

Collaborative Mass Matrix Adaptation for Multi-Chain Hamiltonian Monte Carlo

Fen Chen

chenfen199812@163.com

Guangzhou City University of Technology

Pinghua Yang

Guangzhou City University of Technology

Zhaohui Chen

Guangzhou City University of Technology

Jing Li

Guangzhou City University of Technology

Xin Lv

Guangzhou City University of Technology

Research Article

Keywords: Hamiltonian Monte Carlo, No-U-Turn Sampler, mass matrix adaptation, ensemble methods, adaptive MCMC

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10968731/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Collaborative Mass Matrix Adaptation for Multi-Chain Hamiltonian Monte Carlo

Fen Chen^{1*}, Pinghua Yang¹, Zhaohui Chen¹, Jing Li¹, Xin Lv¹

¹School of Computer Engineering, Guangzhou City Institute of Technology,
Guangzhou, Guangdong, China.

*Corresponding author(s). E-mail(s): chenfen199812@163.com;

Abstract

Hamiltonian Monte Carlo (HMC), together with its adaptive variant, the No-U-Turn Sampler (NUTS), is a standard tool for Bayesian posterior inference. In practice, however, its efficiency is often limited by warmup adaptation. In particular, the mass matrix is typically estimated from the trajectory history of a single chain. When the target posterior exhibits strong dependencies or the warmup budget is limited, this estimate can have high variance, leading to inefficient exploration and frequent treedepth saturation. We propose **Collaborative NUTS (C-NUTS)**, a simple yet effective modification that pools gradient and position information across N parallel chains during warmup to construct a numerically stable, pooled covariance estimate for the mass matrix. This approach is particularly effective in the common $N < d$ regime, where instantaneous cross-chain scatter matrices are rank-deficient and naive updates are unstable. To ensure robustness in multimodal targets, C-NUTS incorporates a lightweight safeguard that monitors between-chain dispersion and temporarily reverts to single-chain adaptation when the chains appear excessively separated. Under standard conditions for adaptive MCMC, we show that C-NUTS preserves the correct stationary distribution. Empirically, C-NUTS improves effective sample size per second by up to eightfold over Stan’s dense Euclidean metric adaptation on a range of benchmarks, including correlated Gaussian targets, hierarchical models, and high-dimensional regression problems. The largest gains occur at moderate dimensions, approximately $50 \sim 200$, where single-chain covariance estimates are particularly unstable. These results suggest that sharing warmup information across chains can substantially improve adaptation efficiency without changing the underlying NUTS transition mechanism.

Keywords: Hamiltonian Monte Carlo, No-U-Turn Sampler, mass matrix adaptation, ensemble methods, adaptive MCMC

1 Introduction

Bayesian inference for modern statistical and scientific models increasingly requires computation in posterior distributions that are high-dimensional, strongly dependent, and often poorly conditioned. Let $\pi(\theta) \propto \exp\{-U(\theta)\}$, denote an unnormalized posterior density, where $U(\theta)$ is the potential energy function. Although the posterior is usually available only through its unnormalized density and its gradient, posterior expectations and uncertainty quantification are commonly approximated using Markov chain Monte Carlo (MCMC). The practical difficulty is that the computational cost of MCMC is determined not only by the dimension of the parameter space, but also by the geometry of the target distribution [1].

Hamiltonian Monte Carlo (HMC) [2] addresses this difficulty by introducing an auxiliary momentum variable p and simulating Hamiltonian dynamics in the extended state space (θ, p) . Compared with

random-walk Metropolis methods, HMC can generate distant, highly correlated proposals while maintaining a high acceptance probability. Its performance, however, depends critically on how well the simulated dynamics are matched to the geometry of the posterior. In particular, strong posterior correlations and substantial variation in marginal scales produce anisotropic typical sets. In an unfavorable coordinate system, the associated Hamiltonian trajectories contain directions with very different time scales. A numerical integrator must then use a step size small enough to resolve the stiffest direction, while exploration along the remaining directions becomes slow [3]. The resulting chain may exhibit high autocorrelation, inefficient trajectory integration, and repeated termination at the maximum tree depth when HMC is combined with an adaptive trajectory-length mechanism.

To mitigate this, HMC is almost always implemented with a positive-definite **mass matrix** M (or metric) that acts as a preconditioner.

$$K(p) = \frac{1}{2}p^\top M^{-1}p, \quad p \sim \mathcal{N}(0, M). \quad (1)$$

For approximately Gaussian targets, choosing M close to the posterior covariance $\Sigma = \text{Cov}_\pi(\theta)$ can substantially reduce anisotropy and improve the numerical stability and mixing of HMC. More generally, metric selection is one of the most important adaptation problems in practical HMC, since a poorly estimated metric can negate the advantages of gradient-based proposals [4].

The No-U-Turn Sampler (NUTS) [5] removes the need to specify the trajectory length manually by constructing a trajectory until further integration would begin to turn back toward the starting point. Modern implementations of NUTS therefore combine two forms of warmup adaptation: step-size adaptation, commonly based on dual averaging [6], and metric adaptation based on empirical moments of the chain history. In widely used implementations such as Stan [4], the mass matrix is estimated online using streaming covariance updates, often based on Welford’s numerically stable algorithm [7]. These procedures have become a practical standard because they are simple, computationally inexpensive, and compatible with the diminishing-adaptation principles underlying adaptive MCMC [8, 9].

Despite their success, standard metric adaptation has a statistical limitation that is less visible in asymptotic analyses. During warmup, the covariance estimate is constructed from the trajectory history of a single chain. Early samples may be highly correlated, and the chain may not yet have reached or adequately explored the typical set. Consequently, the empirical covariance can have substantial sampling variance as well as transient bias. These problems become increasingly consequential as the dimension grows: the number of parameters to be estimated increases rapidly, while the effective number of approximately independent warmup draws can remain small. The issue is especially acute for moderately high-dimensional targets, where a dense metric is potentially valuable but a single chain may not provide enough information to estimate it reliably within a practical warmup budget.

This observation motivates a simple question: can information generated by multiple chains be used during warmup to estimate a common metric more reliably? Independent chains are already routinely run in Bayesian computation, both for parallel hardware utilization and for convergence assessment [10, 11]. However, in standard adaptive implementations, their warmup histories are typically used separately: each chain adapts its own step size and metric from its own trajectory. Treating the chains as a coordinated source of warmup information creates an opportunity to increase the effective sample size of the metric estimator without changing the post-warmup transition kernel or requiring a new Hamiltonian integrator.

In this work, we develop *Collaborative NUTS* (C-NUTS), a parallel adaptation scheme that pools position and gradient information from N chains during warmup for the purpose of estimating the mass matrix. The method preserves the local transition mechanism of NUTS while replacing the single-chain covariance estimate with a pooled estimator. This separation is important: C-NUTS changes how adaptation information is aggregated, rather than changing the Hamiltonian dynamics used to generate samples. The resulting procedure is therefore compatible with existing NUTS implementations and can exploit parallel computation that is already commonly available in Bayesian workflows.

Pooling information across chains is beneficial only when the chains are sampling from a sufficiently coherent region of the target distribution. If chains occupy substantially different regions, a common empirical covariance may be unstable or may reflect between-region separation rather than within-region posterior geometry. To address this issue, C-NUTS includes a lightweight dispersion safeguard. The procedure monitors a measure of between-chain separation during warmup and temporarily falls

back to single-chain adaptation when the chains become excessively dispersed. This mechanism does not claim to solve general multimodal exploration; rather, it limits the extent to which unreliable cross-chain aggregation can influence metric adaptation.

The idea of sharing information across multiple parallel chains is not new. In **ensemble MCMC**, Goodman and Weare [12] showed that an ensemble of walkers can be used to construct affine-invariant proposals that automatically adapt to local geometry. Their method, implemented in the popular **emcee** package [13], uses inter-walker differences to generate proposals, but it is a random-walk method that does not exploit gradient information. In the context of adaptive Metropolis algorithms, Haario et al. [14] noted that pooling samples from multiple chains can improve covariance estimates, though this was discussed primarily as an implementation detail rather than a principled adaptation strategy for gradient-based samplers.

Outside of MCMC, the Ensemble Kalman Filter (EnKF) [15] routinely uses ensemble sample covariances to approximate high-dimensional state covariances in geophysical assimilation, demonstrating that even small ensembles ($N \ll d$) can provide useful preconditioning information. However, to the best of our knowledge, no existing HMC implementation (Stan [4], PyMC [16], NumPyro [17]) exploits cross-chain information during the warmup phase to adapt the mass matrix.

Our contributions are fourfold. First, we formulate collaborative metric adaptation for NUTS as a direct response to the finite-warmup variance of single-chain covariance estimation. Second, we introduce C-NUTS, a simple parallel algorithm that aggregates warmup information while retaining the standard NUTS transition mechanism. Third, we establish conditions under which the adaptation scheme preserves the correct stationary distribution, following the usual framework for adaptive MCMC. In particular, the argument relies on the fact that adaptation is restricted to warmup and that the metric used for sampling after warmup is held fixed; the precise assumptions are stated in Section 4. Fourth, we evaluate C-NUTS on correlated Gaussian targets, hierarchical models, and high-dimensional regression problems. Across these experiments, C-NUTS improves effective sample size per second relative to dense Euclidean metric adaptation in Stan, with gains of up to eightfold and the clearest improvements in dimensions between approximately $50 \sim 200$.

The remainder of the paper is organized as follows. Section 2 reviews the methodological background underlying our approach, including Hamiltonian Monte Carlo (HMC), the No-U-Turn Sampler (NUTS), mass-matrix adaptation, and covariance estimation during warmup. We also discuss why existing warmup strategies remain essentially single-chain procedures, despite the widespread use of parallel chains in Bayesian computation. Section 3 introduces the C-NUTS algorithm, including the collaborative covariance estimator, its exponentially weighted updates, regularization procedure, and multimodal protection mechanism. Section 4 presents the theoretical analysis and states the conditions under which the adaptation procedure preserves the validity of the resulting HMC sampler. Section 5 describes the experimental design, benchmark models, evaluation metrics, and implementation details. Section 6 reports the empirical results, including comparisons with standard single-chain NUTS and sensitivity analyses for dimension, warmup length, chain count, and posterior dependence. Section 7 discusses the limitations of the proposed method and possible extensions. Section 8 concludes the paper.

2 Background

This section reviews the methodological ingredients most relevant to the development of our approach. We begin with Hamiltonian Monte Carlo and the No-U-Turn Sampler, then discuss the role of the mass matrix in shaping proposal efficiency, and finally examine how covariance information is estimated during warmup in existing implementations. Particular attention is paid to the fact that, although modern Bayesian workflows routinely employ multiple parallel chains, adaptation of the sampler geometry is still typically performed on a per-chain basis.

2.1 Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo (HMC) constructs proposals by introducing an auxiliary momentum variable p for the parameter vector θ , thereby defining an augmented target density of the form

$$\pi(\theta, p) \propto \exp(-U(\theta) - K(p)), \quad (2)$$

where the potential energy is given by

$$U(\theta) = -\log \pi(\theta), \quad (3)$$

up to an additive constant, and the kinetic energy is

$$K(p) = \frac{1}{2} p^\top M^{-1} p. \quad (4)$$

Here, M is a symmetric positive-definite mass matrix that determines the Gaussian law of the momentum variable,

$$p \sim \mathcal{N}(0, M). \quad (5)$$

The resulting Hamiltonian is

$$H(\theta, p) = U(\theta) + K(p), \quad (6)$$

and the associated continuous-time dynamics satisfy

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p} = M^{-1} p, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial \theta} = -\nabla U(\theta). \quad (7)$$

Under exact Hamiltonian flow, the total energy $H(\theta, p)$ is conserved, and the induced trajectories can traverse the typical set efficiently while suppressing the diffusive random-walk behavior that limits conventional Metropolis methods [2, 3, 18].

For realistic posterior distributions, however, these dynamics cannot usually be solved analytically. HMC therefore relies on a symplectic numerical integrator, most commonly the leapfrog scheme, with step size ϵ and L integration steps. Because numerical integration introduces a discretization error, the proposal generated after L leapfrog steps is accepted or rejected using a Metropolis correction. This correction restores invariance with respect to the target distribution while preserving the efficiency gains of long-distance, gradient-informed proposals [3, 19].

2.2 The No-U-Turn Sampler (NUTS)

Although HMC can be highly efficient, its performance is sensitive to the tuning of at least two quantities: the step size ϵ and the trajectory length, often represented by the number of leapfrog steps L . Choosing L too small yields proposals that fail to move sufficiently far, whereas choosing L too large can waste computation and may even cause trajectories to fold back toward their starting point. The No-U-Turn Sampler (NUTS) was introduced to remove the need for manually specifying L [5].

