Journal of Machine Learning Research 27 (2026) 1-64 Submitted 6/25; Revised 2/26; Published 3/26

The Within-Orbit Adaptive Leapfrog No-U-Turn Sampler

Nawaf Bou-Rabee NAWAF.BOURABEE@RUTGERS.EDU
Department of Mathematical Sciences
Rutgers University
311 N 5th Street
Camden, NJ 08102, USA

Bob Carpenter BCARPENTER@FLATIRONINSTITUTE.ORG
Center for Computational Mathematics
Flatiron Institute
162 5th Ave
New York, NY 10010, USA

Tore Selland Kleppe TORE.KLEPPE@UIS.NO
Department of Mathematics and Physics
University of Stavanger
Stavanger, 4036 Stavanger, Norway

Sifan Liu SIFAN.LIU@DUKE.EDU
Department of Statistical Science
Duke University
214 Old Chemistry
Durham, NC 27708, USA

Editor: Stephan Mandt

Abstract

Locally adapting parameters within Markov chain Monte Carlo methods while preserving reversibility is notoriously difficult. The success of the No-U-Turn Sampler (NUTS) largely stems from its clever local adaptation of the integration time in Hamiltonian Monte Carlo via a geometric U-turn condition. However, posterior distributions frequently exhibit multiscale geometries with extreme variations in scale, making it necessary to also adapt the leapfrog integrator's step size locally and dynamically. Despite its practical importance, this problem has remained largely open since the introduction of NUTS by Hoffman and Gelman (2014). To address this issue, we introduce the Within-Orbit Adaptive Leapfrog No-U-Turn Sampler (WALNUTS), a generalization of NUTS that adapts the leapfrog step size at fixed intervals of simulated time as the orbit evolves. At each interval, the algorithm selects the largest step size from a dyadic schedule that keeps the energy error below a user-specified threshold. Like NUTS, WALNUTS employs biased progressive state selection to favor states with positions that are further from the initial point along the orbit. Empirical evaluations on multiscale target distributions, including Neal's funnel and the Stock-Watson stochastic volatility time-series model, demonstrate that WALNUTS achieves substantial improvements in sampling efficiency and robustness compared to NUTS.

Keywords: Hamiltonian Monte Carlo, No-U-Turn Sampler, adaptive step size, Markov chain Monte Carlo

1. Introduction and Motivation

Hamiltonian Monte Carlo and the Role of Discretization. Hamiltonian Monte Carlo (HMC) is a Markov chain Monte Carlo (MCMC) method for sampling from probability distributions with continuously differentiable densities $\mu : \mathbb{R}^d \to \mathbb{R}$ that are known up to normalization (Duane et al., 1987; Neal, 2011). HMC generates proposals by simulating a measure-preserving flow on an extended phase space. This is done by introducing an auxiliary momentum variable $\rho \in \mathbb{R}^d$ and a positive-definite mass matrix $M \in \mathbb{R}^{d\times d}$, and evolving the system according to the Hamiltonian dynamics corresponding to the Hamiltonian function $H(\theta, \rho) = -\log \mu(\theta) + \frac{1}{2}\rho^\top M^{-1}\rho$. At all times, both forward and backward, the resulting Hamiltonian flow is time-reversible, exactly conserves phase-space volume and, in the absence of discretization error, the total energy. Hence, this flow preserves the probability measure on $\mathbb{R}^{2d}$ with density proportional to $e^{-H(\theta,\rho)}$.

This measure-preserving property makes Hamiltonian flows a natural tool for constructing MCMC methods. However, in most cases, the Hamiltonian flow cannot be computed analytically and must be approximated numerically, most commonly using the leapfrog integrator (Hairer et al., 2010; Bou-Rabee and Sanz-Serna, 2018). Leapfrog is a time-reversible method that preserves volume exactly but only approximately conserves energy. These geometric properties make it a natural choice for constructing Metropolis proposals, as we briefly review in Section 2.4. Crucially, however, the leapfrog algorithm is only conditionally stable. The numerical trajectory may diverge if the step size is too large relative to the local curvature of the log target density, $\log \mu$, as quantified by the largest eigenvalue in magnitude of its Hessian (Bou-Rabee and Vanden-Eijnden, 2012). This imposes constraints on step size selection, and highlights the importance of carefully discretizing the Hamiltonian dynamics to ensure efficient sampling.

The No-U-Turn Sampler. Two tuning decisions are especially critical to the performance of HMC,

- how far to integrate Hamilton's equations (the integration time), and
- how finely to discretize the dynamics (the step size).

The No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014) addresses the first question by eliminating the need to tune the integration time a priori. Instead, NUTS adaptively constructs an orbit by simulating Hamiltonian dynamics in both forward and backward time directions. The orbit is built recursively using a doubling procedure: at each iteration, a direction (forward or backward) is chosen uniformly at random, and the orbit is extended in that direction by doubling its length. The result is a balanced binary tree of candidate states, where internal nodes represent recursive doubling stages and leaves correspond to individual states.

The iteration continues until a U-turn condition is met (see Figure 1). This condition checks whether the orbit has begun to reverse direction by evaluating inner products between momentum and position vectors across pairs of states, as we briefly review in Section 2.7. To ensure reversibility, this condition is applied recursively across all pairs of left and right subtrees in the binary tree built during orbit expansion. This construction ensures that the probability of any given orbit is independent of the starting point within the orbit.

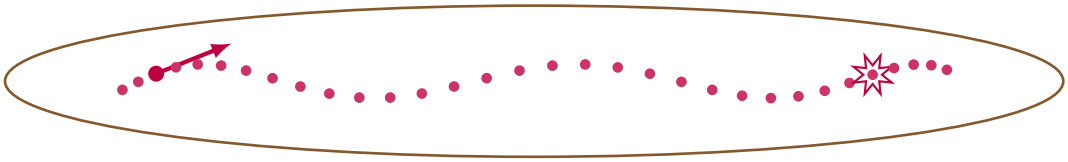


Figure 1: The line indicates a one-standard-deviation contour of a bivariate Gaussian target. The dots represent an orbit of consecutive leapfrog iterates in position space. The large dot and arrow mark the initial position and momentum, while the star marks the next state selected by the sampler. In this illustration, the U-turn condition in NUTS successfully detects the local scale of the target and terminates the trajectory accordingly.

Once the orbit is constructed, a proposal is sampled from the candidate states using a biased progressive sampling scheme that preserves detailed balance (Hoffman and Gelman, 2014; Betancourt, 2017). By adapting the integration time to the local geometry of the target distribution, NUTS achieves robust performance without requiring manual tuning. This local adaptivity of integration time has made NUTS one of the most widely used MCMC methods for sampling from distributions with continuously differentiable densities (Hoffman and Gelman, 2014; Betancourt, 2017; Carpenter et al., 2016; Bou-Rabee and Oberdörster, 2024a).

The Limitations of a Fixed Step Size. While NUTS effectively eliminates the need to tune the integration time by adapting the orbit length to the local geometry, it still uses a single, fixed step size throughout the entire sampling process. This global step size can be limiting in multiscale settings, where the local curvature of the target distribution may vary dramatically across regions. Because the leapfrog integrator is only conditionally stable, a step size that performs well in flat regions may lead to numerical divergence or large energy errors in regions of high curvature. Conversely, calibrating the step size to accommodate the worst-case curvature results in unnecessarily fine discretization and inefficient exploration in flatter regions.

Recent work has introduced orbit-level adaptive step-size variants of NUTS that adjust the leapfrog step size globally for each orbit while preserving detailed balance (Bou-Rabee et al., 2025). However, these methods do not address the need for finer-grained step size adaptation within each orbit.

Introducing WALNUTS. To address this limitation, we introduce WALNUTS (Within-orbit Adaptive Leapfrog No-U-Turn Sampler), a generalization of NUTS that incorporates local step size adaptation within each orbit. The key idea is to allow the leapfrog step size to adapt dynamically to local geometric features of the target distribution, rather than relying on a single, globally calibrated step size. WALNUTS selects the step size independently at each integration step of the orbit, ensuring numerical stability and controlling energy error as the orbit moves through regions of varying curvature. This enables the sampler to integrate stably through high-curvature regions while progressing efficiently through flatter ones without sacrificing detailed balance or requiring manual step size tuning.

To formalize this adaptivity, we distinguish between macro steps and micro steps. A macro step is the unit of simulated time over which candidate states are generated and added to the orbit. Each macro step is implemented by applying the leapfrog integrator over a sequence of micro steps. The micro step size is chosen from a dyadic schedule so that the total energy error over the macro step remains below a user-specified threshold. At each macro step, WALNUTS searches for the coarsest micro step size that satisfies this energy-based criterion. This allows different segments of the orbit to be integrated at different resolutions, depending on local curvature and stability requirements.

The resulting orbit consists of a sequence of states lying on a fixed macro grid, with each macro step integrated using a locally adapted micro step size. Importantly, although the micro step size may vary from one macro step to the next, the map from one candidate state to the next remains volume-preserving: an essential property for maintaining reversibility and correctness, as detailed in Section 3.4. Once the full orbit is constructed, WALNUTS samples a single state from among the macro steps using the same biased progressive scheme as NUTS, which preserves detailed balance and ensures correctness. This local adaptivity allows the algorithm to remain both stable in regions of high curvature and efficient in flatter regions without requiring a globally tuned step size.

Visualization of a WALNUTS Transition Step. Figure 2 illustrates the mechanics of a WALNUTS transition. Each panel shows a different realization, all starting from the same initial condition, in a 2D slice of Neal's funnel distribution. The orbit is color-coded according to the locally selected micro step size, with red indicating larger values and blue smaller ones. These gradients visually demonstrate how WALNUTS adjusts its numerical resolution in response to local curvature. Insets display the corresponding weight profiles used during biased progressive sampling. These weights incorporate not only the target density at each state, as in standard NUTS, but also account for the possibility that the reverse trajectory may follow a different step-size schedule, ensuring that detailed balance is preserved under local step-size adaptivity.

General Applicability and Theoretical Considerations. Although we present adaptive leapfrog integration in the context of NUTS, the underlying idea is more broadly applicable. The same energy-controlled leapfrog step size adaptation can be incorporated, with minimal modification, into other Hamiltonian-based samplers such as standard HMC, multinomial HMC, and other reversible variants, and also non-reversible variants such as the Metropolis-adjusted Kinetic Langevin Algorithm (Bou-Rabee and Oberdörster, 2024b) and generalized HMC (Horowitz, 1991; Turok et al., 2025). The idea may also be relevant for related Hamiltonian descent optimization methods (Fu and Wibisono, 2025).

More generally, while it is well known in geometric integration theory that symplectic integrators with variable step sizes may fail to preserve energy over very long trajectories (Stofer, 1988; Calvo and Sanz-Serna, 1993), this limitation is less restrictive in the context of MCMC. The goals of MCMC differ fundamentally: rather than requiring long-time energy conservation, what matters is the preservation of the correct invariant distribution. For this purpose, it is sufficient that, conditional on the locally chosen micro step size, the sequence of leapfrog micro steps used to perform a single macro integration step remains volume-preserving. This property ensures that the overall transition kernel is reversible with respect to the target distribution, as established in Section 3.4.

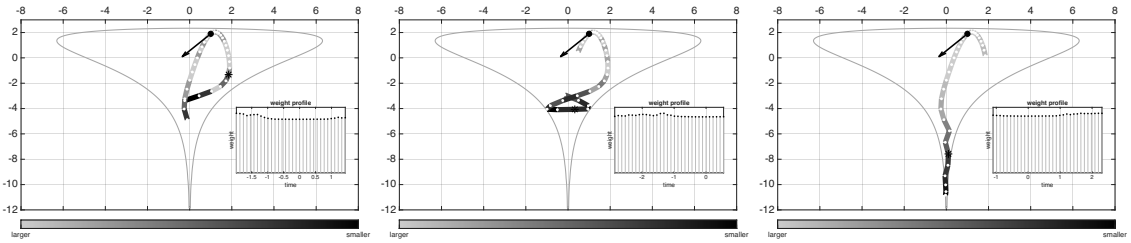


Figure 2: Three stochastic realizations of a WALNUTS transition from the same initial point in a 2D slice of Neal's funnel, each generated using a different random seed. The black dot and arrow indicate the initial position and velocity, and the asterisk marks the selected next state. Each panel shows a full orbit represented by a sequence of white dots. Each macro step is shaded in grayscale according to the micro step size required to keep the energy error below a user-specified threshold, with lighter segments corresponding to larger micro step sizes and darker segments to smaller ones. Insets show the unnormalized sampling weights $w_k \propto \exp(-H(\theta_k, \rho_k))\,\alpha_k$ used in the biased progressive state-selection step, where α_k accounts for the Metropolis acceptance correction induced by within-orbit adaptivity. Differences between panels reflect the stochastic nature of state selection and orbit construction forward and backward in time.

Main Contributions. This paper introduces WALNUTS, a step-size adaptive generalization of the No-U-Turn Sampler (NUTS) that dynamically adjusts the leapfrog step size within each macro step of an orbit. The main contributions of the paper are:

- **Within-orbit step size adaptation.** We propose a principled framework for dynamic step size selection within Hamiltonian trajectories. At each integration interval (macro step), WALNUTS chooses the coarsest step size from a dyadic schedule that keeps the accumulated energy error below a user-specified threshold. This enables stable integration through high-curvature regions while accelerating through flatter regions. In contrast to global or orbit-level adaptation, this within-orbit mechanism provides the fine-grained control needed to handle multiscale targets effectively. To our knowledge, this level of local adaptivity has not previously been achieved in reversible HMC-based samplers.

- **Reversibility under local adaptivity.** We establish that WALNUTS satisfies detailed balance with respect to the target distribution (Theorem 22). To do so, we lift the algorithm to an extended state space $\mathbb{A}$ and introduce a carefully designed involution $\Psi : \mathbb{A} \to \mathbb{A}$, i.e., $\Psi \circ \Psi$ is the identity map (see (21)). As shown in Lemma 17, Ψ preserves the reference measure ζ on $\mathbb{A}$. Lemma 21 further shows that the extended target density is invariant under Ψ. Together, these properties imply that WALNUTS satisfies detailed balance with respect to the extended target on the lifted state space, and, after marginalization, detailed balance with respect to the original target distribution, without requiring a Metropolis correction, analogously to standard NUTS.

This is a subtle and nontrivial result: while individual leapfrog steps are time-reversible and volume-preserving, their composition with locally varying step sizes typically breaks both properties. A key contribution of our analysis is the careful design of the measure-preserving involution Ψ, which situates WALNUTS firmly within the auxiliary-variable-and-involution framework developed in recent work (Andrieu et al., 2020; Glatt-Holtz et al., 2023; Bou-Rabee et al., 2024; Glatt-Holtz et al., 2024), and provides a rigorous foundation for incorporating within-orbit local adaptivity into Hamiltonian-based MCMC samplers.

- **Empirical improvements across multiscale targets.** We benchmark WALNUTS on challenging posterior geometries, including Neal's funnel and the Stock-Watson time-series model, and demonstrate substantial gains in sampling efficiency over standard NUTS. The results highlight that WALNUTS adapts effectively to local curvature without interfering with the integration-time adaptivity central to NUTS. Moreover, WALNUTS successfully fits complex hierarchical Bayesian models that standard NUTS fails to handle reliably.

- **Accessible and efficient implementations.** To promote adoption and reproducibility, we provide both a pedagogically transparent implementation in Python that mirrors the pseudocode in the manuscript, and a memory-efficient C++ version optimized for large-scale applications. These complementary implementations are designed to support both practical deployment and future research. The results and plots can be reproduced from these implementations, which are available from GitHub under a permissive open-source license.[1]

Tuning Considerations. WALNUTS requires three user-specified parameters: a mass matrix, the energy error threshold that controls the micro step size selection at each macro step, and the macro step size itself, which determines the temporal spacing between states along the final orbit. Since the theoretical foundations of warmup adaptation remain underdeveloped, we explore practical strategies for selecting the energy error threshold and macro step size during an initial warmup phase in Appendix C. This phase is used solely to calibrate algorithmic parameters prior to collecting posterior samples. Tuning the mass matrix presents additional challenges, particularly when it is restricted to be diagonal for computational reasons (Kleppe, 2016; Hird and Livingstone, 2023; Tran and Kleppe, 2024). One approach to diagonal mass matrix preconditioning is by approximating the target density with a multivariate normal distribution having a diagonal covariance matrix (Margossian and Saul, 2023, 2024).

Related Work. Here we situate this work in the broader context of locally adaptive HMC methods. The AutoMALA sampler has been proposed as a self-tuning version of the Metropolis-adjusted Langevin algorithm (MALA) (Kleppe, 2016; Biron-Lattes et al., 2024), which can be viewed as a one-step variant of HMC. These methods use a forward–reverse stochastic adaptation scheme that preserves detailed balance. The more recent AutoStep sampler (Liu et al., 2024) builds on similar principles and is closely related to earlier work on the GIST sampler (Bou-Rabee et al., 2024), which provides a general framework for locally adaptive HMC (reviewed in Section 2.3).

1. The code is distributed under the MIT License at `https://github.com/bob-carpenter/walnuts`.

A different but related class of methods involves the delayed rejection algorithm (Mira, 2001; Green and Mira, 2001), which is a generalization of Metropolis-Hastings to a sequence of proposal moves that can start "bold" and become increasingly "timid" (Haario et al., 2006); i.e., if a first proposal with a bold hyper-parameter is rejected, then a second proposal with a timider hyper-parameter is attempted, and so on. The hope is that the delayed rejection probability can be made arbitrarily small with increasingly timid proposals. This strategy has been applied to adaptively select orbit-level leapfrog step sizes in HMC (Modi et al., 2023), and more recently, to adapt the leapfrog step size within each generalized HMC step (Turok et al., 2025). In both approaches, however, key tuning parameters such as path length and friction must be specified in advance and are not adapted locally.

Organization of Paper. The remainder of the paper is organized as follows. Section 2 reviews background on HMC and NUTS, introducing key concepts such as leapfrog integration, auxiliary variables, involutions, and orbits, and states a general reversibility theorem that provides the theoretical foundation for our method. Section 3 presents the WALNUTS algorithm, beginning with the construction of a variable step-size leapfrog integrator in Section 3.1, and then describing the corresponding modifications to orbit construction in Section 3.2 and integration time selection in Section 3.3. Section 3.4 establishes the reversibility of WALNUTS by verifying the conditions of the general theorem introduced in Section 2. Sections 4.2 and 4.3 evaluate the performance of WALNUTS on multivariate Gaussian targets and Neal's funnel distribution, respectively. Section 4.4 applies WALNUTS to a high-dimensional Bayesian inference problem in macroeconomics. Appendix A provides pseudocode to support conceptual understanding of the algorithm, while Appendix B outlines practical strategies for optimizing the implementation and Appendix C concludes with some approaches to parameter tuning to be applied during an initial warmup phase.

2. Preliminaries

We first provide some background on Hamiltonian Monte Carlo and its locally adaptive variants.

2.1 Hamiltonian Flow

Let μ be a given target probability measure on $\mathbb{R}^d$ with unnormalized density also denoted by μ. To sample from μ, HMC-type methods extend the state space $\mathbb{R}^d$ to phase space $\mathbb{R}^{2d}$ by introducing an auxiliary momentum variable $\rho \in \mathbb{R}^d$ and a symmetric positive-definite mass matrix $M \in \mathbb{R}^{d\times d}$. The algorithm approximates a Hamiltonian flow in this extended phase space that preserves the joint distribution $\hat{\mu} = \mu \otimes \mathcal{N}(0, M)$ with density

$$\hat{\mu}(\theta, \rho) \;\propto\; \exp(-H(\theta, \rho)) \;, \quad \text{where} \quad H(\theta, \rho) = -\log \mu(\theta) + \frac{1}{2}\rho^\top M^{-1}\rho \;. \tag{1}$$

The corresponding exact Hamiltonian flow $\varphi_t : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ maps an initial state $(\theta, \rho) \in \mathbb{R}^{2d}$ to (θ_t, ρ_t) at time $t \in \mathbb{R}$, where (θ_t, ρ_t) evolves according to the differential equations

$$\frac{d}{dt}\theta_t = M^{-1}\rho_t, \quad \frac{d}{dt}\rho_t = \nabla \log \mu(\theta_t), \quad \text{with } (\theta_0, \rho_0) = (\theta, \rho) \;. \tag{2}$$

This flow preserves both the Hamiltonian function H and Lebesgue measure on $\mathbb{R}^{2d}$. Consequently, it preserves the extended target distribution $\hat{\mu}$. By "preserves" we mean that the

Hamiltonian flow leaves $\hat{\mu}$ invariant: if the initial state (θ, ρ) is distributed according to $\hat{\mu}$, then for any time t, the evolved state $\varphi_t(\theta, \rho)$ is also distributed according to $\hat{\mu}$.

2.2 Fixed-Step-Size Leapfrog Integrator

Because the exact Hamiltonian flow in (2) cannot be computed in closed form for a general target μ, it is approximated numerically. HMC-type methods almost always use the leapfrog integrator for this purpose, as it is computationally efficient and preserves key geometric properties of the Hamiltonian flow.

Fix a step size $h > 0$. The leapfrog integrator updates (θ, ρ) via the standard *kick-drift-kick* sequence

$$\underbrace{\rho^{(1/2)} \;=\; \rho + \frac{h}{2}\nabla \log \mu(\theta),}_{\text{kick}} \qquad \underbrace{\theta' \;=\; \theta + hM^{-1}\rho^{(1/2)},}_{\text{drift}} \qquad \underbrace{\rho' \;=\; \rho^{(1/2)} + \frac{h}{2}\nabla \log \mu(\theta')\,.}_{\text{kick}}$$

Equivalently, this update can be written compactly as a map $\Phi_h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ that updates a state (θ, ρ) to $(\theta', \rho') = \Phi_h(\theta, \rho)$ according to

$$\theta' \;=\; \theta + M^{-1}\left(h\rho + \frac{h^2}{2}\nabla \log \mu(\theta)\right)\,, \quad \rho' \;=\; \rho + \frac{h}{2}\left(\nabla \log \mu(\theta) + \nabla \log \mu(\theta')\right). \tag{3}$$

Besides being explicit, the leapfrog integrator has several important properties.

- **Volume preservation.** The map preserves phase-space volume, i.e., $|\det(D\Phi_h)| \equiv 1$, where $D\Phi_h$ denotes the Jacobian of Φ_h.
- **Symmetry.** The inverse map satisfies $\Phi_h^{-1} = \Phi_{-h}$.
- **Time reversibility.** The inverse map satisfies

  $$\Phi_h^{-1} = \mathcal{F} \circ \Phi_h \circ \mathcal{F}$$

  where $\mathcal{F} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is the momentum flip involution defined by $\mathcal{F}(\theta, \rho) = (\theta, -\rho)$ for $(\theta, \rho) \in \mathbb{R}^{2d}$. In other words, applying a momentum flip before and after the integrator yields the inverse map.
- **Energy error.** The energy is not exactly conserved; in general, $(H \circ \Phi_h - H) \not\equiv 0$.

For any $L \in \mathbb{Z}$, the L-step leapfrog map $\Phi_h^L : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ allows for both forward ($L > 0$) and backward ($L < 0$) integration steps, and is defined recursively as

$$\Phi_h^{L+1} = \Phi_h \circ \Phi_h^L, \qquad \Phi_h^{L-1} = \Phi_h^{-1} \circ \Phi_h^L,$$

where Φ_h^0 is the identity map.

If $L > 0$, Φ_h^L corresponds to L forward steps of size h. If $L < 0$, backward integration corresponds to applying the inverse map $|L|$ times. By symmetry of the leapfrog integrator, $\Phi_h^{-1} = \Phi_{-h}$, so backward integration may equivalently be interpreted as $|L|$ forward steps with negative step size:

$$\Phi_h^L = \left(\Phi_h^{-1}\right)^{|L|} = \Phi_{-h}^{|L|} \qquad (L < 0).$$

Moreover, the leapfrog integrator is time-reversible: $\Phi_h^{-1} = \mathcal{F} \circ \Phi_h \circ \mathcal{F}$. Iterating this identity L times yields, for all $L \geq 1$,

$$\Phi_h^{-L} = \mathcal{F} \circ \Phi_h^L \circ \mathcal{F}.$$

Combining symmetry with time reversibility therefore gives

$$\Phi_h^{-L} = \Phi_{-h}^{L} = \mathcal{F} \circ \Phi_h^L \circ \mathcal{F}.$$

Thus, backward integration can equivalently be realized by: (i) iterating the inverse map, (ii) using negative step sizes, or (iii) applying a momentum flip before and after the forward map.

These relationships are summarized by the commutative diagram, which holds for all $L \in \mathbb{Z}$:

$$\begin{array}{ccc} (\theta, \rho) & \xrightarrow{\Phi_h^{-L} = \Phi_{-h}^{L}} & (\theta', \rho') \\ {\scriptstyle \mathcal{F}}\downarrow & & \uparrow{\scriptstyle \mathcal{F}} \\ (\theta, -\rho) & \xrightarrow[\Phi_h^L]{} & (\theta', -\rho') \end{array}$$

It is important to note, however, that the composition of time-reversible maps is not necessarily time-reversible; see (Bou-Rabee and Sanz-Serna, 2018, Theorem 4.2). This observation becomes relevant later when we consider compositions of variable step-size leapfrog integrators: although each integrator in the composition is individually time-reversible, their composition generally is not.

2.3 Gibbs Self-Tuning for Locally Adaptive HMC

A general and flexible strategy for constructing Markov chains that are reversible with respect to a given target distribution μ is to augment the state space with an auxiliary variable $v \in \mathbb{V}$, apply a measure-preserving involution on the augmented space $\mathbb{A} = \mathbb{R}^d \times \mathbb{V}$, and incorporate a Metropolis–Hastings correction. This auxiliary-variable-and-involution strategy underlies many advanced MCMC methods and has been formalized in recent literature (Andrieu et al., 2020; Neklyudov et al., 2020; Glatt-Holtz et al., 2023, 2024).

