MNRAS **000**, 1–28 (2019) Preprint 30 March 2019 Compiled using MNRAS LATEX style file v3.0

dynesty: A Dynamic Nested Sampling Package for Estimating Bayesian Posteriors and Evidences

Joshua S. Speagle[1,2]★
[1] *Center for Astrophysics | Harvard & Smithsonian, 60 Garden St., Cambridge, MA, USA*
[2] *NSF Graduate Research Fellow*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We present `dynesty`, a public, open-source, Python package to estimate Bayesian posteriors and evidences (marginal likelihoods) using Dynamic Nested Sampling. By adaptively allocating samples based on posterior structure, Dynamic Nested Sampling has the benefits of Markov Chain Monte Carlo algorithms that focus exclusively on posterior estimation while retaining Nested Sampling's ability to estimate evidences and sample from complex, multi-modal distributions. We provide an overview of Nested Sampling, its extension to Dynamic Nested Sampling, the algorithmic challenges involved in implementing them, and the various approaches taken to solve them. We then examine `dynesty`'s performance on a variety of toy problems along with several astronomical applications. We find in particular problems `dynesty` can provide substantial improvements in sampling efficiency compared to popular MCMC approaches in the astronomical literature. More detailed statistical results related to Nested Sampling are also included in the Appendix.

Key words: methods: statistical – methods: data analysis

1 INTRODUCTION

Much of modern astronomy rests on making inferences about underlying physical models from observational data. Since the advent of large-scale, all-sky surveys such as SDSS (York et al. 2000), the quality and quantity of these data increased substantially (Borne et al. 2009). In parallel, the amount of computational power to process these data also increased enormously. These changes opened up an entire new avenue for astronomers to try and learn about the universe using more complex models to answer increasingly sophisticated questions over large datasets. As a result, the standard statistical inference frameworks used in astronomy have generally shifted away from Frequentist methods such as maximum-likelihood estimation (MLE; Fisher 1922) to Bayesian approaches to estimate the distribution of possible parameters for a given model that are consistent with the data and our current astrophysical knowledge (see, e.g., Trotta 2008; Planck Collaboration et al. 2016; Feigelson 2017).

In the context of Bayesian inference, we are interested in estimating the posterior $P(\boldsymbol{\Theta}|\mathbf{D}, M)$ of a set of parameters $\boldsymbol{\Theta}$ for a given model M conditioned on some data $\mathbf{D}$. This can be written into a form commonly known as Bayes Rule to give

$$P(\boldsymbol{\Theta}|\mathbf{D}, M) = \frac{P(\mathbf{D}|\boldsymbol{\Theta}, M)P(\boldsymbol{\Theta}|M)}{P(\mathbf{D}|M)} \tag{1}$$

where $P(\mathbf{D}|\boldsymbol{\Theta}, M)$ is the likelihood of the data given the parameters of our model, $P(\boldsymbol{\Theta}|M)$ is the prior for the parameters of our model, and

$$P(\mathbf{D}|M) = \int_{\Omega_{\boldsymbol{\Theta}}} P(\mathbf{D}|\boldsymbol{\Theta}, M)P(\boldsymbol{\Theta}|M)\mathrm{d}\boldsymbol{\Theta} \tag{2}$$

is the evidence (i.e. marginal likelihood) for the data given our model, where the integral is taken over the entire domain $\Omega_{\boldsymbol{\Theta}}$ of $\boldsymbol{\Theta}$ (i.e. over all possible parameter combinations). Throughout the rest of the paper, we will refer to these using shorthand notation

$$\mathcal{P}(\boldsymbol{\Theta}_M) = \frac{\mathcal{L}(\boldsymbol{\Theta}_M)\pi(\boldsymbol{\Theta}_M)}{\mathcal{Z}_M} \tag{3}$$

where the subscript M will subsequently be dropped if we are only considering a single model. Here, the posterior $\mathcal{P}(\boldsymbol{\Theta}_M)$ tells us about the parameter estimates from a *given* model M while $\mathcal{Z}_M$ enables us to compare *across* models marginalized over any particular set of parameters using the Bayes factor:

$$R \equiv \frac{\mathcal{Z}_{M_1}}{\mathcal{Z}_{M_2}} \frac{\pi(M_1)}{\pi(M_2)} \tag{4}$$

where $\pi(M_i)$ is the prior belief in model M_i.

★ E-mail: jspeagle@cfa.harvard.edu

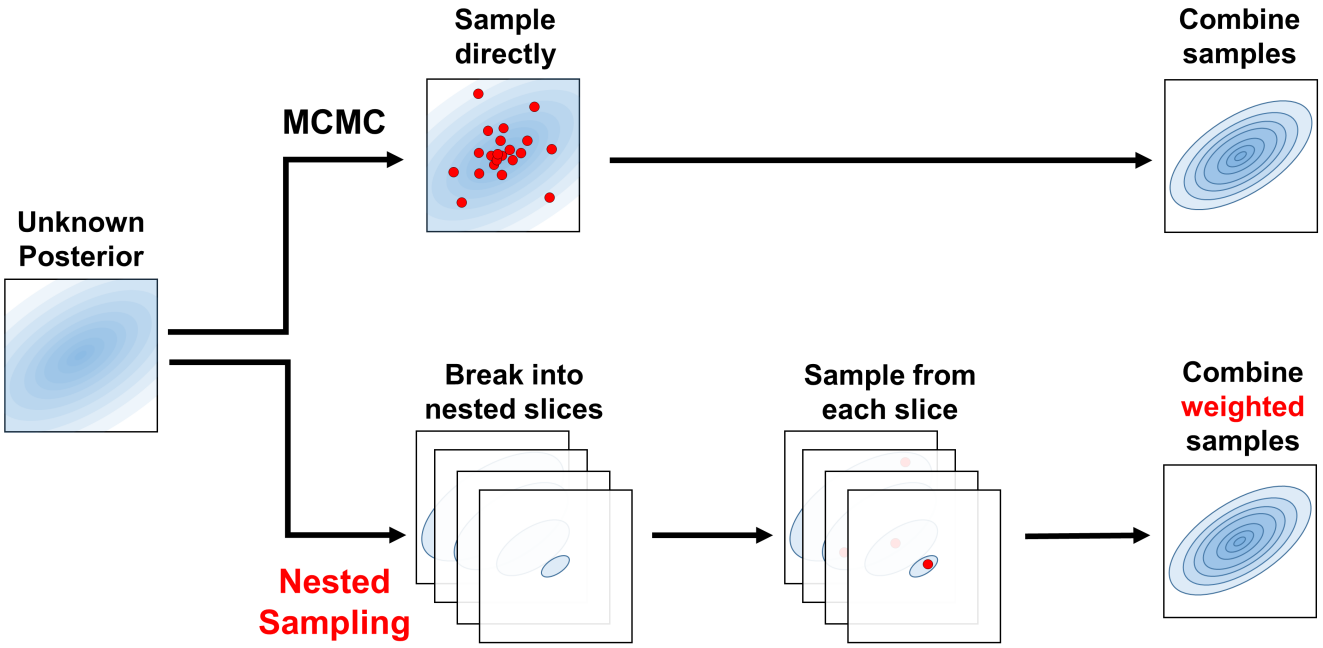

Figure 1. A schematic representation of the different approaches Markov Chain Monte Carlo (MCMC) methods and Nested Sampling methods take to sample from the posterior. While MCMC methods attempt to generate samples directly from the posterior, Nested Sampling instead breaks up the posterior into many nested "slices", generates samples from each of them, and then recombines the samples to reconstruct the original distribution using the appropriate weights.

For complicated data and models, the posterior $\mathcal{P}(\boldsymbol{\Theta})$ is often analytically intractable and must be estimated using numerical methods. These fall into two broad classes: "approximate" and "exact" approaches. Approximate approaches try to find an (analytic) distribution $\mathcal{Q}(\boldsymbol{\Theta})$ that is "close" to $\mathcal{P}(\boldsymbol{\Theta})$ using techniques such as Variational Inference (Blei et al. 2016). These techniques are not the focus of this work and will not be discussed further in this paper.

Exact approaches try to estimate $\mathcal{P}(\boldsymbol{\Theta})$ directly, often by constructing an algorithm that allows us to generate a set of samples $\{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_N\}$ that we can use to approximate the posterior as a weighted collection of discrete points

$$\mathcal{P}(\boldsymbol{\Theta}) \approx \hat{\mathcal{P}}(\boldsymbol{\Theta}) = \frac{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i)}{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)} \tag{5}$$

where $p(\boldsymbol{\Theta}_i)$ is the importance weight associated with each $\boldsymbol{\Theta}_i$ and $\delta(\boldsymbol{\Theta}_i)$ is the Dirac delta function located at $\boldsymbol{\Theta}_i$.

There is a rich literature (see, e.g., Chopin & Ridgway 2015) on the approaches used to generate these samples and their associated weights. The most popular method used in astronomy today is Markov Chain Monte Carlo (MCMC), which generates samples "proportional to" the posterior such that $p_i = 1$. While MCMC has had substantial success over the past few decades (Brooks et al. 2011; Sharma 2017), the most common implementations (e.g., Plummer 2003; Foreman-Mackey et al. 2013; Carpenter et al. 2017) tend to struggle when the posterior is comprised of widely-separated modes. In addition, because it only generates samples *proportional to* the posterior, it is difficult to use those samples to estimate the evidence $\mathcal{Z}_M$ to compare various models.

Nested Sampling (Skilling 2004; Skilling 2006) is an alternative approach to posterior and evidence estimation that tries to resolve some of these issues. By generating samples in nested (possibly disjoint) "shells" of increasing likelihood, it is able to estimate the evidence $\mathcal{Z}_M$ for distributions that are challenging for many MCMC methods to sample from. The final set of samples can also be combined with their associated importance weights p_i to generate associated estimates of the posterior.

Since a large portion of modern astronomy relies on being able to perform Bayesian inference, implementing these methods often can serve as the primary bottleneck for testing hypotheses, estimating parameters, and performing model comparisons. As such, packages that implement these approaches serve an important role enabling science by bridging the gap between writing down a model and estimating its associated parameters. These allow users to perform sophisticated analyses without having to implement many of the aforementioned algorithms themselves. Several prominent examples include the MCMC package `emcee` (Foreman-Mackey et al. 2013) and the Nested Sampling packages `MultiNest` (Feroz et al. 2009, 2013) and `PolyChord` (Handley et al. 2015), which collectively have been used in thousands of papers.

We present `dynesty`, a public, open-source, Python package that implements Dynamic Nested Sampling. `dynesty` is designed to be easy to use and highly modular, with extensive documentation, a straightforward application programming interface (API), and a variety of sampling implementations. It also contains a number of "quality of life" features including well-motivated stopping criteria, plotting functions, and analysis utilities for post-processing results.

The outline of the paper is as follows. In §2 we give an overview of Nested Sampling and discuss the method's ben-

efits and drawbacks. In §3 we describe how Dynamic Nested Sampling is able to resolve some of these drawbacks by allocating samples more flexibly. In §4 we discuss the specific approaches `dynesty` uses to track and sample from complex, multi-modal distributions. In §5 we examine `dynesty`'s performance on a variety of toy problems. In §6 we examine `dynesty`'s performance on several real-world astrophysical analyses. We conclude in §7. For interested readers, more detailed results on many of the methods outlined in the main text are included in Appendix A.

`dynesty` is publicly available on GitHub as well as on PyPI. See `https://dynesty.readthedocs.io` for installation instructions and examples on getting started.

2 NESTED SAMPLING

The general motivation for Nested Sampling, first proposed by Skilling (2004) and later fleshed out in Skilling (2006), stems from the fact that sampling from the posterior $\mathcal{P}(\boldsymbol{\Theta})$ directly is *hard*. Methods such as Markov Chain Monte Carlo (MCMC) attempt to tackle this single difficult problem *directly*. Nested Sampling, however, instead tries to break down this single hard problem into a larger number of *simpler* problems by:

(i) "slicing" the posterior into many simpler distributions,
(ii) sampling from each of those in turn, and
(iii) re-combining the results afterwards.

We provide a schematic illustration of this procedure in Figure 1 and give a broad overview of this process below. For additional details, please see Appendix A.

2.1 Overview

Unlike MCMC methods, which attempt to estimate the posterior $\mathcal{P}(\boldsymbol{\Theta})$ directly, Nested Sampling instead focuses on estimating the evidence

$$\mathcal{Z} \equiv \int_{\Omega_{\boldsymbol{\Theta}}} \mathcal{P}(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} = \int_{\Omega_{\boldsymbol{\Theta}}} \mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \tag{6}$$

As this integral is over the entire multi-dimensional domain of $\boldsymbol{\Theta}$, it is traditionally very challenging to estimate.

Nested Sampling tries to approach this problem by refactoring this integral as one taken over prior volume X of the enclosed parameter space

$$\mathcal{Z} = \int_{\Omega_{\boldsymbol{\Theta}}} \mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} = \int_0^1 \mathcal{L}(X)\mathrm{d}X \tag{7}$$

Here, $\mathcal{L}(X)$ now defines an iso-likelihood contour (or multiple) defining the edge(s) of the volume X, while the prior volume

$$X(\lambda) \equiv \int_{\Omega_{\boldsymbol{\Theta}}:\mathcal{L}(\boldsymbol{\Theta})\geq\lambda} \pi(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \tag{8}$$

is the fraction of the prior where the likelihood $\mathcal{L}(\boldsymbol{\Theta}) \geq \lambda$ is above some threshold λ. Since the prior is normalized, this gives $X(\lambda = 0) = 1$ and $X(\lambda = \infty) = 0$, which define the bounds of integration for equation (7).

As a rough analogy, we can consider trying to integrate over a spherically-symmetric distribution in 3-D. While it is possible to integrate over $\mathrm{d}x\mathrm{d}y\mathrm{d}z$ directly, it often is significantly easier to instead integrate over differential volume elements $\mathrm{d}V = 4\pi r^2$ as a function of radius $r \equiv \sqrt{x^2 + y^2 + z^2}$:

$$\int \mathcal{P}(x, y, z)\mathrm{d}x\mathrm{d}y\mathrm{d}z = \int \mathcal{P}(V(r))\mathrm{d}V(r) = \int \mathcal{P}(r)4\pi r^2\mathrm{d}r$$

Parameterizing the evidence integral this way allows Nested Sampling (in theory) to convert from a complicated D-dimensional integral over $\boldsymbol{\Theta}$ to a simple 1-D integral over X.

While it is straightforward to evaluate the likelihood at a given position $\mathcal{L}(\boldsymbol{\Theta})$, estimating the associated prior volume $X(\boldsymbol{\Theta})$ and its differential $dX(\boldsymbol{\Theta})$ is substantially more challenging. We can, however, generate *noisy estimates* of these quantities by employing the procedure described in Algorithm 1. We elaborate further on this procedure and how it works below.

2.2 Generating Samples

A core element of Nested Sampling is the ability to generate samples from the prior $\pi(\boldsymbol{\Theta})$ subject to a hard likelihood constraint λ. The most naive algorithm that satisfies this constraint is simple rejection sampling: at a given iteration i, generate samples $\boldsymbol{\Theta}_{i+1}$ from the prior $\pi(\boldsymbol{\Theta})$ until $\mathcal{L}(\boldsymbol{\Theta}_{i+1}) \geq \mathcal{L}(\boldsymbol{\Theta}_i)$.

In practice, however, this simple procedure becomes progressively less efficient as time goes on since the remaining prior volume X_{i+1} at each iteration of Algorithm 1 keeps shrinking. We therefore need a way of directly generating samples from the constrained prior:

$$\pi_\lambda(\boldsymbol{\Theta}) \equiv \begin{cases} \pi(\boldsymbol{\Theta})/X(\lambda) & \mathcal{L}(\boldsymbol{\Theta}) \geq \lambda \\ 0 & \mathcal{L}(\boldsymbol{\Theta}) < \lambda \end{cases} \tag{9}$$

Sampling from this constrained distribution is difficult for an arbitrary prior $\pi(\boldsymbol{\Theta})$ since the density can vary drastically from place to place. It is drastically simpler, however, if the prior is standard uniform (i.e. flat from 0 to 1) in all dimensions so that the density interior to λ is constant then X behaves more like a typical volume V. We can accomplish this through the use of the appropriate "prior transform" function $\mathcal{T}$ which maps a set of parameters $\boldsymbol{\Phi}$ with a uniform prior over the D-dimensional "unit cube" to the parameters of interest $\boldsymbol{\Theta}$.[1] Taken together, these transform our original hard problem of sampling from the posterior $\mathcal{P}(\boldsymbol{\Theta})$ directly to instead the much simpler problem of repeatedly sampling uniformly[2] within the transformed constrained prior

$$\pi'_\lambda(\boldsymbol{\Phi}) \equiv \begin{cases} 1/X(\lambda) & \mathcal{L}(\boldsymbol{\Theta} = \mathcal{T}(\boldsymbol{\Phi})) \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Throughout the rest of the text we will henceforth assume $\pi(\boldsymbol{\Theta})$ is a unit cube prior unless otherwise explicitly specified.

Because there is no constraint that this distribution is uni-modal, the constrained prior may define several "blobs" of prior volume that we are interested in sampling from.

[1] In general, there is a uniquely defined prior transform $\mathcal{T}$ for any given $\pi(\boldsymbol{\Theta})$; see the `dynesty` documentation for additional details.
[2] Technically this requirement is overly strict, as Nested Sampling can still be valid even if the samples at each iteration are correlated. See Appendix A for additional discussion.

Algorithm 1: Static Nested Sampling

```
// Initialize live points.
Draw K "live" points {Θ_1, ..., Θ_K} from the prior π(Θ).
// Main sampling loop.
while stopping criterion not met do
    Compute the minimum likelihood L^min among the current set of live points.
    Add the kth live point Θ_k associated with L^min to a list of "dead" points.
    Sample a new point Θ' from the prior subject to the constraint L(Θ') ≥ L^min.
    Replace Θ_k with Θ'.
    // Check whether to stop.
    Evaluate stopping criterion.
end
// Add final live points.
while K > 0 do
    Compute the minimum likelihood L^min among the current set of live points.
    Add the kth live point Θ_k associated with L^min to a list of "dead" points.
    Remove Θ_k from the set of live points.
    Set K = K − 1.
end
```

While sampling from the blob(s) might be hard to do from scratch, because Nested Sampling samples at many different likelihood "levels", structure tends to emerge over time rather than all at once as we transition away from the prior $\pi(\boldsymbol{\Theta})$.

2.3 Estimating the Prior Volume

As shown in Appendix A, generating samples following the strategy in §2.2 based on Algorithm 1 allows us to estimate the (change in) prior volume at a given iteration using the set of "dead" points (i.e. the live points we replaced at each iteration). In particular, it leads to *exponential shrinkage* such that the (log-)prior volume at each iteration changes by

$$\mathbb{E}\left[\Delta \ln \hat{X}_i\right] = \mathbb{E}\left[\ln \hat{X}_i - \ln \hat{X}_{i-1}\right] = -\frac{1}{K} \tag{11}$$

where $\mathbb{E}[\cdot]$ is the expectation value (i.e. mean) and we have adopted the $\hat{x}$ notation to emphasize that we have a noisy estimator of the prior volume X. Using more live points K thus increases our volume resolution by decreasing the rate of this exponential compression. By default, `dynesty` uses $K = 500$ live points, although this should be adjusted depending on the problem at hand.

Once some stopping criterion is reached and sampling terminates after N iterations, the remaining set of K live points are then distributed uniformly within the final prior volume X_N (see Appendix A). These can be "recycled" into the final set of samples by sequentially adding the live points to the list of "dead" points collected at each iteration in order of increasing likelihood. This leads to *uniform shrinkage* of the prior volume such that the (fractional) change in prior volume for the kth live point added this way is

$$\mathbb{E}\left[\frac{\Delta \hat{X}_{N+k}}{\hat{X}_N}\right] = \mathbb{E}\left[\frac{\hat{X}_{N+k} - \hat{X}_{N+k-1}}{\hat{X}_N}\right] = \frac{1}{K+1} \tag{12}$$

where $\hat{X}_N$ is the estimating remaining prior volume at the final Nth iteration.

2.4 Stopping Criterion

Since Nested Sampling is designed to estimate the evidence, a natural stopping criterion (see, e.g., Skilling 2006; Keeton 2011; Higson et al. 2017a) is to terminate sampling when we believe our set of dead points (and remaining live points) give us an integral that encompasses the vast majority of the posterior. In other words, at a given iteration i, we want to terminate sampling if

$$\Delta \ln \hat{\mathcal{Z}}_i \equiv \ln\left(\hat{\mathcal{Z}}_i + \Delta\hat{\mathcal{Z}}_i\right) - \ln\left(\hat{\mathcal{Z}}_i\right) < \epsilon \tag{13}$$

where $\Delta\hat{\mathcal{Z}}_i$ is the estimated remaining evidence we have yet to integrate over and ϵ determines the tolerance. If the final set of live points are excluded from the set of dead points, `dynesty` assumes a default value of $\epsilon = 10^{-2}$ (i.e. $\lesssim 1\%$ of the evidence remaining). If the final set of live points are included, `dynesty` instead uses the slightly more permissive $\epsilon = 10^{-3}(K-1) + 10^{-2}$.