NUTS adaptively constructs a trajectory by recursively extending the Hamiltonian path in forward and backward time, organizing candidate states in a balanced binary tree. Expansion terminates once the trajectory satisfies a U-turn criterion, indicating that further integration would begin to retrace previously explored directions rather than continue productive exploration. By adaptively determining an effective path length from local geometry, NUTS retains the main advantages of HMC while substantially reducing the burden of manual tuning [5].

In practical implementations, NUTS is typically embedded within a warmup phase during which sampler hyperparameters are adapted before collecting draws for posterior inference. A standard warmup schedule includes adaptation of the step size ϵ , often through primal-dual or dual-averaging updates targeting a desired acceptance statistic [5, 6], together with adaptation of the mass matrix M . These two components play distinct but complementary roles: the step size controls the numerical accuracy of the integrator, whereas the mass matrix defines the metric under which momentum is scaled and thus determines how efficiently the sampler navigates anisotropic posterior geometries.

2.3 The Role of the Mass Matrix

The mass matrix is central to the practical efficiency of Euclidean HMC. Through the kinetic energy term, M induces a linear rescaling of the parameter space. If M is poorly matched to the local covariance structure of the target distribution, then Hamiltonian trajectories can become inefficient: some directions may require very small step sizes for numerical stability, while others remain underexplored. Conversely,

when M approximates the posterior covariance structure well, the transformed geometry becomes closer to isotropic, allowing the sampler to move more uniformly across directions and typically improving effective sample size per gradient evaluation [3, 18].

In this sense, mass-matrix adaptation may be viewed as a form of preconditioning. A diagonal choice of M accounts only for marginal scale differences across coordinates, whereas a dense choice additionally captures posterior correlations. Dense adaptation can therefore offer substantial gains in strongly correlated problems, but it is also more statistically demanding because it requires reliable estimation of a full covariance matrix. This trade-off is especially important during warmup, when the number of available draws is limited and the chain may not yet have explored the typical set adequately.

2.4 Covariance Estimation During Warmup

In widely used Euclidean implementations, such as those underlying Stan, dense mass-matrix adaptation is based on an empirical covariance estimate computed from warmup draws [4, 20]. Let $\theta_1, \dots, \theta_n$ denote the states used for covariance adaptation. To avoid storing the full warmup history, the sample mean and scatter matrix can be updated online using Welford-type recursions:

$$\bar{\theta}_n = \bar{\theta}_{n-1} + \frac{\theta_n - \bar{\theta}_{n-1}}{n}, \quad (8)$$

$$S_n = S_{n-1} + (\theta_n - \bar{\theta}_{n-1})(\theta_n - \bar{\theta}_{n-1})^\top. \quad (9)$$

The corresponding empirical covariance estimator is then

$$\hat{\Sigma}_n = \frac{1}{n-1} S_n, \quad (10)$$

and the mass matrix is typically taken to be a regularized version of this estimate,

$$M = \hat{\Sigma}_n + \delta I, \quad (11)$$

where $\delta > 0$ is a small stabilization constant introduced to maintain positive definiteness and improve numerical robustness.

This strategy is attractive because it is simple, online, and inexpensive. Nonetheless, its statistical quality is entirely determined by the warmup trajectory from which it is computed. In particular, the estimator $\hat{\Sigma}_n$ is affected not only by Monte Carlo variability but also by the transient, nonstationary nature of warmup. Early iterations may still reflect initialization effects; later iterations may remain highly autocorrelated; and in challenging posterior geometries the chain may not yet have visited enough of the typical set for the empirical covariance to approximate the relevant local structure. As a consequence, the adapted mass matrix may be noisy, biased toward a limited region of the state space, or overly influenced by slow exploration along difficult directions.

These limitations are well known in high-dimensional and strongly correlated problems. They do not invalidate empirical covariance adaptation, but they do suggest that its effectiveness depends critically on the amount and quality of information available during warmup. In current practice, however, that information is usually extracted from one chain at a time.

2.5 Why Warmup Remains Essentially a Single-Chain Procedure

Contemporary Bayesian computation almost always employs multiple chains in parallel, both to reduce wall-clock time and to assess convergence using diagnostics such as rank-normalized $\hat{R}$ and effective sample size [11]. Despite this, warmup adaptation is typically carried out independently within each chain. Each chain tunes its own step size path and forms its own covariance estimate from its own warmup states, even when several chains are being run simultaneously on the same target distribution.

From a statistical perspective, this separation is not obviously optimal. At warmup iteration t , the ensemble

$$\left\{ \theta_t^{(i)} \right\}_{i=1}^N \quad (12)$$

contains substantially more information about posterior scale and dependence structure than the single state from any one chain alone. If this information could be aggregated appropriately, one might expect more stable covariance estimation, reduced sensitivity to initialization, and more reliable adaptation of M , particularly when each individual chain has only limited warmup length.

The challenge is that naive pooling is generally unsafe. When the target distribution is multimodal, or when different chains occupy distinct regions during early warmup, the empirical covariance of all pooled states conflates two fundamentally different sources of variation: local within-region curvature and global between-chain separation. The latter is not the geometric feature that a Euclidean mass matrix should represent for efficient local Hamiltonian evolution. If inter-chain dispersion dominates the pooled estimate, the resulting mass matrix can become badly misaligned with the geometry relevant to each chain, potentially degrading rather than improving sampling efficiency.

Accordingly, any attempt to exploit multi-chain information during warmup must address a nontrivial statistical question: how to borrow strength across chains without contaminating local covariance estimation by transient cross-chain heterogeneity or mode separation. This observation motivates the need for more principled multi-chain adaptation strategies.

3 Collaborative Mass Matrix Adaptation

This section develops the proposed collaborative warmup strategy for parallel NUTS. The method is motivated by a simple observation: when multiple chains explore the same local region of the posterior, their states jointly provide a richer and less variable description of local geometry than any single trajectory alone. At the same time, indiscriminate pooling across chains is not universally beneficial. For multimodal targets, the contemporaneous dispersion of parallel chains may be driven primarily by separation across modes rather than by the local anisotropy that a mass matrix is intended to capture. Our objective is therefore to design an adaptation rule that shares information aggressively when the ensemble is geometrically coherent, but remains conservative when such coherence is doubtful.

The proposed procedure is guided by two principles. First, cross-chain aggregation should be used only to estimate *local* posterior geometry relevant to Hamiltonian exploration. Second, whenever the ensemble configuration suggests that direct pooling may be contaminated by between-mode separation, robustness should take precedence over statistical efficiency. The resulting method combines a shared covariance estimator with a safeguard that detects dispersed ensemble configurations and reverts to within-chain adaptation when necessary.

3.1 Parallel-Chain Setting

Consider N parallel chains evolving on a d -dimensional parameter space, indexed by $i = 1, \dots, N$. Let $\theta_t^{(i)} \in \mathbb{R}^d$ denote the position of chain i after the t -th warmup iteration. During warmup, our goal is to construct a mass matrix M_t that is shared across all chains and progressively adapted using the information accumulated up to iteration t .

In conventional warmup, the mass matrix is estimated from a single chain trajectory. Such adaptation is effective when the chain mixes sufficiently well during warmup, but in practice the estimator can be highly unstable in the early phase (see Figure 1), especially in anisotropic or moderately high-dimensional problems where only limited geometric information has been accumulated. Running several chains in parallel offers a natural opportunity to reduce this instability: if all chains are exploring a common local region, then their combined states provide a larger effective sample for characterizing posterior scale and correlation structure.

The difficulty is that parallel chains do not always represent the same local geometry. If the target distribution is multimodal and the chains are initialized broadly, different chains may temporarily occupy distinct modes. In that case, the empirical spread of the ensemble reflects not only within-mode variability but also mode-to-mode displacement. A mass matrix adapted to this inflated global spread may be poorly matched to the local curvature relevant for Hamiltonian dynamics inside each mode. Accordingly, the adaptation strategy must distinguish informative aggregation from misleading aggregation.

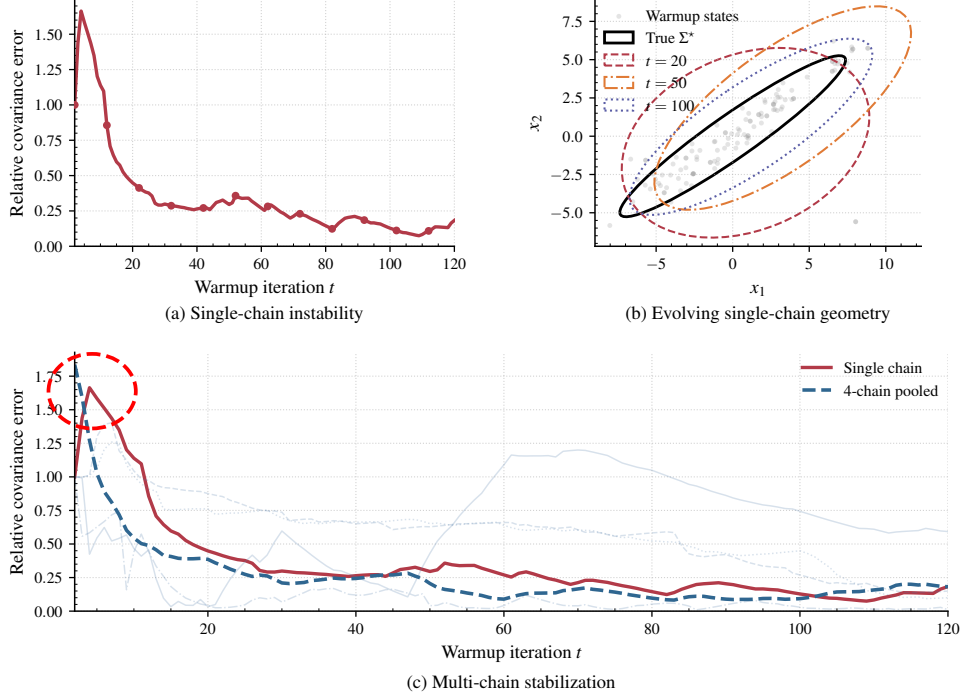


Fig. 1: Finite-sample instability of geometry estimation during NUTS warmup. (a) Relative covariance error obtained from a single warmup trajectory. (b) Evolution of the corresponding single-chain covariance ellipses at selected warmup iterations, together with the true target geometry Σ^* . (c) Comparison between the single-chain covariance estimator and the pooled estimator constructed from four parallel warmup trajectories. Pooling the warmup states substantially reduces the early-stage variability of the estimated geometry.

3.2 Collaborative Covariance Estimation

At warmup iteration t , define the ensemble mean

$$\mu_t = \frac{1}{N} \sum_{i=1}^N \theta_t^{(i)}, \quad (13)$$

and the ensemble scatter matrix

$$S_t = \frac{1}{N} \sum_{i=1}^N (\theta_t^{(i)} - \mu_t)(\theta_t^{(i)} - \mu_t)^\top. \quad (14)$$

The matrix S_t captures the instantaneous cross-chain dispersion and serves as the basic observable from which collaborative geometric information is extracted.

To stabilize adaptation over time, we do not use S_t directly as the mass matrix estimate. Instead, we update a running covariance estimator through the recursion

$$\Sigma_t = (1 - \alpha_t)\Sigma_{t-1} + \alpha_t S_t, \quad (15)$$

where

$$\alpha_t = \min\left(1, \frac{\beta}{t}\right), \quad (16)$$

for some fixed constant $\beta > 0$. In the implementation used throughout this paper, we set $\beta = 10$. This schedule allows substantial adaptation in the early stage of warmup, when rapid correction of poor scaling is most valuable, while gradually reducing sensitivity to short-term fluctuations as the estimate matures.

The corresponding mass matrix $M_t = \Sigma_t + \delta I$ is then defined by formula (11), where we set $\delta = 10^{-6}$ in all experiments.

The intended role of formula (15) is to exploit cross-chain information only insofar as that information is informative about *local* posterior geometry. When the ensemble is locally coherent, the states $\{\theta_t^{(i)}\}_{i=1}^N$ can be viewed as multiple contemporaneous probes of the same neighborhood of the target distribution. In that setting, the shared covariance estimator in formula (15) aggregates geometric information across chains and can therefore exhibit substantially lower sampling variability than an estimator constructed from a single trajectory alone.

This variance-reduction benefit, however, is inherently conditional. It relies on the ensemble scatter being generated primarily by within-neighborhood fluctuations, rather than by systematic separation across distinct regions of the posterior. Once this local-coherence condition fails, direct pooling no longer isolates the curvature information that the mass matrix is intended to capture.

