Gibbsian Polar Slice Sampling

Philip Schär [1] **Michael Habeck** [1] **Daniel Rudolf** [2]

Abstract

Polar slice sampling (Roberts & Rosenthal, 2002) is a Markov chain approach for approximate sampling of distributions that is difficult, if not impossible, to implement efficiently, but behaves provably well with respect to the dimension. By updating the directional and radial components of chain iterates separately, we obtain a family of samplers that mimic polar slice sampling, and yet can be implemented efficiently. Numerical experiments in a variety of settings indicate that our proposed algorithm outperforms the two most closely related approaches, elliptical slice sampling (Murray et al., 2010) and hit-and-run uniform slice sampling (MacKay, 2003). We prove the well-definedness and convergence of our methods under suitable assumptions on the target distribution.

1. Introduction

Bayesian inference heavily relies on efficient sampling schemes of posterior distributions that are defined on high-dimensional spaces with probability density functions that can only be evaluated up to their normalizing constants. We develop a Markov chain method that can be used to approximately sample a large variety of target distributions. For convenience we frame the problem in a black-box setting. We assume that the distribution of interest ν is defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ and given by a density, i.e. a measurable function $\varrho_\nu : \mathbb{R}^d \to \mathbb{R}_+ := [0, \infty)$, such that

$$\nu(A) = \frac{\int_A \varrho_\nu(x)\mathrm{d}x}{\int_{\mathbb{R}^d} \varrho_\nu(x)\mathrm{d}x}, \qquad A \in \mathcal{B}(\mathbb{R}^d).$$

In the following, knowledge of the normalization constant of ϱ_ν, i.e. the denominator of the above ratio, is not required. Since exact sampling is generally infeasible, we aim to produce approximate samples from ν, i.e. realizations of random variables whose distributions are, in some sense, close to ν.

To pursue this goal, we rely on Markov chain Monte Carlo (MCMC), which implements an irreducible and aperiodic Markov kernel P that leaves ν invariant. For such a kernel, well-established theory shows that the distribution of iterates X_n of a Markov chain $(X_n)_{n\in\mathbb{N}_0}$ with transition kernel P converges to ν as $n \to \infty$. An MCMC method generates a realization $(x_n)_{1\leq n\leq N} \subset \mathbb{R}^d$ of $X_1, \dots, X_N$ from $(X_n)_{n\in\mathbb{N}_0}$ and uses some or all of the realized chain iterates x_n as approximate samples of ν. Here, we focus on *slice sampling* MCMC methods, that use auxiliary "slice" or threshold random variables $(T_n)_{n\in\mathbb{N}_0}$. In general, T_n given X_{n-1} follows a uniform distribution over an interval that depends on X_{n-1}. Specifically, we consider *polar slice sampling* (PSS), which was proposed by Roberts & Rosenthal (2002). PSS factorizes the target density ϱ_ν into

$$\begin{aligned} \varrho_\nu(x) &= \varrho_\nu^{(0)}(x)\varrho_\nu^{(1)}(x), \\ \varrho_\nu^{(0)}(x) &= \|x\|^{1-d}, \\ \varrho_\nu^{(1)}(x) &= \|x\|^{d-1}\varrho_\nu(x) \end{aligned} \tag{1}$$

for $x \neq 0$, where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^d$. Given an initial value $x_0 \in \mathbb{R}^d$ with $x_0 \neq 0$ and $\varrho_\nu(x_0) > 0$, PSS recursively realizes the n-th chain iterate x_n from the $(n-1)$-th iterate x_{n-1} as follows: An auxiliary variable t_n is chosen as a realization of[1]

$$T_n \sim \mathcal{U}((0, \varrho_\nu^{(1)}(x_{n-1}))).$$

Given t_n, the next chain iterate is generated using polar coordinates $X_n = R_n\Theta_n$ where the *radius* R_n is a random variable on $\mathbb{R}_+$ and the *direction* Θ_n is a random variable on the $(d-1)$-sphere

$$\mathbb{S}^{d-1} := \{\theta \in \mathbb{R}^d \mid \|\theta\| = 1\} \subset \mathbb{R}^d.$$

To conform with the general slice sampling principle of Besag & Green (1993), the variables (R_n, Θ_n) need to be

[1]Microscopic Image Analysis Group, Friedrich Schiller University Jena, Jena, Germany [2]Faculty of Computer Science and Mathematics, University of Passau, Passau, Germany. Correspondence to: Daniel Rudolf <daniel.rudolf@uni-passau.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]With $\mathcal{U}(G)$ we denote the uniform distribution on a set G w.r.t. some reference measure that is clear from the context, as it is always either the Lebesgue measure, the surface measure on the $(d-1)$-sphere or a product measure of the two.

sampled from the joint uniform distribution

$$\mathcal{U}(\{(r,\theta) \in \mathbb{R}_+ \times \mathbb{S}^{d-1} \mid \varrho_\nu^{(1)}(r\theta) > t_n\}). \tag{2}$$

Standard slice sampling theory then guarantees that the resulting transition kernel has invariant distribution ν. For the convenience of the reader, in Appendix A we elaborate on how PSS can be derived from the general slice sampling principle of Besag & Green (1993). In the following, we refer to the process of sampling X_n given T_n as *X-update* of PSS.

The theoretical analysis of PSS by Roberts & Rosenthal (2002) offered performance guarantees for approximate sampling that are dimension-independent for rotationally invariant, log-concave target densities. Overall, their analysis suggests that PSS works generally robustly in high-dimensional scenarios. Despite this, PSS received little attention in the MCMC literature during the twenty years since its publication. We believe that this lack of engagement is not the result of PSS performing poorly on paper, but rather that of practical challenges in efficiently implementing it. Concurrent work by Rudolf & Schär (2023) supports this view. They prove – again in the rotationally invariant, log-concave setting – dimension-independent spectral gap estimates for PSS, which imply dimension-independence w.r.t. the mean squared error and the asymptotic variance within the central limit theorem of the MCMC average of a summary function.

The practical challenge in implementing PSS is that the polar variables (R_n, Θ_n) need to be jointly drawn uniformly from a high-dimensional set that often has a complicated structure. Therefore, this step is usually implemented by an acceptance/rejection scheme using uniform samples from a tractable superset. In moderate to high dimensions, for target densities ϱ_ν that are not rotationally invariant, the fraction of directions $\theta \in \mathbb{S}^{d-1}$ for which the set

$$\{r \in \mathbb{R}_+ \mid \varrho_\nu^{(1)}(r\theta) > t\}$$

is non-empty becomes tiny for most thresholds t occurring as realizations of T_n. This usually leads to an impractically low acceptance rate of the aforementioned acceptance/rejection scheme, such that – in expectation – an astronomically large number of proposals needs to be drawn during a single transition. In other words, although a valid implementation of PSS is available in principle, the iterations of the sampler are computationally inefficient resulting in exceedingly long simulation runs.

To address this deficiency, we develop an MCMC framework that imitates PSS, but is guaranteed to run in a computationally efficient manner. Imitating here refers to keeping the PSS structure and splitting the difficult joint uniform sampling of radius and direction of (2) into separate steps. Intuition suggests that if the resulting transition mechanism is close to the original version of PSS, then also some of its desirable convergence properties will be inherited. The eventually proposed MCMC algorithm is essentially tuning-free and explores the state space remarkably quickly. We provide a basic theoretical underpinning of our method, proving that it asymptotically samples from the target distribution ν under mild regularity conditions. Moreover, we illustrate its potential to improve upon related methods through a series of numerical experiments.

The remainder of this paper is structured as follows: In Section 2 we propose our modifications of PSS that we term *Gibbsian polar slice sampling*. To provide a better understanding, we present different variants that culminate in a Gibbsian polar slice sampler that incorporates a stepping-out and shrinkage procedure. In Section 3 we provide theoretical support for our methods. We discuss possible extensions and alternative mechanisms that can also be used in our framework in Section 4 and comment on related approaches in Section 5. In Section 6 we present a number of numerical experiments comparing our algorithm to the two most closely related ones that are similarly feasible to use in practice. We conclude with a short discussion in Section 7.

2. Gibbsian Polar Slice Sampling

The idea is to decompose the X-update of PSS by replacing the joint sampling of radius r_n and direction θ_n with separate updates of both variables in a Gibbsian fashion.

Variant 1

Our initial modification of PSS is mostly of theoretical interest. First, we only update the directional component of the last chain iterate x_{n-1}, resulting in an intermediate state $r_{n-1}\theta_n$, where $r_{n-1} = \|x_{n-1}\|$ and θ_n is a realization of

$$\Theta_n \sim \mathcal{U}(\{\theta \in \mathbb{S}^{d-1} \mid \varrho_\nu^{(1)}(r_{n-1}\theta) > t_n\}). \tag{3}$$

We then update the radial component of the intermediate state, resulting in the new chain iterate $x_n := r_n\theta_n$, where r_n is a realization of

$$R_n \sim \mathcal{U}(\{r \in \mathbb{R}_+ \mid \varrho_\nu^{(1)}(r\theta_n) > t_n\}). \tag{4}$$

In contrast to standard Gibbs sampling which cycles over coordinate-wise updates of the current state, we rely on a systematic conditional renewal in terms of the polar transformation components given as radius and direction. Although the modification does not solve the runtime issue, it lays the groundwork for further algorithmic improvements.

At this stage, we would already like to mention Theorem 3.1 which states that ν is the invariant distribution of the transition kernel of the 1st variant of *Gibbsian polar slice sampling* (GPSS). Therefore, GPSS provides a correct MCMC method for targeting ν.

We note that by randomizing the deterministic updating scheme (i.e. rather than always updating the direction first and then the radius, both are sampled in random order), one can obtain the stronger statement that the transition kernel is reversible w.r.t. ν.

Variant 2

The direction update is still a challenging implementation issue, since it requires sampling of a uniform distribution over a $(d-1)$-dimensional set with d possibly being large. Therefore, to sample the next direction, we suitably use (ideal) spherical slice sampling (Habeck et al., 2023), a recently developed MCMC method for (approximate) sampling from distributions on $\mathbb{S}^{d-1}$. The radius update is not changed in this variant (since it consists of a 1-dimensional uniform distribution sampling step).

Let $\mathbb{S}_\theta^{d-2}$ be the *great subsphere w.r.t.* θ, i.e. the set

$$\{\vartheta \in \mathbb{S}^{d-1} \mid \theta^T \vartheta = 0\}$$

of directions in $\mathbb{S}^{d-1}$ that are orthogonal to θ. In (Habeck et al., 2023) it is shown that one can sample uniformly from $\mathbb{S}_\theta^{d-2}$ as follows: Draw $V_1 \sim \mathcal{N}_d(0, I_d)$, set $V_2 := V_1 - (\theta^T V_1)\theta$ and finally $V_3 := V_2/\|V_2\|$, then $V_3 \sim \mathcal{U}(\mathbb{S}_\theta^{d-2})$. Furthermore, for any $y \in \mathbb{S}_\theta^{d-2}$ the set

$$\{\theta \cos(\omega) + y \sin(\omega) \mid \omega \in [0, 2\pi)\}$$

is the unique great circle in $\mathbb{S}^{d-1}$ that contains both θ and y.

A single iteration of this variant imitates the X-update of PSS as follows: First, a reference point y is drawn uniformly from the great subsphere w.r.t. the direction θ_{n-1} of the previous sample. The new direction θ_n is then sampled uniformly from the great circle of $\mathbb{S}^{d-1}$ running through both θ_{n-1} and y, intersected with

$$\{\theta \in \mathbb{S}^{d-1} \mid \varrho_\nu^{(1)}(r_{n-1}\theta) > t_n\}.$$

Then, a new radius is chosen by sampling (4).

Variant 3: The Concrete Algorithm

In practice, the univariate direction and radius updates of the 2nd variant of GPSS still need to be implemented as acceptance/rejection schemes that might exhibit low acceptance rates. Therefore, the final variant of GPSS replaces both updates with adaptive procedures that are essentially guaranteed to be fast and result in an algorithm that empirically converges against the correct target distribution.

In the radius update, we use the stepping-out and shrinkage procedure as proposed for uniform slice sampling by Neal (2003). In the direction update, we incorporate a shrinkage procedure. Actually, our direction update can be interpreted as running the shrinkage-based spherical slice sampler (Habeck et al., 2023). This ultimate variant is readily implemented by Algorithms 1, 2 and 3.