While the remaining evidence $\Delta\hat{\mathcal{Z}}_i$ is unknown, we can in theory construct a strict upper bound on it by assigning

$$\Delta\hat{\mathcal{Z}}_i \leq \mathcal{L}^{\max} X_i \tag{14}$$

where $\mathcal{L}^{\max}$ is the maximum-likelihood value across the entire domain $\boldsymbol{\Omega}_{\boldsymbol{\Theta}}$ and X_i is the prior volume at the current iteration. This is equivalent to treating the remaining likelihood interior to the current sample ($X < X_i$) as a uniform slab with amplitude $\mathcal{L}^{\max}$.

Unfortunately, neither $\mathcal{L}^{\max}$ or X_i is known exactly. However, we can approximate this upper bound by replacing both quantities with associated estimators to get the *rough* upper bound

$$\Delta\hat{\mathcal{Z}}_i \lesssim \mathcal{L}_i^{\max} \hat{X}_i \tag{15}$$

where $\mathcal{L}_i^{\max}$ is the maximum value of the likelihood among the live points at iteration i and $\hat{X}_i$ is the estimated (remaining) prior volume.

While this rough upper bound works well in most cases, because we only have access to the best likelihood $\mathcal{L}_i^{\max}$ sampled by the K live points at a particular iteration there is always a chance that $\mathcal{L}_i^{\max} \ll \mathcal{L}^{\max}$ and that we will terminate early. This can happen if there is an extremely narrow

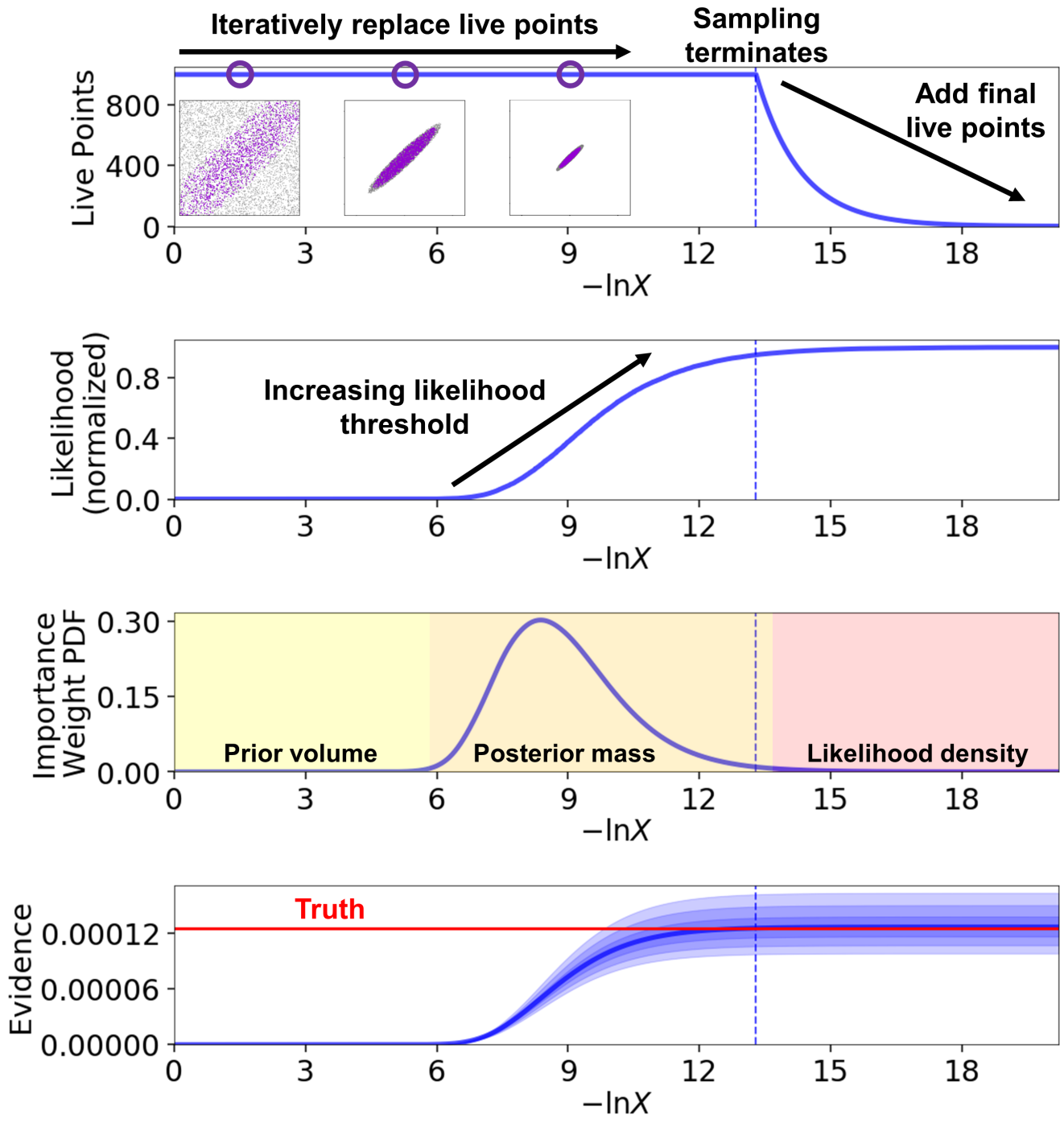

Figure 2. An example highlighting the behavior of a Static Nested Sampling run in `dynesty`. See §2 for additional details. *Top:* The number of live points as a function of prior volume X. Snapshots of their their distribution (purple) with respect to the current bounds (gray; see §4.1) are highlighted in several insets. The number of live points remains constant until sampling terminates, at which point we add the final live points one-by-one to the samples. *Top-middle:* The (normalized) likelihood limit $\mathcal{L}/\mathcal{L}^{\max}$ associated with a the prior volume $X(\mathcal{L})$ in the top panel. This increases monotonically as we sample increasingly smaller regions of the prior. *Bottom-middle:* The importance weight PDF $p(X)$, roughly divided into regions dominated by the prior volume ($\mathrm{d}X$ is large, $\mathcal{L}(X)$ is small; yellow), posterior mass ($\mathrm{d}X$ and $\mathcal{L}(X)$ are comparable; orange), and likelihood density ($\mathrm{d}X$ is small, $\mathcal{L}(X)$ is large; red). The posterior mass is the most important for posterior estimation, while evidence estimation also depends on the prior volume. *Bottom:* The estimated evidence $\hat{\mathcal{Z}}(X)$ (blue line) and its 1, 2, and 3-sigma errors (blue shaded). The true value is shown in red.

likelihood peak within the remaining prior volume that has not yet been discovered by the K live points.

2.5 Estimating the Evidence and Posterior

Once we have a final set of samples $\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_N\}$, we can estimate the 1-D evidence integral using standard numerical techniques. To ensure approximation errors on the numerical integration estimate are sufficiently small, `dynesty` uses the 2nd-order trapezoid rule

$$\hat{\mathcal{Z}} = \sum_{i=1}^{N+K} \frac{1}{2}\left[\mathcal{L}(\boldsymbol{\Theta}_{i-1}) + \mathcal{L}(\boldsymbol{\Theta}_i)\right] \times \left[\hat{X}_{i-1} - \hat{X}_i\right] \equiv \sum_{i=1}^{N+K} \hat{p}_i \tag{16}$$

where $X_0 = X(\lambda = 0) = 1$ and p_i is the estimated importance weight. By default, `dynesty` uses the mean values of $\hat{X}_i$ to compute the mean and standard deviation of $\ln \hat{\mathcal{Z}}$ following Appendix A, although these values can also be simulated explicitly.

We can also estimate the posterior $\mathcal{P}(\boldsymbol{\Theta})$ from the same set of $N+K$ dead points by using the associated importance weights derived above:

$$\hat{\mathcal{P}}(\boldsymbol{\Theta}) = \frac{\sum_{i=1}^{N+K} \hat{p}(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i)}{\sum_{i=1}^{N+K} \hat{p}(\boldsymbol{\Theta}_i)} = \hat{\mathcal{Z}}^{-1} \sum_{i=1}^{N+K} \hat{p}(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i) \tag{17}$$

By default, `dynesty` uses the mean values of $\hat{X}_i$ to compute this posterior estimate.

An illustration of a typical Nested Sampling run is shown in Figure 2.

2.6 Benefits of Nested Sampling

Because of its alternative approach to sampling from the posterior, Nested Sampling has a number of benefits relative to traditional MCMC approaches:

(i) Nested Sampling can estimate the evidence $\mathcal{Z}$ as well as the posterior $\mathcal{P}(\boldsymbol{\Theta})$. MCMC methods generally can only constrain the latter (although see Lartillot & Philippe 2006; Heavens et al. 2017).

(ii) Nested sampling can sample from multi-modal distributions that tend to challenge many MCMC methods.

(iii) While most MCMC stopping criteria based on effective sample sizes can feel arbitrary, Nested Sampling possesses well-motivated stopping criteria focused on evidence estimation.

(iv) MCMC methods need to converge (i.e. "burn in") to the posterior before any samples generated are valid. While optimization techniques can speed up this process, assessing this convergence can be challenging and time-consuming (Gelman & Rubin 1992; Vehtari et al. 2019). Nested Sampling doesn't suffer from similar issues because the method smoothly integrates over the posterior $\mathcal{P}(\boldsymbol{\Theta})$ starting from the prior $\pi(\boldsymbol{\Theta})$.

2.7 Drawbacks

While Nested Sampling has its fair share of benefits that have encouraged its rapid adoption in astronomical Bayesian analyses, it also suffers from a fair share of drawbacks. Most crucially, the standard Nested Sampling implementation outlined in Algorithm 1 focuses *exclusively* on estimating the evidence $\mathcal{Z}$; the posterior $\mathcal{P}(\boldsymbol{\Theta})$ is entirely a by-product of the approach. This creates several immediate drawbacks relative to MCMC, which focuses exclusively on sampling the posterior $\mathcal{P}(\boldsymbol{\Theta})$.

First, because Nested Sampling relies on generating live points from the prior $\pi(\boldsymbol{\Theta})$, the priors *must* be "proper" (i.e. integrable on their own). This disallows the use of "improper" priors that are often used in MCMC analyses, such as a flat prior over $(-\infty, +\infty)$. While the role of improper priors in Bayesian inference is hotly debated (see, e.g., Tak et al. 2018), the inability to use them at all (even if the evidence is still defined) is clearly a drawback.

Second, because most Nested Sampling implementations rely on sampling from uniform distributions (see §2.2), applying them to general distributions requires knowing the appropriate prior transform $\mathcal{T}$. While these are straightforward to define when the prior can be decomposed into separable, independent components, they can be more difficult to derive when the prior involves conditional and/or jointly distributed parameters.

Third, because the evidence depends on the amount of prior volume that needs to be integrated over, the overall expected runtime is sensitive to the relative size of the prior. In other words, while estimating the posterior mostly depends on generating samples close to where the majority of the distribution is located (i.e. the "typical set"; Betancourt 2017), estimating the evidence requires generating samples in the extended tails of the distribution. Using less informative (broader) priors will increase the expected runtime even if the posterior is largely unchanged.

Finally, because the number of live points K is constant, the rate $\Delta \ln X$ at which we integrate over the posterior $\mathcal{P}(\boldsymbol{\Theta})$ is the same regardless of where we are. This means that increasing the number of like points K, which increases the overall runtime, always improves the accuracy of *both* the posterior $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ and evidence $\hat{\mathcal{Z}}$ estimates. In other words, Nested Sampling does not allow users to *prioritize* between estimating the posterior or the evidence, which is not ideal for many analyses that are mostly interested in using Nested Sampling for either option. We focus on improving this behavior in §3.

As with any sampling method, we strongly advocate that Nested Sampling *should not* be viewed as being strictly "better" or "worse" than MCMC, but rather as a tool that can be more or less useful in certain problems. There is no "One True Method to Rule Them All", even though it can be tempting to look for one.

3 DYNAMIC NESTED SAMPLING

In our overview of Nested Sampling in §2, we highlighted four main drawbacks of basic implementations:

(i) They require integrable (proper) priors.
(ii) They generally require a prior transform.
(iii) Their runtime is sensitive to the size of the prior.
(iv) Their rate of posterior integration is always constant.

While the first three drawbacks are essentially inherent to

Algorithm 2: Dynamic Nested Sampling

```
// Initialize first set of live points.
Draw K "live" points {Θ_1, ..., Θ_K} from the prior π(Θ).
// Main sampling loop.
Set L^min = 0 and K_0 = K.
while stopping criterion not met do
    // Get current number of live points.
    Compute the previous number of live points K and the current number of live points K'.
    if K' ≥ K then
        // Add in new live points.
        while K' > K do
            Sample a new point Θ' from the prior subject to the constraint L(Θ') ≥ L^min.
            Add Θ' to the set of live points.
            Set K = K + 1.
        end
        // Replace worst live point.
        Compute the minimum likelihood L^min among the current set of K live points.
        Add the kth live point Θ_k associated with L^min to a list of "dead" points.
        Replace Θ_k with Θ'.
    else
        // Iteratively remove live points.
        while K' < K do
            Compute the minimum likelihood L^min among the current set of K = K' live points.
            Add the kth live point Θ_k associated with L^min to a list of "dead" points.
            Remove Θ_k from the set of live points.
            Set K = K − 1.
        end
    end
    // Check whether to stop.
    Evaluate stopping criterion.
end
// Add final live points.
while there are live points remaining do
    Compute the minimum likelihood L^min among the current set of live points.
    Add the kth live point Θ_k associated with L^min to a list of "dead" points.
    Remove Θ_k from the set of live points.
end
```

Nested Sampling as sampling strategy, the fourth is not. Instead, the inability of Algorithm 1 to "prioritze" estimating the evidence $\mathcal{Z}$ or posterior $\mathcal{P}(\boldsymbol{\Theta})$ is a consequence of the fact that the number of live points K remains constant throughout an entire run, which sets the rate of integration $\Delta \ln X$. As a result, we will henceforth call this procedure "Static" Nested Sampling.

To address this issue, Higson et al. (2017b) proposed a deceptively simple modification: let the number of live points *vary* during runtime. This gives a new "Dynamic" Nested Sampling algorithm whose basic implementation is outlined in Algorithm 2. This simple change is transformative, allowing Dynamic Nested Sampling to focus on sampling the posterior $\mathcal{P}(\boldsymbol{\Theta})$, similar to MCMC approaches, while retaining all the benefits of (Static) Nested Sampling to estimate the evidence $\mathcal{Z}$ and sample from complex, multi-modal distributions. It also possesses well-motivated new stopping criteria for posterior and evidence estimation.

We provide an illustration of the overall approach in Figure 3 and give a broad overview of the basic algorithm below. For additional details, please see Appendix A.

3.1 Allocating Live Points

The singular defining feature of the Dynamic Nested Sampling algorithm is the scheme we use for determining how the number of live points K_i at a given iteration i should vary. Naively, we would like K_i to be larger where we want our resolution to be higher (i.e. a slower rate of integration $\Delta \ln X_i$) and smaller where we are interested in traversing the current region of prior volume more quickly. This allows us to prioritize adding samples in regions of interest.

In general, we would like the number of live points $K(X)$ as a function of prior volume X to follow a particular importance function $\mathcal{I}(X)$ such that

$$K(X) \propto \mathcal{I}(X) \tag{18}$$

While this function can be completely general, since most users are interested in estimating the posterior $\mathcal{P}(\boldsymbol{\Theta})$ and/or evidence $\mathcal{Z}$ more generally, `dynesty` by default follows Higson et al. (2017b) and considers a function of the form:

$$\mathcal{I}(X) = f^{\mathcal{P}}\mathcal{I}^{\mathcal{P}}(X) + (1 - f^{\mathcal{P}})\mathcal{I}^{\mathcal{Z}}(X) \tag{19}$$

where $f^{\mathcal{P}}$ is the relative amount of importance placed on estimating the posterior.

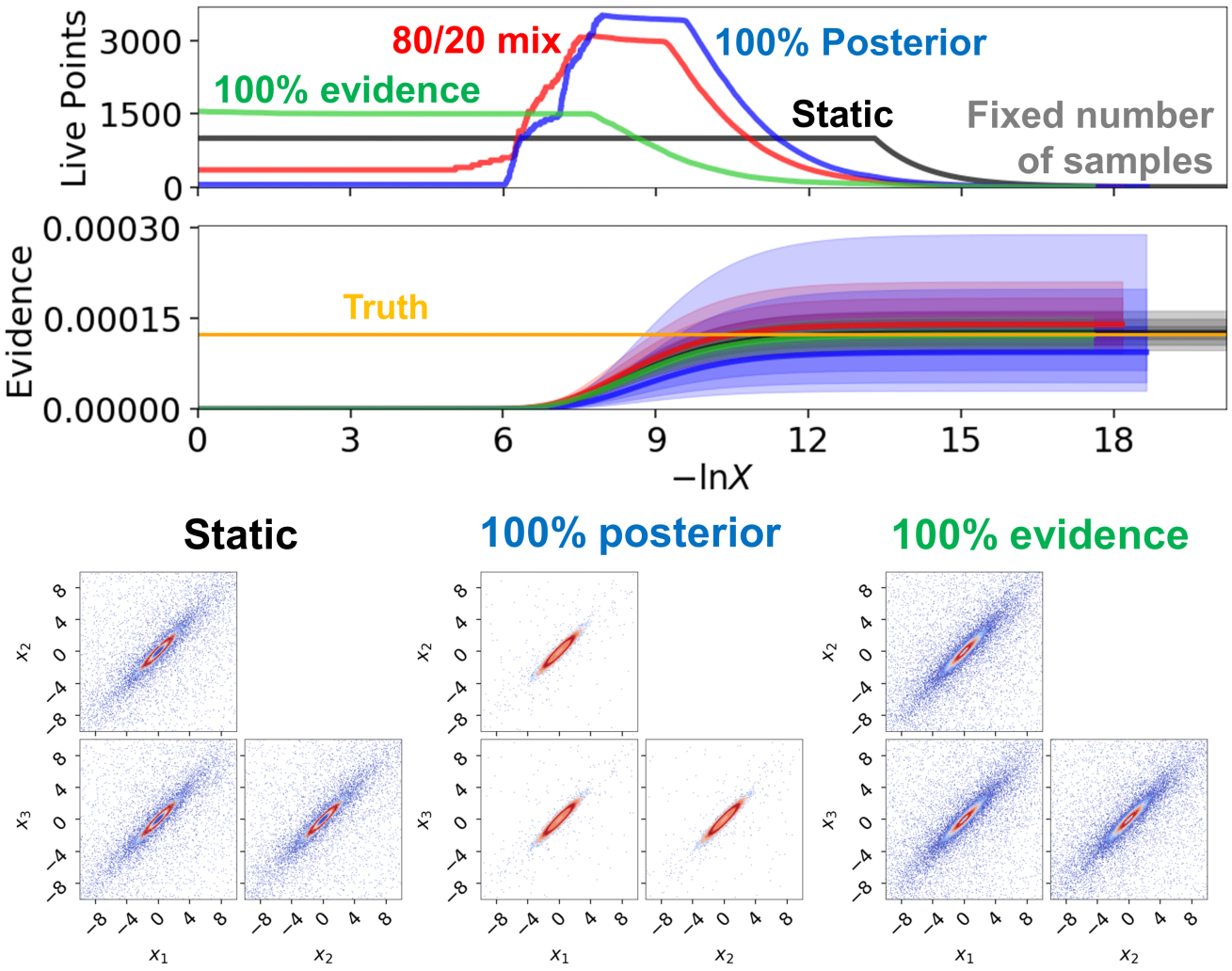

Figure 3. An example highlighting different schemes for live point allocation between Static and Dynamic Nested Sampling run in `dynesty` *with a fixed number of samples*. See §3 for additional details. *Top panels:* As Figure 2, but now highlighting the number of live points (upper) and evidence estimates (lower) for a Static Nested Sampling run (black) and Dynamic Nested Sampling runs focused entirely on estimating the posterior (blue), entirely on estimating the evidence (green), and with an 80%/20% posterior/evidence mixture (the default in `dynesty`; red). *Bottom panels:* The distribution of samples from the targeted 3-D correlated Gaussian distribution in the Static (left), posterior-focused (middle), and evidence-focused (right) runs. Points are color-coded based on their important weight p_i. The posterior-oriented run allocates points almost exclusively around the bulk of the posterior mass, while the evidence-oriented run preferentially allocates them in prior-dominated regions.

We define the posterior importance function as

$$\mathcal{I}^{\mathcal{P}}(X) \equiv p(X) \tag{20}$$

where $p(X)$ is the now the probability density function (PDF) of the importance weight defined in §2.5. This choice just means that we want to allocate more live points in regions where the posterior mass $\propto \mathcal{L}(X)\mathrm{d}X$ is higher.

We define the evidence importance function as

$$\mathcal{I}^{\mathcal{Z}}(X) \equiv \frac{1-\mathcal{Z}(X)/\mathcal{Z}}{\int_0^1 (1-\mathcal{Z}(X)/\mathcal{Z})\mathrm{d}X} \tag{21}$$

where $\mathcal{Z}(X)$ is the evidence integrated *up to* X. This means that we want to allocate more live points when we believe we have not integrated over much of the posterior (i.e. in the prior volume-dominated regime at larger values of X) and fewer as we integrate over larger portions of the posterior mass and become more confident in our estimated value of $\mathcal{Z}$ (see Figure 2).

