margossian, 2019); examples appear in Bayesian statistics (Štrumbelj et al., 2024; Carpenter et al., 2017), machine learning (Baydin et al., 2018), lattice quantum problems (Gattringer & Lang, 2010), and cosmology (Campagne et al., 2023; Horowitz & Lukic, 2025).

Markov Chain Monte Carlo (MCMC; (Metropolis et al., 1953)) is a commonly employed class of sampling methods in which a Markov chain $\{\boldsymbol{x}_i\}_{i=1}^n$ is designed to have a stationary distribution $p(\boldsymbol{x})$ (or close to p), so that the expectation value $\mathbb{E}_p[f]$ can be approximated by $\bar{f} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$. When a (smooth) gradient is available, Hamiltonian Monte Carlo (HMC; Duane et al. (1987); Neal (2011); Betancourt (2018)) and underdamped Langevin Monte Carlo (LMC; Horowitz (1991); Leimkuhler & Matthews (2015)) are state-of-the-art algorithms. Despite their success (Štrumbelj et al., 2024), MCMC often presents a computational bottleneck (Gattringer & Lang, 2010; Leimkuhler & Matthews, 2015; Simon-Onfroy et al., 2025), especially in the high-dimensional applications, forcing practitioners to resort to more approximate methods, such as the Laplace approximation (Millea & Seljak, 2022) or an ensemble Kalman filter (Houtekamer & Mitchell, 2005).

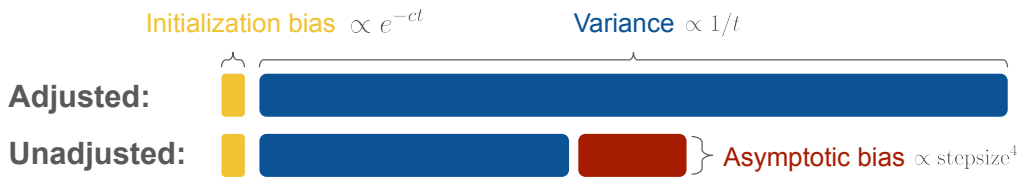

Figure 1: Graphical representation of the bias decomposition from Equation (2). Both adjusted and unadjusted methods have initialization bias, which typically decays exponentially fast for HMC-like algorithms (Margossian & Gelman, 2024) (with a rate denoted by c in the Figure). Both methods also have the variance associated with the finite number of samples taken. It decays inversely proportionally to the number of samples t, with a proportionality constant that is determined by the autocorrelation time. Unadjusted methods additionally have asymptotic bias, which depends strongly on the step size ϵ; as $\mathcal{O}(\epsilon^4)$, for the example from Appendix C.

Bias-variance tradeoff To understand the challenge faced by MCMC, consider the root mean squared error (RMSE) of the Monte Carlo estimator:

$$\text{RMSE}[\bar{f}] = \mathbb{E}_{MC}[(\bar{f} - \mathbb{E}_p[f])^2]^{1/2}, \tag{1}$$

where $\mathbb{E}_{MC}[\cdot]$ denotes the expectation with respect to the Monte Carlo randomness. The RMSE can be decomposed as

$$\text{RMSE}[\bar{f}]^2 = \text{Bias}[\bar{f}]^2 + \text{Var}[\bar{f}], \tag{2}$$

where $\text{Bias}[\bar{f}] = \mathbb{E}_{MC}[\bar{f} - \mathbb{E}_p[f]]$ and $\text{Var}[\bar{f}] = \mathbb{E}_{MC}[(\bar{f} - \mathbb{E}_{MC}[\bar{f}])^2]$. The bias term arises either from the chain not yet reaching its stationary state and being influenced by its initial position (initialization bias), or because the stationary distribution of the chain $\tilde{p}(\boldsymbol{x})$ does not equal the target distribution $p(\boldsymbol{x})$ (asymptotic bias). Initialization bias is an important issue (Margossian et al., 2024), but we will not study it here, noting that for long enough chains it can be eliminated by discarding initial samples (Margossian & Gelman, 2024). The variance term comes from the finite chain length and the correlation between the samples. This decomposition is illustrated in Figure 1.

One of the main practical goals in designing MCMC methods is to obtain a desired RMSE at the lowest possible computational budget. This amounts to balancing the cost of generating samples, correlations between the samples, and the asymptotic bias. For example, the use of Metropolis-Hastings adjustment (Chib & Greenberg, 1995) ensures that a chain satisfies detailed balance, so that the asymptotic bias vanishes, but for finite-length chains, it does not remove the variance. Noting that chains are finite in practice, one could negotiate the tradeoff differently and reduce the variance at the cost of introducing some bias. This strategy is the focus of the present work.

Illustrative example The shortcoming of performing MH is the scaling with the dimension that it implies for the sampler. To develop an intuition, we consider a simple problem, where the target is a product of K independent D-dimensional distributions q:

$$p(\boldsymbol{x}) = \prod_{i=1}^{K} q(x_{iD}, x_{iD+1} \dots x_{iD+D-1}),$$

where D is a small number. Suppose we are interested in the expectation value of a function of only one of those parameters, such as $f(\boldsymbol{x}) = x_1^2$. We will measure the performance as the number of evaluations of $\nabla p(\boldsymbol{x})$ (which is typically the bottleneck of computation and thus a proxy for the wall-clock time) that the sampler uses to get the relative RMSE, i.e. $\text{RMSE}[\bar{f}]/\mathbb{E}_p[f]$ (averaged over 128 chains) below 10%.

Figure 2 shows performance as a function of the dimension of the problem $d = KD$. As can be seen, the number of gradient calls scales as $d^{1/4}$ for Metropolis adjusted methods, in accordance with (Beskos et al., 2013). This is because even though the Hamiltonian and Langevin dynamics operate independently on each copy of the distribution, the energy change Δ_H (defined below), and therefore the MH acceptance probability, involves a sum over all parameters. Optimal performance requires a fixed acceptance rate (Beskos et al., 2013; Neal, 2011), so to compensate for the increasing number of

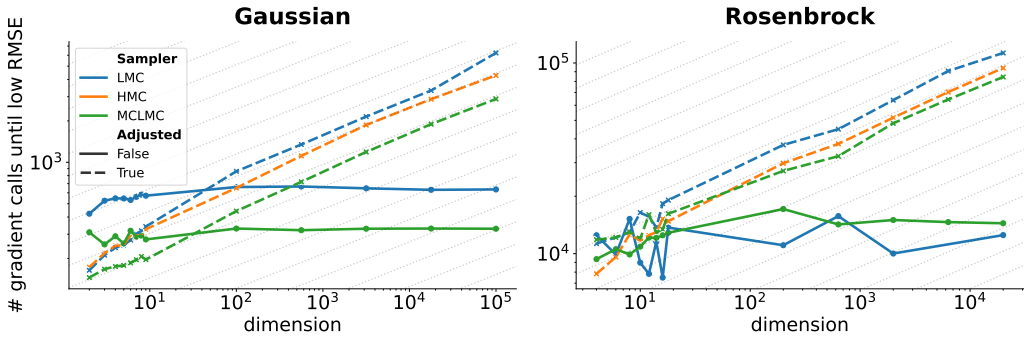

Figure 2: Sampling cost scaling with the dimensionality for product targets. Cost is measured by the number of gradient calls needed to achieve low error. MH adjusted methods are shown in dashed lines, and unadjusted methods are shown in solid lines. Step size for the unadjusted schemes is selected by the scheme proposed in this paper. $d^{1/4}$ power law grey lines are shown in the background. A Standard Gaussian target is shown on the left ($D = 1$), and a product of banana-shaped Rosenbrock distributions ($D = 2$) on the right. In both cases, the cost of the MH adjusted methods scales as $d^{1/4}$, while the cost of the unadjusted method remains constant with dimension.

parameters, the step size needs to decrease, so that the integrator uses more gradient evaluations for a fixed trajectory length. Unadjusted methods, by contrast, do not suffer from this scaling since they operate on each copy independently. They need to ensure that the bias is small, but this is independent of the number of parameters in these examples.

The example in Figure 2 is idealized, but a similar scaling has been observed for mean field models (Durmus & Eberle, 2023) and real problems, like cosmological field level inference (Simon-Onfroy et al., 2025). Thus, particularly for high-dimensional problems, unadjusted samplers may be significantly more efficient. Crucially, however, they are only viable if the step size can be chosen so that the asymptotic bias is small compared to the variance of the finite set of samples taken.

Our contributions We propose an automated step size adaptation scheme for unadjusted HMC and LMC, making them usable in a black-box manner, i.e. *without manually tuning any hyperparameters*. We expect this to have a significant practical impact on applications that require gradient-based sampling in high dimensions. The central idea is that the energy error that arises from integrating with step size ϵ provides a measure of the asymptotic bias, and step size can be adaptively varied in a short initial chain to target an energy error which results in low asymptotic bias relative to the variance. In Section 4 we show analytically for Gaussians that the energy error variance can be used to upper bound the bias. In Section 5, we numerically confirm the analytical results and show that the same upper bound generally extends to non-Gaussian targets. In Section 6 we use the bound to construct our adaptation algorithm. In Section7 we demonstrate its effectiveness on a range of standard benchmarks and on a real-world lattice field theory problem in more than 1 million dimensions.

2 Related work

HMC is a state-of-the-art Markov kernel for densities with smooth gradients, as is *underdamped* Langevin Monte Carlo (LMC). The latter should be distinguished from *overdamped* Langevin dynamics and the corresponding Metropolis Adjusted Langevin Algorithm (MALA; Rossky et al. (1978)). In HMC and LMC, each parameter x_i has an associated momentum variable u_i. The state of the Markov chain $\boldsymbol{x}_n$ is used as an initial condition $\boldsymbol{x}(t = 0) = \boldsymbol{x}_n$ of the dynamics, along with random initial values for the momenta $u_i(0) \sim \mathcal{N}(0, 1)$. The next state in HMC is then generated by solving for $\boldsymbol{x}(t = T)$, where T is a predefined trajectory length and $\boldsymbol{z}(t) \equiv (\boldsymbol{x}(t), \boldsymbol{u}(t))$ are solutions of the Hamiltonian equations with Hamiltonian function $\mathcal{H}(\boldsymbol{z}) = \frac{1}{2} \|\boldsymbol{u}\|^2 + \mathcal{L}(\boldsymbol{x})$:

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{u}(t) \qquad \frac{d}{dt}\boldsymbol{u}(t) = -\nabla\mathcal{L}(\boldsymbol{x}). \tag{3}$$

Note that the value of the Hamiltonian function, also called the energy, is a conserved quantity of the Hamiltonian equations, meaning that the energy difference $\Delta_H(\boldsymbol{z}', \boldsymbol{z}) = H(\boldsymbol{z}') - H(\boldsymbol{z})$ vanishes for exact solutions of the Hamiltonian dynamics: $\Delta_H(\boldsymbol{z}(t), \boldsymbol{z}(0)) = 0$. In LMC, after every step with Hamiltonian dynamics, the velocity is partially randomized, which corresponds to the discretization of the Langevin stochastic differential equation (Leimkuhler & Matthews, 2015).

In statistics, MH adjusted versions of these dynamics are used almost exclusively. However, unadjusted versions are used in some fields with high-dimensional distributions, such as Lattice quantum chromodynamics (Lüscher, 2018; Clark & Kennedy, 2007) and Molecular Dynamics (Leimkuhler & Matthews, 2015). Commonly, underdamped Langevin dynamics or Nosé-Hoover thermostat (Evans & Holian, 1985) are employed. Another promising option is Microcanonical Langevin Monte Carlo (MCLMC; Robnik et al. (2024); Robnik & Seljak (2024); Minary et al. (2003); Steeg & Galstyan (2021)), which makes use of velocity norm preserving dynamics. Domain knowledge and trial runs are used in these fields to select an appropriate step size, which ensures a sufficiently small asymptotic bias. Unadjusted methods have also been analyzed theoretically, establishing a bound on the asymptotic bias (Durmus & Eberle, 2023) and the mixing time (Bou-Rabee & Eberle, 2023; Camrud et al., 2024). For mean-field models, the bound on both is dimension-free (Bou-Rabee & Schuh, 2023) and thus unadjusted methods provably outperform adjusted methods. However, these bounds assume some global knowledge of the distribution that is not available in practice. A notable gap in the above work is an algorithm for choosing the step size ϵ. Without a principled way to do so, general practitioners cannot use unadjusted algorithms in a black-box fashion, which gravely limits their applicability.