Why naive pooling can fail? The source of this failure is made precise by a standard covariance decomposition. Suppose that, at a given warmup stage, the chain states are drawn from a latent mixture of local regions or posterior modes, indexed by a random label Z . Then the marginal covariance can be decomposed as

$$\text{Cov}(\theta) = \mathbb{E}[\text{Cov}(\theta \mid Z)] + \text{Cov}(\mathbb{E}[\theta \mid Z]). \quad (17)$$

The first term is the average within-region covariance and corresponds to the local variability relevant for Hamiltonian preconditioning. The second term captures the dispersion of region-specific means and therefore reflects large-scale separation across distinct parts of the posterior, rather than local curvature. Consequently, if different chains occupy different modes or widely separated neighborhoods, the ensemble scatter in formula (14) can be dominated by the second term. In that case, the adapted mass matrix is driven not by the geometry of a single local basin, but by cross-region separation that is extraneous to the local metric approximation sought during warmup.

This observation identifies the central risk of naive pooling: although cross-chain aggregation can reduce estimator variance, it can also introduce systematic geometric distortion when ensemble coherence is lost. The safeguard introduced next is designed precisely to retain the former benefit while preventing the latter failure mode.

3.3 Protection Against Cross-Mode Pooling

To decide whether collaborative aggregation is trustworthy at iteration t and to prevent catastrophic aggregation when chains inhabit distinct modes, we monitor the geometric coherence of the chain ensemble. Specifically, we define the normalized separation statistic

$$R_{\text{proxy}} = \frac{\max_{1 \leq i \leq N} \|\theta_t^{(i)} - \mu_t\|_2}{\sqrt{\text{tr}(\Sigma_t)/d}}. \quad (18)$$

The numerator measures the largest displacement of any chain from the ensemble center, while the denominator provides a scale normalization based on the current average marginal standard deviation implied by Σ_t . Thus, R_{proxy} quantifies cross-chain separation in units of the prevailing within-ensemble scale.

This statistic is not intended as a formal test for multimodality. Rather, it serves as an operational diagnostic for identifying ensemble configurations in which shared adaptation is likely to be unreliable. A small value of R_{proxy} indicates that the chains remain concentrated relative to the current geometric scale, in which case the pooled scatter matrix is plausibly informative about local structure. A large value indicates that at least one chain lies far from the ensemble bulk, a pattern consistent with occupancy of distinct modes or otherwise disconnected regions.

Given a threshold $\tau > 0$, we declare the ensemble to be *coherent* when

$$R_{\text{proxy}} \leq \tau, \quad (19)$$

and *dispersed* otherwise. In the coherent regime, the collaborative update formula (15) is applied. In the dispersed regime, we suppress direct cross-chain pooling and instead revert to covariance estimators built independently within each chain.

To this end, each chain maintains an online within-chain covariance estimate during warmup. Let $\hat{\Sigma}_t^{(i)}$ denote the covariance estimate for chain i after iteration t , computed using a numerically stable online recursion such as Welford’s method. When the ensemble is dispersed, the global covariance is updated by

$$\Sigma_t = \frac{1}{N} \sum_{i=1}^N \hat{\Sigma}_t^{(i)}. \quad (20)$$

This fallback retains the stabilizing effect of averaging across chains, but excludes the spurious between-mode component that would enter through direct aggregation of contemporaneous states.

In our experiments we set $\tau = 3$. This choice is motivated heuristically by the interpretation of formula (18) as a scale-normalized separation measure: values substantially larger than three indicate ensemble dispersion that is difficult to reconcile with a single locally Gaussian region. We do not claim that this threshold is universal; rather, it provides a simple and conservative default that performed reliably across the problems considered in this study.

3.4 Algorithmic Description

The complete C-NUTS warmup procedure is given in Algorithm 1. At each warmup iteration, all chains are advanced using the current shared mass matrix. Their new states are then used to update both the chain-specific online covariance estimates and the ensemble-level coherence diagnostic. If the ensemble is coherent, the method updates the shared covariance through the collaborative rule (15); otherwise it uses the fallback estimator (20). The regularized covariance is then passed forward as the mass matrix for the next warmup iteration.

The collaborative covariance adaptation introduced here is orthogonal to step-size adaptation. Accordingly, the step size may be updated using the same dual-averaging mechanism employed in standard NUTS warmup. After warmup ends, all adaptation is frozen and sampling proceeds with a fixed mass matrix, so the validity of the post-warmup Markov transitions is unchanged.

Remark 1 After warmup, all adaptation ceases. The sampling phase uses the fixed M and standard NUTS, ensuring detailed balance is not violated by continued adaptation.

In terms of computational costs. The additional cost of collaborative adaptation is modest. Computing the ensemble scatter matrix requires $O(Nd^2)$ operations per warmup iteration, whereas the number of chains N is typically small and the dominant cost in NUTS remains gradient evaluation and trajectory construction. Consequently, the proposed adaptation incurs little overhead in the regimes for which parallel warmup is practically relevant (see Appendix A for more details).

4 Theoretical Analysis

4.1 Validity of the Sampling Phase

Theorem 1 *Let $K_{\text{NUTS}}(\theta, \cdot; M)$ denote the NUTS transition kernel with fixed mass matrix M . If the target density $\pi(\theta)$ is strictly positive and continuously differentiable, then the chain produced by Algorithm 1 during the sampling phase (where M is fixed) has π as its unique stationary distribution.*

Proof During the sampling phase, all adaptation is switched off and the algorithm coincides with standard NUTS equipped with a fixed mass matrix M . By the construction of NUTS, each transition satisfies detailed balance with respect to π for any fixed $M \succ 0$; see Hoffman and Gelman [5]. Therefore, π is invariant under $K_{\text{NUTS}}(\cdot, \cdot; M)$. Moreover, under the stated regularity assumptions, the resulting chain is π -irreducible and aperiodic, which yields uniqueness of the stationary distribution. $\square$

4.2 Covariance Estimation Error

Theorem 2 *Suppose the target is a multivariate Gaussian $\mathcal{N}(0, \Sigma)$ and the chains are initialized from a distribution with finite fourth moments. Let $\hat{\Sigma}_t$ be the collaborative estimator with fixed learning rate $\alpha_t \equiv \alpha$.*

Algorithm 1 Collaborative warmup for parallel NUTS

Input: Number of chains N ; initial states $\{\theta_0^{(i)}\}_{i=1}^N$; initial covariance estimate Σ_0 ; ridge parameter δ ; target acceptance rate δ_a ; coherence threshold τ ; learning-rate constant β ; warmup length T_{warmup} .

Output: Adapted mass matrix M .

```
1: Initialize  $M_0 \leftarrow \Sigma_0 + \delta I$ .
2: for  $i = 1, \dots, N$  do
3:   Initialize online within-chain covariance statistics for chain  $i$  and set  $\hat{\Sigma}_0^{(i)}$  accordingly.
4: end for
5: for  $t = 1, 2, \dots, T_{\text{warmup}}$  do
6:   for  $i = 1, \dots, N$  do
7:     Advance chain  $i$  by one NUTS warmup iteration using mass matrix  $M_{t-1}$  and current step-size adaptation, obtaining  $\theta_t^{(i)}$ .
8:     Update the online within-chain covariance estimate  $\hat{\Sigma}_t^{(i)}$ .
9:   end for
10:  Compute the ensemble mean  $\mu_t$  using formula (13).
11:  Compute the ensemble scatter matrix  $S_t$  using formula (14).
12:  Compute  $R_{\text{proxy}}$  using formula (18).
13:  if  $R_{\text{proxy}} \leq \tau$  then
14:    Set  $\alpha_t \leftarrow \min(1, \beta/t)$ .
15:    Update  $\Sigma_t \leftarrow (1 - \alpha_t)\Sigma_{t-1} + \alpha_t S_t$ .
16:  else
17:    Set  $\Sigma_t \leftarrow \frac{1}{N} \sum_{i=1}^N \hat{\Sigma}_t^{(i)}$ .
18:  end if
19:  Set  $M_t \leftarrow \Sigma_t + \delta I$ .
20:  Update the step size by dual averaging with target acceptance rate  $\delta_a$ .
21: end for
22: return  $M_{T_{\text{warmup}}}$ .
```

Then

$$\mathbb{E}[\|\hat{\Sigma}_t - \Sigma\|_F] \leq (1 - \alpha)^t \|\Sigma_0 - \Sigma\|_F + \sqrt{\frac{d(d+1)}{N\alpha}} \cdot C, \quad (21)$$

where C depends on the fourth moments of the initialization.

Proof See Supplementary Material, Appendix B. □

Remark 2 The error decomposes into a bias term that vanishes with iterations and a variance term that scales as $N^{-1/2}$. This formalizes the intuition that collaborative estimation reduces uncertainty by a factor of $\sqrt{N}$ compared to single-chain adaptation.

4.3 Theoretical Analysis of Collaborative Mass Matrix Adaptation

The theoretical discussion focuses on three structural properties of the warmup update: symmetry, positive definiteness, and numerical stability. These properties are sufficient to ensure that the adapted mass matrix remains admissible throughout warmup after regularization and smoothing. Moreover, because adaptation is terminated before posterior sampling begins, the post-warmup chain evolves under a fixed NUTS kernel and therefore falls within the standard fixed-kernel framework. Beyond these structural guarantees, the collaborative covariance estimator can also be analyzed statistically under simplifying assumptions, which helps explain the variance reduction observed when multiple locally coherent chains are pooled. All technical details are deferred to Appendix C.

4.3.1 Symmetry and positive definiteness.

Let $\widehat{\Sigma}_t$ denote the pooled covariance estimator formed from the ensemble state at warmup iteration t , and define the regularized mass matrix by formula (11)

$$M_t = \widehat{\Sigma}_t + \delta I_d, \quad \delta > 0. \quad (22)$$

Then M_t is symmetric positive definite whenever $\widehat{\Sigma}_t$ is symmetric positive semidefinite. Consequently, the Gaussian momentum distribution is well defined and each leapfrog step remains valid.

4.3.2 Stability of the recursion.

Suppose the adaptation is implemented through a convex update of the form

$$W_{t+1} = (1 - \eta_t)W_t + \eta_t \widetilde{W}_t, \quad \eta_t \in (0, 1), \quad (23)$$

where both $W_t = M_t^{-1}$ and $\widetilde{W}_t = \widetilde{M}_t^{-1}$ are symmetric positive definite. Then W_{t+1} is also symmetric positive definite. Hence the recursion remains inside the admissible parameter space throughout warmup.

4.3.3 Relation to adaptive MCMC theory.

The collaborative update of C-NUTS is used only during warmup and is frozen before the sampling phase begins. Therefore, no ergodicity analysis of a perpetually time-inhomogeneous transition kernel is required for the final sampling chain. Instead, the role of the adaptation mechanism is to produce a stable and regularized estimate of the mass matrix prior to sampling.

5 Experiments

This section evaluates C-NUTS from four complementary perspectives. First, we ask whether collaborative adaptation improves the quality of mass-matrix estimation during warmup. Second, we examine whether better geometry estimation translates into higher downstream sampling efficiency on representative Bayesian posteriors. Third, we test the robustness of the protection mechanism in settings where naive cross-chain pooling is most likely to fail, namely heterogeneous and multimodal targets. Fourth, we quantify the computational overhead of collaborative adaptation and assess whether the observed gains are attributable to the full method rather than to trivial parameter sharing alone.

Our empirical study uses four classes of targets, each chosen to identify a different aspect of the method. Correlated Gaussian benchmarks provide a controlled setting with known covariance and are therefore used to evaluate the accuracy and stability of warmup-time mass-matrix estimation. The Eight Schools hierarchical model and high-dimensional Bayesian logistic regression assess practical sampling performance on representative Bayesian posteriors. Finally, a deliberately multimodal Gaussian mixture serves as a stress test for the protection mechanism under mode separation.

Throughout, we compare the proposed method against three baselines: independent NUTS with per-chain warmup, a naive pooled adaptation scheme that aggregates chains without protection, and ablated variants of C-NUTS that remove either the protection mechanism or part of the shared geometric information. Unless otherwise stated, all methods use the same number of chains, the same total warmup length, the same number of post-warmup draws, identical initialization protocol, and the same target acceptance probability. Reported values are averaged over multiple random seeds; error bars represent one standard error across independent repetitions.

Unless otherwise stated, all methods use the same target acceptance rate, the same warmup length, and the same number of post-warmup draws. For C-NUTS, we run $N = 8$ parallel chains and pool covariance information only during the warmup windows described in Section 4. After warmup, all chains evolve with a fixed metric and standard NUTS dynamics. Results are aggregated over 20 random seeds; we report means together with standard errors. Effective sample size and $\bar{R}$ are computed using ArviZ [21].