3.2 Iterative Dynamic Nested Sampling

As in §2.4, we unfortunately do not have access to X or $\mathcal{I}(X)$ directly. We thus need to use noisy estimators to approximate them, which are only available *after we have already generated samples from the posterior*. In practice then, Dynamic Nested Sampling works as an iterative modification to Static Nested Sampling. We outline this "Iterative" Dynamic Nested Sampling approach, first proposed in Higson

Algorithm 3: Iterative Dynamic Nested Sampling

`// Baseline Nested Sampling run.`
Run Static Nested Sampling (Algorithm 1) with:
(a) K live points
(b) sampled uniformly from the prior $\pi(\boldsymbol{\Theta})$
(c) until the default Static Nested Sampling stopping criterion is met.
`// Main sampling loop.`
while stopping criterion not met **do**
  `// Find region where new samples should be allocated.`
  Compute relative importance $\{\hat{\mathcal{I}}(\hat{X}_i)\}$ over all dead points $\{\boldsymbol{\Theta}_i\}$.
  Use $\{\hat{\mathcal{I}}_i\}$ to assign lower $\mathcal{L}^{\rm low} = \mathcal{L}(\hat{X}^{\rm high})$ and upper $\mathcal{L}^{\rm high} = \mathcal{L}(\hat{X}^{\rm low})$ likelihood bounds.
  `// Batch Nested Sampling run.`
  Run Static Nested Sampling (Algorithm 1) with:
  (a) K' live points
  (b) sampled uniformly from the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$ based on the lower likelihood bound $\lambda = \mathcal{L}^{\rm low}$
  (c) until the likelihood $\mathcal{L}(\boldsymbol{\Theta})$ of the last dead point exceeds the upper likelihood bound $\mathcal{L}^{\rm high}$.
  `// Merge samples from batch.`
  Merge new batch of dead points $\{\boldsymbol{\Theta}'_i\}$ into the previous set of dead points $\{\boldsymbol{\Theta}_i\}$.
  `// Check whether to stop.`
  Evaluate stopping criterion.
end

et al. (2017b) and implemented in `dynesty`, in Algorithm 3. It has five main steps:

(i) Sample the distribution with Static Nested Sampling to lay down a "baseline run" to get a sense where the posterior mass $\mathcal{P}(X)\mathrm{d}X$ is located.
(ii) Evaluate our importance function $\mathcal{I}(X)$ over the existing set of samples.
(iii) Use the computed importances $\mathcal{I}_i$ to decide where to allocate additional live points/samples.
(iv) Add a new "batch" of samples in the region of interest using Static Nested Sampling.
(v) "Merge" the new batch of samples into the previous set of samples.

We then repeat steps (ii) to (v) until some stopping criterion is met. By default, `dynesty` uses $K_{\rm base} = K_{\rm batch} = 250$ points for each run, although this should be adjusted depending on the problem at hand.

Allocating points using an existing set of samples is a two-step process. First, we evaluate a noisy estimate of our importance function over the samples:

$$\hat{\mathcal{I}}_i = f^{\mathcal{P}} \frac{\hat{p}_i}{\sum_{i=1}^{N} \hat{p}_i} + (1 - f^{\mathcal{P}}) \frac{1 - \hat{\mathcal{Z}}_i/(\hat{\mathcal{Z}}_N + \Delta\hat{\mathcal{Z}}_N)}{\sum_{i=1}^{N} 1 - \hat{\mathcal{Z}}_i/(\hat{\mathcal{Z}}_N + \Delta\hat{\mathcal{Z}}_N)} \tag{22}$$

where we are now using the noisy importance weight $\hat{p}_i$ to estimate the posterior and the rough upper limit $\Delta\hat{\mathcal{Z}}_N \sim \mathcal{L}_N^{\rm max}\hat{X}_N$ to estimate the remaining evidence. Then, we use these values to define new regions of prior volume to sample. By default, `dynesty` only samples from a single contiguous range of prior volume $(X^{\rm low}, X^{\rm high}]$ which define an associated (flipped) range in iteration $[i^{\rm low}, i^{\rm high})$ and likelihood $[\mathcal{L}^{\rm low}, \mathcal{L}^{\rm high})$ defined by the simple heuristic

$$\begin{aligned} i^{\rm low} &= \min\left[\min(\{i\}) - n_{\rm pad}, 0\right] \\ i^{\rm high} &= \max\left[\max(\{i\}) + n_{\rm pad}, N\right] \\ &\forall\, i \in [0, N] \text{ s.t. } \hat{\mathcal{I}}_i \geq f_{\rm max} \times \max(\{\hat{\mathcal{I}}_i\}) \end{aligned} \tag{23}$$

where $f_{\rm max}$ serves as a threshold relative to the peak value and $n_{\rm pad}$ pads the starting/ending iteration. In other words, we compute the importance values $\hat{\mathcal{I}}_i$ over the existing set of samples, compute the minimum $i^{\rm low}$ and maximum $i^{\rm high}$ iterations where the importance is above a threshold $f_{\rm max}$ relative to the peak, and shift the final values by $n_{\rm pad}$. By default, `dynesty` assumes $f^{\mathcal{P}} = 0.8$ (80% posterior vs 20% evidence), $f_{\rm max} = 0.8$ (80% thresholding), and $n_{\rm pad} = 1$.

Once we have computed $[i^{\rm low}, i^{\rm high}]$, we can then just start a new Static Nested Sampling run that samples from the *constrained* prior between $[\mathcal{L}^{\rm low}, \mathcal{L}^{\rm high})$. In the case where $\mathcal{L}^{\rm low} = 0$, this is just the original prior $\pi(\boldsymbol{\Theta})$ and our Static Nested Sampling run is identical to Algorithm 1 except with stopping criteria $\mathcal{L}(\boldsymbol{\Theta}) \geq \mathcal{L}^{\rm high}$. If $\mathcal{L}^{\rm low} > 0$, however, then we are instead starting *interior* to the prior and thus not fully integrating over it. So while those new samples will improve the relative posterior resolution $\Delta\ln X_i$ and thus the posterior estimate $\hat{\mathcal{P}}(\boldsymbol{\Theta})$, they will not actually improve the evidence estimate $\hat{\mathcal{Z}}$.

Finally, we need to "merge" our new set of N' samples $\{\boldsymbol{\Theta}'_1, \ldots, \boldsymbol{\Theta}'_{N'}\}$ into our original set of samples $\{\boldsymbol{\Theta}_j\}$. This process is straightforward and can be accomplished following the procedure outlined in Appendix A. We are then left with a combined set of samples $\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_{N+N'}\}$ with new associated prior volumes $\{X_1, \ldots, X_{N+N'}\}$ and a variable number of live points $\{K_1, \ldots, K_{N+N'}\}$ at every iteration.

3.3 Estimating the Prior Volume

As shown in Appendix A, we can reinterpret the results from §2.3 as a consequence of the two different ways Nested Sampling traverses the prior volume. In the first case, where the number of live points $K_i \geq K_{i-1}$ increases or stays the same, we know that we have (possibly) added live points and then *replaced* the one with the lowest likelihood $\mathcal{L}^{\rm min}$. In this case, the prior volume experiences exponential shrinkage

such that

$$\mathbb{E}\left[\Delta \ln \hat{X}_i\right] = -\frac{1}{K_i} \tag{24}$$

In the second case, where the number of live points $K_{j+1} < K_j$ strictly decreases, we know that we have *removed* the live point(s) with the lowest likelihood $\mathcal{L}^{\min}$. For each of the k iterations where this continues to occur, the prior volume experiences uniform shrinkage such that

$$\mathbb{E}\left[\frac{\Delta \hat{X}_{j+k}}{\hat{X}_j}\right] = \frac{1}{K_j + 1} \tag{25}$$

In Static Nested Sampling, these two regimes are cleanly divided, with the main set of dead points traversing the prior volume exponentially and the final set of "recycled" live points traversing it uniformly. In Dynamic Nested Sampling, however, we are constantly switching between exponential and uniform shrinkage as we increase or decrease the number of live points at a given iteration.

3.4 Stopping Criterion

The implementation of Static Nested Sampling outlined in Algorithm 1 generally *exclusively* targets evidence estimation. This gives a natural stopping criterion (see §2.4) to terminate sampling once we believe that we have integrated over a majority of the posterior $\mathcal{P}(\boldsymbol{\Theta})$ such that additional samples will no longer improve our evidence estimate $\hat{\mathcal{Z}}$.

In the Dynamic Nested Sampling case, however, we are no longer *just* interested in computing the evidence. Because we now have the flexibility to vary the number of live points K_i over time, we are also interested in the *properties* of our integral (and the samples that comprise the integrand) in addition to just whether our integral has converged.

This flexibility necessitates the introduction of more complex stopping criteria to assess whether those alternative properties are behaving as expected. Similar to §3.1, we consider a stopping criteria of the form:

$$\mathcal{S} = s^{\mathcal{P}}\mathcal{S}^{\mathcal{P}} + (1 - s^{\mathcal{P}})\mathcal{S}^{\mathcal{Z}} < \epsilon \tag{26}$$

where ϵ is our tolerance, $\mathcal{S}^{\mathcal{P}}$ is the posterior stopping criterion, $\mathcal{S}^{\mathcal{Z}}$ is the evidence stopping criterion, and $s^{\mathcal{P}}$ is the relative amount of weight given to $\mathcal{S}^{\mathcal{P}}$ over $\mathcal{S}^{\mathcal{Z}}$.

We define our stopping criterion to be the amount of *fractional uncertainty* in the current posterior $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ and evidence $\hat{\mathcal{Z}}$ estimates. For the posterior $\mathcal{P}(\boldsymbol{\Theta})$, we start by defining "posterior noise" to be the Kullback-Leibler (KL) divergence

$$H(\hat{\mathcal{P}}'||\hat{\mathcal{P}}) \equiv \mathbb{E}_{\hat{\mathcal{P}}'}\left[\ln \hat{\mathcal{P}}' - \ln \hat{\mathcal{P}}\right] \tag{27}$$

$$= \int_{\Omega_{\boldsymbol{\Theta}}} \hat{\mathcal{P}}'(\boldsymbol{\Theta}) \ln \hat{\mathcal{P}}'(\boldsymbol{\Theta}) \mathrm{d}\boldsymbol{\Theta} - \int_{\Omega_{\boldsymbol{\Theta}}} \hat{\mathcal{P}}'(\boldsymbol{\Theta}) \ln \hat{\mathcal{P}}(\boldsymbol{\Theta}) \mathrm{d}\boldsymbol{\Theta} \tag{28}$$

between the posterior estimate $\hat{\mathcal{P}}'(\boldsymbol{\Theta})$ from a random hypothetical Nested Sampling run with the same setup and our current estimate $\hat{\mathcal{P}}(\boldsymbol{\Theta})$. This can be interpreted as the "information loss" due to random noise in our posterior estimate $\hat{\mathcal{P}}(\boldsymbol{\Theta})$. Our proposed posterior stopping criteria is then

$$\mathcal{S}^{\mathcal{P}} \equiv \frac{1}{\epsilon^{\mathcal{P}}} \frac{\sigma\left[H(\hat{\mathcal{P}}'||\hat{\mathcal{P}})\right]}{\mathbb{E}\left[H(\hat{\mathcal{P}}'||\hat{\mathcal{P}})\right]} \tag{29}$$

where $\epsilon^{\mathcal{P}}$ normalizes the posterior deviation to a desired scale. For the evidence $\mathcal{Z}$, this is just the estimated fractional scatter between the evidence estimates $\hat{\mathcal{Z}}'$ from random hypothetical Nested Sampling runs with the same setup. Following Higson et al. (2017b), we opt to compute this in log-space for convenience:

$$\mathcal{S}^{\mathcal{Z}} \equiv \frac{1}{\epsilon^{\mathcal{Z}}} \sigma\left[\ln \hat{\mathcal{Z}}'\right] \tag{30}$$

where $\epsilon^{\mathcal{Z}}$ normalizes the evidence deviation to a desired scale.

Unsurprisingly, we do not have access to the distribution of all hypothetical Nested Sampling runs with the same setup to compute these exact estimates. However, as with §2.4 and §3.2, we *do* have access to noisy estimates of these quantities via procedures described in Higson et al. (2017a) and outlined in Appendix A for simulating Nested Sampling errors. `dynesty` uses M simulated values of these noisy estimates to estimate the stopping criteria as:

$$\hat{\mathcal{S}} = \frac{s^{\mathcal{P}}}{\epsilon^{\mathcal{P}}} \frac{\sigma\left[\{\hat{H}_1, \ldots, \hat{H}_M\}\right]}{\mathbb{E}\left[\{\hat{H}_1, \ldots, \hat{H}_M\}\right]} + \frac{(1 - s^{\mathcal{P}})}{\epsilon^{\mathcal{Z}}} \sigma\left[\{\ln \hat{\hat{\mathcal{Z}}}_1, \ldots, \ln \hat{\hat{\mathcal{Z}}}_M\}\right] \tag{31}$$

where the $\hat{\hat{\mathcal{Z}}}$ notation just emphasizes that we are constructing a noisy estimator of our already-noisy estimate $\hat{\mathcal{Z}}$. By default, `dynesty` assumes $s^{\mathcal{P}} = 1$ (100% focused on reducing posterior noise), $\epsilon = 1$, $\epsilon^{\mathcal{P}} = 0.02$, $\epsilon^{\mathcal{Z}} = 0.1$, and $M = 128$.

4 IMPLEMENTATION

Now that we have outlined the basic algorithm and approach behind Dynamic Nested Sampling, we now turn our attention to the problem of generating samples from the constrained prior. `dynesty` approaches this problem in two parts:

(i) constructing appropriate bounding distributions that encompass the remaining prior volume over multiple possible modes and
(ii) proposing new live points by generating samples conditioned on these bounds.

`dynesty` contains several options for both constructing bounds and sampling conditioned on them. We provide an broad overview of each of these in turn.

4.1 Bounding Distributions

In general, `dynesty` tries to use the distribution of the current set of live points to try and get a rough idea of the shape and size of the various regions of prior volume that we are currently sampling. These are then used to condition various sampling methods to try and improve the efficiency. There are five bounding methods currently implemented in `dynesty`:

- no bounds (i.e. the unit cube),
- a single ellipsoid,
- multiple ellipsoids,
- many overlapping balls, and
- many overlapping cubes.

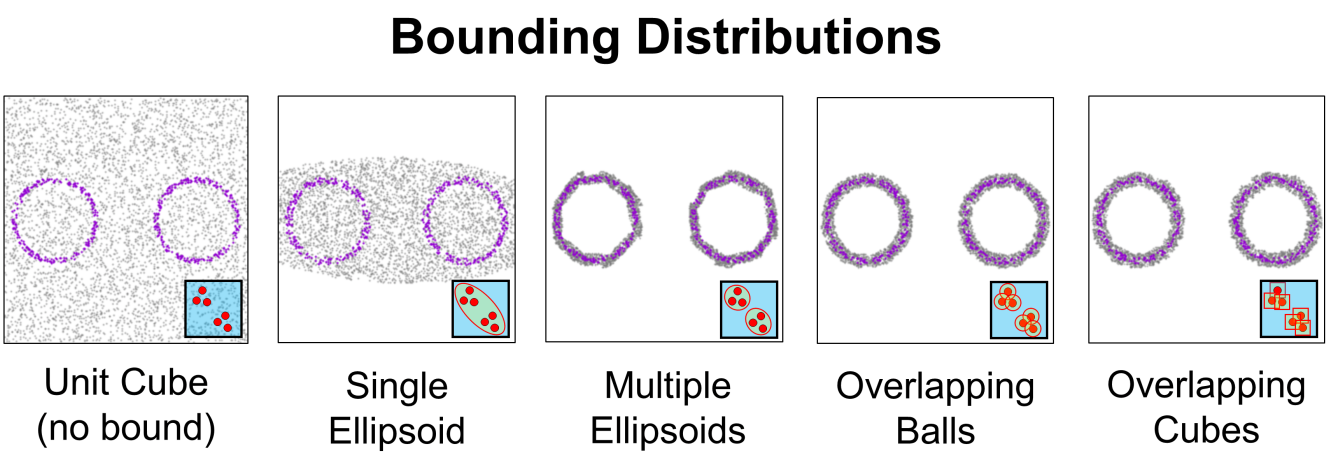

Figure 4. An example highlighting the various bounding distributions implemented in **dynesty**. These include the entire unit cube (left), a single ellipsoid (left-middle), multiple overlapping ellipsoids (middle), overlapping spheres (right-middle), and overlapping cubes (right). The current set of live points are shown in purple while draws from the bounding distribution are shown in grey. A schematic representation of each bounding distribution is shown in the bottom-right-hand corner of each panel. See §4.1 for additional details.

In general, single ellipsoids tend to perform reasonably well at estimating structure when the likelihood is roughly Gaussian and uni-modal. In more complex cases, however, decomposing the live points into separate clusters with their own bounding ellipsoids works reasonably well at locating and tracking structure. In low ($D \lesssim 5$) dimensions, allowing the live points themselves to define emergent structure through many overlapping balls or cubes can perform better provided the $\mathcal{L}(\boldsymbol{\Theta})$ spans similar scales in each of the parameters. Finally, using no bounds at all is only recommended as an option of last resort and is mostly relevant when performing systematics checks or if the number of live points $K \ll D^2/2$ is small relative to the number of possible parameter covariances.

By default, **dynesty** uses multiple ellipsoids. A summary of the various bounding methods can be found in Figure 4. We describe these each in turn below.

4.1.1 Unit Cube

The simplest case of using the entire unit cube (i.e. simple rejection sampling over the entire prior $\pi(\boldsymbol{\Theta})$ with no limits) can be useful in a few edge cases where the number of live points K is small compared to the number of dimensions D, or where users are interested in performing tests to verify sampling behavior.

4.1.2 Single Ellipsoid

As shown in (Mukherjee et al. 2006), a single bounding ellipsoid can be effective if the posterior is unimodal and roughly Gaussian. **dynesty** uses a scaled version of the empirical covariance matrix $\mathbf{C}' = \gamma\mathbf{C}$ centered on the empirical mean $\boldsymbol{\mu}$ of the current set of live points to determine the size and shape of the ellipsoid, where γ is set so the ellipsoid encompasses all available live points.

4.1.3 Multiple Ellipsoids

By default, **dynesty** does not assume the posterior is unimodal or Gaussian and instead tries to bound the live points using a set of (possibly overlapping) ellipsoids. These are constructed using an iterative clustering scheme following the algorithm outlined in Shaw et al. (2007) and Feroz & Hobson (2008) and implemented in the online package **nestle**.[3] In brief, we start by constructing a bounding ellipsoid over the entire collection of live points. We then initialize 2 k-means clusters at the endpoints of the major axes, optimize their positions, assign live points to each cluster, and construct a new pair of bounding ellipsoids for each new cluster of live points. The decomposition is accepted if the combined volume of the subsequent pair of ellipsoids is substantially smaller. This process is then performed recursively until no decomposition is accepted.

By default, **dynesty** tries to be substantially more conservative when decomposing live points into separate clusters and bounding ellipsoids than alternative approaches used in **MultiNest** (Feroz & Hobson 2008; Feroz et al. 2013). This algorithmic choice, which can substantially reduce the overall sampling efficiency, is made in order to avoid "shredding" the posterior into many tiny islands of isolated live point clusters. As shown in Buchner (2016), that behavior can lead to biases in the estimated evidence $\hat{\mathcal{Z}}$ and posterior $\hat{\mathcal{P}}(\boldsymbol{\Theta})$.

4.1.4 Overlapping Balls

An alternate approach to using bounding ellipsoids is to allow the current set of live points themselves to define emergent structure. The simplest approach used in **dynesty** follows Buchner (2016) by assigning a D-dimensional ball (sphere) with radius r to each live point, where r is set using bootstrapping and/or leave-one-out techniques to en-

[3] **dynesty** is built off of **nestle** with the permission of its developer Kyle Barbary.

compass ≥ 1 other live points. One benefit to this approach over using multiple ellipsoids (which can depend sensitively on the clustering schemes) is that it is almost entirely free of tuning parameters, with the overall behavior only weakly dependent on the number of bootstrap realizations.

4.1.5 *Overlapping Cubes*

As with the set of overlapping balls, `dynesty` also implements a similar algorithm based on Buchner (2016) involving overlapping cubes with half-side-length ℓ. As §4.1.4, ℓ is derived using either bootstrapping and/or leave-one-out techniques so that the cubes encompass ≥ 1 other live points.

4.2 Sampling Methods

Once a bounding distribution has been constructed, `dynesty` generates samples conditioned on those bounds. In general, this follows a strategy of

$$f(s\mathbf{C}_b, \boldsymbol{\Theta}) \rightarrow \boldsymbol{\Theta}' \tag{32}$$

where $\mathbf{C}_b$ is the covariance associated with a particular bound b (e.g., an ellipsoid), $\boldsymbol{\Theta}$ is the starting position, $\boldsymbol{\Theta}'$ is the final proposed position, and $s \sim 1$ is a scale-factor that is adaptively tuned over the course of a run to ensure optimal acceptance rates.

