GENERALIZED DIFFERENTIABLE HAMILTONIAN DYNAMIC NESTED SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Nested sampling reduces the evidence integral to a one-dimensional quadrature over prior volume, at the cost of a complex inner problem: at every iteration, a new sample must be drawn from the prior restricted to a likelihood level set. These constrained draws dominate the running time and can bias the estimator. We present Generalized Differentiable Hamiltonian Dynamic Nested Sampling (GDHDNS), built around three complementary ideas to address this bottleneck. First, the live-point covariance provides a natural local preconditioner: its inverse defines a mass matrix that contracts with the constrained region, costs $\mathcal{O}(Nd^2)$, and requires no Hessian. Second, a reflected Hamiltonian flow is integrable inside a convex level set, so its endpoint remains correlated with the point seeding it no matter how long the trajectory; this biases the evidence upward as a result. We break that correlation with a preconditioned slice refresh while retaining the flow's gradient-guided efficiency. Third, we show that batched removal has the same volume schedule as sequential removal, permitting unbiased vectorization, and that a run returns $\partial \log \mathcal{Z}/\partial\phi$ with the evidence. We evaluate GDHDNS on sweeps up to $d = 128$, a strong-lensing reconstruction problem, an interatomic potential, and an SDE mixed-effects model. Across the dimension sweeps it is the only method with unbiased evidence at every tested dimension, while running 5–24$\times$ faster than `dynesty` and up to four orders of magnitude faster than `PolyChord`.

1 INTRODUCTION

Bayesian model comparison rests on the *evidence*, or *marginal likelihood*,

$$\mathcal{Z} = \int \mathcal{L}(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}, \tag{1}$$

with $\mathcal{L}(\boldsymbol{\theta})$ the *likelihood* function and $\pi(\boldsymbol{\theta})$ the *prior distribution*. Estimating it is, however, often intractable because of high-dimensionality and complexity of the likelihood, and is qualitatively harder than estimating posterior expectations, as it depends on how much prior volume the likelihood occupies, including in regions the posterior barely visits. *Nested sampling* (Skilling, 2006) addresses this problem by a Monte Carlo method that transforms the evidence integral in Eq. 1 into a one-dimensional quadrature over the prior volume enclosed by a likelihood contour, and returns weighted *posterior* samples as a by-product. It is a standard tool for evidence estimation across many fields and applications, including cosmology, gravitational-wave astronomy and particle physics (Feroz et al., 2009; Handley et al., 2015; Speagle, 2020; Buchner, 2021; Ashton et al., 2022).

At each iteration, the algorithm removes the lowest-likelihood *live-point*, replaces it with a fresh draw from the prior restricted to the level set $\{\boldsymbol{\theta} : \mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}^*\}$ defined by the likelihood $\mathcal{L}^*$ of that point, and updates the quadrature. The inner problem of producing that constrained draw is the bottleneck of nested sampling: it consumes nearly all of the running time and is where the estimator's bias originates. The cost of a run thus scales as the number of likelihood evaluations, which is $n_{\text{like}} \propto n_{\text{live}} \cdot f_{\text{sampler}} \cdot \mathcal{D}_{\text{KL}}(\mathcal{P} \,\|\, \Pi)$, with n_{live} the number of live points, f_{sampler} the average number of evaluations spent per replacement, and $\mathcal{D}_{\text{KL}}(\mathcal{P} \,\|\, \Pi)$ the Kullback-Leibler divergence between posterior and prior (Skilling, 2006; Ashton et al., 2022). The first and third are set by the problem and the accuracy asked of the estimate; f_{sampler} is the only one a sampler designer controls.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Region samplers build a geometric proxy of the constrained set and draw from it. The proxy must enclose the set, and its excess volume compounds with dimension, so $f_{\text{sampler}} \sim \mathcal{O}(e^{d/d_0})$ for a problem-dependent d_0. *Step samplers* run a short Markov Chain from a surviving live-point and pay $\mathcal{O}(d)$; this is `PolyChord`'s strategy (Handley et al., 2015). *Gradient-guided nested sampling* (GGNS; Lemos et al., 2024) makes the draw with reflective Hamiltonian dynamics: a particle moves in straight lines and reflects off the iso-likelihood surface along the log-likelihood gradient. Every reflection uses the full d-dimensional gradient, so a handful of them carries a point across the region. GGNS pairs this with automatic differentiation, trajectory reuse, a step-size heuristic, pruning, clustering and an $X\mathcal{L}$ termination rule, which made reflective dynamics practical for nested sampling.

However, two limitations emerge from GGNS that we address in this work. First, its fixed Euclidean mass matrix forces a single step-size to resolve the narrowest direction of the constrained region while traversing the widest. In nested sampling, both scales shrink by orders of magnitude during a run, beyond what a step-size controller can accommodate. The live points provide a more suitable metric: their covariance $\widehat{\Sigma}$ estimates the shape of the constrained region and contracts with it. Using $\widehat{\Sigma}^{-1}$ as the mass matrix costs $\mathcal{O}(Nd^2)$, requires neither a Hessian nor a positive-definiteness repair, and keeps the step-size useful throughout a run, where a Riemannian-manifold metric (Girolami & Calderhead, 2011; Betancourt, 2013) would be considerably more expensive and less stable. Second, the reflected flow is not *ergodic*: Gaussian level sets are ellipsoids, and billiards in ellipsoids are integrable. Its endpoint therefore remains correlated with its seed regardless of trajectory length. The live set then behaves as though it contains fewer than N independent points, the volume schedule fails, and $\log \widehat{\mathcal{Z}}$ is biased upward. This comes from the flow's integrability itself, not its discretization scheme; changing the step-size, trajectory length, or momentum refresh does not remove it. A constrained slice refresh does, because its one-dimensional invariance argument makes no assumption about the shape of the region.

Generalized Differentiable Hamiltonian Dynamic Nested Sampling (GDHDNS) combines these two changes with a batched outer loop and differentiable evidence estimation.[1] Section 4.1 preconditions the constrained draw with the live-point covariance. Section 4.2 develops the ergodicity argument and constructs a kernel that composes a persistent reflected flow with a preconditioned slice refresh; each stage preserving the constrained target. Section 4.3 regenerates a fixed fraction of the live set per generation, making all per-point work a single vectorized map, and shows that the resulting volume schedule matches the sequential one. Section 4.4 shows that the log-evidence returned by a run is differentiable with respect to the model parameters, with gradient given by the posterior-averaged score. Setting the mass matrix to the identity, fully refreshing the momentum, omitting the slice refresh, and using a one Newton step reflection recovers GGNS as a special case.

2 Related Work

Nested sampling. Skilling (2006) introduced the algorithm; Chopin & Robert (2010) established consistency and a central limit theorem, Higson et al. (2018) characterized its parameter-estimation error, and Salomone et al. (2025) recast it as sequential Monte Carlo (SMC) to remove the estimator's bias. Ashton et al. (2022) and Buchner (2023) surveyed the field. Implementations differ almost entirely in how they perform the constrained draw: `MultiNest` (Feroz et al., 2009) and `UltraNest` (Buchner, 2021) use ellipsoidal decompositions, `nessai` (Williams et al., 2021) normalizing flows, `PolyChord` (Handley et al., 2015) slice sampling, `dynesty` (Speagle, 2020) both under a dynamic schedule, and `JAXNS` (Albert, 2020) a vectorized reimplementation in `JAX`.

Reflective and gradient-guided constrained sampling. Hamiltonian slice sampling (Neal, 2003; Bloem-Reddy & Cunningham, 2016), Galilean Monte Carlo (Feroz & Skilling, 2013) and constrained HMC (Betancourt, 2011; Duane et al., 1987) replace rejection with *reflection* off the constraint surface, trading many cheap likelihood calls for a few gradient calls. GGNS (Lemos et al., 2024) made this practical with automatic differentiation, trajectory reuse, pruning, clustering and an $X\mathcal{L}$ termination rule, as well as a step-size heuristic that keeps the acceptance rate in a target range. Section 4.2 shows that a reflective flow is not ergodic on convex level sets and composes it with a preconditioned slice refresh. The refresh is ergodic irrespective of the region's shape and retains the gradient-guided flow's efficiency.

[1] Code available on GitHub (link omitted for blind review).

Preconditioning and generalized HMC. Riemannian-manifold HMC (RMHMC) preconditions with a position-dependent metric (Girolami & Calderhead, 2011), SoftAbs builds a smooth positive-definite metric from an indefinite Hessian (Betancourt, 2013), and NeuTra (Hoffman et al., 2019) learns the reparameterization instead. All three need second-order or learned information; our metric is read off the sampler's own state, which is free and already restricted to the region of interest. Partial momentum refreshment (Horowitz, 1991; Neal, 2011) suppresses random-walk behavior, and we carry it across constrained draws rather than within a single one. We deliberately do not adopt a no-U-turn stopping (NUTS) rule (Hoffman & Gelman, 2014): a trajectory length that depends on the trajectory leaves the returned position and the persistent momentum inconsistent, which invalidates the flow's invariance argument. Thus, we fix the flow length and adapt only the step-size.

Dynamic and vectorized nested sampling. *Dynamic nested sampling* (Higson et al., 2019; Speagle, 2020) allocates live points where they contribute most and typically removes several per iteration, and differentiable-programming frameworks (Bradbury et al., 2018) allow the loop to be compiled and vectorized (Albert, 2020). Our contribution here is twofold: we give the bookkeeping needed for batched removal to retain the bias properties of sequential removal (§4.3). Separately, differentiability of the *estimator* also allows an inference procedure to sit inside a learning pipeline, as in simulation-based inference or generative-flow-network training (Bengio et al., 2021; Lemos et al., 2024); GGNS uses gradients only internally, whereas our method explicitly exposes $\partial \log \mathcal{Z}/\partial\phi$ (§4.4) as a by-product of a run.

3 Background

Let $X(\lambda) = \int_{\mathcal{L}(\boldsymbol{\theta})>\lambda} \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$ be the prior mass enclosed by the contour at level λ. Since X is monotone, Eq. 1 becomes $\mathcal{Z} = \int_0^1 \mathcal{L}(X)\, dX$, a one-dimensional integral over prior volume rather than a d-dimensional integral over parameter space. Given N *live points* independently distributed as the prior restricted to $\{\mathcal{L} > \lambda\}$, the contour through the lowest-likelihood one encloses a fraction $t \sim \mathrm{Beta}(N, 1)$ of the current mass, so $\mathbb{E}[\log t] = -1/N$ whatever the target. Removing points one at a time, under this expectation, gives the schedule $\log(X_k) = -\sum_{i\le k} 1/N_i$ and the quadrature:

$$\widehat{\mathcal{Z}} = \sum_k w_k\, \mathcal{L}_k, \qquad w_k = X_{k-1} - X_k, \tag{2}$$

with $\mathcal{L}_k$ the *likelihood* of the k-th removed point. Those points, weighted by $w_k\mathcal{L}_k$, form the posterior representation, and the dominant error is $\sigma(\log \mathcal{Z}) \approx \sqrt{\mathcal{D}_{\mathrm{KL}}(\mathcal{P}\|\Pi)/N}$ (Skilling, 2006). The $\mathrm{Beta}(N, 1)$ law is exact only when the live points are *independent* draws from the constrained prior. A kernel that returns points correlated with the live set therefore invalidates the schedule and biases $\log \widehat{\mathcal{Z}}$, rather than merely increasing its variance. Because the law concerns order statistics alone, it holds under any removal order; §4.3 uses this fact. We implement two *stopping rules*: the standard rule based on the evidence remaining in the live set, and the $X\mathcal{L}$-peak criterion of Lemos et al. (2024). Every reported run uses the former, with the tolerances in Appendix C.