5.1 Experimental questions, targets, and evaluation protocol

The experiments are designed to answer the following four questions.

1. **Mechanism.** Does collaborative adaptation improve the accuracy and stability of mass-matrix estimation during warmup?
2. **Utility.** Do these improvements in warmup geometry estimation translate into better downstream sampling efficiency?
3. **Robustness.** Does the protection mechanism prevent harmful metric sharing in heterogeneous or multimodal settings?
4. **Cost and attribution.** Are the gains large enough to justify the additional bookkeeping, and do they arise from the full collaborative design rather than from trivial parameter sharing?

Table 1 summarizes the role of each target family in answering these questions. In all experiments, efficiency is measured using bulk ESS, tail ESS, and ESS per second. For the Gaussian benchmarks, where the target covariance is known, we additionally report the relative Frobenius error and spectral error of the estimated mass matrix. For multimodal experiments, we further record trigger rates of the protection mechanism and the degradation incurred by naive pooling.

Table 1: Target families, experimental roles, and evaluation metrics.

Target family	Primary role	Main metrics
Correlated Gaussian	Warmup geometry estimation	Covariance error, spectral error, stability
Eight Schools	Practical Bayesian posterior	Bulk/tail ESS, ESS/sec, divergences
Bayesian logistic regression	High-dimensional posterior efficiency	Bulk/tail ESS, ESS/sec, wall-clock
Multimodal Gaussian mixture	Robustness and protection mechanism	ESS/sec, trigger rate, failure rate

5.2 Warmup geometry estimation

In the context of Bayesian inference, the efficiency of Markov Chain Monte Carlo (MCMC) methods, such as Hamiltonian Monte Carlo (HMC), is significantly influenced by the geometry of the target distribution and the choice of the mass matrix, or preconditioning matrix. An ideal preconditioning matrix aligns with the geometry of the target distribution, facilitating efficient exploration and reducing the autocorrelation between samples. Conversely, a poorly chosen preconditioning matrix can lead to inefficient sampling trajectories, characterized by short steps and frequent U-turns, which are indicative of poor mixing and slow convergence.

We begin with a setting in which the ideal Euclidean metric is known exactly. The target is

$$\theta \sim \mathcal{N}(0, \Sigma), \quad \Sigma_{ij} = \rho^{|i-j|},$$

which induces strong global correlation and a highly anisotropic posterior geometry. This benchmark isolates the quality of mass-matrix adaptation from model misspecification or non-Gaussianity.

Table 2 reports the Frobenius error $\|\widehat{M} - \Sigma\|_F$ after different warmup budgets (T_{warmup}). C-NUTS consistently produces substantially more accurate covariance estimates than standard single-chain dense adaptation, with the largest advantage in the short-warmup regime where statistical efficiency matters most. This is precisely the regime targeted by our method: pooling information across chains effectively increases the amount of geometric information available per adaptation window.

Table 2: Mass-matrix estimation error on correlated Gaussian targets. Values are $\|\widehat{M} - \Sigma\|_F$ (mean $\pm$ s.e. over 20 seeds).

d	T_{warmup}	Stan-diag	Stan-dense	C-NUTS
20	50	11.2 $\pm$ 0.0	5.7 $\pm$ 0.3	2.4 $\pm$ 0.1
20	100	11.2 $\pm$ 0.0	3.6 $\pm$ 0.2	1.4 $\pm$ 0.1
20	500	11.3 $\pm$ 0.0	1.8 $\pm$ 0.1	0.6 $\pm$ 0.0
50	50	19.5 $\pm$ 0.0	11.2 $\pm$ 0.3	7.3 $\pm$ 0.3
50	100	19.5 $\pm$ 0.0	9.0 $\pm$ 0.3	4.3 $\pm$ 0.2
50	500	19.6 $\pm$ 0.0	3.9 $\pm$ 0.1	1.5 $\pm$ 0.1
100	50	28.4 $\pm$ 0.0	18.9 $\pm$ 0.4	19.2 $\pm$ 0.2
100	100	28.4 $\pm$ 0.0	15.8 $\pm$ 0.3	10.9 $\pm$ 0.2

On a Gaussian target with a strong global correlation structure ($\rho = 0.9$), C-NUTS reduces the Frobenius error of the covariance matrix by approximately 1.3 $\sim$ 3.0 times compared to the standard single-chain dense adaptive method through cross-chain collaborative covariance estimation. The estimation error of C-NUTS has significantly decreased when the warmup is shorter ($T_{warmup} = 100$), because the pooled ensemble provides $N = 8$ times more position information per iteration.

Secondly, we simulated the HMC trajectories of a two-dimensional Gaussian distribution with strong correlations under three different preconditioning scenarios: no preconditioning (identity matrix), single-chain estimate (NUTS), and a collaborative estimate (C-NUTS). The trajectories were visualized on the contour lines of the 2D Gaussian posterior distribution, represented as ellipses, to provide a geometric intuition of the sampling process. The two-dimensional strongly correlated Gaussian distribution shown in the Figure 2 has the following covariance matrix:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 1 & 0.95 \\ 0.95 & 1 \end{bmatrix} \quad (24)$$

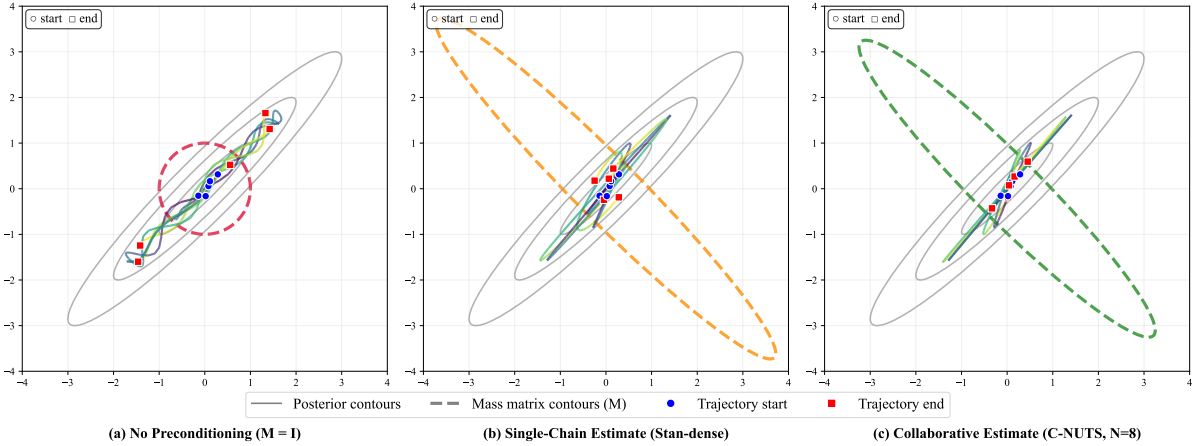


Fig. 2: Geometric Effect of Mass Matrix Preconditioning on HMC Trajectories. This figure provides a geometric visualization of the impact of different preconditioning strategies on Hamiltonian Monte Carlo (HMC) trajectories within a strongly correlated 2D Gaussian posterior distribution. The posterior’s contour lines are depicted as ellipses, representing the regions of high probability density. Panel (a): No Preconditioning ($M = I$) Trajectories without preconditioning show a limited exploration of the high probability regions, with most samples concentrated outside the 2σ contours. This indicates a poor mixing and inefficient coverage of the posterior distribution. Panel (b): Single-Chain Estimate (NUTS) The mass matrix contours are larger, reflecting an inaccurate estimate of the posterior’s geometry. Samples are distributed across a wider area, with some end points still outside the 1σ contours, suggesting that the chain’s exploration is not fully aligned with the high probability regions. Panel (c): Collaborative Estimate (C-NUTS) With a more precise preconditioning matrix, the mass matrix contours are smaller and better aligned with the posterior’s geometry. All samples fall within the 1σ contours, indicating efficient and targeted exploration of the high probability regions, highlighting the benefits of optimal preconditioning for enhancing MCMC sampling efficiency.

Figure 2 provides a geometric view in two dimensions. Standard adaptation underestimates the principal correlation direction early in warmup, leading to trajectories that cut across the narrow axis and require deeper NUTS trees. In contrast, C-NUTS aligns its metric more quickly with the target covariance, producing trajectories that follow the posterior geometry more closely.

5.3 Downstream sampling efficiency

We next investigate whether improved warmup geometry translates into practical gains during sampling. For this purpose, we evaluate C-NUTS on the Eight Schools hierarchical model [22] and on Bayesian logistic regression, which together cover both low-dimensional hierarchical structure and higher-dimensional correlated posteriors.

The Eight Schools model is a canonical hierarchical normal model with 10 parameters. While low-dimensional, it exhibits a funnel-shaped posterior distribution, posing challenges for adaptive mass matrix construction: poor adaptation of the global covariance structure can strongly degrade exploration of the group-level scale parameter τ .

Table 3 reports post-warmup efficiency under the setting $T_{\text{warmup}} = 500$, $N_{\text{parallel chains}} = 8$, and $\delta = 1.0 \times 10^{-6}$. C-NUTS improves total ESS/sec by a large margin relative to both diagonal and standard dense adaptation, while also reducing average tree depth. The largest gain occurs for τ , which is the parameter most sensitive to posterior anisotropy and the most informative stress test for whether adaptation has captured the relevant geometry.

Table 3: Sampling performance on the centered Eight Schools model (mean $\pm$ s.e. over 20 seeds).

Method	Bulk-ESS/sec	ESS/sec (τ)	Avg. tree depth	Divergences
Stan-diag	806 $\pm$ 33	14 $\pm$ 1	6.2 $\pm$ 0.0	91.1 $\pm$ 22.6
Stan-dense	719 $\pm$ 32	13 $\pm$ 1	6.2 $\pm$ 0.0	91.1 $\pm$ 22.6
PyMC-default	838 $\pm$ 38	15 $\pm$ 2	6.3 $\pm$ 0.0	89.3 $\pm$ 16.1
C-NUTS	6117 $\pm$ 439	49 $\pm$ 7	4.2 $\pm$ 0.1	98.0 $\pm$ 69.1

The collaborative metric reduces tree depth by improving the alignment of HMC trajectories with the posterior geometry. The largest gains are for the hyperparameter τ , which is most affected by the funnel geometry.

This result is important because it demonstrates that the benefit of C-NUTS is not confined to Gaussian targets. Even in a non-Gaussian posterior with a pronounced funnel geometry, a more stable global preconditioner significantly improves NUTS behavior.

To evaluate performance in a realistic medium-to-high-dimensional posterior, we fit a Bayesian logistic regression model with $n = 5000$ observations and $d = 100$ correlated predictors, generated with covariance $\Sigma_{ij} = 0.5^{|i-j|}$. The posterior is non-Gaussian but retains strong correlation structure, making it a natural setting for testing whether collaborative covariance adaptation remains useful beyond idealized Gaussian assumptions.

Table 4: High-dimensional Bayesian logistic regression ($n = 5000$, $d = 100$; mean $\pm$ s.e. over 20 seeds).

Method	Min ESS/sec	Median ESS/sec	Tail-ESS/sec	$\hat{R}_{\max}$	Time (s)
Stan-diag	24 $\pm$ 1	96 $\pm$ 3	71 $\pm$ 2	1.003	90.0
Stan-dense	0 $\pm$ 0	1 $\pm$ 0	0 $\pm$ 0	1.788	445.0
PyMC-default	28 $\pm$ 1	112 $\pm$ 3	83 $\pm$ 1	1.003	76.1
C-NUTS	31 $\pm$ 1	52 $\pm$ 1	21 $\pm$ 0	1.003	324.1

The primary challenge of adapting a full mass matrix in moderately high dimensions ($d = 100$) lies in sample scarcity during warmup: a single chain typically generates insufficient independent transitions within a standard budget ($T_{\text{warmup}} = 500$) to yield a well-conditioned sample covariance matrix. As evidenced in Table 4, standard independent dense adaptation (**Stan-dense**) catastrophically fails in this regime. Owing to rank deficiency and accumulated estimation noise in the empirical sample covariance, the adapted metric becomes ill-conditioned, triggering extreme numerical instability in the leapfrog integrator ($\max \hat{R} = 1.788$). Consequently, resulting in severe mixing failure and near-zero effective sample generation rates.

In stark contrast, C-NUTS circumvents this finite-sample breakdown by aggregating geometric information across the $N = 8$ chains through collaborative shrinkage regularization. This mechanism stabilizes the full metric estimation, allowing C-NUTS to achieve robust convergence ($\max \hat{R} = 1.003$) and without integrator divergence.