More recently, this strategy was extended to a broad class of locally adaptive HMC samplers through the GIST (Gibbs Self-Tuning) framework (Bou-Rabee et al., 2024). GIST constructs adaptive samplers by Gibbs sampling the HMC algorithm's tuning parameters conditionally on the current state, such as path length, step size, and mass matrix. This unifying framework includes randomized HMC (Bou-Rabee and Sanz-Serna, 2017; Bou-Rabee and Eberle, 2022; Kleppe, 2022), multinomial HMC (Betancourt, 2017; Xu et al., 2021), the No-U-Turn Sampler (Hoffman and Gelman, 2014; Betancourt, 2017), and the Apogee to Apogee Path Sampler (Sherlock et al., 2023) as special cases.

We now describe a single transition step of a GIST sampler. Let $\mathbb{V}$ denote the auxiliary variable space, and let $p_a(v \mid \theta)$ be a conditional probability density on $\mathbb{V}$ given $\theta \in \mathbb{R}^d$. In the locally adaptive HMC samplers we will study, v typically consists of the momentum variable ρ along with additional tuning parameters. The extended target density on the

augmented space $\mathbb{A} = \mathbb{R}^d \times \mathbb{V}$ is defined by

$$\hat{\mu}(\theta, v) \;=\; \mu(\theta)\; p_a(v \mid \theta)\;,$$

which is interpreted as a density with respect to a reference measure ζ on $\mathbb{A}$. This reference measure specifies how we assign "volume" or "weight" to regions of the augmented state space. For the original state variable $\theta \in \mathbb{R}^d$, we use the standard Lebesgue measure. For the auxiliary variable $v \in \mathbb{V}$, we use a measure which may be the counting measure (if v is discrete) or the Lebesgue measure (if v is continuous).

Given the conditional density $p_a(v \mid \theta)$, the extended density $\hat{\mu}(\theta, v)$, an involution $\Psi : \mathbb{A} \to \mathbb{A}$ and the current state $\theta_n \in \mathbb{R}^d$, a transition step of a GIST sampler computes an updated state $\theta_{n+1} \in \mathbb{R}^d$ as follows:

1. **Auxiliary variable refreshment.** Sample $v_n \sim p_a(\cdot \mid \theta_n)$.
2. **Involution-based proposal.** Compute a proposal $(\tilde{\theta}_{n+1}, \tilde{v}_{n+1}) = \Psi(\theta_n, v_n)$.
3. **Metropolis correction.** Accept the proposal with probability

$$\alpha(\theta_n, v_n) = \min\left(1, \frac{\hat{\mu}(\tilde{\theta}_{n+1}, \tilde{v}_{n+1})}{\hat{\mu}(\theta_n, v_n)}\right)$$

and set

$$\theta_{n+1} = \begin{cases} \tilde{\theta}_n & \text{with probability } \alpha(\theta_n, v_n), \\ \theta_n & \text{otherwise.} \end{cases}$$

Consider the Markov chain on $\mathbb{R}^d$ obtained by iterating this GIST transition step from a given initial distribution. Since the auxiliary variables are fully resampled in each step and discarded after they are used, the marginal chain on θ inherits reversibility with respect to μ, as formalized below.

Theorem 1 ((Bou-Rabee et al., 2024)) *Suppose $\Psi : \mathbb{A} \to \mathbb{A}$ is an involution (i.e., $\Psi^2 = \mathrm{id}$) and preserves the reference measure ζ on $\mathbb{A}$. Then, the Markov chain defined by the GIST sampler is reversible with respect to μ.*

In the following subsections, we illustrate how this GIST framework gives rise to increasingly sophisticated samplers, beginning with HMC with a fixed integration time, followed by biased progressive HMC with a randomized integration time, and culminating in the No-U-Turn Sampler.

2.4 Fixed-Integration-Time Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) with a fixed integration time not only fits naturally into the auxiliary-variable framework based on measure-preserving involutions, but serves as a canonical example of this framework. In this setting, the auxiliary variable is a momentum vector $\rho \in \mathbb{R}^d$ with conditional density

$$p_a(\rho \mid \theta) = \mathcal{N}(\rho \mid 0, M) = \frac{1}{(2\pi)^{d/2} \det(M)^{1/2}} \exp\left(-\tfrac{1}{2}\rho^\top M^{-1} \rho\right),$$

corresponding to a Gaussian distribution with mean zero and covariance matrix M, independent of θ. The augmented space is $\mathbb{A} = \mathbb{R}^{2d}$, the reference measure ζ is the standard Lebesgue measure on $\mathbb{R}^{2d}$, and the augmented target distribution is given by (1).

The involution $\Psi : \mathbb{A} \to \mathbb{A}$ is defined by simulating Hamiltonian dynamics using $i \in \mathbb{N}$ leapfrog steps with a fixed step size $h > 0$, followed by a momentum flip:

$$\Psi = \mathcal{F} \circ \Phi_h^i,$$

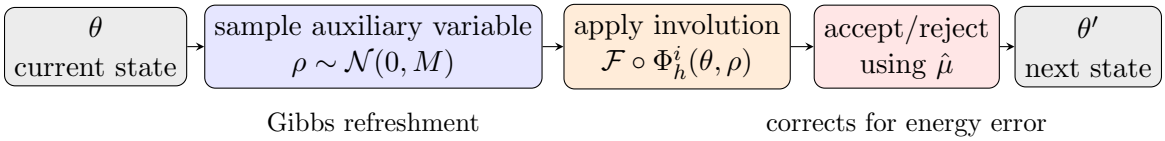
where Φ_h^i is the i-step leapfrog integrator for the Hamiltonian in (1) with mass matrix M, and $\mathcal{F}(\theta, \rho) = (\theta, -\rho)$ denotes the momentum-flip map. This composition defines a volume-preserving (i.e., ζ-preserving) involution on the augmented space $\mathbb{A}$.

Thus, HMC with fixed integration time corresponds to the following instance of a GIST sampler. The steps of this sampler are summarized below and illustrated schematically in the diagram that follows.

1. **Auxiliary variable refreshment.** Sample $\rho_n \sim \mathcal{N}(0, M)$.
2. **Involution-based proposal.** Compute $(\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1}) = \Psi(\theta_n, \rho_n) = \mathcal{F} \circ \Phi_h^i(\theta_n, \rho_n)$.
3. **Metropolis correction.** Accept the proposal with probability

$$\begin{aligned} \alpha\Big((\theta_n, \rho_n), (\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})\Big) &= \min\left(1,\ \frac{\hat{\mu}(\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})}{\hat{\mu}(\theta_n, \rho_n)}\right) \\ &= \min\Big(1,\ \exp\Big(H(\theta_n, \rho_n) - H(\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})\Big)\Big). \end{aligned}$$

After the accept/reject step, the auxiliary variable is discarded. The proposal map $\Psi = \mathcal{F} \circ \Phi_h^L$ is a volume-preserving involution: it satisfies $\Psi^2 = \mathrm{id}$ and $|\det(D\Psi)| = 1$. By Theorem 1, it follows that the resulting Markov chain on θ is reversible with respect to μ.



To better understand why the Metropolis correction is necessary, note that applying Ψ simply swaps the roles of the current and proposed states without changing phase space volume. If Ψ exactly preserved the joint density (i.e., if $\hat{\mu} \circ \Psi = \hat{\mu}$), no Metropolis correction would be needed. However, because the leapfrog integrator introduces energy error, Ψ does not exactly preserve $\hat{\mu}$, and the acceptance probability serves to correct for this discrepancy.

By definition of the acceptance probability, we have

$$\hat{\mu}(\theta_n, \rho_n)\ \alpha\Big((\theta_n, \rho_n), (\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})\Big) = \hat{\mu}(\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})\ \alpha\Big((\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1}), (\theta_n, \rho_n)\Big)\ ,$$

so the product $\hat{\mu}(\theta_n, \rho_n)\ \alpha((\theta_n, \rho_n), (\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1}))$ is symmetric in the current state (θ_n, ρ_n) and the proposed state $(\tilde{\theta}_{n+1}, \tilde{\rho}_{n+1})$. This symmetry, combined with the volume-preserving and involutive nature of Ψ, ensures that HMC is reversible with respect to $\hat{\mu}$, and thus that the marginal chain on θ is reversible with respect to μ.

As we will see in the next subsection, this logic extends naturally to biased progressive HMC, where the auxiliary variables include a random integration time.

2.5 Randomized Integration Time with Biased Progressive HMC

Biased progressive HMC introduces randomness into the integration time while preserving reversibility. This is achieved through an extended set of auxiliary variables that consist of a leapfrog trajectory, called an *orbit*, and a randomly selected index that labels a point along this orbit. This index determines the next state of the chain and can be interpreted as a randomized integration time.

We introduce precise notation and structure here not only to support the proofs that follow, but also because this perspective helps clarify key ideas underlying NUTS. In particular, it provides a clear view of how NUTS operates on an extended space, where randomized integration times emerge naturally and reversibility is maintained without Metropolis-style rejections.

Let $h > 0$ be a fixed step size. Throughout this paper, "orbit" refers to a sequence of phase space points computed using the leapfrog integrator and evaluated on a uniformly spaced time grid, defining

$$t_k \;=\; kh, \quad \text{where} \quad k \in \mathbb{Z}\,.$$

Definition 2 *An* orbit $\mathcal{O}$ *is a finite ordered sequence of states in* $\mathbb{R}^{2d}$ *generated by applying the leapfrog integrator with fixed step size* $h > 0$ *to a given initial condition* $(\theta_0, \rho_0) \in \mathbb{R}^{2d}$*. It takes the form:*

$$\mathcal{O} \;=\; ((\theta_a, \rho_a), (\theta_{a+1}, \rho_{a+1}), \ldots, (\theta_b, \rho_b))\,,$$

where $a, b \in \mathbb{Z}$ *with* $a \le b$*. Each pair* $(\theta_k, \rho_k) = \Phi_h^k(\theta_0, \rho_0)$ *represents the state at time* $t_k = kh$ *obtained by applying leapfrog steps forward (if* $k > 0$*) or backward (if* $k < 0$*) in time from the initial point, for*

$$k \in a{:}b = \{a, a+1, \ldots, b\}.$$

The length *of the orbit, denoted* $|\mathcal{O}|$*, is* $b - a + 1$*.*

We define a simple operation to concatenate orbits, which will be used in orbit construction.

Definition 3 *Let* $a, b, c \in \mathbb{Z}$ *with* $a \le b < c$*. Given two orbits*

$$\mathcal{O} = ((\theta_a, \rho_a), \ldots, (\theta_b, \rho_b)) \quad \textit{and} \quad \widetilde{\mathcal{O}} = ((\theta_{b+1}, \rho_{b+1}), \ldots, (\theta_c, \rho_c))\,,$$

their concatenation*, denoted* $\mathcal{O} \odot \widetilde{\mathcal{O}}$*, is defined as*

$$\mathcal{O} \odot \widetilde{\mathcal{O}} \;=\; ((\theta_a, \rho_a), \ldots, (\theta_c, \rho_c))\,,$$

with length $|\mathcal{O} \odot \widetilde{\mathcal{O}}| = c - a + 1$*.*

Biased progressive HMC augments the state space with the following auxiliary variables:

- a momentum vector $\rho \in \mathbb{R}^d$,
- an orbit $\mathcal{O} \subset \mathbb{R}^{2d}$ generated from the initial condition (θ, ρ) via forward and/or backward integration,

- an index $i \in \mathbb{Z}$ that selects a state within $\mathcal{O}$ to serve as the next state of the Markov chain.

We consider orbits of fixed length 2^m for a fixed $m \in \mathbb{N}$. Each orbit is determined by an initial condition (θ, ρ) and the index $b \in 0{:}(2^m - 1)$ labelling its rightmost element.

The augmented state space is

$$\mathbb{A} \;:=\; \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{N} \times \mathbb{Z},$$

with each augmented state denoted by $z = (\theta, \rho, b, i) \in \mathbb{A}$. Let ζ denote the reference measure on $\mathbb{A}$, given by the product of Lebesgue measure on the continuous components and counting measure on the discrete components. The joint density on $\mathbb{A}$ takes the form

$$p_{\text{joint}}(z) \;\propto\; e^{-H(\theta,\rho)} \cdot p_{\text{orbit}}(b \mid \theta, \rho) \cdot p_{\text{index}}(i \mid \theta, \rho, b),$$

where the proportionality symbol reflects that $e^{-H(\theta,\rho)}$ may not be normalized. Here, $p_{\text{orbit}}(b \mid \theta, \rho) = \text{Unif}(b \mid 0{:}(2^m - 1))$, i.e., b is uniformly distributed over $0{:}(2^m - 1)$, and $p_{\text{index}}(i \mid \theta, \rho, b)$ is defined implicitly by the orbit construction procedure. To construct the orbit, we draw m independent Bernoulli(1/2) random variables $B_1, \ldots, B_m$, which determine the rightmost index of the orbit as

$$b = \sum_{j=1}^{m} B_j 2^{j-1} \,,$$

and the leftmost index as $a = b - 2^m + 1$. Given (θ, ρ), the orbit $\mathcal{O}$ and integration time index i are sampled through the following recursive procedure.

1. **Initialization.** Start with the singleton orbit $\mathcal{O}_0 = ((\theta_0, \rho_0))$ where $(\theta_0, \rho_0) = (\theta, \rho)$, and set $a_0 = b_0 = i_0 = 0$. Sample m i.i.d. Bernoulli(1/2) random variables $B = (B_1, \ldots, B_m)$.

2. **Recursive construction.** For each $k = 0, 1, \ldots, m-1$:

   (a) **Current orbit.** Let $\mathcal{O}_k$ be the current orbit with index range $a_k{:}b_k$.

   (b) **Extension.** Build an extension orbit $\mathcal{O}_k^{\text{ext}}$ using leapfrog steps over the index range $a_k^{\text{ext}}{:}b_k^{\text{ext}}$ where

$$a_k^{\text{ext}} = \begin{cases} b_k + 1 & \text{if } B_{k+1} = 1, \\ a_k - 2^k & \text{if } B_{k+1} = 0, \end{cases} \qquad b_k^{\text{ext}} = \begin{cases} b_k + 2^k & \text{if } B_{k+1} = 1, \\ a_k - 1 & \text{if } B_{k+1} = 0. \end{cases}$$

   (c) **Proposal index.** Sample i_k^{ext} from a categorical distribution over $a_k^{\text{ext}}{:}b_k^{\text{ext}}$ with probabilities proportional to the weights $w_j \propto e^{-H(\theta_j,\rho_j)}$ where $j \in a_k^{\text{ext}}{:}b_k^{\text{ext}}$.

   (d) **Metropolis acceptance.** Accept i_k^{ext} with probability:

$$\alpha((a_k, b_k), (a_k^{\text{ext}}, b_k^{\text{ext}})) = \min\left(1, \frac{\sum_{j \in a_k^{\text{ext}}:b_k^{\text{ext}}} e^{-H(\theta_j,\rho_j)}}{\sum_{j \in a_k:b_k} e^{-H(\theta_j,\rho_j)}}\right) . \tag{4}$$

   If accepted, $i_{k+1} = i_k^{\text{ext}}$; otherwise, $i_{k+1} = i_k$.

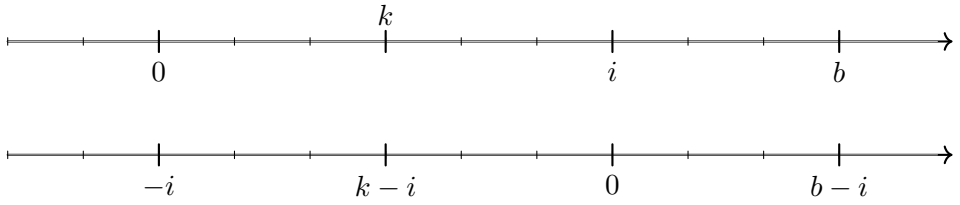


Figure 3: Re-indexing under Ψ: every index k is relabeled as $k - i$, so that i becomes the new origin while the underlying orbit remains unchanged.

(e) **Update orbit.** Concatenate to form the next orbit,

$$\mathcal{O}_{k+1} = \begin{cases} \mathcal{O}_k \odot \mathcal{O}_k^{\text{ext}} & \text{if } B_{k+1} = 1, \\ \mathcal{O}_k^{\text{ext}} \odot \mathcal{O}_k & \text{if } B_{k+1} = 0. \end{cases}$$

Accordingly, the index range updates as

$$(a_{k+1}, b_{k+1}) = \begin{cases} (a_k,\, b_k^{\text{ext}}), & \text{if } B_{k+1} = 1, \\ (a_k^{\text{ext}},\, b_k), & \text{if } B_{k+1} = 0. \end{cases}$$

3. **Output.** Return the final orbit $\mathcal{O}$ of length 2^m, the final rightmost index $b = b_m$, and the selected index $i = i_m$.

Reversibility of biased progressive HMC follows from three simple identities, which we state and prove next: (i) Ψ is a ζ-preserving involution, (ii) the orbit kernel is invariant under re-indexing, and (iii) the index kernel satisfies a detailed balance identity.

Lemma 4 (Involution) *The mapping*

$$\Psi(\theta, \rho, b, i) = (\Phi_h^i(\theta, \rho),\, b - i,\, -i)$$

is a ζ-preserving involution on $\mathbb{A}$.

The mapping Ψ acts by shifting the origin of the orbit from index 0 to index i, reversing the direction of traversal. This re-indexing is illustrated in Figure 3.

Proof Applying Ψ twice gives

$$\Psi^2(\theta, \rho, b, i) = \big(\Phi_h^{-i}(\Phi_h^i(\theta, \rho)),\, b,\, i\big) = (\theta, \rho, b, i),$$

since Φ_h^{-i} is the inverse of Φ_h^i. Thus Ψ is an involution. Leapfrog preserves Lebesgue measure and $(b, i) \mapsto (b - i, -i)$ preserves counting measure, so Ψ preserves ζ. ■

Lemma 5 (Orbit shift identity) *For all (θ, ρ, b, i),*

$$p_{\text{orbit}}(b \mid \theta, \rho) = p_{\text{orbit}}(b - i \mid \theta_i, \rho_i).$$

Proof Since $p_{\text{orbit}}(b \mid \theta, \rho) = \text{Unif}(0{:}(2^m - 1))$ is independent of (θ, ρ), we have

$$p_{\text{orbit}}(b \mid \theta, \rho) = \frac{1}{2^m} = p_{\text{orbit}}(b - i \mid \theta_i, \rho_i).$$

■

Lemma 6 (Index detailed balance) *For all (θ, ρ, b, i),*

$$e^{-H(\theta,\rho)}\, p_{\text{index}}(i \mid \theta, \rho, b) = e^{-H(\theta_i,\rho_i)}\, p_{\text{index}}(-i \mid \theta_i, \rho_i, b - i).$$

Proof For each doubling $k = 0, \ldots, m-1$, set

$$\alpha_{k,k+1} := \alpha\big((a_k, b_k), (a_k^{\text{ext}}, b_k^{\text{ext}})\big) = \min\left(1,\, \frac{\sum_{j \in a_k^{\text{ext}}:b_k^{\text{ext}}} e^{-H(\theta_j,\rho_j)}}{\sum_{j \in a_k:b_k} e^{-H(\theta_j,\rho_j)}}\right).$$

The index evolves according to

$$i_{k+1} = \begin{cases} i_k^{\text{ext}}, & \text{with probability } \alpha_{k,k+1}, \\ i_k, & \text{with probability } 1 - \alpha_{k,k+1}, \end{cases} \qquad i_0 = 0,$$

where i_k^{ext} is drawn from the Gibbs distribution on the extension block, that is,

$$p(i_k^{\text{ext}} = j \mid \theta, \rho, b) = \frac{e^{-H(\theta_j,\rho_j)}}{\sum_{\ell \in a_k^{\text{ext}}:b_k^{\text{ext}}} e^{-H(\theta_\ell,\rho_\ell)}} \mathbf{1}\big\{j \in a_k^{\text{ext}}{:}b_k^{\text{ext}}\big\}.$$

If $i = 0$, then every proposal is rejected, hence

$$p_{\text{index}}(0 \mid \theta, \rho, b) = \prod_{k=0}^{m-1} \big(1 - \alpha_{k,k+1}\big).$$

Since $\Psi(\theta, \rho, b, 0) = (\theta, \rho, b, 0)$, both sides of the identity coincide.

Assume $i \neq 0$. By the update rule, there is a unique doubling k at which an acceptance occurs and no later acceptance occurs. Then $i \in a_k^{\text{ext}}{:}b_k^{\text{ext}}$ and

$$p_{\text{index}}(i \mid \theta, \rho, b) = \Big(\prod_{\ell=k+1}^{m-1} \big(1 - \alpha_{\ell,\ell+1}\big)\Big)\, \alpha_{k,k+1}\, \frac{e^{-H(\theta_i,\rho_i)}}{\sum_{\ell \in a_k^{\text{ext}}:b_k^{\text{ext}}} e^{-H(\theta_\ell,\rho_\ell)}}.$$

Multiplying by $e^{-H(\theta_0,\rho_0)}$ and using the elementary balance identity

$$\alpha\big((a_k, b_k), (a_k^{\text{ext}}, b_k^{\text{ext}})\big) \sum_{j \in a_k:b_k} e^{-H(\theta_j,\rho_j)} = \alpha\big((a_k^{\text{ext}}, b_k^{\text{ext}}), (a_k, b_k)\big) \sum_{j \in a_k^{\text{ext}}:b_k^{\text{ext}}} e^{-H(\theta_j,\rho_j)},$$

yields

$$e^{-H(\theta_0,\rho_0)}\, p_{\text{index}}(i \mid \theta, \rho, b) = \Big(\prod_{\ell=k+1}^{m-1} \big(1 - \alpha_{\ell,\ell+1}\big)\Big)\, \alpha\big((a_k^{\text{ext}}, b_k^{\text{ext}}), (a_k, b_k)\big)\, \frac{e^{-H(\theta_0,\rho_0)}\, e^{-H(\theta_i,\rho_i)}}{\sum_{j \in a_k:b_k} e^{-H(\theta_j,\rho_j)}}.$$

Now apply $\Psi(\theta, \rho, b, i) = (\theta_i, \rho_i, b - i, -i)$. As illustrated in Figure 3, this re-indexing makes (θ_i, ρ_i) the new initial state and relabels all indices by $k' = k - i$. Under this relabeling, the same recursive construction proceeds from index 0 (formerly i), and at the last accepted doubling the roles of the current and extension blocks are exchanged. Therefore

$$e^{-H(\theta_0,\rho_0)}\, p_{\text{index}}(i \mid \theta, \rho, b) = e^{-H(\theta_i,\rho_i)}\, p_{\text{index}}(-i \mid \theta_i, \rho_i, b - i),$$

as claimed. ■

Proposition 7 (Invariance of the extended density) *The extended density satisfies*

$$p_{\text{joint}} \circ \Psi = p_{\text{joint}}.$$

Proof Let $\tilde{p}_{\text{joint}}$ denote the unnormalized joint density

$$\tilde{p}_{\text{joint}}(\theta, \rho, b, i) = e^{-H(\theta_0,\rho_0)}\, p_{\text{orbit}}(b \mid \theta_0, \rho_0)\, p_{\text{index}}(i \mid \theta_0, \rho_0, b),$$

so that $p_{\text{joint}} = \tilde{p}_{\text{joint}}/Z$ for a constant Z.

Fix $z = (\theta, \rho, b, i) \in \mathbb{A}$. By Lemma 5,

$$p_{\text{orbit}}(b \mid \theta_0, \rho_0) = p_{\text{orbit}}(b - i \mid \theta_i, \rho_i),$$

and by Lemma 6,

$$e^{-H(\theta_0,\rho_0)}\, p_{\text{index}}(i \mid \theta_0, \rho_0, b) = e^{-H(\theta_i,\rho_i)}\, p_{\text{index}}(-i \mid \theta_i, \rho_i, b - i).$$

Substituting gives

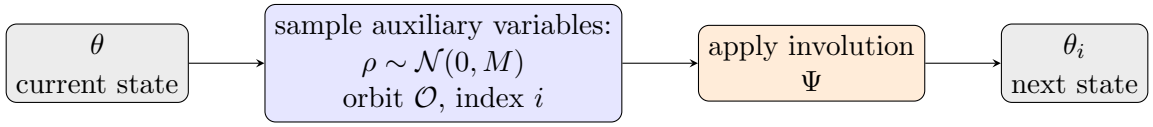
$$\tilde{p}_{\text{joint}}(z) = \tilde{p}_{\text{joint}}(\Psi z).$$

Since Z is constant, it follows that

$$p_{\text{joint}}(z) = p_{\text{joint}}(\Psi z),$$

and hence $p_{\text{joint}} \circ \Psi = p_{\text{joint}}$. ■

The diagram below summarizes the update steps of biased progressive HMC. No Metropolis correction is required, since the proposal involution Ψ preserves both the reference measure ζ (Lemma 4) and the joint density (Proposition 7). By Theorem 1, the induced Markov chain on θ is therefore reversible with respect to μ.



No Metropolis correction is needed since Ψ preserves the joint density.

In Section 3.4, we extend this reversibility argument to the full WALNUTS algorithm, where both the integration time and the step size are adapted. We now turn to an important special case of biased progressive HMC in which the leapfrog integrator is replaced by the exact Hamiltonian flow.

2.6 Idealized Case: Exact Biased Progressive HMC

To better understand the behavior of biased progressive HMC, it is instructive to consider an idealized version in which the leapfrog integrator Φ_h with fixed step size $h > 0$ is replaced with the exact Hamiltonian flow φ_h. In this setting, no numerical integration error is introduced, and the Hamiltonian is conserved exactly at each point along the orbit. As a result, all the Metropolis-style acceptance probabilities are equal to one, and every proposal index is accepted.