The constrained draw is then given by the following problem: given $\mathcal{L}^*$ and points known to satisfy $\mathcal{L} > \mathcal{L}^*$, produce a point distributed as π conditioned on $\mathcal{L} > \mathcal{L}^*$ and independent of the starting points. *Reflective Hamiltonian dynamics* (Neal, 2011) address it by simulating the dynamics of a fictitious particle. With position $\boldsymbol{\theta}$, momentum $\boldsymbol{p}$ and velocity $\boldsymbol{v} = \boldsymbol{M}^{-1}\boldsymbol{p}$ for a mass matrix $\boldsymbol{M}$, the particle moves freely, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}\,\Delta t$, until the next step would cross $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}^*$, where it reflects,

$$\boldsymbol{p}' = \boldsymbol{p} - 2\,\frac{\boldsymbol{n}^\top \boldsymbol{M}^{-1}\boldsymbol{p}}{\boldsymbol{n}^\top \boldsymbol{M}^{-1}\boldsymbol{n}}\,\boldsymbol{n}, \qquad \boldsymbol{n} = \nabla \log \mathcal{L}(\boldsymbol{\theta}), \tag{3}$$

and continues until a fixed trajectory length is reached. Eq. 3 preserves the *kinetic energy* $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}$ and reduces to $\boldsymbol{p}' = \boldsymbol{p} - 2(\boldsymbol{n}\cdot\boldsymbol{p})\boldsymbol{n}$ for $\boldsymbol{M} = \boldsymbol{I}$.

4 Methods

4.1 Preconditioning From the Live Ensemble

A fixed Euclidean metric gives a sampler one step-size to serve two *conflicting* demands: small enough to resolve the narrowest direction of the constrained region, and large enough to traverse

the widest. On an anisotropic target that ratio is already large at the first iteration, and both scales then shrink by orders of magnitude over a run, so a step-size tuned at the start is *useless* at the end. A Hessian-based position-dependent metric (Girolami & Calderhead, 2011; Betancourt, 2013) addresses the geometry but fits poorly here: it costs $\mathcal{O}(d^3)$ per evaluation once regularized, and measures the curvature of the likelihood rather than the shape of the set the sampler is confined to. On Neal's funnel, the likelihood curvature along the wide coordinates vanishes precisely where the funnel is widest, and the induced steps overshoot the prior support.

The live points, however, answer this question directly since they are a sample from the set the kernel must explore. We therefore take

$$\widehat{\Sigma} \;=\; \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{\theta}_i-\bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_i-\bar{\boldsymbol{\theta}})^\top, \qquad \boldsymbol{M} \;=\; \widehat{\Sigma}^{-1}, \tag{4}$$

where $\bar{\boldsymbol{\theta}}$ is the live-point *mean* and $\widehat{\Sigma}$ is recomputed once per generation. The resulting velocity, $\boldsymbol{v}=\widehat{\Sigma}\boldsymbol{p}$, has covariance $\widehat{\Sigma}$. A unit step therefore has the order of the current width in every direction, and the metric contracts with the region. One covariance and Cholesky factorization are required per generation, at $\mathcal{O}(Nd^2+d^3)$ cost amortized over the regenerated points.

Two limitations follow from the estimate: *(i)* a covariance describes a *single mode*; for a multimodal live set, it interpolates between modes, so whitening may do no more than avoid harm. Mode-aware preconditioning would require the clustering used by Handley et al. (2015) or Lemos et al. (2024), and §5.4 shows the cost of omitting it; and *(ii)* a covariance is a *global summary*, so it misses targets whose scale changes with position. In §4.2, we mix in coordinate directions scaled by the live points to address this case.

4.2 The Constrained-Sampling Kernel

The kernel starts from a uniformly chosen *surviving* live-point and returns a point above $\mathcal{L}^*$. It combines a *persistent* reflected Hamiltonian flow with a short preconditioned slice refresh, both using the metric in Eq. 4. Each preserves $\mathcal{U}\{\mathcal{L}>\mathcal{L}^*\}\times\mathcal{N}(\mathbf{0},\boldsymbol{M})$, so the composition preserves the constrained prior and is *ergodic*. The flow handles smooth targets efficiently and the refresh supplies the missing ergodicity.

Why a reflective flow alone is not enough. An exact reflective flow inside a region with a smooth boundary is a *billiard*, and billiards are not generically ergodic. For a Gaussian likelihood, the constrained region is an *ellipsoid*, and the billiard in an ellipsoid is completely integrable: each trajectory is tangent to a fixed confocal quadric and confined to the *bounded shell*. No trajectory length makes such a flow forget its starting point, because the conserved quantities do not depend on time, and strict convexity is enough for the qualitative conclusion, since caustics persist near any smooth strictly convex boundary. In nested sampling, the replacement then stays *correlated* with its seed, the live set behaves as if it had fewer than N independent members, and the schedule of §3 fails. Preconditioning does not help: whitening an ellipsoid produces a ball, whose billiard is the most degenerate case of all.

Persistent reflected flow. Each live-point carries a *momentum*. When it seeds a replacement, that momentum is partially refreshed by a *Horowitz rotation*,

$$\boldsymbol{p} \;\leftarrow\; \cos(\delta)\,\boldsymbol{p} \;+\; \sin(\delta)\,\boldsymbol{\eta}, \qquad \boldsymbol{\eta}\sim\mathcal{N}(\mathbf{0},\boldsymbol{M}), \quad \delta\in(0,\tfrac{\pi}{2}], \tag{5}$$

which preserves the momentum marginal exactly for every δ, unlike the multiplicative perturbation of GGNS. Taking $\delta<\pi/2$ retains directed motion across successive replacements instead of resampling the direction each time; $\delta=\pi/2$ recovers *independent* momenta. The particle then moves for a *fixed* number n_f of integration steps, with velocity $\boldsymbol{v}=\boldsymbol{M}^{-1}\boldsymbol{p}=\widehat{\Sigma}\boldsymbol{p}$. Whenever a step would leave the constrained region, the crossing is located and the momentum reflected there by Eq. 3 with $\boldsymbol{M}^{-1}=\widehat{\Sigma}$, the surface normal $\boldsymbol{n}$ being the score $\nabla\log\mathcal{L}$ on the likelihood contour and the face normal $\boldsymbol{e}_i$ on the prior box. Contour crossings are bracketed and refined by n_b bisections with the lower end held on the valid side; a single Newton step, as in GGNS, can land outside a curved contour and leak mass across it. The prior box is part of the constrained region, so its walls are reflected with the same $\boldsymbol{M}$-metric rule rather than by flipping a coordinate, which is exact only for a diagonal metric. The length n_f is fixed rather than adaptive: a stopping rule that depends on the realized trajectory

destroys the reversibility of the flow unless it is made symmetric at considerable extra cost (Hoffman & Gelman, 2014), and the resulting bias is not washed out because the momentum persists across replacements. A fixed length keeps the flow *reversible* by construction; only Δt is adapted, by dual averaging towards a target number of reflections per flow (Appendix C).

Preconditioned slice refresh. The endpoint of the flow then undergoes n_s constrained *slice updates*. Each update draws either a *whitened* direction $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\widehat{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\top$, with probability $(1-\rho)$, or a single coordinate axis scaled by that coordinate's live spread. The two branches serve *complementary* purposes: the whitened branch handles global anisotropy and correlation, while the axis branch, combined with the doubling step-out below, finds a scale for each coordinate and can therefore handle geometry whose scale varies with position. Along the chosen direction, we place a bracket through the current point, double its width until both ends leave the constrained region, then sample within it and shrink towards the current point until a proposal satisfies $\mathcal{L}(\boldsymbol{\theta}') > \mathcal{L}^*$. Doubling with shrinkage leaves the uniform measure on $\{\mathcal{L} > \mathcal{L}^*\}$ invariant (Neal, 2003), regardless of the metric estimate. The update supplies the *ergodicity* absent from the flow, is one-dimensional, makes no assumption about the shape of the region, and allows the bracket to extend through the boundary, so the move is not confined to the component in which it started.

Setting $\boldsymbol{M} = \boldsymbol{I}$, $\delta = \pi/2$, $n_s = 0$ and a one Newton-step reflection reduces the kernel to GGNS. Algorithm 1 and Appendix C give the full procedure and the per-problem settings.

4.3 Batched Removal and the Volume Schedule

Sequential nested sampling regenerates one point at a time, producing a short, branchy computation for a single point. Vectorizing the outer loop *amortizes* that overhead across many points but must retain the quadrature. Each generation of our loop removes the $k = \lceil fN \rceil$ lowest-likelihood live points at once, sets the new contour at the highest of them, and regenerates all k under one `vmap` above that common contour, with the batch fraction f trading statistical resolution for throughput.

Proposition 1 (Batched removal preserves the schedule)**.** *Let $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ be independent draws from the prior restricted to $\{\mathcal{L} > \lambda_0\}$, enclosing prior mass X_0. Remove the k lowest-likelihood points in increasing order of likelihood and let X_j be the mass enclosed by the contour through the j-th removed point. Then $X_j/X_{j-1} \sim \mathrm{Beta}(N-j+1, 1)$ independently across j, so $\mathbb{E}[\log(X_j/X_{j-1})] = -1/(N-j+1)$. After all k removals the $(N-k)$ survivors are independent draws from the prior restricted to $\{\mathcal{L} > \lambda_k\}$, with λ_k the likelihood of the k-th removed point.*

The proof in Appendix A maps each point to its *enclosed* prior mass, which is uniform by the probability integral transform, and proceeds recursively. Within a batch, the schedule is identical to that of k sequential removals: the live count decreases as $N, (N-1), \ldots, (N-k+1)$. Contracting X with N throughout would understate shrinkage by a factor that grows with f and bias $\log \widehat{\mathcal{Z}}$ upward. The survivors remain independent draws from the constrained prior, so they can seed the next generation without bias.

4.4 Differentiating the Evidence

Let ϕ be a *parameter* of the model, such as a likelihood or noise parameter. Differentiating Eq. 1 under the integral sign gives the score identity $\partial_\phi \log \mathcal{Z} = \mathbb{E}_{\mathrm{post}}[\partial_\phi \log \mathcal{L}]$, the expectation taken under the posterior. The evidence is a one-dimensional quadrature over prior volume, so it is *differentiable* in ϕ whenever the likelihood is, and automatic differentiation can be applied to the quadrature itself.