3 Measuring asymptotic bias

Origin of bias Generically, Hamiltonian equations like (3) cannot be solved exactly, so a numerical integrator like velocity Verlet (Leimkuhler & Matthews, 2015) is used to approximate it. Velocity Verlet is an example of a splitting method, where to solve the dynamics numerically, one first analytically solves for $\boldsymbol{x}$ at fixed $\boldsymbol{u}$ and vice versa. The first solution is called the position update and is given by $\Phi_\epsilon^T(\boldsymbol{z}) = (\boldsymbol{x} + \epsilon \boldsymbol{u}, \boldsymbol{u})$ for Equation (3), while the second is called the velocity update and is given by $\Phi_\epsilon^V(\boldsymbol{z}) = (\boldsymbol{x}, \boldsymbol{u} - \epsilon \nabla \mathcal{L}(\boldsymbol{x}))$. The joint solution is then approximated by a composition of these maps, which for velocity Verlet is,

$$\boldsymbol{z}(t+\epsilon) \approx \Phi_\epsilon(\boldsymbol{z}(t)) = (\Phi_{\epsilon/2}^V \circ \Phi_\epsilon^T \circ \Phi_{\epsilon/2}^V)(\boldsymbol{z}(t)). \tag{4}$$

Due to this approximation, the stationary distribution $\tilde{p}(\boldsymbol{x})$ of the sampler no longer equals the desired target distribution $p(\boldsymbol{x})$ and expectation values acquire an asymptotic bias, which vanishes in the limit of step size going to zero (Durmus & Eberle, 2023). In the case of LMC, the update is additionally complemented by partial refreshments of the velocity (Leimkuhler & Matthews, 2015):

$$\boldsymbol{z}(t+\epsilon) \approx (\Phi_{\epsilon/2}^O \Phi_\epsilon \circ \Phi_{\epsilon/2}^O)(\boldsymbol{z}(t)), \tag{5}$$

where $\Phi_\epsilon^O(\boldsymbol{z}) = (\boldsymbol{x},\, e^{-\epsilon/L}\boldsymbol{u} + (\sqrt{1 - e^{-2\epsilon/L}})\mathbf{n})$ and $\mathbf{n} \sim \mathcal{N}(0, I)$. The parameter L determines the amount of momentum refreshment and plays a similar role as the trajectory length in HMC.

Summary statistics When minimizing bias, it is important to determine the expectation with respect to which the bias is defined. For instance, an important expectation in Bayesian statistics (where quantification of uncertainty is key) is of the second moments, so that we are concerned with expectations of the form $\mathbb{E}_p[x_i x_j]$, or more generally the covariance matrix $\Sigma_p = \mathbb{E}_q[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^T]$. For Gaussian distributions, the covariance matrix contains all the information about the posterior, but even for non-Gaussian distributions, they are often used as a summary statistics.

From a theoretical perspective, it may also be interesting to consider the bias of a wider set of functions, such as any Lipschitz-continuous functions. In what follows, we consider a metric that controls the more general bias and a metric measuring the covariance matrix bias.

Covariance matrix error To quantify expectation value error of the second moments, we introduce a scalar measure of the covariance matrix error, which we define as

$$b_{cov}^2(\Sigma_p, \Sigma_q) \equiv \frac{1}{d} \operatorname{Tr}\big\{(I - \Sigma_p^{-1}\Sigma_q)^2\big\}, \tag{6}$$

where Σ_p is the true covariance matrix of the target distribution p and Σ_q is the covariance matrix of some other distribution q. In the simple case where the covariance matrices are diagonal, b_{cov} is the relative error of the variance estimate, averaged over the parameters, i.e., $b_{cov}^2(\Sigma_p, \Sigma_q) = \frac{1}{d}\sum_{i=1}^{d}([\Sigma_p]_{ii} - [\Sigma_q]_{ii})^2/[\Sigma_p]_{ii}^2$. This convergence metric is often used in practice (Grumitt et al., 2022; Robnik et al., 2024). The diagonal form is preferable in high dimensions as it does not require storage of the full covariance matrix. Nonetheless, we will measure error with (6) when feasible because it additionally penalizes the off-diagonal terms and has a number of nice properties: it is a divergence on the space of positive-definite matrices (meaning that it is non-negative, and zero if and only if the two matrices are the same), can be connected to the effecive sample size and is invariant to the linear change of basis of the parameter space. Proofs of these properties are provided in Appendix A.

Wasserstein distance The Wasserstein distance $\mathcal{W}_\nu(p, q)$ between densities p and q is (Kantorovich, 1960):

$$\mathcal{W}_\nu(p, q) = \left(\inf_{\pi \in \Pi(p,q)} \int \|\boldsymbol{x} - \boldsymbol{x}'\|^\nu \, \pi(\boldsymbol{x}, \boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}'\right)^{1/\nu}, \tag{7}$$

where $\Pi(p, q)$ is the set of probability densities on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals p and q. Wasserstein distance has several nice properties: it is invariant to change of basis, is a metric on the space of distributions, and most importantly in the present context, it upper bounds the bias of Lipschitz-continuous functions. That is, for any Lipschitz-continuous function f, with Lipschitz constant L (meaning that $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| < L\,\|\boldsymbol{x} - \boldsymbol{x}'\|$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$), the bias associated with this function is upper bounded by $\mathcal{W}_1(p, \tilde{p})$, which is in turn upper bounded by $\mathcal{W}_2(p, \tilde{p})$:

$$\mathbb{E}_p[f] - \mathbb{E}_{\tilde{p}}[f] \leq L\mathcal{W}_1(p, \tilde{p}) \leq L\mathcal{W}_2(p, \tilde{p}). \tag{8}$$

The first inequality is Kantorovich-Rubinstein duality (Villani, 2003), and the second is a consequence of Jensen's inequality. We will focus on $\mathcal{W}_2$.

Energy error variance per dimension Our goal is to control some measure of the bias. We will focus on b_{cov}^2, because it is of practical interest in various fields like Bayesian inference and on $\mathcal{W}_2$, because it provides a bound on the bias of all Lipschitz continuous functions. We will do this by monitoring the energy error $\Delta_H(\Phi_\epsilon(\boldsymbol{z}(t)), \boldsymbol{z}(t))$ induced by numerical integration with step size ϵ. Computing the energy error is a side product of the integration step, for example for HMC, the position update energy change is $\mathcal{L}(\boldsymbol{x} + \epsilon\boldsymbol{u}) - \mathcal{L}(\boldsymbol{x})$ and $\frac{1}{2}\|\boldsymbol{u} - \epsilon\nabla\mathcal{L}(\boldsymbol{x})\|^2 - \frac{1}{2}\|\boldsymbol{u}\|^2$ for the velocity update. For all models of practical interest this incurs a negligible cost compared to the gradient evaluation, because $\mathcal{L}(\boldsymbol{x})$ is a side product of the gradient evaluation.

We define the Energy Error Variance Per Dimension (EEVPD),

$$\text{EEVPD} = \text{Var}_{\boldsymbol{x}\sim\tilde{p}, \boldsymbol{u}\sim N(0,I)}[\Delta_H(\Phi_\epsilon(\boldsymbol{x}, \boldsymbol{u}), (\boldsymbol{x}, \boldsymbol{u}))]/d. \tag{9}$$

Crucially, this is a quantity that can easily be estimated in practice: $\boldsymbol{x} \sim \tilde{p}$, $\boldsymbol{u} \sim \mathcal{N}(0, I)$ is the stationary distribution of the chain, so computing EEVPD amounts to collecting the samples from the stationary chain, evaluating the one-step energy error for each of those samples and computing their variance. This can be done online, using a running average of the first and second moment, so that the step size ϵ can be adaptively varied to target a desired value of EEVPD.

4 Analytic results for Gaussian distributions

We begin by showing that EEVPD can be used to control the asymptotic covariance matrix bias $b_{cov}^2(\Sigma, \tilde{\Sigma})$ and Wasserstein distance $\mathcal{W}_2(p, \tilde{p})$ for Gaussian target distributions, $p = \mathcal{N}(0, \Sigma)$. The key tool is that the stationary distribution $\tilde{p}$ of unadjusted HMC and LMC for Gaussians is known exactly (Gouraud et al., 2025) and is also Gaussian, with $\tilde{p} = \mathcal{N}(0, \tilde{\Sigma})$. $\tilde{\Sigma}$ has the same eigenvectors as Σ, but its eigenvalue associated to the i-th eigenvector is

$$\tilde{\sigma}_i^2 = \frac{\sigma_i^2}{1 - \epsilon^2/4\sigma_i^2}, \tag{10}$$

where σ_i^2 is the corresponding eigenvalue of Σ. Therefore the EEVPD has a closed form:

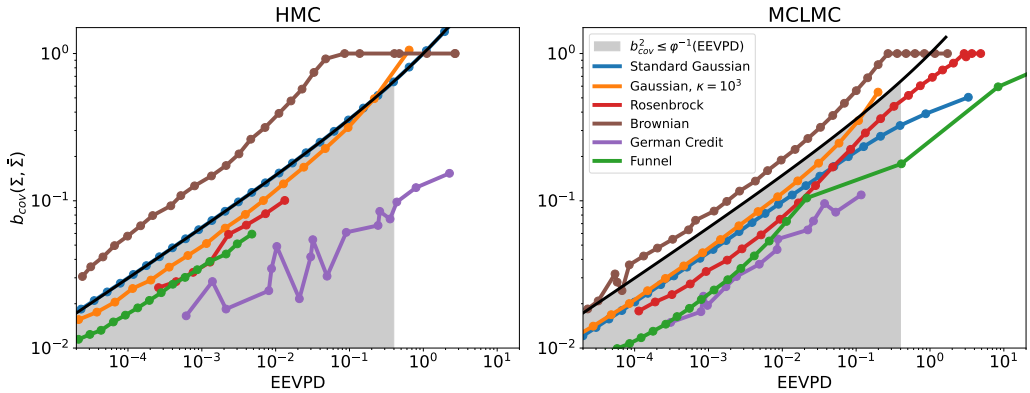

Figure 3: The asymptotic covariance matrix error $b_{cov}(\Sigma, \tilde{\Sigma})$ as a function of EEVPD. Unadjusted HMC (left), and unadjusted MCLMC (right) are shown. The relation is shown for various problems from Section 5. The analytical equality for the Standard Gaussians from Theorem 4.2 is shown in black and agrees perfectly with the numerical results for HMC (left). The inequality for arbitrary Gaussian distributions is shown as a shaded grey region, and also applies perfectly for Gaussians up to EEVPD = 0.397, as per Theorem 4.2. We see that most targets abide by this inequality.

Lemma 4.1. *For a Gaussian distribution with covariance matrix eigenvalues $\{\sigma_i^2\}_{i=1}^d$ and an HMC or LMC sampler with a stable velocity Verlet integrator, meaning that step size $\epsilon < 2\min_i \sigma_i$,*

$$\text{EEVPD} = \frac{1}{d}\sum_{i=1}^{d} E(\epsilon^2/\sigma_i^2),$$

where $E(y) = \frac{y^3}{16(1-y/4)}$.