Algorithm 1 Gibbsian Polar Slice Sampling

Input: target density ϱ_ν, initial value $x_0 \in \mathbb{R}^d$ with $x_0 \neq 0$ and $\varrho_\nu(x_0) > 0$, initial interval length $w > 0$
Define $\varrho_\nu^{(1)} : x \mapsto \|x\|^{d-1} \varrho_\nu(x)$
Set $r_0 := \|x_0\|$ and $\theta_0 := x_0/r_0$
for $n = 1, 2, \ldots$ **do**
  Draw $T_n \sim \mathcal{U}((0, \varrho_\nu^{(1)}(x_{n-1})))$, call result t_n
  $\theta_n :=$ Geodesic_Shrinkage$(\varrho_\nu^{(1)}, t_n, r_{n-1}, \theta_{n-1})$
  $r_n :=$ Radius_Shrinkage$(\varrho_\nu^{(1)}, t_n, r_{n-1}, \theta_n, w)$
  $x_n := r_n \theta_n$
end for
return $(x_n)_{n \geq 0}$

Algorithm 2 Geodesic Shrinkage

Input: transform $\varrho_\nu^{(1)}$ of target density, current threshold t_n, current radius r_{n-1}, current direction θ_{n-1}
Draw $Y \sim \mathcal{U}(\mathbb{S}_{\theta_{n-1}}^{d-1})$, call the result y
Draw $\omega_{\max} \sim \mathcal{U}([0, 2\pi])$, set $\omega_{\min} := \omega_{\max} - 2\pi$
repeat
  Draw $\Omega \sim \mathcal{U}([\omega_{\min}, \omega_{\max}])$, call result ω
  Set $\theta_n := \theta_{n-1} \cos\omega + y \sin\omega$
  if $\omega < 0$ **then** $\omega_{\min} := \omega$ **else** $\omega_{\max} := \omega$
until $\varrho_\nu^{(1)}(r_{n-1}\theta_n) > t_n$
return θ_n

Briefly a single iteration works as follows: An element y of the great subsphere is determined as in variant 2 and then the new direction θ_n is sampled via a shrinkage procedure on the great circle of $\mathbb{S}^{d-1}$ running through both θ_{n-1} and y. Finally, a new radius is chosen via a stepping-out and shrinkage procedure on the ray emanating from the origin in direction θ_n, where shrinkage can be performed around the old radius r_{n-1} because it satisfies the target condition by construction.

3. Validation – Theoretical Support

We provide some basic theoretical underpinning of the proposed methods with a focus on the 2nd variant of GPSS. The reasons for this are threefold: First, in principle this variant can also be implemented by using univariate acceptance/rejection schemes[2]. Second, we want to avoid any deep discussion about the stepping-out procedure that is involved in Algorithm 3. Third, since Algorithms 2 and 3 both

[2]For the direction update this is immediately possible, since $[0, 2\pi]$ is a superset of the corresponding acceptance region. For the radius update this is more complicated and may require some structural knowledge about ϱ_ν.

Algorithm 3 Radius Shrinkage

Input: transform $\varrho_\nu^{(1)}$ of target density, current threshold t_n, current radius r_{n-1}, current direction θ_n, initial interval length $w > 0$
Sample $U \sim \mathcal{U}([0,1])$, call the result u
$r_{\min} := \max(r_{n-1} - u \cdot w, 0)$
$r_{\max} := r_{n-1} + (1-u) \cdot w$
while $r_{\min} > 0$ **and** $\varrho_\nu^{(1)}(r_{\min}\theta_n) > t_n$ **do**
  $r_{\min} := \max(r_{\min} - w, 0)$
end while
while $\varrho_\nu^{(1)}(r_{\max}\theta_n) > t_n$ **do**
  $r_{\max} := r_{\max} + w$
end while
Sample $R_n \sim \mathcal{U}([r_{\min}, r_{\max}])$, call the result r_n
while $\varrho_\nu^{(1)}(r_n\theta_n) \leq t_n$ **do**
  if $r_n < r_{n-1}$ **then** $r_{\min} := r_n$ **else** $r_{\max} := r_n$
  Sample $R_n \sim \mathcal{U}([r_{\min}, r_{\max}])$, call the result r_n
end while
return r_n

contain a shrinkage procedure, no explicit representation of the corresponding transition kernels is available to our knowledge. The proofs of the following statements can be found in Appendix B.

Theorem 3.1. *The transition kernels corresponding to the 1st and 2nd variant of GPSS admit ν as invariant distribution.*

To verify that ν is an invariant distribution, we argue that the direction and radius update are both individually reversible w.r.t. ν. This could also serve as a strategy to prove the invariance of the 3rd variant of GPSS. For the direction update (Algorithm 2), this can be done by virtue of results of (Habeck et al., 2023; Hasenpflug et al., 2023). For the radius update, the interplay of stepping-out and shrinkage makes it difficult to prove reversibility rigorously. However, intuitively the arguments of Neal (2003) apply.

Under weak regularity assumptions on ϱ_ν we also get a convergence statement.

Theorem 3.2. *Assume that the target density ϱ_ν is strictly positive, i.e. $\varrho_\nu \colon \mathbb{R}^d \to (0,\infty)$. Then, for ν-almost every initial value x_0, the distribution of an iterate X_n of a Markov chain $(X_n)_{n\in\mathbb{N}_0}$ with $X_0 = x_0$ and transition kernel corresponding to the 2nd variant of GPSS converges to ν in total variation as $n \to \infty$.*

Under an additional restrictive structural requirement the convergence result also holds for the 3rd variant of GPSS. Namely, if we assume that the radius shrinkage of Algorithm 3 with threshold t_n, current radius r_{n-1} and direction θ_n realizes sampling w.r.t.

$$\mathcal{U}(\{r \in \mathbb{R}_+ \mid \varrho_\nu^{(1)}(r\theta_n) > t_n\}).$$

For example, this is true if $\varrho_\nu^{(1)}$ is unimodal along rays. This actually is a scenario where the direction update relies on the shrinkage-based Algorithm 2, but the radius update is exact as in the 2nd variant of GPSS.

4. Alternative Transition Mechanisms

We emphasize that our 3rd variant of GPSS is just one of many possible ways to create a valid and efficiently implementable method that is based on the 2nd variant. For example, many of the ideas in Section 2.4 of (Murray et al., 2010) for modifying the shrinkage procedure used by elliptical slice sampling easily transfer to the shrinkage procedure used in the direction update of GPSS. For the radius update, one could drop stepping-out, i.e. just place an interval of size $w > 0$ randomly around the current r_n and then use the shrinkage procedure right away. With the same choice of the hyperparameter w this reduces the number of target density evaluations per iteration at the cost of also reducing the speed at which the sampler explores the distribution of interest. One could also replace our radius update by the update mechanism of latent slice sampling (Li & Walker, 2023). In principle, one could even use Metropolis-Hastings transitions for either one of the updates. As long as each transition mechanism leaves the corresponding distribution in either (3) or (4) invariant, the resulting sampler should be well-behaved. In this sense, GPSS provides a flexible framework that leads to a variety of different samplers.

5. Related Work

The radius update we use for GPSS in the 3rd variant has previously been considered in Section 4.2.2 of (Thompson, 2011), where it was suggested to alternate it with a standard update on $\mathbb{R}^d$, i.e. one that does not keep the radius fixed. Intuitively, our GPSS improves upon this approach by introducing a dedicated direction update, which, by not wasting effort on exploring the entire sample space, can more efficiently determine a (hopefully) good new direction.

Furthermore, although GPSS was developed as imitating classical polar slice sampling, it also bears some resemblance to two other slice sampling methods, namely *elliptical slice sampling* (ESS) (Murray et al., 2010) and *hit-and-run uniform slice sampling*[3] (HRUSS). Both methods have been analyzed theoretically to some extent, ESS in (Natarovskii et al., 2021; Hasenpflug et al., 2023), HRUSS in (Latuszyński & Rudolf, 2014) and an idealized version of HRUSS in (Rudolf & Ullrich, 2018). Moreover, a sophisticated sampling algorithm that employs a large number of ESS-based Markov chains running in parallel has been

[3]The idea of HRUSS is already formulated in the paragraph 'How slice sampling is used in real problems' in Section 29.7 in (MacKay, 2003)

suggested in (Nishihara et al., 2014).

Both ESS and HRUSS follow the same basic principle as all other slice sampling approaches: In iteration n, they draw a threshold t_n w.r.t. the value of some fixed function $\varrho_\nu^{(1)}$ at the latest sample x_{n-1}. They then determine the next sample x_n by approximately sampling from some distribution restricted to the *slice* (or level set) of $\varrho_\nu^{(1)}$ at threshold t_n.

For a given threshold, ESS draws an auxiliary variable from a mean zero multivariate Gaussian and performs shrinkage on the zero-centered ellipse running through both the auxiliary point and the latest sample, using the latter as the reference point for shrinkage. This is very similar to the direction update in the 3rd variant of GPSS[4], where we draw the auxiliary variable y from the uniform distribution on the great subsphere w.r.t. the direction θ_{n-1} of the latest sample x_{n-1} and then perform shrinkage on the unique great circle of $\mathbb{S}^{d-1}$ that contains both y and θ_{n-1}.

HRUSS on the other hand uses neither ellipses nor circles. For a given threshold, it proceeds by choosing a random direction (uniformly from $\mathbb{S}^{d-1}$) and determining the next sample x_n by performing stepping-out and shrinkage procedures on the line through the latest sample x_{n-1} in this direction. This is obviously very similar to the radius update used in the 3rd GPSS variant. The two major differences are that the latter does not use an auxiliary variable to determine the direction along which it performs the update, and that HRUSS performs an update on an entire line, whereas GPSS only considers a ray (by requiring radii to be non-negative).

Based on these comparisons, we may expect an iteration of our 3rd GPSS variant to be roughly as costly as one of ESS and one of HRUSS combined. Various experiments suggest this to be a good rule of thumb, though GPSS actually tends to be faster than ESS if the latter is not well-tuned (we explain what this means in Section 6). Although HRUSS is clearly the fastest among the three, it usually takes large amounts of very small steps, so that it tends to lag behind the other two samplers when considering metrics like effective sample size per time unit.

6. Experiments

We illustrate the strengths of our 3rd variant of GPSS by a series of numerical experiments in which we compare its performance with those of ESS and HRUSS[5]. Source code allowing the reproduction (in nature) of our experimental results is provided in a github repository[6]. Some remarks on the pitfalls in implementing our method are provided in Appendix C, additional sampling statistics in Appendix D, an additional experiment on Bayesian logistic regression in Appendix E, and further illustrative plots in Appendix F.

Before we discuss the individual experiments, some explanation of our general approach to using and comparing these methods is in order. The positive hyperparameter w in HRUSS determines the initial size of the search interval in the stepping-out procedure, playing the same role as the eponymous parameter of GPSS in Algorithm 3. Therefore, when we compare the two samplers in some fixed setting, we always use the same value of w for both of them. We note, however, that neither parameter has much of an influence on the sampler's performance as long as they are not chosen orders of magnitude too small. Accordingly, we did not carefully tune them in any of the experiments.

As ESS is technically only intended for posterior densities with mean zero Gaussian priors, whenever we are in a different setting, we artificially introduce a mean zero Gaussian prior and use the target divided by the prior as likelihood. As we will see in the following, the performance of the method can be quite sensitive to the choice of the covariance matrix for this artificial prior, so we sometimes consider both a "naive" (or "untuned") and a "sophisticated" (or "tuned") choice in our comparisons.

To make the sophisticated choices, we typically rely on the fact that the radii of samples from $\mathcal{N}_d(0, I_d)$ scatter roughly around $\sqrt{d}$, but also on an understanding of the desired range of radii gained either through theoretical analysis of the target density or by examining samples generated by GPSS. In one of our experiments, where the variables are highly correlated in the sense that their joint density – our target density – is very far from rotational invariance, we even use the empirical covariance of samples generated by GPSS as the covariance matrix for ESS.

6.1. Multivariate Standard Cauchy Distribution

First we considered the multivariate generalization of the pathologically heavy-tailed standard Cauchy distribution, i.e. the distribution ν with $\varrho_\nu(x) = (1+\|x\|^2)^{-(d+1)/2}$. We chose the sample space dimension to be $d = 100$, initialized all samplers with $x_0 := (1, \dots, 1)^T$ and ran each of them for $N = 10^6$ iterations. Since ν is rotationally invariant and the covariance of naive ESS was already on a reasonable scale, we refrained from using a tuned version of ESS here.

As the canonical test statistics mean and covariance are undefined for ν, we instead measured the sample quality by letting the samplers estimate a probability w.r.t. ν. For

[4] In fact, both approaches can even be implemented to use the same random variables, d samples from $\mathcal{N}(0, 1)$, for determining the one-dimensional object to perform shrinkage on.

[5] All of our experiments were conducted on a workstation equipped with an AMD Ryzen 5 PRO 4650G CPU.