Proposition 2. *Let $\{\boldsymbol{\theta}_k\}$ be the dead points of a run performed at ϕ_0 and $\{w_k\}$ their prior-volume weights from Eq. 2. Define $\log \widehat{\mathcal{Z}}(\phi) = \mathrm{logsumexp}_k(\log w_k + \log \mathcal{L}(\boldsymbol{\theta}_k; \phi))$. Then $\log \widehat{\mathcal{Z}}(\phi_0)$ is the run's evidence estimate and*

$$\nabla_\phi \log \widehat{\mathcal{Z}}(\phi)\big|_{\phi_0} = \sum_k p_k \, \partial_\phi \log \mathcal{L}(\boldsymbol{\theta}_k; \phi_0), \qquad p_k = \frac{w_k \mathcal{L}(\boldsymbol{\theta}_k; \phi_0)}{\sum_j w_j \mathcal{L}(\boldsymbol{\theta}_j; \phi_0)}, \tag{6}$$

which is the run's posterior-weighted estimate of $\mathbb{E}_{\mathrm{post}}[\partial_\phi \log \mathcal{L}]$.

In practical terms, a single run at ϕ_0 thus returns both the evidence and its gradient; the latter being the Monte Carlo estimator of the score identity built from the run's own posterior representation. Its

accuracy is that of any posterior expectation from the same run, at the cost of one *extra pass* over the *dead points*. Appendix B proves the proposition and considers $\phi \neq \phi_0$, where $\log \widehat{\mathcal{Z}}(\phi)$ becomes an importance-reweighting estimate that degrades as the likelihood surfaces diverge. Appendix D measures its useful range and gives a diagnostic for when to rerun the sampler. Two restrictions are worth stating here: *(i)* the weights w_k are held *fixed*, so ϕ must enter through the likelihood; a prior hyperparameter must first be moved into the likelihood by reparameterization; and *(ii)* this is differentiation of the estimator at fixed sampler decisions, not differentiation through the sampler itself.

5 Experiments

5.1 Setup

We compare GDHDNS against three other *state-of-the-art* evidence estimators: `dynesty` (Speagle, 2020), `PolyChord` (Handley et al., 2015), and an adaptive tempered sequential Monte Carlo sampler built on `blackjax` (Del Moral et al., 2006; Cabezas et al., 2024). Every method receives the same likelihood and uniform prior, and the same live-point budget where applicable. SMC uses $\max(N, 512)$ particles, target ESS of 0.5, and ten HMC moves of eight leapfrog steps at each tempering stage. `PolyChord` uses its default $5d$ repeats, except on SDEMEM, where the per-point Kalman filter makes this impractical and we use d. We count likelihood evaluations by instrumentation for the nested samplers and analytically for SMC. Results are means and across-seed standard deviations over five seeds.

Likewise, reference values are exact where possible: closed form for the Gaussian, one-dimensional quadrature for the funnel, analytic linear Gaussian evidence for lensing, and a four million sample Monte Carlo estimate for the banana. The Lennard-Jones targets have no closed form, so we use an independent *parallel-tempering* run (Earl & Deem, 2005) with 18 temperatures and 96 chains. SDEMEM has no reference evidence and is used only to assess *posterior sampling*. GDHDNS uses the same kernel throughout: $n_f = 16$ integration steps, $n_b = 8$ contour-crossing bisections, and Horowitz angle $\delta = 1.2$ ($\cos\delta \approx 0.36$). The live count N, number of refresh slice updates n_s, axis probability ρ, and batch fraction f vary by problem; Appendix C gives the settings. The sweeps use $N = 300$ at every dimension and n_s grows linearly with d. Timings were measured on an Apple M4 Pro running `JAX` (v. 0.10).

5.2 Synthetic Targets and Dimensionality Scaling

We use three targets with tractable references, each a likelihood on a uniform prior. The **isotropic Gaussian** $\mathcal{L}(\boldsymbol{\theta}) = \prod_i \mathcal{N}(\boldsymbol{\theta}_i; 0, 1)$ is the standard calibration test. The **banana** (Haario et al., 1999) applies the shear $u_1 = \boldsymbol{\theta}_1 + b(\boldsymbol{\theta}_0^2 - \sigma_0^2)$ with $b = 0.1$, $\sigma_0 = 10$ to a diagonal Gaussian, producing a strongly curved ridge. **Neal's funnel** (Neal, 2003), with $v \sim \mathcal{N}(0, 3^2)$ and $x_i \mid v \sim \mathcal{N}(0, e^v)$, spans orders of magnitude of scale inside one target and is a known challenge for samplers (Betancourt & Girolami, 2015). The Gaussian and banana are run at $d = 2, 4, 8, 16, 32, 64, 128$, the funnel at $d = 2, 4, 8, 16, 32$.

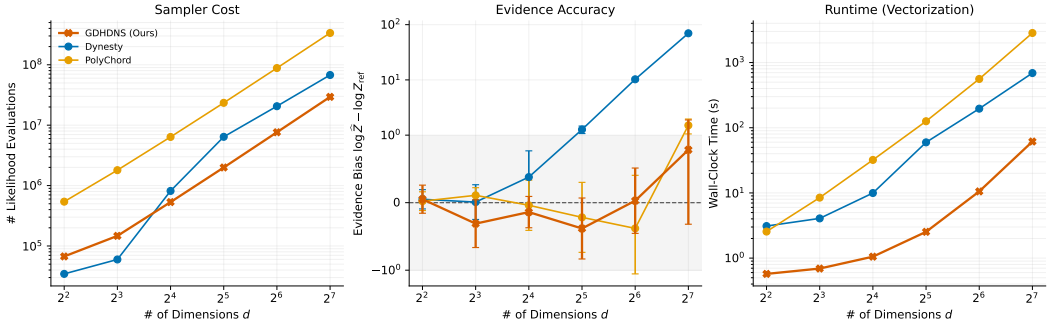
All methods are accurate at low dimension (Table 1), but their costs differ. GDHDNS is 6–100$\times$ faster than `dynesty` and `PolyChord`. Tempered SMC is comparably fast, but loses 1.3 nats[2] on the 8-D Gaussian and 2.1 on the 10-D funnel. This is a familiar tempering failure mode (Del Moral et al., 2006): intermediate distributions do not cover the target support, preventing transitions between modes.

The differences become clearer as dimension grows (Figure 1). On the 64-D Gaussian, `dynesty` is high by 10.3 nats, rising to 69.8 at 128. `PolyChord` remains accurate but, from $d = 8$, requires 11–12$\times$ as many likelihood evaluations and, above $d = 32$, 46–53$\times$ as much wall-clock time. The funnel reverses this ordering: `PolyChord`'s bias reaches $+3.3$ nats at 16 dimensions and $+8.9$ at 32, while a run takes more than 25 hours, compared with 9 seconds for GDHDNS. GDHDNS is the only method whose bias remains within its *across-seed* spread at every point in both sweeps: $+0.03 \pm 0.48$ on the 64-D Gaussian and $+0.03 \pm 0.40$ on the 32-D funnel. Its cost also grows more

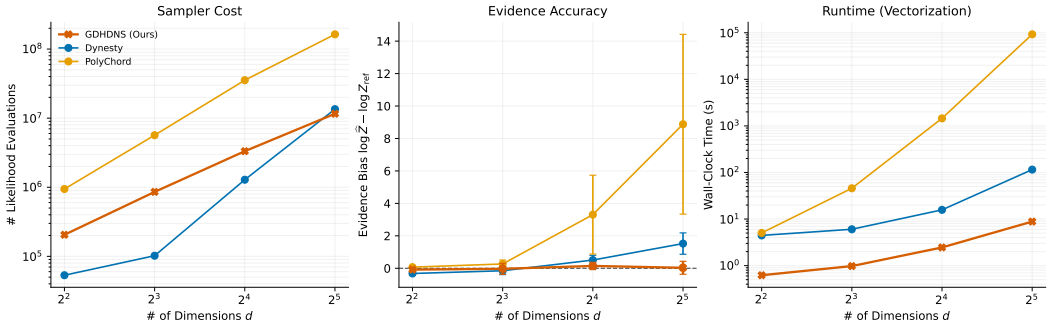
[2] A *nat* is the unit of $\log \mathcal{Z}$ under the natural logarithm, so an error of one nat is a factor of $e \approx 2.7$ in the evidence itself.

Table 1: Evidence and cost on the three synthetic targets with tractable references, five seeds, mean $\pm$ across-seed standard deviation. Scaling sweeps are in Figure 1. Best bias and best time in **bold**.

Target	**Method**	**Bias** $(\log \widehat{\mathcal{Z}} - \log \mathcal{Z}_{\text{ref}})$	n_{like}	**Time (s)**
Banana (2-D)	GDHDNS (ours)	$\mathbf{-0.008 \pm 0.042}$	5.3×10^4	**0.45**
	`dynesty`	-0.029 ± 0.059	5.3×10^4	6.6
	`PolyChord`	$+0.162 \pm 0.165$	2.4×10^5	6.1
	Tempered SMC	-0.037 ± 0.144	1.3×10^6	0.60
Gaussian (8-D)	GDHDNS (ours)	$+0.026 \pm 0.381$	2.0×10^5	**0.65**
	`dynesty`	-0.031 ± 0.220	$\mathbf{6.4 \times 10^4}$	4.3
	`PolyChord`	$+0.053 \pm 0.142$	2.4×10^6	11.5
	Tempered SMC	-1.288 ± 1.571	1.2×10^6	0.64
Funnel (10-D)	GDHDNS (ours)	$+0.029 \pm 0.337$	2.6×10^6	2.3
	`dynesty`	-0.043 ± 0.370	$\mathbf{1.1 \times 10^6}$	14.8
	`PolyChord`	$+1.851 \pm 3.424$	1.8×10^7	234.0
	Tempered SMC	-2.147 ± 1.275	2.9×10^6	**0.89**



(a) Isotropic Gaussian, $d = 4$ to 128.



(b) Neal's funnel, $d = 4$ to 32.

Figure 1: Dimension scaling over the synthetic targets: likelihood evaluations (*left*), evidence bias (*center*) and wall-clock time (*right*). `dynesty`'s bias diverges on the Gaussian above 32 dimensions and `PolyChord`'s on the funnel above 16; GDHDNS remains accurate at a wall-clock cost an order of magnitude below either baseline. Appendix F shows the same biases at selected dimensions with per-panel scales.

slowly than `dynesty`'s: it uses more evaluations at $d \leq 8$, fewer above $d \approx 16$, and $3.2\times$ fewer by $d = 32$ on the Gaussian.