This idealization provides valuable insight into the behavior of biased progressive HMC when leapfrog integration errors are negligible. In particular, the distribution of the final integration time index i becomes independent of the target distribution, depending only on the randomized orbit construction procedure. This yields a clean mathematical description of the sampling law for i, and serves as a useful benchmark for understanding how the actual algorithm deviates from ideal behavior in practice.

Under exact integration, the orbit is constructed by repeatedly applying the exact Hamiltonian flow map φ_h rather than the leapfrog integrator. The orbit construction follows the same recursive doubling procedure as before, but without any accept/reject steps. At each doubling step, 2^k new states are generated by applying φ_h, either forward or backward in time with equal probability. After m such steps, the final orbit consists of 2^m consecutive states, each lying exactly on the Hamiltonian trajectory through the initial point.

The integration time index i is then drawn uniformly from the extension added during the final doubling step. Although this rule is simple (just uniform sampling from the last added segment), the resulting distribution of i is nonuniform, since the location of the final extension depends on the random sequence of doubling directions. We now briefly review this construction and describe its implications.

Let $m \geq 1$ be the number of doublings. Begin with the singleton orbit $\mathcal{O}_0 = ((\theta, \rho))$ and initial indices $a_0 = b_0 = 0$. For each doubling step $k \in \{0, \ldots, m-1\}$, extend the current orbit by appending 2^k consecutive states generated by repeatedly applying the exact Hamiltonian flow φ_h to either the left or right end, each with probability $1/2$. After m doublings, the resulting orbit contains 2^m phase space states indexed by consecutive integers that include 0.

Let $\mathcal{O}_{m-1}^{\text{ext}}$ denote the extension added during the final doubling step ($k = m-1$), with index range $a_k^{\text{ext}}{:}b_k^{\text{ext}}$ and length 2^{m-1}. The final integration time index is then sampled uniformly from this index range:

$$i \;\sim\; \text{Unif}\big(a_k^{\text{ext}}{:}b_k^{\text{ext}}\big)\,,$$

giving each index in the last-added extension equal weight. The next results describe the distribution of i, as illustrated in Figure 4, and provide a closed-form expression for $\mathbb{E}[\,|i|\,]$.

Theorem 8 *Fix an integer $m \geq 1$ and set $N = 2^{m-1}$. Let i denote the integration time index selected by the exact biased progressive sampling procedure with uniform weights. Then i follows the symmetric discrete triangular law:*

$$\Pr\big[i = k\big] \;=\; \frac{1}{2N^2}\,\min(|k|, 2N - |k|)\;\mathbf{1}\{1 \leq |k| \leq 2N-1\} \tag{5}$$

for $k \in (1-2N){:}(2N-1)$.

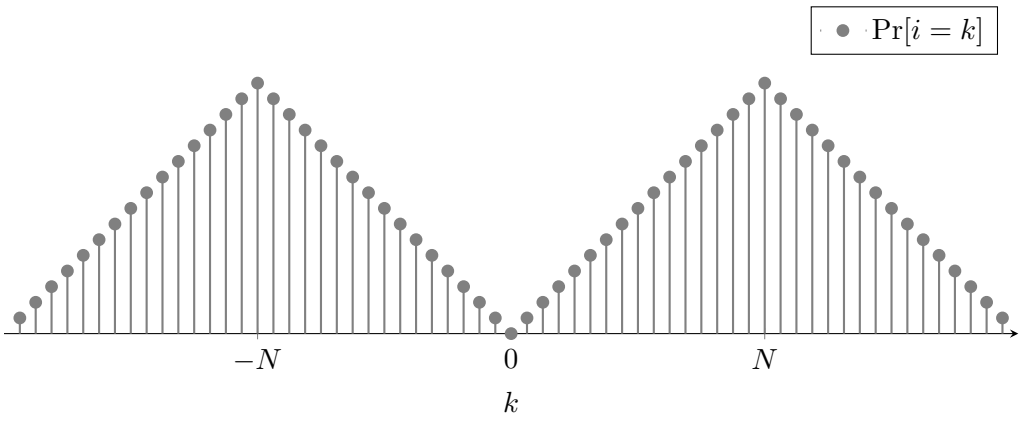


Figure 4: *Probability mass function of the integration time index i in* exact *biased progressive HMC with $m = 5$. The triangular shape reflects the random location of the final extension orbit from which i is uniformly sampled.*

Proof At each doubling step ($k = 0, \ldots, m-1$), the Bernoulli variable B_{k+1} indicates whether the extension orbit $\mathcal{O}_k^{\text{ext}}$ of length 2^k is appended to the right ($B_{k+1} = 1$) or to the left ($B_{k+1} = 0$) of the current orbit $\mathcal{O}_k$. In the final doubling step, the orbit $\mathcal{O}_{m-1}$ has length $N = 2^{m-1}$ with right endpoint index:

$$b_{m-1} = \sum_{j=1}^{m-1} B_j \, 2^{j-1} \, .$$

which implies $b_{m-1} \sim \text{Unif}(0{:}(N-1))$. The corresponding left endpoint index is $a_{m-1} = b_{m-1} - N + 1$.

The final bit B_m determines whether the last extension of length N is appended to the right or left:

- If $B_m = 1$, then $a_{m-1}^{\text{ext}} = b_{m-1} + 1$ and $b_{m-1}^{\text{ext}} = b_{m-1} + N$.
- If $B_m = 0$, then $a_{m-1}^{\text{ext}} = a_{m-1} - N$ and $b_{m-1}^{\text{ext}} = a_{m-1} - 1$.

The index i is then sampled uniformly from this extension. Let $c \sim \text{Unif}(1{:}N)$, independent of all previous random variables. Then

$$\begin{aligned} i &= (b_{m-1} + c)B_m + (a_{m-1} - c)(1 - B_m), \\ &= (b_{m-1} + c) \cdot B_m + (b_{m-1} - N + 1 - c) \cdot (1 - B_m), \end{aligned}$$

where the second line uses the identity $a_{m-1} = b_{m-1} - N + 1$. This expression shows that i is distributed as the convolution of two independent discrete uniform variables, yielding the symmetric triangular PMF given in the theorem. ■

$\min\big(\rho_a \cdot (\theta_b - \theta_a),\ \rho_b \cdot (\theta_b - \theta_a)\big) > 0$ $\qquad$ $\min\big(\rho_a \cdot (\theta_b - \theta_a),\ \rho_b \cdot (\theta_b - \theta_a)\big) < 0$

Figure 5: Position-space projections of two orbits: one that does not satisfy the U-turn condition (left), and one that does (right). For each case, the velocity vectors ρ_a and ρ_b, as well as the displacement vector $\theta_b - \theta_a$, are shown, assuming the mass matrix $M = I$.

The following corollary computes the expected magnitude of the integration time index under this distribution, providing a simple benchmark for the typical integration time in this idealized setting.

Corollary 9 *It holds that* $\mathbb{E}[\,|i|\,] = 2^{m-1}$.

Proof From Theorem 8, the magnitude $|i|$ has a triangular distribution on $\{1, \ldots, 2N-1\}$ that is symmetric about N. Hence its expectation equals the center of symmetry, namely $\mathbb{E}[|i|] = N = 2^{m-1}$. ■

Next, we describe how the orbit length can be locally adapted without breaking reversibility using a geometric stopping rule based on a U-turn condition.

2.7 Locally Adaptive Path Length with The No-U-Turn Sampler

The key innovation of the No-U-Turn Sampler (NUTS) is to locally adapt the orbit length, and thus integration time, by terminating the doubling procedure in biased progressive HMC when a U-turn condition is met (see Figure 5).

Definition 10 *An orbit* $\mathcal{O} = ((\theta_a, \rho_a), \ldots, (\theta_b, \rho_b))$ *satisfies the* U-turn condition *if:*

$$\min\big(\rho_a \cdot M^{-1}(\theta_b - \theta_a),\ \rho_b \cdot M^{-1}(\theta_b - \theta_a)\big) \ < \ 0\,. \tag{6}$$

To ensure reversibility, NUTS also checks a *sub-U-turn condition*. Because NUTS constructs orbits through iterative doubling, the orbit length is always a power of two: $|\mathcal{O}| = 2^m$ for some $m \in \mathbb{N}$. For such orbits, we define a hierarchy of sub-orbits generated by recursive halving:

$$\Big\{\mathcal{O}_{i,j} \ : \ i \in \big\{0, 1, \ldots, \log_2 |\mathcal{O}|\big\},\ j \in \big\{1, 2, \ldots, 2^i\big\}\Big\} \tag{7}$$

where, for each $i \in \{0, 1, \ldots, \log_2 |\mathcal{O}|\}$, $\mathcal{O}_{i,j}$ are defined to be the unique orbits of size $|\mathcal{O}|2^{-i}$ such that

$$\mathcal{O} \;=\; \mathcal{O}_{i,1} \odot \mathcal{O}_{i,2} \odot \cdots \odot \mathcal{O}_{i,2^i} \;.$$

Definition 11 *An orbit $\mathcal{O}$ is said to satisfy the* sub-U-turn condition *if at least one of its sub-orbits, as defined in* (7)*, satisfies the U-turn condition given in Definition 10.*

Sub-U-turn checks play a key role in ensuring reversibility. When the orbit construction procedure is rerun starting from the i-th point along the final orbit, the sequence of doubling steps required to recover the same final orbit is uniquely determined by the binary representation of $b - i$. However, if sub-U-turn checks are not enforced, the algorithm may terminate too early during this rerun, producing a different final orbit. This breaks the symmetry of the orbit construction with respect to the starting point along the orbit.

Given an initial state $(\theta, \rho) \in \mathbb{R}^{2d}$ and a maximum number of doublings $m_{\max} \in \mathbb{N}$, NUTS constructs an adaptive-length orbit as follows.

1. **Initialization.** Start with the singleton orbit $\mathcal{O}_0 = ((\theta, \rho))$, and set the index range $a_0 = b_0 = 0$. Sample $m_{\max}$ i.i.d. Bernoulli(1/2) random variables $B = (B_1, \ldots, B_{m_{\max}})$.

2. **Doubling steps.** For each step $k \in 0{:}(m_{\max} - 1)$, perform the following:

   (a) **Current orbit.** Let the current orbit be

   $$\mathcal{O}_k = ((\theta_{a_k}, \rho_{a_k}), \ldots, (\theta_{b_k}, \rho_{b_k}))$$

   with length $|\mathcal{O}_k| = 2^k$ and index range $a_k{:}b_k$.

   (b) **Extension.** Construct an orbit $\mathcal{O}_k^{\text{ext}}$ of length 2^k over the index range $a_k^{\text{ext}}{:}b_k^{\text{ext}}$, defined by the value of B_{k+1}:

   $$a_k^{\text{ext}} = \begin{cases} b_k + 1 & \text{if } B_{k+1} = 1, \\ a_k - 2^k & \text{if } B_{k+1} = 0, \end{cases} \quad b_k^{\text{ext}} = \begin{cases} b_k + 2^k & \text{if } B_{k+1} = 1, \\ a_k - 1 & \text{if } B_{k+1} = 0. \end{cases}$$

   (c) **Sub-U-turn check.** If $\mathcal{O}_k^{\text{ext}}$ satisfies the sub-U-turn condition, terminate and set $\mathcal{O} = \mathcal{O}_k$.

   (d) **Orbit update.** If not, define the updated orbit: $\mathcal{O}_{k+1} = \begin{cases} \mathcal{O}_k \odot \mathcal{O}_k^{\text{ext}} & \text{if } B_{k+1} = 1, \\ \mathcal{O}_k^{\text{ext}} \odot \mathcal{O}_k & \text{if } B_{k+1} = 0. \end{cases}$

   (e) **U-turn check.** If $\mathcal{O}_{k+1}$ satisfies the U-turn condition or if $k + 1 = m_{\max}$, terminate and set $\mathcal{O} = \mathcal{O}_{k+1}$.

3. **Output.** The final orbit $\mathcal{O}$ has length 2^m for some $m \leq m_{\max}$ and is fully determined by its length and rightmost index.

The integration time index is then selected exactly as in biased progressive HMC. For clarity, the integration time index selection has been omitted in the above description, as it follows identically from the biased progressive HMC case described in Section 2.5.

3. WALNUTS

WALNUTS addresses a key limitation in traditional NUTS: fixed step size integrators can fail to capture local variations in geometry, especially in stiff or funnel-shaped targets. By introducing a locally adaptive variable step-size leapfrog integrator within a fixed macro time grid, WALNUTS improves robustness while maintaining reversibility. It does so using the auxiliary variable GIST framework introduced in Section 2.3, which provides a principled foundation for both path length and step size adaptation.

At a high level, each transition step of WALNUTS uses a locally adaptive leapfrog integrator with variable micro step sizes to construct an orbit $\mathcal{O}$ along a macro time grid. Once the orbit is built, WALNUTS selects the next state by sampling a point in $\mathcal{O}$, thereby randomizing the integration time. The general framework for orbit construction and state selection was introduced in Section 2; what follows highlights the modifications needed to incorporate a variable step-size leapfrog integrator.

3.1 Variable Step-Size Leapfrog Integrator

The leapfrog (or Verlet) integrator is the standard method used to numerically approximate Hamiltonian dynamics in HMC (Bou-Rabee and Sanz-Serna, 2018; Hairer et al., 2010; Leimkuhler and Reich, 2004). In this section, we develop a locally adaptive extension that varies the step size within each macro time interval based on local energy error.

Macro/Micro grids and integration notation. In our framework, we assume a uniform *macro* time grid to anchor the overall trajectory, while allowing for variable *micro* step sizes within each macro interval to locally control integration errors. Each macro interval is subdivided into a number of micro steps, with finer subdivisions used in regions of high curvature or energy variation. This two-level structure enables local adaptivity while preserving a globally consistent time grid, as illustrated in Figure 6. We restrict attention to uniform macro grids throughout; the advantages of this choice for reversibility are discussed in Remark 16.

To formalize this, we introduce a fixed macro step size $h > 0$ and define the evenly-spaced time grid

$$t_k \;:=\; kh \quad \text{where} \quad k \in \mathbb{Z}\,.$$

Each macro interval $[t_k, t_{k+1}]$ is subdivided into $\ell_{k,k+1}$ micro steps, each of size $h\ell_{k,k+1}^{-1}$. We denote by $\{\ell_{k,k+1}\}_{k\in\mathbb{Z}}$ the sequence of micro step counts associated with each macro interval. For any $m, n \in \mathbb{Z}$, the evolution of the numerical solution from time t_n to t_m using these variable step sizes is defined by the following composition of leapfrog maps:

$$\Phi_{m,n} = \begin{cases} \Phi^{\ell_{m-1,m}}_{h\ell^{-1}_{m-1,m}} \circ \cdots \circ \Phi^{\ell_{n,n+1}}_{h\ell^{-1}_{n,n+1}} & \text{if } m > n \\ \Phi^{\ell_{m,m+1}}_{-h\ell^{-1}_{m,m+1}} \circ \cdots \circ \Phi^{\ell_{n-1,n}}_{-h\ell^{-1}_{n-1,n}} & \text{if } m < n \end{cases} \tag{8}$$

with $\Phi_{m,m}$ defined as the identity map. The forward evolution ($m > n$) advances the numerical solution from t_n to t_m using positive micro-step sizes, while the backward evolution ($m < n$) is implemented by composing leapfrog maps with negative micro-step sizes.

If the micro step counts $\{\ell_{k,k+1}\}_{k\in\mathbb{Z}}$ are fixed in advance or treated as auxiliary variables in an enlarged state space, then $\Phi_{m,n}$ is volume-preserving, as it is a composition of volume-preserving leapfrog maps (Bou-Rabee and Sanz-Serna, 2018, Proposition 2.3). However, if

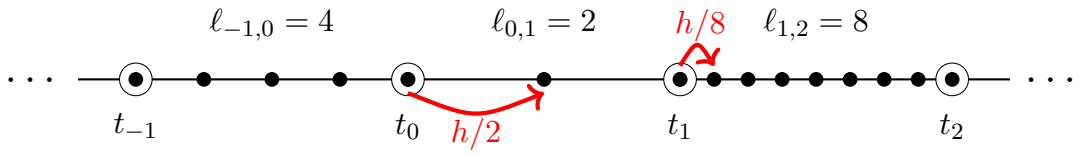


Figure 6: The time grid used to advance the numerical solution from t_0 to t_1 consists of two microsteps, each of size $h/2$. Similarly, the transition from t_1 to t_2 is achieved using eight microsteps, each of size $h/8$.

$\{\ell_{k,k+1}\}_{k\in\mathbb{Z}}$ depends on the state during integration, then volume preservation is no longer guaranteed. In either case, $\Phi_{m,n}$ is not, in general, time-reversible: although each leapfrog step is time-reversible, the composition of time-reversible maps with varying step sizes does not, in general, yield a time-reversible map (Bou-Rabee and Sanz-Serna, 2018, Theorem 4.2).

Micro step selection using local energy error. In locally adaptive HMC, the primary goal of local step size adaptivity is to select the largest leapfrog step size that ensures the energy error remains below a specified threshold. Here, we describe a general procedure for determining such step sizes. Given an initial condition $(\theta, \rho) \in \mathbb{R}^{2d}$, a macro step size h, and a prescribed maximum allowable energy error $\delta > 0$, we define

$$\text{micro}(\theta, \rho, \mu, M, h, \delta) \;:=\; \min\{\ell \in 2^{\mathbb{N}} : \, H_\ell^+ - H_\ell^- \le \delta\}, \tag{9}$$

where $2^{\mathbb{N}} = \{2^n : n \in \mathbb{N}\}$.

For a fixed $\ell \in 2^{\mathbb{N}}$, we integrate forward over $[0, h]$ using micro step size $h\ell^{-1}$. We denote by $(\theta^{(j)}, \rho^{(j)})$, $0 \le j \le \ell$, the resulting *micro iterates*, defined by

$$(\theta^{(j)}, \rho^{(j)}) = \Phi_{h\ell^{-1}}^j(\theta, \rho), \qquad 0 \le j \le \ell,$$

so that $(\theta^{(0)}, \rho^{(0)}) = (\theta, \rho)$. We then set

$$H_\ell^+ = \max_{0\le j\le \ell} H(\theta^{(j)}, \rho^{(j)}),$$
$$H_\ell^- = \min_{0\le j\le \ell} H(\theta^{(j)}, \rho^{(j)}).$$

Thus, $\text{micro}(\theta, \rho, \mu, M, h, \delta)$ returns the minimal power of two such that the micro-level leapfrog trajectory over $[0, h]$ remains within energy error δ.

Remark 12 *The index-selection weights penalize increases in energy. Large negative energy drift reflects departure from the regime in which the numerical integrator faithfully tracks Hamiltonian dynamics. By time reversibility, every decrease in energy corresponds to an increase of the same magnitude along the reversed trajectory (Bou-Rabee and Sanz-Serna, 2018, Figure 6.1). Hence, large negative and positive energy errors are structurally paired. Substantial energy drift (of either sign) signals loss of geometric fidelity and increases the risk of practical non-reversibility of the step-size control mechanism. Controlling the absolute energy error therefore provides a natural step-size selection criterion.*

Remark 13 *In the optimized implementation of WALNUTS, the energy error used to assess the validity of a micro trajectory is computed as the absolute difference of the Hamiltonian evaluated at the endpoints, i.e.,* $|H(\theta^{(0)}, \rho^{(0)}) - H(\theta^{(\ell)}, \rho^{(\ell)})|$. *This choice improves practical performance by reducing mismatches between forward and backward micro steps, and ensures that the energy error threshold* δ *has a consistent interpretation across different values of the step-size reduction factor* ℓ. *Reversibility, however, is independent of the specific deterministic rule used to select* ℓ, *and therefore this implementation modification does not alter the reversibility guarantee.*

The next lemma shows that the micro step-size factor $\ell = \text{micro}(\theta, \rho, \mu, M, h, \delta)$ is always greater than or equal to the value obtained by applying the same procedure at the endpoint of the corresponding forward trajectory, but with the momentum reversed.

Lemma 14 *For any* $(\theta, \rho) \in \mathbb{R}^{2d}$, *and energy error threshold* $\delta > 0$, *the forward step-size reduction factor satisfies:*

$$\ell = \text{micro}(\theta, \rho, \mu, M, h, \delta) \geq \text{micro}\left(\theta', -\rho', \mu, M, h, \delta\right), \quad \textit{where} \quad (\theta', \rho') = \Phi^{\ell}_{h\ell^{-1}}(\theta, \rho)\ .$$

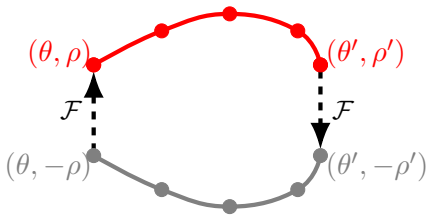
Proof Let $\ell = \text{micro}(\theta, \rho, \mu, M, h, \delta)$. By definition of micro in (9), we have:

$$\max\{H(\theta^{(j)}, \rho^{(j)}) \,:\, 0 \leq j \leq \ell\} - \min\{H(\theta^{(j)}, \rho^{(j)}) \,:\, 0 \leq j \leq \ell\} \leq \delta \tag{10}$$

where the leapfrog iterates are given by

$$(\theta^{(j)}, \rho^{(j)}) = \Phi^{j}_{h\ell^{-1}}(\theta, \rho), \quad 0 \leq j \leq \ell\ .$$

For the backward evolution, we consider the trajectory starting from (θ', ρ'). As shown in the grey dots in the image on the right, by applying a momentum flip $\mathcal{F}$, the backward orbit is given by:



$$\Phi^{j}_{h\ell^{-1}} \circ \mathcal{F}(\theta', \rho') = (\theta^{(\ell-j)}, -\rho^{(\ell-j)}), \quad 0 \leq j \leq \ell\ .$$

Since $H \circ \mathcal{F} \equiv H$, it follows that the backward trajectory at step size $h\ell^{-1}$ also satisfies (10). Thus, the backward critical step-size reduction factor cannot exceed ℓ, ensuring that $\ell \geq \text{micro}\,(\theta', -\rho', \mu, M, h, \delta)$. ■

Randomized step size selection within orbits. On a given macro step, Lemma 14 yields only an upper bound for the backward reduction factor and does not guarantee that the forward and backward reduction factors coincide. To make the orbit construction robust to this lack of a symmetry guarantee between forward and backward reductions, we introduce randomization in the step-size selection.

Let p_{micro} be the distribution from which the step-size reduction factor is sampled given the critical step-size reduction factor. Given an initial point (θ_0, ρ_0) at time t_0, we recursively define the sequence of points (θ_j, ρ_j) at times t_j as follows:

- **Forward recursion** ($j > 0$):

$$(\theta_j, \rho_j) = \Phi^{\ell_{j-1,j}}_{h\ell^{-1}_{j-1,j}}(\theta_{j-1}, \rho_{j-1}),$$

  where the step-size reduction factor $\ell_{j-1,j}$ is sampled from the conditional distribution:

$$\ell_{j-1,j} \sim p_{\text{micro}}(\cdot \mid \text{micro}(\theta_{j-1}, \rho_{j-1}, \mu, M, h, \delta)).$$

- **Backward recursion** ($j < 0$):

$$(\theta_j, \rho_j) = \Phi^{\ell_{j,j+1}}_{-h\ell^{-1}_{j,j+1}}(\theta_{j+1}, \rho_{j+1}),$$

  where $\ell_{j,j+1}$ is sampled from the analogous distribution:

$$\ell_{j,j+1} \sim p_{\text{micro}}(\cdot \mid \text{micro}(\theta_{j+1}, -\rho_{j+1}, \mu, M, h, \delta)).$$

A simple choice for the distribution p_{micro} is a uniform distribution over two candidate micro step-size factors: the deterministic baseline value $\widetilde{\ell} = \texttt{micro}(\theta, \rho, \mu, M, h, \delta)$, as defined in (9), and its immediate successor $\widetilde{\ell} + 1$. Specifically,

$$p_{\text{micro}}(k \mid \widetilde{\ell}) = \frac{1}{2}\,\mathbf{1}\Big\{k \in \{\widetilde{\ell},\ \widetilde{\ell} + 1\}\Big\}\,.$$

Note that the support of p_{micro} is limited to $\{\widetilde{\ell}, \widetilde{\ell} + 1\}$. This restriction is justified by Lemma 14, which guarantees that the forward micro step-size factor is always greater than or equal to the backward one over the same macro step. Therefore, if a given step size yields an acceptable energy error in the forward direction, it will also do so in the backward direction. This allows us to safely omit smaller micro step factors from consideration.