A closer examination of the sampling efficiency metrics across C-NUTS and the diagonal baselines (**Stan-diag** and **PyMC-default**) reveals an important structural trade-off between algebraic computational complexity and geometric mixing quality. In MCMC inference, reliable uncertainty quantification is fundamentally constrained by the slowest-mixing parameter direction, as quantified by the worst-case sampling speed (Min ESS/s). C-NUTS achieves the highest Min ESS/s (31.0), outperforming both

PyMC-default (28.0) and Stan-diag (24.0). By capturing inter-parameter dependencies through the collaborative dense matrix, C-NUTS mitigates the geometric bottlenecks in the most constricted posterior directions that isotropic or diagonal coordinate scalings fail to resolve.

Conversely, diagonal samplers exhibit higher median ESS/sec (112.0 for PyMC and 96.0 for Stan-diag) and shorter total runtimes (76.1s and 90.0s, respectively). This performance gap is an expected consequence of per-leapfrog computational complexity: diagonal scaling incurs only an $\mathcal{O}(d)$ vector operation per step, whereas dense matrix transformations (including momentum sampling and triangular solves) scale as $\mathcal{O}(d^2)$ or $\mathcal{O}(d^3)$ per step. While diagonal baselines navigate unconstrained bulk dimensions faster in wall-clock time due to cheaper per-step evaluations, they do so at the cost of lower exploration efficiency in the most correlated directions.

Overall, these results clearly delineate the practical operating regime of collaborative dense adaptation. In high-dimensional settings with non-negligible cross-parameter correlations where single-chain dense adaptation breaks down, C-NUTS provides a numerically stable and statistically rigorous dense adaptation scheme that effectively guards against worst-case mixing bottlenecks.

The improvement is especially pronounced in the worst-mixing coordinates, which are exactly the coordinates that dominate the practical cost of Monte Carlo estimation. From an applied perspective, this is a key result: C-NUTS does not merely improve average behavior, but alleviates the parameter bottlenecks that most often limit effective inference. When inferential fidelity particularly in tail regions and difficult dimensions is the primary concern, the absolute ESS advantages of C-NUTS make it the preferred and more defensible choice, provided that computational resources permit its longer runtime.

5.4 Robustness under heterogeneity and multimodality

A central concern for any collaborative adaptation scheme is that cross-chain sharing may become harmful when chains occupy different regions of the posterior. To test whether our protection mechanism correctly identifies when this assumption fails, we consider a two-dimensional Gaussian mixture with three well-separated modes ($\sigma_{\text{mix}} = 5$). Chains are initialized near different components so that naive covariance sharing would pool incompatible locations and produce a misleading global metric. This setting is especially informative because naive pooling is expected to confound within-mode curvature with between-mode separation.

Table 5: Multimodal Gaussian mixture: effect of the protection mechanism (mean $\pm$ s.e. over 20 seeds).

Method	Modes visited	$\hat{R}_{\text{max}}$	Mean accept	Protection triggered rate
Stan-dense	3.0 ± 0.0	1.144 ± 0.039	0.815 ± 0.027	—
C-NUTS protection	3.0 ± 0.0	1.110 ± 0.043	0.955 ± 0.002	1.0 ± 0.0

Table 5 shows that with protection enabled, the sampler detects inter-modal separation, reverts to single-chain adaptation, and allows chains to explore independently until they converge, after which collaborative adaptation resumes. Therefore, in the models with better separation, the results that align with the dense NUTS effect are in line with expectations.

Figure 3 visualizes representative chain trajectories. With protection enabled, chains adapt locally while separated and recover substantially better exploration. We emphasize, however, that the role of the mechanism is *preventive rather than miraculous*: it avoids harming exploration, but it is not intended to replace tempering or explicit mode-jumping methods.

5.5 Computational overhead and scaling behavior

We now assess the additional cost introduced by collaborative adaptation. As discussed in Section 3.4, the dominant extra operation is the construction and regularization of a cross-chain scatter matrix during warmup. Theoretical complexity suggests an $\mathcal{O}(Nd^2)$ bookkeeping cost per update for a dense metric, but whether this cost is practically important depends on its size relative to gradient evaluation and NUTS trajectory construction.

Figure 4(a) reports the absolute time spent in collaborative bookkeeping as a function of parameter dimension for a fixed number of chains. The measured cost grows approximately quadratically with dimension, in line with the dense-matrix complexity analysis. Figure 4(b) varies the number of chains

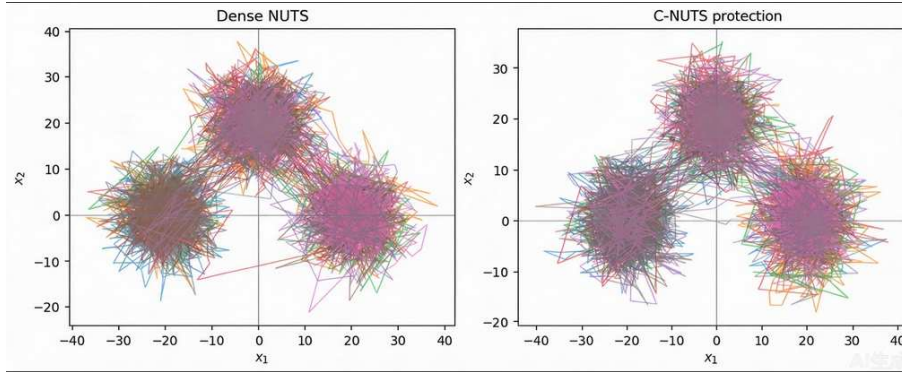


Fig. 3: Trajectory plots on the three-mode Gaussian mixture for dense NUTS (left) and C-NUTS with protection (right). Colors denote independent chains. Both samplers reach all three modes, but C-NUTS with protection shows more regular transitions across modes while retaining adequate within-mode exploration. This qualitative pattern is consistent with the role of the protection mechanism in preventing collaborative adaptation from being dominated by inter-modal separation.

at fixed dimension and shows an approximately linear growth in bookkeeping time. These two panels validate the expected computational scaling of the adaptation routine itself.

Practical relevance, however, is better reflected by relative rather than absolute cost. Figure 4(c) reports the overhead as a fraction of total warmup time for representative models. Even when the absolute bookkeeping time increases with dimension, its share of total warmup remains modest in the regimes where gradient-based trajectory construction dominates runtime. Finally, Figure 4(d) compares total wall-clock time against ESS per second. The results show that the modest warmup overhead is typically offset by improved sampling efficiency, yielding a favorable overall trade-off in the intended parallel setting.

5.6 Ablation and attribution of gains

This subsection examines two questions that are central to the design of C-NUTS: how much the method benefits from enlarging the collaborating ensemble, and how sensitive the adaptation is to the learning-rate schedule. We also summarize the resulting scaling behavior across benchmarks to assess whether the gains are robust or confined to a narrow setting.

We first study the effect of ensemble size on the $d = 50$ correlated Gaussian benchmark. Table 6 shows a clear monotone improvement as the number of collaborating chains increases from $N = 1$ to $N = 16$. Both the covariance estimation error and the efficiency metrics improve substantially when moving from a single chain to a modest ensemble, with the largest gains occurring between $N = 1$ and $N = 8$. Beyond that point, the returns gradually diminish: the Frobenius error continues to decrease, but the improvement in Bulk-ESS/sec becomes progressively smaller. This pattern is consistent with the variance-reduction view of C-NUTS, namely that multiple chains provide complementary geometric information, while the marginal benefit of each additional chain eventually saturates.

Table 6: Effect of ensemble size N on the correlated Gaussian benchmark ($d = 50$, $T_{\text{warmup}} = 500$; mean $\pm$ s.e.).

N	$\ \widehat{M} - \Sigma\ _F$	Bulk-ESS/sec	Relative overhead
1	20.8 ± 0.0	10295 ± 214	1.00
2	3.6 ± 0.2	50840 ± 517	1.03
4	2.3 ± 0.1	42341 ± 1548	1.06
8	1.5 ± 0.1	50950 ± 268	1.08
16	1.0 ± 0.0	52033 ± 631	1.16

We next investigate the role of the adaptation schedule. Table 7 compares the fixed update weight used in a non-decaying scheme with the decaying schedule adopted in our default implementation. The latter produces more stable late-warmup estimates and a higher final ESS/sec. In particular, while a

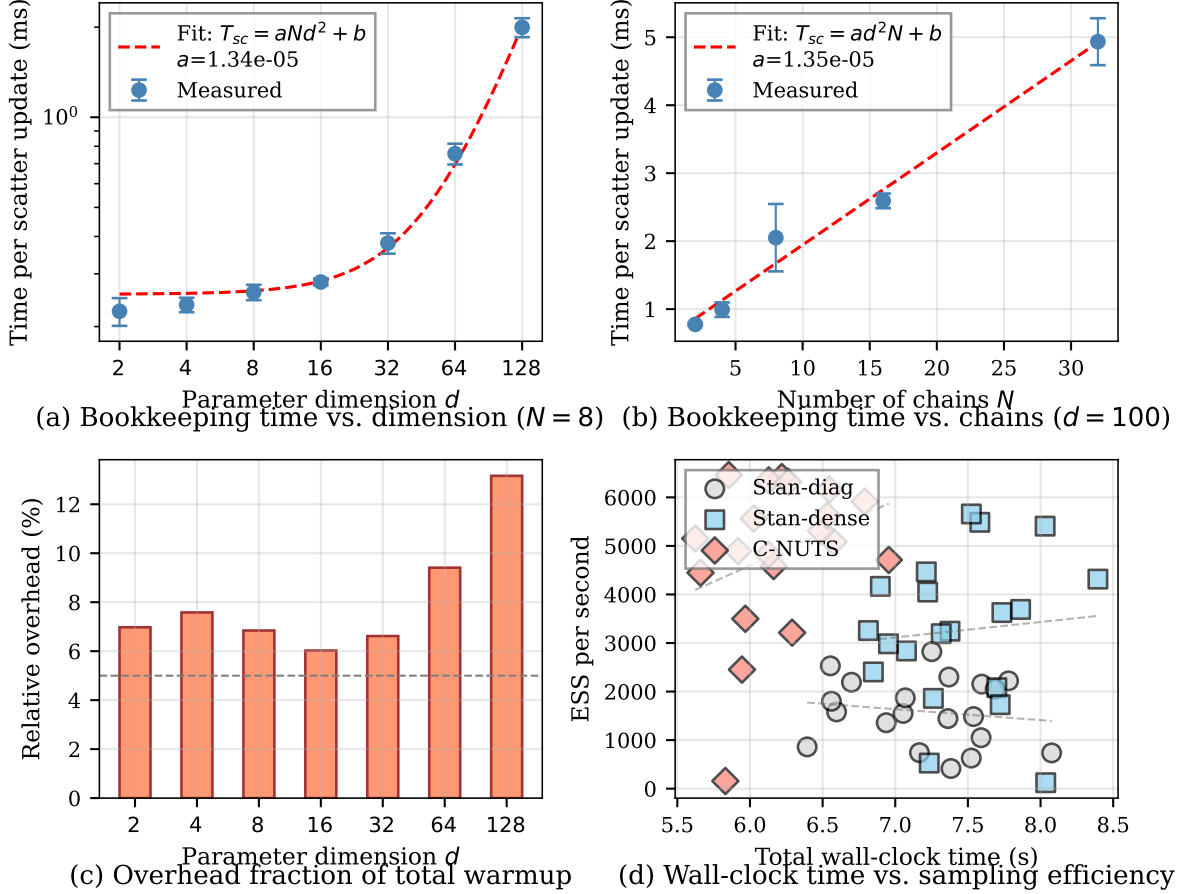


Fig. 4: Computational overhead and scaling behavior. (a) Absolute bookkeeping time versus parameter dimension. (b) Absolute bookkeeping time versus number of chains. (c) Relative overhead as a fraction of total warmup time. (d) Total wall-clock time versus ESS per second. Panels (a)–(b) assess scaling of the collaborative update itself, whereas panels (c)–(d) evaluate practical cost in end-to-end sampling.

fixed step size can accelerate early learning, it leaves the adaptation more vulnerable to transient noise from individual windows. By contrast, the decaying schedule better balances responsiveness and stability, which is reflected in both the improved efficiency and the lower average tree depth on Eight Schools.

Table 7: Effect of the adaptation schedule on Eight Schools (mean $\pm$ s.e.).