The key result of this section is then:

Theorem 4.2. *For a Gaussian distribution $\mathcal{N}(0, \Sigma)$ with covariance matrix eigenvalues $\{\sigma_i^2\}_{i=1}^d$ and HMC or LMC sampler with a stable velocity Verlet integrator, meaning that the step size $\epsilon < 2\min_i \sigma_i$, the covariance matrix bias is upper bounded by*

$$b_{cov}^2(\Sigma, \tilde{\Sigma}) \leq \varphi^{-1}(\text{EEVPD}),$$

as long as EEVPD < 0.397. *Here,*

$$\varphi(x) = \frac{4x^{3/2}}{(1 + x^{1/2})^2}.$$

Similarly, the Wasserstein distance between the target and stationary distributions is upper bounded by

$$\mathcal{W}_2(p, \tilde{p})^2/d \leq \epsilon^2 \varphi_W^{-1}(\text{EEVPD}),$$

as long as EEVPD < 6.75. *Here, $\varphi_W \equiv E \circ W^{-1}$ with $W(y) = \frac{2(1-y/8-\sqrt{1-y/4})}{y(1-y/4)}$. Both bounds are sharp and realized if and only if the target is isotropic, i.e. $\Sigma_p \propto I$.*

All the proofs are given in Appendix B. Note that the conditions on EEVPD are not a severe limitation in practice, see for example Table 2, where significantly lower values of EEVPD are used.

5 Numeric results for non-Gaussian distributions

We now examine the validity of Theorem 4.2 for standard Bayesian inference benchmark problems. We focus on verifying the bound on b_{cov}, due to the difficulty of numerically computing the Wasserstein distance for the problems considered here. Even though the exact asymptotic error-EEVPD

relation of Theorem 4.2 applies only for HMC and LMC, unadjusted MCLMC has a notion of energy (Robnik et al., 2024), so we will also test MCLMC here. Benchmark problem descriptions are in Appendix D.1. For each problem, we show the asymptotic value of $b_{cov}(\Sigma, \tilde{\Sigma})^2$ as a function of EEVPD in Figure 3. Asymptotic b_{cov} is computed by running unadjusted chains with different step sizes, each using 10^8 gradient calls. We eliminate the initial 10^4 calls to eliminate the initialization bias and use the subsequent samples to compute the expectation values for the covariance matrix and EEVPD. We monitor $b_{cov}(\Sigma, \tilde{\Sigma})^2$ from Equation (6) and check that it has converged to the asymptotic value. If the convergence has not yet been achieved, i.e., the bias is still decaying, we do not show these measurements on the plots. This happens for some of the harder problems at small step sizes, where the chains are not long enough for the variance to become negligible. We have checked that the variation between the chains is negligible, but nonetheless average the results over 4 independent chains.

Numerical results for unadjusted HMC on Gaussians agree perfectly with Theorem 4.2: for the standard Gaussian, the equality holds, while for the Ill-conditioned Gaussian, the inequality holds for $\mathrm{EEVPD} < 0.397$ as per the theorem. The inequality also applies to the majority of non-Gaussian benchmark problems, illustrating its broader applicability. One exception is the Brownian motion example, where it is off by approximately a factor of 1.5 at small ϵ, meaning that one would think one has $< 2\%$ asymptotic error, when in fact it was 3%. The Rosenbrock and Funnel examples are only shown at small step sizes, because the problem becomes numerically unstable at higher step sizes, incurring divergences. Very similar results also apply to MCLMC, except that the bias at a fixed EEVPD is usually lower for MCLMC. This suggests that the tuning scheme we develop for unadjusted HMC can also be applied to MCLMC, but a slightly larger EEVPD may be used.

6 AUTOMATIC TUNING SCHEME

Stepsize The core findings of sections 4 and 5 are that by controlling EEVPD, we can in turn control the asymptotic bias of expectations of interest. Our tuning scheme for step size is therefore straightforward: for any unadjusted sampler with an appropriate notion of energy, we keep a running estimate of EEVPD, and adaptively vary ϵ to target a desired value of EEVPD in a stochastic optimization scheme. In practice, an optimization algorithm such as dual averaging (Nesterov, 2009; Hoffman & Gelman, 2014) can be used here, but the choice of optimizer has little effect on the results; we use the scheme in Appendix D.4. We note that a running average estimate of EEVPD converges significantly faster than say, a running average of the second moment. This is because the energy changes in the subsequent steps are very mildly correlated. For example, integrated autocorrelation time for EEVPD time series is around 2 for standard Gaussian problem in d = 100 and around 5 for the Brownian motion problem from Appendix D. Therefore, the variance of EEVPD estimate is small, even for a relatively small number of samples. We observe that for the experiments considered in this work, the EEVPD converges to the desired value in an order of a few tens of steps.

It remains to select the desired bias tolerance and the corresponding desired EEVPD. This depends on the application and the accuracy requirements; we, nonetheless, provide some guidance. Certainly, asymptotic bias squared should be smaller than the mean squared error tolerance (because mean squared error is composed of the bias and the variance, which are both non-negative), but by how much? Appendix C shows that for estimating $\mathbb{E}[x^2]$ of a standard normal distribution with finite chain length of unadjusted HMC, the optimal squared bias should be one fifth of the mean squared error tolerance. This suggests that EEVPD of 3×10^{-4} should be used if 10% relative RMSE is required and 3×10^{-7} if relative RMSE 1% is required. For convenience, a conversion table 2 is provided in appendix.

Momentum decoherence scale A final hyperparameter of unadjusted HMC is the number of steps between momentum refreshes, and for unadjusted LMC, the amount of noise in each partial refresh. Both can be understood in terms of a single hyperparameter, the *momentum decoherence length* L. For HMC, L is simply the number of steps in a trajectory times the step size, since this is precisely the length along the trajectory after which momentum completely decoheres. For LMC, L is given in Section 3. In either case, we select L based on the autocorrelation length of the the chain, as in (Robnik et al., 2024; 2025).

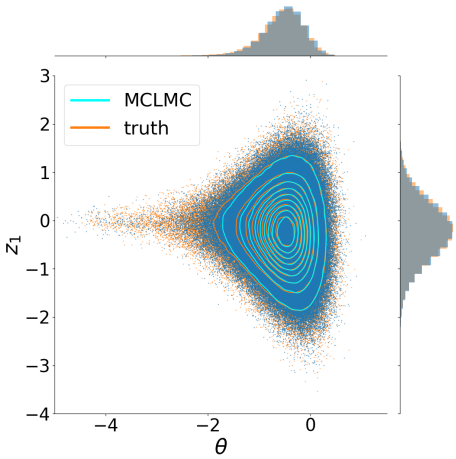

Figure 4: Posterior density for the funnel problem. The 2D marginal distribution in the $\theta - z_1$ plane and the corresponding 1D marginals are shown. The contours are obtained by kernel density estimation, and samples are shown as dots. The ground truth, obtained by a very long NUTS chain, is shown in orange, and unadjusted MCLMC is shown in blue. Both chains are 10^7 samples long to eliminate the variance error. The two methods give practically indistinguishable posteriors, demonstrating that the discretization bias was successfully suppressed by the EEVPD control.

7 Experiments

Our central contribution is a scheme which makes unadjusted HMC, LMC and MCLMC *black-box samplers*, in the sense of requiring no manual tuning on the part of a user. It is therefore natural to compare performance to the state of the art black-box sampler, the No-U-Turn Sampler (NUTS; Hoffman & Gelman (2014)). In addition, it is of interest to compare unadjusted samplers to their adjusted counterparts.

Samplers With this in mind, we report results on NUTS, unadjusted LMC (uLMC), adjusted LMC (aLMC), in particular, the version proposed in (Riou-Durand & Vogrinc, 2023), which is also known as Metropolis Adjusted Langevin Trajectories (MALT), unadjusted MCLMC (uMCLMC), and adjusted MCLMC (aMCLMC). While the other algorithms are evaluated with their respective tuning schemes, for aLMC, we perform a grid search to demonstrate that black-box uLMC outperforms even optimal aLMC. We omit reporting of unadjusted and adjusted HMC, since the performance and implementation closely resemble that of LMC, and LMC is generally considered the preferred option (Riou-Durand & Vogrinc, 2023). Further details of the experiment are provided in Appendix D.5. For the unadjusted methods, we use the scheme from Section 6 and take EEVPD of 3×10^{-4}, corresponding to the RMSE = 10%, which will be our notion of convergence. For MCLMC we use a slightly larger value of 5×10^{-4}, as suggested by Figure 3. In Appendix E, we perform an ablation study for LMC, to examine the change in performance as desired EEVPD is varied. The performance does not change much in a range of reasonable EEVPD, and the value of 3×10^{-4} is conservative, in the sense that larger values improve performance. The only exception is Stochastic Volatility, where we find that a smaller value is needed; we use 5×10^{-7} for HMC, and 2×10^{-8} for MCLMC.

Evaluation metric In addition to $b^2_{cov}(\Sigma, \Sigma_{\text{sampler}})$ we consider a more standard metric of convergence: following Hoffman & Sountsov (2022) we define the squared error of the expectation value $\mathbb{E}[f(\boldsymbol{x})]$ as

$$b^2(f) = \frac{(\mathbb{E}_{\text{sampler}}[f] - \mathbb{E}[f])^2}{\text{Var}[f]}, \tag{11}$$

and consider the average second-moment error across parameters, $b^2_{avg} \equiv d^{-1} \sum_{i=1}^{d} b^2(x_i^2)$. b^2_{avg} can be interpreted as the accuracy equivalent to 100 effective samples (Hoffman & Sountsov, 2022). In typical applications, computing the gradients $\nabla \log p(\boldsymbol{x})$ dominates the total sampling cost, so we take the number of gradient evaluations as a proxy of a wall-clock time. As in (Hoffman & Sountsov, 2022), we measure the sampler's performance as the number of gradient calls n needed to achieve low error, $b^2_{avg} < 0.01$ or $b^2_{cov} < 0.01$. We estimate the standard deviation of results by a bootstrapping procedure (see Table 4). We avoid using the effective sample size from the chain autocorrelation (Gelman et al., 2013) because it would give unadjusted methods an unfair advantage, given that it only measures the Monte Carlo variance error, but not the asymptotic bias.

	aLMC *grid search*	**uLMC** *EEVPD*	**aMCLMC** *acc. rate*	**uMCLMC** *EEVPD*	**uLMC** *grid search*	**NUTS** *acc. rate*
Standard Gaussian	803	**563**	436	**246**	568	2391
Rosenbrock	19,862	**16,820**	18,214	**10,688**	8410	27,070
Brownian	4667	**2168**	2876	**1628**	2407	5334
German Credit	7924	**4730**	6123	**3960**	4423	10,484
Item Response	5234	**2020**	5470	**1612**	930	6944
Stochastic Volatility	**37,904**	40,131	38,357	**16,854**	26,982	30,234

Table 1: Number of gradient calls needed to get b^2_{avg} below 0.01. Lower is better. The row below the method name denotes how the step size was determined.

Bayesian benchmarks Table 1 shows the results on a set of common benchmarks for Bayesian inference adapted from the Inference Gym (Sountsov et al., 2020) and described in appendix D.1. The corresponding results for the b_{cov} metric are shown in Table 3 in the appendix. The problems vary in dimensionality (36–2429) and are both synthetic (the first three problems) and with real data (the last three problems). We see that the unadjusted algorithms almost always perform better than their adjusted counterparts and better than NUTS. This is especially impressive since the hyperparameters of the adjusted samplers were found by grid search, or were shown to be near optimal in the case of aMCLMC (Robnik et al., 2025). Brownian motion was the only problem where EEVPD-based bound in Section 5 has failed, but here the unadjusted samplers nonetheless converge to the desired error and achieve close to optimal performance. Tables 1 and 3 also show unadjusted LMC with hyperparameters (L, ϵ) determined by grid search. This shows that the performance of LMC with our tuning is close to optimal, except for Rosenbrock and Item response where performance is off by a factor of two, due to our conservative choice of EEVPD (see Appendix E).