Remark 15 *The recently proposed step size adaptive NUTS algorithm (Bou-Rabee et al., 2025) uses the same random number of leapfrog micro steps within each macro step. Specifically, for each macro step spanning the index range a:b, the number of microsteps is given by a random variable ℓ such that $\ell_{a,a+1} = \ell_{a+1,a+2} = \cdots = \ell_{b-1,b} = \ell$ with ℓ drawn from a distribution that depends on (θ, ρ), a, and b.*

3.2 Orbit Construction with variable step-size Leapfrog Integration

Orbit construction proceeds exactly as in NUTS, except that fixed step size leapfrog integration is replaced by variable step-size leapfrog integration during the orbit extension steps described in Section 2.7. Specifically, by (8), the overall evolution from time $t_0 = 0$ to time t_j is given by

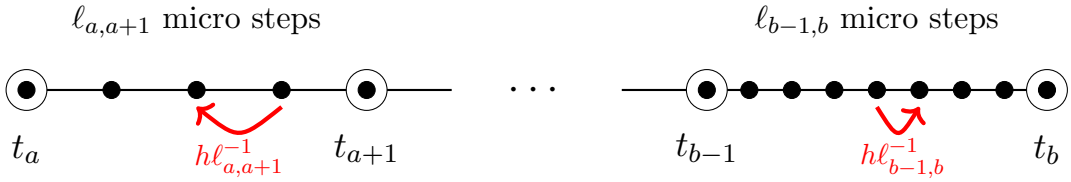
$$\Phi_{j,0}|_{\boldsymbol{\ell}} = \begin{cases} \Phi^{\ell_{j-1,j}}_{h\ell^{-1}_{j-1,j}} \circ \cdots \circ \Phi^{\ell_{0,1}}_{h\ell^{-1}_{0,1}} & \text{if } j > 0 \\ \Phi^{\ell_{j,j+1}}_{-h\ell^{-1}_{j,j+1}} \circ \cdots \circ \Phi^{\ell_{-1,0}}_{-h\ell^{-1}_{-1,0}} & \text{if } j < 0 \end{cases} \tag{11}$$

where $\Phi_{0,0}$ is the identity map. Each composition uses a different number of micro steps, governed by the micro step-size reduction factors $\ell_{k,k+1}$ stored in the tuple $\boldsymbol{\ell}$. Given two

endpoint indices $a, b \in \mathbb{Z}$ with $a \leq b$, the orbit over the index range $a{:}b = \{a, a+1, \ldots, b\}$, starting at (θ_0, ρ_0) at time $t_0 = 0$, is defined by

$$\mathcal{O} = (\Phi_{a,0}|_{\boldsymbol{\ell}}(\theta_0, \rho_0), \ldots, \Phi_{b,0}|_{\boldsymbol{\ell}}(\theta_0, \rho_0)) = ((\theta_a, \rho_a), \ldots, (\theta_b, \rho_b)).$$

The figure below illustrates this construction. Each macro-step interval $[t_k, t_{k+1}]$ is subdivided into $\ell_{k,k+1}$ micro steps. The number of micro steps may vary across intervals, as shown by the differing densities of black points in the two macro-steps depicted. These locally adaptive refinements allow WALNUTS to dynamically adjust the resolution of leapfrog integration within each macro step, e.g., by taking more micro steps between $b-1$ and b than between a and $a+1$:



3.3 Randomized Integration Time via Biased Progressive Sampling

In WALNUTS, the index of the next state in the Markov chain is selected via biased progressive sampling, as in biased progressive HMC. This index corresponds to a randomly chosen point along the orbit and may be interpreted as a randomized integration time.

We use categorical(S, q) to denote the categorical distribution supported on a discrete set S with the probability of selecting $x \in S$ proportional to $q(x)$. In view of Lemma 14,

$$p_{\text{micro}}(\ell_{k,k+1} \mid \text{micro}(\theta_k, \rho_k, \mu, M, h, \delta))$$

may not equal

$$p_{\text{micro}}(\ell_{k,k+1} \mid \text{micro}(\theta_{k+1}, -\rho_{k+1}, \mu, M, h, \delta)) \; .$$

Figure 7 illustrates this asymmetry at the orbit level: when the orbit is viewed from different starting indices, the macro-step directions can differ only on the interval between the two starting points, while outside that interval the directions agree.

To ensure reversibility, we assign a weight to each point in the orbit, defined recursively as follows.

- The initial weight is
$$w_0 = \mu(\theta_0) e^{-\frac{1}{2}\rho_0^\intercal M^{-1} \rho_0} \; . \tag{12}$$

- For $j > 0$,
$$w_j = \frac{\mu(\theta_j) e^{-\frac{1}{2}(\rho_j)^\intercal M^{-1} \rho_j}}{\mu(\theta_{j-1}) e^{-\frac{1}{2}(\rho_{j-1})^\intercal M^{-1} \rho_{j-1}}} \frac{p_{\text{micro}}(\ell_{j-1,j} \mid \text{micro}(\theta_j, -\rho_j, \mu, M, h, \delta))}{p_{\text{micro}}(\ell_{j-1,j} \mid \text{micro}(\theta_{j-1}, \rho_{j-1}, \mu, M, h, \delta))} w_{j-1} \; . \tag{13}$$

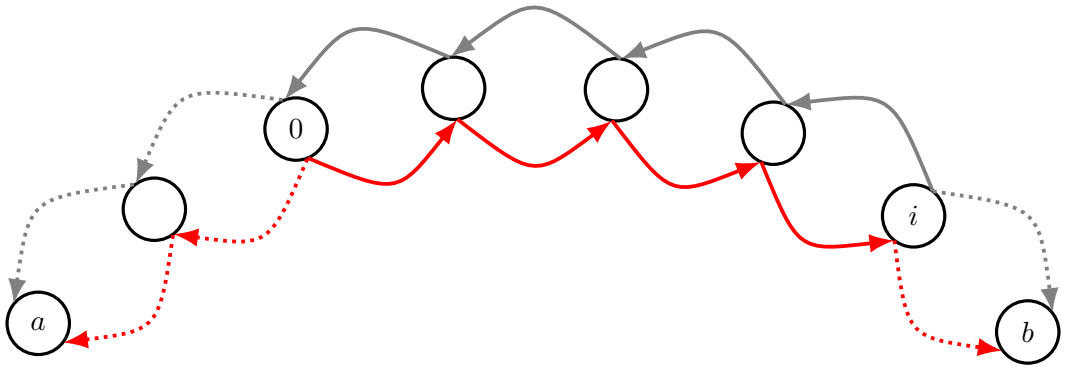


Figure 7: Illustration of orbit construction in WALNUTS. Red arrows indicate the directions of macro steps starting from the initial state indexed at 0. Gray arrows indicate the macro-step directions when the orbit is constructed starting from the i-th state. The dotted arrows mark subintervals where the macro steps from the two starting indices move in the same direction. On these subintervals, the conditional distribution of the step-size reduction factor is the same; any difference in the conditional distribution can occur only on the interval between $\min(i, 0)$ and $\max(i, 0)$.

- While for $j < 0$,

$$w_j = \frac{\mu(\theta_j) e^{-\frac{1}{2}(\rho_j)^\intercal M^{-1} \rho_j}}{\mu(\theta_{j+1}) e^{-\frac{1}{2}(\rho_{j+1})^\intercal M^{-1} \rho_{j+1}}} \frac{p_{\text{micro}}(\ell_{j,j+1} \mid \text{micro}(\theta_j, \rho_j, \mu, M, h, \delta))}{p_{\text{micro}}(\ell_{j,j+1} \mid \text{micro}(\theta_{j+1}, -\rho_{j+1}, \mu, M, h, \delta))} w_{j+1} \,. \tag{14}$$

Given an orbit with endpoint indices $a, b \in \mathbb{Z}$, we define the corresponding collection of weights by

$$\mathcal{W} = (w_a, \ldots, w_b) \,.$$

Biased progressive sampling in WALNUTS selects the index of the next state iteratively as follows. The process begins with the initial index $i_0 = 0$. At the k-th doubling step ($k \in \{0, \ldots, m_{\max} - 1\}$), let $\mathcal{O}_k$, $\mathcal{W}_k$ denote the current orbit and its weights, and $\mathcal{O}_k^{\text{ext}}$, $\mathcal{W}_k^{\text{ext}}$ denote the proposed extension and its weights. An extension index is sampled according to

$$i_k^{\text{ext}} \sim \text{categorical}(a_k^{\text{ext}}{:}b_k^{\text{ext}}, \mathcal{W}_k^{\text{ext}})$$

and accepted with probability given by the acceptance function

$$\alpha(\mathcal{W}_k, \mathcal{W}_k^{\text{ext}}) := \min\left(1, \frac{\sum \mathcal{W}_k^{\text{ext}}}{\sum \mathcal{W}_k}\right) \,. \tag{15}$$

Here, the expression $\sum \mathcal{W}$ is shorthand for the sum of all the weights in the collection $\mathcal{W}$, i.e., $\sum \mathcal{W} = \sum_{w \in \mathcal{W}} w$. The index update is:

$$i_{k+1} = \begin{cases} i_k^{\text{ext}} & \text{with probability } \alpha(\mathcal{W}_k, \mathcal{W}_k^{\text{ext}}), \\ i_k & \text{otherwise.} \end{cases} \tag{16}$$

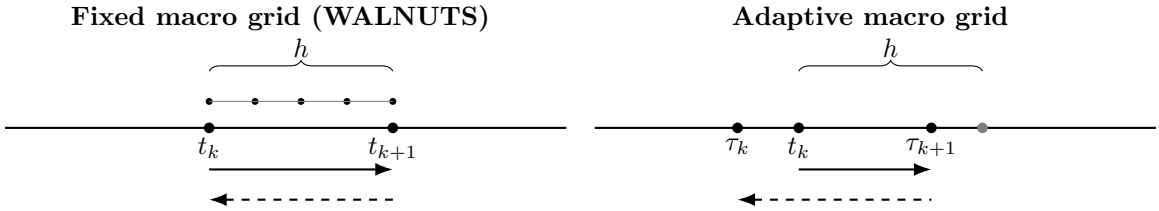


Figure 8: *Left:* With a fixed macro grid $t_{k+1} - t_k = h$, refinements remain anchored to the same macro interval $[t_k, t_{k+1}]$, so forward and backward traversals target the identical time interval. *Right:* With an adaptive macro grid, the forward step from t_k may terminate at $\tau_{k+1} < t_k + h$, and the backward check from τ_{k+1} may land at $\tau_k < t_k$, so the forward and backward macrosteps target different time intervals of the trajectory.

This Metropolis-style rule favors integration times in the extension orbit while ensuring reversibility.

Remark 16 *A key advantage of WALNUTS is that it operates on a* fixed, evenly spaced *macro time grid* $\{t_k\}_{k\in\mathbb{Z}}$ *with* $t_{k+1} - t_k = h$. *Each call to* micro *in* (9) *refines the numerical resolution* within the same macro interval $[t_k, t_{k+1}]$, *so forward and backward constructions are anchored to identical time segments of the underlying Hamiltonian trajectory; see Figure 8 (left).*

In contrast, some schemes (used e.g. by Biron-Lattes et al., 2024; Kleppe, 2016) do not fix the macro grid in advance. Instead, the macro endpoint itself is determined adaptively by the local error control. Starting from a nominal step size h, *the step size is reduced until an error criterion is met, and a single accepted step is taken at that refined size. As a result, the forward move from* t_k *may terminate at an endpoint* $\tau_{k+1} \neq t_k + h$, *so the macro grid is effectively irregular.*

Reversibility is then checked by integrating backward from τ_{k+1} *using the same adaptive rule. Because the macro grid is not fixed, the backward integration may land at* $\tau_k \neq t_k$. *Thus the forward and backward macrosteps target different time intervals of the Hamiltonian trajectory; see Figure 8 (right). WALNUTS avoids this issue by fixing the macro grid in advance and adapting only the micro grid within each prescribed macro interval.*

3.4 Reversibility of WALNUTS

We present a rigorous proof of the reversibility of WALNUTS by interpreting it as a GIST sampler, as described in Section 2.3, and verifying that it satisfies the conditions of Theorem 1. While the overall structure of the proof is concise, establishing reversibility requires careful handling of the variable step-size leapfrog integrator and its interaction with orbit construction and state selection.

To streamline the proof, we introduce some additional notation. For brevity, we write $\ell_j = \ell_{j,j+1}$ for the micro step-size factor associated with the macro step interval $[t_j, t_{j+1}]$. By

construction, ℓ_j denotes the step-size reduction factor used when expanding the orbit away from the initial point (θ_0, ρ_0): forward from (θ_j, ρ_j) to $(\theta_{j+1}, \rho_{j+1})$ if $j > 0$, and backward from $(\theta_{j+1}, \rho_{j+1})$ to (θ_j, ρ_j) if $j < 0$. With this notation, (11) simplifies to:

$$\Phi_{j,0}|_{\boldsymbol{\ell}} = \begin{cases} \Phi^{\ell_{j-1}}_{h\ell_{j-1}^{-1}} \circ \cdots \circ \Phi^{\ell_0}_{h\ell_0^{-1}} & \text{if } j > 0 \\ \Phi^{\ell_j}_{-h\ell_j^{-1}} \circ \cdots \circ \Phi^{\ell_{-1}}_{-h\ell_{-1}^{-1}} & \text{if } j < 0 \end{cases} \tag{17}$$

and $\Phi_{0,0}$ is the identity map.

As described in Section 3.2, a WALNUTS orbit

$$\mathcal{O} \;=\; ((\theta_a, \rho_a), \ldots, (\theta_b, \rho_b))$$

is uniquely determined by

- an initial point $(\theta_0, \rho_0) \in \mathbb{R}^{2d}$,
- m: the number of doublings,
- b: the rightmost index of the orbit, and
- $\boldsymbol{\ell} = \{\ell_j\}_{a \le j \le b-1}$: a collection of 2^{m-1} micro step-size factors, one for each macro step.

The leftmost index is given by $a = b - 2^m + 1$. Accordingly, we use $(\theta_0, \rho_0, \mathcal{O})$ and the tuple $(\theta_0, \rho_0, m, b, \boldsymbol{\ell})$ interchangeably. Once the micro step-size factors $\boldsymbol{\ell}$ are sampled, they determine the sequence of forward and backward leapfrog steps used to generate the orbit from the initial point (θ, ρ).

We treat $m, b, \boldsymbol{\ell}$ and the integration time index $i \in a{:}b$ as auxiliary variables, and define the augmented state

$$z \;:=\; (\theta_0, \rho_0, m, b, \boldsymbol{\ell}, i)$$

taking values in the augmented state space

$$\mathbb{A} \;:=\; \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}^{2^m - 1} \times \mathbb{Z}\,.$$

Let $p_{\text{orbit}}(m, b, \boldsymbol{\ell} \mid \theta_0, \rho_0)$ be the conditional distribution of the orbit given its starting point, as defined by the orbit construction procedure in Section 3.2. The conditional over the integration time $p_{\text{index}}(i \mid \theta_0, \rho_0, m, b, \boldsymbol{\ell})$ is defined by biased progressive sampling, as described in Section 3.3. To verify the conditions of Theorem 1, it is helpful to also consider a simpler alternative. To distinguish between these two integration time selection strategies, we introduce the following notation, which extends to the corresponding extended target distributions as well:

- **Biased progressive sampling** (p^{BP}) is the method used in WALNUTS. It defines $p^{\text{BP}}_{\text{index}}$ recursively. Let $\mathcal{O}_k$ denote the orbit at k-th doubling step and $\mathcal{W}_k$ its associated weights. Then,

$$\begin{aligned} p^{\text{BP}}_{\text{index}}(j \mid \theta_0, \rho_0, \mathcal{O}_{k+1}) = \alpha(\mathcal{W}_k, \mathcal{W}_k^{\text{ext}}) \cdot \frac{w_j}{\sum \mathcal{W}_k^{\text{ext}}} \cdot \mathbf{1}\{j \in a_k^{\text{ext}}{:}b_k^{\text{ext}}\} \\ + \big(1 - \alpha(\mathcal{W}_k, \mathcal{W}_k^{\text{ext}})\big) \cdot p^{\text{BP}}_{\text{index}}(j \mid \theta_0, \rho_0, \mathcal{O}_k) \cdot \mathbf{1}\{j \in a_k{:}b_k\}, \end{aligned} \tag{18}$$

  where $\mathcal{W}_k^{\text{ext}}$ are the weights associated with the extension orbit $\mathcal{O}_k^{\text{ext}}$, and $\alpha(\mathcal{W}_k, \mathcal{W}_k^{\text{ext}})$ is the acceptance probability defined in (15).

- **Multinomial sampling** (p^{MN}) draws the integration time index from a categorical distribution proportional to the orbit weights:

$$p^{\text{MN}}_{\text{index}}(j \mid \theta_0, \rho_0, \mathcal{O}) = \frac{w_j}{\sum \mathcal{W}}, \quad \text{for } j \in a{:}b, \tag{19}$$

  where $\mathcal{W} = (w_a, \ldots, w_b)$ are the weights associated with the orbit $\mathcal{O}$.

The superscripts "BP" and "MN" also apply to the corresponding extended target distributions. Let ζ denote the reference measure on $\mathbb{A}$, given by the product of Lebesgue measure on the first two components and counting measure on the remaining components. The extended target density with respect to ζ under biased progressive sampling is given by

$$p^{\text{BP}}_{\text{joint}}(z) \;\propto\; e^{-H(\theta_0,\rho_0)} \cdot p_{\text{orbit}}(m, b, \boldsymbol{\ell} \mid \theta_0, \rho_0) \cdot p^{\text{BP}}_{\text{index}}(i \mid \theta_0, \rho_0, m, b, \boldsymbol{\ell}) \tag{20}$$

and the analogous definition $p^{\text{MN}}_{\text{joint}}$ is obtained by replacing $p^{\text{BP}}_{\text{index}}$ with $p^{\text{MN}}_{\text{index}}$. The proportionality symbol reflects that $e^{-H(\theta,\rho)}$ is not necessarily normalized.

For any integer $k \in \mathbb{Z}$, we introduce the shift operator $\mathcal{S}^k$ which acts on $\boldsymbol{\ell}$ as follows:

$$(\mathcal{S}^k \boldsymbol{\ell})_j := \ell_{j-k} \; .$$

This operator maps the micro step-size factor associated with the macro step $[t_j, t_{j+1}]$ to the one associated with $[t_{j-k}, t_{j-k+1}]$. In terms of this shift operator, define the map $\Psi : \mathbb{A} \to \mathbb{A}$ by

$$\Psi(z) \;=\; (\,\Phi_{i,0}|_{\boldsymbol{\ell}}\,(\theta_0, \rho_0), m, b - i, \mathcal{S}^i \boldsymbol{\ell}, -i) \; . \tag{21}$$

This map is carefully designed to have the key properties stated in the lemmas that follow: in particular, Ψ will be shown to be a ζ-preserving involution (Lemma 17) under which the extended target distribution remains invariant (Lemma 21). These properties are essential for understanding the reversibility of WALNUTS.

Lemma 17 *The mapping Ψ is a ζ-preserving involution on $\mathbb{A}$.*

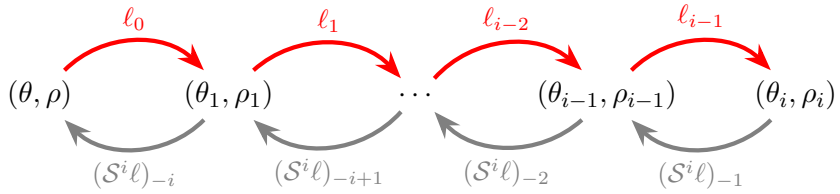
Proof The map Ψ is an involution, since for all $z \in \mathcal{Z}$,

$$\begin{aligned} \Psi \circ \Psi(z) \;&=\; \Psi(\,\Phi_{i,0}|_{\boldsymbol{\ell}}\,(\theta_0, \rho_0), m, b - i, \mathcal{S}^i \boldsymbol{\ell}, -i) \\ &=\; (\,\Phi_{-i,0}|_{\mathcal{S}^i \boldsymbol{\ell}}\,(\theta_i, \rho_i), m, b, \boldsymbol{\ell}, i) \;=\; z \; . \end{aligned}$$

In the last step, we used the identity

$$\Phi_{-i,0}|_{\mathcal{S}^i \boldsymbol{\ell}} \circ \Phi_{0,i}|_{\boldsymbol{\ell}} \;=\; \mathrm{id}_{\mathbb{R}^{2d}} \; ,$$

which follows from the definition of $\Phi_{0,i}|_{\boldsymbol{\ell}}$ in (17) and the fact that $(\mathcal{S}^i \boldsymbol{\ell})_j = \ell_{j-i}$ reindexes the micro step-size factors appropriately for reversal. The following diagram summarizes the composition for $i > 0$, with red arrows indicating the forward trajectory constructed by $\Phi_{0,i}|_{\boldsymbol{\ell}}$ and gray arrows indicating the reversed trajectory constructed by $\Phi_{-i,0}|_{\mathcal{S}^i \boldsymbol{\ell}}$, which returns the state to its original position:



The transition from (θ_0, ρ_0) to (θ_i, ρ_i) is a composition of multiple leapfrog steps, therefore the transition is measure preserving in the first two components. The counting measure on $\mathbb{Z}$ is invariant under reflection. The counting measure on $\mathbb{N}$ is invariant under translation. Thus, Ψ preserves ζ. ■

For notational brevity, we define

$$\begin{aligned} p_j^+ &= p_{\text{micro}}(\ell_j \mid \text{micro}(\theta_j, \rho_j, \mu, M, h, \delta)) \,, \\ p_j^- &= p_{\text{micro}}(\ell_j \mid \text{micro}(\theta_{j+1}, -\rho_{j+1}, \mu, M, h, \delta)) \,. \end{aligned}$$

For any integers $j \le k$, we write

$$\begin{aligned} p_{j:k}^+ &= p_j^+ \cdot p_{j+1}^+ \cdots p_k^+, \\ p_{j:k}^- &= p_j^- \cdot p_{j+1}^- \cdots p_k^-. \end{aligned}$$

If $j > k$, we use the convention that $p_{j:k}^+ = 1$ and $p_{j:k}^- = 1$.

Lemma 18 *The orbit selection kernel of WALNUTS is given by*

$$p_{\text{orbit}}(m, b, \boldsymbol{\ell} \mid \theta_0, \rho_0) = \frac{1}{2^m} \cdot p_{0:b-1}^+ \cdot p_{a:-1}^-.$$

Proof A WALNUTS orbit is generated by first sampling a sequence of Bernoulli variables $B = (B_1, \ldots, B_{m_{\max}})$ and micro step-size factors $\{\ell_j\}_{a_{m_{\max}} \le j < b_{m_{\max}}}$. Given B and $\{\ell_j\}_{a_{m_{\max}} \le j < b_{m_{\max}}}$, the selection of $\mathcal{O} = \mathcal{O}_m$ is deterministic with the NUTS-style doubling procedure. Further, one of the following conditions must hold.

1. $\mathcal{O}_m$ does not satisfy the sub-U-turn condition, but its extension $\mathcal{O}_m^{\text{ext}}$ does.
2. $\mathcal{O}_m$ satisfies the U-turn condition, but none of its sub-orbits do.
3. The maximum number of doublings is reached, i.e., $m = m_{\max}$.

We write this deterministic selection as $\mathcal{O}_m = \text{NUTS}(B, \{\ell_j\}_{a_{m_{\max}} \le j < b_{m_{\max}}})$. Then, the probability of selecting orbit $\mathcal{O}_m$ is

$$p_{\text{orbit}}(\mathcal{O}_m \mid \theta_0, \rho_0) = \sum_{B'} \sum_{\ell'} p(B') \cdot p(\ell' \mid B') \cdot \mathbf{1}\{\text{NUTS}(B', \ell') = \mathcal{O}_m\},$$

where the sums range over all possible values of $B' \in \{0,1\}^{m_{\max}}$ and $\ell' = \{\ell'_j\}_{a_{m_{\max}} \le j < b_{m_{\max}}}$, and where

$$\begin{aligned} p(B') =& \frac{1}{2^{m_{\max}}}, \\ p(\ell' \mid B') =& \prod_{j=0}^{b_{\max}-1} p_{\text{micro}}(\ell'_j \mid \text{micro}(\theta'_j, \rho'_j, \mu, M, h, \delta)) \\ & \cdot \prod_{j=a_{\max}}^{-1} p_{\text{micro}}(\ell'_j \mid \text{micro}(\theta'_{j+1}, -\rho'_{j+1}, \mu, M, h, \delta)), \end{aligned}$$

and (θ'_j, ρ'_j) represents the j-th state in the orbit constructed from B', ℓ'.

The indicator $\mathbf{1}\{\text{NUTS}(B', \ell') = \mathcal{O}_m\}$ is equal to 1 if and only if

$$(B'_1, \ldots, B'_m) = (B_1, \ldots, B_m) \text{ and } \ell'_j = \ell_j \text{ for all } a_m \le j \le b_m - 1.$$

Hence, the probability of selecting $\mathcal{O}_m$ is equal to the probability of matching the first m Bernoulli steps and the corresponding micro step factors:

$$\begin{aligned} \frac{1}{2^m} \cdot \prod_{j=0}^{b_m-1} p_{\text{micro}}(\ell_j \mid \text{micro}(\theta_j, \rho_j, \delta)) \cdot \prod_{j=a_m}^{-1} p_{\text{micro}}(\ell_j \mid \text{micro}(\theta_{j+1}, -\rho_{j+1}, \delta)) \\ = \frac{1}{2^m} \cdot p^+_{0:b-1} \cdot p^-_{a:-1}. \end{aligned}$$

■

Lemma 19 *The index selection kernel for multinomial sampling is given by*

$$p^{\text{MN}}_{\text{index}}(j \mid \theta_0, \rho_0, \mathcal{O}) = \frac{e^{-H(\theta_j, \rho_j)}}{Z(\theta_0, \rho_0, \mathcal{O})} \cdot \begin{cases} \dfrac{p^-_{0:j-1}}{p^+_{0:j-1}} & 0 \le j \le b \\ \dfrac{p^+_{j:-1}}{p^-_{j:-1}} & a \le j < 0 \end{cases}$$

where the normalizing constant is given by

$$Z(\theta_0, \rho_0, \mathcal{O}) = \sum_{j=0}^{b} e^{-H(\theta_j, \rho_j)} \frac{p^-_{0:j-1}}{p^+_{0:j-1}} + \sum_{j=a}^{-1} e^{-H(\theta_j, \rho_j)} \frac{p^+_{j:-1}}{p^-_{j:-1}}.$$

Lemma 19 follows directly from the definitions of $p^{\text{MN}}_{\text{index}}$ in (19) and the weights w_j in (12), (13) and (14).