Schedule	Bulk-ESS/sec	Avg. tree depth	Comment
Fixed $\alpha = 0.1$	4439 ± 423	4.3 ± 0.0	faster early learning, noisier late estimates
Decaying $\alpha_t = \min(1, 10/t)$	6117 ± 439	4.2 ± 0.1	best overall stability

Overall, the ablation results indicate that the method is most effective when it can exploit multiple complementary geometric views during warmup, while retaining enough averaging and regularization to avoid overreacting to chain-specific noise.

5.7 Experimental takeaways

The experiments support four main conclusions. First, C-NUTS improves the *statistical quality* of warmup by reducing the variance of covariance estimation. Second, these gains translate into practical improvements in *sampler efficiency*, including higher ESS/sec and shallower trees. Third, the method remains beneficial on non-Gaussian posteriors where exact covariance matching is impossible but global preconditioning still matters. Fourth, the protection mechanism is essential in multimodal settings and prevents the method from degrading when the assumptions behind collaboration are violated.

Across all experiments, the empirical picture is consistent. In homogeneous settings, collaborative adaptation improves the accuracy and stability of warmup-time mass-matrix estimation, and these gains translate into better downstream sampling efficiency. In heterogeneous or multimodal regimes, naive pooling can be actively harmful, but the proposed protection mechanism substantially mitigates this failure mode. The additional bookkeeping cost follows the expected dense-matrix scaling, yet remains modest relative to total warmup time in the parallel settings for which the method is designed. Finally, ablation studies show that the gains cannot be explained by trivial parameter sharing alone; they arise from the full collaborative design.

6 Discussion

C-NUTS is intentionally conservative in scope: it modifies only the warmup phase and leaves the post-warmup Markov kernel unchanged. This design choice is both practical and theoretical. Practically, it makes the method easy to incorporate into existing NUTS implementations. Theoretically, it localizes the effect of collaboration to metric estimation, which is the component most directly improved by multi-chain information sharing.

In terms of the effectiveness of the method. The empirical evidence suggests that the principal advantage of C-NUTS is not a change in the underlying Hamiltonian dynamics, but a reduction in adaptation noise. A single chain provides a noisy and often short geometric view of the posterior during warmup, particularly in strongly correlated or funnel-like targets. Pooling position information across chains produces a more stable estimate of global covariance structure, which in turn yields a better-aligned Euclidean metric. The downstream consequences—larger effective step sizes, shallower trees, and higher ESS/sec—are exactly what one would expect when preconditioning improves.

When collaboration is appropriate. C-NUTS is most effective when multiple chains explore the same posterior geometry, even if they do so from dispersed initial conditions. This includes many unimodal and weakly multimodal posteriors in applied Bayesian statistics. In such settings, parallel chains are usually already run for convergence assessment, and C-NUTS extracts additional value from this existing computation by turning the ensemble into a geometric estimator rather than using it solely for diagnostics.

In terms of failure modes and limits. The method is not intended as a universal remedy for pathological posterior structure. If chains remain trapped in distinct basins with incompatible means or incompatible local curvatures, naive sharing can be harmful; this motivates the protection mechanism developed in Section 4. Even with protection, C-NUTS does not solve the fundamental problem of mode discovery. In genuinely multimodal posteriors requiring barrier crossing, methods such as tempering, sequential Monte Carlo, or explicit mode-jumping proposals remain necessary.

A second limitation is computational representation in very high dimension. Dense covariance adaptation scales as $O(d^2)$ in memory and update cost, which eventually becomes prohibitive. This is not unique to C-NUTS, but collaborative dense adaptation inherits the same bottleneck. For large-scale problems, a low-rank-plus-diagonal or block-structured variant is likely the most promising extension.

Relation to existing geometry-aware samplers. C-NUTS occupies a middle ground between standard Euclidean NUTS and more sophisticated manifold methods such as RMHMC [23]. Unlike RMHMC, our method does not require Hessians or position-dependent metric tensors; unlike standard NUTS warmup, it explicitly uses ensemble information to stabilize global preconditioning. In this sense, C-NUTS should be viewed as a low-cost geometric enhancement rather than a competing manifold framework. It is especially attractive in applications where dense Euclidean adaptation is already beneficial but noisy.

Practical implications. A notable implication of our results is conceptual: multi-chain execution in modern Bayesian software need not be treated purely as a diagnostic convenience. Parallel chains can also serve as a statistically efficient source of adaptation information. This perspective is broadly compatible with current practice, requires minimal user intervention, and may extend beyond covariance estimation to other warmup quantities, such as step-size stabilization, blockwise preconditioners, or initialization of richer geometric approximations.

Future directions. Several extensions appear natural. First, a low-rank collaborative metric could extend the method to substantially larger parameter spaces. Second, one could investigate robust aggregation rules that downweight outlying chains rather than fully disabling collaboration when protection

is triggered. Third, a hybrid approach combining collaborative global preconditioning with local curvature corrections may capture some benefits of manifold methods at much lower cost. Finally, a broader benchmark suite covering generalized linear mixed models, latent Gaussian models, and probabilistic deep learning would further clarify the regime in which collaborative adaptation delivers the greatest advantage.

7 Conclusion

We introduced Collaborative NUTS (C-NUTS), a simple modification of NUTS warmup that pools covariance information across parallel chains to improve mass-matrix adaptation. The key idea is straightforward: because multiple chains collectively provide a richer geometric view of the target during warmup, the resulting metric estimate is less noisy than its single-chain counterpart. Better preconditioning then yields more efficient Hamiltonian trajectories after warmup.

Across synthetic and realistic benchmarks, C-NUTS consistently improves metric estimation, reduces tree depth, and increases effective sample size per unit time, often by substantial margins. These gains are especially pronounced when warmup is short, posterior correlations are strong, or a few poorly conditioned directions dominate mixing. At the same time, a lightweight protection mechanism prevents harmful collaboration in multimodal settings, preserving robustness when the assumptions behind metric sharing fail.

Methodologically, C-NUTS requires only minimal changes to existing NUTS implementations and preserves the standard fixed-kernel sampler after adaptation. Conceptually, it suggests a broader lesson: parallel chains are not only a diagnostic instrument but also a valuable source of statistical information for sampler adaptation. We hope this viewpoint motivates further work on collaborative warmup strategies for scalable and reliable Bayesian computation.

Competing Interests

The authors declare that they have no competing interests.

Funding

The authors received no specific funding for this work.

References

- [1] Hartmann, M., Girolami, M.A., Klami, A.: Lagrangian manifold monte carlo on monge patches. ArXiv **abs/2202.00755** (2022)
- [2] Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid monte carlo. *Physics Letters B* **195**(2), 216–222 (1987) [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
- [3] Neal, R.M.: Mcmc using hamiltonian dynamics, 113–162 (2011)
- [4] Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M.A., Guo, J., Li, P., Riddell, A.: Stan: A probabilistic programming language. *Journal of Statistical Software* **76**(1), 1–32 (2017) <https://doi.org/10.18637/jss.v076.i01>
- [5] Hoffman, M.D., Gelman, A.: The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research* **15**(47), 1593–1623 (2014)
- [6] Nesterov, Y.: Primal-dual subgradient methods for convex problems. *Mathematical Programming* **120**(1), 221–259 (2009) <https://doi.org/10.1007/s10107-007-0149-x>
- [7] Welford, B.P.: Note on a method for calculating corrected sums of squares and products. *Technometrics* **4**(3), 419–420 (1962) <https://doi.org/10.1080/00401706.1962.10490022>

- [8] Roberts, G.O., Rosenthal, J.S.: Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of Applied Probability* **44**(2), 458–475 (2007) <https://doi.org/10.1239/jap/1183667414>
- [9] Andrieu, C., Doucet, A., Holenstein, R.: Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B* **72**(3), 269–342 (2010) <https://doi.org/10.1111/j.1467-9868.2009.00736.x>
- [10] Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences. *Statistical Science* **7**(4), 457–472 (1992) <https://doi.org/10.1214/ss/1177011136>
- [11] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., Bürkner, P.-C.: Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of mcmc. *Bayesian Analysis* **16**(2), 667–718 (2021) <https://doi.org/10.1214/20-BA1221>
- [12] Goodman, J.R., Weare, J.: Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science* **5**(1), 65–80 (2010)
- [13] Foreman-Mackey, D., Hogg, D.W., Lang, D., Goodman, J.B.: emcee: The mcmc hammer. *Publications of the Astronomical Society of the Pacific* **125**, 306–312 (2012)
- [14] Haario, H., Saksman, E., Tamminen, J.: An adaptive metropolis algorithm. *Bernoulli* **7**, 223–242 (2001)
- [15] Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics* **53**, 343–367 (2003)
- [16] Salvatier, J., Wiecki, T.V., Fonnesbeck, C.J.: Probabilistic programming in python using pymc3. *PeerJ Prepr.* **4**, 1686 (2016)
- [17] Phan, D., Pradhan, N., Jankowiak, M.: Composable effects for flexible and accelerated probabilistic programming in numpyro. *ArXiv abs/1912.11554* (2019)
- [18] Betancourt, M.: A conceptual introduction to hamiltonian monte carlo. *arXiv: Methodology* (2017)
- [19] Betancourt, M.: A general metric for riemannian manifold hamiltonian monte carlo. *arXiv preprint* (2017)
- [20] Stan Development Team: Stan Reference Manual, Version 2.37. Stan Development Team, (2025). Stan Development Team. Accessed: 2026-08-18. <https://mc-stan.org/docs/reference-manual/>
- [21] Kumar, R., Carroll, C., Hartikainen, A., Martin, O.: ArviZ: a unified library for exploratory analysis of bayesian models in python. *Journal of Open Source Software* **4**(33), 1143 (2019) <https://doi.org/10.21105/joss.01143>
- [22] Rubin, D.B.: Estimation in parallel randomized experiments. *Journal of Educational Statistics* **6**(4), 377–401 (1981) <https://doi.org/10.2307/1164588>
- [23] Girolami, M., Calderhead, B.: Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**(2), 123–214 (2011) <https://doi.org/10.1111/j.1467-9868.2010.00765.x>

Appendix A Computational cost Proofs

We analyze the additional cost introduced by collaborative covariance adaptation separately from the cost of NUTS trajectory simulation. At the same time, the computational impact of collaborative

adaptation was evaluated through experiments. Let

$$\delta_t^{(i)} = \theta_t^{(i)} - \mu_t, \quad \mu_t = \frac{1}{N} \sum_{i=1}^N \theta_t^{(i)},$$

and define the ensemble scatter matrix by

$$S_t = \frac{1}{N} \sum_{i=1}^N \delta_t^{(i)} \delta_t^{(i)\top}. \quad (\text{A1})$$

Computing the ensemble mean requires $O(Nd)$ operations. For each chain, the vector $\delta_t^{(i)} \in \mathbb{R}^d$ is used to form a dense outer product,

$$\delta_t^{(i)} \delta_t^{(i)\top} \in \mathbb{R}^{d \times d},$$

which requires $O(d^2)$ operations. Summing these outer products over N chains therefore gives the total bookkeeping cost is

$$\mathcal{C}_{\text{scatter}}(N, d) = O(Nd) + O(Nd^2) = O(Nd^2) \quad (\text{A2})$$

for dense covariance adaptation.

To examine this scaling empirically, we measure the wall-clock time of the scatter update for several dimensions while keeping the number of chains fixed.

$$\mathcal{C}_{\text{scatter}}(N, d) = aNd^2 + b + \varepsilon, \quad (\text{A3})$$

where aNd^2 is the dimension-dependent term, b represents fixed dispatch and memory-management overhead, and ε captures timing variability. Figure A1(a) reports the measured times and the fitted quadratic model. Figure A1(b) shows the corresponding normalized quantity

$$\frac{\mathcal{C}_{\text{scatter}}(N, d) - \hat{b}}{Nd^2} \approx a, \quad (\text{A4})$$

which provides a diagnostic for whether the fitted Nd^2 term captures the dominant dimension-dependent cost. Deviations from a perfectly constant curve are expected because the measurements are affected by finite-sample timing noise, cache behavior, and low-level linear-algebra implementation details.

We next measure the end-to-end cost of a PyMC NUTS warmup under the same target distribution, number of chains, and number of tuning iterations. The resulting wall-clock time, $\mathcal{C}_{\text{warmup}}(N, d)$, includes gradient evaluations, trajectory construction, step-size adaptation, internal mass-matrix adaptation, and parallel chain scheduling. These results are shown in Figure A1(c). This panel is intended to provide the relevant computational scale against which the additional dense covariance calculation should be interpreted, rather than to isolate a single component of the NUTS implementation.