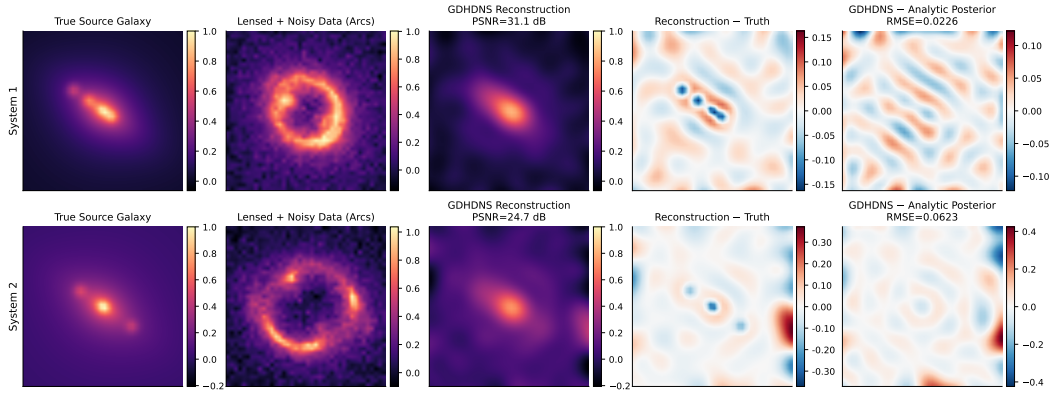


Figure 2: Strong lensing source reconstruction at 1600 dimensions for two systems (*rows*). Columns: true source, lensed and noisy data, GDHDNS reconstruction with PSNR, its residual against the truth, and its difference from the analytic posterior mean with RMSE. The second system's broken arcs leave part of the source weakly constrained, which is where its residual concentrates.

5.3 Strong Gravitational Lensing Galaxy Image Reconstruction

To further test the method on a high-dimensional structured target, we use a strong gravitational *lensing* galaxy reconstruction problem (Adam et al., 2022). A source image $\boldsymbol{s} \in \mathbb{R}^{n\times n}$, an inclined exponential disk with a Sérsic bulge and star-forming clumps (Sérsic, 1963), is mapped to the image plane by a singular *isothermal* ellipsoid deflection operator $\boldsymbol{A}$ (Kormann et al., 1994) and corrupted by Gaussian noise, giving $\boldsymbol{d} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I})$, $\sigma_n^2 = 0.01$, under a *correlated* Gaussian source prior $\boldsymbol{s} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$ with $\boldsymbol{C}$ a squared exponential kernel of length scale $\ell = 0.1$ and amplitude $\sigma_s^2 = 0.5$. The forward model is linear and Gaussian, so the posterior mean, posterior covariance and evidence are all analytic, which gives an exact reference in arbitrary dimension. The likelihood is $\mathcal{L}(\boldsymbol{s}) \propto \exp[-\frac{1}{2}(\boldsymbol{d} - \boldsymbol{A}\boldsymbol{s})^\top (\boldsymbol{d} - \boldsymbol{A}\boldsymbol{s})/\sigma_n^2]$. Taking $n = 40$ gives 1600 parameters.

At that dimension, GDHDNS recovers the source at 31.1 dB PSNR against the *ground-truth* and differs from the analytic posterior mean by an RMSE of 0.023 on a source scaled to $[0, 1]$ (Figure 2, top). A second system with a larger, more elliptical, lens produces a broken-arc configuration in which parts of the source are constrained by far fewer rays than others; the reconstruction falls to 24.7 dB and RMSE 0.062, with the residual concentrated in those regions. Evidence estimation is less successful: decorrelating a 1600-dimensional point requires $\mathcal{O}(d)$ slice updates, while our budget provides fewer, and $\log \widehat{\mathcal{Z}}$ is high by 7.8 ± 4.8 nats. In any case, *reconstruction quality* is less sensitive to this error because it depends on locating the posterior correctly, rather than on an exact volume schedule.

5.4 Lennard-Jones Interatomic Potentials

Rugged multimodal landscapes are complex for samplers that follow local geometry and offer little useful information to the ensemble metric. For n atoms at positions $\boldsymbol{x} \in \mathbb{R}^{3n}$, the *Lennard-Jones potential* is

$$E(\boldsymbol{x}) = 4\epsilon \sum_{i<j} \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right], \tag{7}$$

with r_{ij} the interatomic distance. We sample $\mathcal{L}(\boldsymbol{x}) \propto e^{-\beta E(\boldsymbol{x})}$ at $\beta = 3$, a warm solid whose mass sits near the global minimum but which retains many competing basins, and use LJ7 (21-D) and LJ13 (39-D), whose minima are known (Wales & Doye, 1997).

On LJ7, the posterior energy histograms agree with parallel tempering across the distribution, and GDHDNS reaches -16.05 against the global minimum -16.51 (Figure 6, Appendix F). LJ13 is harder: the bulk is recovered, but the low-energy tail is underrepresented and the lowest energy reached is -40.52, compared with -44.33. As anticipated in §4.1, when the live set spans several basins, its covariance describes none of them. The LJ13 minimum is separated from competing

basins by barriers that neither a whitened slice direction nor a coordinate move crosses often enough. This is the setting in which Handley et al. (2015) and Lemos et al. (2024) use clustering. Even so, the evidence remains accurate, and the sampler locates the global minimum and samples the bulk of the distribution well enough to give a reasonable estimate of $\log \widehat{\mathcal{Z}}$.

5.5 A Hierarchical Stochastic Differential Equation Mixed-Effects Model

Lastly, we test the method on a hierarchical **stochastic differential equation mixed-effects model** (SDEMEM), focusing on *posterior sampling* rather than evidence estimation. An SDEMEM describes repeated noisy measurements of several individuals whose *latent states* follow an SDE with individual-specific parameters (Picchini et al., 2010). Such models arise when a population is observed and each member follows its own trajectory, as in pharmacokinetics (Donnet & Samson, 2013), tumor growth (Picchini & Forman, 2019), epidemic dynamics (Narci et al., 2022), and repeated physiological measurement (Oravecz et al., 2009). They are demanding because the population parameters and per-individual random effects must be inferred jointly, coupling the two levels of the posterior and increasing its dimension with the study. We use a bivariate *Ornstein-Uhlenbeck* process adapted from Barragán-Ramírez et al. (2026) with a Gaussian observation model for each individual's latent state.

For individual m the latent state $\boldsymbol{X}_t^{(m)} \in \mathbb{R}^2$ follows the SDE:

$$\begin{aligned} d\boldsymbol{X}_t^{(m)} &= \left(\boldsymbol{\beta} \odot \boldsymbol{b}^{(m)}\right)\left(\boldsymbol{\alpha} - \boldsymbol{X}_t^{(m)}\right) dt + \boldsymbol{\sigma}\, d\boldsymbol{W}_t^{(m)}, \qquad \boldsymbol{b}^{(m)} = \exp\left(\omega \boldsymbol{Z}^{(m)}\right), \\ \boldsymbol{Y}_t^{(m)} &= \boldsymbol{X}_t^{(m)} + \boldsymbol{\varepsilon}_t^{(m)}, \qquad \boldsymbol{\varepsilon}_t^{(m)} \sim \mathcal{N}\left(\mathbf{0}, \operatorname{diag}(\boldsymbol{\xi}^2)\right), \qquad Z_{ij}^{(m)} \sim \mathcal{N}(0,1), \end{aligned} \tag{8}$$

with $\odot$ the elementwise product, $\boldsymbol{\beta} \in \mathbb{R}^{2\times 2}$ a population drift matrix coupling the two processes, and *random effects* $\boldsymbol{b}^{(m)} \in \mathbb{R}^{2\times 2}$ entering multiplicatively. We simulate $M = 5$ individuals at $T = 20$ times with $\Delta t = 0.1$:

$$\boldsymbol{\alpha} = (1.0, -0.5)^\top, \quad \boldsymbol{\beta} = \left(\begin{smallmatrix} 0.8 & 0.2 \\ 0.15 & 0.5 \end{smallmatrix}\right), \quad \boldsymbol{\sigma} = (0.5, 0.3)^\top, \quad \boldsymbol{\xi} = (0.15, 0.10)^\top, \quad \omega = 0.3, \tag{9}$$

The posterior has 31 dimensions: the eleven *population* parameters in Eq. 9 and four *random effects* per individual. The marginal likelihood is tractable exactly. The OU transition is the matrix exponential $\boldsymbol{\Phi}^{(m)} = \exp(-\boldsymbol{A}^{(m)}\Delta t)$ with $\boldsymbol{A}^{(m)} = \boldsymbol{\beta} \odot \boldsymbol{b}^{(m)}$; van Loan's method (Van Loan, 1978) gives the process-noise covariance, and a Kalman filter marginalizes the latent states. We reject points for which $\boldsymbol{A}^{(m)}$ is not *contractive*, namely when the Routh-Hurwitz conditions $\operatorname{tr}(\boldsymbol{A}^{(m)}) > 0$ and $\det(\boldsymbol{A}^{(m)}) > 0$ fail and no *stationary distribution* exists. The posterior concentrates near the truth for every population parameter, and GDHDNS has the highest effective samples per second while converging ($\widehat{R} = 1.03$). Priors, traces, and diagnostics are in Appendix E.

6 Conclusion

Constrained sampling is the main cost and a principal source of error in nested sampling. GDHDNS uses the live-point covariance as a mass matrix that follows the contracting constrained region, avoiding a fixed metric and its associated step-size compromises. It also addresses a limitation of gradient-guided nested sampling: reflective billiards are integrable rather than ergodic on convex level sets, and therefore cannot alone decorrelate a live-point. A preconditioned slice refresh breaks this dependence while retaining the flow. With a batched outer loop whose volume schedule matches sequential removal, the method delivers unbiased evidence throughout both dimension sweeps and returns $\partial \log \mathcal{Z}/\partial\phi$ alongside $\log \mathcal{Z}$ from one run. Its main limitation is the use of a single global covariance. On LJ13 it misses part of the low-energy tail, and at 1600 dimensions the $\mathcal{O}(d)$ slice updates needed to decorrelate a point exceed our budget. Promising extensions include clustering the live set for mode-aware preconditioning and differentiating through the sampler dynamics to learn schedules and tune hyperparameters.

AI Use Statement

In this work, we used generative AI tools to assist with software implementation and with the preparation of this manuscript. On the implementation side, LLM-based coding assistants were used to

draft and refactor parts of the reference implementation, specifically the `JAX` kernels, the experiment harness, the caching layer and the plotting code. On the writing side, they were used to draft and edit passages of the text, to condense the manuscript so that it meets the page limit, and for routine editorial work: improving readability, tightening captions and tables, or checking LaTeX formatting.

We did not use generative AI tools to develop the theoretical content of this paper. The method, the ergodicity argument of §4.2, the statements of Propositions 1 and 2 and their proofs, the design of the experimental protocol and the interpretation of the results are the authors' own work. Generative AI was not used to generate synthetic data sets, to propose or refine hypotheses, or to perform data analysis. The remaining categories in the disclosure list, namely translation, transcription of research material and the design of surveys or interviews, are not applicable to this work.