Lemma 20 *The extended target density with multinomial sampling is invariant under Ψ in (21), i.e., $p^{\text{MN}}_{\text{joint}} \circ \Psi = p^{\text{MN}}_{\text{joint}}$.*

Proof The case $i = 0$ is trivial, as Ψ is the identity. We therefore assume $i > 0$; the case $i < 0$ is analogous and omitted. By Lemmas 18 and 19, the extended target evaluated at z satisfies:

$$\begin{aligned} p^{\mathrm{MN}}_{\mathrm{joint}}(z) &\propto e^{-H(\theta_0,\rho_0)} \cdot \frac{1}{2^m} p^+_{0:b-1} p^-_{a:-1} \cdot \frac{1}{Z(\theta_0,\rho_0,\mathcal{O})} e^{-H(\theta_i,\rho_i)} \frac{p^-_{0:i-1}}{p^+_{0:i-1}} \\ &\propto e^{-H(\theta_0,\rho_0)-H(\theta_i,\rho_i)} \frac{1}{2^m} p^-_{a:i-1} p^+_{i:b-1} \frac{1}{Z(\theta_0,\rho_0,\mathcal{O})} \,. \end{aligned} \tag{22}$$

Now consider the transformed point $z' = \Psi(z)$, which shifts the proposal index i to 0 and the initial index 0 to $-i$, and reindexes the micro step-size factors via the shift operator: $\mathcal{S}^i\boldsymbol{\ell}$ satisfies $(\mathcal{S}^i\boldsymbol{\ell})_j = \ell_{j-i}$. Thus, micro step factors and integration times in the reverse orbit are properly aligned, and hence, we simply write $(\theta_i, \rho_i, \mathcal{O}) = (\theta_i, \rho_i, m, \mathcal{S}^i\boldsymbol{\ell}, b-i)$, which corresponds to the same orbit now viewed from the new initial point (θ_i, ρ_i). Evaluating the extended target at z' yields

$$\begin{aligned} p^{\mathrm{MN}}_{\mathrm{joint}}(z') &\propto e^{-H(\theta_i,\rho_i)} \cdot \frac{1}{2^m} p^+_{i:b-1} p^-_{a:i-1} \cdot \frac{1}{Z(\theta_i,\rho_i,\mathcal{O})} e^{-H(\theta_0,\rho_0)} \frac{p^+_{0:i-1}}{p^-_{0:i-1}} \\ &\propto e^{-H(\theta_0,\rho_0)-H(\theta_i,\rho_i)} \frac{1}{2^m} p^-_{a:-1} p^+_{0:b-1} \frac{1}{Z(\theta_i,\rho_i,\mathcal{O})} \,. \end{aligned} \tag{23}$$

Comparing (22) to (23), to prove $p^{\mathrm{MN}}_{\mathrm{joint}}(z') = p^{\mathrm{MN}}_{\mathrm{joint}}(z)$, it suffices to show

$$Z(\theta_i,\rho_i,\mathcal{O}) \cdot p^-_{0:i-1} \;=\; Z(\theta_0,\rho_0,\mathcal{O}) \cdot p^+_{0:i-1} \,. \tag{24}$$

Let $\mu_j := e^{-H(\theta_j,\rho_j)}$ denote the unnormalized weight at index j. We first expand $Z(\theta_0,\rho_0,\mathcal{O})$:

$$\begin{aligned} Z(\theta_0,\rho_0,\mathcal{O}) \cdot p^+_{0:i-1} &= \Big(\sum_{j=0}^{b} \mu_j \frac{p^-_{0:j-1}}{p^+_{0:j-1}} + \sum_{j=a}^{-1} \mu_j \frac{p^+_{j:-1}}{p^-_{j:-1}} \Big) \cdot p^+_{0:i-1} \\ &= \sum_{j=0}^{i} \mu_j p^-_{0:j-1} p^+_{j:i-1} + \sum_{j=i+1}^{b} \mu_j \frac{p^-_{0:j-1}}{p^+_{i:j-1}} + \sum_{j=a}^{-1} \mu_j \frac{p^+_{j:i-1}}{p^-_{j:-1}} \\ &= \sum_{j=0}^{i-1} \mu_j p^-_{0:j-1} p^+_{j:i-1} + \sum_{j=i}^{b} \mu_j \frac{p^-_{0:j-1}}{p^+_{i:j-1}} + \sum_{j=a}^{-1} \mu_j \frac{p^+_{j:i-1}}{p^-_{j:-1}}. \end{aligned} \tag{25}$$

Here we used that $p^+_{j:k} = 1$ and $p^-_{j:k} = 1$, if $j > k$. Next, we expand $Z(\theta_i,\rho_i,\mathcal{O})$:

$$\begin{aligned} Z(\theta_i,\rho_i,\mathcal{O}) \cdot p^-_{0:i-1} &= \Big(\sum_{j=i}^{b} \mu_j \frac{p^-_{i:j-1}}{p^+_{i:j-1}} + \sum_{j=a}^{i-1} \mu_j \frac{p^+_{j:i-1}}{p^-_{j:i-1}} \Big) \cdot p^-_{0:i-1} \\ &= \sum_{j=i}^{b} \mu_j \frac{p^-_{0:j-1}}{p^+_{i:j-1}} + \sum_{j=0}^{i-1} \mu_j p^-_{0:j-1} p^+_{j:i-1} + \sum_{j=a}^{-1} \mu_j \frac{p^+_{j:i-1}}{p^-_{j:-1}}. \end{aligned} \tag{26}$$

The right-hand sides of (25) and (26) are identical term-by-term. Therefore, (24) holds, and the invariance $p^{\text{MN}}_{\text{joint}} \circ \Psi = p^{\text{MN}}_{\text{joint}}$ follows. ■

Lemmas 17 and 20 verify the conditions of Theorem 1 in the case of WALNUTS with multinomial integration time sampling. Consequently, the associated transition kernel is reversible. We do not state this as a separate theorem, however, since our primary focus is on WALNUTS with biased progressive sampling.

The following result is key to understanding why WALNUTS, like biased progressive HMC and NUTS, always accepts the proposed next state without the need for a Metropolis correction. In the GIST framework (Section 2.3), the acceptance probability for transitioning from one extended state to another is determined by a Metropolis ratio of extended target densities. Lemma 21 shows that this Metropolis ratio is always equal to one, thereby ensuring automatic acceptance and preserving detailed balance without the need to explicitly compute an acceptance probability.

Lemma 21 *The extended target density with biased progressive sampling in* (20) *is invariant under* Ψ *in* (21)*, i.e.,* $p^{\text{BP}}_{\text{joint}} \circ \Psi = p^{\text{BP}}_{\text{joint}}$.

Proof The proof is by induction on the number of doublings m. For the base case $m = 1$, assume $a = 0$, $b = 1$, and $i = 1$. The claim for $i = 0$ is trivial, and the proof for $a = -1$, $b = 0$ follows by symmetry. The orbit consists of two states, (θ_0, ρ_0) and (θ_1, ρ_1), and the orbit and index selection kernels under biased progressive sampling are:

$$p_{\text{orbit}}(m, b, \ell_0 \mid \theta_0, \rho_0) = \frac{1}{2} p_0^+, \quad p_{\text{orbit}}(m, b-1, \ell_0 \mid \theta_1, \rho_1) = \frac{1}{2} p_0^-,$$

$$p^{\text{BP}}_{\text{index}}(1 \mid \theta_0, \rho_0, \mathcal{O}) = \min(1, \frac{e^{-H(\theta_1,\rho_1)} \frac{p_0^-}{p_0^+}}{e^{-H(\theta_0,\rho_0)}}), \quad p^{\text{BP}}_{\text{index}}(-1 \mid \theta_1, \rho_1, \mathcal{O}) = \min(1, \frac{e^{-H(\theta_0,\rho_0)} \frac{p_0^+}{p_0^-}}{e^{-H(\theta_1,\rho_1)}}).$$

Thus, the extended target evaluated at $z = (\theta_0, \rho_0, m, b, \boldsymbol{\ell}, i) = (\theta_0, \rho_0, 1, 1, \ell_0, 1)$ is

$$\begin{aligned} p^{\text{BP}}_{\text{joint}}(\theta_0, \rho_0, 1, 1, \ell_0, 1) &\propto e^{-H(\theta_0,\rho_0)} \cdot p_0^+ \cdot \min(1, \frac{e^{-H(\theta_1,\rho_1)} \frac{p_0^-}{p_0^+}}{e^{-H(\theta_0,\rho_0)}}) \\ &= \min(e^{-H(\theta_0,\rho_0)} \cdot p_0^+,\ e^{-H(\theta_1,\rho_1)} \cdot p_0^-). \end{aligned}$$

Similarly, the extended target evaluated at $z' = \Psi(z)$ is

$$\begin{aligned} p^{\text{BP}}_{\text{joint}}(\theta_1, \rho_1, 1, 0, \ell_0, -1) &\propto e^{-H(\theta_1,\rho_1)} \cdot p_0^- \cdot \min(1, \frac{e^{-H(\theta_0,\rho_0)} \frac{p_0^+}{p_0^-}}{e^{-H(\theta_1,\rho_1)}}) \\ &= \min(e^{-H(\theta_0,\rho_0)} \cdot p_0^+,\ e^{-H(\theta_1,\rho_1)} \cdot p_0^-). \end{aligned}$$

This confirms that the extended target is invariant under Ψ when the orbit contains two states.

Assume that the claim is true when $\mathcal{O}$ has size $2, 2^2, \ldots, 2^m$. We now prove it holds when $\mathcal{O}$ has size 2^{m+1}. There are two cases depending on whether (θ_i, ρ_i) belongs to $\mathcal{O}^{\text{ext}}_m$ or $\mathcal{O}_m$.

Case 1: $(\theta_i, \rho_i) \in \mathcal{O}_m^{\text{ext}}$**.** By (18), the probability of choosing i under biased progressive sampling is

$$\begin{aligned}
p_{\text{index}}^{\text{BP}}(i \mid \theta_0, \rho_0, \mathcal{O}_{m+1}) &= \alpha(\mathcal{W}_m, \mathcal{W}_m^{\text{ext}}) \cdot \frac{w_i}{\sum \mathcal{W}_m^{\text{ext}}} \\
&= \alpha(\mathcal{W}_m, \mathcal{W}_m^{\text{ext}}) \cdot \frac{w_i}{\sum \mathcal{W}_{m+1}} \frac{\sum \mathcal{W}_{m+1}}{\sum \mathcal{W}_m^{\text{ext}}} \\
&= \alpha(\mathcal{W}_m, \mathcal{W}_m^{\text{ext}}) \cdot p_{\text{index}}^{\text{MN}}(i \mid \theta_0, \rho_0, \mathcal{O}_{m+1}) \cdot \frac{\sum \mathcal{W}_{m+1}}{\sum \mathcal{W}_m^{\text{ext}}} \\
&= p_{\text{index}}^{\text{MN}}(i \mid \theta_0, \rho_0, \mathcal{O}_{m+1}) \cdot \min\left(\frac{1}{\sum \mathcal{W}_m^{\text{ext}}}, \frac{1}{\sum \mathcal{W}_m}\right) \cdot \sum \mathcal{W}_{m+1}.
\end{aligned}$$

Thus,

$$\begin{aligned}
&p_{\text{joint}}^{\text{BP}}(\theta_0, \rho_0, m, b, \boldsymbol{\ell}, i) \\
&\quad \propto e^{-H(\theta_0, \rho_0)} \cdot p_{\text{orbit}}(m, b, \boldsymbol{\ell} \mid \theta_0, \rho_0) \cdot p_{\text{index}}^{\text{MN}}(i \mid \theta_0, \rho_0, \mathcal{O}_{m+1}) \\
&\qquad \cdot \min\left(\frac{1}{\sum \mathcal{W}_m^{\text{ext}}}, \frac{1}{\sum \mathcal{W}_m}\right) \cdot \sum \mathcal{W}_{m+1} \\
&\quad \propto p_{\text{joint}}^{\text{MN}}(\theta_0, \rho_0, \mathcal{O}_{m+1}, i) \cdot \min\left(\frac{1}{\sum \mathcal{W}_m^{\text{ext}}}, \frac{1}{\sum \mathcal{W}_m}\right) \cdot \sum \mathcal{W}_{m+1}.
\end{aligned}$$

Similarly, for the transformed point:

$$\begin{aligned}
&p_{\text{index}}^{\text{BP}}(-i \mid \theta_i, \rho_i, \mathcal{O}_{m+1}) \\
&\quad = p_{\text{index}}^{\text{MN}}(-i \mid \theta_i, \rho_i, \mathcal{O}_{m+1}) \cdot \min\left(\frac{1}{\sum \mathcal{W}_m^{\text{ext}}}, \frac{1}{\sum \mathcal{W}_m}\right) \cdot \sum \mathcal{W}_{m+1}.
\end{aligned}$$

and

$$\begin{aligned}
&p_{\text{joint}}^{\text{BP}}(\theta_i, \rho_i, m, b, \boldsymbol{\ell}, -i) \\
&\quad = p_{\text{joint}}^{\text{MN}}(\theta_i, \rho_i, \mathcal{O}, -i) \cdot \min\left(\frac{1}{\sum \mathcal{W}_m^{\text{ext}}}, \frac{1}{\sum \mathcal{W}_m}\right) \cdot \sum \mathcal{W}_{m+1}.
\end{aligned}$$

By Lemma 20, we have

$$p_{\text{joint}}^{\text{MN}}(\theta_0, \rho_0, \mathcal{O}, i) = p_{\text{joint}}^{\text{MN}}(\theta_i, \rho_i, \mathcal{O}, -i),$$

and hence, $p_{\text{joint}}^{\text{BP}}(\theta_0, \rho_0, \mathcal{O}, i) = p_{\text{joint}}^{\text{BP}}(\theta_i, \rho_i, \mathcal{O}, -i)$. This concludes the proof for the case $(\theta_i, \rho_i) \in \mathcal{O}_m^{\text{ext}}$.

Case 2: $(\theta_i, \rho_i) \in \mathcal{O}_m$**.** By (18), we have

$$p_{\text{index}}^{\text{BP}}(i \mid \theta_0, \rho_0, \mathcal{O}_{m+1}) = (1 - \alpha(\mathcal{W}_m, \mathcal{W}_m^{\text{ext}})) \cdot p_{\text{index}}^{\text{BP}}(i \mid \theta_0, \rho_0, \mathcal{O}_m).$$

Applying the induction hypothesis to $\mathcal{O}_m$ gives

$$\begin{aligned}
&e^{-H(\theta_0, \rho_0)} \cdot p_{\text{orbit}}(\mathcal{O}_m \mid \theta, \rho) \cdot p_{\text{index}}^{\text{BP}}(i \mid \theta, \rho, \mathcal{O}_m) \\
&\quad = e^{-H(\theta_i, \rho_i)} \cdot p_{\text{orbit}}(\mathcal{O}_m \mid \theta_i, \rho_i) \cdot p_{\text{index}}^{\text{BP}}(-i \mid \theta_i, \rho_i, \mathcal{O}_m).
\end{aligned}$$

Suppose $\mathcal{O}_m^{\text{ext}}$ is the right half of $\mathcal{O}_{m+1}$, and let b_m be the rightmost index of $\mathcal{O}_m$. (The argument for the other direction is the same.) Then we have

$$p_{\text{orbit}}(\mathcal{O}_{m+1} \mid \theta, \rho) = \frac{1}{2} p_{\text{orbit}}(\mathcal{O}_m \mid \theta, \rho) \cdot \prod_{j=b_m}^{j=b_{m+1}-1} p_j^+,$$

$$p_{\text{orbit}}(\mathcal{O}_{m+1} \mid \theta_i, \rho_i) = \frac{1}{2} p_{\text{orbit}}(\mathcal{O}_m \mid \theta_i, \rho_i) \cdot \prod_{j=b_m}^{j=b_{m+1}-1} p_j^+.$$

Combining these identities confirms that

$$p_{\text{joint}}^{\text{BP}}(\theta_0, \rho_0, \mathcal{O}_{m+1}, i) = p_{\text{joint}}^{\text{BP}}(\theta_i, \rho_i, \mathcal{O}_{m+1}, -i).$$

■

Lemmas 17 and 21 verify the conditions of Theorem 1 for WALNUTS with biased progressive integration time sampling. As a result, the following reversibility result holds.

Theorem 22 *The WALNUTS transition kernel is reversible with respect to the target distribution μ.*

3.5 Pseudocode Implementation of WALNUTS

WALNUTS requires as inputs an unnormalized target density μ, a mass matrix M, a macro step size $h > 0$, a maximum number of doublings $m_{\max}$, and a leapfrog energy error threshold $\delta > 0$. The probability distribution p_{micro}, which governs the sampling of micro step-size reduction factors, is defined internally in terms of the output of the `micro` subroutine (see Listing 6). Complete pseudocode for all components is summarized below and detailed in Appendix A. We emphasize that, unlike NUTS, WALNUTS incorporates variable step-size leapfrog integration as a central component of its design.

The main WALNUTS algorithm is given in Listing 1, which implements orbit construction and biased progressive state selection. Orbit expansion is handled by the `extend-orbit-forward` and `extend-orbit-backward` procedures (Listings 4 and 5, respectively), which generate new segments of the variable step-size leapfrog trajectory along with corresponding weights. The U-turn and sub-U-turn conditions used to terminate orbit growth are checked by the `U-turn` and `sub-U-turn` subroutines (Listings 2 and 3). Finally, the local step size selection is handled by the `micro` subroutine (Listing 6), which adaptively determines the number of micro steps needed to keep the energy error within the specified threshold δ.

A reference implementation of WALNUTS, including a command-line interface, is publicly available at `https://github.com/flatironinstitute/walnuts`.

4. Empirical Evaluations

The results and plots presented below can be reproduced using a Python implementation of WALNUTS, which is available on GitHub under a permissive open-source license.[2]

2. The research code is available at `https://github.com/bob-carpenter/walnuts`.

4.1 Setup for Numerical Experiments

Throughout this and the following sections, unless otherwise specified, WALNUTS is run with a default micro step distribution p_{micro} defined as follows. Given a current state $(\theta, \rho) \in \mathbb{R}^{2d}$, the function call

$$\widetilde{\ell} = \texttt{micro}(\theta, \rho, \mu, M, h, \delta)$$

returns the smallest power of two $\widetilde{\ell} \in 2^{\mathbb{N}}$ such that the energy error of the leapfrog trajectory satisfies $H^+_{\widetilde{\ell}} - H^-_{\widetilde{\ell}} \le \delta$, as defined in (9).

The *randomized two-point* distribution is then defined by:

$$p_{\text{micro}}(\ell \mid \widetilde{\ell}) = \begin{cases} 2/3 & \text{if } \ell = \widetilde{\ell}, \\ 1/3 & \text{if } \ell = \widetilde{\ell} + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{R2P}$$

The corresponding WALNUTS method using this distribution is referred to as *WALNUTS–R2P*.

In contrast, the *deterministic* variant corresponds to the degenerate case:

$$p_{\text{micro}}(\ell \mid \widetilde{\ell}) = \begin{cases} 1 & \text{if } \ell = \widetilde{\ell}, \\ 0 & \text{otherwise.} \end{cases} \tag{D}$$

The corresponding method is referred to as *WALNUTS–D*. In this variant, the step-size reduction factor selected in the forward macrostep must match exactly that obtained in the backward flow. If a mismatch occurs at a given macro interval, the corresponding conditional probability is zero, and all subsequent weights along the orbit are zero. Thus forward–reverse consistency is enforced locally at each macrostep through the orbit weights. This differs from forward–reverse adaptation schemes such as AutoMALA (Kleppe, 2016; Biron-Lattes et al., 2024) and AutoStep (Liu et al., 2024), where consistency is enforced via a global proposal rejection. These two variants will be compared in the numerical studies that follow.

When measuring performance in terms of gradient evaluations, both the forward and backward micro computations are counted, even though backward micro computations may be run concurrently and thus can be coded so as to add negligible wall clock time with parallel processors.

To avoid the looping phenomenon where a trajectory returns to its starting point, we jitter the macro step size throughout. NUTS and WALNUTS rely on the U-turn condition to adaptively truncate their orbit lengths. However, for certain fixed step sizes, this condition can become ineffective: the orbit selection procedure may fail to detect a U-turn and instead expand to the maximum allowed orbit length and severely impair mixing (Bou-Rabee and Oberdörster, 2024a, Figure 4(a) and 5(b)). This looping behavior arises from resonance between the doubling procedure and a natural frequency of the underlying Hamiltonian dynamics, causing the physical orbit length to repeatedly miss the U-turn detection range. A simple and practical remedy is to break the periodicity by jittering the leapfrog step size either once per orbit or at each leapfrog step. As shown empirically, such randomization mitigates looping behavior and improves performance (Bou-Rabee and Oberdörster, 2024a, Figure 5(c) and 5(d)). Throughout our experiments, we uniformly jitter the macro step size by $\pm 20\%$ at each integration step.

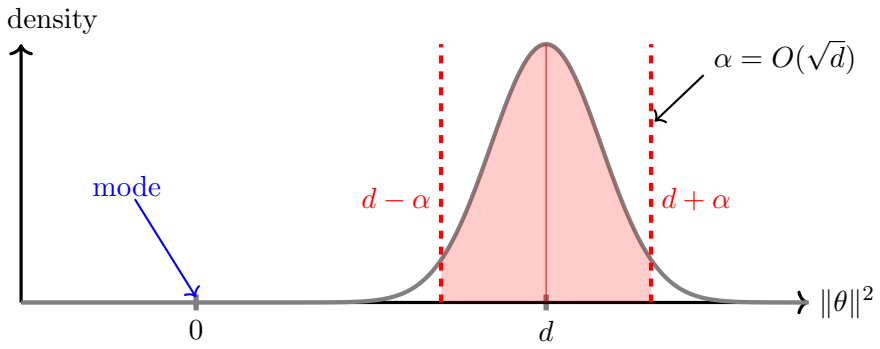


Figure 9: **Concentration of High-Dimensional Gaussian.** Illustration of the concentration of $\|\theta\|^2$ for $\theta \sim \mathcal{N}(0, \mathrm{I}_d)$. The shaded region shows the high-probability shell defined in (27), centered around $\|\theta\|^2 = d$ with width $O(\sqrt{d})$. The mode at $\theta = 0$, corresponding to $\|\theta\|^2 = 0$, lies well outside this concentration region.

4.2 High-Dimensional Gaussian

Geometry of Gaussian concentration. This part evaluates the performance of WALNUTS on a d-dimensional standard Gaussian target with density $\mu(\theta) \propto \exp(-\|\theta\|^2/2)$ for $\theta \in \mathbb{R}^d$. We compare WALNUTS to the standard No-U-Turn Sampler (NUTS) using a fixed step size leapfrog integrator. A $d \times d$-identity mass matrix $M = \mathrm{I}_d$ is used, so that $\rho \sim \mathcal{N}(0, \mathrm{I}_d)$.

It is well known that in high dimensions, the standard Gaussian concentrates its mass in a thin spherical shell around the sphere of radius $\sqrt{d}$. Specifically,

$$\mu(D_\alpha^c) \le 2\exp(-\alpha^2/8d), \quad D_\alpha = \left\{\theta \in \mathbb{R}^d : \left|\|\theta\|^2 - d\right| \le \alpha\right\}, \quad \alpha > 0, \tag{27}$$

as shown, for example, in (Vershynin, 2018, Lemma 1). This implies that most of the probability mass lies within a shell of width $O(\sqrt{d})$ around the sphere of radius $\sqrt{d}$. Figure 9 illustrates this concentration phenomenon.

We consider two initialization strategies: one in which chains start at the mode $\theta = 0$, which lies well outside the concentration region; and one in which chains are initialized from the stationary distribution. Cold starts introduce substantial transient behavior. For HMC-type methods with fixed step size leapfrog integrators, controlling the energy error during this transient phase typically requires step sizes of order $O(d^{-1/2})$. We emphasize that the initialization at $\theta = 0$ is intentionally pathological; while such cold starts can often be avoided in practice, robustness to this type of initialization remains desirable. By contrast, once the chain enters the concentration region, step sizes of order $O(d^{-1/4})$ are sufficient for mixing, as quantified for NUTS in (Bou-Rabee and Oberdörster, 2024a, Theorem 2). We emphasize that this is a sufficient condition for mixing and not a stability condition: leapfrog integration remains stable uniformly in d as long as $h \le 2$ in this example.

To make this precise, let $\theta \in D_\alpha$ and $\rho \sim \mathcal{N}(0, \mathrm{I}_d)$. Then the leapfrog energy error after i steps satisfies

$$\left|H \circ \Phi_h^i(\theta,\rho) - H(\theta,\rho)\right| = \frac{1}{8}h^2 \left|\|\Pi(\Phi_h^i(\theta,\rho))\|^2 - \|\theta\|^2\right| = O(h^2 d^{1/2}), \quad i \in \mathbb{Z}\,, \tag{28}$$

where Π is the projection on the position component. This bound follows from the fact that the leapfrog integrator preserves the modified Hamiltonian $H_h(\theta,\rho) = H(\theta,\rho) - \frac{1}{8}h^2\|\theta\|^2$, i.e., $H_h \circ \Phi_h \equiv H_h$, and that the leapfrog trajectory remains in a shell of comparable radius with high probability.

This estimate reveals two distinct regimes. When the chain is initialized at the mode, $\|\theta\|^2 = 0$, while $\|\Pi(\Phi_h^i(\theta,\rho))\|^2 = O(d)$ with high probability. As a result, the energy error is $O(h^2 d)$ during the transient phase. Once the chain reaches the concentration region D_α, both $\|\theta\|^2$ and $\|\Pi(\Phi_h^i(\theta,\rho))\|^2$ are close to d, and their difference is only $O(d^{1/2})$ with high probability. This reduces the energy error to $O(h^2 d^{1/2})$, allowing for larger step sizes and more efficient sampling.