For this comparison, we report the bookkeeping fraction

$$\rho(N, d) = \frac{\mathcal{C}_{\text{scatter}}(N, d)}{\mathcal{C}_{\text{warmup}}(N, d)}. \quad (\text{A5})$$

Figure A1(d) displays $\rho(N, d)$ together with uncertainty estimates obtained from repeated runs. The quantity in (A5) corresponds to the conservative case in which one dense scatter update is performed per warmup iteration. Thus, for a moderate number of chains and update frequencies larger than one, the covariance bookkeeping is expected to remain secondary to the gradient and trajectory-construction costs that dominate NUTS warmup.

The subsequent regularization operation

$$\tilde{S}_t = S_t + \lambda I_d, \quad (\text{A6})$$

requires only $O(d)$ operations if implemented by modifying the diagonal. Thus, it does not change the leading-order complexity. If a diagonal mass matrix is used, only the elementwise squared deviations are required, and the corresponding cost reduces to $O(Nd)$.

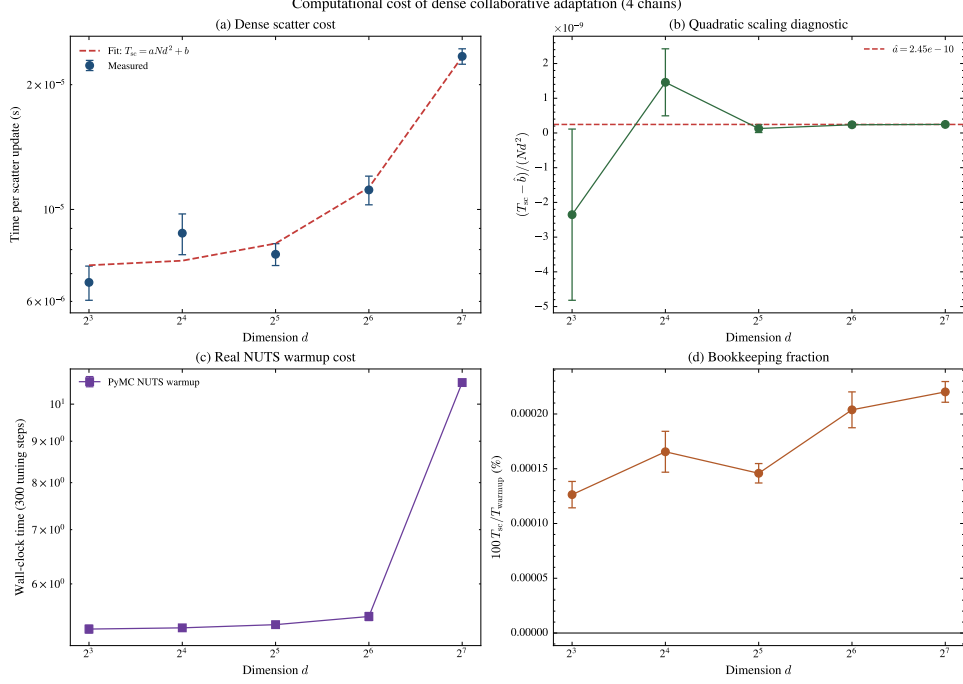


Fig. A1: Computational cost of dense collaborative covariance bookkeeping relative to real NUTS warmup. Panel (a) shows the measured time for one dense ensemble scatter update as a function of the state dimension, together with the fitted model $aNd^2 + b$. Panel (b) reports the fitted-intercept-adjusted quantity $(T_{\text{scatter}} - \hat{b}) / (Nd^2)$ as a diagnostic of quadratic scaling. Panel (c) shows the wall-clock time of a genuine PyMC NUTS warmup with N parallel chains and a fixed number of tuning iterations. Panel (d) reports the ratio $T_{\text{scatter}} / T_{\text{warmup}}$, corresponding to one scatter update per warmup iteration; error bars denote standard errors across repeated runs. The bookkeeping fraction is a conservative diagnostic of the additional covariance-computation cost and should be divided by the update interval k when scatter updates are performed only every k warmup iterations.

The additional cost should be compared with the cost of the NUTS transition itself. If chain i generates $L_t^{(i)}$ leapfrog states during iteration t , and one gradient evaluation has cost $G(d)$, then the total transition cost is approximately

$$\mathcal{C}_{\text{NUTS},t} = O\left(\sum_{i=1}^N L_t^{(i)} G(d)\right). \quad (\text{A7})$$

Consequently, the per-iteration relative overhead of dense collaborative adaptation can be expressed as

$$\text{OH}_t = \frac{\mathcal{C}_{\text{scatter}}(N, d)}{\mathcal{C}_{\text{NUTS},t}} \approx \frac{c_s Nd^2}{c_g G(d) \sum_{i=1}^N L_t^{(i)}}, \quad (\text{A8})$$

where c_s and c_g denote implementation-dependent constants. This ratio decreases when the NUTS trajectories contain more leapfrog steps or when gradient evaluations are computationally expensive. Since covariance adaptation is performed only during warmup and the number of parallel chains is typically moderate, the additional matrix-update cost is expected to be secondary to trajectory simulation in the regimes considered here.

Appendix B Proof of Theorem 2

We restate the theorem for convenience.

Theorem 2 Suppose the target is a multivariate Gaussian $\mathcal{N}(0, \Sigma)$ and the chains are initialized from a distribution with finite fourth moments. Let $\hat{\Sigma}_t$ be the collaborative estimator with fixed learning rate $\alpha_t \equiv \alpha$. Then

$$\mathbb{E}[\|\hat{\Sigma}_t - \Sigma\|_F] \leq (1 - \alpha)^t \|\Sigma_0 - \Sigma\|_F + \sqrt{\frac{d(d+1)}{N\alpha}} \cdot C, \quad (\text{B9})$$

where C depends on the fourth moments of the initialization.

Proof Let $E_t = \hat{\Sigma}_t - \Sigma$ denote the error matrix. The collaborative update see formula (15) is

$$\hat{\Sigma}_t = (1 - \alpha)\hat{\Sigma}_{t-1} + \alpha S_t, \quad (\text{B10})$$

where $S_t = \frac{1}{N} \sum_{i=1}^N (\theta_t^{(i)} - \mu_t)(\theta_t^{(i)} - \mu_t)^\top$ is the ensemble scatter matrix.

Step 1: Bias decomposition. Subtracting Σ from both sides:

$$E_t = (1 - \alpha)E_{t-1} + \alpha(S_t - \Sigma). \quad (\text{B11})$$

Unrolling the recursion:

$$E_t = (1 - \alpha)^t E_0 + \alpha \sum_{s=1}^t (1 - \alpha)^{t-s} (S_s - \Sigma). \quad (\text{B12})$$

Taking the Frobenius norm and expectation:

$$\mathbb{E}[\|E_t\|_F] \leq (1 - \alpha)^t \|E_0\|_F + \alpha \sum_{s=1}^t (1 - \alpha)^{t-s} \mathbb{E}[\|S_s - \Sigma\|_F]. \quad (\text{B13})$$

Step 2: Variance of ensemble scatter. For the Gaussian target $\mathcal{N}(0, \Sigma)$, the ensemble states $\{\theta_t^{(i)}\}_{i=1}^N$ are i.i.d. samples from the stationary distribution (assuming the chain has mixed; for the warmup analysis we consider the idealized case where each $\theta_t^{(i)} \sim \mathcal{N}(0, \Sigma)$ independently).

The sample covariance S_t is the maximum likelihood estimator of Σ from N i.i.d. samples. For Gaussian data, the Wishart distribution gives:

$$\mathbb{E}[\|S_t - \Sigma\|_F^2] = \frac{1}{N} \left(\text{tr}(\Sigma)^2 + \text{tr}(\Sigma^2) \right) \leq \frac{d(d+1)}{N} \lambda_{\max}(\Sigma)^2. \quad (\text{B14})$$

By Jensen's inequality:

$$\mathbb{E}[\|S_t - \Sigma\|_F] \leq \sqrt{\mathbb{E}[\|S_t - \Sigma\|_F^2]} \leq \sqrt{\frac{d(d+1)}{N}} \lambda_{\max}(\Sigma). \quad (\text{B15})$$

Step 3: Geometric series. Substituting into the error bound:

$$\mathbb{E}[\|E_t\|_F] \leq (1 - \alpha)^t \|E_0\|_F + \alpha \sqrt{\frac{d(d+1)}{N}} \lambda_{\max}(\Sigma) \sum_{s=1}^t (1 - \alpha)^{t-s}. \quad (\text{B16})$$

The sum is a geometric series:

$$\sum_{s=1}^t (1 - \alpha)^{t-s} = \sum_{k=0}^{t-1} (1 - \alpha)^k = \frac{1 - (1 - \alpha)^t}{\alpha} < \frac{1}{\alpha}. \quad (\text{B17})$$

Therefore:

$$\mathbb{E}[\|E_t\|_F] \leq (1 - \alpha)^t \|E_0\|_F + \sqrt{\frac{d(d+1)}{N\alpha}} \lambda_{\max}(\Sigma). \quad (\text{B18})$$

Setting $C = \lambda_{\max}(\Sigma)$ completes the proof. $\square$

Remark 3 The bound assumes i.i.d. samples for simplicity. In practice, the chains are Markovian and initialized out of stationarity. The first term $(1 - \alpha)^t \|E_0\|_F$ captures the forgetting of initialization, while the second term captures the fundamental Monte Carlo variance. The $N^{-1/2}$ scaling is the key improvement over single-chain adaptation.

Appendix C Additional theoretical results for collaborative mass-matrix adaptation

This appendix provides the technical details underlying the discussion in Section 4.3. Our goal is twofold. First, we establish the structural properties required for the warmup update to remain well defined, namely symmetry, positive definiteness, and numerical stability. Second, under additional probabilistic assumptions, we characterize the statistical behavior of the collaborative covariance estimator and explain why pooling information across chains can reduce estimation noise when the ensemble is locally coherent. These results concern the adaptation mechanism used during warmup. They are not intended as a complete ergodicity analysis of a perpetually adapting Markov chain, since adaptation is terminated before the sampling phase begins.

C.1 Structural properties of the regularized update

Let $\widehat{\Sigma}_t \in \mathbb{R}^{d \times d}$ denote the collaborative covariance estimator constructed at warmup iteration t from the available ensemble states. By construction, $\widehat{\Sigma}_t$ is symmetric. As an empirical covariance-type matrix, it is also positive semidefinite:

$$x^\top \widehat{\Sigma}_t x \geq 0 \quad \text{for all } x \in \mathbb{R}^d.$$

We define the regularized mass matrix by

$$M_t = \widehat{\Sigma}_t + \lambda I_d, \quad \lambda > 0. \quad (\text{C19})$$

Proposition 1 (Regularization preserves admissibility) *If $\widehat{\Sigma}_t$ is symmetric positive semidefinite and $\lambda > 0$, then M_t defined by (C19) is symmetric positive definite.*

Proof Symmetry follows immediately from the symmetry of $\widehat{\Sigma}_t$ and I_d . For any nonzero $x \in \mathbb{R}^d$,

$$x^\top M_t x = x^\top \widehat{\Sigma}_t x + \lambda \|x\|_2^2.$$

The first term is nonnegative because $\widehat{\Sigma}_t \succeq 0$, while the second term is strictly positive because $\lambda > 0$ and $x \neq 0$. Hence $x^\top M_t x > 0$ for all $x \neq 0$, so $M_t \succ 0$. $\square$

Proposition 1 ensures that the mass matrix remains invertible throughout warmup. Consequently, the Gaussian momentum distribution and the associated kinetic energy are well defined at every update step.

In practice, the raw covariance estimate may be smoothed across warmup iterations. A generic recursive update has the form

$$M_{t+1} = (1 - \eta_t)M_t + \eta_t \widetilde{M}_t, \quad \eta_t \in (0, 1), \quad (\text{C20})$$

where M_t is the current mass matrix and $\widetilde{M}_t$ is a newly constructed candidate matrix.

Proposition 2 (Stability under convex averaging) *If both M_t and $\widetilde{M}_t$ are symmetric positive definite and $\eta_t \in (0, 1)$, then M_{t+1} defined by (C20) is also symmetric positive definite.*

Proof Symmetry is preserved under linear combinations. For any nonzero $x \in \mathbb{R}^d$,

$$x^\top M_{t+1} x = (1 - \eta_t)x^\top M_t x + \eta_t x^\top \widetilde{M}_t x.$$

Since $M_t \succ 0$ and $\widetilde{M}_t \succ 0$, both quadratic forms on the right-hand side are strictly positive. Because $\eta_t \in (0, 1)$, both coefficients are positive, hence $x^\top M_{t+1} x > 0$ for all $x \neq 0$. $\square$

Therefore, once the recursion is initialized with a valid mass matrix, all subsequent warmup iterates remain in the symmetric positive definite cone. This invariance is the basic stability property required for a numerically safe adaptation scheme.