We have reviewed all AI-assisted work. AI-assisted code was read, tested and verified by the authors, and validated against the analytic references described in §5.1; every number reported in this paper is the output of a run of that verified code. AI-assisted text was checked line by line against the underlying results for factual accuracy, and all citations were verified against the primary sources. We take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

Ethics Statement

This work is *methodological*: it develops a general-purpose algorithm for Bayesian evidence estimation and posterior sampling. It involves no human subjects, personally identifiable information, or new data collection. Every experiment reported here uses either analytically defined test distributions (the Gaussian, the banana, Neal's funnel, and the linear-Gaussian lensing model), a standard physical potential (Lennard-Jones clusters; Wales & Doye, 1997), or data simulated from the stochastic differential equation model specified in §5.5. Accordingly, we identify no direct risk of harm, discrimination, or privacy violation arising from the method or its evaluation. We have read and adhere to the ICLR Code of Ethics.

Reproducibility Statement

The complete implementation of the proposed method is available at the repository linked in §1. Each experiment is defined by a version-controlled configuration file, and runs are content-addressed by a hash of that configuration, so a cached result is reused only when the configuration that produced it is identical; baseline library versions are pinned for the same reason. All reported quantities are means over five fixed seeds with the across-seed standard deviation, and the hardware and software versions behind every timing are stated in §5.1.

The method itself is specified in full in §4, with the complete procedure given as Algorithm 1 and every per-problem hyperparameter in Table 2, alongside the implementation details, namely the step-size adaptation, the crossing refinement and the collision cap, that are needed to reproduce the kernel exactly. Propositions 1 and 2 are stated with their assumptions in the main text and proved in Appendices A and B. The reference values against which we measure bias are exact wherever available, and their derivation is given with each target in §5. The targets are specified completely enough to be re-implemented independently, including the SDEMEM priors, true parameter values and simulation protocol in Appendix E.

References

Alexandre Adam, Adam Coogan, Nikolay Malkin, Ronan Legin, Laurence Perreault-Levasseur, Yashar Hezaveh, and Yoshua Bengio. Posterior samples of source galaxies in strong gravitational lenses with score-based priors. In *Machine Learning and the Physical Sciences Workshop, NeurIPS*, 2022.

Joshua G. Albert. JAXNS: A high-performance nested sampling package based on JAX. *arXiv preprint arXiv:2012.15286*, 2020.

Greg Ashton, Noam Bernstein, Johannes Buchner, Xi Chen, Gábor Csányi, Andrew Fowlie, Farhan Feroz, Matthew Griffiths, Will Handley, Michael Habeck, et al. Nested sampling for physical scientists. *Nature Reviews Methods Primers*, 2:39, 2022.

G. Barragán-Ramírez, S. Infante, I. Becerra, and A. Hernández. A comparative study of Monte Carlo methods and numerical schemes for SDE mixed-effects models. *Statistics, Optimization & Information Computing*, 2026.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

Michael Betancourt. Nested sampling with constrained Hamiltonian Monte Carlo. In *AIP Conference Proceedings*, volume 1305, pp. 165–172, 2011.

Michael Betancourt. A general metric for Riemannian manifold Hamiltonian Monte Carlo. In *Geometric Science of Information*, volume 8085 of *Lecture Notes in Computer Science*, pp. 327–334, 2013.

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. In *Current Trends in Bayesian Methodology with Applications*, pp. 79–101. Chapman and Hall/CRC, 2015.

Benjamin Bloem-Reddy and John P. Cunningham. Slice sampling on hamiltonian trajectories. In *International Conference on Machine Learning (ICML)*, pp. 3050–3058, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs. 2018. `http://github.com/google/jax`.

Johannes Buchner. UltraNest – a robust, general purpose Bayesian inference engine. *Journal of Open Source Software*, 6(60):3001, 2021.

Johannes Buchner. Nested sampling methods. *Statistics Surveys*, 17:169–215, 2023.

Alberto Cabezas, Adrien Corenflos, Junpeng Lao, Rémi Louf, et al. BlackJAX: Composable Bayesian inference in JAX. *arXiv preprint arXiv:2402.10797*, 2024.

Nicolas Chopin and Christian P. Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

Sophie Donnet and Adeline Samson. A review on estimation of stochastic differential equations for pharmacokinetic/pharmacodynamic models. *Advanced Drug Delivery Reviews*, 65(7):929–939, 2013.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

David J. Earl and Michael W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

Farhan Feroz and John Skilling. Exploring multi-modal distributions with nested sampling. In *AIP Conference Proceedings*, volume 1553, pp. 106–113, 2013.

Farhan Feroz, Michael P. Hobson, and M. Bridges. MultiNest: An efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*, 398(4):1601–1614, 2009.

Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 73(2):123–214, 2011.

Heikki Haario, Eero Saksman, and Johanna Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14(3):375–395, 1999.

Will J. Handley, Michael P. Hobson, and Anthony N. Lasenby. PolyChord: Next-generation nested sampling. *Monthly Notices of the Royal Astronomical Society*, 453(4):4384–4398, 2015.

Edward Higson, Will Handley, Michael Hobson, and Anthony Lasenby. Sampling errors in nested sampling parameter estimation. *Bayesian Analysis*, 13(3):873–896, 2018.

Edward Higson, Will Handley, Mike Hobson, and Anthony Lasenby. Dynamic nested sampling: An improved algorithm for parameter estimation and evidence calculation. *Statistics and Computing*, 29:891–913, 2019.

Matthew Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

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

Alan M. Horowitz. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2):247–252, 1991.

R. Kormann, P. Schneider, and M. Bartelmann. Isothermal elliptical gravitational lens models. *Astronomy and Astrophysics*, 284:285–299, 1994.

Pablo Lemos, Nikolay Malkin, Will Handley, Yoshua Bengio, Yashar Hezaveh, and Laurence Perreault-Levasseur. Improving gradient-guided nested sampling for posterior inference. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024.

Romain Narci, Maud Delattre, Catherine Larédo, and Elisabeta Vergu. Inference in gaussian state-space models with mixed effects for multiple epidemic dynamics. *Journal of Mathematical Biology*, 85(4):40, 2022.

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

Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pp. 113–162. Chapman and Hall/CRC, 2011.

Zita Oravecz, Francis Tuerlinckx, and Joachim Vandekerckhove. A hierarchical Ornstein–Uhlenbeck model for continuous repeated measurement data. *Psychometrika*, 74(3):395–418, 2009.

Umberto Picchini and Julie Lyng Forman. Bayesian inference for stochastic differential equation mixed effects models of a tumour xenography study. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(4):887–913, 2019.

Umberto Picchini, Andrea De Gaetano, and Susanne Ditlevsen. Stochastic differential mixed-effects models. *Scandinavian Journal of Statistics*, 37(1):67–90, 2010.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

Robert Salomone, Leah F. South, Christopher Drovandi, Dirk P. Kroese, and Adam M. Johansen. Unbiased and consistent nested sampling via sequential Monte Carlo. *Journal of the Royal Statistical Society: Series B*, 87(4):1221–1238, 2025.

José Luis Sérsic. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletín de la Asociación Argentina de Astronomía*, 6:41–43, 1963.

John Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, 2006.

Joshua S. Speagle. dynesty: A dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493(3):3132–3158, 2020.

Charles Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.

David J. Wales and Jonathan P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

Michael J. Williams, John Veitch, and Chris Messenger. Nested sampling with normalizing flows for gravitational-wave inference. *Physical Review D*, 103(10):103006, 2021.

A Proof of Proposition 1

Let X_0 be the prior mass enclosed by the current contour, and, for a point $\boldsymbol{\theta}$, define $u(\boldsymbol{\theta}) = X(\mathcal{L}(\boldsymbol{\theta}))/X_0$ for the fraction of that mass enclosed by its contour.

Proof. If $\boldsymbol{\theta}$ is drawn from the prior restricted to $\{\mathcal{L} > \lambda_0\}$, then $u(\boldsymbol{\theta})$ is uniform on $[0, 1]$. This is the probability integral transform for the monotone map $\lambda \mapsto X(\lambda)$, which makes the nested-sampling shrinkage law independent of the target. For $t \in [0, 1]$,

$$\Pr[u(\boldsymbol{\theta}) \le t] = \frac{1}{X_0} \int_{\mathcal{L}(\boldsymbol{\theta}') > X^{-1}(tX_0)} \pi(\boldsymbol{\theta}')\, d\boldsymbol{\theta}' = \frac{X(X^{-1}(tX_0))}{X_0} = t. \tag{10}$$

For N independent live points, $u_1, \ldots, u_N$ are therefore *i.i.d.* $\mathcal{U}[0, 1]$, and the lowest-likelihood point is the one with the largest u, since X is decreasing in λ.

The largest of N *i.i.d.* uniforms has c.d.f. $F(t) = t^N$, hence has distribution $\mathrm{Beta}(N, 1)$. Thus $X_1/X_0 = u_{(N)} \sim \mathrm{Beta}(N, 1)$ and $\mathbb{E}[\log(X_1/X_0)] = -1/N$. Conditional on $u_{(N)} = t$, the remaining $(N - 1)$ values are *i.i.d.* uniform on $[0, t]$. After rescaling by t, they are independent draws from the prior restricted to the new contour. Repeating the argument with $(N - 1)$ points and mass X_1, and noting that the conditional law of the rescaled survivors does not depend on $u_{(N)}$, gives independence across j. By induction, the j-th removal contributes an independent factor $X_j/X_{j-1} \sim \mathrm{Beta}(N - j + 1, 1)$ with $\mathbb{E}[\log(X_j/X_{j-1})] = -1/(N - j + 1)$. After k removals, the $(N - k)$ survivors are *i.i.d.* draws from the prior restricted to $\{\mathcal{L} > \lambda_k\}$. Replacing the removed points with fresh draws above λ_k restores N *i.i.d.* draws from that constrained prior, as required for the next generation. ■

The argument does not depend on *when* replacements are generated, only on the order in which removals are counted. The batched loop can therefore *regenerate* at the end of a batch while accounting for removals in increasing likelihood order and decreasing the live count within the batch. Holding the count at N would replace the correct increment $\sum_{j=1}^{k} 1/(N - j + 1)$ with k/N, understate the contraction of X by a factor that grows with $f = k/N$, and bias $\log \widehat{\mathcal{Z}}$ upward.