`dynesty` implements four main approaches to generating samples:

- uniform sampling,
- random walks,
- multivariate slice sampling, and
- Hamiltonian slice sampling.

These each are designed for different regimes. Uniform sampling can be relatively efficient in lower dimensions where the bounding distribution can approximate the prior volume better, but tends to struggle in higher dimensions since it is extremely sensitive to the size of the bounds. Random walks are less sensitive to the size of the bounding distribution and so tend to work better than uniform sampling in moderate dimensional spaces but still struggle in high-dimensional spaces because of the exponentially increasing amount of volume it needs to explore. Multivariate and Hamiltonain slice sampling often performs better in these high-dimensional regimes by avoiding sampling directly from the volume and taking advantage of gradients, respectively.

By default, `dynesty` resorts to uniform sampling when the number of dimensions $D < 10$, random walks when $10 \leq D \leq 20$, and Hamiltonian/multivariate slice sampling when $D > 20$ if a gradient is/is not provided. A summary of the various sampling methods can be found in Figure 5. We describe these each in turn below.

4.2.1 *Uniform Sampling*

If we assume that our bounding distribution $B(\boldsymbol{\Theta})$ encloses the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$, the most direct approach to generating samples from the bounds is to sample from them uniformly. This procedure by construction produces entirely independent samples between each iteration i, and tends to work best when the volume of the bounds $X_B(\lambda)$ is roughly the same order of magnitude as the current prior volume $X(\lambda)$ (leading to $\gtrsim 10\%$ acceptance rates).

In general, the procedure for generating uniform samples from overlapping bounds is straightforward (see, e.g., Feroz & Hobson 2008; Buchner 2016):

(i) Pick a bound b at random with probability $p_b \propto X_b$ proportional to its volume X_b.
(ii) Sample a point $\boldsymbol{\Theta}_b$ uniformly from the bound.
(iii) Accept the point with probability $1/q$, where $q \geq 1$ is the number of bounds $\boldsymbol{\Theta}_b$ lies within.

This approach ensures that any proposed sample will be drawn from the bounding distributing $B(\boldsymbol{\Theta})$ comprised of the *union* of all bounds, which has a volume $X_B \leq \sum_{b=1}^{N_b} X_b$ that is strictly less than or equal to the sum of the volumes of each individual bound.

Generating samples uniformly from the bounds in §4.1 falls into two cases: cubes and ellipsoids. Generating points from an D-cube centered at $\boldsymbol{\Theta}_b$ with half-side-length ℓ is trivial and can be accomplished via:

(i) Generate D iid uniform random numbers $\mathbf{U} = \{U_1, \ldots, U_D\}$ from $[-\ell, \ell]$.
(ii) Set $\boldsymbol{\Theta}' = \mathbf{U} + \boldsymbol{\Theta}_b$.

Generating points from an ellipsoid centered at $\boldsymbol{\Theta}_b$ with covariance $\mathbf{C}_b$ with matrix square-root $\mathbf{C}_b^{1/2}$ is also straightforward but slightly more involved:

(i) Generate D iid standard normal random numbers $\mathbf{Z} = \{Z_1, \ldots Z_D\}$.
(ii) Compute the normalized vector $\mathbf{V} \equiv \mathbf{Z}/||\mathbf{Z}||$.
(iii) Draw a standard uniform random number U and compute $\mathbf{S} \equiv U^D\mathbf{V}$.
(iv) Set $\boldsymbol{\Theta} = \mathbf{C}_b^{1/2}\mathbf{S} + \boldsymbol{\Theta}_b$.

Step (ii) creates a random vector $\mathbf{V}$ that is uniformly distributed on the *surface* of the D-sphere. Step (iii) randomly moves $\mathbf{V} \rightarrow \mathbf{S}$ to an interior radius $r \in (0, 1)$ based on the fact that the volume of a D-sphere scales as $V(r) \propto r^D$. Finally, step (iv) adjusts the scale, shape, and center to match that of the bounding ellipsoid.

4.2.2 *Random Walks*

An alternative approach to sampling uniformly within the bounding distribution $B(\boldsymbol{\Theta})$ is to instead to try and propose new positions by "evolving" a given live point $\boldsymbol{\Theta}_k \rightarrow \boldsymbol{\Theta}'$ to a new position. Since $\mathcal{L}(\boldsymbol{\Theta}_k) \geq \mathcal{L}_i^{\min}$ at a given iteration by definition, this procedure also guarantees that we will be generating samples exclusively within the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$.

One straightforward approach to "evolving" a live point to a new position is to consider sampling from the constrained prior using a simple Metroplis-Hastings (MH; Metropolis et al. 1953; Hastings 1970) MCMC algorithm:

(i) Propose a new position $\boldsymbol{\Theta}' \sim Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}_k)$ from the proposal distribution $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}_k)$ starting from $\boldsymbol{\Theta}_k$.
(ii) Move to $\boldsymbol{\Theta}'$ with probability $A = \frac{\pi_\lambda(\boldsymbol{\Theta}')}{\pi_\lambda(\boldsymbol{\Theta}_k)} \frac{Q(\boldsymbol{\Theta}_k|\boldsymbol{\Theta}')}{Q(\boldsymbol{\Theta}'|\boldsymbol{\Theta}_k)}$. Otherwise, stay at $\boldsymbol{\Theta}_k$.
(iii) Repeat (i)-(ii) for $N_{\rm walks}$ iterations.

Since the constrained prior is flat (see §2.2), the ratio of the

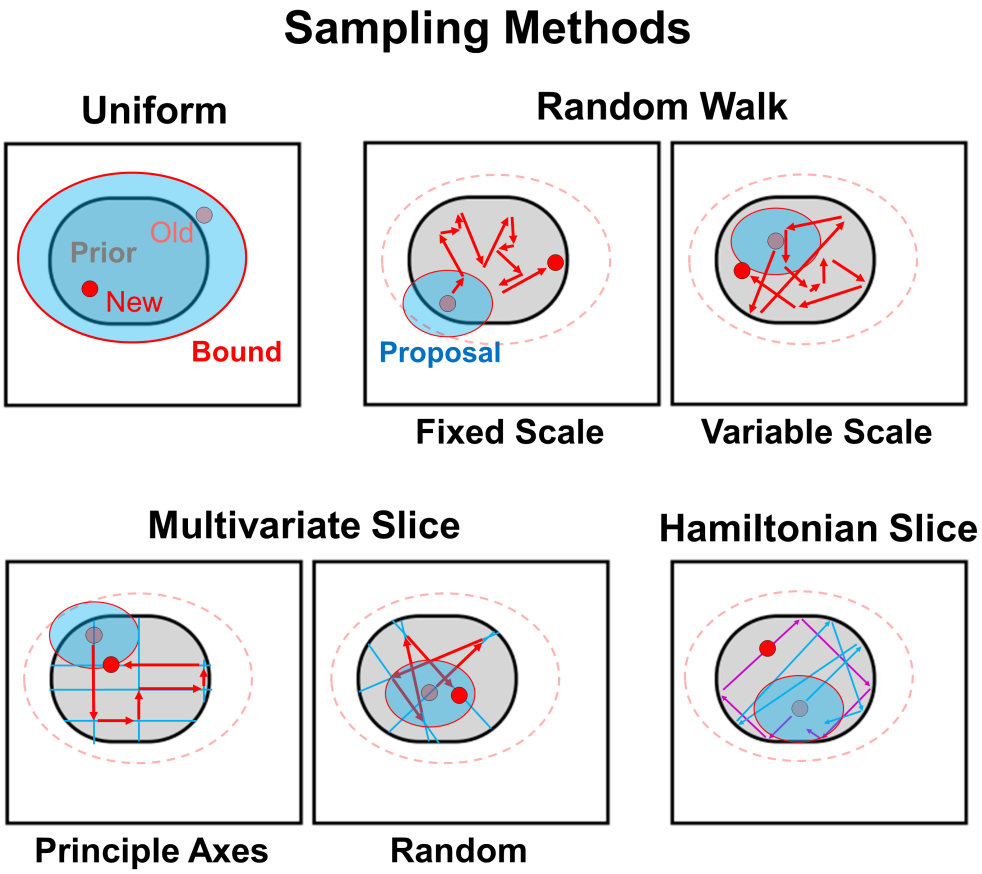

Figure 5. A schematic illustration of the different sampling methods implemented in `dynesty`. These include: uniform sampling from the bounding distribution (top-left), random walks proposals starting from a random live point based on the bounding distribution (top-right) with either fixed or variable scale-lengths for proposals, multivariate slice sampling proposals starting from a random live point (bottom-left) using either the principle axes or a random direction sampled from the bounding distribution, and Hamiltonian slice sampling away from a random live point forwards and backwards in time (bottom-right). See §4.2 for additional details.

constrained prior values is by definition 1. Likewise, if we choose a symmetric proposal distribution $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}_k)$, then the ratio of the proposal distributions also evaluates to 1. This procedure then reduces to simply accepting a new point if it is within the constrained prior with $\mathcal{L}(\boldsymbol{\Theta}_i) \geq \lambda$ and rejecting it otherwise. By default, `dynesty` takes $N_{\rm walks} = 25$.

`dynesty` implements two forms of the proposal $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}_k)$. The default option is to propose new positions uniformly from an associated ellipsoid centered on $\boldsymbol{\Theta}_k$ with covariance $\mathbf{C}_b$, where $\mathbf{C}_b$ is one of the bounding distributions that encompasses $\boldsymbol{\Theta}_k$ (selected randomly). The second follows the same form as the first, except the covariance $\mathbf{C}_b$ is re-scaled at each subsequent proposal $t \leq N_{\rm walks}$ by γ following the procedure outlined in Sivia & Skilling (2006):

$$\alpha(t) = \begin{cases} e^{1/N_{\rm acc}(t)} \times \gamma(t-1) & \frac{N_{\rm acc}(t)}{t} > f_{\rm acc} \\ e^{-1/N_{\rm rej}(t)} \times \gamma(t-1) & \frac{N_{\rm acc}(t)}{t} < f_{\rm acc} \\ \gamma(t-1) & \frac{N_{\rm acc}}{t} = f_{\rm acc} \end{cases} \tag{33}$$

where $N_{\rm acc}(t)$ and $N_{\rm rej}(t)$ is the total number of accepted and rejected proposals by iteration t, respectively, $f_{\rm acc}$ is the desired acceptance fraction, and $\gamma(t = 0) = 1$. By default, `dynesty` targets $f_{\rm acc} = 0.5$.

4.2.3 Multivariate Slice Sampling

In higher dimensions, rejection sampling-based methods such as the random walk proposals outlined in §4.2.2 can become progressively more inefficient. To remedy this, `dynesty` includes slice sampling (Neal 2003) routines designed to sample from the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$. These are based on the "stepping out" method proposed in Neal (2003) and Jasa & Xiang (2012), which works as follows in the single-variable case starting from the position x_k of the kth live point:

(i) Draw a standard uniform random number U.
(ii) Set the left bound $L = x_k - wU$ and the right as $R = L + w$ where w is the starting "window".
(iii) While $\mathcal{L}(L) \geq \lambda$, extend the position of the left bound L by w. Repeat this procedure for R.
(iv) Sample a point $x' \sim \mathrm{Unif}(L, R)$ uniformly on the interval from L to R.
(v) If $\mathcal{L}(x') > \lambda$, accept x'. Otherwise, reassign the corresponding bound to be x' (L if $x' < x$ and R otherwise) and repeat steps (iv)-(v).

When sampling in higher dimensions, the single-variable update outlined above can be interpreted as a Gibbs sampling update (Geman & Geman 1987) where instead of draw-

ing $\boldsymbol{\Theta}$ directly we instead update each component in turn

$$\boldsymbol{\Theta}' \sim \pi_{\lambda}(\boldsymbol{\Theta}) \Rightarrow \begin{cases} \Theta_1 \sim \pi_{\lambda}(\Theta_1|\boldsymbol{\Theta}_{\setminus 1}) \\ \quad\vdots \\ \Theta_D \sim \pi_{\lambda}(\Theta_D|\boldsymbol{\Theta}_{\setminus D}) \end{cases} \tag{34}$$

where $\boldsymbol{\Theta}_{\setminus i}$ are the set of $D-1$ parameters excluding Θ_i. We then repeat this procedure for $N_{\rm slices}$ iterations. By default `dynesty` takes $N_{\rm slices} = 5$.

This procedure is generally robust, although it can introduce longer correlation times if there are strong covariances between parameters. To mitigate this, `dynesty` by default executes single-variable slice sampling updates along the principle axes $\mathbf{V}_b \equiv \{\mathbf{v}_{1,b}, \ldots, \mathbf{v}_{D,b}\}$ associated with the covariance $\mathbf{C}_b$ from a given bound b. This allows us to automatically set both the direction $\mathbf{v}_{i,b}$ and associated scale $||\mathbf{v}_{i,b}||$ of the window while trying to reduce the correlations among sets of parameters.

Alternately, instead of executing a full Gibbs update by rotating through the entire set of parameters in turn, we can sample along a random trajectory $\mathbf{v}'$ through the prior instead. This procedure is similar to that implemented in `PolyChord` (Handley et al. 2015), except that rather than "whitening" the set of live points using the associated $\mathbf{C}_b$ we instead draw $\mathbf{v}'$ from the *surface* of the corresponding bound with covariance $\mathbf{C}_b$. Provided a suitable number of $N_{\rm slices} \sim D$, this procedure also can generate suitably independent new positions $\boldsymbol{\Theta}'$.

4.2.4 Hamiltonian Slice Sampling

Over the past two decades, sampling methods have increasingly attempted to incorporate gradients to improve their overall performance, especially in high-dimensional spaces. The most common class of methods are based on Hamiltonian Monte Carlo (HMC; Neal 2012; Betancourt 2017), whereby a particle at a given position $\mathbf{x}$ is assigned a mass matrix $\mathbf{M}$ and some momentum $\mathbf{p}$ and allowed to sample from the joint distribution

$$P(\mathbf{x}, \mathbf{p}|\mathbf{M}) \propto \exp\left[-\mathcal{H}(\mathbf{x}, \mathbf{p}|\mathbf{M})\right] \tag{35}$$

where

$$\mathcal{H}(\mathbf{x}, \mathbf{p}|\mathbf{M}) \equiv U(\mathbf{x}) + K(\mathbf{p}|\mathbf{M}) \equiv -\ln\left[\pi(\mathbf{x})\mathcal{L}(\mathbf{x})\right] + \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p} \tag{36}$$

is the Hamiltonian of the system with a "potential energy" $U(\mathbf{x})$ and "kinetic energy" $K(\mathbf{p}|\mathbf{M})$, and T is the transpose operator. Typically, proposals are generated by sampling the momentum from the corresponding multivariate Normal (Gaussian) distribution

$$\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{M}) \,, \tag{37}$$

with mean $\mathbf{0}$ and covariance $\mathbf{M}$, evolving the system via Hamilton's equations from $\mathcal{H}(\mathbf{x}, \mathbf{p}) \rightarrow \mathcal{H}(\mathbf{x}', \mathbf{p}')$, and then accepting the new position based on the MH acceptance criteria outlined in §4.2.2. In other words, at each iteration we randomly assign a given particle some mass and velocity and then have it explore the potential defined by the (log-)posterior.

As with the previous methods, this approach simplifies dramatically when sampling over the constrained prior $\pi_{\lambda}(\boldsymbol{\Theta})$. In that case, since the distribution is flat, the momentum remains unchanged until the particle hits the hard likelihood boundary, at which point it reflects so that

$$\mathbf{p}' = \mathbf{p} - 2\mathbf{h}\frac{\mathbf{p}\cdot\mathbf{h}}{||\mathbf{h}||^2} \tag{38}$$

where $\mathbf{h}$ is the gradient at the point of reflection. This version of the algorithm is referred to elsewhere as Galilean Monte Carlo (Skilling 2012; Feroz & Skilling 2013) or reflective slice sampling (Neal 2003).

In practice, since we have to evolve the system discretely, there are a few additional caveats to consider. Most importantly, the use of discrete time-steps means reflection will not occur right *at* the boundary of the constrained prior but slightly *beyond* it, which does not guarantee reflections will end up back inside the constrained prior. This behavior, which arises from larger time-steps, "terminates" the particle's trajectory in that particular direction and leads to inefficient sampling that isn't able to explore the full parameter space.

On the other hand, using extremely small time-steps means spending the vast majority of time evaluating positions along a straight line, which is also non-optimal. `dynesty` by default attempts to compromise between these two behaviors by optimizing the time-step so that $f_{\rm move} \sim 0.9$ of total steps are spent moving forward passively instead of reflecting or terminating. In addition, `dynesty` by default caps the total number of time-steps to $N_{\rm move} = 100$ to prevent trajectories from being evolved indefinitely.

Similar to algorithms such as the No U-Turn Sampler (NUTS; Hoffman & Gelman 2011), `dynesty` also considers trajectories evolved forwards and backwards in time to broaden the range of possible positions explored in a given proposal. While these roughly double the number of overall time-steps, they substantially improve overall behavior by exploring larger regions of the constrained prior.

`dynesty` employs two additional schemes to try and further mitigate discretization effects on the sampling procedure described above. First, the time-step used at a given iteration is allowed to vary randomly by up to 30% following recommendations from Neal (2012). This helps to suppress resonant behavior that can arise from poor choices of time-steps without substantially impacting overall performance. Second, rather than merely accepting positions at the end of a trajectory, `dynesty` instead tries to sample uniformly from the *entire trajectory* by treating it as a set of slices defined by $(\boldsymbol{\Theta}^i_L, \boldsymbol{\Theta}^i, \boldsymbol{\Theta}^i_R)$ left-inner-right position tuples. New samples are then proposed via the following scheme:

(i) Compute the length ℓ_i of each line segment $(\boldsymbol{\Theta}^i_L, \boldsymbol{\Theta}^i_R)$.
(ii) Selecting a line segment i at random proportional to its length.
(iii) Sample a point $\boldsymbol{\Theta}'$ uniformly on the line segment defined by $(\boldsymbol{\Theta}^i_L, \boldsymbol{\Theta}^i_R)$.
(iv) If $\mathcal{L}(\boldsymbol{\Theta}') > \lambda$, accept $\boldsymbol{\Theta}'$. Otherwise, reassign the corresponding bound to be $\boldsymbol{\Theta}'$ ($\boldsymbol{\Theta}^i_L$ if $\boldsymbol{\Theta}'$ is on the line segment $[\boldsymbol{\Theta}^i_L, \boldsymbol{\Theta}^i)$ and $\boldsymbol{\Theta}^i_R$ otherwise) and repeat steps (i)-(iv).

While there are a variety of possible approaches to applying HMC-like methods to Nested Sampling other than the basic procedure outlined above, we defer any detailed comparisons between them to possible future work.

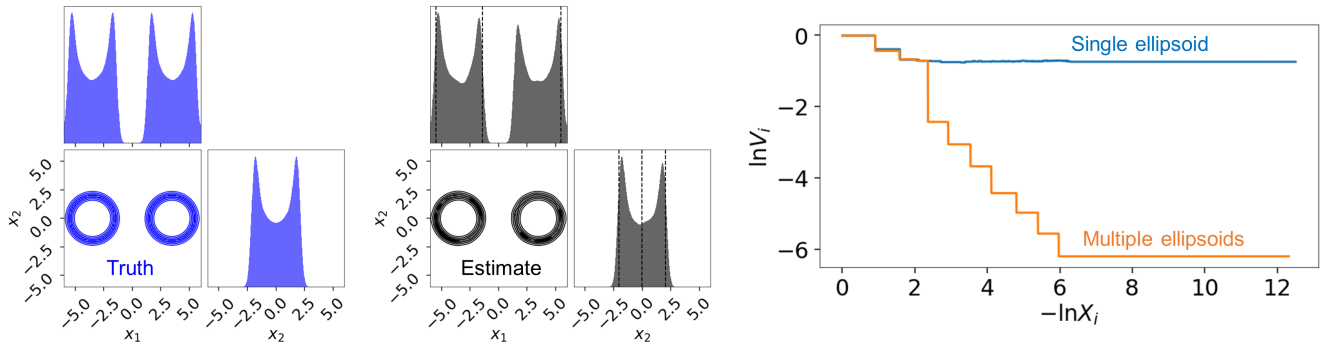

Figure 6. Illustration of `dynesty`'s performance using multiple bounding ellipsoids and uniform sampling over 2-D Gaussian shells (highlighted in Figure 4) meant to test the code's bounding distributions. *Left*: A smoothed corner plot showing the exact 1-D and 2-D marginalized posteriors of the target distribution. *Middle*: As before, but now showing the final distribution of weighted samples. *Right:* The volume of the bounding distribution when using a single ellipsoid (blue) versus multiple ellipsoids (orange) over the course of the run. Since a single ellipsoid is a poor model for this distribution, its volume quickly saturates as it becomes unable to accurately capture the distribution of live points. Allowing the bounding distribution to be modeled by multiple ellipsoids allows for `dynesty` to capture the more complex structure as the live points move increasingly into organized rings.

5 TESTS

Here, we examine `dynesty`'s performance on a variety of toy problems designed to stress-test various aspects of the code. Additional tests can also be found online.