WALNUTS adapts to this variation in leapfrog energy error through its two-level step size mechanism. The macro step size h is tuned for efficient mixing in the high-density shell, i.e., $h = O(d^{-1/4})$. The micro step size, on the other hand, is adjusted locally via the `micro` routine to ensure that the total energy error over each macro interval remains below a fixed threshold δ. When initialized far from the concentration region, like the mode, WALNUTS selects smaller micro step sizes to compensate for elevated energy error. Once the orbit enters the high-probability shell, the micro step size increases accordingly. In this way, WALNUTS dynamically adapts its step size based on local energy errors, maintaining energy accuracy cheaply.

Numerical results for high-dimensional Gaussian. The first set of simulations, summarized in Figure 10, considers a setting in which NUTS and WALNUTS-R2P chains are initialized at the mode of a d-dimensional standard Gaussian target. In these simulations, NUTS is tuned for the transient phase using a step size $h = d^{-1/2}$, which, as discussed above, is optimal during the transient phase. WALNUTS-R2P is instead tuned for the stationary regime using a macro step size $h = d^{-1/4}$ (and $\delta = 0.3$), corresponding to the optimal (fixed) step size in the stationary regime. From the upper set of plots in Figure 10, it is seen that transient-tuned NUTS arrives at the high-probability shell slightly faster in terms of MCMC iterations than WALNUTS, with the gap growing modestly with increasing dimension.

The mid-level set of plots show the smallest $h\tilde{\ell}^{-1}$ used by WALNUTS in each orbit, which may be interpreted as the smallest micro step size used (modulo macro step size jittering and randomization of step-size factors (R2P)) selected by `micro`. As seen in the plots, WALNUTS uses smaller step sizes during the transient phase and gradually transitions to behavior similar to optimally tuned NUTS in the stationary phase.

The lower set of plots shows the cumulative number of gradient evaluations for WALNUTS and NUTS. As expected, due to the lack of step size adaptivity for NUTS, the benefit of small fixed step size during the transient phase is soon overwhelmed by the cost of such small step sizes in the stationary phase.

Though it is seen that the initial WALNUTS step sizes are somewhat larger than $d^{-1/2}$ (indicated by black line in the mid-level plots) for this choice of δ, and thereby arriving at the high probability shell somewhat slower than NUTS, this choice of δ appears to be reasonable when computational cost is taken into account.

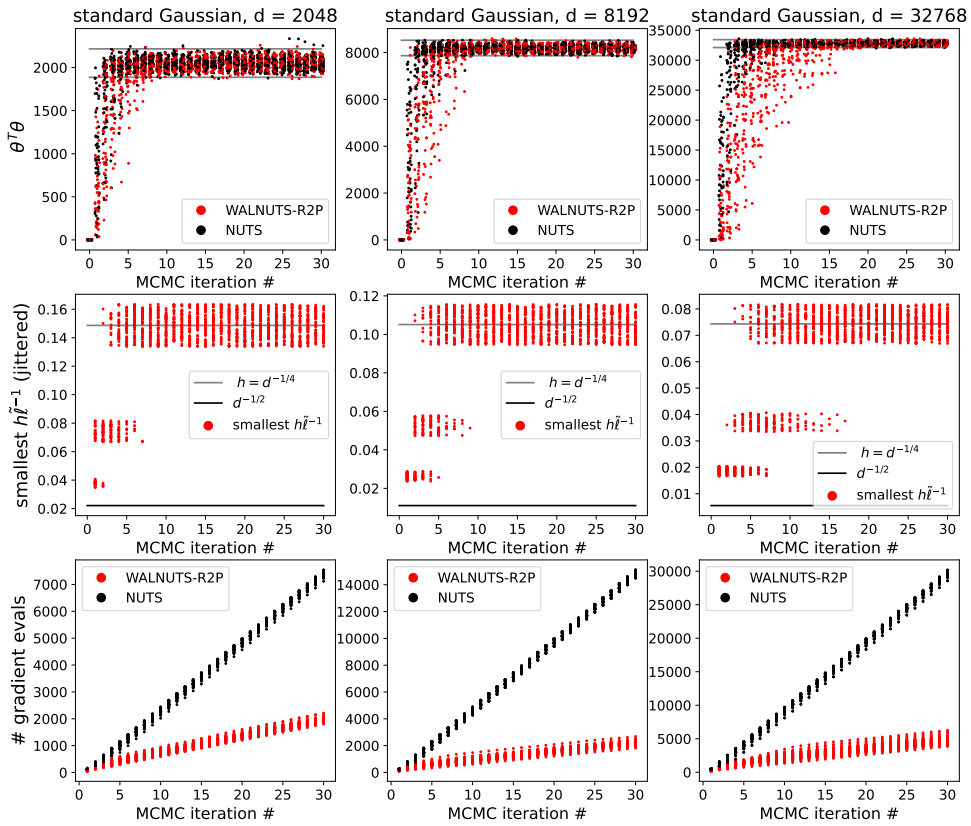


Figure 10: **Cold-Start Behavior and Step Size Adaptation in High-Dimensional Gaussians.** In all cases, 50 independent chains were initialized at the mode $\theta = 0$ and run for 30 MCMC iterations. WALNUTS was tuned for the stationary regime using $h = d^{-1/4}$ and $\delta = 0.3$, while NUTS was tuned specifically for the cold-start regime using $h = d^{-1/2}$. The upper panels show $\|\theta\|^2$ per MCMC iteration for a standard Gaussian target distribution in the indicated dimension d. Red horizontal lines indicate the intervals covering 99% of the probability mass of $\|\theta\|^2$. The middle panels show the smallest value of $h\tilde{\ell}^{-1}$ observed within each orbit (jittered for readability), which may be interpreted as the local micro step size selected by `micro`. The lower panels show the cumulative gradient count after each MCMC iteration for the same simulations.

Figure 11 shows the estimated effective sample size (ESS) per 1000 gradient evaluations for warm-start simulations targeting a standard Gaussian distribution, focusing on ESS for the scalar quantity $\|\theta\|^2$. All three methods—NUTS, WALNUTS-D, and WALNUTS-R2P—used the same macro step size $h = 1.4\, d^{-1/4}$. For this relatively simple target,

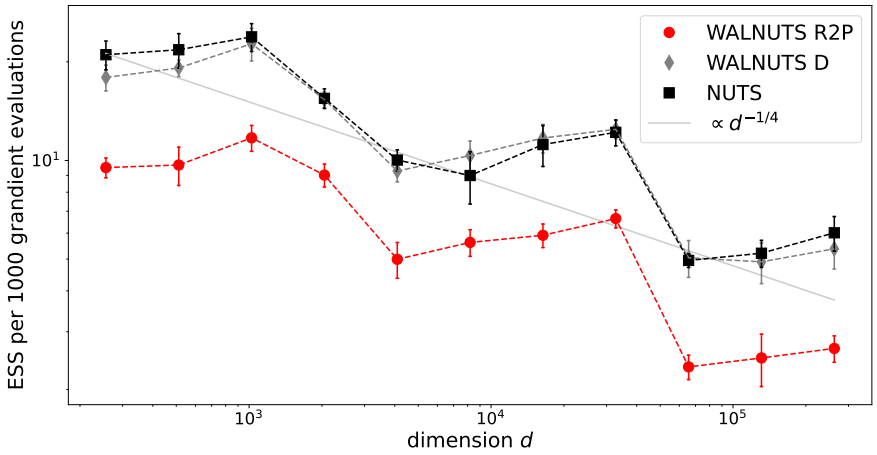


Figure 11: **Efficiency of WALNUTS in High-Dimensional Gaussians.** Effective sample size (ESS) per 1000 gradient evaluations for a standard Gaussian target in varying dimensions, under warm-start initialization. Results for each dimension are based on 10 chains, each run for 1000 iterations. Mean ESS per 1000 gradient evaluations is indicated by symbols; vertical bars show 95% confidence intervals for the exact ESS. All methods used a step size of $h = 1.4\, d^{-1/4}$, which was found by trial and error to be near-optimal for NUTS. Both WALNUTS variants used $\delta = 0.3$.

which is well-suited to fixed step size HMC methods like NUTS, WALNUTS-R2P exhibits somewhat lower efficiency, primarily due to the added cost of frequently applying higher simulation fidelity (i.e., using $\ell = \tilde{\ell}+1$). WALNUTS-D and NUTS show comparable efficiency overall. At higher dimensions, WALNUTS-D and NUTS exhibit comparable efficiency, with differences well within Monte Carlo variability. The step-like pattern in the results reflects dimension-dependent shifts in the number of doublings used by the no-U-turn condition.

To summarize these numerical results, WALNUTS demonstrates substantial promise in two key respects. First, it effectively handles transient regimes without getting stuck near the initialization point, avoiding the cold-start issues that can hinder other methods. Second, for target distributions where standard HMC-based methods like NUTS are known to perform well, WALNUTS achieves comparable efficiency. These results suggest that WALNUTS combines robustness to initialization with competitive performance in well conditioned, high-dimensional settings.

4.3 Neal's Funnel

Geometry of Neal's funnel. Consider the $(d+1)$-dimensional funnel distribution introduced by Neal (Neal, 2003), defined over $\theta = (\omega, x)$ with $\omega \in \mathbb{R}$ and $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, where

$$\omega \sim \mathcal{N}(0, 9), \quad x \mid \omega \sim \mathcal{N}\left(0, \exp(\omega)\, \mathrm{I}_d\right) \ . \tag{29}$$

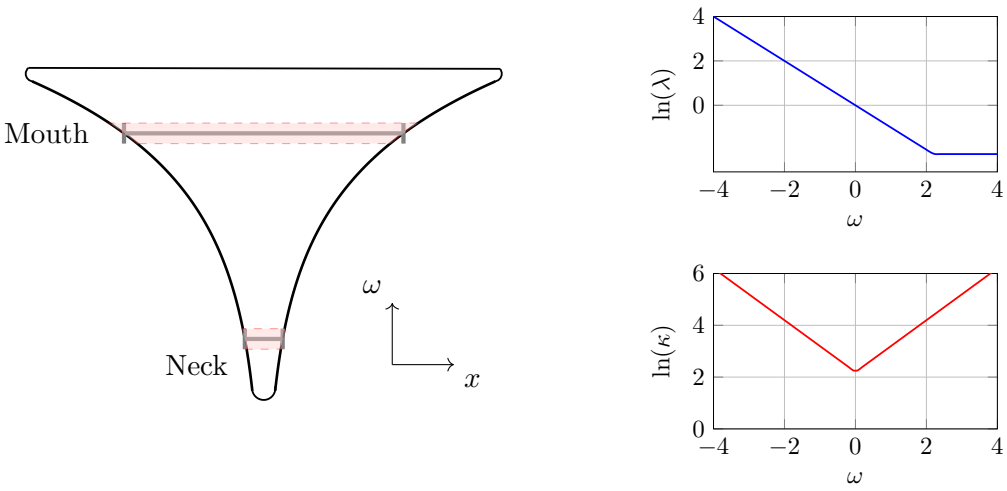


Figure 12: **Curvature, Conditioning, and Scale Invariance in Neal's Funnel.** Illustration of Neal's funnel geometry (left), along with the spectral radius $\lambda(\omega) = \max(1/9, e^{-\omega})$ (top right; see (30)) and the condition number $\kappa(\omega) = 9 \cdot \max(e^{\omega}, e^{-\omega})$ (bottom right; see (31)) plotted as functions of the funnel axis variable ω. The neck and mouth regions have the same shape up to scale: the width at level ω is proportional to $e^{\omega/2}$, with relative rate of change $\frac{d}{d\omega}\text{width}(\omega) = \frac{1}{2} \cdot \text{width}(\omega)$. The translucent bands show a mouth segment and its horizontally rescaled copy overlaid on the neck, highlighting the funnel's local scale invariance: each horizontal slice has the same shape up to scale. This structure makes it difficult for methods such as ensemble samplers to use local geometric cues to guide movement into or out of the neck.

The joint distribution $\mu(\omega, x)$ exhibits a funnel-like geometry, with ω forming the funnel axis and may be thought of as a model problem for Bayesian hierarchical models. Several authors (Betancourt and Girolami, 2015; Kleppe, 2022; Bou-Rabee et al., 2025) have found that HMC-like methods with fixed step sizes may fail to properly explore such target distributions.

This difficulty stems from the model's extreme variation in scale: for large positive ω, the conditional variances $\text{Var}(x_i \mid \omega) = e^{\omega}$ are large and the target is wide and diffuse. This region corresponds to the mouth of the funnel. Conversely, for large negative ω, the conditional variances shrink exponentially, concentrating the mass near a narrow neck. These features are illustrated in Figure 12, where the funnel's "wide mouth" and "narrow neck" encode this dramatic variation in scale.

The challenge of sampling from Neal's funnel can be quantified in terms of the curvature of the potential energy associated with the target distribution given by $U(\omega, x) = -\log \mu(\omega, x)$, and its curvature varies drastically across regions of the domain. From (29), and up to an additive constant independent of (ω, x), this takes the explicit form

$$U(\omega, x) = \frac{\omega^2}{18} + \frac{d}{2}\omega + \frac{1}{2}e^{-\omega}\|x\|^2.$$

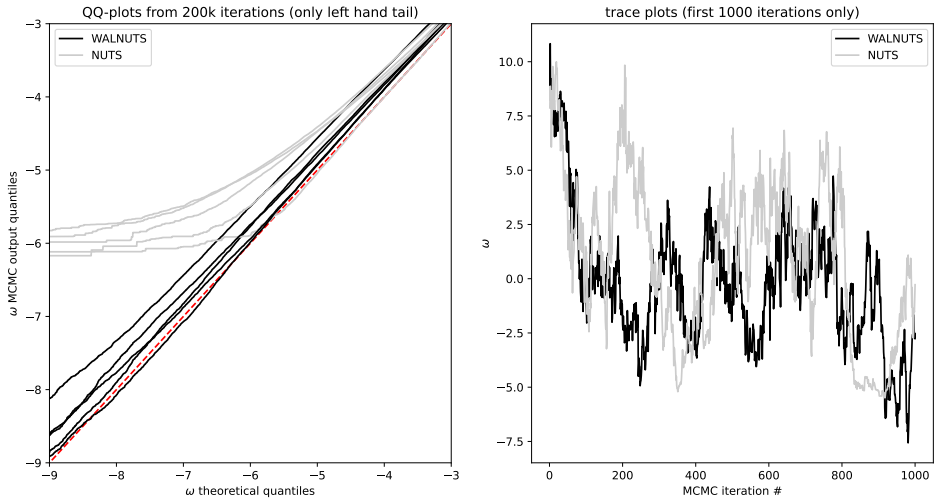


Figure 13: **Tail accuracy and mixing in Neal's funnel: WALNUTS vs. NUTS** The left panel shows QQ-plots (restricted to the far left tail) for WALNUTS-R2P (black) and NUTS (gray), each based on 200,000 iterations. Multiple curves of the same color correspond to independent runs (with different random seeds) of the respective sampler. WALNUTS was run with tuning parameters $\delta = 0.21$ and $h = 0.36$, while NUTS used a step size $h_{\text{NUTS}} = 0.11$, chosen so that the average orbit length matched the number of gradient evaluations used by WALNUTS. With comparable computational cost (NUTS used 104% of WALNUTS's gradient evaluations), WALNUTS successfully explores the full support of the target, while NUTS fails to reach deep into the neck of the funnel. The right panel shows trace plots of ω, illustrating that both samplers mix very slowly. This poor mixing explains the variability in the QQ-plots, particularly in the far tail.

Its curvature varies drastically across regions of the domain. For analytical tractability, we focus on the Hessian evaluated along the funnel axis at $(\omega, 0)$, where it is diagonal,

$$D^2U(\omega, 0) = \text{diag}\left(\tfrac{1}{9}, e^{-\omega}, \ldots, e^{-\omega}\right) \ .$$

The corresponding *spectral radius*, defined as the largest eigenvalue of the Hessian, is given by

$$\lambda(\omega) := \max\left(\tfrac{1}{9}, e^{-\omega}\right) \ . \tag{30}$$

Since the local Lipschitz constant L of ∇U is controlled by the spectral radius of D^2U, this implies that the leapfrog integrator is numerically stable only if

$$h \leq \frac{2}{\sqrt{L}} \ \leq \ 2e^{-\omega/2}.$$

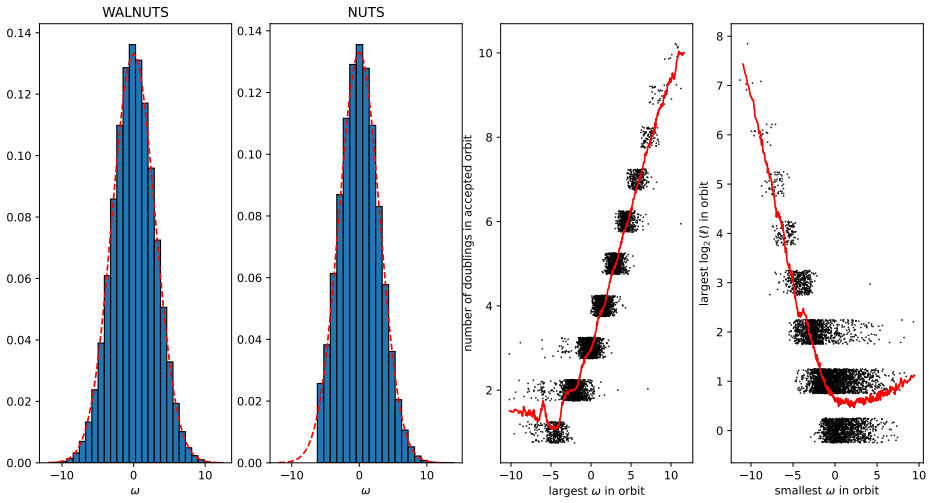


Figure 14: **Diagnostics for WALNUTS and NUTS on Neal's Funnel.** Various diagnostics for WALNUTS-R2P (with $\delta = 0.21$, macro step size $h = 0.36$, 1 million iterations) applied to Neal's funnel distribution (29). The left panel shows a histogram of sampled ω values overlaid with the true $N(0, 9)$ density. The second panel shows the corresponding histogram for NUTS, using 104% of WALNUTS's total gradient evaluations. The third panel plots (thinned and jittered) the number of doublings in accepted WALNUTS orbits versus the largest ω visited during each orbit. The right panel plots (thinned and jittered) the maximum number of micro step size doublings ($\log_2(\ell)$) within each orbit versus the smallest ω visited. In both rightmost panels, nonparametric KNN regression curves are overlaid in red.

Although this calculation is performed along the axis $x = 0$, the full Hessian away from the axis satisfies

$$\lambda_{\max}(D^2U(\omega, x)) \geq \max(1/9, e^{-\omega}),$$

so the stability restriction derived above remains valid (and may be conservative) off-axis. As a result, exploring the neck region ($\omega \ll 0$) requires exponentially small step sizes to maintain stability, while larger step sizes are needed to mix efficiently in the mouth region ($\omega \gg 0$).

Furthermore, the *condition number* of the Hessian matrix evaluated along the funnel axis (i.e., the ratio of the largest to smallest eigenvalue of $D^2U(\omega, 0)$) is given by

$$\kappa(\omega) := 9 \cdot \max(e^{\omega}, e^{-\omega}) \,, \tag{31}$$

which grows exponentially in $|\omega|$. This makes the funnel an extreme example of an ill-conditioned target. Any fixed-step-size implementation of NUTS must adopt a small enough leapfrog step size to avoid numerical divergences in the neck, thereby incurring significant

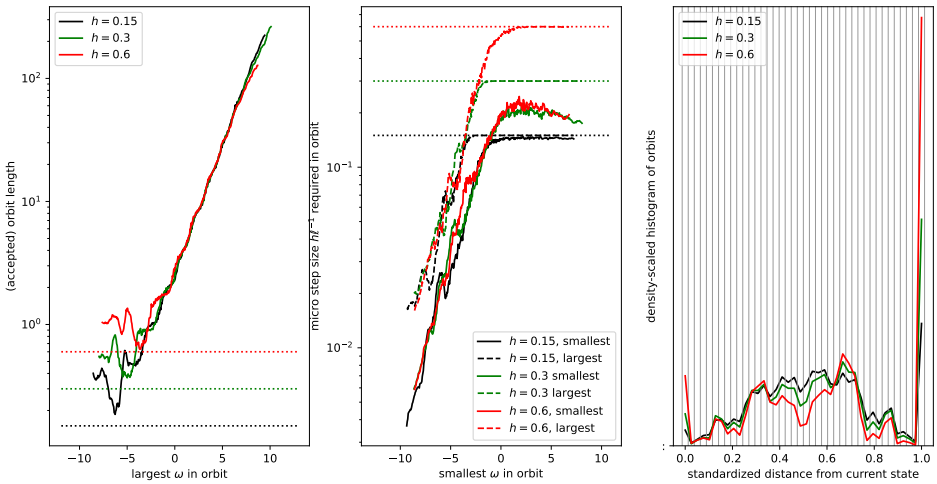


Figure 15: **Influence of macro-step size on WALNUTS in Neal's Funnel.** This figure illustrates the effect of the macro step size h on WALNUTS behavior when applied to Neal's funnel distribution (29), based on 50,000 MCMC iterations for each value of h. The left panel shows KNN regression curves of orbit length (integration time) as a function of the largest ω encountered in the orbit, for three macro step sizes ($h = 0.15$, 0.3, 0.6; shown as dotted lines), with $\delta = 0.1$ fixed throughout. Orbit lengths are broadly similar across values of h, except in the left-hand tail, where many accepted orbits consist of a single integration step. The middle panel shows KNN regression curves of the largest and smallest micro step sizes $h\ell^{-1}$ used in each accepted orbit, plotted against the smallest ω visited. For large ω, smaller macro step sizes act as a ceiling on $h\ell^{-1}$, limiting step size even when larger values would be allowed by the energy error threshold. The right panel shows histograms of the relative time distance between the initial and selected states in each accepted orbit. The distributions are broadly consistent with the idealized triangular law from biased progressive HMC.

inefficiency in the mouth. Conversely, if tuned for the mouth, such a method becomes numerically unstable or inaccurate in the neck. Since both regions carry significant posterior mass, this trade-off cannot be ignored.

These challenges make Neal's funnel a rigorous test for samplers that adapt to local geometry. WALNUTS addresses this issue by dynamically adjusting the step size *within* each leapfrog path. Specifically, WALNUTS first chooses a macro step size h that is suitable for the well-conditioned regions of the target (e.g., the mouth of the funnel), allowing rapid progress when the target is flat. Within each macro step, however, WALNUTS reduces the micro step size until the local energy error falls below the user-specified tolerance δ. This gives rise to a sequence of micro step sizes, each of which may vary depending on the local curvature along the orbit. As a result, the variable step-size leapfrog integrator within

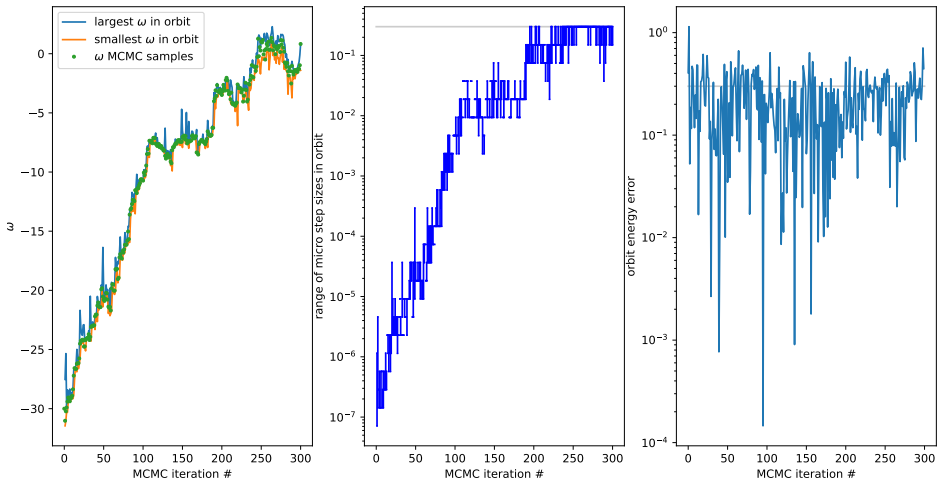


Figure 16: **WALNUTS Diagnostics from a Cold Start Deep in the Neck of Neal's Funnel.** The left panel shows the first 300 WALNUTS samples of ω from a run on Neal's funnel distribution, initialized in a cold state deep in the neck at $\omega = -30$, with $x_i = 0$ for $i = 1, \ldots, 10$. Green dots indicate sampled values; solid lines show the ω-range of each orbit. The middle panel displays the range of micro step sizes $h\ell^{-1}$ used per orbit, with the macro step size $h = 0.3$ shown as a horizontal grey line. The right panel shows the total energy error per orbit, defined as the difference between the largest and smallest Hamiltonian values. The local energy error tolerance $\delta = 0.3$ is indicated by a dashed line.

WALNUTS automatically uses finer resolution when passing through stiff regions such as the narrow neck (where stability conditions are more restrictive), and coarser resolution in flatter regions where larger steps are feasible (see Figure 2).

Remark 23 (Mode of Neal's Funnel) *Analogously to the case of a high-dimensional Gaussian, the potential energy $U(\omega, x) = -\log \mu(\omega, x)$ attains its minimum at a mode that lies in a region of extremely low posterior mass. Completing the square in U, we find that the unique minimizer is given by*

$$(\omega^*, x^*) = \left(-\tfrac{9d}{2}, 0\right) \ .$$

Indeed, expanding the expression for U,

$$U(\omega, x) = \frac{\omega^2}{18} + \frac{\|x\|^2}{2e^\omega} + \frac{d}{2}\omega + \mathit{const}\,,$$

we see that U is minimized at $x = 0$ and $\omega = -\frac{9d}{2}$. This mode lies deep in the neck of the funnel, far in the lower tail of the marginal $\omega \sim \mathcal{N}(0, 9)$, and hence, in a region with

negligible posterior mass. Much like the mode of a high-dimensional standard Gaussian, it is not representative of typical samples from the target.