[6] https://github.com/microscopic-image-analysis/Gibbsian_Polar_Slice_Sampling

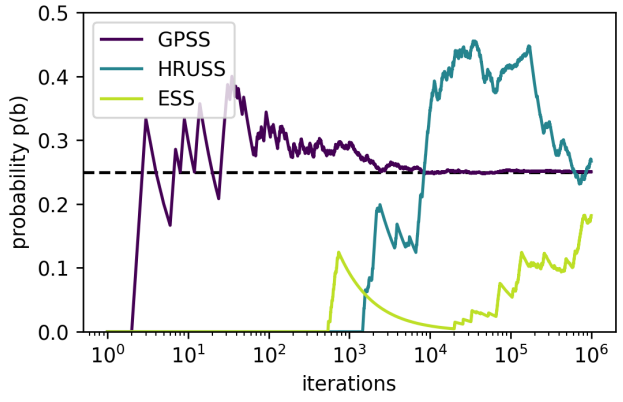

Figure 1. Progression over iterations of the estimates of $p(b)$ from (5) in the multivariate standard Cauchy experiment. Note the logarithmic scale. The dashed black line denotes the ground truth, which, using our knowledge of the target's rotational invariance, we could compute by numerically solving a one-dimensional integral.

this we chose the probability of the event that $\|Z\| > b$ and simultaneously $Z_1 > 0$, where $Z = (Z_1, \dots, Z_d)^T \sim \nu$ and $b > 0$. Described in formulas, the samplers estimated

$$p(b) := \nu(\{z \in \mathbb{R}^d \mid \|z\| > b,\ z_1 > 0\}), \tag{5}$$

where z_1 is the first entry of $z = (z_1, \dots, z_d)^T \in \mathbb{R}^d$. Note that the condition $\|Z\| > b$ measures how well the sample radii reflect those that would occur in exact sampling from ν, whereas the condition $Z_1 > 0$ detects if the sample directions (i.e. the samples divided by their radii), are biased towards either one of the two half-spaces separated by the hyperplane through all but the first coordinate axes. Hence both radii and directions need to be sampled well in order for the estimate of $p(b)$ to quickly approach the true value.

The progressions of the samplers' estimates of $p(b)$ are shown in Figure 1. It can be seen that those produced by GPSS converge to the true value orders of magnitude faster than those by both HRUSS and ESS. In Appendix F, Figure 11, we provide a peek into the sampling behind these results by displaying the progression of radii and log radii over $N_{\text{window}} = 5 \cdot 10^4$ iterations. As exact sampling from ν is tractable (using samples from the multivariate normal and the χ^2-distribution), we also display traces of exact i.i.d. samples for comparison.

Figure 11 suggests that the samples produced by GPSS are comparable in quality to i.i.d. ones, and shows that those by HRUSS and ESS are certainly not. Upon closer examination, the reason for this becomes evident: Although all samplers take trips to the distribution's tails, those taken by HRUSS and ESS last several orders of magnitude longer than those of GPSS into the same distance. Consequently, HRUSS and ESS make their excursions to the extremely far-off parts of the tails (say, the points of radii in the thousands or more) with vanishingly small frequency. Thus GPSS needs a much shorter chain to properly reflect the tails of the target distribution. We acknowledge, however, that the advantage GPSS has over HRUSS in this setting would be considerably smaller if the target density was not centered around the origin.

6.2. Hyperplane Disk

Next we considered the target density

$$\varrho_\nu(x) = \exp\Big(-(\textstyle\sum_{i=1}^d x_i)^2 - \|x\|^2\Big) \tag{6}$$

for $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$. Intuitively, the sum term within the density leads to a concentration of its probability mass around a hyperplane, i.e. a $(d-1)$-dimensional subspace, given by the set of points $x \in \mathbb{R}^d$ for which the sum term vanishes. The norm term ensures that the function is integrable with Gaussian tails in all directions, which intuitively further concentrates the distribution around a circular disk within the hyperplane. We set $d = 200$ and initialized all chains in an area of high probability mass. We then ran each sampler for $N = 10^4$ iterations. To tune ESS, we took all N samples generated by GPSS, computed their empirical covariance matrix and used it as the covariance parameter for the artificial Gaussian prior of ESS.

The results are shown in Appendix F, Figure 12. The progression of the sample radii suggests that GPSS has a considerable advantage over all other approaches. However, impressions based on sample radii should be taken with some caution, since GPSS is the only method that specifically updates the radii during sampling. Accordingly, when considering empirical step sizes, the advantage of GPSS is less pronounced, but still present. For example, the mean step sizes are ≈ 5.0 for GPSS, ≈ 3.5 for tuned ESS, ≈ 2.4 for untuned ESS and ≈ 0.6 for HRUSS.

There are two drawbacks regarding the performance of tuned ESS, which by the aforementioned aspects is the closest competitor to our method in this setting. On the one hand, as noted before, we tuned ESS using the samples generated by GPSS, thus relying on the robust performance of GPSS to compute a good estimate of the target distribution's true covariance matrix. On the other hand, the computational overhead of sampling from a multivariate Gaussian with non-diagonal covariance matrix slows tuned ESS down significantly. As a result, tuned ESS consistently ran slower than all the other samplers in this experiment. The advantage of GPSS over tuned ESS is remarkable, not just because only the latter uses a proposal distribution adjusted to the shape of the target distribution, but also because the tails of the target are Gaussian, which in principle should benefit ESS (by virtue of its proposal distribution being Gaussian

as well).

We attribute the relatively poor performance of HRUSS to the curse of dimensionality: As a result of the target density being narrowly concentrated around a hyperplane of a very high-dimensional space, from any given point in the target's high probability region there are very few directions in which one can take large steps without leaving the high-probability region. Nevertheless, HRUSS isotropically samples the direction along which it will move, such that most of the time only small steps along the sampled direction are allowed.

6.3. Axial Modes

Our third experiment was concerned with the target density

$$\varrho_\nu(x) = \|x\|_\infty^4 \exp(-\|x\|_1). \tag{7}$$

Due to the counteracting forces of ∞-norm and 1-norm, ϱ_ν possesses $2d$ fairly isolated modes, 2 along each coordinate axis, see Figure 8 (appendix) for an illustration.

This particular structure enables an interesting quantitative diagnostic for the performance of MCMC methods targeting this distribution: For any given $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$, one can use the absolute values of its components to assign it to a pair of modes (that lie on the same coordinate axis) via

$$\text{axis}(x) := \operatorname*{argmax}_{1 \le i \le d} |x_i|.$$

For a finite chain of samples $(x^{(n)})_{n=1,\dots,N} \subset \mathbb{R}^d$ that was generated as the output of some MCMC method, numerous quantitative diagnostics can then be applied to the values $(\text{axis}(x^{(n)}))_{n=1,\dots,N}$. For example, one can say that the chain *jumped between modes* in step i if and only if

$$\text{axis}(x^{(i)}) \neq \text{axis}(x^{(i-1)}).$$

One can then count the *total number of jumps* within the N iterations (which provides information about how quickly the chain moved back and forth between the mode pairs) and compare these values between different chains. Alternatively, one could compute the *mean dwelling time*, i.e. the average number of iterations the chain spent at a mode pair until jumping to the next. This may be a more helpful quantity than the total number of jumps, because it is essentially independent of the number N of iterations. It may also be worthwhile to consider the *maximum dwelling time*, i.e. the largest number of iterations the chain spent at a mode pair without leaving, as this is more suitable than mean dwelling time and total number of jumps for detecting whether a chain occasionally gets stuck at a mode pair for excessively many iterations.

We ran each sampler for $N = 10^5$ iterations in each of the dimensions $d = 10, 20, \dots, 100$. As initialization we used $x_0 := (5, 1, \dots, 1)^T \in \mathbb{R}^d$. For the ESS covariance we considered both the usual naive choice $\Sigma = I_d$ and the carefully hand-tuned choice $\Sigma = (5 + d/10)^2/d \cdot I_d$.

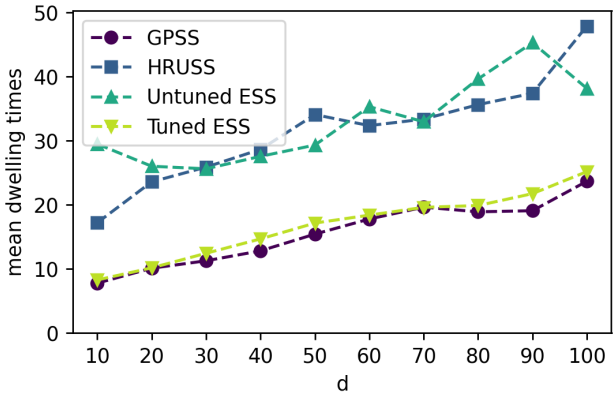

Figure 2. Progression over dimensions d of the mean dwelling time, determined based on $N = 10^5$ iterations, in the axial modes experiment, as described in Section 6.3.

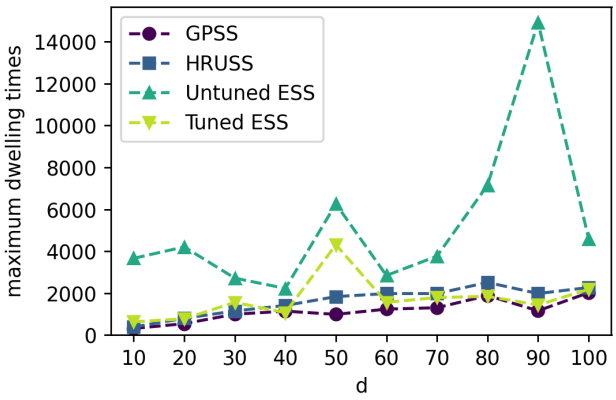

Figure 3. Progression over dimensions d of the maximum dwelling times within $N = 10^5$ iterations in the axial modes experiment, as described in Section 6.3.

In Figures 2 and 3 we display for each sampler the progression over the dimensions d of mean and maximum dwelling time. In terms of mean dwelling time, GPSS has a clear advantage over both untuned ESS and HRUSS, only the carefully tuned ESS is competitive with it. In terms of maximum dwelling time, GPSS is slightly ahead of all other samplers, even tuned ESS. In Appendix F, Figure 13 we provide a peek into the sampling behind these results by displaying the progression of currently visited mode pair in the last $N_{\text{window}} = 2 \cdot 10^4$ iterations of the samplers' runs for the highest dimension $d = 100$.

6.4. Neal's Funnel

In our fourth experiment, we considered an arguably even more challenging target density that was originally proposed by Neal (2003) and is commonly termed *Neal's funnel*. It is given by

$$\varrho_\nu(x) = \mathcal{N}(x_1; 0, 9) \prod_{i=2}^{d} \mathcal{N}(x_i; 0, \exp(x_1)), \qquad (8)$$

where we write $x = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ and denote by $\mathcal{N}(z; \mu, \sigma^2)$ the density of $\mathcal{N}(\mu, \sigma^2)$ evaluated at z. As the name suggests, the density is shaped like a funnel, having both a very narrow and a very wide region of high probability mass, which smoothly transition into one another. Besides being a challenging target, the funnel is of particular interest to us because its marginal distribution in the first coordinate is simply $\mathcal{N}(0, 9)$ and a sampler needs to explore both the narrow and the wide part of the funnel equally well in order to properly approximate this marginal distribution via the marginals of its samples (i.e. the set containing each of its samples truncated after the first component).

We ran the slice samplers for the funnel in dimension $d = 10$ (as suggested by Neal) and initialized all of them with $x_0 := (2, 0, \ldots, 0)^T \in \mathbb{R}^d$. For ESS we used the relatively well-tuned covariance $\Sigma = \mathrm{diag}(9, 70, \ldots, 70)$. For this experiment we also extend our comparison to a sampler that is only related to GPSS by the fact that it is also an MCMC method. Namely, we compare GPSS with the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014), which is widely regarded to be the current state-of-the-art in MCMC sampling[7]. We used the implementation of NUTS provided by the probabilistic programming Python library PyMC (Salvatier et al., 2016).

In this experiment we study the convergence to several target quantities. To enable a fair comparison of the samplers, we took differences in the speed of the iterations into account. This was achieved by allocating a fixed time budget of 1 minute to each sampler and letting it run for sufficiently many iterations to deplete this time budget, while tracking how much time had elapsed after each completed iteration[8]. We assessed the convergence ten times per second, resulting in 600 measurement times in total. Finally, we determined for each measurement time t and each sampler s the exact number of iterations $n_{t,s} \in \mathbb{N}$ it had completed up to that time (using the aforementioned logs) and estimated the target quantities from only the $n_{t,s}$ samples produced in these iterations. As target quantities we considered mean, standard deviation, 0.001-quantile and 0.999-quantile. To generate estimates of these quantities from marginal samples, we simply used their empirical versions.