5.1 Gaussian Shells

One standard problem that tests the efficiency of the ability of bounding distributions to transition between a flat surface to separated, elongated structures is the D-dimensional "Gaussian shells" from Feroz & Hobson (2008). The likelihood of the distribution is defined as

$$\mathcal{L}(\boldsymbol{\Theta}) = \mathrm{circ}(\boldsymbol{\Theta}|\mathbf{c}_1, r_1, w_1) + \mathrm{circ}(\boldsymbol{\Theta}|\mathbf{c}_2, r_2, w_2) \tag{39}$$

where

$$\mathrm{circ}(\boldsymbol{\Theta}|\mathbf{c}, r, w) = \frac{1}{\sqrt{2\pi w^2}} \exp\left[-\frac{1}{2}\frac{(||\boldsymbol{\Theta} - \mathbf{c}|| - r)^2}{w^2}\right] \tag{40}$$

Following Feroz et al. (2013), we take the centers $\mathbf{c}_1$ and $\mathbf{c}_2$ of the two positions to be -3.5 and 3.5 in the first dimension and 0 in all others, respectively, the radius $r = 2$, and the width $w = 0.1$. Our prior is defined to be uniform from $[-6, 6]$ to encompass the majority of the likelihood and ensure a smooth transition between the uni-modal starting distribution and the multi-modal target distribution.

We illustrate `dynesty`'s performance in the 2-D case in Figure 6. The default configuration options in `dynesty` (multiple ellipsoid bounds with uniform sampling) lead to a roughly 10% sampling efficiency over the course of $\sim$ 20k iterations when using Dynamic Nested Sampling and lead to excellent posterior estimates. We also see that the multi-ellipsoidal decomposition algorithm works as expected, with the total volume of the bounding distribution decreasing dramatically as the live points begin to organize themselves within the two shells.

5.2 Eggbox

Another distribution we consider to test the ability of `dynesty` to track and evolve multiple modes is the 2-D "Eggbox" likelihood from Feroz & Hobson (2008), which we defined as

$$\mathcal{L}(x, y) = \exp\left\{\left[2 + \cos\left(\frac{5\pi(x-1)}{2}\right)\sin\left(\frac{5\pi(y-1)}{2}\right)\right]^5\right\} \tag{41}$$

This distribution is periodic over the 2-D unit cube, with 13 localized modes contained within a given period. We take our prior to be standard uniform in x and y to limit sampling to one period.

The resulting posterior and evidence estimates from several posterior-oriented and evidence-oriented Dynamic Nested Sampling runs are shown in Figure 7. `dynesty` is able to sample from this distribution quite effectively, with average sampling efficiencies ranging from $20 - 40\%$ when sampling uniformly from the multiple ellipsoids or overlapping balls.

5.3 Exponential Wave

We next apply `dynesty` to a signal reconstruction problem with multiple modes and periodic boundary conditions. Our model is a transformed periodic single from 0 to 2π:

$$y(x) = \exp\left[n_a \sin(f_a x + p_a) + n_b \sin(f_b x + p_b)\right] \tag{42}$$

where we observe noisy data points drawn from

$$\hat{y}(x) \sim \mathcal{N}\left(y(x), \sigma^2\right) \tag{43}$$

The likelihood for this model is Gaussian over the corresponding observed datapoints such that

$$\ln \mathcal{L}(\boldsymbol{\Theta}) = -\frac{1}{2}\sum_{i=1}^{N} \ln\left(2\pi\sigma^2\right) + \frac{[\hat{y}_i - y(x_i|\boldsymbol{\Theta})]^2}{\sigma^2} \tag{44}$$

and has seven free parameters: two controlling the relevant amplitudes (n_a, n_b), two controlling the frequencies (f_a, f_b),

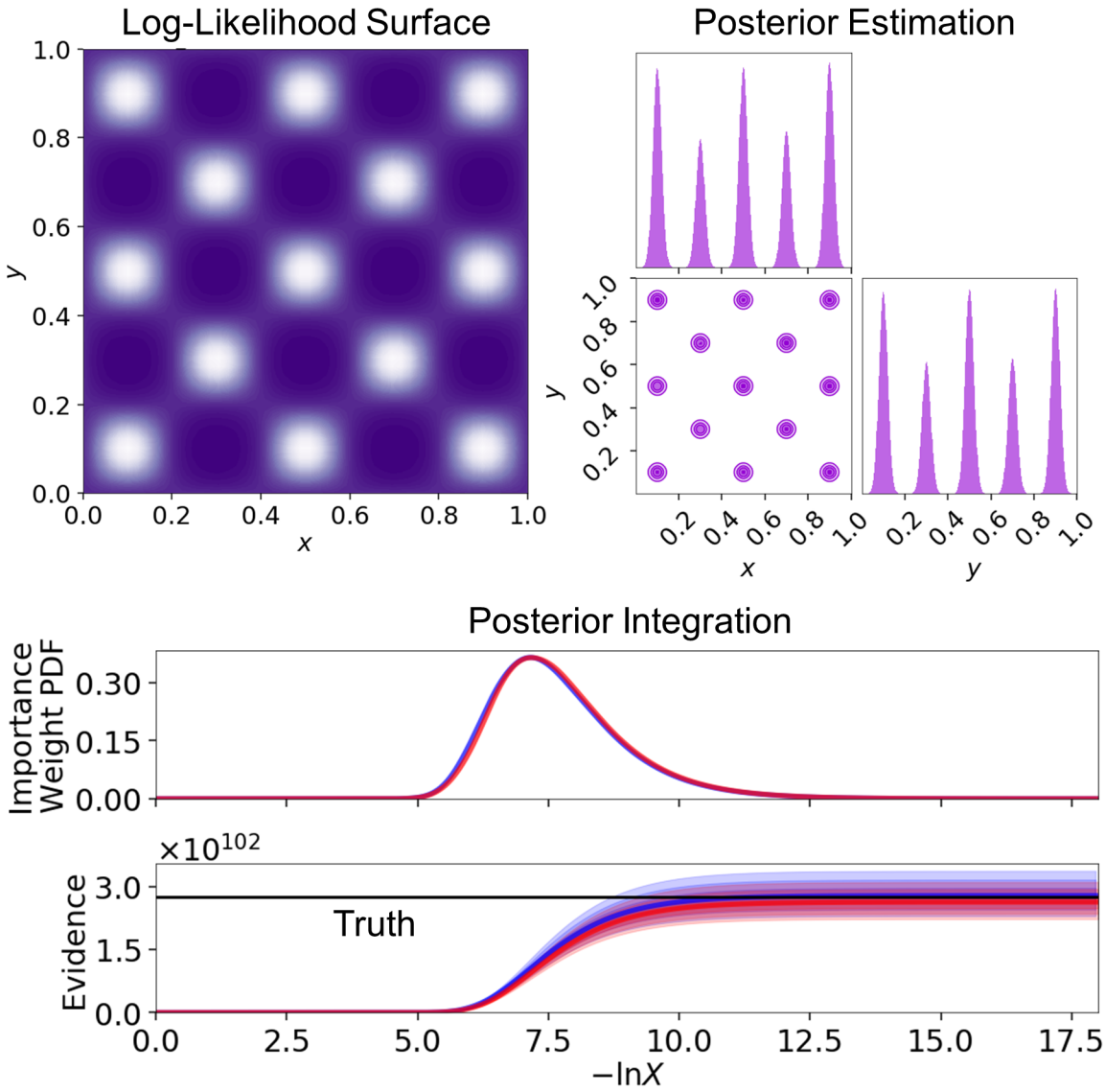

Figure 7. Illustration of `dynesty`'s performance using multiple bounding ellipsoids and overlapping balls with uniform sampling over the 2-D "Eggbox" distribution meant to test the code's bounding distributions. *Top left*: The true log-likelihood surface of the Eggbox distribution. *Top right*: A smoothed corner plot showing the 1-D and 2-D marginalized posteriors of the final distribution of weighted samples from a posterior-oriented Dynamic Nested Samplig run. *Bottom*: The importance weight PDF $p(X)$ (top) and corresponding evidence estimate $\hat{\mathcal{Z}}$ with 1, 2, and 3-sigma uncertainties (bottom) from two independent evidence-oriented Dynamic Nested Sampling runs using multiple ellipsoids (blue) and overlapping balls (red) as bounding distributions.

two controlling the phases (p_a, p_b), and one controlling the scatter σ.

We take our true model parameters to be $f_a = 1.05$, $f_b = 4.2$, $n_a = 0.8$, $n_b = 0.3$, $p_a = 0.1$, $p_b = 2.4$, and $\sigma = 0.2$ so that a solution is close to the boundary. We assign our prior to be uniform or log-uniform in all parameters with $\log n_a \in [-2, 2)$, $\log f_a \in [-2, 2)$, $p_a \in [0, 2\pi)$, $\log n_b \in [-2, 2)$, $\log f_b \in [-2, 2)$, $p_b \in [0, 2\pi)$, and $\log \sigma \in [-2, 0)$, where the priors in p_a and p_b are periodic.

We illustrate `dynesty`'s performance on this problem in Figure 8. We find `dynesty` is able to robustly recover both modes in this problem, including the solution near the boundary.

5.4 200-D Gaussian

We next examine `dynesty`'s behavior in higher dimensions by testing its performance on a 200-D multivariate Gaussian likelihood with mean $\boldsymbol{\mu} = \mathbf{0}$ and covariance $\mathbf{C} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. We assign an identical prior (iid Gaussian with $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{C} = \mathbf{I}$), such that the posterior will also be iid Gaussian with mean $\boldsymbol{\mu} = \mathbf{0}$ but with covariance $\mathbf{C} = (1/2)\,\mathbf{I}$.

We sample from this distribution using Hamiltonian Slice Sampling with the analytic log-likelihood gradient. To further highlight the efficiency of these proposals to explore the posterior, we use a small ($K = 50$) number of live points so that we are highly undersampled relative to the 200-D

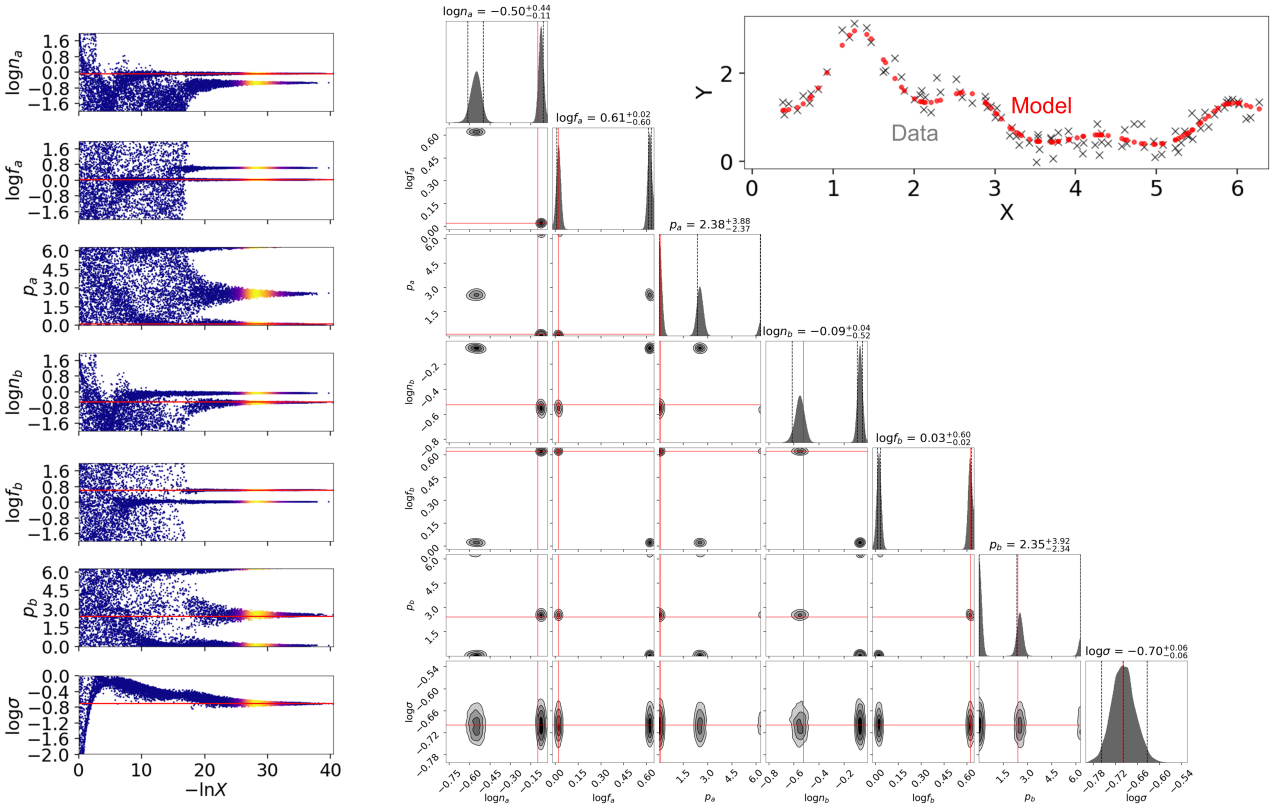

Figure 8. Illustration of **dynesty**'s performance using multiple bounding ellipsoids and multivariate slice sampling over principle axes to model an "Exponential Wave" signal meant to test the code's bounding distributions and incorporation of periodic boundary conditions. *Left*: Trace plots showing the 1-D positions of samples (dead points) over the course of the run, colored by their estimated importance weight PDF $p(X)$. The true model parameters are shown highlighted in red. We see that even though the underlying structure of the distribution spans many different scales and emerges in different stages, **dynesty** is able to confidently identify the final two modes and converge to the underlying model parameters. *Middle*: A corner plot showing the 1-D and 2-D marginalized posteriors from the distribution of the final weighted samples. The true model parameter values are shown in red. The 2.5%, 50%, and 97.5% percentiles (i.e. the 2-sigma credible region) are shown as vertical dashed lines. *Top right*: The noisy data (gray crosses) and underlying model (red points).

space. Since **dynesty** by default uses the empirical covariance (i.e. the MLE estimate) to construct any bounding ellipsoids, this process is dominated by shot noise that can substantially affect the covariance. We consequently impose no bounding distribution (which happens to also be optimal for this problem).

As shown in Figure 9, we find **dynesty** is able to achieve unbiased recovery of the mean, covariance, and evidence under these conditions. The typical sampling efficiency we achieve for this problem is roughly 0.1% (i.e. 1000 likelihood calls per iteration), which translates to roughly 5 per dimension.

5.5 Comparison to MCMC

Nested Sampling and MCMC sampling are different tools designed for different types of problems. Here we perform a limited comparison to highlight the advantages/disadvantages of each methodology.

We consider a simple linear regression problem where our model is

$$y(x) = mx + b \tag{45}$$

and we observe noisy data from

$$\hat{y}_i \sim \mathcal{N}\left(y(x_i), \sigma_i^2 + [f\,y(x_i)]^2\right) \tag{46}$$

where σ_i^2 is the measured variance and f corresponds to an additional fractional systematic uncertainty that we would like to infer in addition to m and b. The likelihood is again Gaussian:

$$\ln \mathcal{L}(m, b, f) = -\frac{1}{2}\sum_{i=1}^{N} \ln\left[2\pi(\sigma_i^2 + f^2(mx_i + b)^2)\right] + \frac{[\hat{y}_i - (mx_i + b)]^2}{\sigma^2 + f^2(mx_i + b)^2} \tag{47}$$

This problem is unimodal and only has three parameters, making it very tractable to both Nested Sampling and MCMC methods.

We choose our priors to be uniform so that $m \in [-5, 0.5)$, $b \in [0, 10)$, and $\ln f \in [-10, 1]$, which are substantially broader than the likelihood distribution but not so broad that the runtime of **dynesty** will be dominated merely integrating over the prior.

We run **dynesty** in three configurations to sample from

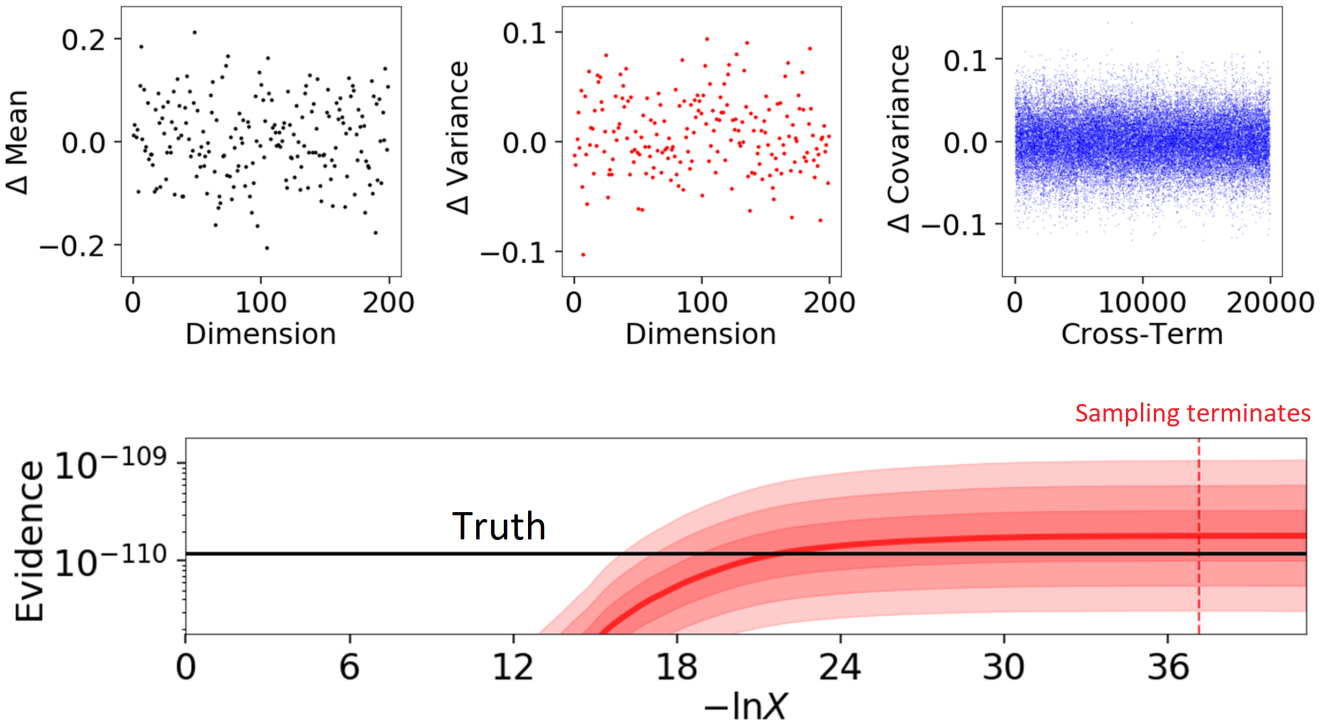

Figure 9. Illustration of `dynesty`'s performance sampling from a 200-D Gaussian using Hamiltonian Slice Sampling (§4.2.4) with gradients and no bounding distribution with only $K = 50$ live points. *Top*: Offsets in the recovered mean (left, black), variance (center, red), and covariance cross-terms (right, blue) relative to an expected mean of $\boldsymbol{mu} = \mathbf{0}$ and covariance of $\mathbf{C} = (1/2)\mathbf{I}$. *Bottom*: The estimated evidence $\hat{\mathcal{Z}}$ (red line) along with the 1, 2, and 3-sigma errors (shaded). The true value is shown in black, along with the location where sampling terminates (dotted red vertical line).

this posterior distribution, using the default settings whenever possible to highlight performance in a "typical" use case. First, we set the weight function to give the posterior 100% of the importance when allocating live points in order to imitate MCMC-like behavior. Then, we revert to the default 80%/20% posterior/evidence weighting scheme to see how much our posterior estimate degrades as we spend a larger fraction of runtime trying to improve our evidence estimates. Finally, we switch out the default sampling mode (uniform sampling) for random walks to forcibly decrease the overall sampling efficiency.

We compare these results to two MCMC alternatives. The first is `emcee` (Foreman-Mackey et al. 2013), which is by far the most common MCMC sampler used in astronomical analyses today. We opt to run it in its default configuration, which uses the "stretch move" from (Goodman & Weare 2010) to make proposals, with $K = 150$ walkers. To be fair in our treatment of the prior, we initialize the walkers randomly from the prior (i.e. the same starting conditions as `dynesty`) and include the time they take to burn in to the posterior when comparing results. We set the burn-in to occur in stages to avoid walkers getting "stuck" in far-flung regions, where at each stage the worst 50% of walkers are reinitialized in a Gaussian ball around the best 50% of walkers.

The second alternative is a standard MH MCMC sampler with a Gaussian proposal distribution. We take the covariance to be the same as that of the posterior distribution determined from the final set of weighted `dynesty` samples to create an relatively optimal proposal distribution. We then run with an identical setup to `emcee` ($K = 150$ chains, burn-in occurring in stages) to maintain consistency between approaches.