B Proof of Proposition 2

We first state the identity estimated by the quadrature. We may differentiate Eq. 1 under the integral sign when $\partial_\phi \mathcal{L}(\boldsymbol{\theta}; \phi)$ is dominated, in a neighbourhood of ϕ_0, by an integrable function of $\boldsymbol{\theta}$:

$$\begin{aligned} \partial_\phi \log \mathcal{Z}(\phi) &= \frac{1}{\mathcal{Z}(\phi)} \int \partial_\phi \mathcal{L}(\boldsymbol{\theta}; \phi)\, \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &= \int \partial_\phi \log \mathcal{L}(\boldsymbol{\theta}; \phi)\, \frac{\mathcal{L}(\boldsymbol{\theta}; \phi)\pi(\boldsymbol{\theta})}{\mathcal{Z}(\phi)}\, d\boldsymbol{\theta} = \mathbb{E}_{\mathrm{post}}[\partial_\phi \log \mathcal{L}]\,. \end{aligned} \tag{11}$$

The posterior on the right is evaluated at ϕ and is estimated by Eq. 6.

Proof of Proposition 2. Let $f_k(\phi) = \log w_k + \log \mathcal{L}(\boldsymbol{\theta}_k; \phi)$, so $\log \widehat{\mathcal{Z}}(\phi) = \mathrm{logsumexp}_k\, f_k(\phi)$. The completed run fixes the dead points $\boldsymbol{\theta}_k$ and prior-volume weights w_k; only $\mathcal{L}(\boldsymbol{\theta}_k; \phi)$ depends on ϕ. The gradient of $\mathrm{logsumexp}$ is the softmax-weighted average of the argument gradients:

$$\nabla_\phi \log \widehat{\mathcal{Z}}(\phi) = \sum_k \frac{e^{f_k(\phi)}}{\sum_j e^{f_j(\phi)}} \nabla_\phi f_k(\phi) = \sum_k p_k(\phi)\, \partial_\phi \log \mathcal{L}(\boldsymbol{\theta}_k; \phi), \tag{12}$$

where $p_k(\phi) = w_k \mathcal{L}(\boldsymbol{\theta}_k; \phi) / \sum_j w_j \mathcal{L}(\boldsymbol{\theta}_j; \phi)$. Evaluating at ϕ_0 gives Eq. 6. There, $p_k(\phi_0)$ is the normalized nested-sampling posterior weight $w_k \mathcal{L}_k / \widehat{\mathcal{Z}}$ from Eq. 2. The sum is therefore the run's Monte Carlo estimate of $\mathbb{E}_{\mathrm{post}}[\partial_\phi \log \mathcal{L}]$, which equals $\partial_\phi \log \mathcal{Z}$ by Eq. 11. Its accuracy is that of any posterior expectation from the same run. ■

Behaviour away from ϕ_0. For $\phi \neq \phi_0$, the expression remains an estimator, but of a different kind. Writing $\log \widehat{\mathcal{Z}}(\phi) = \log \widehat{\mathcal{Z}}(\phi_0) + \log \sum_k p_k(\phi_0)\, r_k(\phi)$ with $r_k(\phi) = \mathcal{L}(\boldsymbol{\theta}_k;\phi)/\mathcal{L}(\boldsymbol{\theta}_k;\phi_0)$ gives an *importance-sampling* estimate of $\mathcal{Z}(\phi)/\mathcal{Z}(\phi_0)$ using the dead points as proposals. Its quality depends on the spread of r_k, summarized by the *effective sample size* (ESS):

$$\mathrm{ESS}(\phi) = \frac{\left[\sum_k p_k(\phi_0) r_k(\phi)\right]^2}{\sum_k p_k(\phi_0)^2 r_k(\phi)^2}, \tag{13}$$

This quantity is available from the run and indicates whether a given ϕ remains within reweighting range. Appendix D measures that range for a problem with an analytic answer. Two restrictions remain. The weights w_k are constant, so ϕ must enter through $\mathcal{L}$; a prior hyperparameter must first be moved into the likelihood by reparameterization. Moreover, the dead points are fixed, so this differentiates the estimator at fixed sampler decisions rather than the sampler dynamics.

C Algorithm, Implementation and Hyperparameters

Algorithm 1 Generalized Differentiable Hamiltonian Dynamic Nested Sampling (GDHDNS). Block (c) is the constrained kernel of §4.2: a persistent reflected flow composed with a preconditioned slice refresh. Setting $\boldsymbol{M} \equiv \boldsymbol{I}$, $\delta = \pi/2$, $n_s = 0$ and a one Newton-step crossing recovers GGNS.

Require: likelihood $\mathcal{L}$, prior π with an inverse-CDF sampler, live count N, batch fraction $f \in (0,1)$, flow steps n_f, bisections n_b, Horowitz angle δ, slice steps n_s, axis probability ρ, tolerance.
Ensure: $\log \mathcal{Z}$, its uncertainty σ, and weighted posterior samples $\mathcal{D}$.

$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N \sim \pi$; $\mathcal{L}_i \leftarrow \mathcal{L}(\boldsymbol{\theta}_i)$; $X \leftarrow 1$; $\mathcal{Z} \leftarrow 0$; $\mathcal{D} \leftarrow \emptyset$
while not terminated **do** ▷ stop when the live set's remaining evidence falls under tolerance
  $\bar{\boldsymbol{\theta}} \leftarrow \frac{1}{N}\sum_i \boldsymbol{\theta}_i$; $\widehat{\Sigma} \leftarrow \frac{1}{N-1}\sum_i (\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}})^\top$; factor $\widehat{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\top$ ▷ (a) Eq. 4
  $k \leftarrow \lceil fN \rceil$; relabel so $\mathcal{L}_{(1)} \le \cdots \le \mathcal{L}_{(N)}$; $\mathcal{L}^* \leftarrow \mathcal{L}_{(k)}$
  for $j = 1, \ldots, k$ **do** ▷ (b) bank the batch under the decreasing live count
    $n \leftarrow N - j + 1$; $w \leftarrow X(1 - e^{-1/n})$ ▷ Proposition 1
    $\mathcal{Z} \leftarrow \mathcal{Z} + w\,\mathcal{L}_{(j)}$; $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{\theta}_{(j)}, w\mathcal{L}_{(j)})\}$; $X \leftarrow X\, e^{-1/n}$
  end for
  for each of the k vacated slots, in parallel **do** ▷ (c) refill above $\mathcal{L}^*$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, a uniformly chosen surviving live-point; $\boldsymbol{p} \leftarrow$ the carried momentum
    $\boldsymbol{p} \leftarrow \cos(\delta)\,\boldsymbol{p} + \sin(\delta)\,\boldsymbol{L}^{-\top}\boldsymbol{z}$, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$; $\boldsymbol{p}_0 \leftarrow \boldsymbol{p}$ ▷ Horowitz refresh, Eq. 5
    for n_f steps of duration Δt **do** ▷ persistent reflected flow
      move $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta t\, \widehat{\Sigma}\boldsymbol{p}$, stopping at the first likelihood-contour or box crossing
      at a crossing: refine it by n_b bisections, reflect $\boldsymbol{p}$ (Eq. 3), continue for the remainder
    end for
    if the flow reached the collision cap **then** ▷ discard an incomplete flow
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$; $\boldsymbol{p} \leftarrow -\boldsymbol{p}_0$ ▷ reversing $\boldsymbol{p}$ ensures *reversibility*
    end if
    for n_s slice updates **do** ▷ preconditioned slice refresh
      $\boldsymbol{u} \leftarrow \boldsymbol{L}\boldsymbol{z}$, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ with probability $(1-\rho)$, else $\boldsymbol{u} \leftarrow s_c \boldsymbol{e}_c$ for a random axis c
      bracket $[\boldsymbol{\theta} - r\boldsymbol{u},\ \boldsymbol{\theta} + (1-r)\boldsymbol{u}]$, $r \sim \mathcal{U}(0,1)$; double while both ends satisfy $\mathcal{L} > \mathcal{L}^*$
      propose $\boldsymbol{\theta}' \sim \mathcal{U}(\text{bracket})$, shrinking towards $\boldsymbol{\theta}$ until $\mathcal{L}(\boldsymbol{\theta}') > \mathcal{L}^*$; $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}'$
    end for
    install $\boldsymbol{\theta}$, $\mathcal{L}(\boldsymbol{\theta})$ and $\boldsymbol{p}$ into the slot
  end for
end while
fold in the survivors at weight X/N each; $\sigma \leftarrow \sqrt{\mathcal{D}_{\mathrm{KL}}(\mathcal{P}\|\Pi)/N}$
return $\log \mathcal{Z}$, σ, $\mathcal{D}$

Implementation notes. Momentum is stored with each live-point, so persistence spans replacements: a replacement inherits its seed's momentum and partially refreshes it through Eq. 5. We use $\delta = 1.2$ throughout ($\cos(\delta) \approx 0.36$). This strong, partial refresh breaks the ellipsoidal billiard invariant while retaining directed motion.

The flow has $n_f = 16$ steps. Nesterov dual averaging adapts its duration Δt to target a number of *reflections per flow*. Raw likelihood-call counts are unsuitable because they include root refinements

and vary with contour curvature. We pass the error through $\tanh$ so an unbounded count cannot destabilize a controller designed for a bounded statistic.

Contour crossings are bracketed and refined by $n_b = 8$ bisections, keeping the lower endpoint valid. Box crossings are solved exactly because the hit time for an axis-aligned face is available in closed form. Both use the same $\boldsymbol{M}$-metric reflection. A cap of 32 collisions per step bounds the inner loop on pathological contours. When a lane reaches the cap, we discard it, restore its seed, and reverse its momentum. This is an involution on $(\boldsymbol{\theta}, \boldsymbol{p})$, so the flow remains reversible. The cap therefore cannot install a variable-length trajectory or introduce a point not produced by the dynamics. Before factorization, we shrink $\widehat{\Sigma}$ towards the identity by $10^{-8}\mathrm{tr}(\widehat{\Sigma})/d$ for numerical stability.

Table 2: GDHDNS settings per problem. The flow is identical everywhere ($n_f = 16$ steps, $n_b = 8$ bisections, $\delta = 1.2$); only the live count N, the slice updates n_s in the refresh, the axis probability ρ and the batch fraction f vary.

Problem	d	N	n_s	ρ	f	**Termination**
Banana	2	400	4	0.5	0.25	$\Delta \log \mathcal{Z} < 10^{-2}$
Gaussian	8	400	6	0.5	0.25	$\Delta \log \mathcal{Z} < 10^{-2}$
Funnel	10	500	25	0.4	0.25	$\Delta \log \mathcal{Z} < 10^{-2}$
Lensing	1600	800	5	0.2	0.40	$\Delta \log \mathcal{Z} < 0.4$
LJ7	21	500	15	0.6	0.25	$\Delta \log \mathcal{Z} < 0.1$
LJ13	39	800	25	0.6	0.25	$\Delta \log \mathcal{Z} < 0.1$
SDEMEM	31	500	10	0.3	0.25	$\Delta \log \mathcal{Z} < 0.05$
Gaussian sweep	4 – 128	300	$\max(5, \lceil 0.75d \rceil)$	1.0	0.15	$\Delta \log \mathcal{Z} < 0.1$
Funnel sweep	4 – 32	300	$\max(10, \lceil 2.5d \rceil)$	0.5	0.20	$\Delta \log \mathcal{Z} < 0.1$

D Differentiable Evidence

Proposition 2 implies that a run returns the gradient of its own log-evidence at no additional sampling cost. We test this on a 64-dimensional linear Gaussian model, where $\log \mathcal{Z}(\sigma)$ is analytic in the observation noise level σ. We generate data at $\sigma_0 = 0.1$, perform six independent runs at σ_0, and do not rerun the sampler.