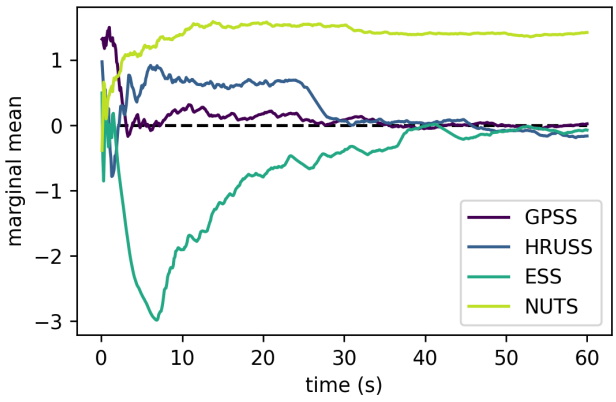

Figure 4. Progression over runtime of the empirical mean of the marginal samples in the experiment on Neal's funnel (8). The dashed black line denotes the ground truth.

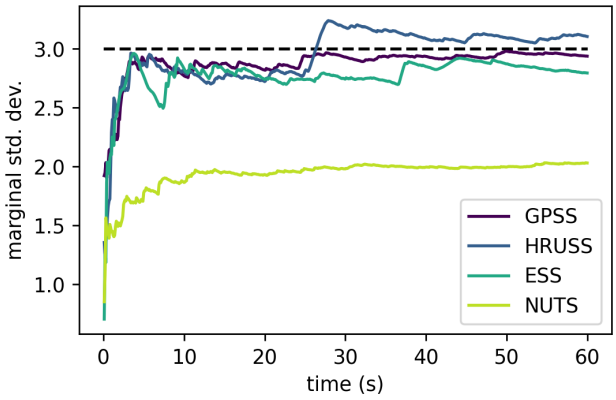

Figure 5. Progression over runtime of the empirical standard deviation of the marginal samples in the experiment on Neal's funnel (8). The dashed black line denotes the ground truth.

The progression over time of each sampler's approximations to these target quantities are shown in Figures 4, 5, 6 and 7. Additionally, we provide the marginal histograms of all samples generated within the time budget as well as a peek into the progression of the marginal samples and sample radii in Appendix F, Figure 14.

It can be clearly seen from the first four figures that GPSS performs better overall than any of the other methods. We note that HRUSS is more competitive with it than ESS and that GPSS converges well despite completing only $3.39 \cdot 10^5$ iterations in the allotted time, which is less than half the $7.95 \cdot 10^5$ iterations completed by HRUSS, and about the same as the $2.94 \cdot 10^5$ completed by ESS. In other words, had

[7] Note, however, that NUTS needs the target density to be differentiable and requires oracle access to its gradient, neither of which is true for the slice sampling methods we consider.

[8] Except for NUTS, where we only approximated this by assuming the runtime per iteration to be constant.

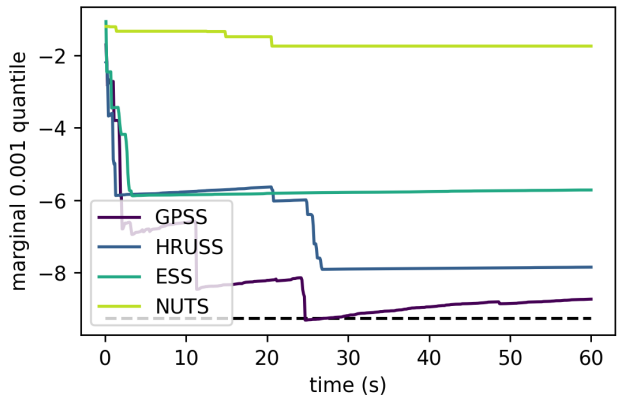

Figure 6. Progression over runtime of the empirical 0.001-quantile of the marginal samples in the experiment on Neal's funnel (8). The dashed black line denotes the ground truth.

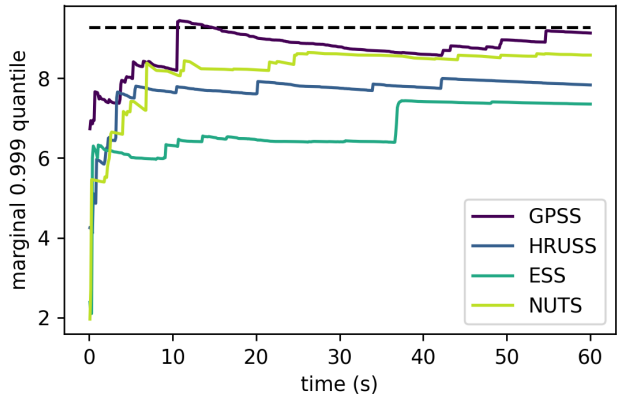

Figure 7. Progression over runtime of the empirical 0.999-quantile of the marginal samples in the experiment on Neal's funnel (8). The dashed black line denotes the ground truth.

we used the same number of iterations for all slice samplers – like in other experiments – the convergence results would attribute GPSS an even larger advantage. Regarding the performance of NUTS, we observe that it is very successful at retrieving the 0.999-quantile, but, due to its refusal to enter the narrow part of the funnel, it performs worst among the four methods not just for the 0.001-quantile, but also for mean and standard deviation.

7. Discussion

We introduced a Gibbsian polar slice sampling (GPSS) framework as a general Markov chain approach for approximate sampling from distributions on $\mathbb{R}^d$ given by an unnormalized Lebesgue density. The efficiently implementable version that we propose is essentially tuning-free. It has only a single hyperparameter with little impact on the method's performance, as long as it is not chosen orders of magnitude too small. GPSS can quickly produce samples of high quality, and numerical experiments indicate advantages compared to related approaches in a variety of settings.

Although GPSS is generally quite robust, its performance slowly deteriorates when the distance between the target distribution's center of mass and the coordinate origin is increased. This could potentially be avoided by automatically centering the target on the origin. However, such a modification would likely follow an adaptive MCMC approach and therefore result in a method that no longer fits the "strict" MCMC framework.

Particularly good use cases for GPSS appear to be heavy-tailed target distributions. For example, one could apply GPSS to intractable posterior distributions resulting from Cauchy priors (perhaps on just some of the variables) and likelihoods that do not change the nature of the tails. As illustrated in Section 6.1, GPSS can have enormous advantages over related methods when heavy tails are involved. Another type of target for which GPSS could be of practical use are distributions with strong funneling, which naturally occur in Bayesian hierarchical models (cf. Neal (2003)). By nature, these targets call for methods with variable step sizes, because they contain both very narrow regions, which necessitate small step sizes, and very wide regions, in which much larger step sizes are advantageous to speed up the exploration of the sample space. Due to their use of variable step sizes, slice sampling methods might be considered a natural choice and, as demonstrated in Section 6.4, GPSS appears to perform better than related slice samplers for such targets.

We envision that many more applications for GPSS will be found. Moreover, we think it may be possible to derive qualitative or even quantitative geometric convergence guarantees for GPSS. Aside from giving helpful insight into how well GPSS retains the desirable theoretical properties of PSS, this would further justify using GPSS in real-world applications, where the sample quality is often hard to validate. Finally, we emphasize that the 2nd variant of GPSS, the intermediate step between idealized framework and efficiently implementable method, can also be used as the foundation for a variety of other hybrid samplers (cf. Section 4).

Acknowledgements

We thank the anonymous referees for their suggestions. PS and MH gratefully acknowledge funding by the Carl Zeiss Foundation within the program "CZS Stiftungsprofessuren" and the project "Interactive Inference". We are grateful for the support of the DFG within project 432680300 – SFB 1456 subprojects A05 and B02.

References

Besag, J. and Green, P. J. Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society Series B*, 55(1):25–37, 1993.

Blackard, J. A. *Comparison of neural networks and discriminant analysis in predicting forest cover types*. PhD thesis, Colorado State University, 1998.

Blackard, J. A., Dean, D. J., and Anderson, C. W. Covertype data set. URL https://archive.ics.uci.edu/ml/datasets/Covertype.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. *Markov Chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, 2018.

Habeck, M., Hasenpflug, M., Kodgirwar, S., and Rudolf, D. Geodesic slice sampling on the sphere. arXiv preprint arXiv:2301.08056, 2023.

Hasenpflug, M., Natarovskii, S., and Rudolf, D. Reversibility of elliptical slice sampling revisited. arXiv preprint arXiv:2301.02426, 2023.

Hoffman, M. D. and Gelman, A. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

Latuszyński, K. and Rudolf, D. Convergence of hybrid slice sampling via spectral gap. arXiv preprint arXiv:1409.2709, 2014.

Li, Y. and Walker, S. G. A latent slice sampling algorithm. *Computational Statistics and Data Analysis*, 179, 2023.

MacKay, D. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

Murray, I., Adams, R., and MacKay, D. Elliptical slice sampling. *Journal of Machine Learning Research*, 9: 541–548, 2010.

Natarovskii, V., Rudolf, D., and Sprungk, B. Geometric convergence of elliptical slice sampling. *Proceedings of the 38th International Conference on Machine Learning*, 139:7969–7978, 2021.

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

Nishihara, R., Murray, I., and Adams, R. Parallel MCMC with generalized elliptical slice sampling. *Journal of Machine Learning Research*, 15, 2014.

Roberts, G. O. and Rosenthal, J. S. The polar slice sampler. *Stochastic Models*, 18(2):257–280, 2002.

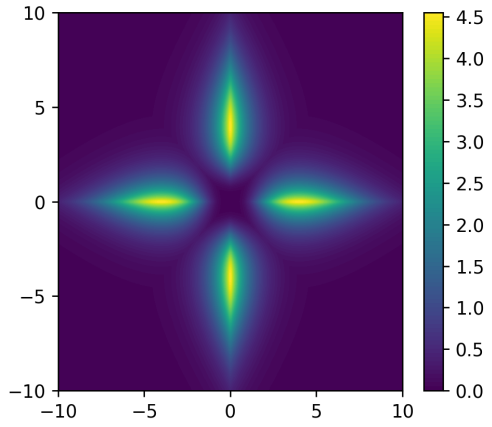

Figure 8. Illustration of the axial modes target density (7) in dimension $d = 2$.

Rudolf, D. and Schär, P. Dimension-independent spectral gap of polar slice sampling. arXiv preprint arXiv:2305.03685, 2023.

Rudolf, D. and Ullrich, M. Comparison of hit-and-run, slice sampler and random walk metropolis. *Journal of Applied Probability*, 55, 2018.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, 2(e55), 2016.

Schilling, R. L. *Measures, Integrals and Martingales*. Cambridge University Press, 2005.

Thompson, M. B. *Slice Sampling with Multivariate Steps*. PhD thesis, University of Toronto, 2011.

Tierney, L. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

A. Formal Derivation

In order to explain how PSS as described in Section 1 can be derived from the usual slice sampling framework laid out in (Besag & Green, 1993), we provide two technical tools. The first one is a well-known identity from measure theory.

Proposition A.1. *Any integrable real-valued function g on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ satisfies*

$$\int_{\mathbb{R}^d} g(x)dx = \int_0^\infty \int_{\mathbb{S}^{d-1}} g(r\theta)r^{d-1}\sigma_d(d\theta)dr,$$

where σ_d is the surface measure on $(\mathbb{S}^{d-1}, \mathcal{B}(\mathbb{S}^{d-1}))$.

Proof. See for example Theorem 15.13 in (Schilling, 2005). □

Though this identity does not by itself involve any probabilistic quantities, we apply it in a stochastic setting to obtain our second tool, which we term *sampling in polar coordinates*.

Corollary A.2. *Let ξ be a distribution on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ with probability density ϱ_ξ. Then by Proposition A.1 we get for any $A \in \mathcal{B}(\mathbb{R}^d)$ that*

$$\begin{aligned}\xi(A) &= \int_{\mathbb{R}^d} \mathbb{1}_A(x)\varrho_\xi(x)dx \\ &= \int_{\mathbb{S}^{d-1}} \int_0^\infty \mathbb{1}_A(r\theta)\varrho_\xi(r\theta)r^{d-1}dr\sigma_d(d\theta).\end{aligned}$$

Consequently a random variable $X \sim \xi$ can be sampled in polar coordinates as $X := R \cdot \Theta$ by sampling (R, Θ) from the joint distribution with probability density

$$(r, \theta) \mapsto \varrho_\xi(r\theta)r^{d-1}\mathbb{1}_{\mathbb{R}_+}(r)$$

w.r.t. $\lambda_1 \otimes \sigma_d$, where λ_1 denotes the Lebesgue measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$.