The metric we use to compare between methods is the overall "sampling efficiency", which we define to be the ratio of the estimated effective sample size (ESS) $N_{\rm ESS}$ relative to the number of likelihood calls $N_{\rm call}$:

$$f_{\rm samp} \equiv \frac{N_{\rm ESS}}{N_{\rm call}} \tag{48}$$

For `dynesty`, since the samples are all independent but assigned varying importance weights, we choose to estimate the ESS by counting the number of *unique* samples after using systematic resampling to redraw a set up equally-weighted samples.[4]

For the MCMC approaches, we use the standard definition of ESS as

$$N_{\rm ESS} = \frac{N}{1 + 2\sum_{t=1}^{\infty}\rho(t)} \tag{49}$$

where $\rho(t)$ is the auto-correlation after t iterations and we

[4] Using multinomial resampling, which introduces additional sampling noise (Douc et al. 2005; Hol et al. 2006), reduces the ESS by roughly 25% but does not affect our overall conclusions.

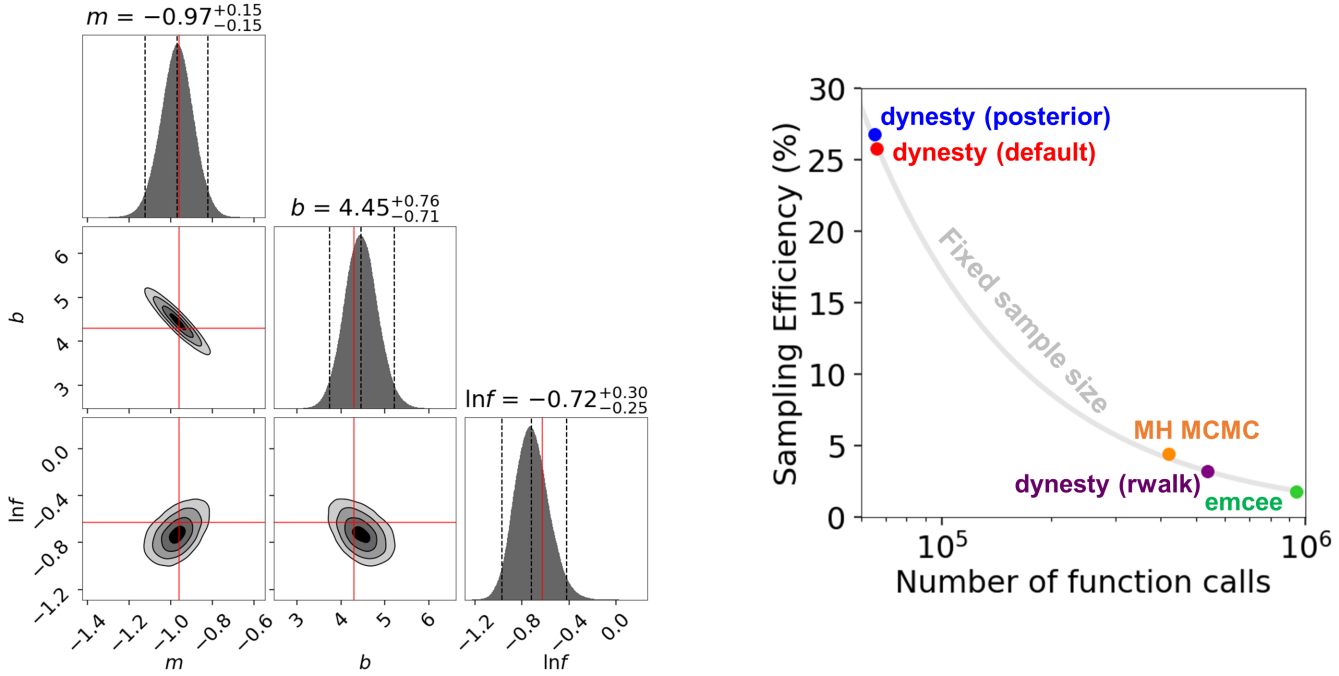

Figure 10. Comparison between `dynesty` and common MCMC alternatives inferring the slope m, intercept b, and (log-)fractional uncertainty $\ln f$ in a simple linear regression problem. See §5.5 for additional details. *Left*: A corner plot showing the 1-D and 2-D marginalized posteriors for the slope m, intercept b, and (log-)fractional uncertainty $\ln f$, with their true values in red. The 2.5%, 50%, and 97.5% percentiles (i.e. the 2-sigma credible region) are shown as vertical dashed lines. We see the posterior is well-constrained and roughly Gaussian. *Right*: The posterior sampling efficiency of `dynesty`, `emcee`, and simple MH MCMC plotted as a function of the total number of likelihood function calls. The predicted efficiency for a fixed effective sample size is shown in gray. We see that `dynesty` optimized for posterior estimation (blue) can be up to 10x more efficient than `emcee` at generating independent samples from the posterior, and 5x more efficient than MH MCMC. As expected, decreasing the emphasis on posterior vs evidence estimation to 80% (red) or using a less efficient sampling method such as random walks (right) also reduces the overall efficiency.

compute the ESS on a per-chain basis. To avoid introducing unnecessary variance into the sum, we opt to truncate it after the auto-correlation stabilizes around 0 (typically after $t \gtrsim 150$). Finally, to be conservative we set the sum to be the maximum value among the individual 1-D sums computed for each parameter. These choices tend to decrease the ESS by $\sim 25\%$ relative to more optimistic ones but does not affect our overall conclusions.

We compare the five different cases above and summarize the results in Figure 10. In all cases, we try to generate enough samples to give similar ESS between each approach based on `dynesty`'s default stopping criterion, which gives $N_{\rm ESS} \sim 17000$. We see that `dynesty` with uniform sampling within multiple bounding ellipsoids is roughly an order of magnitude more efficient at generating independent samples in this problem than MH MCMC and `emcee`, while MH MCMC is slightly more efficient than generating samples with `dynesty` using random walks. `emcee` is the least efficient option, with only an $\sim 2\%$ efficiency driven by long auto-correlation times.

As discussed earlier, all methods experience some amount of overhead transitioning from the prior-dominated to posterior-dominated region. While this leads to $\lesssim 5\%$ of samples being discarded for burn-in for the MCMC cases, it leads to a reduction in the ESS of $\sim 25\%$ for `dynesty`. The fact that `dynesty` performs well even in this case illustrates how important Dynamic Nested Sampling is for ensuring samples are efficiently allocated during runtime.

This result highlights the basic argument first outlined in §2, illustrating that using Nested Sampling to sample from many simpler distributions in turn can sometimes be more effective than trying to sample from the posterior distribution directly with MCMC. In general, Nested Sampling performs well in cases like these where the likelihood varies smoothly in a given region and the prior has reasonable bounds. In other cases where the prior is large or fewer samples from the posterior are needed, MCMC methods are more than sufficient.

6 APPLICATIONS

In addition to the toy problems in §5, `dynesty` has also been applied in several packages and ongoing studies and shown to perform well when applied to real astronomical analyses. These include applications analyzing gravitational waves (Ashton et al. 2018), exoplanets (Diamond-Lowe et al. 2018; Espinoza et al. 2018; Günther et al. 2019), transients (Guillochon et al. 2018), galaxies (Leja et al. 2018a,b), and 3-D dust mapping (Zucker et al. 2018, 2019). We highlight two of these applications below that the author has been personally involved in.

In Leja et al. (2018b), the authors modeled roughly 60k

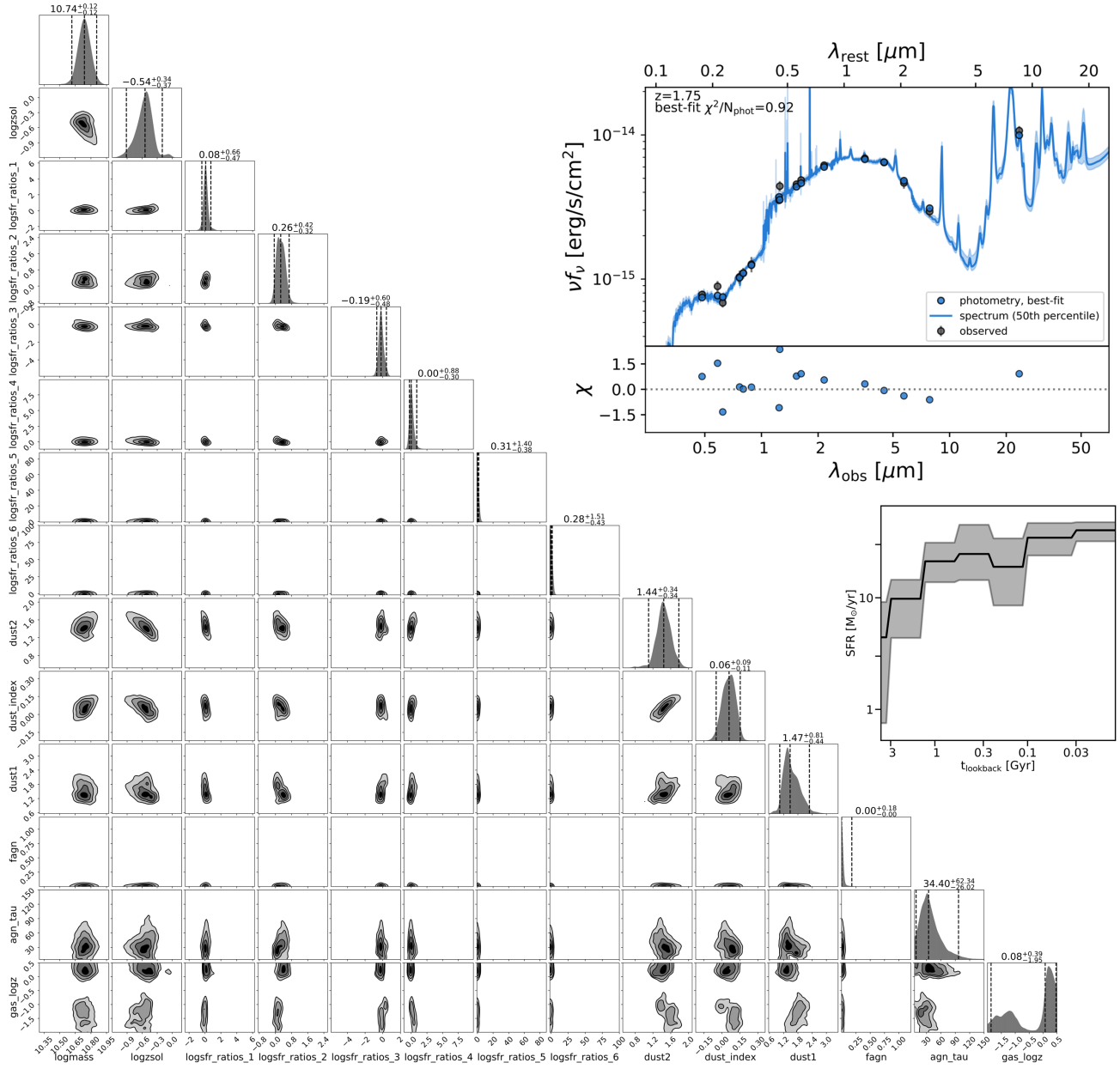

Figure 11. Galaxy SED for object AEGIS 17 from the 3D-HST survey modeled with `Prospector` using `dynesty`. *Left*: A corner plot showing the 1-D and 2-D marginalized posteriors for the 14-parameter galaxy model. The 2.5%, 50%, and 97.5% percentiles (i.e. the 2-sigma credible region) are shown as vertical dashed lines. The posterior includes a bi-modal solution for the gas-phase metallicity. *Top right*: The modeled galaxy SED marginalized over the posterior. The 1-sigma (16-84% credible region) is also shown, along with the error-normalized residuals. The underlying model provides a reasonable fit to the observed data. *Right middle*: The median reconstructed star formation history as a function of look-back time along with the associated 16-84% credible region.

galaxy spectral energy distributions (SEDs) from the 3D-HST survey (Brammer et al. 2012) over a redshift range of $0.5 < z < 2.5$. To conduct this analysis, they used the Bayesian SED fitting code `Prospector` (Johnson et al. in prep.), utilizing `dynesty` as their primary sampler, to sample from a 14-parameter model involving stellar mass, a non-parametric star formation history, stellar and gas metallicites, dust properties, and contributions from possible Active Galactic Nuclei. Compared to previous studies where `emcee` had been used to sample from the posterior (Leja et al. 2017, 2018c), the authors found that `dynesty` provided over an order of magnitude more efficient sampling and was able to characterize a wide variety of posteriors. The results for a typical galaxy are shown in Figure 11.

In Zucker et al. (2019), the authors used a combination of distance and reddening estimates to nearby stars from SED modeling (Speagle et al. in prep.) and Gaia parallax measurements (Gaia Collaboration et al. 2018) to derive distances to dozens of local molecular clouds. The distances to these clouds, however, are sensitive to the number and distribution of stars immediately in front of them as these stars help constrain the location of the "jump" in dust extinc-

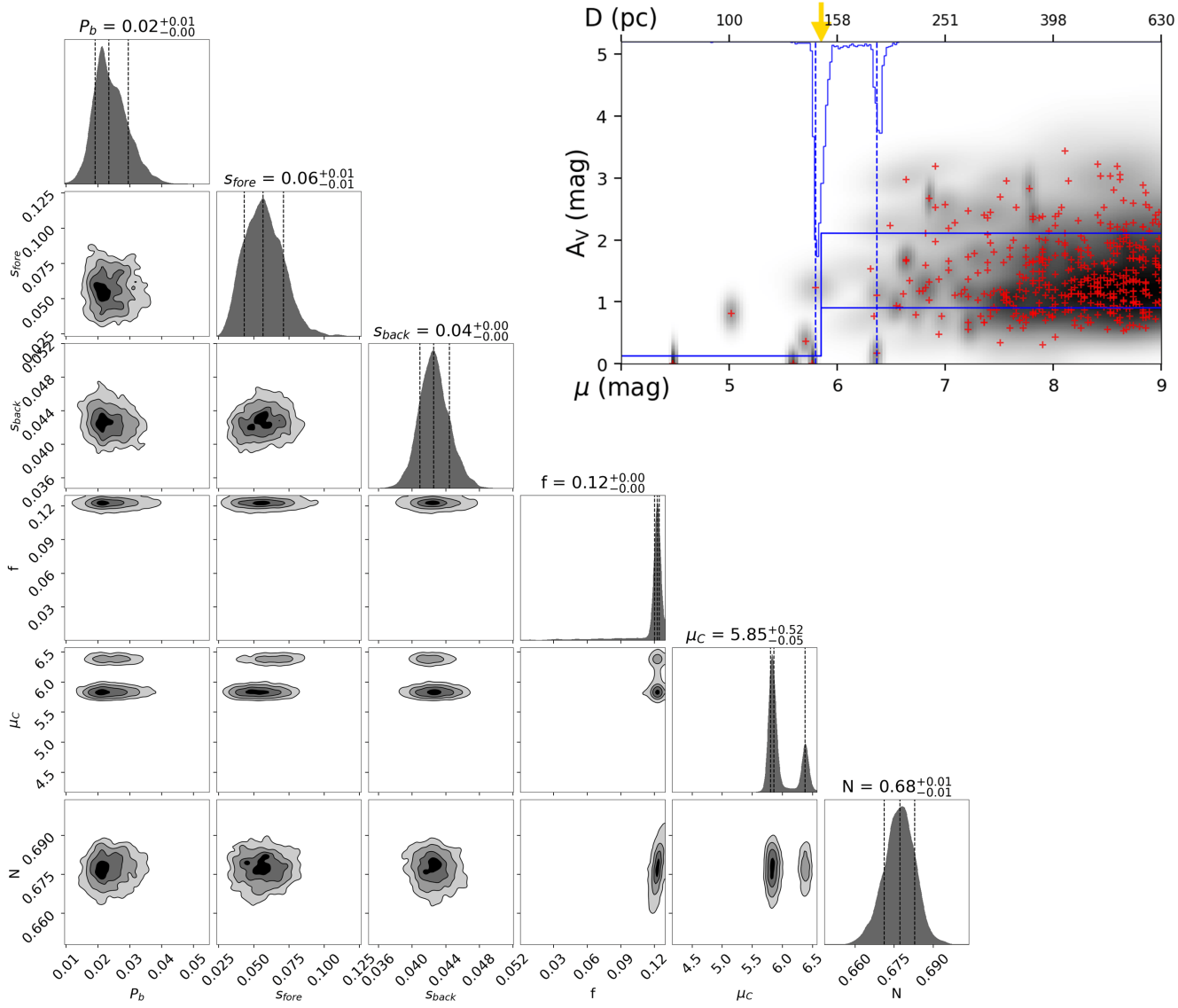

Figure 12. Line-of-sight dust extinction (reddening) model for a sight-line in the Chameleon molecular cloud estimated with **dynesty**. *Left*: A corner plot showing the 1-D and 2-D marginalized posteriors for the 6-parameter line-of-sight model. The 16%, 50%, and 84% percentiles (i.e. the 1-sigma credible region) are shown as vertical dashed lines. The posterior includes a bi-modal solution for the cloud distance μ_C as well as an extended tail for the foreground dust reddening f. *Top right*: The line-of-sight model from the estimated posterior. Individual distance-extinction posteriors for stars used in the fit as shown in grayscale, with most probable distance and extinction shown as a red cross. The blue line shows the typical extinction profile inferred for the sightline. The range of distance estimates is shown as the inverted blue histogram at the top of each panel, with the median cloud distance marked via the vertical blue line and yellow arrow and the 16-84% credible ranges marked via the vertical blue dashed lines. The horizontal blue lines show the estimated 1-sigma scatter in extinction behind the cloud.

tion associated with the cloud. In cases where there are only a small number of foreground stars, this constraint can be quite weak, leading to extended posteriors with multi-modal solutions. This, along with the overall performance illustrated in Figure 10, motivated the use of **dynesty** to sample from the 6-parameter cloud distance model used in the analysis. We highlight one such multi-modal case in Chameleon in Figure 12.

These examples, along with others listed earlier, are large-scale professional applications of **dynesty** that illustrate **dynesty** can work well in theory and in practice.

7 CONCLUSION

With Bayesian inference techniques now a large part of modern astronomical analyses, it has become increasingly important to develop and provide tools to the community that can help to "bridge the gap" between writing the underlying model and estimating the corresponding posterior $\mathcal{P}(\boldsymbol{\Theta})$.

Tools such as `emcee`, `MultiNest`, and `PolyChord`, which provide Markov Chain Monte Carlo and Nested Sampling implementations, have been heavily used and highly cited.

In this paper we presented an overview of `dynesty`, a public, open-source, Python package that implements Dynamic Nested Sampling to enable flexible Bayesian inference over complex, multi-modal distributions. Building on previous work in the literature, we described the basics behind the Dyamic Nested Sampling approaches employed in the code, how we implement them, and how we use a variety of bounding and sampling methods to enable efficient inference. We then showcased `dynesty`'s performance on several toy problems as well as real astronomical application, highlighting its ability to estimate challenging posterior distributions both in theory and in practice.

While we have shown `dynesty` can perform similarly or better than existing MCMC approaches in one simple case, the real test for any package is based on users applying it to their analysis problems. We hope that `dynesty` will prove useful to the community and help facilitate exciting new science over the coming years.

ACKNOWLEDGEMENTS

JSS is grateful to Rebecca Bleich for her support and patience.

This project is the culmination of many individual efforts, not all of whom can be thanked here. First and foremost, JSS would like to thank Daniel Eisenstein, Charlie Conroy, and Doug Finkbeiner for their patience while he pursued this project, Catherine Zucker for her constant stream of feedback during development, and Johannes Buchner for incredibly insightful and inspiring conversations. JSS would also like to thank Johannes Buchner, Hannah Diamond-Lowe, Joel Leja, Locke Patton, and Catherine Zucker for feedback on earlier drafts that substantially improved the quality of this work. JSS would further like to thank Johannes Buchner, Phil Cargile, Ben Cook, James Guillochon, and Ben Johnson for their direct and indirect contributions to the `dynesty` codebase, as well as Kyle Barbary and collaborators for their contributions to `nestle` (upon which `dynesty` was initially based). JSS is also grateful to many beta-testers who provided invaluable feedback during `dynesty`'s development and suffered through many bugfixes, including Gregory Ashton, Ana Bonaca, Phil Cargile, Tansu Daylan, Hannah Diamond-Lowe, Philipp Eller, Jonathan Fraine, Maximilian Günther, Daniela Huppenkothen, Joel Leja, Sandro Tacchella, Ashley Villar, Catherine Zucker, and Joe Zuntz.

This work has benefited from several software packages including `numpy` (van der Walt et al. 2011), `scipy` (Oliphant 2007), `matplotlib` (Hunter 2007), and `corner` (Foreman-Mackey 2016).