ϕ^4 field theory We test our scheme on a real-world problem from statistical physics, the ϕ^4 model in $1024^2 > 10^6$ dimensions (see Appendix D.3). We focus on the unadjusted microcanonical sampler since it dominated in all the above experiments and compare it to the NUTS baseline. We find that NUTS obtains $b^2_{\mathrm{avg}} < 0.01$ in $133,266$ gradient calls. Adjusted MCLMC yields the same in $39,240$ calls, while unadjusted MCLMC remarkably only needs 1344 gradient calls, a 100 fold improvement over NUTS. We view this as an exemplary case of constant efficiency with varying dimension of an unadjusted method (as shown in figure 5), here on a non-product, realistic problem.

Marginal posterior Figure 4 shows the marginal posterior density for the funnel problem, obtained by the unadjusted MCLMC algorithm with asymptotic bias of 1% requirement. The posterior is practically indistinguishable from a very long NUTS chain, showing that the discretization error is very small, as desired and demonstrating that our scheme produces accurate marginal posteriors.

8 Conclusions

We have shown that unadjusted gradient-based samplers can be turned into fully automatic black-box algorithms. With our adaptation scheme, unadjusted HMC, LMC and MCLMC do not need manual tuning and consistently outperform adjusted methods like NUTS, especially in high dimensions. This makes them a scalable alternative for a broad range of applications, from probabilistic programming to computational science, and opens new directions for efficient Bayesian inference.

Limitations The proposed scheme automatically selects a step size that in general performs well but it is certainly possible that there are models where bias is larger than expected. We therefore encourage the practice of validating results common in numerical analysis (Dalla Brida & Lüscher, 2017): rerun the sampler with a smaller step size and check that results do not change significantly. Since the asymptotic bias depends strongly on the step size this provides strong evidence that the asymptotic bias is negligible. We note that running a few parallel MCMC chains for validation purposes is a common practice (Gelman & Rubin, 1992; Margossian & Gelman, 2024), even for Metropolis adjusted chains, which still have the initialization bias, and is therefore not a drastic change to the standard MCMC workflow.

REPRODUCIBILITY STATEMENT

Proofs are provided in Appendix B. The technical details behind the experiments are in Appendix D. The code itself is provided in anonymous GitHub repositories. Samplers are implemented in https://anonymous.4open.science/r/blackjax-2118/. Code for benchmarking is in https://anonymous.4open.science/r/sampler-benchmarks-5B6A/README.md and the code to specifically obtain the results in this paper is https://anonymous.4open.science/r/sampler-benchmarks-5B6A/uHMC_paper/uHMC_paper_results.md.

REFERENCES

Michael S Albergo, Denis Boyda, Daniel C Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racaniere, Danilo Jimenez Rezende, and Phiala E Shanahan. Introduction to normalizing flows for lattice field theory. *arXiv preprint arXiv:2101.08176*, 2021.

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18, 2018. ISSN 1532-4435. URL https://ora.ox.ac.uk/objects/uuid%3Af2b14395-73ac-4e91-a8ca-f37168dcf589?utm_source=chatgpt.com. Publisher: Journal of Machine Learning Research.

Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19 (5A):1501–1534, November 2013. ISSN 1350-7265. doi: 10.3150/12-BEJ414. URL https://projecteuclid.org/journals/bernoulli/volume-19/issue-5A/Optimal-tuning-of-the-hybrid-Monte-Carlo-algorithm/10.3150/12-BEJ414.full. Publisher: Bernoulli Society for Mathematical Statistics and Probability.

Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018. URL http://arxiv.org/abs/1701.02434. arXiv:1701.02434 [stat].

Nawaf Bou-Rabee and Andreas Eberle. Mixing time guarantees for unadjusted Hamiltonian Monte Carlo. *Bernoulli*, 29(1):75–104, February 2023. ISSN 1350-7265. doi: 10.3150/21-BEJ1450. URL https://projecteuclid.org/journals/bernoulli/volume-29/issue-1/Mixing-time-guarantees-for-unadjusted-Hamiltonian-Monte-Carlo/10.3150/21-BEJ1450.full. Publisher: Bernoulli Society for Mathematical Statistics and Probability.

Nawaf Bou-Rabee and Katharina Schuh. Convergence of unadjusted Hamiltonian Monte Carlo for mean-field models. *Electronic Journal of Probability*, 28(none):1–40, January 2023. ISSN 1083-6489, 1083-6489. doi: 10.1214/23-EJP970. URL https://projecteuclid.org/journals/electronic-journal-of-probability/volume-28/issue-none/Convergence-of-unadjusted-Hamiltonian-Monte-Carlo-for-mean-field-models/10.1214/23-EJP970.full. Publisher: Institute of Mathematical Statistics and Bernoulli Society.

Alberto Cabezas, Adrien Corenflos, Junpeng Lao, and Rémi Louf. Blackjax: Composable bayesian inference in jax. 2024. URL https://arxiv.org/abs/2402.10797.

Jean-Eric Campagne, François Lanusse, Joe Zuntz, Alexandre Boucaud, Santiago Casas, Minas Karamanis, David Kirkby, Denise Lanzieri, Yin Li, and Austin Peel. JAX-COSMO: An End-to-End Differentiable and GPU Accelerated Cosmology Library. *The Open Journal of Astrophysics*, 6: 10.21105/astro.2302.05163, April 2023. ISSN 2565-6120. doi: 10.21105/astro.2302.05163. URL http://arxiv.org/abs/2302.05163. arXiv:2302.05163 [astro-ph].

Evan Camrud, Alain Durmus, Pierre Monmarché, and Gabriel Stoltz. Second order quantitative bounds for unadjusted generalized Hamiltonian Monte Carlo, May 2024. URL http://arxiv.org/abs/2306.09513. arXiv:2306.09513 [math].

Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76:1–32, January 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01. URL https://doi.org/10.18637/jss.v076.i01.

Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

M. A. Clark and A. D. Kennedy. Asymptotics of fixed point distributions for inexact Monte Carlo algorithms. *Physical Review D*, 76(7):074508, October 2007. doi: 10.1103/PhysRevD.76.074508. URL https://link.aps.org/doi/10.1103/PhysRevD.76.074508. Publisher: American Physical Society.

Mattia Dalla Brida and Martin Lüscher. SMD-based numerical stochastic perturbation theory. *The European Physical Journal C*, 77(5):308, May 2017. ISSN 1434-6052. doi: 10.1140/epjc/s10052-017-4839-0. URL https://doi.org/10.1140/epjc/s10052-017-4839-0.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987. ISSN 0370-2693. doi: 10.1016/0370-2693(87)91197-X. URL https://www.sciencedirect.com/science/article/pii/037026938791197X.

Alain Oliviero Durmus and Andreas Eberle. Asymptotic bias of inexact Markov Chain Monte Carlo methods in high dimension, April 2023. URL http://arxiv.org/abs/2108.00682. arXiv:2108.00682 [cs, math, stat].

Denis J Evans and Brad Lee Holian. The nose-hoover thermostat. *Journal of Chemical Physics*, 83 (8):4069–4074, 1985.

Christof Gattringer and Christian Lang. *Quantum chromodynamics on the lattice: an introductory presentation*, volume 788. Springer Science & Business Media, 2009.

Christof Gattringer and Christian B. Lang. *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, volume 788 of *Lecture Notes in Physics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-01849-7 978-3-642-01850-3. doi: 10.1007/978-3-642-01850-3. URL https://link.springer.com/10.1007/978-3-642-01850-3.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, November 1992. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177011136. URL https://projecteuclid.org/journals/statistical-science/volume-7/issue-4/Inference-from-Iterative-Simulation-Using-Multiple-Sequences/10.1214/ss/1177011136.full. Publisher: Institute of Mathematical Statistics.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. Bayesian data analysis third edition. *Chapman and Hall/CRC*, 2013.

Mathis Gerdes, Pim de Haan, Corrado Rainone, Roberto Bondesan, and Miranda CN Cheng. Learning lattice quantum field theories with equivariant continuous flows. *arXiv preprint arXiv:2207.00283*, 2022.

Nigel Goldenfeld. *Lectures on phase transitions and the renormalization group*. CRC Press, 2018.

Nicolaï Gouraud, Pierre Le Bris, Adrien Majka, and Pierre Monmarché. Hmc and underdamped langevin united in the unadjusted convex smooth case. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1):278–303, 2025.

Andreas Griewank and Andrea Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition. Society for Industrial and Applied Mathematics, January 2008. ISBN 978-0-89871-659-7 978-0-89871-776-1. doi: 10.1137/1.9780898717761. URL http://epubs.siam.org/doi/book/10.1137/1.9780898717761. Edition: Second.

Richard D. P. Grumitt, Biwei Dai, and Uros Seljak. Deterministic Langevin Monte Carlo with Normalizing Flows for Bayesian Inference, October 2022. URL http://arxiv.org/abs/2205.14240. arXiv:2205.14240 [cond-mat, physics:physics, stat].

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. ISSN 1533-7928. URL http://jmlr.org/papers/v15/hoffman14a.html.

Matthew D. Hoffman and Pavel Sountsov. Tuning-Free Generalized Hamiltonian Monte Carlo. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 7799–7813. PMLR, May 2022. URL https://proceedings.mlr.press/v151/hoffman22a.html. ISSN: 2640-3498.

Alan M. Horowitz. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2):247–252, October 1991. ISSN 0370-2693. doi: 10.1016/0370-2693(91)90812-5. URL https://www.sciencedirect.com/science/article/pii/0370269391908125.

Benjamin Horowitz and Zarija Lukic. Differentiable Cosmological Hydrodynamics for Field-Level Inference and High Dimensional Parameter Constraints, February 2025. URL http://arxiv.org/abs/2502.02294. arXiv:2502.02294 [astro-ph].

Peter L Houtekamer and Herschel L Mitchell. Ensemble kalman filtering. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3269–3289, 2005.

L. V. Kantorovich. Mathematical Methods of Organizing and Planning Production. *Management Science*, 6(4):366–422, July 1960. ISSN 0025-1909. doi: 10.1287/mnsc.6.4.366. URL https://pubsonline.informs.org/doi/10.1287/mnsc.6.4.366. Publisher: INFORMS.

Ben Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, volume 39 of *Interdisciplinary Applied Mathematics*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16374-1 978-3-319-16375-8. doi: 10.1007/978-3-319-16375-8. URL https://link.springer.com/10.1007/978-3-319-16375-8.

Martin Lüscher. Stochastic locality and master-field simulations of very large lattices. *EPJ Web Conf.*, 175:01002, 2018. doi: 10.1051/epjconf/201817501002. _eprint: 1707.09758.

Charles C. Margossian. A review of automatic differentiation and its efficient implementation. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1305, 2019. ISSN 1942-4795. doi: 10.1002/widm.1305. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1305. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1305.

Charles C. Margossian and Andrew Gelman. For how many iterations should we run Markov chain Monte Carlo?, February 2024. URL http://arxiv.org/abs/2311.02726. arXiv:2311.02726 [stat].

Charles C. Margossian, Matthew D. Hoffman, Pavel Sountsov, Lionel Riou-Durand, Aki Vehtari, and Andrew Gelman. Nested R^: Assessing the Convergence of Markov Chain Monte Carlo When Running Many Short Chains. *Bayesian Analysis*, -1(-1):1–28, January 2024. ISSN 1936-0975, 1931-6690. doi: 10.1214/24-BA1453. URL https://projecteuclid.org/journals/bayesian-analysis/advance-publication/Nested-R%cb%86--Assessing-the-Convergence-of-Markov-Chain-Monte/10.1214/24-BA1453.full. Publisher: International Society for Bayesian Analysis.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

Marius Millea and Uroš Seljak. Marginal unbiased score expansion and application to cmb lensing. *Physical Review D*, 105(10):103531, 2022.