We can now apply the second tool to PSS. In the framework of Besag & Green (1993), the X-update of slice sampling for a target density factorized as in (1) is to be performed by sampling X_n from the distribution with unnormalized density

$$x \mapsto \varrho_\nu^{(0)}(x)\mathbb{1}_{(t_n,\infty)}(\varrho_\nu^{(1)}(x)) = \|x\|^{1-d}\mathbb{1}_{(t_n,\infty)}(\varrho_\nu^{(1)}(x)).$$

By Corollary A.2, this can equivalently be done by sampling X_n in polar coordinates as $X_n := R_n\Theta_n$, where (R_n, Θ_n) is drawn from the joint distribution with unnormalized density

$$\begin{aligned}(r, \theta) &\mapsto r^{1-d}\mathbb{1}_{(t_n,\infty)}(\varrho_\nu^{(1)}(r\theta))\, r^{d-1}\mathbb{1}_{\mathbb{R}_+}(r) \\ &= \mathbb{1}_{(t_n,\infty)}(\varrho_\nu^{(1)}(r\theta))\mathbb{1}_{\mathbb{R}_+}(r)\end{aligned}$$

w.r.t. $\lambda_1 \otimes \sigma_d$. As this distribution is simply (2), PSS as described in Section 1 is equivalent to the slice sampler resulting from factorization (1) in the general slice sampling framework of Besag & Green (1993).

B. Proofs

We assume some familiarity with transition kernels and Markov chains throughout this section. For details we refer to the introductory sections of (Douc et al., 2018).

We introduce some notation and provide a few observations. Let $C_\nu := \int_{\mathbb{R}^d} \varrho_\nu(x)\mathrm{d}x$, so that

$$C_\nu\, \nu(\mathrm{d}x) = \varrho_\nu^{(0)}(x)\varrho_\nu^{(1)}(x)\mathrm{d}x.$$

For $t > 0$ define the level set

$$L(t) := \{x \in \mathbb{R}^d \mid \varrho_\nu^{(1)}(x) > t\}.$$

Note that, by

$$\mathbb{1}_{(0,\varrho_\nu^{(1)}(x))}(t) = \mathbb{1}_{L(t)}(x),$$

the transition kernel corresponding to PSS for ν can be expressed as

$$P(x, A) := \frac{1}{\varrho_\nu^{(1)}(x)} \int_0^\infty \mu_t(A)\mathbb{1}_{L(t)}(x)\mathrm{d}t$$

for $x \in \mathbb{R}^d$, $A \in \mathcal{B}(\mathbb{R}^d)$, where we set

$$\mu_t(A) := \frac{\int_A \varrho_\nu^{(0)}(x)\mathbb{1}_{L(t)}(x)\mathrm{d}x}{\int_{\mathbb{R}^d} \varrho_\nu^{(0)}(x)\mathbb{1}_{L(t)}(x)\mathrm{d}x}$$

for $t > 0$. It is known that $\nu P = \nu$ (Roberts & Rosenthal, 2002). We formulate a criterion for invariance w.r.t. imitations of polar slice sampling.

Lemma B.1. *Suppose that for each $t > 0$ we have a transition kernel $U_X^{(t)}$ on $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$ with $\mu_t U_X^{(t)} = \mu_t$. Then the transition kernel Q on $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$ given by*

$$Q(x, A) := \frac{1}{\varrho_\nu^{(1)}(x)} \int_0^\infty U_X^{(t)}(x, A)\mathbb{1}_{L(t)}(x)dt$$

satisfies $\nu Q = \nu$.

Proof. Let $A \in \mathcal{B}(\mathbb{R}^d)$ arbitrary. Then

$$\begin{aligned}C_\nu \cdot \nu Q(A) &= C_\nu \int_{\mathbb{R}^d} Q(x, A)\nu(\mathrm{d}x) \\ &= \int_{\mathbb{R}^d} \int_0^\infty U_X^{(t)}(x, A)\mathbb{1}_{L(t)}(x)\mathrm{d}t\varrho_\nu^{(0)}(x)\mathrm{d}x \\ &= \int_0^\infty \int_{\mathbb{R}^d} U_X^{(t)}(x, A)\varrho_\nu^{(0)}(x)\mathbb{1}_{L(t)}(x)\mathrm{d}x\mathrm{d}t \\ &= \int_0^\infty \left(\int_{\mathbb{R}^d} U_X^{(t)}(x, A)\mu_t(\mathrm{d}x)\right) \\ &\qquad \cdot \left(\int_{\mathbb{R}^d} \varrho_\nu^{(0)}(x)\mathbb{1}_{L(t)}(x)\mathrm{d}x\right)\mathrm{d}t \\ &= \int_0^\infty \mu_t(A) \int_{\mathbb{R}^d} \varrho_\nu^{(0)}(x)\mathbb{1}_{L(t)}(x)\mathrm{d}x\mathrm{d}t \\ &= \int_{\mathbb{R}^d} \int_0^\infty \mu_t(A)\mathbb{1}_{L(t)}(x)\mathrm{d}t\varrho_\nu^{(0)}(x)\mathrm{d}x \\ &= C_\nu \int_{\mathbb{R}^d} P(x, A)\nu(\mathrm{d}x) \\ &= C_\nu \cdot \nu P(A) = C_\nu \cdot \nu(A),\end{aligned}$$

which shows $\nu Q = \nu$. □

Note that the framework we rely on was previously used in Lemma 1 in (Latuszyński & Rudolf, 2014) to analyze the reversibility of hybrid uniform slice sampling. Further

note that in proving Lemma B.1 we never use the precise factorization of ϱ_ν into $\varrho_\nu^{(0)}$ and $\varrho_\nu^{(1)}$ that is dictated by PSS. Hence the result also holds true for any other slice sampling scheme that adheres to this format. Moreover, in the previous lemma it is assumed that $U_X^{(t)}$ is a transition kernel on $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$. For given $t > 0$ it is sometimes more convenient to define a corresponding transition kernel $U_X^{(t)}$ on $L(t) \times \mathcal{B}(L(t))$. In such cases we consider $\overline{U}_X^{(t)}$ as extension of $U_X^{(t)}$ to $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$ given as

$$\overline{U}_X^{(t)}(x, A) = \begin{cases} U_X^{(t)}(x, A \cap L(t)) & x \in L(t) \\ \mathbb{1}_A(x) & x \notin L(t). \end{cases}$$

In the following we write $U_X^{(t)}$ for $\overline{U}_X^{(t)}$ and consider $U_X^{(t)}$ as extension if necessary. We turn our attention to the 1st variant of GPSS.

Lemma B.2. *For $T_n = t$ the transition kernel $U_X^{(t)}$ on $(L(t), \mathcal{B}(L(t)))$ of the X-update of the 1st variant of GPSS takes the form $U_X^{(t)} = U_{D_1}^{(t)} U_R^{(t)}$ with kernels*

$$U_{D_1}^{(t)}(r\theta, A) = \frac{\int_{\mathbb{S}^{d-1}} \mathbb{1}_A(r\tilde{\theta})\sigma_d(d\tilde{\theta})}{\int_{\mathbb{S}^{d-1}} \mathbb{1}_{L(t)}(r\tilde{\theta})\sigma_d(d\tilde{\theta})},$$
$$U_R^{(t)}(r\theta, A) = \frac{\int_0^\infty \mathbb{1}_A(\tilde{r}\theta)d\tilde{r}}{\int_0^\infty \mathbb{1}_{L(t)}(\tilde{r}\theta)d\tilde{r}}, \tag{9}$$

where $r \in \mathbb{R}_+, \theta \in \mathbb{S}^{d-1}, x = r\theta \in L(t)$ and $A \in \mathcal{B}(L(t))$. Moreover, $U_{D_1}^{(t)}$ and $U_R^{(t)}$ are reversible w.r.t. μ_t.

Proof. The overall X-update consists of consecutively realizing (3) and (4). Obviously, the direction update $r_{n-1}\theta_{n-1} \mapsto r_{n-1}\theta_n$ (according to (3)) corresponds to $U_{D_1}^{(t)}$ and the radius update $r_{n-1}\theta_n \mapsto r_n\theta_n$ (according to (4)) corresponds to $U_R^{(t)}$. Consequently, the transition kernel of the X-update is $U_X^{(t)} = U_{D_1}^{(t)} U_R^{(t)}$, that is, $U_X^{(t)}$ is the product of the kernels $U_{D_1}^{(t)}$ and $U_R^{(t)}$.

Now we prove the claimed invariance property. We use the fact that $\varrho_\nu^{(0)}(x) = \|x\|^{1-d}$. To simplify the notation, set

$$c_t := \left(\int_{\mathbb{R}^d} \|x\|^{1-d} \mathbb{1}_{L(t)}(x) \mathrm{d}x \right)^{-1}, \tag{10}$$

so that the restriction of μ_t to $L(t)$ can be written as

$$\mu_t(\mathrm{d}x) = c_t \|x\|^{1-d} \mathrm{d}x.$$

Observe that Proposition A.1 yields for all $A, B \in \mathcal{B}(L(t))$

$$\begin{aligned} &\int_A U_{D_1}^{(t)}(x, B)\mu_t(\mathrm{d}x) \\ &= c_t \int_A U_{D_1}^{(t)}(x, B)\|x\|^{1-d}\mathrm{d}x \\ &= c_t \int_0^\infty \int_{\mathbb{S}^{d-1}} \mathbb{1}_A(r\theta) U_{D_1}^{(t)}(r\theta, B)\sigma_d(\mathrm{d}\theta)\mathrm{d}r \\ &= c_t \int_0^\infty \frac{\int_{\mathbb{S}^{d-1}} \mathbb{1}_A(r\theta)\sigma_d(\mathrm{d}\theta) \int_{\mathbb{S}^{d-1}} \mathbb{1}_B(r\tilde{\theta})\sigma_d(\mathrm{d}\tilde{\theta})}{\int_{\mathbb{S}^{d-1}} \mathbb{1}_{L(t)}(r\tilde{\theta})\sigma_d(\mathrm{d}\tilde{\theta})}\mathrm{d}r. \end{aligned}$$

Similarly, again by Proposition A.1, we obtain

$$\begin{aligned} &\int_A U_R^{(t)}(x, B)\mu_t(\mathrm{d}x) \\ &= c_t \int_A U_R^{(t)}(x, B)\|x\|^{1-d}\mathrm{d}x \\ &= c_t \int_{\mathbb{S}^{d-1}} \int_0^\infty \mathbb{1}_A(r\theta) U_R^{(t)}(r\theta, B)\mathrm{d}r\sigma_d(\mathrm{d}\theta) \\ &= c_t \int_{\mathbb{S}^{d-1}} \frac{\int_0^\infty \mathbb{1}_A(r\theta)\mathrm{d}r \int_0^\infty \mathbb{1}_B(\tilde{r}\theta)\mathrm{d}\tilde{r}}{\int_0^\infty \mathbb{1}_{L(t)}(\tilde{r}\theta)\mathrm{d}\tilde{r}}\sigma_d(\mathrm{d}\theta). \end{aligned}$$

As the last expression in each of these two computations is symmetric in A and B, they show both $U_{D_1}^{(t)}$ and $U_R^{(t)}$ to be reversible w.r.t. μ_t. □

We also provide a suitable representation for the 2nd variant of GPSS.

Lemma B.3. *For $T_n = t$ the transition kernel $U_X^{(t)}$ on $(L(t), \mathcal{B}(L(t)))$ of the X-update of the 2nd variant of GPSS takes the form $U_X^{(t)} = U_{D_2}^{(t)} U_R^{(t)}$ with $U_R^{(t)}$ being specified as in (9) and a suitable[9], w.r.t. μ_t reversible, kernel $U_{D_2}^{(t)}$.*

Proof. We require some further notation: For $r \in \mathbb{R}_+$ and $\theta, \vartheta \in \mathbb{S}^{d-1}$ let $g^{(\theta,\vartheta)} \colon [0, 2\pi] \to \mathbb{S}^{d-1}$ be given by $g^{(\theta,\vartheta)}(\alpha) = \theta\cos(\alpha) + \vartheta\sin(\alpha)$ and define

$$L(t, r) := \{\theta \in \mathbb{S}^{d-1} \mid r\theta \in L(t)\}$$
$$L(t, r, \theta, \vartheta) := \{\alpha \in [0, 2\pi] \mid g^{(\theta,\vartheta)}(\alpha) \in L(t, r)\}.$$

Define the transition kernel $S^{(t,r)}$ on $L(t, r) \times \mathcal{B}(\mathbb{S}^{d-1})$ as

$$S^{(r,t)}(\theta, C) := \int_{\mathbb{S}_\theta^{d-2}} \int_{L(t,r,\theta,\vartheta)} \frac{\mathbb{1}_C(g^{(\theta,\vartheta)}(\alpha))\,\mathrm{d}\alpha}{\lambda_1(L(t, r, \theta, \vartheta))}\xi_\theta(\mathrm{d}\vartheta),$$

where $\theta \in L(t, r)$ and $C \in \mathcal{B}(\mathbb{S}^{d-1})$, with ξ_θ being the uniform distribution on $\mathbb{S}_\theta^{d-2}$. The transition kernel $S^{(r,t)}$ coincides with the transition kernel of the ideal geodesic slice sampler on the sphere that is reversible w.r.t. the uniform distribution on $L(t, r)$, see Lemma 14 in (Habeck et al.,

[9] We provide an explicit expression of $U_{D_2}^{(t)}$ in the proof of the statement.