From runs that only saw σ_0, reweighting recovers the shape of $\log \mathcal{Z}(\sigma)$ over $\sigma \in [0.06, 0.20]$, a three-fold range (Figure 3, left). The curve is about two nats above the analytic one because of the runs' evidence bias at 64 dimensions. The relevant claim concerns *differences* in $\log \mathcal{Z}$, shown in the middle panel. Their error is below 0.05 over $\sigma \in [0.096, 0.145]$ and below 0.5 over $[0.072, 0.200]$, then grows quickly. The *effective sample size* in Eq. 13 explains this pattern: it is 31% of the dead points at σ_0 and remains so over most of the range, before collapsing where the error increases.

The gradient is also accurate throughout. The auto-differentiated $\partial \log \mathcal{Z}/\partial\sigma$ tracks the analytic derivative across the interval (Figure 3, right). Its largest discrepancy is 110 at $\sigma = 0.06$, where the derivative is 1394: a 7.9% relative error at the edge of the sweep and smaller elsewhere. This is sufficient for empirical Bayes. Gradient ascent from $\sigma = 0.2$ converges to $\sigma^\star = 0.107$, compared with the maximum-evidence value 0.103 and data-generating value 0.100, using the six existing runs rather than a fresh run for each candidate σ.

E SDEMEM: Specification and Results

Table 3 gives the eleven population parameters, their priors, and the values used to simulate the data. The random effects are $b_{ij}^{(m)} = \exp(\omega z_{ij}^{(m)})$, where $z_{ij}^{(m)} \sim \mathcal{N}(0, 1)$ is drawn once at a fixed seed and inferred jointly with the population parameters. With $M = 5$, this gives $11 + 4M = 31$ dimensions. Each latent state is initialized from the stationary distribution of its own drift, obtained by solving $A^{(m)}P + PA^{(m)\top} = \mathrm{diag}(\sigma^2)$, and observed at $T = 20$ time points with $\Delta t = 0.1$. All priors are mapped from the unit hypercube by inverse CDF, so the sampler's uniform box coincides with the model prior.

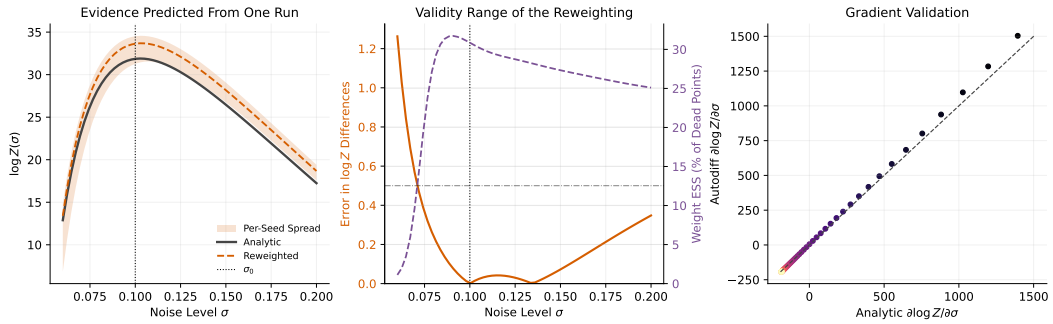


Figure 3: Differentiable evidence on a 64-D linear-Gaussian model, from six runs performed once at $\sigma_0 = 0.1$. *Left*: $\log \mathcal{Z}(\sigma)$ predicted by reweighting the fixed dead points, against the analytic curve, with the per-seed spread shaded; the vertical offset is the runs' own evidence bias at this dimension. *Center*: the error in log-evidence *differences* relative to σ_0, against the effective sample size of the importance weights that governs it. *Right*: the auto-differentiated $\partial \log \mathcal{Z}/\partial\sigma$ against its analytic value.

Table 3: SDEMEM population parameters: prior and the true value used to generate the data. $\mathcal{LN}(m, s)$ denotes a log-normal with log-mean m and log-sd s.

Parameter	Prior	True value	Role
α_1	$\mathcal{N}(0, 3^2)$	1.00	long-run mean, component 1
α_2	$\mathcal{N}(0, 3^2)$	-0.50	long-run mean, component 2
β_{11}	$\mathcal{LN}(\log 0.5,\ 0.8)$	0.80	self mean-reversion, component 1
β_{12}	$\mathcal{LN}(\log 0.3,\ 0.8)$	0.20	coupling $2 \to 1$
β_{21}	$\mathcal{LN}(\log 0.3,\ 0.8)$	0.15	coupling $1 \to 2$
β_{22}	$\mathcal{LN}(\log 0.5,\ 0.8)$	0.50	self mean-reversion, component 2
σ_1	$\mathcal{LN}(\log 0.4,\ 0.9)$	0.50	diffusion, component 1
σ_2	$\mathcal{LN}(\log 0.4,\ 0.9)$	0.30	diffusion, component 2
ξ_1	$\mathcal{LN}(\log 0.12,\ 0.9)$	0.15	measurement sd, component 1
ξ_2	$\mathcal{LN}(\log 0.12,\ 0.9)$	0.10	measurement sd, component 2
ω	$\mathcal{LN}(\log 0.3,\ 0.7)$	0.30	random-effect sd

Table 4: SDEMEM posterior sampling diagnostics (31-D, five seeds). ESS is the `coda` spectral effective sample size (Plummer et al., 2006), averaged over population parameters. ESS/N is relative to the number of returned samples and ESS/evals to the likelihood budget; $\widehat{R}_{\max}$ is the largest Gelman–Rubin statistic (Gelman & Rubin, 1992) across parameters and seeds. The two efficiency measures are distinct: a method can return well-mixed samples while using a large likelihood budget.

Method	ESS	ESS/N	ESS/sec	$\widehat{R}_{\max}$	ESS/evals (%)	Time (s)
GDHDNS (ours)	1879	0.20	**40.6**	1.03	0.64	**46.5**
`dynesty`	**5137**	0.28	6.8	1.00	0.26	755.4
`PolyChord`	2179	0.18	7.7	1.00	0.29	283.8
Tempered SMC	211	0.41	6.3	4.55	0.05	32.6

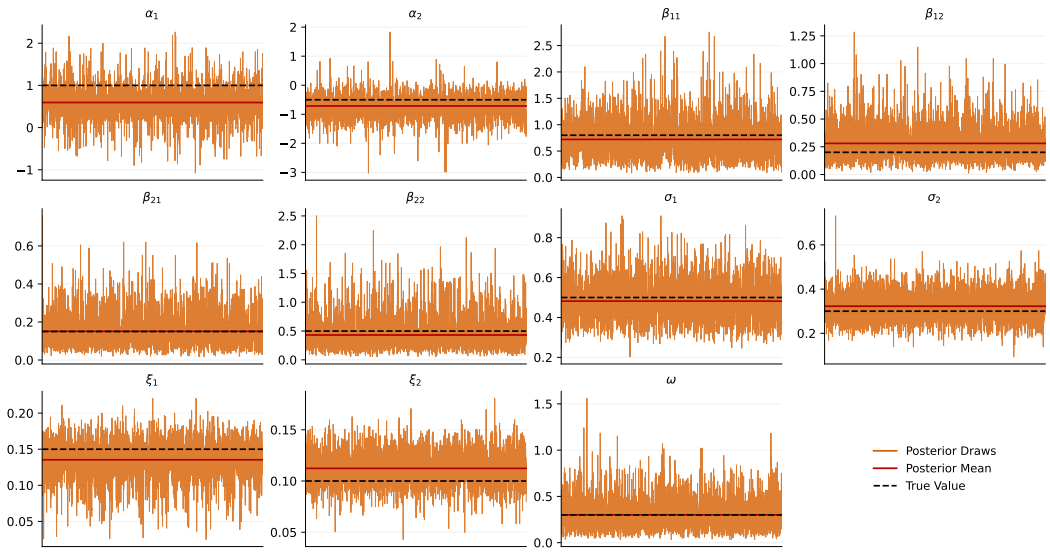


Figure 4: GDHDNS posterior traces for the eleven SDEMEM population parameters. Orange: equally weighted posterior draws; red: posterior mean; dashed black: the true value from Table 3.

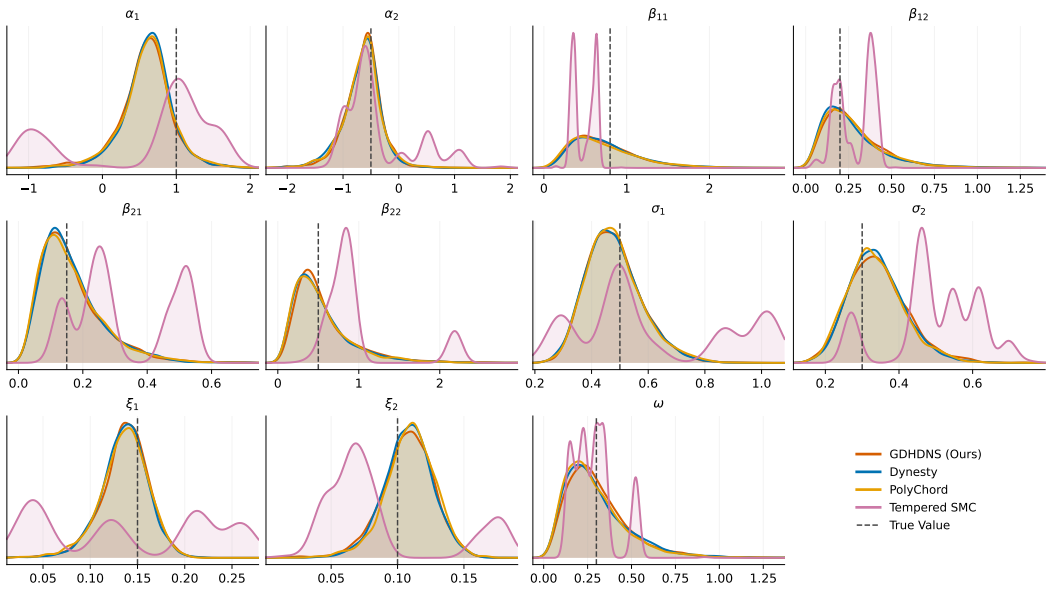
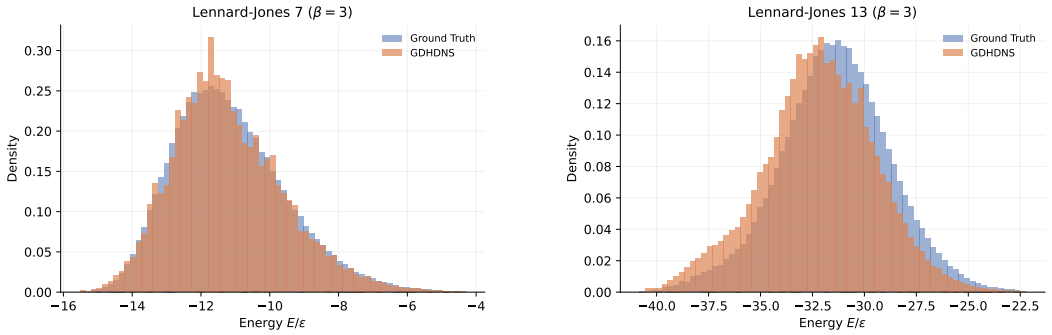


Figure 5: SDEMEM population parameter marginals for all four samplers (dashed: true value). The three nested samplers agree and cover the truth; tempered SMC is degenerate here, which is the $\widehat{R} = 4.55$ of Table 4.