Numerical results for Neal's funnel. Figure 13, left panel, shows QQ-plots of the marginal distribution of ω based on MCMC output using either WALNUTS-R2P or NUTS, with comparable computational budgets. WALNUTS provides a significantly more accurate representation of the far left tail, demonstrating its ability to explore high-curvature regions deep in the neck of the funnel. However, both methods exhibit slow mixing, as evidenced by the trace plots in the right panel of Figure 13. Notably, the poor mixing persists across the entire support of the target distribution, suggesting that the orbit length selection mechanism in NUTS (and inherited by WALNUTS) tends to produce trajectories that are too short on average. Addressing this issue is an important direction for future work.

Figure 14 presents diagnostic information from a warm-start run of WALNUTS applied to the funnel distribution (29), using 1 million MCMC iterations with parameters $\delta = 0.21$ and macro step size $h_0 = 0.36$, selected during an initial warmup phase as described in Appendix C. The far-left panel shows a histogram of sampled ω values, which indicates no apparent difficulty exploring the left-hand tail of the distribution (see also Figure 13). For comparison, the second panel from the left shows the corresponding histogram for NUTS, run at similar computational cost (104% of the gradient evaluations used by WALNUTS), which reveals a clear failure to explore the left-hand tail.

The second panel from the right displays the number of orbit doublings used in each accepted orbit, plotted against the largest value of ω visited during that orbit. This shows that WALNUTS adapts by generating longer trajectories in the broad mouth of the funnel (i.e., regions with large ω), where the target is flatter and longer integration times are beneficial.

The rightmost panel of Figure 14 shows the largest number of micro step size halvings (i.e., $\log_2(\ell)$) used within each orbit, plotted against the smallest value of ω visited in that orbit. It is evident that WALNUTS automatically selects smaller step sizes in the narrow neck of the funnel (i.e., regions with small ω), where stability constraints are more restrictive, while using larger step sizes in the broad mouth where the geometry is more forgiving.

Figure 15 examines how the macro step size h influences orbit (integration time) length, micro step sizes, and the potential accumulation of local integration errors. The left panel shows KNN regressions of orbit length as a function of the largest ω visited within each orbit. For large values of ω, all considered macro step sizes result in similar orbit lengths. In the narrow neck of the funnel, orbit lengths vary because they often consist of only a single integration step, and are therefore bounded below by the macro step size.

The middle panel of Figure 15 shows KNN regressions of the largest and smallest micro step sizes used by `micro` in each orbit, plotted as a function of the smallest ω visited within that orbit. For small values of ω, all macro step sizes yield similar results, since the upper bound imposed by the macro step size on the micro step sizes is rarely active. As ω increases, however, the results begin to diverge because the macro step size increasingly constrains the allowable micro step sizes.

The rightmost panel of Figure 15 presents histograms of the relative orbit time distance between the initial state (θ_0, ρ_0) and the selected state $(\tilde{\theta}, \tilde{\rho})$, normalized by the total accepted orbit time. In the idealized setting of biased progressive HMC with exact Hamiltonian

simulation and fixed orbit length, this statistic follows a symmetric triangular distribution centered at 0.5 (see Theorem 8). The empirical histograms are broadly consistent with this behavior and do not exhibit signs of degeneracy. The peaks at standardized distances 0 and 1 are attributable to orbits consisting of a single integration step, which occur more frequently as the macro step size increases.

Figure 16 shows a single cold-start simulation consisting of 300 WALNUTS MCMC iterations, initialized at $\omega = -30$ (i.e., 10 standard deviations below the mean) with $x_i = 0$ for $i = 1, \ldots, 10$. The left panel displays the MCMC samples as green dots, along with the extent of each orbit in the ω-direction. The sampler has no difficulty moving through the highly ill-conditioned regions encountered early in the simulation. The middle panel shows the range of micro step sizes used in each orbit, again indicating that WALNUTS adapts as expected. The right panel plots the total energy error per orbit, defined as the difference between the largest and smallest Hamiltonian values, suggesting that energy errors do not exhibit problematic constructive accumulation.

The choice of initial configuration is motivated by the fact that the funnel distribution has a single mode at $\omega = -45$ and $x_i = 0$ for all i (see Remark 23). While there is no theoretical obstacle to starting the chain at the mode, extrapolation from Figure 16 suggests that doing so would require micro step sizes near machine precision and would entail considerable computational effort.

This example supports the conclusion that the proposed locally adaptive step size procedure functions as intended, successfully identifying regions that require smaller step sizes. As discussed above, however, orbit-length selection remains a challenge for this model: NUTS-based strategies lead to slow mixing, and WALNUTS is not immune to this issue. While this suggests that further refinement of orbit-length selection methods, particularly for multiscale target distributions, could improve performance, such developments are beyond the scope of the present work (more on this point in Section 5 below).

4.4 The Stock-Watson model

As a real-world example, we consider the Stock and Watson inflation rate model (Stock and Watson, 2007). Quarterly US inflation rates $y_t,\ t = 1, \ldots, T$ are modeled using three latent, nonlinearly coupled random walk factors (z_t, x_t, τ_t) as follows:

$$
\begin{aligned}
z_t | z_{t-1}, \sigma &\sim \mathcal{N}(z_{t-1}, \sigma^2),\ t = 2, \ldots, T-1, && (32)\\
x_t | x_{t-1}, \sigma &\sim \mathcal{N}(x_{t-1}, \sigma^2),\ t = 2, \ldots, T, && (33)\\
\tau_t | \tau_{t-1}, z_{t-1} &\sim \mathcal{N}(\tau_{t-1}, \exp(z_{t-1})),\ t = 2, \ldots, T, && (34)\\
y_t | \tau_t, x_t &\sim \mathcal{N}(\tau_t, \exp(x_t)),\ t = 1, \ldots, T. && (35)
\end{aligned}
$$

The description of the model is completed with the prior $\sigma^{-2} \sim \text{Gamma}(5, 0.5)$, using the rate parameterization. The data set used is the same as in (Kleppe, 2022), consisting of $T = 252$ observations between 1955 (first quarter) and 2018 (first quarter). Figure 17 provides posterior quantiles of the three latent processes, and it is seen that the balance between persistent volatility (i.e., $\exp(z_t)$) and transient volatility (i.e., $\exp(x_t)$) shows substantial temporal variation (see Stock and Watson, 2007, who first pointed out these effects).

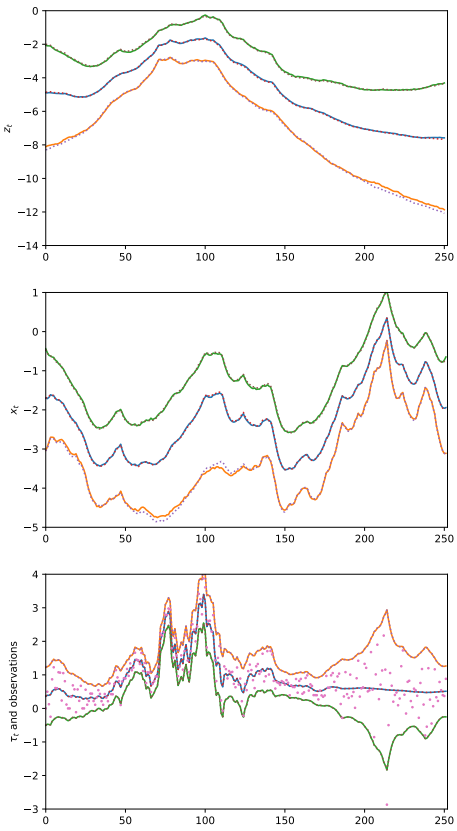


Figure 17: **Posterior distributions of the latent factors (z, x, τ) under the Stock Watson model.** The top panel shows the 0.05, 0.5, and 0.95 quantiles of the persistent volatility factor z_t: solid lines correspond to WALNUTS, and dotted lines to NUTS. The middle panel presents the same quantiles for the transient volatility factor x_t. The bottom panel displays the posterior median of τ_t along with the bands $\tau_t \pm 2\exp(0.5x_t)$ (medians plotted), representing a scale-aware uncertainty interval. Observations y_t are shown as dots. Results from WALNUTS-D are nearly identical and omitted for clarity.

Due to the nonlinear coupling of the latent factors, the posterior distribution of this model is known to exhibit substantial non-Gaussian structure, including pronounced funnel-like geometries. To facilitate sampling, we reparameterize the model in terms of the innovations of the latent processes: $([z_1, (z_2 - z_1)\sigma^{-1}, \ldots, (z_{T-1} - z_{T-2})\sigma^{-1}], [x_1, (x_2 - x_1)\sigma^{-1}, \ldots, (x_T - x_{T-1})\sigma^{-1}], [\tau_1, (\tau_2 - \tau_1)\exp(-0.5z_1), \ldots, (\tau_T - \tau_{T-1})\exp(-0.5z_{T-1})], \log\sigma^2)$. This transfor-

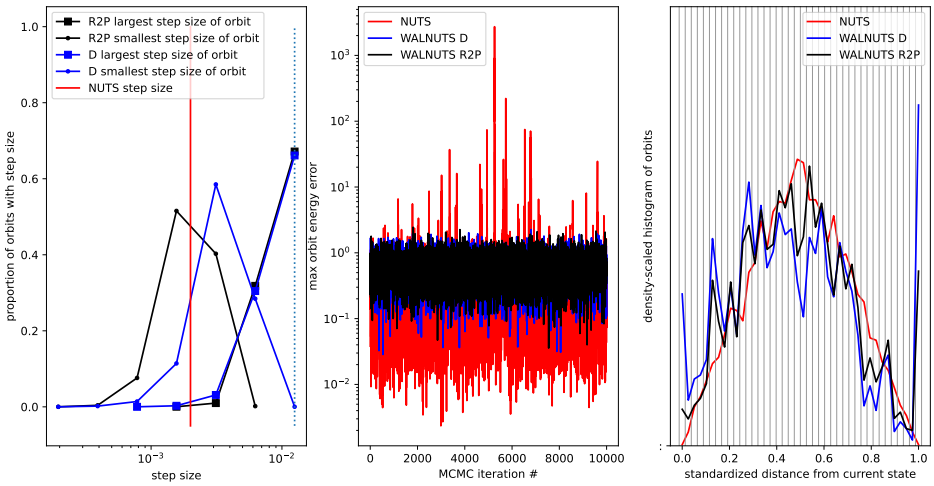


Figure 18: **Distribution of step sizes and orbit energy errors for the Stock-Watson model.** The left panel shows the distribution of the largest (dots) and smallest (squares) micro step sizes used in each WALNUTS orbit (both R2P and D variants). The red vertical line indicates the NUTS step size, while the dotted line marks the largest micro step size considered by WALNUTS ($h/8$). The middle panel presents the total energy error per orbit, computed as $\max_j H(\theta^{(j)}, \rho^{(j)}) - \min_j H(\theta^{(j)}, \rho^{(j)})$, for NUTS and both versions of WALNUTS. Computational costs are comparable between WALNUTS-R2P and NUTS, while WALNUTS-D requires approximately 60% of the gradient evaluations of the former two. The right panel displays the integration time statistic normalized by orbit length. These distributions do not raise any concerns. Note that the smaller step size used by NUTS results in more frequent orbit doublings, which explains the more triangular-like distribution of this statistic.

mation yields a posterior that can be sampled using an identity mass matrix. However, it is important to emphasize that the transformation does not eliminate the underlying nonlinearities. In particular, this parametrization of the model is known to require very small step sizes when using fixed step size HMC, motivating more elaborate preconditioning strategies as explored in (Kleppe, 2019).

Posterior summaries from runs of 10,000 iterations of NUTS (with fixed step size $h = 0.002$) and WALNUTS (with local error tolerance $\delta = 0.3$, macro step size $h = 0.1$, and a minimum of 8 micro-steps per macro-step, enforced by starting the loop in `micro` at index 3 instead of 0) are shown in Figure 17. These parameter settings result in approximately equal computational cost. At first glance, the posterior estimates produced by NUTS and WALNUTS appear similar.

However, Figure 18 reveals important differences in the energy error diagnostics. The energy errors from NUTS exhibit a substantial number of large deviations (e.g., greater than

2), indicating that many transitions diverge and that numerical instability is occurring along certain trajectories. This suggests that some regions of posterior support are effectively inaccessible to NUTS at this step size, even though such pathologies are not evident from the trace plots in Figure 17. A practitioner encountering such behavior would rightly be concerned about convergence.

In contrast, WALNUTS shows no such issues: energy errors remain stable across iterations, even though only local error control is enforced. The left panel of Figure 18 shows the distribution of the largest and smallest micro step sizes used by the two variants of WALNUTS. *Remarkably, despite frequently using step sizes larger than NUTS, WALNUTS maintains stable energy errors by selectively using smaller steps only where needed.*

Crucially, without a stable reference such as WALNUTS, there would be no principled way to determine whether the NUTS samples are in fact accurate, or whether they are implicitly truncating the smaller-scale regions of the posterior. A practitioner encountering repeated large energy errors of this magnitude, particularly when concentrated in geometrically challenging regions, would therefore have legitimate reason to question convergence, even if standard trace plots appear benign.

5. Conclusion and Outlook

WALNUTS introduces a robust and flexible locally adaptive variant of HMC that performs well in difficult sampling problems where state-of-the-art methods like NUTS often struggle. Despite its locally adaptive nature, WALNUTS is not significantly more expensive than NUTS for well-conditioned targets, especially when warmup is appropriately tuned. When NUTS works, it works; but WALNUTS is designed for when it doesn't. It offers a principled fallback that expands the range of models that can be reliably sampled without manual tuning or aggressive preconditioning. In the near term, WALNUTS can be integrated into probabilistic programming languages such as Stan, PyMC, or NumPyro, where its plug-and-play adaptivity and robustness offer practical benefits with minimal user tuning.

Importantly, the error-controlled adaptive integration scheme at the heart of WALNUTS is not specific to NUTS. It is a general-purpose tool that can be incorporated into a wide variety of HMC-type methods, including standard HMC, randomized HMC (Bou-Rabee and Sanz-Serna, 2017; Deligiannidis et al., 2021; Bou-Rabee and Eberle, 2022) (and other Hamiltonian-based PDMPs (Chevallier et al., 2025)), kinetic Langevin samplers (Bou-Rabee and Oberdörster, 2024b), and generalized HMC variants (Turok et al., 2025). The approach we present here also opens up new directions for developing Hamiltonian-based samplers that remain robust and efficient even in challenging inference settings where the local geometry of the target distribution varies substantially. Below, we outline several promising directions for future work.

- **Comparisons to non-reversible step-size tuning.** A natural direction for future work is a systematic comparison between reversible within-trajectory adaptivity, as developed in WALNUTS, and non-reversible step-size adaptation procedures. Understanding the relative strengths, limitations, and regimes of effectiveness of these complementary approaches—reversible versus non-reversible step-size tuning—would help clarify the broader methodological landscape of adaptive HMC. This includes both empirical benchmarking and theoretical analysis of stability, robustness, and mixing

behavior; see, for example, (Kleppe, 2016, 2022) for related work on non-reversible step-size adaptation strategies.

- **Mass matrix adaptivity.** A natural extension is to introduce local adaptivity into the mass matrix, allowing it to vary with position to better capture heterogeneous curvature in the target distribution. We emphasize, however, that this setting is substantially more delicate than scalar step-size adaptivity. Although step sizes are one-dimensional and admit a natural ordering that enables threshold-based selection rules, matrix-valued parameters lack such an ordering structure. Extending WALNUTS-style local adaptivity to non-scalar parameters would therefore require different tools and should be viewed as a longer-term research direction rather than a direct consequence of the present framework.

  Nonetheless, ideas from Riemannian HMC (Girolami and Calderhead, 2011) and recent advances in adaptive mass matrix tuning (Kleppe, 2016; Whalley et al., 2024; Hird and Livingstone, 2023; Tran and Kleppe, 2024) suggest possible avenues for incorporating position-dependent preconditioning while maintaining computational tractability. Any such extension would aim to preserve the simplicity and efficiency of WALNUTS by avoiding expensive full Hessian evaluations. Another promising direction is the use of Adam-like adaptive mechanisms within HMC (Leimkuhler et al., 2025).

- **Ensemble-based approaches.** Another approach to preconditioning WALNUTS is to run an ensemble of chains, following the strategy of affine-invariant methods (Foreman-Mackey et al., 2013; Goodman and Weare, 2010), including the recently introduced affine-invariant HMC variant (Chen, 2025). These methods leverage the positions of parallel chains to adapt proposals to the geometry of the target distribution. Global affine invariance is not sufficient to handle highly multiscale or hierarchical targets such as Neal's funnel where local structure can vary dramatically across regions. Still, ensemble-based strategies are promising. They provide a natural way to estimate low-rank structure and local scaling, which might be hard to detect with a single locally adaptive chain. Combining these techniques with WALNUTS's local step size adaptation could yield samplers that better capture local geometry.

- **Improving the U-Turn Diagnostic.** Another important direction is to revisit the U-turn diagnostic, which remains a heuristic inherited from standard NUTS. A more principled approach to deciding when and how to truncate orbit construction could improve both robustness and sampling efficiency. We emphasize, however, that poor mixing in challenging models such as Neal's funnel is not solely an orbit-length issue. It is also closely related to high-dimensional concentration and energy-level separation effects. In d dimensions, the kinetic energy after momentum refreshment is typically distributed like a χ^2_d random variable and therefore concentrates sharply around its mean with fluctuations of order $O(\sqrt{d})$.

  As a result, each HMC update can typically exchange only $O(\sqrt{d})$ energy between kinetic and potential components. In contrast, Neal's funnel spans an $O(d)$ range of potential energies, creating what is commonly described as an intrinsic *entropic barrier* (Yao et al., 2022). Thus, modifying orbit-length criteria alone cannot fundamentally

eliminate diffusive mixing across energy levels. Nonetheless, Figure 13 illustrates that orbit truncation heuristics can materially influence exploration behavior in practice.

The left panel shows that WALNUTS more accurately captures the far left tail of the marginal distribution of ω, demonstrating its ability to explore high-curvature regions deep in the neck of the funnel. However, the trace plots in the right panel reveal slow mixing for both WALNUTS and NUTS, with poor performance persisting across the entire support. This suggests that the orbit length selection mechanism often terminates integration prematurely. Addressing this issue by developing alternatives to the U-turn diagnostic one is an important direction for future work.

Taken together, these directions suggest the potential for a new class of locally adaptive HMC methods that offer improved robustness across a broad range of challenging scenarios, including models with extreme anisotropy, funnel geometries, or high dimensionality. From an empirical perspective, this motivates more systematic benchmarking including comparisons with alternative micro-step selection criteria. From a theoretical perspective, important open questions include developing quantitative convergence or mixing-time bounds for WALNUTS and identifying regimes in which such adaptation provably improves robustness or efficiency.

We view these questions as complementary to the present contribution, which focuses on establishing a rigorous and practically implementable framework for locally adaptive step-size selection within leapfrog trajectories, and on demonstrating its effectiveness in representative multiscale settings where fixed step-size methods are ineffective.

Acknowledgments

We thank the Editor, Stephan Mandt, and three anonymous reviewers for their thoughtful comments, which led to numerous improvements to the paper. We are also grateful to Stefan Oberdörster and Jimmy Huy Tran for their helpful comments and suggestions. Kleppe acknowledges support from Finansmarkedsfondet, grant #337601. Bou-Rabee was partially supported by NSF grant No. DMS-2111224.

Appendix A. Pseudocode Implementation of WALNUTS

`WALNUTS`$(\theta, \mu, M, h, m_{\max}, \delta)$

Inputs:	$\theta \in \mathbb{R}^d$	initial state
	$\mu : \mathbb{R}^d \to (0, \infty)$	unnormalized target density
	$M \in \mathbb{R}^{d \times d}$	mass matrix
	$h > 0$	macro step size
	$m_{\max} \in \mathbb{N}$	maximum number of doublings
	$\delta > 0$	maximum energy error
Return:	$\tilde{\theta} \in \mathbb{R}^d$	next state

$\rho \sim \mathcal{N}(0, M)$
$(\tilde{\theta}, \tilde{\rho}) = (\theta_0, \rho_0) = (\theta, \rho)$
$w_0 = \mu(\theta_0)\ e^{-\frac{1}{2}\rho_0^\intercal M^{-1} \rho_0}$
$\mathcal{O} = ((\theta_0, \rho_0))$
$\mathcal{W} = (w_0)$
$B \sim \text{Unif}(\{0, 1\}^{m_{\max}})$
for i from 1 to $m_{\max}$:
    $\mathcal{O}^{\text{old}} = \mathcal{O} = ((\theta_a, \rho_a), \ldots, (\theta_b, \rho_b))$, $\mathcal{W}^{\text{old}} = \mathcal{W} = (w_a, \ldots, w_b)$
    if $B_i = 1$:
        $\mathcal{O}^{\text{ext}}, \mathcal{W}^{\text{ext}} =$ `extend-orbit-forward`$(\theta_b, \rho_b, w_b, \mu, M, h, \delta, 2^{i-1})$
        $\mathcal{O} = \mathcal{O}^{\text{old}} \odot \mathcal{O}^{\text{ext}}$, $\mathcal{W} = \mathcal{W}^{\text{old}} \odot \mathcal{W}^{\text{ext}}$
    else:
        $\mathcal{O}^{\text{ext}}, \mathcal{W}^{\text{ext}} =$ `extend-orbit-backward`$(\theta_a, \rho_a, w_a, \mu, M, h, \delta, 2^{i-1})$
        $\mathcal{O} = \mathcal{O}^{\text{ext}} \odot \mathcal{O}^{\text{old}}$, $\mathcal{W} = \mathcal{W}^{\text{ext}} \odot \mathcal{W}^{\text{old}}$
    if `sub-U-turn`$(\mathcal{O}^{\text{ext}}, M)$:
        break
    $u \sim \text{Unif}((0, 1))$
    if $u \leq \dfrac{\sum \mathcal{W}^{\text{ext}}}{\sum \mathcal{W}^{\text{old}}}$:
        $(\tilde{\theta}, \tilde{\rho}) \sim \text{categorical}\left(\mathcal{O}^{\text{ext}}, \mathcal{W}^{\text{ext}}\right)$
    if `U-turn`$(\mathcal{O}, M)$:
        break
return $\tilde{\theta}$

Listing 1: *WALNUTS algorithm with biased progressive state selection.*

`U-turn`$(\mathcal{O}, M)$

Inputs: $\mathcal{O} = \big((\theta^{\text{left}}, \rho^{\text{left}}), \ldots, (\theta^{\text{right}}, \rho^{\text{right}})\big)$ orbit, with $|\mathcal{O}|$ a power of 2

return $(\rho^{\text{right}})^\top M^{-1}(\theta^{\text{right}} - \theta^{\text{left}}) < 0$ or $(\rho^{\text{left}})^\top M^{-1}(\theta^{\text{right}} - \theta^{\text{left}}) < 0$

Listing 2: *Check if the orbit $\mathcal{O}$ satisfies the U-turn condition in Definition 10.*

`sub-U-turn`$(\mathcal{O}, M)$

Inputs: $\mathcal{O} = \mathcal{O}^{\text{left}} \odot \mathcal{O}^{\text{right}}$ orbit, with $|\mathcal{O}^{\text{left}}| = |\mathcal{O}^{\text{right}}|$ a power of 2

if length$(\mathcal{O}) < 2$: return `False`

return `U-turn`$(\mathcal{O}, M)$ or `sub-U-turn`$(\mathcal{O}^{\text{left}}, M)$ or `sub-U-turn`$(\mathcal{O}^{\text{right}}, M)$

Listing 3: *Check if the orbit $\mathcal{O}$ satisfies the sub-U-turn condition. The function* ***`U-turn`*** *is given in Listing 2.*

`extend-orbit-forward`$(\theta_b, \rho_b, w_b, \mu, M, h, \delta, L)$

Inputs:	$(\theta_b, \rho_b) \in \mathbb{R}^{2d}$	initial position, momentum
	$w_b \in \mathbb{R}_{\geq 0}$	initial weight
	$\mu : \mathbb{R}^d \to (0, \infty)$	unnormalized target density
	$M \in \mathbb{R}^{d \times d}$	mass matrix
	$h > 0$	macro step size
	$\delta > 0$	maximum energy error
	$L \in \mathbb{N}$	number of macro steps
Return:	$\mathcal{O}^{\text{ext}} \in \big(\mathbb{R}^{2d}\big)^L$	orbit
	$\mathcal{W}^{\text{ext}} \in (\mathbb{R}_{\geq 0})^L$	weights

for i from $b+1$ to $b+L$:

$$\ell \sim p_{\text{micro}}(\cdot \mid \text{micro}(\theta_{i-1}, \rho_{i-1}, \mu, M, h, \delta))$$

$$(\theta_i, \rho_i) = \Phi^{\ell}_{h\ell^{-1}}(\theta_{i-1}, \rho_{i-1})$$

$$w_i = \frac{\mu(\theta_i) e^{-\frac{1}{2}(\rho_i)^\intercal M^{-1} \rho_i}}{\mu(\theta_{i-1}) e^{-\frac{1}{2}(\rho_{i-1})^\intercal M^{-1} \rho_{i-1}}} \frac{p_{\text{micro}}(\ell \mid \text{micro}(\theta_i, -\rho_i, \mu, M, h, \delta))}{p_{\text{micro}}(\ell \mid \text{micro}(\theta_{i-1}, \rho_{i-1}, \mu, M, h, \delta))} w_{i-1}$$

return $\mathcal{O}^{\text{ext}} = ((\theta_{b+1}, \rho_{b+1}), (\theta_{b+2}, \rho_{b+2}), \ldots, (\theta_{b+L}, \rho_{b+L}))$ and
$\mathcal{W}^{\text{ext}} = (w_{b+1}, w_{b+2}, \ldots, w_{b+L})$

