Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation

Edward Higson, Will Handley, Mike Hobson, Anthony Lasenby
University of Cambridge, Cambridge, UK
E-mail: e.higson@mrao.cam.ac.uk

Summary. We introduce dynamic nested sampling: a generalisation of the nested sampling algorithm in which the number of “live points” varies to allocate samples more efficiently. In empirical tests the new method increases parameter estimation accuracy by up to a factor of \( \sim 8 \) compared to standard nested sampling with the same number of samples — equivalent to speeding up the computation by a factor of \( \sim 64 \). We also show that the accuracy of Bayesian evidence estimates can be improved, and that both can be improved simultaneously. Unlike in standard nested sampling more accurate results can be obtained by continuing the calculation for longer. We have implemented dynamic nested sampling in the publicly available PERFECTNS package, which is used for the numerical tests in this paper. Our approach can be easily included in popular nested sampling software packages such as MULTINEST and POLYCHORD.

Keywords: nested sampling, parameter estimation, Bayesian evidence

1. Introduction

Nested sampling (Skilling, 2006) is a numerical method for Bayesian computation which simultaneously provides both posterior samples and Bayesian evidence estimates. The original development of the algorithm was motivated by evidence calculation, but the MULTINEST (Feroz and Hobson, 2008; Feroz et al., 2008, 2013) and POLYCHORD (Handley et al., 2015a,b) software packages are now extensively used for parameter estimation from posterior samples (for example in DES Collaboration, 2017). Nested sampling performs well compared to Markov chain Monte Carlo (MCMC)-based parameter estimation for multi-modal and degenerate posteriors, and the POLYCHORD implementation is well suited to high dimensional parameter estimation problems.

Nested sampling maintains a set of samples from the prior, called live points, and iteratively updates them subject to the constraint that new samples have increasing likelihoods. Conventionally a fixed number of live points is used — we term this standard nested sampling. In this case the expected fractional shrinkage of the prior volume remaining is the same at each step, and as a result many samples are typically taken from regions of the prior that are remote from the bulk of the posterior. The allocation of samples in standard nested sampling is set by the likelihood and the prior, and cannot be changed depending on whether calculating the evidence or obtaining posterior samples is the primary goal.

We propose modifying the nested sampling algorithm by dynamically varying the number of live points in order to maximise the accuracy of a calculation for some number
of posterior samples, subject to practical constraints. We term this more general approach dynamic nested sampling, with standard nested sampling representing the special case where the number of live points is constant. Dynamic nested sampling is particularly effective for parameter estimation, as standard nested sampling typically spends most of its computational effort iterating towards the posterior peak. This produces posterior samples with negligible weights which make little contribution to the calculation’s accuracy, as discussed in our previous analysis of sampling errors in nested sampling parameter estimation (Higson et al., 2017). We also achieve significant improvements in the accuracy of evidence calculations, and show both evidence and parameter estimates can be improved simultaneously. Our approach can be easily incorporated into existing standard nesting sampling software such as MultiNest and POLYCHORD; we are currently working on its inclusion in the forthcoming POLYCHORD 2 package.

Dynamically weighting sampling in order to target regions of higher posterior mass has been used in the statistical physics literature, such as in multi-canonical sampling (see for example Okamoto, 2004). Other variants of nested sampling include diffusive nested sampling (Brewer et al., 2011) and superposition enhanced nested sampling (Martiniani et al., 2014), which have been implemented as stand alone software packages. In particular, dynamic nested sampling shares some similarities with DNeST4 (Brewer and Foreman-Mackey, 2016), in which diffusive nested sampling is followed by additional sampling targeting regions of high posterior mass. In the case of nested sampling on unimodal posteriors, efficiency can be greatly increased using nested importance sampling (Chopin and Robert, 2010) or by performing nested sampling using an auxiliary prior which approximates the posterior as described in Cameron and Pettitt (2014). Dynamic nested sampling differs from these alternatives in that it can be used to improve the efficiency of popular standard nested sampling software such as MultiNest and POLYCHORD, in particular in high dimensional problems, while maintaining their strengths in sampling degenerate and multimodal distributions.