Peter Minary, Glenn J. Martyna, and Mark E. Tuckerman. Algorithms and novel applications based on the isokinetic ensemble. I. Biophysical and path integral molecular dynamics. *The Journal of Chemical Physics*, 118(6):2510–2526, February 2003. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.1534582. URL https://pubs.aip.org/jcp/article/118/6/2510/438540/Algorithms-and-novel-applications-based-on-the.

Radford M. Neal. *MCMC using Hamiltonian dynamics*. May 2011. doi: 10.1201/b10905. URL http://arxiv.org/abs/1206.1901. arXiv:1206.1901 [physics, stat].

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, August 2009. ISSN 1436-4646. doi: 10.1007/s10107-007-0149-x. URL https://doi.org/10.1007/s10107-007-0149-x.

I. Olkin and F. Pukelsheim. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, December 1982. ISSN 0024-3795. doi: 10.1016/0024-3795(82)90112-4. URL https://www.sciencedirect.com/science/article/pii/0024379582901124.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. In *Proceedings of the 2nd Symposium on Advances in Approximate Bayesian Inference*, 2019. URL https://arxiv.org/abs/1912.11554. arXiv preprint arXiv:1912.11554.

Lionel Riou-Durand and Jure Vogrinc. Metropolis Adjusted Langevin Trajectories: a robust alternative to Hamiltonian Monte Carlo, December 2023. URL http://arxiv.org/abs/2202.13230. arXiv:2202.13230 [math, stat].

Jakob Robnik and Uros Seljak. Fluctuation without dissipation: Microcanonical Langevin Monte Carlo. In *Proceedings of the 6th Symposium on Advances in Approximate Bayesian Inference*, pp. 111–126. PMLR, July 2024. URL https://proceedings.mlr.press/v253/robnik24a.html. ISSN: 2640-3498.

Jakob Robnik, G. Bruno De Luca, Eva Silverstein, and Uroš Seljak. Microcanonical Hamiltonian Monte Carlo. *The Journal of Machine Learning Research*, 24(1):311:14696–311:14729, March 2024. ISSN 1532-4435.

Jakob Robnik, Reuben Cohn-Gordon, and Uros Seljak. Metropolis Adjusted Microcanonical Hamiltonian Monte Carlo. September 2025. URL https://openreview.net/forum?id=Y38oEwTLTQ.

Peter J Rossky, Jimmie D Doll, and Harold L Friedman. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

Hugo Simon-Onfroy, François Lanusse, and Arnaud de Mattia. Benchmarking field-level cosmological inference from galaxy redshift surveys, April 2025. URL http://arxiv.org/abs/2504.20130. arXiv:2504.20130 [astro-ph].

Pavel Sountsov, Alexey Radul, and contributors. Inference gym, 2020. URL https://pypi.org/project/inference_gym.

Greg Ver Steeg and Aram Galstyan. Hamiltonian Dynamics with Non-Newtonian Momentum for Rapid Sampling, December 2021. URL http://arxiv.org/abs/2111.02434. arXiv:2111.02434 [physics].

Ingmar Vierhaus. Simulation of phi 4 theory in the strong coupling expansion beyond the ising limit. Master's thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2010.

Cédric Villani. *Topics in optimal transportation*. Graduate studies in mathematics. American Mathematical Society, Providence, RI, 2003. ISBN 978-0-8218-3312-4. URL http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&doc_number=010185636&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA. OCLC: 51477002.

Erik Štrumbelj, Alexandre Bouchard-Côté, Jukka Corander, Andrew Gelman, Håvard Rue, Lawrence Murray, Henri Pesonen, Martyn Plummer, and Aki Vehtari. Past, Present and Future of Software for Bayesian Inference. *Statistical Science*, 39(1):46–61, February 2024. ISSN 0883-4237, 2168-8745. doi: 10.1214/23-STS907. URL https://projecteuclid.org/journals/statistical-science/volume-39/issue-1/Past-Present-and-Future-of-Software-for-Bayesian-Inference/10.1214/23-STS907.full. Publisher: Institute of Mathematical Statistics.

A COVARIANCE MATRIX ERROR

Let SPD(d) be the space of symmetric positive definite matrices of size $d \times d$.

Lemma A.1. $b^2_{cov}(A, B) = \frac{1}{d}\operatorname{Tr}\{(I - A^{-1}B)^2\}$ *is a divergence on SPD(d), meaning that for all $A, B \in$ SPD(d),*

1. $b^2_{cov}(A, B) \geq 0$
2. $b^2_{cov}(A, B) = 0$ *if and only if* $A = B$.

Proof. Without loss of generality, we may assume that A is diagonal with positive entries, because the trace is invariant under the change of basis and A must be diagonal in some basis because it is positive-definite. $R \equiv I - A^{-1}B$ then has $R_{ii} = 1 - A^{-1}_{ii}B_{ii}$ on the diagonal and $R_{ij} = -A^{-1}_{ii}B_{ij}$ off the diagonal. The trace is

$$\operatorname{Tr}\{R^2\} = \sum_{i=1}^{d}(1 - A^{-1}_{ii}B_{ii})^2 + \sum_{i \neq j} A^{-1}_{ii}B_{ij}A^{-1}_{jj}B_{ji},$$

where the first and the second term are the contribution from the diagonal and off-diagonal elements respectively. Both terms are non-negative: the first term because it is a sum of squares, the second because all factors A_{ii}, A_{jj} and $B_{ij}B_{ji} = (B_{ij})^2$ are non-negative. This already proves (1).

The implication $b^2_{cov}(A, A) = 0$ in (2) is trivial. To prove the other implication in (2), now suppose $b^2_{cov}(A, B) = 0$. This implies that both terms in the above equation are zero, as they are both non-negative. In the second term, $A_{ii} > 0$ are non-zero for any i, so the term can only be zero if $B_{ij} = 0$ for any $i \neq j$. The first term can only be non-zero if $A_{ii} = B_{ii}$ for any i. We have shown that $A_{ij} = B_{ij}$ for any i, j, thus $A = B$. □

Note that the above result is not obvious from the fact that we are computing a trace of the matrix squared. There are non-zero matrices whose square is zero.

b^2_{cov} has another nice property – it can be related to the effective sample size in a covariance matrix independent way:

Lemma A.2. *Let $\boldsymbol{x}^{(k)}$ for $k = 1, 2 \ldots n$ be exact i.i.d samples from $p = \mathcal{N}(0, \Sigma)$ and let $\bar{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T$, be the empirical estimate for Σ. Then*

$$\mathbb{E}_{MC}[b^2_{cov}(\Sigma, \bar{\Sigma})] = (d + 1)/n,$$

where $\mathbb{E}_{MC}[\cdot]$ is the expectation with respect to the sample realizations.

Proof. The empirical estimate is unbiased:

$$\mathbb{E}_{MC}[\bar{\Sigma}] = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}_{MC}[\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T] = \Sigma,$$

and

$$\mathbb{E}_{MC}[\bar{\Sigma}_{ab}\bar{\Sigma}_{cd}] = \Sigma_{ab}\Sigma_{cd} + \frac{1}{n}\big(\Sigma_{ac}\Sigma_{bd} + \Sigma_{ad}\Sigma_{bc}\big).$$

The expected error of the empirical covariance matrix is then:

$$\begin{aligned}\mathbb{E}_{MC}[b_{cov}^2] &= 1 - 2\frac{1}{d}[\Sigma^{-1}]_{ij}\mathbb{E}_{MC}[\bar{\Sigma}_{ij}] + \frac{1}{d}[\Sigma^{-1}]_{ij}\mathbb{E}_{MC}[\bar{\Sigma}_{jk}\bar{\Sigma}_{li}][\Sigma^{-1}]_{kl} \\ &= 1 - 2 + \frac{1}{d}[\Sigma^{-1}]_{ij}\left(\Sigma_{jk}\Sigma_{li} + \frac{1}{n}\big(\Sigma_{jl}\Sigma_{ki} + \Sigma_{ji}\Sigma_{kl}\big)\right)[\Sigma^{-1}]_{kl} = \frac{d+1}{n},\end{aligned}$$

where Einstein's convention of summing over the repeated indices was used. □

This result implies that for Gaussian distributions, b_{cov}^2 could be used to define the effective sample size (ESS) of the estimate. Concretely, given some samples whose empirical covariance matrix error is b_{cov}^2, we can define their effective sample size to be $n_{\text{eff}} = (d+1)/b_{cov}^2$, i.e., this is the number of exact samples that would yield the same covariance matrix error as given samples. For example, given a target distribution p in $d = 9$, suppose it takes 1000 samples from a Markov chain to achieve $b_{cov}^2 = 0.1$. This would correspond to $(9+1)/0.1 = 100$ effective samples, so 0.1 effective samples per step. Although this result rigorously only holds for Gaussian target distributions it offers some interpreteability to the value of b_{cov}.

Note that we could also take the more transparently non-negative definition $b_F^2(A, B) = \frac{1}{d}\operatorname{Tr}\{(I - A^{-1}B)((I - A^{-1}B)^T\} = \frac{1}{d}\left\|I - A^{-1}B\right\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. However, b_F^2 cannot be related to the effective sample size in a covariance matrix independent way. Instead,

$$\begin{aligned}\mathbb{E}_{MC}[b_F^2] &= \frac{1}{d}\mathbb{E}_{MC}[\big(\delta_{ij} - [\Sigma^{-1}]_{ik}\bar{\Sigma}_{kj}\big)\big(\delta_{ij} - [\Sigma^{-1}]_{il}\bar{\Sigma}_{lj}\big)] \\ &= 1 - 2\frac{1}{d}[\Sigma^{-1}]_{ij}\mathbb{E}_{MC}[\bar{\Sigma}_{ij}] + \frac{1}{d}[(\Sigma^{-1})^2]_{ij}\mathbb{E}_{MC}[\bar{\Sigma}^2]_{ji} = \frac{1}{n}\big(1 + \frac{1}{d}\operatorname{Tr}\{\Sigma\}\operatorname{Tr}\{\Sigma^{-1}\}\big),\end{aligned} \tag{12}$$

so we will note use this definition here.

Finally, b_{cov} is invariant to the linear change of basis

Lemma A.3. *For an invertible matrix A, and a change of basis $\boldsymbol{x}' = A\boldsymbol{x}$, the covariance matrix error does not change, that is, $b_{cov}^2(\Sigma_{p(\boldsymbol{x})}, \Sigma_{q(\boldsymbol{x})}) = b_{cov}^2(\Sigma_{p(\boldsymbol{x}')}, \Sigma_{q(\boldsymbol{x}')})$.*

Proof. The covariance matrix transforms as

$$[\Sigma_{p(\boldsymbol{x}')}]_{ij} = \int x_i' x_j' p(\boldsymbol{x}') d\boldsymbol{x}' = [A\,\Sigma_{p(\boldsymbol{x})}A^T]_{ij},$$

hence the covariance matrix bias is invariant:

$$\begin{aligned}b_{cov}^2(\Sigma_{p(\boldsymbol{x})}, \Sigma_{q(\boldsymbol{x})}) &= \frac{1}{d}\operatorname{Tr}\Big\{\big(I - (A\Sigma_{p(\boldsymbol{x})}A^T)^{-1}(A\Sigma_{q(\boldsymbol{x})}A^T)\big)^2\Big\} \\ &= \frac{1}{d}\operatorname{Tr}\Big\{\big(I - (A^T)^{-1}\Sigma_{p(\boldsymbol{x})}^{-1}A^{-1})(A\Sigma_{q(\boldsymbol{x})}A^T)\big)^2\Big\} \\ &= \frac{1}{d}\operatorname{Tr}\Big\{\big(I - (A^T)^{-1}\Sigma_{p(\boldsymbol{x})}^{-1}\Sigma_{q(\boldsymbol{x})}A^T\big)^2\Big\} \\ &= \frac{1}{d}\operatorname{Tr}\Big\{(A^T)^{-1}\big(I - \Sigma_{p(\boldsymbol{x})}^{-1}\Sigma_{q(\boldsymbol{x})}\big)^2 A^T\Big\} \\ &= b_{cov}^2(\Sigma_{p(\boldsymbol{x}')}, \Sigma_{q(\boldsymbol{x}')}).\end{aligned}$$