2023). Denote the uniform distribution on $L(t,r)$ as $\gamma^{(r,t)}$, i.e.

$$\gamma^{(r,t)}(\mathrm{d}\theta) = \frac{\mathbb{1}_{L(t,r)}(\theta)\sigma_d(\mathrm{d}\theta)}{\sigma_d(L(t,r))}.$$

Now observe that the transition kernel of the direction update $U_{D_2}^{(t)}$ specified in the 2nd variant of GPSS for $x = r\theta \in L(t)$ with $r \in \mathbb{R}_+$ and $\theta \in \mathbb{S}^{d-1}$ is given by

$$U_{D_2}^{(t)}(r\theta, A) = S^{(r,t)}(\theta, A^{(r)}), \tag{11}$$

where for $A \in \mathcal{B}(\mathbb{R}^d)$ we denote

$$A^{(r)} := \{\vartheta \in \mathbb{S}^{d-1} \mid r\vartheta \in A\}.$$

Note that the radius update of the 2nd variant of GPSS coincides with the one of the 1st variant, i.e. $U_R^{(t)}$ is given by (9). Hence the total X-update takes the form $U_X^{(t)} = U_{D_2}^{(t)} U_R^{(t)}$. It remains to prove the reversibility of $U_{D_2}^{(t)}$ w.r.t. μ_t: For $A, B \in \mathcal{B}(L(t))$ we have with c_t as in (10) that

$$\begin{aligned}
&\int_A U_{D_2}^{(t)}(x,B)\mu_t(\mathrm{d}x) \\
&= c_t \int_A U_{D_2}^{(t)}(x,B)\|x\|^{1-d}\mathrm{d}x \\
&= c_t \int_0^\infty \int_{\mathbb{S}^{d-1}} \mathbb{1}_A(r\theta) U_{D_2}^{(t)}(r\theta, B)\sigma_d(\mathrm{d}\theta)\mathrm{d}r \\
&= c_t \int_0^\infty \int_{A^{(r)}} S^{(r,t)}(\theta, B^{(r)})\gamma^{(r,t)}(\mathrm{d}\theta)\cdot\sigma_d(L(t,r))\,\mathrm{d}r,
\end{aligned}$$

which is, by the reversibility of $S^{(r,t)}$ w.r.t. $\gamma^{(r,t)}$, symmetric in A and B. Consequently, $U_{D_2}^{(t)}$ is reversible w.r.t. μ_t and the claimed statement is proven. □

Theorem 3.1 is now an easy consequence.

Proof of Theorem 3.1. By Lemmas B.2 and B.3, given $T_n = t$, the transition kernel of the X-update of the 1st and 2nd variant of GPSS is $U_X^{(t)} := U_{D_i}^{(t)} U_R^{(t)}$ for $i = 1, 2$ respectively. By the reversibility of the individual kernels w.r.t. μ_t, see Lemma B.2 and Lemma B.3, we have

$$\mu_t U_X^{(t)} = \mu_t U_{D_i}^{(t)} U_R^{(t)} = \mu_t U_R^{(t)} = \mu_t,$$

proving that $U_X^{(t)}$ leaves μ_t for $i = 1, 2$ invariant. By Lemma B.1, this yields that the variants of GPSS leave the target distribution ν invariant. □

We add another auxiliary result.

Lemma B.4. *For $(t,\theta) \in (0,\infty)\times\mathbb{S}^{d-1}$ let*

$$L(t,\theta) := \{r \in \mathbb{R}_+ \mid r\theta \in L(t)\}.$$

Then $\lambda_1(L(t, y/\|y\|)) < \infty$ holds for $\lambda_1 \otimes \lambda_d$-almost all $(t,y) \in \mathbb{R}_+ \times \mathbb{R}^d$.

Proof. By Proposition A.1 follows

$$\begin{aligned}
\int_{\mathbb{R}^d} \varrho_\nu(x)\mathrm{d}x &= \int_{\mathbb{S}^{d-1}} \int_0^\infty \varrho_\nu^{(1)}(r\theta)\mathrm{d}r\sigma_d(\mathrm{d}\theta) \\
&= \int_0^\infty \int_{\mathbb{S}^{d-1}} \int_0^\infty \mathbb{1}_{L(t)}(r\theta)\mathrm{d}r\sigma_d(\mathrm{d}\theta)\mathrm{d}t \\
&= \int_0^\infty \int_{\mathbb{S}^{d-1}} \lambda_1(L(t,\theta))\sigma_d(\mathrm{d}\theta)\mathrm{d}t.
\end{aligned}$$

From this and the fact that ϱ_ν is integrable by assumption, it is immediate that $\lambda_1(L(t,\theta)) < \infty$ for $\lambda_1 \otimes \sigma_d$-almost all $(t,\theta) \in \mathbb{R}_+ \times \mathbb{S}^{d-1}$. By defining

$$F := \{(t,\theta) \in \mathbb{R}_+ \times \mathbb{S}^{d-1} \mid \lambda_1(L(t,\theta)) = \infty\},$$

we can alternatively express this result as $(\lambda_1 \otimes \sigma_d)(F) = 0$. With another application of Proposition A.1, this yields

$$\begin{aligned}
&(\lambda_1 \otimes \lambda_d)(\{(t,y) \in \mathbb{R}_+ \times \mathbb{R}^d \mid \lambda_1(L(t, y/\|y\|)) = \infty\}) \\
&= (\lambda_1 \otimes \lambda_d)(\{(t,y) \in \mathbb{R}_+ \times \mathbb{R}^d \mid (t, y/\|y\|) \in F\}) \\
&= \int_{\mathbb{R}^d} \int_0^\infty \mathbb{1}_F(t, y/\|y\|)\mathrm{d}t\mathrm{d}y \\
&= \int_0^\infty \int_{\mathbb{S}^{d-1}} \int_0^\infty \mathbb{1}_F(t,\theta)\mathrm{d}t\,\sigma_d(\mathrm{d}\theta) r^{d-1}\mathrm{d}r \\
&= \int_0^\infty (\lambda_1 \otimes \sigma_d)(F)\, r^{d-1}\mathrm{d}r = 0,
\end{aligned}$$

which proves the lemma's claim. □

Now we turn to the proof of Theorem 3.2.

Proof of Theorem 3.2. We use the same notation as in the proof of Theorem 3.1. Moreover, we know from the proof[10] of Theorem 15 in (Habeck et al., 2023) that there exists a constant $\varepsilon > 0$ (independent of t, r) such that

$$S^{(r,t)}(\theta, D) \geq \varepsilon\,\sigma_d(D \cap L(t,r)),$$

for any $\theta \in L(t,r)$ and $D \in \mathcal{B}(\mathbb{S}^{d-1})$. By (11) this implies for $x = r\theta \in L(t)$ and $B \in \mathcal{B}(L(t))$ that

$$\begin{aligned}
U_{D_2}^{(t)}(r\theta, B) &= S^{(r,t)}(\theta, B^{(r)}) \geq \varepsilon\,\sigma_d(B^{(r)} \cap L(t,r)) \\
&= \varepsilon \int_{\mathbb{S}^{d-1}} \mathbb{1}_{B\cap L(t)}(r\vartheta)\sigma_d(\mathrm{d}\vartheta) \\
&= \varepsilon \int_{\mathbb{S}^{d-1}} \mathbb{1}_{L(t)}(r\vartheta)\delta_{r\vartheta}(B)\sigma_d(\mathrm{d}\vartheta),
\end{aligned}$$

where δ_z denotes the Dirac-measure at $z \in \mathbb{R}^d$. Concisely written down, the former inequality yields

$$U_{D_2}^{(t)}(r\theta, \mathrm{d}y) \geq \varepsilon \int_{\mathbb{S}^{d-1}} \mathbb{1}_{L(t)}(r\vartheta)\delta_{r\vartheta}(\mathrm{d}y)\sigma_d(\mathrm{d}\vartheta).$$

[10] The theorem applies in their notation with $p(\theta) = \mathbb{1}_{L(t,r)}(\theta)$, $C = L(t,r)$ and $\beta = 1$, where also Remark 1 of (Habeck et al., 2023) should be taken into account.

For $\vartheta \in \mathbb{S}^{d-1}$ we use $L(t,\vartheta)$ as defined in Lemma B.4 and note that the normalizing constant within $U_R^{(t)}$ satisfies $\int_0^\infty \mathbb{1}_{L(t)}(s\vartheta)\mathrm{d}s = \lambda_1(L(t,\vartheta))$. Taking the representation of the X-update of the 2nd variant of GPSS into account we obtain (using the same variables as earlier)

$$\begin{aligned}
&U_X^{(t)}(r\theta, B) = U_{D_2}^{(t)} U_R^{(t)}(r\theta, B)\\
&= \int_{\mathbb{R}^d} U_R^{(t)}(y, B) U_{D_2}^{(t)}(r\theta, \mathrm{d}y)\\
&\geq \varepsilon \int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}^d} U_R^{(t)}(y, B) \mathbb{1}_{L(t)}(r\vartheta)\delta_{r\vartheta}(\mathrm{d}y)\sigma_d(\mathrm{d}\vartheta)\\
&= \varepsilon \int_{\mathbb{S}^{d-1}} U_R^{(t)}(r\vartheta, B)\mathbb{1}_{L(t)}(r\vartheta)\sigma_d(\mathrm{d}\vartheta)\\
&= \varepsilon \int_{\mathbb{S}^{d-1}} \int_0^\infty \frac{\mathbb{1}_B(\tilde{r}\vartheta)}{\lambda_1(L(t,\vartheta))}\mathbb{1}_{L(t)}(r\vartheta)\mathrm{d}\tilde{r}\,\sigma_d(\mathrm{d}\vartheta)\\
&= \varepsilon \int_B \frac{\mathbb{1}_{L(t)}(ry/\|y\|)}{\|y\|^{d-1}\lambda_1(L(t,y/\|y\|))}\mathrm{d}y,
\end{aligned}$$

where the last equality follows by Proposition A.1. Therefore the transition kernel Q corresponding to the 2nd variant of GPSS satisfies for any $A \in \mathcal{B}(\mathbb{R}^d)$ and $0 \neq x \in \mathbb{R}^d$ with $x = r\theta$ that

$$\begin{aligned}
Q(x,A) &= \frac{1}{\varrho_\nu^{(1)}(x)} \int_0^{\varrho_\nu^{(1)}(x)} U_X^{(t)}(x, A \cap L(t))\mathrm{d}t\\
&\geq \frac{\varepsilon}{\varrho_\nu^{(1)}(x)} \int_A \int_0^{\varrho_\nu^{(1)}(x)} \frac{\mathbb{1}_{L(t)}(ry/\|y\|)\mathbb{1}_{L(t)}(y)}{\|y\|^{d-1}\lambda_1(L(t,y/\|y\|))}\mathrm{d}t\,\mathrm{d}y\\
&= \frac{\varepsilon}{\varrho_\nu^{(1)}(x)} \int_A \int_0^{\min\{\varrho_\nu^{(1)}(x), \varrho_\nu^{(1)}(y), \varrho_\nu^{(1)}(ry/\|y\|)\}}\\
&\qquad \cdot \frac{\|y\|^{1-d}}{\lambda_1(L(t,y/\|y\|))}\mathrm{d}t\,\mathrm{d}y.
\end{aligned}$$

By Lemma B.4 we have that the mapping

$$(t,y) \mapsto \lambda_1(L(t,y/\|y\|))$$

is $\lambda_1 \otimes \lambda_d$-almost surely finite, so that we obtain that the function

$$(t,y) \mapsto \frac{\|y\|^{1-d}}{\lambda_1(L(t,y/\|y\|))}$$

is $\lambda_1 \otimes \lambda_d$-almost surely strictly larger than zero. By the assumption and definition of $\varrho_\nu^{(1)}$ we have that $\varrho_\nu^{(1)} > 0$ on $\mathbb{R}^d \setminus \{0\}$, such that $Q(x,A) > 0$ whenever $\lambda_d(A) > 0$. We apply the former implication: Let $A \in \mathcal{B}(\mathbb{R}^d)$ such that $\nu(A) > 0$. Then, by the absolute continuity of ν w.r.t. λ_d we obtain that $\lambda_d(A) > 0$ and thus $Q(x,A) > 0$ for any $0 \neq x \in \mathbb{R}^d$. This yields that the transition kernel of the 2nd variant of GPSS is ν-irreducible and aperiodic, as defined in Section 3 in (Tierney, 1994).

Since by Theorem 3.1 we also know that ν is an invariant distribution of Q, applying Theorem 1 of (Tierney, 1994) yields that ν is the unique invariant distribution. Moreover, the same theorem gives that the distribution of X_n converges ν-almost surely (regarding the initial state) to ν in the total variation distance. □

C. Implementation Notes

The code we provide not only allows for the reproduction of our experimental results, but also contains an easily usable, general purpose implementation of GPSS in Python 3.10, based on numpy.