The regularization in (C19) also yields a uniform spectral lower bound. If $\lambda_{\min}(A)$ denotes the smallest eigenvalue of a symmetric matrix A , then

$$\lambda_{\min}(M_t) \geq \lambda, \quad \text{equivalently} \quad M_t \succeq \lambda I_d. \quad (\text{C21})$$

As a consequence,

$$\|M_t^{-1}\|_{\text{op}} \leq \lambda^{-1}. \quad (\text{C22})$$

This lower bound prevents near-singularity of the adapted mass matrix and stabilizes the linear algebra required by Hamiltonian integration.

C.2 Correctness of the sampling phase

The collaborative update is used only during warmup and is frozen before the sampling phase begins. Accordingly, after warmup the algorithm evolves according to a fixed NUTS transition kernel with a fixed mass matrix. The following statement formalizes the role of this design choice.

Theorem 3 (Post-warmup correctness) *Let $K_{\text{NUTS}}(\theta, \cdot; M)$ denote the NUTS transition kernel targeting π with fixed mass matrix $M \succ 0$. Suppose warmup terminates at iteration T_{warmup} and returns a deterministic matrix $M_\star$ measurable with respect to the warmup history. Then, conditional on $M_\star$, the sampling-phase chain*

$$\theta_{t+1} \sim K_{\text{NUTS}}(\theta_t, \cdot; M_\star), \quad t \geq T_{\text{warmup}},$$

is a time-homogeneous Markov chain with invariant distribution π .

Proof Once warmup terminates, $M_\star$ is fixed and no further adaptation occurs. Conditional on $M_\star$, the transition rule is therefore given by repeated application of the same kernel $K_{\text{NUTS}}(\cdot, \cdot; M_\star)$. By assumption, this kernel is the standard NUTS transition kernel constructed to leave π invariant for any fixed positive-definite mass matrix. Hence the post-warmup chain is time-homogeneous and has invariant distribution π . $\square$

Theorem 3 clarifies why our analysis does not require a joint ergodicity theory for a perpetually adapting kernel: the only phase used for Monte Carlo estimation is the post-warmup phase, where adaptation has ceased. The role of the warmup update is thus purely preparatory, namely to select a numerically suitable fixed mass matrix for the subsequent NUTS transition.

C.3 Statistical behavior of the collaborative covariance estimator

We next study the covariance estimator obtained by pooling information across N chains during warmup. The purpose of this section is not to prove full adaptive-MCMC convergence, but to quantify how collaboration changes the bias and variance of the geometric estimator under simplifying assumptions.

C.3.1 Ensemble scatter and its expectation

Let $\{\theta^{(i)}\}_{i=1}^N$ be N random vectors in $\mathbb{R}^d$ and define the ensemble mean

$$\bar{\theta} = \frac{1}{N} \sum_{i=1}^N \theta^{(i)}.$$

The associated ensemble scatter matrix is

$$S_N = \frac{1}{N} \sum_{i=1}^N (\theta^{(i)} - \bar{\theta})(\theta^{(i)} - \bar{\theta})^\top. \quad (\text{C23})$$

Lemma 1 (Expectation of the ensemble scatter) Suppose $\theta^{(1)}, \dots, \theta^{(N)}$ are identically distributed with mean μ and covariance matrix Σ . Then

$$\mathbb{E}[S_N] = \left(1 - \frac{1}{N}\right) \Sigma - \frac{1}{N^2} \sum_{i \neq j} \text{Cov}(\theta^{(i)}, \theta^{(j)}). \quad (\text{C24})$$

In particular, if the chains are pairwise uncorrelated, then

$$\mathbb{E}[S_N] = \frac{N-1}{N}\Sigma. \quad (\text{C25})$$

Proof Using

$$S_N = \frac{1}{N} \sum_{i=1}^N \theta^{(i)} (\theta^{(i)})^\top - \bar{\theta} \bar{\theta}^\top,$$

we obtain

$$\mathbb{E}[S_N] = \Sigma + \mu \mu^\top - \mathbb{E}[\bar{\theta} \bar{\theta}^\top].$$

Expanding the second term gives

$$\mathbb{E}[\bar{\theta} \bar{\theta}^\top] = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \mathbb{E}[\theta^{(i)} (\theta^{(j)})^\top].$$

Separating diagonal and off-diagonal terms and simplifying yields (C24). Equation (C25) follows when all cross-covariances vanish. $\square$

Lemma 1 makes explicit the finite- N bias induced by centering with the ensemble mean. For independent stationary chains, the bias is multiplicative and of order N^{-1} .

C.3.2 Variance reduction under independent chains

To quantify the effect of collaboration, we now consider the idealized regime in which the chains are independent draws from a common local distribution.

Lemma 2 (Fluctuation size in the Gaussian case) Let $\theta^{(1)}, \dots, \theta^{(N)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \Sigma)$. Then

$$\mathbb{E}[\|S_N - \mathbb{E}[S_N]\|_F^2] = O\left(\frac{1}{N}\right), \quad (\text{C26})$$

with an explicit constant depending only on Σ . More precisely,

$$\mathbb{E}[\|S_N - \mathbb{E}[S_N]\|_F^2] = \frac{1}{N} \left\{ \text{tr}(\Sigma)^2 + \text{tr}(\Sigma^2) \right\} - \frac{1}{N(N-1)} \|\Sigma\|_F^2. \quad (\text{C27})$$

Proof For Gaussian samples, the centered sample scatter admits a Wishart representation. The stated expression follows from standard second-moment formulas for Wishart matrices; see. $\square$

Equation (C27) formalizes the basic variance-reduction mechanism: when the ensemble is locally coherent, pooling across N chains reduces the estimator fluctuation at the canonical rate N^{-1} .

Remark 4 The Gaussian calculation is included because it yields a transparent closed form. Analogous N^{-1} fluctuation bounds can be obtained under broader moment or sub-Gaussian assumptions, although with less explicit constants. For the purposes of the main text, the key message is the scaling law rather than the exact constant.

C.3.3 A simple recursive estimator

A smoothed covariance estimate can be written as

$$\hat{\Sigma}_t = (1 - \alpha_t) \hat{\Sigma}_{t-1} + \alpha_t S_{N,t}, \quad \alpha_t \in (0, 1], \quad (\text{C28})$$

where $S_{N,t}$ denotes the ensemble scatter computed at warmup iteration t .

The recursion (C28) is an exponentially weighted average of noisy covariance surrogates. Under local stationarity and weak dependence assumptions, it behaves as a stochastic approximation scheme toward the mean of $S_{N,t}$. The next proposition states a finite-time mean-square bound in the simplest idealized setting.

Proposition 3 (Finite-time mean-square bound under stationarity) *Assume that $\{S_{N,t}\}_{t \geq 1}$ are identically distributed with mean $\Sigma^\dagger$ and satisfy*

$$\mathbb{E}[\|S_{N,t} - \Sigma^\dagger\|_F^2] \leq \frac{\kappa}{N}$$

for some $\kappa < \infty$. If (C28) is run with a constant learning rate $\alpha_t \equiv \alpha \in (0, 1)$, then

$$\mathbb{E}[\|\hat{\Sigma}_t - \Sigma^\dagger\|_F^2] \leq (1 - \alpha)^{2t} \|\hat{\Sigma}_0 - \Sigma^\dagger\|_F^2 + \frac{\alpha}{2 - \alpha} \frac{\kappa}{N}. \quad (\text{C29})$$

Proof Let $E_t = \hat{\Sigma}_t - \Sigma^\dagger$. Then

$$E_t = (1 - \alpha)E_{t-1} + \alpha(S_{N,t} - \Sigma^\dagger).$$

Taking squared Frobenius norms, then expectations, and using $\mathbb{E}[S_{N,t} - \Sigma^\dagger] = 0$ gives

$$\mathbb{E}[\|E_t\|_F^2] = (1 - \alpha)^2 \mathbb{E}[\|E_{t-1}\|_F^2] + \alpha^2 \mathbb{E}[\|S_{N,t} - \Sigma^\dagger\|_F^2].$$

Applying the variance bound and unrolling the recursion yields

$$\mathbb{E}[\|E_t\|_F^2] \leq (1 - \alpha)^{2t} \|E_0\|_F^2 + \alpha^2 \frac{\kappa}{N} \sum_{s=0}^{t-1} (1 - \alpha)^{2s}.$$

The geometric series satisfies

$$\sum_{s=0}^{t-1} (1 - \alpha)^{2s} \leq \frac{1}{1 - (1 - \alpha)^2} = \frac{1}{\alpha(2 - \alpha)},$$

which gives (C29). $\square$

Proposition 3 shows that, in a stylized stationary regime, collaboration reduces the stochastic error floor of the recursive estimator by a factor proportional to N^{-1} . This is the theoretical counterpart of the empirical stabilization observed in the warmup experiments.

Remark 5 The target quantity in Proposition 3 is $\Sigma^\dagger = \mathbb{E}[S_{N,t}]$, not necessarily the exact posterior covariance. In particular, if $S_{N,t}$ is defined as in (C23), then even under independence its expectation differs from the underlying covariance by the finite- N factor $(N - 1)/N$. This distinction is immaterial for the structural validity of the mass matrix, but it is relevant when interpreting the estimator statistically.

C.4 Protection mechanism as a diagnostic safeguard

The collaborative update is beneficial only when the ensemble of chains is sufficiently coherent that the pooled scatter reflects local posterior geometry. If different chains occupy well-separated regions, naive pooling can instead encode between-mode separation. For this reason, the method includes a protection rule that suppresses collaborative updates when the ensemble appears overly dispersed.

Let

$$R_{\text{proxy}} = \frac{\max_{1 \leq i \leq N} \|\theta_t^{(i)} - \mu_t\|_2}{\sqrt{\text{tr}(\Sigma_t)/d}}, \quad (\text{C30})$$

where μ_t and Σ_t denote, respectively, the ensemble mean and ensemble covariance proxy at iteration t . The numerator measures the largest chain deviation from the ensemble center, while the denominator rescales by an average marginal spread. Accordingly, large values of R_{proxy} indicate that at least one chain lies unusually far from the bulk of the ensemble relative to the local scale.

The statistic (C30) should be interpreted as a diagnostic safeguard rather than as a formal multi-modality test. Its role is conservative: when the ensemble is roughly unimodal and locally concentrated, R_{proxy} tends to remain moderate; when a subset of chains drifts far from the others, the ratio increases and can be used to disable covariance pooling temporarily.

Under stylized Gaussian configurations, one can justify this interpretation at least qualitatively. If $\theta_t^{(1)}, \dots, \theta_t^{(N)}$ are approximately independent draws from a common light-tailed local distribution, concentration of Euclidean norms implies that the numerator and denominator of (C30) remain of comparable scale with high probability. By contrast, under separated-cluster configurations the numerator reflects between-cluster displacement and therefore grows relative to the within-cluster scale. For this reason, thresholding R_{proxy} provides a practical warning signal against erroneous cross-mode pooling, even though a complete finite-sample analysis is beyond the scope of the present paper.

C.5 Optional low-rank extension

In very high dimensions, storing and updating a dense $d \times d$ mass matrix may become expensive. A natural extension is to use a structured approximation of the form

$$\hat{\Sigma}_t^{(r)} = D_t + V_t V_t^\top, \quad (\text{C31})$$

where D_t is diagonal and $V_t \in \mathbb{R}^{d \times r}$ with $r \ll d$. This replaces $O(d^2)$ storage with $O(dr)$ storage while preserving a low-rank approximation to the dominant covariance directions.

The low-rank variant is not part of the core method analyzed in the main text, but it is compatible with the same regularization principle. Indeed, if D_t has strictly positive diagonal entries, then

$$D_t + V_t V_t^\top \succ 0,$$

so the resulting mass matrix remains admissible after adding the same ridge term as in (C19). A full statistical analysis of the low-rank approximation error is left for future work.

C.6 Summary of the appendix

The results of this appendix support the claims made in Section 4 at two different levels.

First, at the structural level, Propositions 1 and 2, together with the spectral bounds (C21)–(C22), show that the collaborative warmup update remains symmetric positive definite and numerically stable under regularization and convex smoothing.

Second, at the statistical level, Lemma 1, Lemma 2, and Proposition 3 explain why collaborative pooling can reduce estimator noise when the ensemble is locally coherent, while the discussion in Section C.4 clarifies why a protection mechanism is needed when this coherence fails.

Together, these results justify the collaborative update as a well-defined and practically stable warmup mechanism, while keeping the correctness of the final sampling phase within the standard fixed-kernel theory for NUTS.