□

B PROOFS

B.1 LEMMA 4.1: EEVPD

Proof. We will work in the eigenbasis, where the dynamics is decoupled. It then suffices to analyze each dimension separately. Let $x_i(t)$ and $u_i(t)$ be components of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ along the dimension

that we analyze and σ_i^2 the eigenvalue of the covariance matrix in that direction. The velocity Verlet integrator update (4) with step size ϵ can be written compactly as (Gouraud et al., 2025)

$$\Phi_\epsilon(\boldsymbol{z}_i) = \begin{bmatrix} \cos h & \alpha\sigma_i \sin h \\ -\alpha^{-1}\sigma_i^{-1}\sin h & \cos h \end{bmatrix} \boldsymbol{z}_i \equiv A\boldsymbol{z}_i.$$

Here, $\boldsymbol{z}_i(t) = (x_i(t), u_i(t))$, $\alpha = (1 - y_i/4)^{-1/2}$, $y_i = \epsilon^2/\sigma_i^2$ and $\sin h = \sqrt{y_i}/\alpha$. The energy error is

$$\Delta_H^i \equiv \Delta_H(\Phi_\epsilon(\boldsymbol{z}_i), \boldsymbol{z}_i) = \frac{1}{2}\Phi_\epsilon(\boldsymbol{z}_i)^T D \Phi_\epsilon(\boldsymbol{z}_i) - \frac{1}{2}\boldsymbol{z}_i^T D \boldsymbol{z}_i = \frac{1}{2}\boldsymbol{z}_i^T M \boldsymbol{z}_i,$$

where $D = \mathrm{Diag}(1/\sigma_i^2, 1)$ and

$$M = A^T D A - D = (1-\alpha^2)\begin{bmatrix} (\alpha\sigma_i)^{-2}\sin^2 h & -(\alpha\sigma_i)^{-1}\sin h\cos h \\ -(\alpha\sigma_i)^{-1}\sin h\cos h & -\sin^2 h \end{bmatrix}.$$

Denote by $\tilde{p}_i$ the stationary distribution for $\boldsymbol{z}_i$, namely $x_i \sim \mathcal{N}(0, \tilde{\sigma}_i)$, $u_i \sim \mathcal{N}(0,1)$, where $\tilde{\sigma}_i = \sigma_i/\alpha$ is taken from Equation (10).

The contribution to the EEVPD from $\boldsymbol{z}_i$ is

$$\mathrm{Var}_{\tilde{p}_i}[\Delta_H^i] = \mathbb{E}_{\tilde{p}_i}[(\Delta_H^i)^2] - \mathbb{E}_{\tilde{p}}[\Delta_H^i]^2.$$

The expectation value in the second term is

$$\mathbb{E}_{\tilde{p}}[\Delta_H^i] = \frac{1}{2}\big(M_{11}\mathbb{E}_{\tilde{p}}[x^2] + M_{22}\mathbb{E}_{\tilde{p}}[u^2]\big)$$

and for the first

$$\begin{aligned}\mathbb{E}_{\tilde{p}}[(\Delta_H^i)^2] &= \frac{1}{4}\mathbb{E}_{\tilde{p}}[(M_{11}x^2 + 2M_{12}xu + M_{22}u^2)^2] \\ &= \frac{1}{4}\big(M_{11}^2\mathbb{E}_{\tilde{p}}[x^4] + M_{22}^2\mathbb{E}_{\tilde{p}}[u^4] + (4M_{12}^2 + 2M_{11}M_{22})\mathbb{E}_{\tilde{p}}[x^2]\mathbb{E}_{\tilde{p}}[u^2]\big).\end{aligned}$$

In both expressions we have dropped the vanishing contributions that contain $\mathbb{E}_{\tilde{p}}[xu] = 0$, $\mathbb{E}_{\tilde{p}}[xu^3] = 0$ or $\mathbb{E}_{\tilde{p}}[x^3u] = 0$. Combining both terms together and using $\mathbb{E}_{\tilde{p}}[x^4] = 3\mathbb{E}_{\tilde{p}}[x^2]^2$ and $\mathbb{E}_{\tilde{p}}[u^4] = 3\mathbb{E}_{\tilde{p}}[u^2]^2$ we get

$$\mathrm{Var}[\Delta_H^i] = \frac{1}{4}\big(2M_{11}^2\mathbb{E}_{\tilde{p}}[x^2]^2 + 2M_{22}^2\mathbb{E}_{\tilde{p}}[u^2]^2 + 4M_{12}^2\mathbb{E}_{\tilde{p}}[x^2u^2]\big).$$

Inserting $\mathbb{E}_{\tilde{p}}[x^2] = \sigma^2\alpha^2$ and $\mathbb{E}_{\tilde{p}}[u^2] = 1$ gives

$$\mathrm{Var}[\Delta_H^i] = \frac{(1-\alpha^2)^2}{2}\big(\sin^4 h + \sin^4 h + 2\sin^2 h\cos^2 h\big) = (1-\alpha^2)^2\sin^2 h = \frac{y^3}{16(1-y/4)},$$

which is $E(y)$ from the statement of the theorem. EEVPD is thus

$$\mathrm{EEVPD} = \frac{1}{d}\sum_{i=1}^{d}\mathrm{Var}[\Delta_H^i] = \frac{1}{d}\sum_{i=1}^{d}E(y_i).$$

□

B.2 Theorem 4.2: Bias Bounds

Proof. Let's start with the covariance matrix bias. Due to Equation (10), the asymptotic covariance error (6) is

$$b_{cov}^2(\Sigma, \tilde{\Sigma}) = \frac{1}{d}\sum_{i=1}^{d}\big(1 - (1-\epsilon^2/4\sigma_i^2)^{-1}\big)^2 = \frac{1}{d}\sum_{i=1}^{d}B(\epsilon^2/\sigma_i^2),$$

where $B(y) = \frac{y^2}{16(1-y/4)^2}$. We thus see that φ from the statement of the theorem is $\varphi = E \circ B^{-1}$. Lemma B.1 shows that $\varphi(x)$ restricted to $0 < x < 11 - 4\sqrt{7}$ is a convex, monotonically increasing function. By Jensen's inequality this implies that

$$\varphi(b_{cov}(\Sigma, \tilde{\Sigma})^2) \leq \mathrm{EEVPD},$$

as long as $\varphi(b_{cov}^2) < \varphi(11-4\sqrt{7}) = (-134+52\sqrt{7})/9 \approx 0.397674$. The assumption of the theorem that $\mathrm{EEVPD} < 0.397$ is a sufficient condition for this to hold. Since $\varphi(x)$ is not a linear function, Jensen's inequality becomes an equality if and only if $\sigma_i = \sigma_j$ for all i, j. φ is a monotonically increasing function so it is a bijection and its inverse is also a monotonically increasing function. The inverse can therefore be applied to both sides of the above inequality to yield the desired result.

The proof of the Wasserstein distance part of the theorem is similar. For zero-mean Gaussian distributions with covariance matrices Σ_p and Σ_q, the Wasserstein distance reduces to (Olkin & Pukelsheim, 1982)

$$\mathcal{W}_2(\mathcal{N}(0, \Sigma_p), \mathcal{N}(0, \Sigma_q))^2 = \mathrm{Tr}\Big\{\Sigma_p + \Sigma_q - 2\big(\Sigma_p^{1/2}\Sigma_q\Sigma_p^{1/2}\big)^{1/2}\Big\}.$$

By using Equation (10) for unadjusted HMC or LMC with the velocity Verlet integrator we therefore get

$$\mathcal{W}_2(p, \tilde{p})^2 = \epsilon^2 \sum_{i=1}^{d} W(\epsilon^2/\sigma_i^2),$$

where

$$W(y) = \frac{2(1 - y/8 - \sqrt{1 - y/4})}{y(1 - y/4)}$$

is as in the statement of the theorem. Lemma B.2 shows that $\varphi_W(x)$ is a convex, monotonically increasing function for $0 < \varphi_W(x) < 27/4 = 6.75$. By Jensen's inequality this implies that

$$\varphi_W(w) \leq \mathrm{EEVPD},$$

as long as $\mathrm{EEVPD} < 6.75$. Here $w = \mathcal{W}_2(p, \tilde{p})^2/d\epsilon^2$. Since φ_W is not a linear function, Jensen's inequality becomes an equality if and only if $\sigma_i = \sigma_j$ for all i, j. φ_W is a monotonically increasing function so it is a bijection and its inverse is also a monotonically increasing function. The inverse can therefore be applied to both sides of the above inequality to yield

$$w \leq \varphi_W^{-1}(\mathrm{EEVPD}).$$

□

B.3 Convexity

Lemma B.1. *$\varphi(x) = 4x^{3/2}/(1 + x^{1/2})^2$ is monotonically increasing for $x > 0$ and convex for $0 < x < 11 - 4\sqrt{7}$.*

Proof. $\varphi(x)$ is monotonically increasing because its derivative

$$\varphi'(x) = \frac{2x^{1/2}(3 + x^{1/2})}{(1 + x^{1/2})^3}$$

is positive (all terms are positive). To show that it is convex, we compute its second derivative

$$\varphi''(x) = \frac{3 - 4x^{1/2} - x}{x^{1/2}(1 + x^{1/2})^4}$$

The denominator is positive for $x > 0$. The numerator is a quadratic polynomial $p(y) = 3 - 4 - y^2$ in $y = \sqrt{x}$. Its roots are $y_{1,2} = -2 \pm \sqrt{7}$. Since $p(0) = 3 > 0$ the numerator is positive for $0 < y < -2 + \sqrt{7}$ corresponding to $0 < x < (-2 + \sqrt{7})^2 = 11 - 4\sqrt{7}$ so $\varphi(x)$ restricted to this interval is convex. □

Lemma B.2. *$\varphi_W(x)$ from Theorem 4.2 is monotonically increasing and convex for $0 < \varphi_W(x) < 27/4$.*

Proof. To prove that φ_W is monotonically increasing we will show that its derivative is positive. We cannot solve for for W^{-1} explicitly, but nontheless

$$\varphi_W'(w) = E'(W^{-1}(w))(W^{-1})'(w) = \frac{E'(y)}{W'(y)},$$

where we have denoted $y = W^{-1}(w)$. We will show that both denominator and numerator are positive for $0 \leq y < 4$, i.e. in the range where the velocity Verlet integrator is stable.

The numerator is

$$E'(y) = \frac{3y^2(1 - y/6)}{(1 - y/4)^2}$$

which is positive for $0 < y < 4$.

To simplify the denominator we use the reparametrization $y = 4\sin^2\xi$, such that $0 \leq \xi < \pi/2$. In the new parametrization

$$W(\xi(y)) = \frac{\sin^4(\xi/2)}{4\sin^2(2\xi)}.$$

We get

$$W'(y) = \frac{1 + \cos\xi - \cos^2\xi}{128\cos^4(\xi/2)\cos^4(\xi)} \geq \frac{\cos\xi}{128\cos^4(\xi/2)\cos^4(\xi)} > 0,$$

where we have used that $-\cos^2\xi > -1$.

To prove that φ_W is convex, we will show that its second derivative is positive. We have

$$\varphi''_W(w) = E''(W^{-1}(w))((W^{-1})'(w))^2 + E'(W^{-1}(w))(W^{-1})''(w) = \frac{E''(y)}{W'(y)^2} - \frac{E'(y)W''(y)}{W'(y)^3}$$

Which after some algebra reduces to

$$\varphi''_W(y) = \frac{(2 - s)(s^2 - 1)(s^4 + 4s^3 + 7s^2 + 6s - 6)}{8(s^2 + s - 1)},$$

where we have first reparametrized $y(\xi) = 4\sin^2\xi$ and then $s(\xi) = 1/\cos(\xi)$. The range $0 < y < 4$ corresponds to $0 < \xi < \pi/2$ which in turn corresponds to $1 < s < \infty$.