Those still seeking to implement GPSS themselves, perhaps in another programming language, can pretty much follow Algorithms 1, 2 and 3. Still we find it appropriate to give two pieces of advice. First, the target density ϱ_ν, its transform $\varrho_\nu^{(1)}$ and the thresholds t_n should all be moved into log-space in order to make the implementation numerically stable. Second, upon generating a direction proposal, i.e. immediately following the code-adaptation of line 6 of Algorithm 2, the proposal should be re-normalized.

Although in theory the proposal should already be normalized, in practice the operations involved in generating it always introduce small numerical errors. Individually, these errors are far too small to matter, but, due to the way in which each direction variable depends on that from the previous iteration, without re-normalization the errors accumulate and lead the direction variables to be more and more unnormalized. If the sampler goes through many thousands of iterations, these accumulating errors eventually introduce large amounts of bias into the sampling, because the initial proposal set in the direction update is no longer a great circle but rather an elongated ellipse. Of course the re-normalization is not necessary if the direction variables are extracted from the full samples whenever required instead of being stored separately.

D. Additional Sampling Statistics

Table 1. Target density evaluations per iteration (TDE/I) and integrated autocorrelation times (IAT) for the experiments presented in Sections 6.1 and 6.2. The IATs were computed w.r.t. the sample log radii in the Cauchy experiment and w.r.t. the sample radii in the hyperplane experiment.

	CAUCHY		HYPERPLANE	
SAMPLER	TDE/I	IAT	TDE/I	IAT
GPSS	6.90	8.59	12.23	1.09
HRUSS	8.46	51346.93	7.78	684.57
NAIVE ESS	5.86	35543.94	7.91	199.17
TUNED ESS	–	–	6.51	188.13

E. Bayesian Logistic Regression

In this section we examine how well GPSS performs in a more classical machine learning setting, namely Bayesian logistic regression with a mean-zero Gaussian prior. We begin by giving a brief overview of the problem. Bayesian logistic regression is a binary regression problem. Supposing the training data to be given as pairs $(a^{(i)}, b^{(i)})$, $i = 1, ..., m$, where $a^{(i)} \in \mathbb{R}^d$ and $b^{(i)} \in \{-1, 1\}$, and choosing the prior distribution as $\mathcal{N}_d(0, 0.1^2 I_d)$, the posterior density of interest is given by

$$\varrho_\pi(x) = \mathcal{N}_d(x; 0, 0.1^2 I_d) \prod_{i=1}^{m} \frac{1}{1 + \exp\big(-b^{(i)} \langle a^{(i)}, x \rangle\big)},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $\mathbb{R}^d$.

We consider the *CoverType data set* (Blackard, 1998; Blackard et al.), which has cartographic variables as its features and forest cover types as its labels. The raw data set has $m = 581{,}012$ instances, 54 features and 7 different labels. As one of the labels occurs in about the same number of instances (283,301) as all the other labels combined (297,711), we make the regression problem binary by mapping this label to $+1$ and all other labels to -1. As the data set is quite large and we want to keep all of our experiments easily reproducible (in particular avoiding prohibitively large execution times), we use only 10% of it as training data and the remaining 90% as test data. Moreover, we normalized the data (i.e. each feature was affine-linearly transformed to have mean zero and standard deviation one) and added a constant feature (to enable a non-zero intercept of the regression line), rendering the problem $d = 55$ dimensional.

We then ran the usual three slice samplers and NUTS for $N = 5000$ iterations each, initializing them equally with a draw x_0 from the prior distribution $\mathcal{N}_d(0, 0.1^2 I_d)$. Viewing the first $N_b = 5000$ iterations as burn-in, we computed sample means for each sampler based only on the latter $N_a = 5000$ iterations and then used these sample means as predictors to compute training and testing accuracies. For reference, we also solved the problem[11] with the default solver of the classical python machine learning package `scikit-learn` (typically abbreviated sklearn).

As can be seen in Table 2, all five methods solved the logistic regression problem equally well, achieving virtually identical accuracies on both training and test data. Nevertheless, some differences between the samplers' behaviors become evident when considering the usual sampling metrics, see Table 3: On the one hand, GPSS used as many target density evaluations as HRUSS and ESS combined, on the other hand it also achieved more than twice the mean step size of either method (which should in principle help GPSS explore the target distribution's mode more thoroughly, though evidently this did not lead to it finding a better predictor than the other samplers, cf. Table 2). In terms of IAT, the three slice samplers perform relatively similarly[12]. Notably, NUTS, for which target density evaluations are neither available nor a sensible metric (due to the sampler's use of gradient information), vastly outperformed all three slice samplers in terms of both IAT and mean step size.

Table 2. Training and testing accuracies of the different solvers in the Bayesian logistic regression experiment.

SOLVER	TRAIN ACC	TEST ACC
GPSS	0.75520	0.75571
HRUSS	0.75472	0.75583
ESS	0.75506	0.75585
NUTS	0.75510	0.75573
SKLEARN	0.75562	0.75580

Table 3. Target density evaluations per iteration (TDE/I), integrated autocorrelation time (IAT) and mean step size (MSS) for the Bayesian logistic regression experiment. The IATs were computed w.r.t. the marginal samples consisting of the second entry of each sample. Whereas the TDE/I were computed based on the entire sampling runs, both IAT and MSS are only based on the samples generated after the burn-in period.

SAMPLER	TDE/I	IAT	MSS
GPSS	23.55	311.37	0.0268
HRUSS	9.12	179.30	0.0125
ESS	10.81	116.17	0.0115
NUTS	–	1.18	0.2942

Aside from classification accuracies, another important aspect to consider when using samplers for logistic regression is how long they need to reach the target distribution's mode from wherever they are initialized. For this we refer to Figures 9 and 10, where it can be seen that, at least in this case, GPSS and HRUSS converged towards the mode about equally fast, with the convergence of ESS seemingly being slightly slower. NUTS again worked vastly better than all slice samplers, converging almost instantaneously. This may seem remarkable at first glance, but is to be expected when considering that NUTS – unlike the slice samplers – relies on gradient information, which is extremely valuable in solving logistic regression tasks.

Overall, the results do not suggest Bayesian logistic regres-

[11] Note that sklearn technically solved a slightly different problem, because it only performed maximum likelihood without considering the prior. However, this did not appear to have a significant effect in practice, see Table 2.

[12] Note that the IAT values varied significantly between different runs of this experiment.

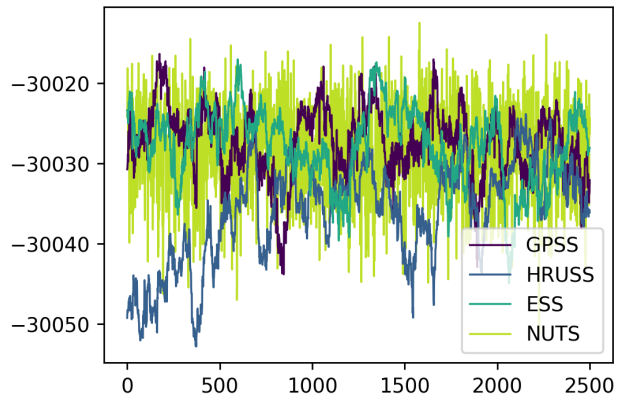

Figure 9. Log target density values for the Bayesian logistic regression experiment, over the course of each sampler's $N_a = 2500$ iterations after the burn-in period.

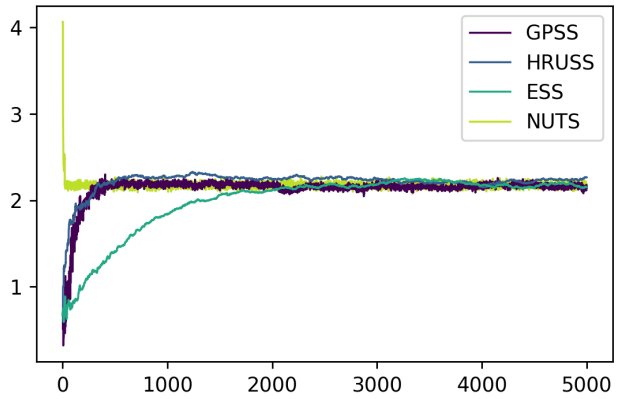

Figure 10. Sample radii (Euclidean norms) for the Bayesian logistic regression experiment over the course of each sampler's $N = 5000$ iterations.

sion to be a particularly worthwhile application of GPSS, which is not surprising to us based on our observations about the method's apparent strengths and weaknesses (cf. Sections 6 and 7).

F. Additional Results of Numerical Experiments

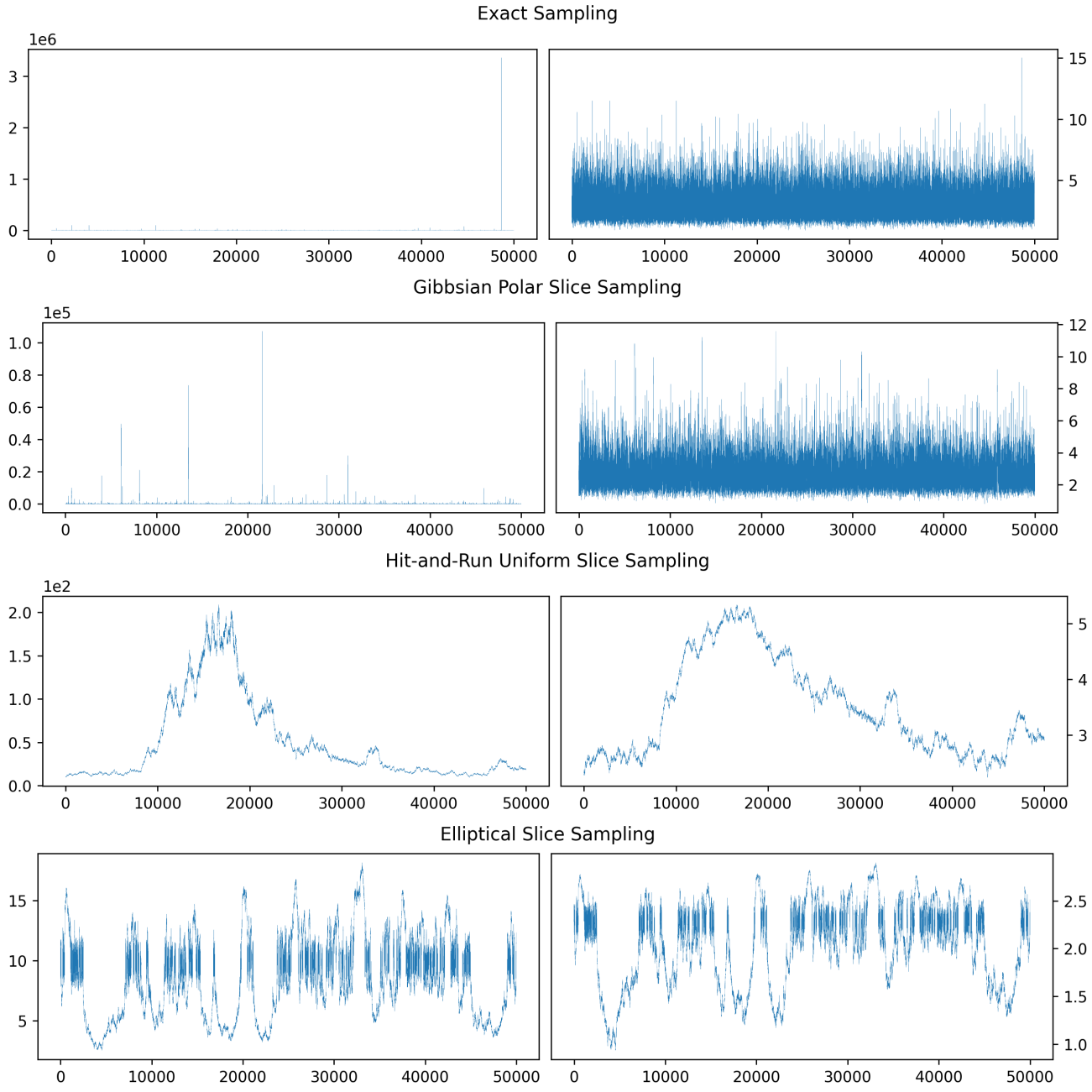

Figure 11. Sample runs for the multivariate standard Cauchy distribution, i.e. the density $\varrho_\nu(x) = (1 + \|x\|^2)^{-(d+1)/2}$, in dimension $d = 100$. The plots in the left column display the progression of the radii (Euclidean norms) of the individual samples over the course of $N = 5 \cdot 10^4$ iterations. Their counterparts on the right show the logarithms of these values. Note that we include the log radii plots to provide more insight into the short-term behavior of exact sampling and the GPSS chain, which can not be ascertained from the radii plots due to the magnitude of their respective outliers.