REFERENCES

Ashton G., et al., 2018, arXiv e-prints, p. arXiv:1811.02042
Betancourt M., 2017, arXiv e-prints, p. arXiv:1701.02434
Blei D. M., Kucukelbir A., McAuliffe J. D., 2016, arXiv e-prints,
Blitzstein J., Hwang J., 2014, Introduction to Probability. Chapman & Hall/CRC Texts in Statistical Science, CRC Press/Taylor & Francis Group, https://books.google.com/books?id=ZwSlMAEACAAJ
Borne K., et al., 2009, in astro2010: The Astronomy and Astrophysics Decadal Survey. p. P6 (arXiv:0909.3892)
Brammer G. B., et al., 2012, The Astrophysical Journal Supplement Series, 200, 13
Brooks S., Gelman A., Jones G., Meng X.-L., 2011, Handbook of Markov Chain Monte Carlo. CRC press
Buchner J., 2016, Statistics and Computing, 26, 383
Carpenter B., et al., 2017, Journal of Statistical Software, Articles, 76, 1
Chopin N., Ridgway J., 2015, preprint, (arXiv:1506.08640)
Chopin N., Robert C. P., 2010, Biometrika, 97, 741
Diamond-Lowe H., Berta-Thompson Z., Charbonneau D., Kempton E. M. R., 2018, AJ, 156, 42
Douc R., Cappé O., Moulines E., 2005, arXiv e-prints, p. cs/0507025
Efron B., 1979, Ann. Statist., 7, 1
Espinoza N., Kossakowski D., Brahm R., 2018, arXiv e-prints, p. arXiv:1812.08549
Feigelson E. D., 2017, in Brescia M., Djorgovski S. G., Feigelson E. D., Longo G., Cavuoti S., eds, IAU Symposium Vol. 325, Astroinformatics. pp 3–9 (arXiv:1612.06238), doi:10.1017/S1743921317003453
Feroz F., Hobson M. P., 2008, MNRAS, 384, 449
Feroz F., Skilling J., 2013, in von Toussaint U., ed., American Institute of Physics Conference Series Vol. 1553, American Institute of Physics Conference Series. pp 106–113 (arXiv:1312.5638), doi:10.1063/1.4819989
Feroz F., Hobson M. P., Bridges M., 2009, MNRAS, 398, 1601
Feroz F., Hobson M. P., Cameron E., Pettitt A. N., 2013, preprint, (arXiv:1306.2144)
Fisher R. A., 1922, Philosophical Transactions of the Royal Society of London Series A, 222, 309
Foreman-Mackey D., 2016, The Journal of Open Source Software, 24
Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, PASP, 125, 306
Gaia Collaboration et al., 2018, A&A, 616, A1
Gelman A., Rubin D. B., 1992, Statistical Science, 7, 457
Geman S., Geman D., 1987, in Fischler M. A., Firschein O., eds, , Readings in Computer Vision. Morgan Kaufmann, San Francisco (CA), pp 564 – 584, doi:https://doi.org/10.1016/B978-0-08-051581-6.50057-X, http://www.sciencedirect.com/science/article/pii/B978008051581650057X
Goodman J., Weare J., 2010, Communications in Applied Mathematics and Computer Science, 5, 65
Guillochon J., Nicholl M., Villar V. A., Mockler B., Narayan G., Mandel K. S., Berger E., Williams P. K. G., 2018, The Astrophysical Journal Supplement Series, 236, 6
Günther M. N., et al., 2019, arXiv e-prints, p. arXiv:1903.06107
Handley W. J., Hobson M. P., Lasenby A. N., 2015, MNRAS, 450, L61
Hastings W., 1970, Biometrika, 57, 97
Heavens A., Fantaye Y., Mootoovaloo A., Eggers H., Hosenie Z., Kroon S., Sellentin E., 2017, arXiv e-prints, p. arXiv:1704.03472
Higson E., Handley W., Hobson M., Lasenby A., 2017a, arXiv e-prints, p. arXiv:1703.09701
Higson E., Handley W., Hobson M., Lasenby A., 2017b, arXiv e-prints, p. arXiv:1704.03459
Higson E., Handley W., Hobson M., Lasenby A., 2019, MNRAS, 483, 2044
Hoffman M. D., Gelman A., 2011, arXiv e-prints, p. arXiv:1111.4246
Hol J. D., Schon T. B., Gustafsson F., 2006, in 2006 IEEE Nonlinear Statistical Signal Processing Workshop. pp 79–82, doi:10.1109/NSSPW.2006.4378824

Hunter J. D., 2007, Computing in Science Engineering, 9, 90
Jasa T., Xiang N., 2012, Acoustical Society of America Journal, 132, 3251
Keeton C. R., 2011, MNRAS, 414, 1418
Lartillot N., Philippe H., 2006, Systematic Biology, 55, 195
Leja J., Johnson B. D., Conroy C., van Dokkum P. G., Byler N., 2017, ApJ, 837, 170
Leja J., Carnall A. C., Johnson B. D., Conroy C., Speagle J. S., 2018a, arXiv e-prints, p. arXiv:1811.03637
Leja J., et al., 2018b, arXiv e-prints, p. arXiv:1812.05608
Leja J., Johnson B. D., Conroy C., van Dokkum P., 2018c, ApJ, 854, 62
Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E., 1953, J. Chem. Phys., 21, 1087
Mukherjee P., Parkinson D., Liddle A. R., 2006, ApJ, 638, L51
Nagaraja H. N., 2006, Order Statistics from Independent Exponential Random Variables and the Sum of the Top Order Statistics. Birkhäuser Boston, Boston, MA, pp 173–185, doi:10.1007/0-8176-4487-3_11, https://doi.org/10.1007/0-8176-4487-3_11
Neal R. M., 2003, Ann. Statist., 31, 705
Neal R. M., 2012, arXiv e-prints, p. arXiv:1206.1901
Oliphant T. E., 2007, Computing in Science Engineering, 9, 10
Planck Collaboration et al., 2016, A&A, 594, A20
Plummer M., 2003, in Proceedings of the 3rd International Workshop on Distributed Statistical Computing.
Salomone R., South L. F., Drovandi C. C., Kroese D. P., 2018, arXiv e-prints, p. arXiv:1805.03924
Sharma S., 2017, Annual Review of Astronomy and Astrophysics, 55, 213
Shaw J. R., Bridges M., Hobson M. P., 2007, MNRAS, 378, 1365
Sivia D., Skilling J., 2006, Data analysis: a Bayesian tutorial. Oxford science publications, Oxford University Press, https://books.google.com/books?id=6O8ZAQAAIAAJ
Skilling J., 2004, in Fischer R., Preuss R., Toussaint U. V., eds, American Institute of Physics Conference Series Vol. 735, American Institute of Physics Conference Series. pp 395–405, doi:10.1063/1.1835238
Skilling J., 2006, Bayesian Anal., 1, 833
Skilling J., 2012, in Goyal P., Giffin A., Knuth K. H., Vrscay E., eds, American Institute of Physics Conference Series Vol. 1443, American Institute of Physics Conference Series. pp 145–156, doi:10.1063/1.3703630
Tak H., Ghosh S. K., Ellis J. A., 2018, MNRAS, 481, 277
Trotta R., 2008, Contemporary Physics, 49, 71
Vehtari A., Gelman A., Simpson D., Carpenter B., Bürkner P.-C., 2019, arXiv e-prints, p. arXiv:1903.08008
York D. G., et al., 2000, AJ, 120, 1579
Zucker C., Schlafly E. F., Speagle J. S., Green G. M., Portillo S. K. N., Finkbeiner D. P., Goodman A. A., 2018, ApJ, 869, 83
Zucker C., Speagle J. S., Schlafly E. F., Green G. M., Finkbeiner D. P., Goodman A. A., Alves J., 2019, arXiv e-prints, p. arXiv:1902.01425
van der Walt S., Colbert S. C., Varoquaux G., 2011, Computing in Science Engineering, 13, 22

APPENDIX A: DETAILED NESTED SAMPLING RESULTS

While we presented a broad overview of Nested Sampling in the main text, we glossed over much of the statistical background. We include more detailed results and discussion below.

The outline of these results are as follows. In §A1 we outline the basic setup for Nested Sampling. In §A2 we derive statistical properties in the single live point case. In §A3 we discuss the process of utilizing multiple live points. In §A4 we derive properties in the many live point case. In §A5 we extend these results to encompass varying numbers of live points. Finally, in §A6 we discuss various error properties of Nested Sampling as well as schemes to estimate them.

A1 Setup

Following Skilling (2006), Feroz et al. (2013), and others, we start by (re-)defining Bayes Rule

$$\mathcal{P}(\boldsymbol{\Theta}) = \frac{\mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})}{\mathcal{Z}} \tag{A1}$$

where $\mathcal{P}(\boldsymbol{\Theta})$ is the posterior, $\mathcal{L}(\boldsymbol{\Theta})$ is the likelihood, $\pi(\boldsymbol{\Theta})$ is the prior, and

$$\mathcal{Z}_M = \int_{\Omega_{\boldsymbol{\Theta}}} \mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \tag{A2}$$

is the evidence.

To evaluate this integral, Nested Sampling seeks to transform it from one over position $\boldsymbol{\Theta}$ to one over prior volume X where

$$X(\lambda) \equiv \int_{\Omega_{\boldsymbol{\Theta}}\,\mathrm{s.t.}\,\mathcal{L}(\boldsymbol{\Theta})>\lambda} \pi(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \equiv \int_{\Omega_{\boldsymbol{\Theta}}} \pi_\lambda(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \tag{A3}$$

defines the prior volume within a given iso-likelihood contour of level λ, assuming our priors are integrable, and

$$\pi_\lambda(\boldsymbol{\Theta}) \equiv \begin{cases} \pi(\boldsymbol{\Theta})/X(\lambda) & \mathcal{L}(\boldsymbol{\Theta}) \geq \lambda \\ 0 & \mathcal{L}(\boldsymbol{\Theta}) < \lambda \end{cases} \tag{A4}$$

is the constrained prior. Note that $X \in (0, 1]$ since the integral over the entire prior is $x(\lambda = 0) = 1$ while the value as $\lambda \to \infty$ should approach 0 if the maximum-likelihood value $\mathcal{L}_{\max}$ is a singular point.

Since $\lambda \in [0, \infty)$, this allows us to redefine the evidence integral as

$$\mathcal{Z} = \int_0^\infty X(\lambda)\mathrm{d}\lambda \tag{A5}$$

Provided the inverse $\mathcal{L}(X)$ of $X(\mathcal{L}(\boldsymbol{\Theta}) = \lambda)$ exists (i.e. there are no flat "slabs" of likelihood anywhere, only contours), we can rewrite this integral in terms of the prior volume associated with a particular iso-likelihood contour:

$$\mathcal{Z} = \int_0^1 \mathcal{L}(X)\mathrm{d}X \tag{A6}$$

This is now a 1-D integral over X that we can approximate using a discrete set of N points using, e.g., a Riemann sum

$$\hat{\mathcal{Z}} = \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{\Theta}_i) \times (X_i - X_{i-1}) \equiv \sum_{i=1}^{N} p(\boldsymbol{\Theta}_i) \tag{A7}$$

where $X_0 = 1$ and $p(\boldsymbol{\Theta}_i)$ is the (un-normalized) importance weight. These values can also be used to approximate the posterior:

$$\hat{\mathcal{P}}(\boldsymbol{\Theta}) = \frac{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i)}{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)} \tag{A8}$$

A2 Using a Single Live Point

Unfortunately, the exact value of $X(\lambda)$ at a given likelihood level $\lambda = \mathcal{L}(\boldsymbol{\Theta})$ is unknown. We can, however, construct an estimator $\hat{X}$ with a known statistical distribution. Looking back at the definition of the prior volume $X(\mathcal{L})$, we see that it defines a cumulative distribution function (CDF) over $\mathcal{L}$. We can then define the associated probability density function (PDF) for $\mathcal{L}$ as

$$P(\mathcal{L}) \equiv \frac{\mathrm{d}X(\mathcal{L})}{\mathrm{d}\mathcal{L}} = \frac{\mathrm{d}}{\mathrm{d}\mathcal{L}} \int_{\Omega_{\boldsymbol{\Theta}}} \pi_{\mathcal{L}}(\boldsymbol{\Theta}) \mathrm{d}\boldsymbol{\Theta} \tag{A9}$$

Assuming we can sample $\mathcal{L}$ from its PDF $P(\mathcal{L})$, we can use the Probability Integral Transform (PIT) to subsequently constrain the distribution of $X(\mathcal{L})$. In other words:

$$\mathcal{L}' \sim P(\mathcal{L}) \quad \Rightarrow \quad X(\mathcal{L}') \sim \mathrm{Unif} \tag{A10}$$

where $X \sim f(X)$ notation implies the random variable X is drawn from $f(X)$ and Unif is the standard Uniform distribution (i.e. flat from 0 to 1). This can be directly extended to cases where we are interested in sampling relative to a given threshold λ as

$$\mathcal{L}' \sim P(\mathcal{L}|\mathcal{L} > \lambda) \quad \Rightarrow \quad \frac{X(\mathcal{L}')}{X(\lambda)} \sim \mathrm{Unif} \tag{A11}$$

While this does not appear to make things any easier, it actually helps us out enormously. That's because, at fixed λ, $X(\lambda)$ is actually a CDF over the constrained prior $\pi_{\lambda}(\boldsymbol{\Theta})$. That means we can bypass λ and $P(\mathcal{L})$ altogether and just sample from $\pi_{\lambda}(\boldsymbol{\Theta})$ directly to satisfy the PIT:

$$\boldsymbol{\Theta}' \sim \pi_{\lambda}(\boldsymbol{\Theta}) \quad \Rightarrow \quad \frac{X(\mathcal{L}(\boldsymbol{\Theta}'))}{X(\lambda)} \sim \mathrm{Unif} \tag{A12}$$

Various methods for sampling from the constrained prior $\pi_{\lambda}(\boldsymbol{\Theta})$ subject to a suitable prior transform $\mathcal{T}$ (see §2.2) are outlined in §4.

Before moving on, we want to quickly note that while the above scheme is *sufficient* for generating values of $\mathcal{L}' \sim P(\mathcal{L})$ it is by no means *necessary*. As a counter-example, we can imagine a function $f(t) \rightarrow \boldsymbol{\Theta}_t$ that traces out a singular path through the distribution with support over $\mathcal{L}(f) \in [\mathcal{L}^{\min}, \mathcal{L}^{\max}]$. Let us furthermore assume that we construct $f(t)$ such that we spend more "time" t where the likelihood PDF is higher so that the PDF $P(t) \propto P(\mathcal{L}(\boldsymbol{\Theta}_t))$. Finally, let's define the constrained function $f_{\lambda}(t)$ to simply be the portion of the path with $\mathcal{L}(\boldsymbol{\Theta}_t) > \lambda$. While this path by no means encompasses the prior, it is clear that

$$t' \sim f_{\lambda}(t) \quad \Rightarrow \quad \frac{X(\mathcal{L}(\boldsymbol{\Theta}_{t'}))}{X(\lambda)} \sim \mathrm{Unif} \tag{A13}$$

This result proves we can in theory satisfy the PIT for Nested Sampling using correlated samples provided they probe enough of the *local* portion of the prior to obtain sufficient coverage over the range of possible likelihoods (see also Salomone et al. 2018). It also provides support for why Nested Sampling works so well in practice even when samples are not fully independent.

For a given prior volume X_{i-1} associated with a given likelihood level $\lambda_{i-1} = \mathcal{L}(\boldsymbol{\Theta}_{i-1})$ after $i-1$ iterations of this procedure, this implies the current prior volume X_i will be

$$\hat{X}_i = U_i \hat{X}_{i-1} = \prod_{j=1}^{i} U_j \tag{A14}$$

where

$$U_1, \ldots, U_i \overset{\mathrm{iid}}{\sim} \mathrm{Unif} \tag{A15}$$

are independent and identically distributed (iid) random variables drawn from the standard Uniform distribution and we have taken $X_0 \equiv X(\lambda = 0) = 1$. As we do not actually know the values of $U_1, \ldots, U_j$, we consider $\hat{X}_i$ to be a noisy estimator of X_i.

While sampling, we obviously need to assign a value for $\hat{X}_i$ to determine, e.g., whether to stop. While we can easily simulate random values of $U_1, \ldots, U_i$, if we want these values to be consistent then a reasonable choice is the expectation value (arithmetic mean):

$$\mathbb{E}\left[\hat{X}_i\right] = \mathbb{E}\left[\prod_{j=1}^{i} U_j\right] = \prod_{j=1}^{i} \mathbb{E}\left[U_j\right] = \left(\frac{1}{2}\right)^i \tag{A16}$$

Alternately, we might also be interested in the expectation value of $\ln \hat{X}_i$ (geometric mean):

$$\mathbb{E}\left[\ln \hat{X}_i\right] = \sum_{j=1}^{i} \mathbb{E}\left[\ln U_j\right] \sim -\sum_{j=1}^{i} \mathbb{E}\left[E_j\right] = -i \tag{A17}$$

where we have used the fact that

$$U \sim \mathrm{Unif} \Rightarrow -\ln U \sim \mathrm{Expo} \tag{A18}$$

where Expo is the standard Exponential distribution and

$$E_1, \ldots, E_i \overset{\mathrm{iid}}{\sim} \mathrm{Expo} \tag{A19}$$

Various stopping criteria are discussed in the main text (§2.4 and §3.4) and so are not discussed further here.

A3 Combining Live Points

Following Higson et al. (2017a), let's consider the case where we have two independent live points following the basic sampling approach described above. These each form a set of samples with increasing likelihood

$$\mathcal{L}_{N_1}^{[1]} > \cdots > \mathcal{L}_1^{[1]} > 0$$
$$\mathcal{L}_{N_2}^{[2]} > \cdots > \mathcal{L}_1^{[2]} > 0$$

where the $[\cdot]$ superscript notation indicates the index of the associated live point. We now want to "merge" these two sets of ordered samples together to get a single hypothetical ordered list:

$$\mathcal{L}_{N_1}^{[1]} > \mathcal{L}_{N_2}^{[2]} > \cdots > \mathcal{L}_2^{[1]} > \mathcal{L}_2^{[2]} > \mathcal{L}_1^{[2]} > \mathcal{L}_1^{[1]} > 0$$
$$\rightarrow \mathcal{L}_N > \cdots > \mathcal{L}_1 > 0$$

where $N = N_1 + N_2$.

Independently, we know that the prior volume at a given iteration for each live point is just

$$\hat{X}_i^{[j]} = \prod_{n=1}^{i} U_n^{[j]}$$

What we want to know, however, is the distribution of $\hat{X}_i$ of the *merged* list. Considering each sample independently implies $X_2 = X_1^{[2]}$ and $X_1 = X_1^{[1]}$ follow the same distribution (i.e. the first sampled prior volume for each run is similarly distributed). However, considering them *together* (based on

the merged list) implies $X_2 = X_1^{[2]}$ is *strictly less than* $X_1 = X_1^{[1]}$ since $\mathcal{L}_2 > \mathcal{L}_1$. This tells us that $\hat{X}_i$ *cannot* follow the same distribution from the associated independent runs that comprise it.

With this finding in hand, we now consider an approach for sampling from the prior volume using two live points. At each iteration i, we remove the one with the lowest likelihood $\lambda = \mathcal{L}_i^{\min}$ and replace it with a new point sampled from the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$. After N iterations, we will end up with a sorted list of likelihoods $\mathcal{L}_N > \cdots > \mathcal{L}_1 > 0$. If, however, we look at each live point individually (i.e. ignoring the other live point), we would find that each live point's evolution would comprise a list of independent samples with ordered likelihoods that would each be identical to $\mathcal{L}_{N_1}^{[1]} > \cdots > \mathcal{L}_1^{[1]} > 0$ and $\mathcal{L}_{N_2}^{[2]} > \cdots > \mathcal{L}_1^{[2]} > 0$, respectively! Therefore, we see that this procedure for sampling with two live points is identical to combining two sets of independent samples derived using one live point each.

The above procedure can be immediately generalized to K live points, producing the (Static) Nested Sampling procedure outlined in Algorithm 1. We will return to this duality between K independent Nested Sampling runs and a single Nested Sampling run with K live points in §A6.

A4 Using Many Live Points

Now that we have established a procedure for running Nested Sampling with K live points, we need to characterize how this affects our estimates $\hat{X}_i$ of the prior volume. At any given iteration i, we know that the current set of prior volumes $\{X_i^{[1]}, \ldots, X_i^{[K]}\}$ associated with our K live points are uniformly distributed within the prior volume from the previous iteration X_{i-1} so that

$$X_i^{[j]} = U^{[j]} X_{i-1} \tag{A20}$$

where

$$U^{[1]}, \ldots, U^{[K]} \overset{\text{iid}}{\sim} \text{Unif} \tag{A21}$$

We are now want to replace the live point with the lowest likelihood $\mathcal{L}_i^{\min}$ corresponding to the largest prior volume. This means we are now interested in the *ordered* list of prior volumes

$$X_i^{(j)} = U^{(j)} X_{i-1} \tag{A22}$$

where (j) now indicates the position in the *ordered list* (from smallest to largest) rather than the live point index $[j]$ and

$$U^{(j)} = \min_j \left(\left\{ U^{[1]}, \ldots, U^{[K]} \right\} \right) \tag{A23}$$

is the jth standard uniform order statistic, where $\min_j$ selects the jth smallest point (so $j = 1$ is the smallest and $j = K$ is the largest).

Using the Renyí Representation, it can be shown (Nagaraja 2006) that we can represent the *joint* distribution of our K standard uniform order statistics $\{U^{(1)}, \ldots, U^{(K)}\}$ such that

$$U^{(j)} = \frac{\sum_{n=1}^{j} E_n}{\sum_{n=1}^{K+1} E_n} \tag{A24}$$

where

$$E_1, \ldots, E_{K+1} \overset{\text{iid}}{\sim} \text{Expo} \tag{A25}$$

The marginal distribution for $U^{(j)}$ is then (Blitzstein & Hwang 2014):

$$U^{(j)} \sim \text{Beta}\,(j, K + 1 - j) \tag{A26}$$

where $\text{Beta}\,(\alpha, \beta)$ is the Beta distribution.