All the factors except for $(2 - s)$ are non-negative at $s = 1$ and monotonically increasing for $s > 1$, so they are all positive for $s > 1$. The second derivative is then positive for $1 < s < 2$, corresponding to $0 < \xi < \pi/3$ or $0 < y < 3$. $\varphi_W(x)$ is thus monotonically increasing and convex for $0 < \varphi_W(x) < E(3) = 27/4$. □

C BIAS-VARIANCE TRADEOFF

We have shown how to control the covariance matrix bias to be below some desired threshold. However, typically bias is not of direct interest, but instead we want to control the error of the expectation values, which additionally contains the variance, i.e. the second term in Equation (2). The variance of a stationary chain is

$$\mathrm{Var}[\bar{f}] = \mathrm{Var}_p[f](1 + 2\sum_{k=1}^{n}(1 - k/n)\rho_k)/n \equiv \mathrm{Var}_p[f]\tau_{\mathrm{int}}/n, \tag{13}$$

where the autocorrelation coefficients are $\rho_k = \mathbb{E}[(f(x_i) - \mathbb{E}_p[f])(f(x_{i+k}) - \mathbb{E}_p[f])]/\mathrm{Var}_p[f]$.

We would therefore like to know how to optimally set the bias, given some error tolerance. Here we will provide a heuristic, based on estimating the second moment $\mathbb{E}[x^2]$ of a one-dimensional standard Gaussian target with velocity Verlet unadjusted HMC.

In this case, $\rho_k = \rho^k$ (Gouraud et al., 2025), where

$$\rho = \cos^2\big(\frac{T}{\epsilon}\arcsin(\alpha\epsilon/\sigma)\big). \tag{14}$$

This makes the sum in Equation (13) expressable in terms of geometric series:

$$\tau_{\mathrm{int}} = 1 + 2S(\rho) - 2\rho S'(\rho)/n = \frac{1 + \rho}{1 - \rho}\left(1 - \frac{2\rho}{n}\frac{1 - \rho^n}{1 - \rho^2}\right), \tag{15}$$

relative RMSE tolerance	Bias tolerance	EEVPD
50%	22%	3.0×10^{-2}
10%	4.5%	3.3×10^{-4}
5%	2.2%	4.3×10^{-5}
1%	0.45%	3.5×10^{-7}

Table 2: Tabulated values of EEVPD (third column) that ensure a desired asymptotic covariance matrix bias (second column). A useful, quick recipe to compute approximation for small b_{cov}^2 is $\mathrm{EEVPD} = \varphi(b_{cov}^2) \approx 4b_{cov}^3$. Optimal asymptotic bias should be smaller than the given relative root mean square error tolerance (Equation (1)), for example, one can use the prescription $\mathrm{Bias}^2 < \mathrm{RMSE}^2/5$ from Appendix C. This error is given in the first column.

where $S(\rho) = \sum_{k=1}^{n} \rho^k = \rho(1-\rho^n)/(1-\rho)$. The variance at the lowest order in the small-ϵ-expansion is then

$$\mathrm{Var}[x^2] \asymp \frac{2L}{N\epsilon} \lim_{\epsilon \to 0} \tau_{\mathrm{int}}(\epsilon) = c_v/\epsilon, \tag{16}$$

where c_v is constant, independent of the step size and $\asymp$ denotes asymptotic equivalence as $\epsilon \to 0$. The bias of the second moment in this limit is

$$\mathrm{Bias}[x^2] = \frac{1}{1-\epsilon^2/4} - 1 \asymp \epsilon^2/4 = c_b\epsilon^2, \tag{17}$$

where we have used Equation (10) and denoted $c_b = 1/4$.

Combining Equations (16) and (17) we get for the RMSE in the small step size limit:

$$\mathrm{RMSE}^2 = c_b\epsilon^4 + c_v/\epsilon, \tag{18}$$

where c_b and c_v are defined above and are independent of the step size. We would like to set the step size so that it minimizes the RMSE. The optimum is found at

$$0 = \frac{d}{d\epsilon}\mathrm{RMSE}^2 = 4c_b\epsilon^3 - c_v/\epsilon^2, \tag{19}$$

which gives the optimal step size $\epsilon_{\mathrm{opt}} = (c_v/4c_b)^{1/5}$. At the optimal step size, bias squared is one fifth of the error squared:

$$\frac{\mathrm{Bias}^2}{\mathrm{RMSE}^2} = \frac{1}{1 + \frac{c_v/\epsilon_{\mathrm{opt}}}{c_v\epsilon_{\mathrm{opt}}^4}} = \frac{1}{5}. \tag{20}$$

This is the prescription that we use in Table 2 and in the numerical experiments of Section 7. It is based on the Gaussian assumption, so it might suboptimal for the non-Gaussian distributions. However, the samples would still eventually converge below the required RMSE, as long as the EEVPD-based bound holds. We note that the results in Section 7 do not even exhibit a significant decrease in efficiency compared to the grid search results and all unadjusted samplers converge to the desired accuracy.

D Experiment Details

D.1 Benchmark Problems

The following benchmarks are used:

- A Standard Gaussian in $d = 100$.
- An ill-conditioned Gaussian in $d = 100$ and condition number $\kappa = 1000$. The eigenvalues of the covariance matrix are equally spaced in log.
- A Rosenbrock function with $Q = 0.1$ from Grumitt et al. (2022). This is a banana shaped target in two dimensions, see Figure 8 in Robnik et al. (2024). We use a product of 18 independent copies, so the total dimension of the target is 36. An exception is Figure 2 where we study the performance as a function of the number of copies.

- A Funnel problem in 101 dimensions: this is a hierarchical Bayesian model with a funnel shape (Grumitt et al., 2022). The goal is to infer the hierarchical parameter θ and the latent variables $\{z_i\}_{i=1}^{100}$, given the noisy observations $y_i \sim \mathcal{N}(z_i, 1)$. The prior is Neal's funnel (Neal, 2011): $\theta \sim \mathcal{N}(0, 3)$, $z_i \sim \mathcal{N}(0, e^{\theta/2})$. We set $\theta_{\text{true}} = 0$ and generate the data with the generative process described above. Given this data we then sample from the posterior for θ and $\{z_i\}_{i=1}^{100}$.
- A Brownian motion example from the Inference Gym (Sountsov et al., 2020), where it is named `BrownianMotionUnknownScalesMissingMiddleObservations`. This is a 32 dimensional hierarchical Bayesian model where Brownian motion with unknown innovation noise and measurement noise is fitted to the noisy and partially missing data.
- The German Credit model, also known as Sparse logistic regression (`GermanCredit-NumericSparseLogisticRegression`) is a 51-dimensional Bayesian hierarchical model, where logistic regression is used to model the approval of the credit based on the information about the applicant.
- An Item Response theory model (`SyntheticItemResponseTheory`), which is a 501-dimensional hierarchical problem where students' ability is inferred, given the test results.
- Stochastic Volatility is a 2429-dimensional hierarchical non-Gaussian random walk fit to the S&P500 returns data, adapted from NumPyro (Phan et al., 2019).
- Lattice ϕ^4 field theory: see section D.3 for details. We take the number of points in the lattice in each of the two directions to be 1024, so that the problem has $1024^2 = 1048576$ dimensions.

The ground truth covariance matrix for the first two problems is known exactly. For the Rosenbrock function, we compute it by drawing exact samples from the posterior. For the other problems, we obtain the ground truth by running very long NUTS chains.

We note that even though these benchmark problems are the same as in (Hoffman & Sountsov, 2022) and the evaluation metric is the same, the results cannot be compared, because Hoffman & Sountsov (2022) run a large number of chains in parallel (4096 chains) and combines the samples at each fixed time to compute the expectation values. We on the other hand are interested in the more standard regime where one chain is sequentially run for a longer time and samples at different times are combined to compute the expectation values. This of course results in longer time to convergence, but lower total calculation cost. Furthermore NUTS in (Hoffman & Sountsov, 2022) was adapted to the many-short-chains regime and is therefore not the same algorithm as a more standard NUTS used here.

D.2 Covariance matrix bias

We here provide the results complementary to Section 7 for the b_{cov} metric. We do not show it for the higher dimensional problems (Item response and Stochastic Volatility), because their covariance matrix is impractically large. Qualitatively the conclusions are as in Section 7:

- Unadjusted versions of the algorithms tend to perform better than their adjusted counterparts.
- EEVPD based tuning yields close to optimal performance when compared to grid search. An exception is the Rosenbrock distribution, where the conservative choice of EEVPD yields some decrease in performance, but the unadjusted algorithm still performs better than the adjusted one.
- MCLMC performs better than LMC.

A notable difference is that all samplers need a larger number of gradient evaluations here, because b_{cov} is a more stringent metric than b_{avg}.

D.3 Lattice ϕ^4 field theory

This is a classic interacting lattice field theory on the plane. We will adopt its treatment from Robnik & Seljak (2024). In a continuum, the ϕ scalar function $\phi(x, y)$ on the plane. The probability density

	aLMC *grid search*	uLMC *EEVPD*	aMCLMC *acc. rate*	uMCLMC *EEVPD*	uLMC *grid search*	NUTS *acc. rate*
Standard Gaussian	79,841	**64,254**	43,389	**26,032**	65,604	240,456
Rosenbrock	659,359	**415,988**	540866	**348,048**	271,825	852,135
Brownian	**93,787**	112,242	76,931	**41,838**	73,632	146,333
German Credit	462,843	**380,569**	462,770	**249,674**	306904	756,792

Table 3: Number of gradient calls needed to get b^2_{cov} below 0.01.

on the field configuration space is proportional to $e^{-S[\phi]}$, where the action is

$$S[\phi(x,y)] = \int \big(-\phi\,\partial^2\phi + m^2\phi^2 + \lambda\phi^4\big)dxdy. \tag{21}$$

The parameters of the theory are the mass $m^2 < 0$, and the quartic coupling $\lambda > 0$.

The system is interesting as it exhibits spontaneous symmetry breaking, and belongs to the same universality class as the Ising model. It does not admit analytic solutions due to the quartic interaction term. It is numerically solved by discretizing the field on a lattice and making the lattice spacing as fine as possible (Gattringer & Lang, 2009). The discretized field is specified by a vector of values on a lattice ϕ_{ij} for $i, j = 1, 2, \ldots L$. The dimensionality of the configuration space is $d = L^2$. We will impose periodic boundary conditions, such that $\phi_{i,L+1} = \phi_{i1}$ and $\phi_{L+1,j} = \phi_{1j}$. The lattice action is (Vierhaus, 2010)

$$S_{\text{lat}}[\phi] = \sum_{i,j=1}^{L} 2\phi_{ij}\big(2\phi_{ij} - \phi_{i+1,j} - \phi_{i,j+1}\big) + m^2\phi_{ij}^2 + \lambda\phi_{ij}^4. \tag{22}$$

We will fix $m^2 = -4$ (which removes the diagonal terms ϕ_{ij}^2 in the action) as is common (Albergo et al., 2021; Gerdes et al., 2022). In all experiments we fix $\bar{\lambda} = 4$, where $\bar{\lambda} = L^{1/\nu}(\lambda - \lambda_C)/\lambda_C$. Here, $\nu = 1$ and $\lambda_C = 4.25$ are taken from (Vierhaus, 2010). This choice ensures that all of the experiments are performed at approximately the same location in the phase diagram in disordered phase, with minimal dependence on the lattice size (Gerdes et al., 2022; Goldenfeld, 2018).