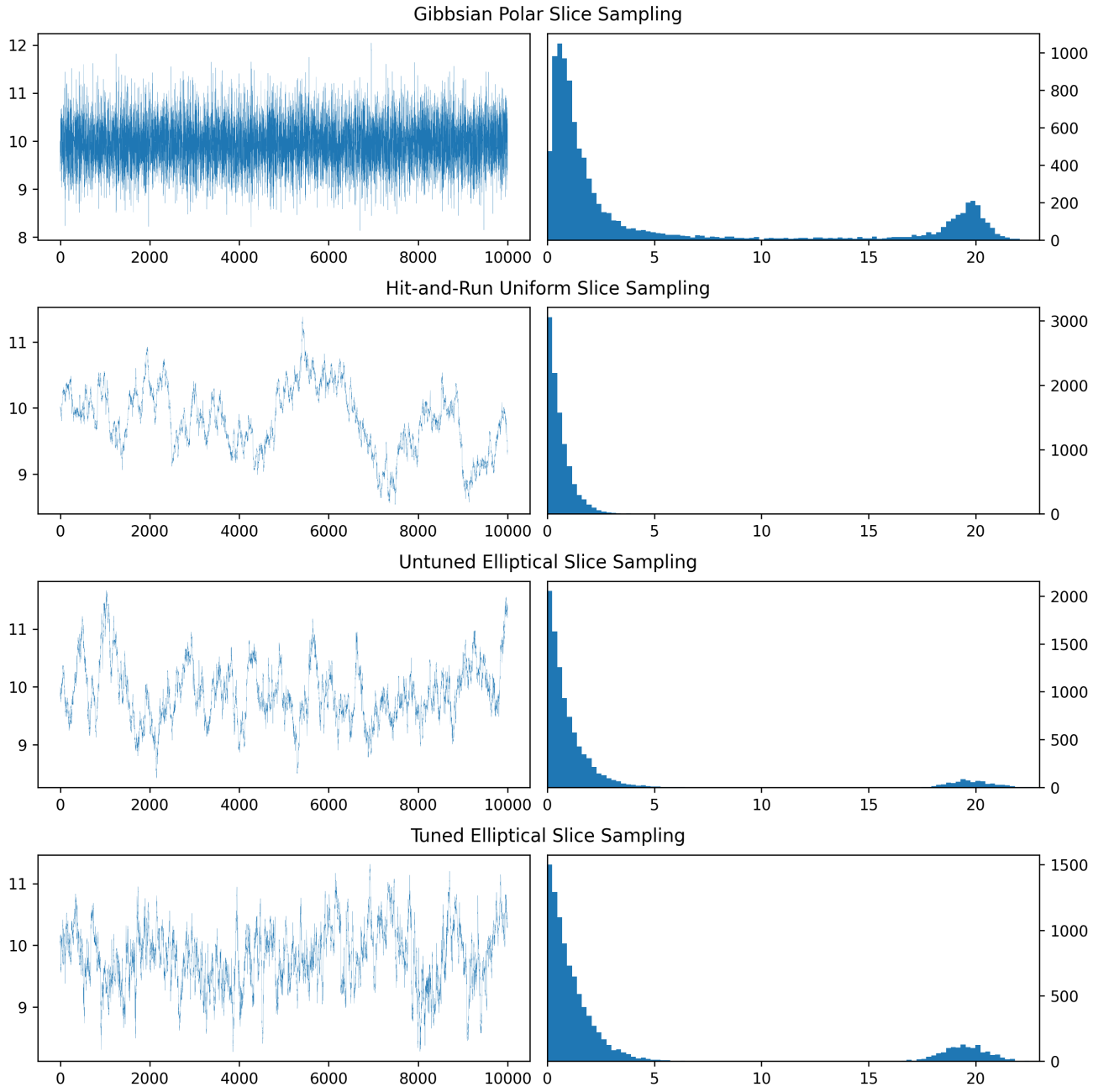

Figure 12. Sample runs and step size histograms for the hyperplane disk density (6) in dimension $d = 200$. The plots in the left column display the progression of the radii (Euclidean norms) of the individual samples over the course of $N = 10^4$ iterations. The plots in the right column show histograms of the Euclidean distances between each two consecutive samples. The descriptor *tuned ESS* refers to ESS where the artificial Gaussian prior is given the empirical covariance of the samples generated by GPSS as its covariance parameter Σ. Untuned ESS corresponds, as usual, to $\Sigma := I_d$.

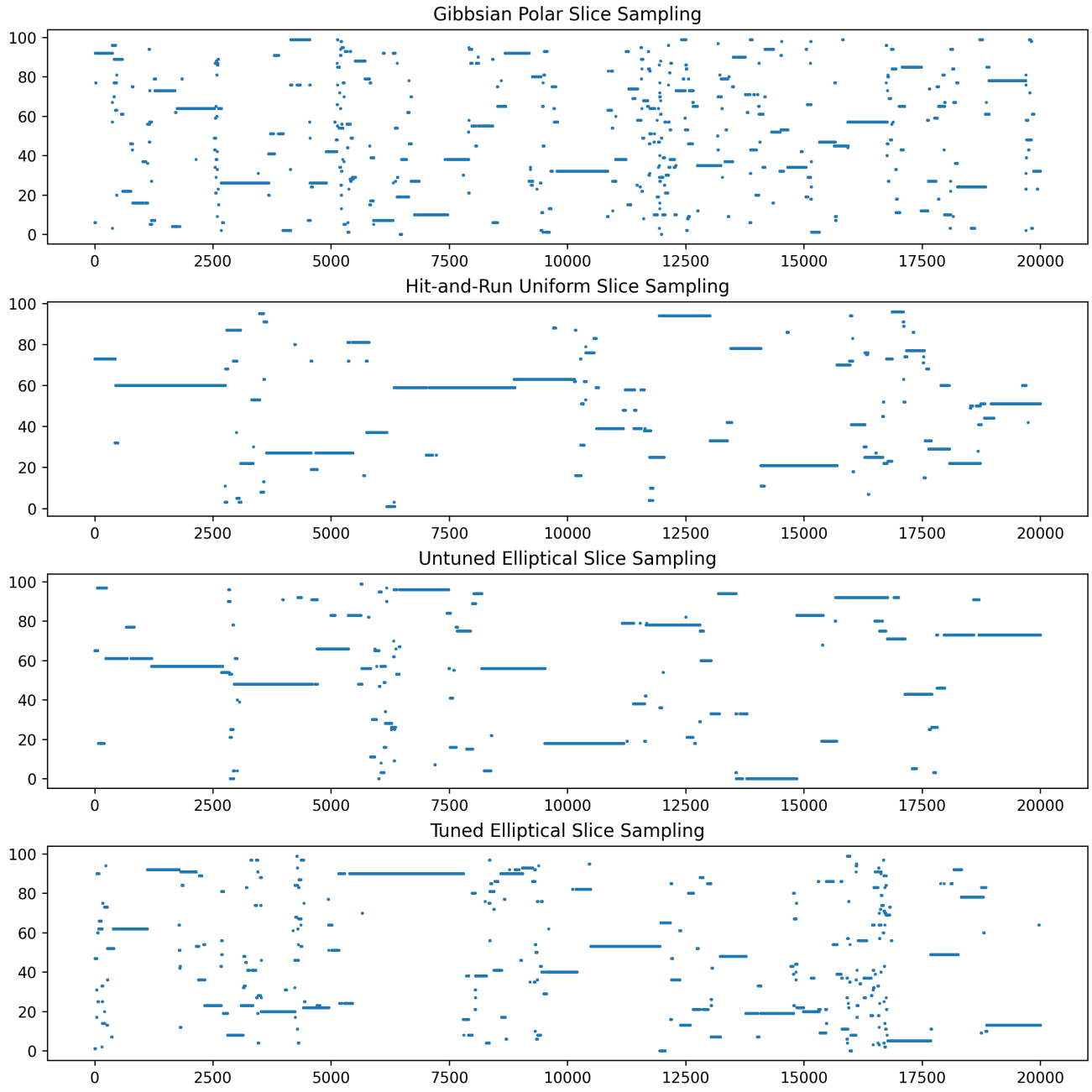

Figure 13. Progression of currently visited mode pair over the last $N_{\text{window}} = 2 \cdot 10^4$ iterations for the axial modes target density (7) in dimension $d = 100$. Here the covariance used by untuned ESS was $\Sigma = I_d$ and that used by tuned ESS $\Sigma = (5 + d/10)^2/d \cdot I_d$.

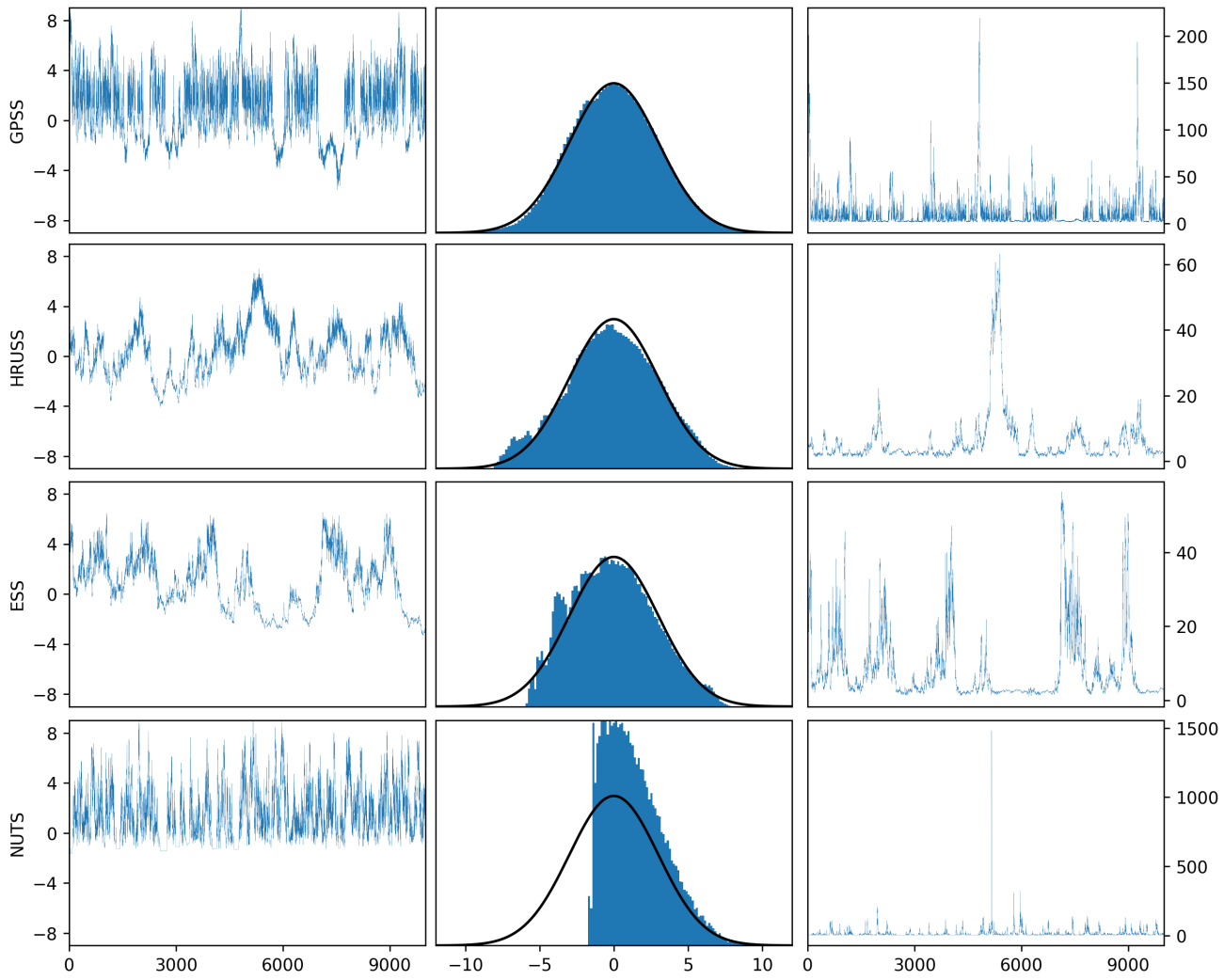

Figure 14. Marginal traces, marginal histograms and sample radii for Neal's funnel (8) in dimension $d = 10$. The plots in the left column show the progression of the first coordinate component of each sample over the course of each sampler's final $N_{\text{window}} = 10^4$ iterations. The plots in the middle column display histograms of the first coordinate component of all samples each sampler generated within the awarded time budget, with the thick black line marking the target marginal distribution. The plots in the right column show the progression of sample radii (Euclidean norms) over the course of each sampler's final $N_{\text{window}} = 10^4$ iterations. In particular, the quantities displayed in the left and right column of each row are derived from the same N_{window} samples.