The paper proceeds as follows: Section 2 contains background on nested sampling, and Section 3 establishes useful results about the effects of varying the number of live points. Our dynamic nested sampling algorithm for increasing efficiency for general nested sampling calculations is presented in Section 4; its accurate allocation of live points for a priori unknown posterior distributions is illustrated in Figure 4. In the manner described by Keeton (2011) we test dynamic nested sampling using analytical cases where one can obtain uncorrelated samples from the prior space within some likelihood contour using standard techniques, and we term the resulting procedure perfect nested sampling (in both standard and dynamic versions). Numerical tests of dynamic nested sampling’s performance compared to standard nested sampling use our publicly available PERFECTNS package, and are described in Section 5. This includes a discussion of the effects of likelihood, priors and dimensionality on the improvements from dynamic nested sampling; in particular we find large efficiency gains for high-dimensional parameter estimation problems. We discuss including dynamic nested sampling in nested sampling software for use on practical problems such as multimodal posteriors in Section 6.
2. Background: the nested sampling algorithm

This section provides background on the nested sampling algorithm based on Higson et al. (2017), and sets out our notation. For a more detailed discussion see Higson et al. (2017) and Skilling (2006).

For a given likelihood $L(\theta)$ and prior $\pi(\theta)$, nested sampling will simultaneously compute the Bayesian evidence

$$Z = \int L(\theta)\pi(\theta)\,d\theta$$

(1)

and samples from the posterior distribution

$$P(\theta) = \frac{L(\theta)\pi(\theta)}{Z}.$$  

(2)

The algorithm begins by sampling some number of live points randomly from the prior $\pi(\theta)$. In standard nested sampling, at each iteration $i$ the point with the lowest likelihood $L_i$ is replaced by a new point sampled from the region of prior with likelihood $L(\theta) > L_i$ and the number of live points remains constant throughout the process. This process is continued until some termination condition is met, producing a list of samples (referred to as dead points) which are then used for evidence and parameter estimation. We term the finished nested sampling process a run.

Nested sampling calculates the evidence (1) as a one-dimensional integral

$$Z = \int_0^1 L(X)\,dX,$$

(3)

where $X(L)$ is the fraction of the prior with likelihood greater than $L$ and $L(X) \equiv X^{-1}(L)$. The prior volumes $X_i$ corresponding to the dead points $i$ are unknown but can be modelled statistically as $X_i = t_i X_{i-1}$, where $X_0 = 1$. For a given number of live points $n$, each shrinkage ratio $t_i$ is independently distributed as the largest of $n$ random variables from the interval $[0, 1]$ and so (Higson et al., 2017):

$$P(t_i) = nt_i^{n-1}, \quad \mathbb{E}[\log t_i] = -\frac{1}{n}, \quad \text{Var}[\log t_i] = \frac{1}{n^2}.$$  

(4)

In standard nested sampling the number of live points $n$ is some constant value for all $t_i$ — the iteration of the algorithm in this case is illustrated schematically in Figure 1.

2.1. Evidence estimation

Nested sampling calculates the evidence (3) as a quadrature sum over the dead points

$$Z(\mathbf{t}) \approx \sum_{i \in \text{dead}} L_i w_i(\mathbf{t}),$$

(5)

where $\mathbf{t} = \{t_1, t_2, \ldots, t_{\text{dead}}\}$ are the unknown set of shrinkage ratios for each dead point and each $t_i$ is an independent random variable with distribution (4). The $w_i$ are appropriately chosen quadrature weights; we use the trapezium rule such that $w_i(\mathbf{t}) = \frac{1}{2}(X_{i-1}(\mathbf{t}) - X_{i+1}(\mathbf{t}))$, where $X_i(\mathbf{t}) = \prod_{k=0}^i t_k$. Given that the shrinkage ratios $\mathbf{t}$ are a
priori unknown, one typically calculates an expected value and error on the evidence (5) using the distribution (4). The dominant source of error in evidence estimates from perfect nested sampling is the statistical variation in the unknown volumes of the prior “shells” \( w_i(t) \).

2.2. Parameter estimation

Nested sampling parameter estimation uses the dead points to construct a set of posterior samples with weights proportional to their share of the posterior mass:

\[
p_i(t) = \frac{w_i(t) \mathcal{L}_i}{\sum_i w_i(t) \mathcal{L}_i} = \frac{w_i(t) \mathcal{L}_i}{\mathcal{Z}(t)},
\]

Neglecting any implementation-specific errors not present in perfect nested sampling, the dominant sampling errors in estimating some parameter or function of parameters \( f(\theta) \) come from two sources Higson et al. (2017):

(i) approximating the relative point weights \( p_i(t) \) with their expectation \( \mathbb{E}[p_i(t)] \) using (4);

(ii) approximating the mean value of a function of parameters over an entire iso-likelihood contour with its value at a single point \( f(\theta_i) \).