Listing 4: *Generates an orbit of L macro steps and associated weights over the index range $(b+1)$:$(b+L)$.*

`extend-orbit-backward` $(\theta_a, \rho_a, w_a, \mu, M, h, \delta, L)$

Inputs:	$(\theta_a, \rho_a) \in \mathbb{R}^{2d}$	initial position, momentum
	$w_a \in \mathbb{R}_{\geq 0}$	initial weight
	$\mu : \mathbb{R}^d \to (0, \infty)$	unnormalized target density
	$M \in \mathbb{R}^{d \times d}$	mass matrix
	$h > 0$	macro step size
	$\delta > 0$	maximum energy error
	$L \in \mathbb{N}$	number of macro steps
Return:	$\mathcal{O}^{\text{ext}} \in \left(\mathbb{R}^{2d}\right)^L$	orbit
	$\mathcal{W}^{\text{ext}} \in \mathbb{R}^L$	weights

for i from $a-1$ down to $a-L$:

$$\ell \sim p_{\text{micro}}(\cdot \mid \text{micro}(\theta_{i+1}, -\rho_{i+1}, \mu, M, h, \delta))$$

$$(\theta_i, \rho_i) = \Phi^{\ell}_{-h\ell^{-1}}(\theta_{i+1}, \rho_{i+1})$$

$$w_i = \frac{\mu(\theta_i) e^{-\frac{1}{2}(\rho_i)^\intercal M^{-1} \rho_i}}{\mu(\theta_{i+1}) e^{-\frac{1}{2}(\rho_{i+1})^\intercal M^{-1} \rho_{i+1}}} \frac{p_{\text{micro}}(\ell \mid \text{micro}(\theta_i, \rho_i, \mu, M, h, \delta))}{p_{\text{micro}}(\ell \mid \text{micro}(\theta_{i+1}, -\rho_{i+1}, \mu, M, h, \delta))} w_{i+1}$$

return $\mathcal{O}^{\text{ext}} = ((\theta_{a-L}, \rho_{a-L}), \ldots, (\theta_{a-2}, \rho_{a-2}), (\theta_{a-1}, \rho_{a-1}))$ and
$\mathcal{W}^{\text{ext}} = (w_{a-L}, \ldots, w_{a-2}, w_{a-1})$

Listing 5: *Generates an orbit of L macro steps and associated weights over the index range $(a-L)$:$(a-1)$.*

`micro`$(\theta, \rho, \mu, M, h, \delta)$

Inputs:	$(\theta, \rho) \in \mathbb{R}^{2d}$	initial position, momentum
	$h_0 > 0$	macro step size
	$\mu : \mathbb{R}^d \to (0, \infty)$	unnormalized target density
	$M \in \mathbb{R}^{d \times d}$	mass matrix
	$h > 0$	macro step size
	$\delta > 0$	maximum energy error
Return:	$\ell \in \mathbb{N}$	step-size reduction factor

$(\theta^{(0)}, \rho^{(0)}) = (\theta, \rho)$
for i from 0 to ∞
    $\ell = 2^i$
    $h_{\text{micro}} = h\, \ell^{-1}$
    $H_{\max} = H_{\min} = -\log \mu(\theta^{(0)}) + \frac{1}{2}(\rho^{(0)})^\intercal M^{-1} \rho^{(0)}$
    for j from 0 to $\ell - 1$
        $\rho^{(j+1/2)} = \rho^{(j)} + \frac{1}{2}\, h_{\text{micro}}\, \nabla \log \mu(\theta^{(j)})$
        $\theta^{(j+1)} = \theta^{(j)} + h_{\text{micro}}\, M^{-1} \rho^{(j+1/2)}$
        $\rho^{(j+1)} = \rho^{(j+1/2)} + \frac{1}{2}\, h_{\text{micro}}\, \nabla \log \mu(\theta^{(j+1)})$
        $H^{(j+1)} = -\log \mu(\theta^{(j+1)}) + \frac{1}{2}(\rho^{(j+1)})^\intercal M^{-1} \rho^{(j+1)}$
        $H_{\max} = \max(H^{(j+1)}, H_{\max})$
        $H_{\min} = \min(H^{(j+1)}, H_{\min})$
    if $H_{\max} - H_{\min} \le \delta$
        return ℓ

Listing 6: *Computes the smallest power-of-two integer ℓ such that ℓ leapfrog steps of micro step size $h_{\text{micro}} = h\ell^{-1}$ keep the energy error within the user-specified threshold δ, as defined in* (9).

Appendix B. Practical Implementation

This section outlines practical implementation strategies that reduce the computational and memory overhead of WALNUTS. The pseudocode in Listing 1, including dependencies, is intended to provide the reader with a clear conceptual understanding of the WALNUTS transition kernel. However, directly implementing this version can lead to substantial redundant computations and high memory usage. In particular, storing all positions and momenta in an orbit $\mathcal{O}$ may require $O(d \cdot 2^{m_{\max}})$ memory, where d is the dimension of the state space and $m_{\max}$ is the maximum number of doublings.

B.1 Error-controlled leapfrog integration steps

In the pseudocode for orbit expansion (Listings 4 and 5), the calls to `micro` and the leapfrog integration steps used to build the final orbit are implemented as separate function calls for both forward and backward directions. While this modular structure aids conceptual understanding and code readability, it introduces computational overhead, as the same leapfrog steps may be recomputed multiple times.

In practice, the contents of the `for` loops in `extend-orbit-forward` and `extend-orbit-backward` are consolidated into a single routine that performs all relevant operations together. This combined implementation reduces redundant computation by reusing results where possible. The forward case is described below; the backward case is analogous.

- Compute $\widetilde{\ell} = \texttt{micro}(\theta_{i-1}, \rho_{i-1}, \mu, M, h, \delta)$, and store $(\theta_i, \rho_i) = \Phi^{\widetilde{\ell}}_{h\widetilde{\ell}^{-1}}(\theta_{i-1}, \rho_{i-1})$, which is already computed as part of the `micro` procedure.
- Sample $\ell \sim p_{\text{micro}}(\cdot|\widetilde{\ell})$.
- If $\ell = \widetilde{\ell} = 0$: reuse the stored result (θ_i, ρ_i).
- If $\ell = \widetilde{\ell} > 0$: use the precomputed result $(\theta_i, \rho_i) = \Phi^{\widetilde{\ell}}_{h\widetilde{\ell}^{-1}}(\theta_{i-1}, \rho_{i-1})$. To compute the weight factor, we then run $\texttt{micro}(\theta_i, -\rho_i, \mu, M, h, \delta)$ using step sizes corresponding to micro step count 2^j for $j = 0, 1, \ldots, \log_2(\ell) - 1$. (There is no need to run the backward `micro` iteration for $j = \log_2(\ell)$ or higher due to reversibility of the integration; see Lemma 14.)
- Otherwise (i.e., if $\ell \neq \widetilde{\ell}$): proceed as in `extend-orbit-forward`. In this case, there are no obvious computational savings compared to the original algorithm, and the backward `micro` loop generally cannot be terminated early.

Note that there can be a substantial computational cost difference between the cases $\ell = \widetilde{\ell}$ and $\ell \neq \widetilde{\ell}$. This may partly explain why WALNUTS configured with $p_{\texttt{micro}}(\widetilde{\ell} \mid \widetilde{\ell}) = 1$ often appears to run faster in practice, even if it terminates orbit construction slightly earlier and more frequently.

B.2 Memory-efficient orbit construction

As also described in (Hoffman and Gelman, 2014), a more memory-efficient implementation, requiring only $O(d \cdot m_{\max})$ memory, is possible by checking the sub-U-turn condition and

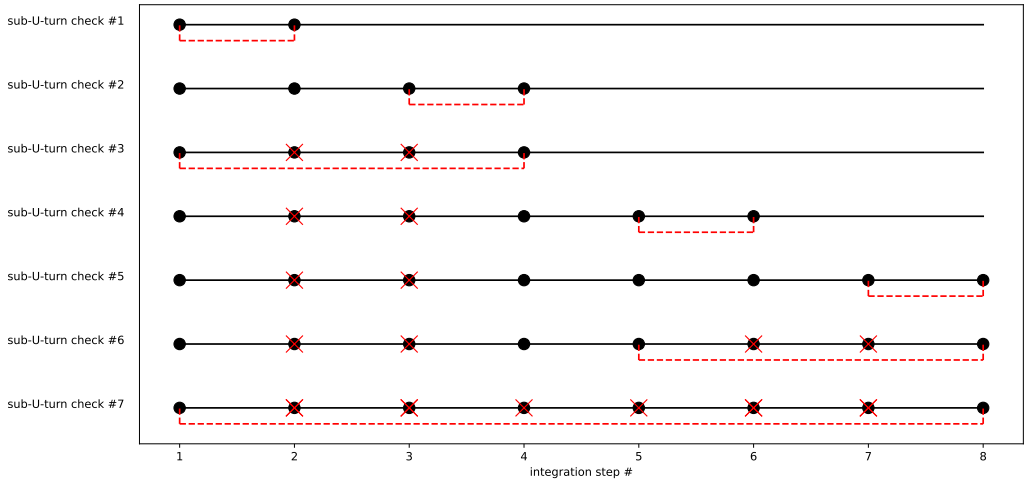


Figure 19: Sequence of sub-U-turn checks during forward orbit expansion with 8 integration steps. Black dots indicate computed integration steps; red Xs mark steps that have been discarded. Red underbraces denote the pairs of states involved in each U-turn check. A state can be safely deleted once it is covered by a completed U-turn check that spans beyond it on both sides.

performing sampling from categorical $\left(\mathcal{O}^{\text{ext}}, \mathcal{W}^{\text{ext}}\right)$ concurrently during the execution of `extend-orbit-forward` or `extend-orbit-backward`. Since this approach is thoroughly explained in (Hoffman and Gelman, 2014), we provide only a brief summary here.

The categorical sampling step is conceptually straightforward. Suppose we wish to sample a random index $I \in \{1, \ldots, N\}$ according to a sequence of (possibly unnormalized) weights $w_1, \ldots, w_N$. This can be done in an online manner using the following recursive procedure:

- Initialize $I = 1$ and $S_1 = w_1$.
- For $i = 2, \ldots, N$:
  - Set $S_i = S_{i-1} + w_i$,
  - Draw $U_i \sim \text{Unif}(0, 1)$,
  - With probability w_i / S_i, set $I = i$; otherwise, leave I unchanged.

At the end of this procedure, the variable I is distributed according to the categorical distribution with probabilities proportional to $w_1, \ldots, w_N$.

Sub-U-turn checks can also be performed concurrently with the `extend-orbit-forward` and `extend-orbit-backward` routines. In fact, the algorithm in Listing 3 is designed to operate sequentially on the states of the orbit as they are computed. Once a U-turn check

has been performed between two states, say i and j, the intermediate states $i+1, \ldots, j-1$ are no longer needed and can be discarded to reduce memory usage.

Figure 19 illustrates this concurrent procedure for `extend-orbit-forward` with 8 integration steps. New integration steps are computed in iterations 1, 2, 4, and 5 of the algorithm. At no point is the full sub-orbit stored in memory.

Concurrent U-turn checks can also lead to computational savings: if a U-turn is detected early (e.g., during check #3 in Figure 19), then the remaining integration steps can be skipped, as no proposal will be drawn from that region and the orbit expansion stops.

Similar considerations apply when the weight factor from the reversibility check becomes zero. Specifically, if the reversibility check yields zero weights on both sides of the current state (θ_0, ρ_0), the orbit-building process can be terminated early.

Appendix C. Exploring Tuning Strategies

This section focuses on tuning the macro step size h, which determines the temporal spacing between candidate states along the final orbit, and the energy error threshold δ, which controls local step size refinement within each macro step. WALNUTS is designed to be robust to user input, but performance can be significantly improved by selecting h and δ carefully in a warmup phase; that is, an initial portion of the Markov chain devoted to adapting algorithm parameters prior to collecting samples from the posterior. In this part, we explore approaches to tuning both parameters with the goal of balancing efficiency and robustness.

C.1 Energy error threshold δ

To appropriately set the energy error threshold δ in the micro step selection function

$$\text{micro}(\theta, \rho, \mu, M, h, \delta) := \min \left\{ \ell \in 2^{\mathbb{N}} \mid H_\ell^+ - H_\ell^- \leq \delta \right\}, \tag{36}$$

we aim to control the local energy error introduced by each integration step while ensuring robust global behavior of the sampler.

When choosing the energy error threshold $\delta > 0$, the primary objective is to prevent numerical divergences by bounding the energy error introduced in each macro integration step. An additional, equally important objective is to control the total change in energy along an orbit $\mathcal{O}$. Doing so helps ensure that the distribution of the integration time index i, which determines the next step of the chain, is broadly spread across the orbit.

To see this, consider the idealized setting in which the leapfrog integrator is replaced by the exact Hamiltonian flow, and assume for simplicity that the orbit length is fixed. In this case, there is no energy error, and the normalized integration time index i follows an exact symmetric triangular distribution centered at 0.5 (see Theorem 8). When the orbit length is tuned to match the local scale of the target, this integration time distribution enables effective exploration along that scale.

In practice, the leapfrog integrator induces an energy error, which can distort this ideal integration time distribution and lead to biased or inefficient proposals, particularly if the energy error varies substantially along the orbit. Controlling the energy error is

therefore essential: it allows the sampler to approximate the idealized behavior locally, so that movement occurs effectively along the relevant scale of the target.

While it is possible to combine WALNUTS's local step size adaptation with global orbit-level adaptation, as proposed in (Bou-Rabee et al., 2025), doing so may compromise the fine-grained adaptivity that WALNUTS is designed to achieve; namely, reducing the step size only in regions where it is truly necessary. Instead, we advocate a strategy that tunes the local energy error threshold δ so that the *global* energy change along the orbit satisfies

$$H_{\mathcal{O}}^{+} - H_{\mathcal{O}}^{-} < \Delta, \tag{37}$$

with high probability, where $H_{\mathcal{O}}^{+}$ and $H_{\mathcal{O}}^{-}$ denote the maximum and minimum Hamiltonians observed along the orbit, and $\Delta > 0$ is a global energy error threshold. As shown in (Bou-Rabee et al., 2025), the quantity $\exp(-(H_{\mathcal{O}}^{+} - H_{\mathcal{O}}^{-}))$ is strongly correlated with the probability $P(i \neq 0)$ of selecting a nontrivial proposal. Enforcing the bound (37) with a prescribed probability level p_a (e.g., $p_a = 0.95$) therefore promotes stable and reliable sampling.

WALNUTS employs a *variable step-size leapfrog integrator*, which adaptively adjusts the step size within each macro integration interval to satisfy a local energy error threshold δ. Specifically, the step size is recursively halved until the estimated energy error in a single step satisfies

$$|H_{\ell}^{+} - H_{\ell}^{-}| \leq \delta.$$

If an orbit consists of n macro steps, each satisfying the above bound, then in the worst case, where all local energy errors accumulate constructively, then the total energy error satisfies

$$H_{\mathcal{O}}^{+} - H_{\mathcal{O}}^{-} \leq n\delta.$$

While this is a conservative upper bound, it highlights the need for orbit-level control. In practice, the signs of energy errors often fluctuate, especially in well-conditioned regions of the target, leading to partial cancellation. Nonetheless, the *envelope* of the energy error (i.e., the difference between the largest and smallest Hamiltonians along the orbit) may grow with orbit length. To capture this possibility, we model the accumulated energy variation as

$$H_{\mathcal{O}}^{+} - H_{\mathcal{O}}^{-} \approx \mathcal{K}\delta,$$

where $\mathcal{K}$ is a *local-to-orbit inflation factor*, a random variable whose distribution depends on the number of integration steps, the macro step size h, and the local geometry of the target distribution. In the ideal case of uncorrelated stepwise errors, $\mathcal{K}$ remains $\mathcal{O}(1)$ even for long orbits, due to frequent sign changes.

To ensure that (37) holds with probability at least p_a, we propose adapting δ during warmup by recording the observed inflation factor

$$\mathcal{K} = \frac{H_{\mathcal{O}}^{+} - H_{\mathcal{O}}^{-}}{\delta} \tag{38}$$

for each constructed orbit. These values are collected across warmup iterations, and δ is then updated according to

$$\delta = \frac{\Delta}{q_{p_a}^{\mathcal{K}}}, \tag{39}$$

where $q_{p_a}^{\mathcal{K}}$ denotes the empirical p_a-quantile of the recorded inflation factors. Since each orbit contributes a single value of $\mathcal{K}$, this adaptation procedure is both memory-efficient and computationally inexpensive.

This strategy enables automatic and interpretable tuning of δ, aligning local step size control with global energy error constraints, and thereby supporting the stability and robustness of the WALNUTS sampler.

C.2 Macro step size h (given δ)

Recall that the function $\text{micro}(\theta, \rho, \mu, M, h, \delta)$ returns the number of step size halvings needed to reduce the energy error below the threshold δ, as defined in (36). In particular, if no micro step size halving is needed, then $\text{micro}(\theta, \rho, \mu, M, h, \delta) = 1$.

To tune the macro step size h, the primary goal is efficiency. If h is too small, WALNUTS will take unnecessarily many leapfrog steps, even in regions where larger steps would be sufficient. On the other hand, if h is too large, the `micro` routine will frequently trigger step size halving to satisfy the energy error threshold δ, which wastes computation. Moreover, overly large values of h can cause the integration paths to become too coarse to detect U-turns or other important geometric features, potentially leading to poor proposal quality.

To navigate this trade-off, we propose choosing h such that the probability of needing no step size halving is equal to a target value $\Gamma \in (0, 1)$:

$$P\left(\texttt{micro}(\theta, \rho, \mu, M, h_0, \delta) = 1\right) = \Gamma \tag{40}$$

This criterion means that, in a typical region of phase space, the initial macro step size h is accepted without refinement in a fraction Γ of cases. A value such as $\Gamma = 0.8$ typically works well in practice: most steps proceed without modification, while `micro` still adjusts the step size in regions where smaller steps are needed.

In short, this objective helps ensure that h is large enough to be efficient across most of the target distribution, without compromising robustness or geometric resolution in more challenging areas.

References

Christophe Andrieu, Anthony Lee, and Sam Livingstone. A general perspective on the Metropolis-hastings kernel. *arXiv*, 2012.14881, 2020.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv*, 1701.02434, 2017.

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79(30):2–4, 2015.

Miguel Biron-Lattes, Nikola Surjanovic, Saifuddin Syed, Trevor Campbell, and Alexandre Bouchard-Côté. autoMALA: Locally adaptive Metropolis-adjusted Langevin algorithm. In *27th International Conference on Artificial Intelligence and Statistics*, volume PMLR 238, 2024.

Nawaf Bou-Rabee and Andreas Eberle. Couplings for Andersen dynamics in high dimension. *Ann. Inst. H. Poincaré Probab. Statist*, 58(2):916–944, 2022.

Nawaf Bou-Rabee and Stefan Oberdörster. Mixing of the no-U-turn sampler and the geometry of Gaussian concentration. *arXiv*, 2410.06978, 2024a.

Nawaf Bou-Rabee and Stefan Oberdörster. Mixing of Metropolis-adjusted Markov chains via couplings: The high acceptance regime. *Electronic Journal of Probability*, 29:1–27, 2024b.

Nawaf Bou-Rabee and Jesús María Sanz-Serna. Randomized Hamiltonian Monte Carlo. *Ann. Appl. Probab.*, 27(4):2159–2194, 2017.

Nawaf Bou-Rabee and Jesús María Sanz-Serna. Geometric integrators and the Hamiltonian Monte Carlo method. *Acta Numerica*, 27:113–206, 2018.

Nawaf Bou-Rabee and Eric Vanden-Eijnden. A patch that imparts unconditional stability to explicit integrators for Langevin-like equations. *J Comput Phys*, 231:2565–2580, 2012.

Nawaf Bou-Rabee, Bob Carpenter, and Milo Marsden. Gist: Gibbs self-tuning for locally adaptive Hamiltonian Monte Carlo. *arXiv*, 2404.15253, 2024.

Nawaf Bou-Rabee, Bob Carpenter, Tore Selland Kleppe, and Milo Marsden. Incorporating local step-size adaptivity into the no-U-turn sampler using Gibbs self-tuning. *The Journal of Chemical Physics*, 163(8):084119, 2025.

Mari Paz Calvo and Jesús María Sanz-Serna. The development of variable-step symplectic integrators, with application to the two-body problem. *SIAM Journal on Scientific Computing*, 14(4):936–952, 1993.

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 20:1–37, 2016.

Yifan Chen. New affine invariant ensemble samplers and their dimensional scaling. *arXiv*, 2505.02987, 2025.

Augustin Chevallier, Sam Power, and Matthew Sutton. Towards practical PDMP sampling: Metropolis adjustments, locally adaptive step-sizes, and NUTS-based time lengths. *arXiv*, 2503.11479, 2025.

George Deligiannidis, Daniel Paulin, Alexandre Bouchard-Côté, and Arnaud Doucet. Randomized Hamiltonian Monte Carlo as scaling limit of the bouncy particle sampler and dimension-free convergence rates. *The Annals of Applied Probability*, 31(6):2612 – 2662, 2021.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte-Carlo. *Physics Letters B*, 195:216–222, 1987.

Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.

Qiang Fu and Andre Wibisono. Hamiltonian descent algorithms for optimization: Accelerated rates via randomized integration time. *arXiv*, 2505.12553, 2025.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J R Statist Soc B*, 73:123–214, 2011.

Nathan Glatt-Holtz, Justin Krometis, and Cecilia Mondaini. On the accept–reject mechanism for Metropolis–Hastings algorithms. *The Annals of Applied Probability*, 33(6B), 2023. doi: 10.1214/23-aap1948.

Nathan E. Glatt-Holtz, Andrew J. Holbrook, Justin A. Krometis, Cecilia F. Mondaini, and Ami Sheth. Sacred and profane: from the involutive theory of MCMC to helpful Hamiltonian hacks, 2024.

Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, 2010.

Peter J Green and Antonietta Mira. Delayed rejection in reversible jump Metropolis–Hastings. *Biometrika*, 88(4):1035–1053, 2001.

Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. DRAM: efficient adaptive MCMC. *Statistics and Computing*, 16:339–354, 2006.

E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer, 2010.

Max Hird and Samuel Livingstone. Quantifying the effectiveness of linear preconditioning in Markov chain Monte Carlo. *arXiv*, 2312.04898, 2023.

M. D. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

A. M. Horowitz. A generalized guided Monte-Carlo algorithm. *Phys Lett B*, 268:247–252, 1991.

Tore Selland Kleppe. Adaptive step size selection for Hessian-based manifold Langevin samplers. *Scandinavian Journal of Statistics*, 43(3):788–805, 2016.

Tore Selland Kleppe. Dynamically rescaled Hamiltonian Monte Carlo for Bayesian hierarchical models. *Journal of Computational and Graphical Statistics*, 28(3):493–507, 2019.

Tore Selland Kleppe. Connecting the dots: Numerical randomized Hamiltonian Monte Carlo with state-dependent event rates. *Journal of Computational and Graphical Statistics*, 31 (4):1238–1253, 2022.

B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.

Benedict Leimkuhler, René Lohmann, and Peter Whalley. A Langevin sampling algorithm inspired by the Adam optimizer. *arXiv*, 2504.18911, 2025.

Tiange Liu, Nikola Surjanovic, Miguel Biron-Lattes, Alexandre Bouchard-Côté, and Trevor Campbell. AutoStep: Locally adaptive involutive MCMC. *arXiv*, 2410.18929, 2024.

Charles C Margossian and Lawrence K Saul. The shrinkage-delinkage trade-off: An analysis of factorized Gaussian approximations for variational inference. In *Uncertainty in Artificial Intelligence*, pages 1358–1367. PMLR, 2023.

Charles C Margossian and Lawrence K Saul. Variational inference in location-scale families: Exact recovery of the mean and correlation matrix. *arXiv*, 2410.11067, 2024.

Antonietta Mira. On Metropolis-Hastings algorithms with delayed rejection. *Metron*, 59 (3-4):231–241, 2001.

Chirag Modi, Alex Barnett, and Bob Carpenter. Delayed rejection Hamiltonian Monte Carlo for sampling multiscale distributions. *Bayesian Analysis*, 1(1):1–28, 2023.

R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2011.

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705 – 767, 2003.

Kirill Neklyudov, Max Welling, Evgenii Egorov, and Dmitry Vetrov. Involutive MCMC: a unifying framework. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7273–7282, 2020.

Chris Sherlock, Szymon Urbas, and Matthew Ludkin. The apogee to apogee path sampler. *Journal of Computational and Graphical Statistics*, 32(4):1436–1446, 2023.

James H. Stock and Mark W. Watson. Why has U.S. inflation become harder to forecast? *Journal of Money, Credit and Banking*, 39(s1):3–33, 2007.

Daniel Martin Stofer. *Some geometric and numerical methods for perturbed integrable systems*. PhD thesis, ETH Zurich, 1988.

Jimmy Huy Tran and Tore Selland Kleppe. Tuning diagonal scale matrices for HMC. *Statistics and Computing*, 34(6):196, 2024.

Gilad Turok, Chirag Modi, and Bob Carpenter. Sampling from multiscale densities with delayed rejection generalized Hamiltonian Monte Carlo. In *AISTATS*, 2025.

Roman Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018.

Peter A Whalley, Daniel Paulin, and Benedict Leimkuhler. Randomized time Riemannian manifold Hamiltonian Monte Carlo. *Statistics and Computing*, 34(1):48, 2024.

Kai Xu, Tor Erlend Fjelde, Charles Sutton, and Hong Ge. Couplings for multinomial Hamiltonian Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 3646–3654. PMLR, 2021.

Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for non-mixing Bayesian computations: The curse and blessing of multimodal posteriors. *Journal of Machine Learning Research*, 23(79):1–45, 2022.