As in the other experiments, we take the average squared bias of the second moments as the convergence metric, $b^2_{avg} \equiv d^{-1}\sum_{k,l=1}^{L} b^2(|\widetilde{\phi}_{kl}|^2)$. The only difference is that we here consider the second moments of the Fourier transform $\widetilde{\phi}$ instead of the field itself. The Fourier transform is defined as $\widetilde{\phi}_{kl} = L^{-1}\sum_{nm=1}^{L}\phi_{nm}e^{-2\pi i(kn+lm)/L}$. We use only 2 chains for ϕ^4 at 1024^2 dimensions, since this is what fits on a GPU.

Scaling We examine the scaling of the performance with lattice size L, as shown in Figure 5. We observe that up to 10^6 dimensions, uMCLMC performance is constant, which gives rise to its outstanding performance relative to adjusted methods at 1024^2 dimensions.

D.4 Step size adaptation

We here describe the scheme for quickly adapting the step size ϵ to achieve a desired EEVPD. We verified that it indeed yields the desired EEVPD in the experiments performed in this work, but we do not provide any convergence guarantees. More generally, one can instead use a more established dual averaging algorithm (Hoffman & Gelman, 2014; Nesterov, 2009).

Suppose we did a leapfrog step with size ϵ_k and found some energy error Δ_H^k. Using only this knowledge and the scaling from Equation (4.1) for small step sizes, $\Delta_H \propto \epsilon^6$, we could estimate the optimal step size to use in the next iteration as $\epsilon_{k+1} = \xi_k^{-1/6}$ where

$$\xi_k = \frac{(\Delta_H^k)^2}{d}\frac{1}{\alpha\,\epsilon_k^6} \tag{23}$$

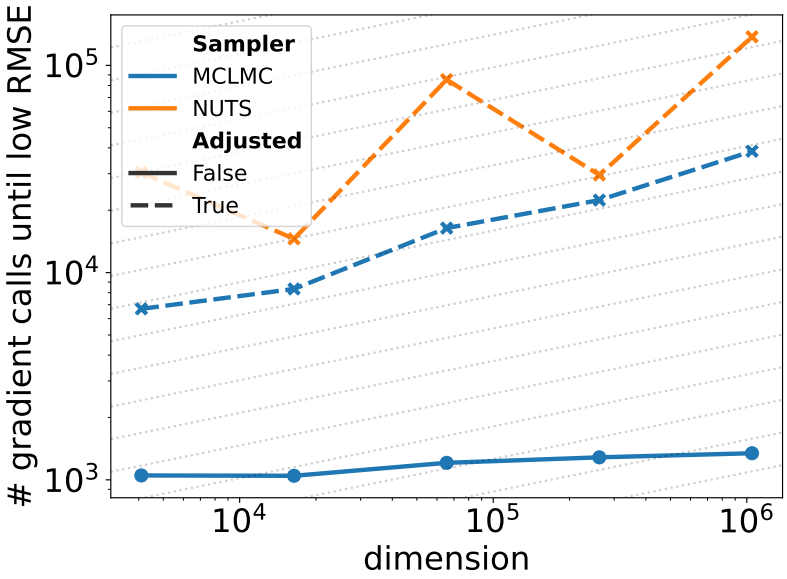

Figure 5: ϕ^4 performance scaling as the lattice size is increased. Cost of the Metropolis adjusted MCLMC and NUTS grows with the number of parameters as $d^{1/4}$ (Grey lines in the background), while unadjusted MCLMC performance stays constant.

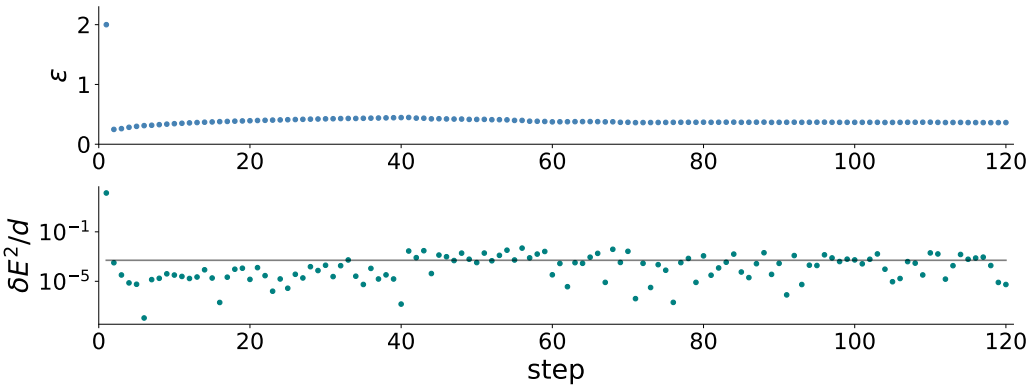

Figure 6: The step size adaptation algorithm from Section D.4, applied to the Rosenbrock target distribution in $d = 36$ with MCLMC sampler. The sequential algorithm was initialized from the standard Gaussian distribution with a random initial velocity orienation. Top: the step size as a function of leapfrog integration steps. Bottom: per dimension squared energy error for each step. The algorithm quickly converges to the targeted EEVPD = 0.001, shown with a black line.

and α is the desired EEVPD. As we do more leapfrog steps, we can improve our estimate by averaging the energy errors. Our estimate of the optimal step size is then a weighted sum:

$$\epsilon_{n+1} = \left(\frac{\sum_{k=1}^{n} w(\xi_k)\gamma^{n-k}\, \xi_k}{\sum_{k=1}^{n} w(\xi_k)\gamma^{n-k}} \right)^{-1/6}. \tag{24}$$

We have introduced two types of weights:

- The weights w parametrize our trust in the predictions from the too large and too small ϵ. We take the log-normal penalty

$$w(\xi) = \exp\left\{ -\frac{1}{2}(\log \xi)^2/\sigma_\xi^2 \right\}, \tag{25}$$

  with $\sigma_\xi = 1.5$.
- γ is the forgetting factor. It is related to the effective sample size n of the estimate (if w were constant) by $\gamma = \frac{n-1}{n+1}$. n is also the number of steps after which the weights have decayed to $e^{-2} = 0.13$. In general, we don't want n to be too small, so that EEVPD is well determined and yet not too large during the burn-in such that the initially heavily biased estimates are forgotten quickly. We find $n = 50$ to work well on all the experiments considered in this work. An example run is shown in Figure 6.

The pseudocode for the proposed algorithm is shown in 1.

Data: initial condition $(\boldsymbol{x}, \boldsymbol{u})$,
initial step size $\epsilon > 0$,
number of integration steps $N > 0$,
desired EEVPD $\alpha > 0$.
Result: step size ϵ
$A, B \leftarrow 0$;
for $n \leftarrow 0$ **to** N **do**
  $(\boldsymbol{x}, \boldsymbol{u}), \Delta E \leftarrow \Phi_\epsilon(\boldsymbol{x}, \boldsymbol{u})$;
  $\xi \leftarrow$ Equation (23) $(\Delta E, \epsilon, \alpha)$;
  $A \leftarrow A\gamma + \xi w(\xi)$;
  $B \leftarrow B\gamma + w(\xi)$;
  $\epsilon \leftarrow (A/B)^{-1/6}$;
end

Algorithm 1: Step size adaptation

D.5 Details of experiments

We use an run of NUTS to find a diagonal preconditioning matrix. Preconditioning can be viewed as reparametrization of the parameter space, so the goal here is to compare different samplers on exactly the same problem, i.e., same target, same parametrization. In practice, however, we recommend using unadjusted methods for preconditioning which is in fact the default behavior in our implementation, even when adjusted methods are later used for sampling. This is because the unadjusted methods converge significantly faster and the bias in the preconditioner is acceptable even if one desires asymptotically unbiased samples.

We determine step size by running a short warm-up chain. For unadjusted methods, we use the algorithm from D.4 to select a step size that ensures a desired EEVPD. For adjusted MCLMC we use the dual averaging algorithm (Nesterov, 2009) from Hoffman & Gelman (2014) to adapt the step size to achieve an acceptance rate of 90%.

We take the tuning steps as our burn-in, initializing the chain with the final state returned by the tuning procedure. For each model we run at least 128 chains, and take the median of the error across chains at each step. This reduces the error in the quantities in Tables 1 and 3. The errors are shown in Table 4. They are calculated by bootstrap: for a given model, we produce a set of chains (at least 128), and calculate the bias b_{avg} or b_{cov} at each step of the chain. We then resample (with replacement)

Model	metric	aLMC	uLMC	aMCLMC	uMCLMC	NUTS
Standard Gaussian	b_{avg}	0.42%	1.02%	1.51%	0.77%	0.13%
	b_{cov}	0.06%	0.32%	0.31%	0.29%	0.06%
Rosenbrock	b_{avg}	0.07%	5.69%	0.06%	2.05%	0.02%
	b_{cov}	0.03%	1.64%	0.01%	0.90%	0.01%
Brownian Motion	b_{avg}	0.37%	6.14%	0.38%	2.77%	0.16%
	b_{cov}	0.05%	4.96%	0.06%	0.49%	0.03%
German Credit	b_{avg}	0.06%	7.49%	0.10%	2.25%	0.04%
	b_{cov}	0.05%	11.58%	0.08%	0.93%	0.05%
Item Response	b_{avg}	0.62%	8.08%	1.23%	4.11%	0.44%
Stochastic Volatility	b_{avg}	0.04%	8.88%	0.06%	1.87%	0.02%

Table 4: Relative error associated with Tables 1 and 3.

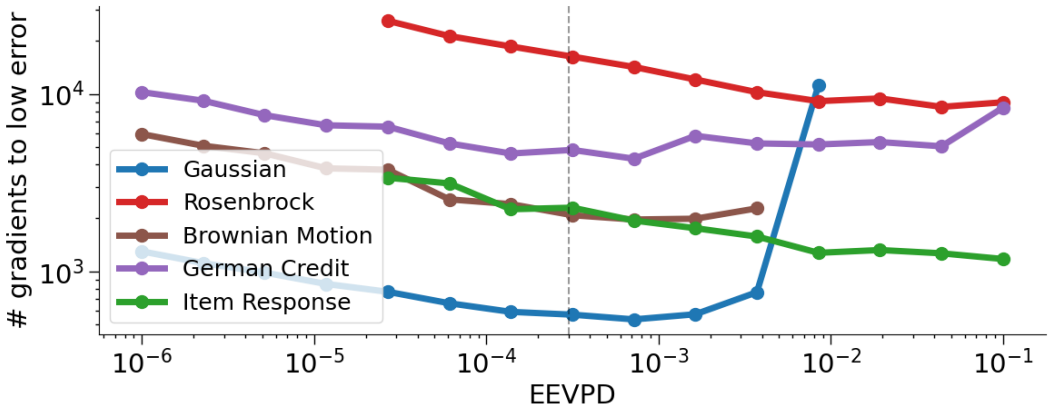

Figure 7: Performance of unadjusted LMC as a function of EEVPD (which in turn sets the step size). As can be seen, performance does not change much within a resonable range of EEVPD. The value $\text{EEVPD} = 3 \times 10^{-4}$ that we use in Section 7 is shown as a vertical dashed line.

100 times from this set, and compute our final metric (number of gradients to low bias) 100 times. We take the standard deviation of this list of length 100 as the estimate of the error and report it relatively to the values in Tables 1 and 3.

NUTS is run using the BlackJax (Cabezas et al., 2024) implementation, with the provided window adaptation scheme. For adjusted LMC (MALT), we perform a search over different values of trajectory length, and at each, choose ϵ to target an acceptance rate of 0.8.

D.6 Computing architecture

ϕ^4 field theory example was run on the NVIDIA A100 GPU (40GB). The other experiments were run on 128 CPU cores, where each core is a 2x AMD EPYC 7763 (Milan) CPU.

E Ablation study

Here, we investigate how performance of unadjusted LMC, with the tuning algorithm proposed in Section 6 varies with the EEVPD value being targeted. Figure 7 shows that our choice of 3×10^{-4} is within the safe range for problems that we consider, albeit it is somewhat conservative for the Item Response and Rosenbrock problems.