Using these results, we see that the prior volume based on K live points at iteration i evolves as

$$\hat{X}_i = \prod_{j=1}^{i} U_j^{(K)} \tag{A27}$$

where $U_1^{(K)}, \ldots, U_i^{(K)}$ are iid draws of the Kth standard uniform order statistic with marginal distribution $\text{Beta}\,(K, 1)$. The arithmetic mean is

$$\mathbb{E}\left[\hat{X}_i\right] = \prod_{j=1}^{i} \mathbb{E}\left[U_j^{(K)}\right] = \left(\frac{K}{K+1}\right)^i \tag{A28}$$

The geometric mean is

$$\mathbb{E}\left[\ln \hat{X}_i\right] = \sum_{j=1}^{i} \mathbb{E}\left[\ln U_j^{(K)}\right] = -\frac{i}{K} \tag{A29}$$

As discussed in §2.3, after we terminate sampling we can add the final set of K live points to our set of N samples. These will then just follow the final set of $\{U^{(1)}, \ldots, U^{(K)}\}$ standard uniform order statistics relative to $\hat{X}_N$ with an arithmetic mean

$$\mathbb{E}\left[\hat{X}_{N+k}\right] = \left(\frac{K+1-k}{K+1}\right)\left(\frac{K}{K+1}\right)^N \tag{A30}$$

and geometric mean

$$\mathbb{E}\left[\ln \hat{X}_{N+k}\right] = -\frac{N}{K} - \left[\psi(K+1) - \psi(K+1-k)\right] \tag{A31}$$

where $\psi(\cdot)$ is the digamma function.

A5 Using a Varying Number of Live Points

As discussed in §3, there's no inherent reason why the number of number of live points must remain constant from iteration to iteration. Indeed, we can interpret adding the final set of live points to the list of samples from §A4 as simply allowing the nested sampling run to continue while continually decreasing the number of live points. From this viewpoint, we have $K_1 = \cdots = K_N = K$ live points over iteration $i = 1$ to N, but only $K_{N+k} = K+1-k$ live points at iteration $i = N+k$.

The change in the number of live points also changes the overall behavior of the Nested Sampling run before and after adding the final set of live points. We can highlight these by rewriting the results from §A4 as:

$$\ln \mathbb{E}\left[\hat{X}_{N+k}\right] = \underbrace{\sum_{i=1}^{N} \ln\left(\frac{K}{K+1}\right)}_{\text{Exponential Shrinkage}} + \underbrace{\sum_{j=1}^{k} \ln\left(\frac{K+1-k}{K+2-k}\right)}_{\text{Uniform Shrinkage}} \tag{A32}$$

This neatly decomposes the two "modes" in which Nested Sampling can traverse the prior. While "replacing" the worst live point (i.e. $K_i = K_{i-1}$), the prior volume shrinks *exponentially* by a constant factor at each iteration. However, when

"removing" live points (i.e. $K_i < K_{i-1}$), we instead shrink *uniformly* by a variable factor at each iteration.

We can now generalize this behavior to the case where K_i is allowed to vary at each iteration (Higson et al. 2017b). This now generates two distinct classes of behavior. When $K_i \geq K_{i-1}$, we add $K_i - K_{i-1} \geq 0$ live points to our existing set of live points, after which we replace the one with the worst likelihood $\mathcal{L}_i^{\min}$. This then gives a distribution for the prior volume shrinkage of $\mathrm{Beta}(K_i, 1)$.

If $K_i < K_{i-1}$, on the other hand, we instead have removed $K_{i-1} - K_i$ live points from the previous set of live points. The expected shrinkage is then based on the associated K_i standard uniform order statistic $U^{(K_i)}$ from the initial set of K_{i-1} values. Although in theory we should consider cases where the number of live points can decrease by an arbitrary amount, in practice when following iterative schemes such as the one outlined in Algorithm 3 we only need to consider the case where $K_{i-1} - K_i = 1$.

Taken together, these two types of behavior then give a mean estimate of:

$$\begin{aligned}\ln \mathbb{E}\left[\hat{X}_j\right] = &\sum_{i=1}^{n_1} \ln\left(\frac{K_i}{K_i+1}\right) + \sum_{i=1}^{n_2} \ln\left(\frac{K_{N_1}+1-i}{K_{N_1}+2-i}\right) \\ &+ \sum_{i=1}^{n_3} \ln\left(\frac{K_{N_2+i}}{K_{N_2+i}+1}\right) + \cdots + \sum_{i=1}^{n_{M-1}} \ln\left(\frac{K_{N_{M-2}+i}+1-i}{K_{N_{M-2}+i}+1}\right) \\ &+ \sum_{i=1}^{n_M} \ln\left(\frac{K_{N_{M-1}}+1-i}{K_{N_{M-1}}+2-i}\right)\end{aligned} \tag{A33}$$

where n_m is the number of contiguous samples for which either exponential or uniform shrinkage dominates, $N_m = \sum_{k=1}^{i} n_k$ is the total number of iterations that have occurred up to that point, and M is the number of contiguous regions prior to iteration $j = N_M$ where one mode of shrinkage dominates. Note that for illustrative purposes here we have assumed the final samples are experiencing uniform shrinkage.

To summarize, varying the number of live points at each iteration simply involves dynamically switching between exponential and uniform shrinkage over the course of a Nested Sampling run. While this adds additional bookkeeping, it remains straightforward to estimate the prior volume $\hat{X}_i$ at any particular iteration.

A6 Nested Sampling Errors

We now turn our attention to characterizing various error properties of Nested Sampling, following the basic approach of Higson et al. (2017a, 2019). Similar to other sampling approaches, we expect some amount of "sampling noise" in our evidence $\hat{\mathcal{Z}}$ and posterior estimates $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ arising from the fact that we are approximating a continuous distribution (and smooth integral) with a discrete set of N samples. We expect that as the number of live points at each iteration $K_i \to \infty$ such that change in prior volume $X_i - X_{i-1} \to 0$ and the total number of samples $N \to \infty$, these sampling errors will become negligible.

Unlike other sampling approaches such as Markov Chain Monte Carlo (MCMC), however, Nested Sampling, contains an *additional* source of noise arising from our use of noisy estimators $X \to \hat{X}_i$ of the prior volume at a given iteration i (Skilling 2006). This "statistical noise" translates to a noisy estimator of the importance weight $p(\boldsymbol{\Theta}_i) \to \hat{p}(\boldsymbol{\Theta}_i)$, which in turn gives noisy estimators for our previous evidence estimate

$$\begin{aligned}\hat{\mathcal{Z}} &= \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{\Theta}_i) \times (X_i - X_{i-1}) \\ &\approx \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{\Theta}_i) \times (\hat{X}_i - \hat{X}_{i-1}) \equiv \sum_{i=1}^{N} \hat{p}(\boldsymbol{\Theta}_i)\end{aligned} \tag{A34}$$

and our previous posterior estimate

$$\hat{\mathcal{P}}(\boldsymbol{\Theta}) = \frac{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i)}{\sum_{i=1}^{N} p(\boldsymbol{\Theta}_i)} \approx \frac{\sum_{i=1}^{N} \hat{p}(\boldsymbol{\Theta}_i)\delta(\boldsymbol{\Theta}_i)}{\sum_{i=1}^{N} \hat{p}(\boldsymbol{\Theta}_i)} \tag{A35}$$

Similar with the sampling noise, we also expect the statistical noise to become negligible as the number of live points at each iteration $K_i \to \infty$ such that our estimate $\hat{X}_i \to X_i$ and the total number of samples $N \to \infty$.

We can highlight the decomposition of these two noise sources by considering trying to evaluate the expectation value of a target function $f(\boldsymbol{\Theta})$ with respect to the posterior (Chopin & Robert 2010; Higson et al. 2017a):

$$\mathbb{E}_{\mathcal{P}}[f] = \int_{\Omega_{\boldsymbol{\Theta}}} f(\boldsymbol{\Theta})\mathcal{P}(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} = \frac{1}{\mathcal{Z}} \int_0^1 \tilde{f}(X)\mathcal{L}(X)\mathrm{d}X \tag{A36}$$

where

$$\tilde{f}(X) = \mathbb{E}_{\pi}[f(\boldsymbol{\Theta}) | \mathcal{L}(\boldsymbol{\Theta}) = \mathcal{L}(X)] \tag{A37}$$

is the expectation value of $f(\boldsymbol{\Theta})$ on the associated iso-likelihood contour $\mathcal{L}(\boldsymbol{\Theta}) = \mathcal{L}(X)$ with respect to the prior $\pi(\boldsymbol{\Theta})$. Using the same Riemann sum approximation as §A1, Nested Sampling would approximate this integral as:

$$\mathbb{E}_{\mathcal{P}}[f] \approx \sum_{i=1}^{N} \tilde{f}(X_i) \frac{\mathcal{L}(X_i)(X_i - X_{i-1})}{\mathcal{Z}} = \sum_{i=1}^{N} \tilde{f}(X_i)p(X_i) \tag{A38}$$

$$\approx \sum_{i=1}^{N} f(\boldsymbol{\Theta}_i)p(X_i) \tag{A39}$$

$$\approx \sum_{i=1}^{N} f(\boldsymbol{\Theta}_i)\hat{p}(\boldsymbol{\Theta}_i) \tag{A40}$$

We can see the two error types enter in cleanly through the final two approximations. In equation (A39), we introduce sampling noise by replacing $\tilde{f}(X)$, which is averaged over the entire iso-likelihood contour, with the estimate $f(\boldsymbol{\Theta}_i)$ evaluated at a single point. Then, in equation (A40), we replace the true importance weight $p(X_i)$ at a given prior volume with its noisy estimate $\hat{p}(\boldsymbol{\Theta}_i)$ based on our noisy estimators for the prior volume $\hat{X}_i$.

A6.1 Statistical Uncertainties

In §A2, §A4, and §A5, we derived the analytic distribution for our prior volume estimator $\hat{X}_i$ at iteration i under a variety of assumptions. While the distribution for $\hat{\mathcal{Z}}$ and $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ is not analytic, it is straightforward to draw from them. First, we simulate values of the prior volumes

$$\hat{X}'_1, \ldots, \hat{X}'_N \sim P(\hat{X}_1, \ldots, \hat{X}_N) \tag{A41}$$

by drawing a combination of $\mathrm{Beta}(K_i, 1)$-distributed random variables (when $K_i \geq K_{i-1}$) and standard uniform order

statistics (when $K_i < K_{i-1}$) and iteratively computing each $\hat{X}'_i$ using the procedures outlined earlier. Then, we simply compute the corresponding evidence $\hat{\mathcal{Z}}'$ and posterior $\hat{\mathcal{P}}'(\boldsymbol{\Theta})$ estimates.

While we can simulate the prior volumes and trace their impact on $\hat{\mathcal{Z}}$ and $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ explicitly, it is also helpful to derive a rough estimate of their impact. Since the posterior $\mathcal{P}(\boldsymbol{\Theta})$ can be arbitrarily complex, we will focus on the evidence $\mathcal{Z}$ for which this analysis is more tractable.

There have previously been two main approaches for deriving the uncertainty, which focus either on trying to derive $\mathbb{V}\left[\hat{\mathcal{Z}}\right]$ (Keeton 2011) or $\mathbb{V}\left[\ln\hat{\mathcal{Z}}\right]$ (Skilling 2006). Here we will focus on the latter, which gives a cleaner (if less precise) result.

We first start with the Static Nested Sampling case using a constant number of live points K. To estimate the evidence $\mathcal{Z}$, we must integrate over the unnormalized posterior $\mathcal{P}(\boldsymbol{\Theta}) \propto \pi(\boldsymbol{\Theta})\mathcal{L}(\boldsymbol{\Theta})$. This occurs after a certain number of iterations N have passed given a fixed stopping criterion.

There are two factors that contribute to the overall N. The first is the rate of integration: at any given iteration i, the prior volume decreases by $\Delta\ln X \approx 1/K$. As a result, it must be the case that $N \propto 1/K$.

The second is the total amount of prior volume that needs to be integrated over. This roughly scales as the Kullback-Leibler (KL) divergence (i.e. "information gain") between the prior $\pi(\boldsymbol{\Theta})$ and posterior $\mathcal{P}$

$$H(\mathcal{P}||\pi) \equiv H \equiv \int_{\Omega_{\boldsymbol{\Theta}}} \mathcal{P}(\boldsymbol{\Theta}) \ln\frac{\mathcal{P}(\boldsymbol{\Theta})}{\pi(\boldsymbol{\Theta})} \mathrm{d}\boldsymbol{\Theta} \tag{A42}$$

$$= \frac{1}{\mathcal{Z}}\int_0^1 \mathcal{L}(X)\ln\mathcal{L}(X)\mathrm{d}X - \ln\mathcal{Z} \tag{A43}$$

Since N is a discrete number that is typically large, it is reasonable to assume that it follows a Poisson distribution such that

$$\mathbb{E}[N] = \mathbb{V}[N] \sim \frac{H}{\Delta\ln X} \tag{A44}$$

This leads to a rough uncertainty in $\ln\hat{\mathcal{Z}}$ of

$$\sigma\left[\ln\hat{\mathcal{Z}}\right] \sim \sigma\left[\ln\hat{X}_N\right] \sim \sigma\left[\ln N\right](\Delta\ln X)$$

$$\sim \sqrt{H(\Delta\ln X)} = \sqrt{\frac{H}{K}} \tag{A45}$$

We now extend this result to encompass a variable number of live points K_i at each iteration. We first rewrite our estimate of the variance as

$$\mathbb{V}\left[\ln\hat{\mathcal{Z}}\right] = \mathbb{V}\left[\sum_{i=1}^{N}\left(\ln\hat{\mathcal{Z}}_i - \ln\hat{\mathcal{Z}}_{i-1}\right)\right] \equiv \mathbb{V}\left[\sum_{i=1}^{N}\Delta\ln\hat{\mathcal{Z}}_i\right]$$

$$\approx \sum_{i=1}^{N}\mathbb{V}\left[\Delta\ln\hat{\mathcal{Z}}_i\right] \tag{A46}$$

where the final approximation assumes the distribution of evidence updates is independent at each iteration i and $\ln\hat{\mathcal{Z}}_0 = 0$. If we further assume that the distribution of the actual evidence estimates $\hat{\mathcal{Z}}_i$ themselves are roughly independent at each iteration i and that the number of live points K_i changes sufficiently slowly such that $\Delta\ln X_i \approx \Delta\ln X_{i-1}$, we find

$$\mathbb{V}\left[\Delta\ln\hat{\mathcal{Z}}_i\right] \approx \mathbb{V}\left[\ln\hat{\mathcal{Z}}_i\right] - \mathbb{V}\left[\Delta\ln\hat{\mathcal{Z}}_{i-1}\right]$$

$$\sim (H_i - H_{i-1})(\Delta\ln X_i) \equiv (\Delta H_i)(\Delta\ln X_i) \tag{A47}$$

Substituting this in to our original expression and taking $\Delta\ln X_i \approx 1/K_i$ then gives a modified error estimate

$$\sigma\left[\ln\hat{\mathcal{Z}}\right] \sim \sqrt{\sum_{i=1}^{N}\frac{\Delta H_i}{K_i}} \tag{A48}$$

While the modified estimator in equation (A48) is less reliable that our original estimate, it is somewhat reassuring that in the special case $K_1 = \cdots = K_N = K$ it reduces to the original estimator derived in equation (A45).

A6.2 Sampling Uncertainties

Unlike the statistical uncertainties on the prior volume estimator $\hat{X}_i$, we do not have analytic expression or ways to explicitly simulate from the distribution characterized by our sampling uncertainties. In fact, it is doubtful we will ever have access to these except in special cases, since they rely on having access to the distribution of all possible paths (or varying lengths) live points can take through the distribution over the course of a Nested Sampling run.

This however, does not mean we cannot attempt to construct an estimate of this distribution. To do this, we follow Higson et al. (2017a) and turn to bootstrapping, which serves as a generic and robust tool for attempting to simulate the impact of sampling uncertainties with limited support (Efron 1979). Since in most cases we have many thousands of samples from our distribution and sample with $K > 100$ live points, Nested Sampling is almost always in a regime where bootstrapping should be viable.

Naively, we might expect to simply be able simulate values of, e.g., $\hat{\mathcal{Z}}$ by just bootstrapping the underlying set of live points. However, this leads to three immediate complications:

(i) This approach creates multiple samples at the same position. It is unclear how these points need to be ordered to assign them associated prior volumes.

(ii) This approach conserves the total number of samples N, which clearly must be allowed to change if we really want to simulate from all possible live point paths (with varying path-lengths).

(iii) This approach can leave out samples initially drawn from the prior. These points are crucial for establishing the normalization needed to estimate the evidence, and so removing them drastically distorts our evidence estimates.

We address each of these in turn.

First, the ambiguous ordering, while at first glance a serious issue, is actually a non-concern since the impact on any derived quantity is actually completely insensitive to the ordering. For the evidence, since the likelihood $\mathcal{L}(\boldsymbol{\Theta})$ is identical among the points, their contribution to $\hat{\mathcal{Z}}$ will remain unchanged. Likewise, because they occupy the same position $\boldsymbol{\Theta}$, their contribution to the posterior estimate $\hat{\mathcal{P}}(\boldsymbol{\Theta})$ is also unchanged. This implies that any ordering scheme (e.g., random) will suffice.

To resolve the second issue, we now turn to the problem of simulating all possible live point paths along with their possibly varying path-lengths. Bootstrapping over all the samples by construction destroys this information by ignoring the paths of each individual live point. Analogous to the discussion in §A3, we can characterize these individual paths as being the collection of K lists of positions $\boldsymbol{\Theta}_1^{[j]} \rightarrow \cdots \rightarrow \boldsymbol{\Theta}_{N_j}^{[j]}$ traversed by each live point. Sampling from the space of all possible live point paths thus is equivalent to bootstrapping from these individual K "strands" and then merging the K resampled strands $\{\ldots, \{\boldsymbol{\Theta}_1^{[j']} \rightarrow \cdots \rightarrow \boldsymbol{\Theta}_{N_{j'}}^{[j']}\}, \ldots\}$ into a new Nested Sampling run.

Unfortunately, this procedure still can run afoul of the third issue when the number of live points K_i is not constant. Going back to the discussion in §A3 and the Iterative Dynamic Nested Sampling scheme outlined in Algorithm 3, we see that increasing the number of live points at some iteration $i > 1$ means that those additional live points *were sampled interior to the prior* at some associated likelihood threshold $\mathcal{L}(\boldsymbol{\Theta}_i)$. Since these live points provide no information about the overall normalization (only the normalization relative to $\hat{X}_i$), they are totally uninformative on their own when it comes to estimating the evidence $\hat{\mathcal{Z}}$.

To account for this, we need to perform a *stratified* bootstrap over the set of $K_{\rm int}$ "interior" strands (i.e. strands with starting positions interior to the prior) and $K_{\rm anc}$ "anchor" strands (i.e. strands sampled directly from the prior that "anchor" the interior strands). Once the set of $K_{\rm int}$ interior strands and $K_{\rm anc}$ strands have been resampled, we can then merge the new collection into a new Nested Sampling run. Following this scheme is then sufficient for simulating the evidence $\hat{\tilde{\mathcal{Z}}}$ and posterior $\hat{\tilde{\mathcal{P}}}(\boldsymbol{\Theta})$ estimates, where we have used $\hat{\tilde{\mathcal{Z}}}$ notation to indicate a we used bootstrapping rather than prior volume simulation.

Note that one interesting corollary of our bootstrap estimates is that we expect the total number of samples $\tilde{N}$ to change. For a sufficient number of live points, this distribution is likely to be roughly Poisson. Assuming that the associated $\Delta\tilde{H}_i \approx \Delta H_i$ and $\tilde{K}_i \approx K_i$ from our bootstrapped Nested Sampling run are similar to the original, this immediately leads us to an estimate of $\sigma\left[\ln\hat{\mathcal{Z}}\right]$ identical to that in equation (A48). Although it has the exact same form, note that this error term *is completely independent* from the previous case.

A6.3 Combined Uncertainties

The full uncertainties associated with a given Nested Sampling run involve both the statistical uncertainties described in §A6.1 and the sampling uncertainties described in §A6.2. Simulating from this combined error distribution is straightforward and can be done by the following procedure:

(i) Resample the set of underlying $K_{\rm anc}$ anchor and $K_{\rm int}$ interior strands using stratified bootstrap resampling.
(ii) Merge the resampled strands into a single run.
(iii) Simulate the values of the prior volumes.

We can then calculate the evidence $\hat{\tilde{\mathcal{Z}}}'$ and posterior $\hat{\tilde{\mathcal{P}}}'(\boldsymbol{\Theta})$ estimates accordingly. The combined uncertainty on the evidence that we estimate from both sources is then roughly

$$\sigma\left[\ln\hat{\mathcal{Z}}\right] \sim \sqrt{2\sum_{i=1}^{N}\frac{\Delta H_i}{K_i}} \tag{A49}$$

based on the identical error estimates derived in §A6.1 and §A6.2.

This paper has been typeset from a TEX/LATEX file prepared by the author.