F Additional Results and Posterior Visualizations

Figure 6 shows the Lennard-Jones energy histograms discussed in §5.4. Figure 7 gives evidence bias at selected dimensions on per-panel scales; Figure 1 shows the corresponding log-scale sweeps. For the banana, Figures 8 and 9 show per-method *weighted* posteriors and their overlay with the exact contours. All four samplers recover the curved ridge and low-mass tails. The speckling within the 99% envelope is mostly a rendering artifact of the low-density fill. Figure 10 shows corner plots for the 8-D Gaussian and 10-D funnel.



(a) LJ7 (21-D). (b) LJ13 (39-D).

Figure 6: Boltzmann energy histograms at $\beta = 3$ with the GDHDNS posterior (orange) against a parallel-tempering reference (blue). LJ7 agrees across the distribution; on LJ13 the bulk is recovered but the low-energy tail is underpopulated, which is the multimodality limitation of the ensemble metric discussed in §5.4.

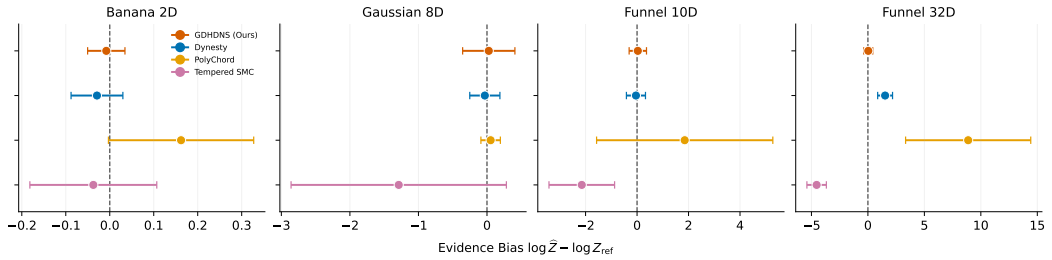


Figure 7: Evidence bias $\log \widehat{\mathcal{Z}} - \log \mathcal{Z}_{\mathrm{ref}}$ (mean and across-seed spread over five seeds) for the banana (2-D) and selected dimensions of the Gaussian and funnel sweeps. All methods agree on the easy target. At high dimension, each baseline fails on a different geometry, whereas GDHDNS remains unbiased in both sweeps.

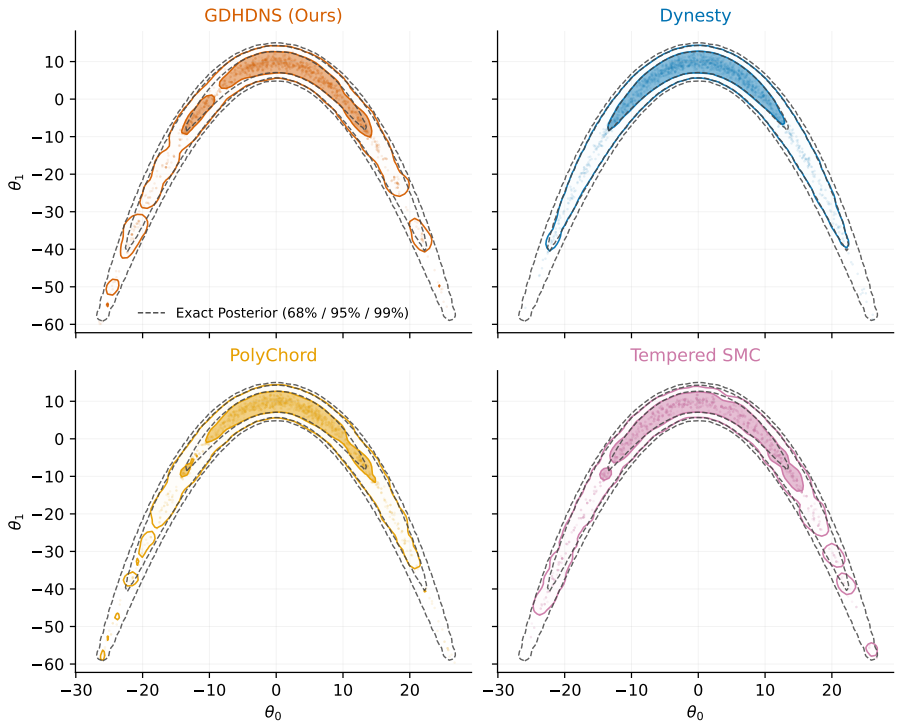


Figure 8: Per-method *weighted* posterior ($68/95\%$ filled) for the Banana experiment, with the exact posterior ($68/95/99\%$, dashed) overlaid in every panel.

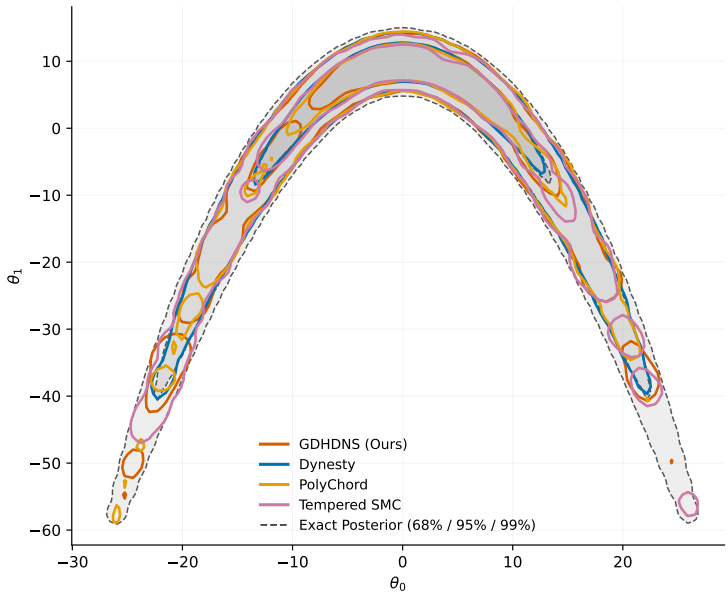
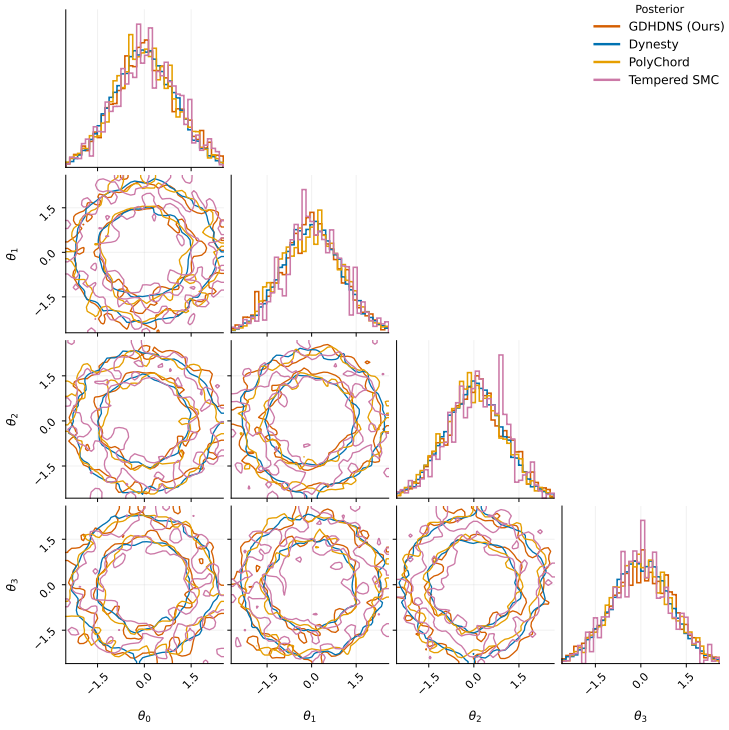
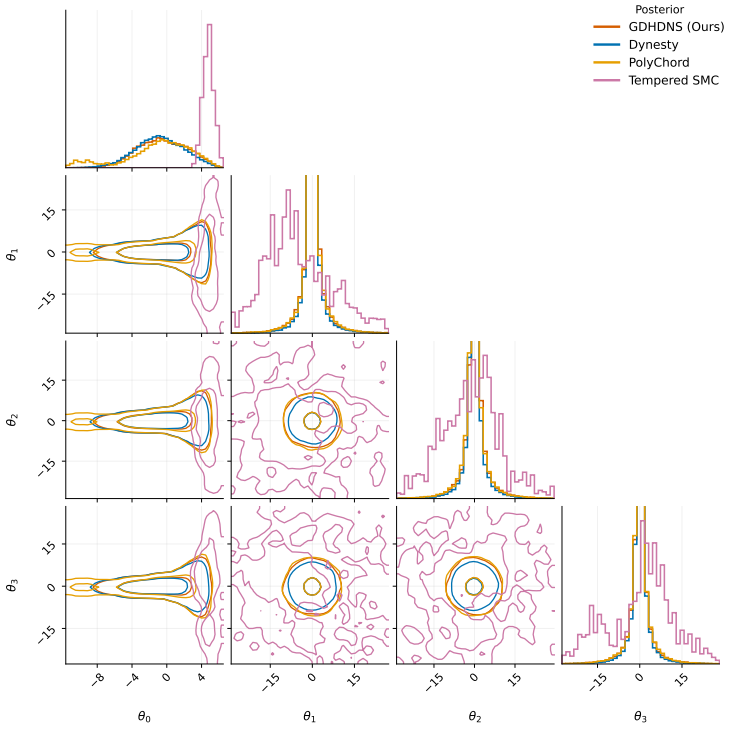


Figure 9: Sampler's $68/95\%$ posterior contours for the Banana experiment overlaid on the shaded exact posterior ($68/95/99\%$).



(a) Gaussian (8-D).



(b) Funnel (10-D).

Figure 10: Corner plots of the *weighted* posteriors (first four dimensions) for the Gaussian (8-D) and the funnel (10-D).