2.3. Combining and dividing nested sampling runs

Skilling (2006) describes how several standard nested sampling runs \( r = 1, 2, \ldots \) with constant live points \( n^{(r)} \) may be combined simply by merging the dead points and sorting by likelihood value. The combined sequence of dead points is equivalent to a single nested sampling run with \( n_{\text{combined}} = \sum_r n^{(r)} \) live points.
Higson et al. (2017) gives an algorithm for the reverse procedure — decomposing a nested sampling run with \( n \) live points into a set of \( n \) valid nested sampling runs, each with 1 live point. These single live point runs, which we term *threads*, are the smallest unit from which valid nested sampling runs can be constructed and will prove useful in developing dynamic nested sampling.

3. Variable numbers of live points

Before presenting our algorithm for dynamic nested sampling in Section 4, we first establish some basic results for a nested sampling run in which the number of live points varies. Such runs are valid as successive shrinkage ratios \( t_i \) are independently distributed (Skilling, 2006).

Let us define \( n_i \) as the number of live points present for the prior shrinkage ratio \( t_i \) between dead points \( i-1 \) and \( i \). In this notation all information about the number of live points for a nested sampling run can be expressed as a list of numbers \( n = \{ n_1, n_2, \ldots, n_{n\text{dead}} \} \) which correspond to the shrinkage ratios \( t = \{ t_1, t_2, \ldots, t_{n\text{dead}} \} \). Nested sampling calculations for variable numbers of live points differ from the constant live point case only in the use of different \( n_i \) in calculating the distribution of each \( t_i \) from (4).

Skilling (2006)’s method for combining constant live point runs, mentioned in Section 2.3, can be extended to accommodate variable numbers of live points by requiring that at any likelihood the live points of the combined run equals the sum of the live points of the constituent runs at that likelihood, as illustrated in Figure 2. Conversely if variable live point runs are divided into their constituent threads, some threads will start and finish part way through the run. On iso-likelihood contours when the number of live points is increased there is no longer a unique division into threads as there is for standard nested sampling.

In addition the variable live point framework provides a natural way to include the final set of live points remaining when a standard nested sampling run terminates in a calculation. These are uniformly distributed in the region of the prior with \( L(\theta) > L_{\text{terminate}} \), and can be treated as samples from a dynamic nested sampling run with the number of live points reducing by 1 as each of the points remaining after termination is passed until the final point \( i \) has \( n_i = 1 \). This allows the final live points of standard nested sampling runs to be combined with variable live point runs.

The remainder of this section analyses the effects of local variations in the number of live points on the accuracy of nested sampling evidence calculation and parameter estimation. The dynamic nested sampling algorithm in Section 4 uses these results to allocate additional live points.

†In order for (4) to be valid, the number of live points must remain constant across the shrinkage ratios \( t_i \) between successive dead points. We therefore only allow the number of live points to change on iso-likelihood contours \( L(\theta) = L_i \) where a dead point \( i \) is present. This restriction has negligible effects for typical calculations, and is automatically satisfied by most nested sampling implementations.
3.1. Effects on calculation accuracy

Nested sampling calculates the evidence $Z$ as the sum of sample weights (5); the dominant sampling errors are from statistically estimating shrinkage ratios $t_i$ which affect the weights of all subsequent points. In Appendix B we show that the reduction in evidence errors achieved by taking additional samples to increase the local number of live points $n_i$ is inversely proportional to $n_i$, and is approximately proportional to the evidence contained in subsequent points. This makes sense as the dominant evidence errors are from statistically estimating shrinkages $t_i$ which affect all subsequent points $j \geq i$.

In nested sampling parameter estimation, sampling errors come both from taking a finite number of samples in any region of the prior and from the stochastic estimation of their normalised weights $p_i$ from (6). Typically standard nested sampling takes many samples with negligible posterior mass as illustrated in Figure 1; these make little contribution to estimates of parameters or to the accuracy of samples’ normalised weights. From (4) the expected separation between points in log $X$ (approximately proportional to the posterior mass they each represent) is $1/n_i$. As a result increasing the number of live points wherever the dead points’ posterior weights $p_i \propto L_i w_i$ are greatest distributes posterior mass more evenly among the samples. This improves the accuracy of the statistically estimated weights $p_i$, and can dramatically increase the information content (Shannon entropy of the samples)

$$H = \exp \left( - \sum_i p_i \log p_i \right)$$

which is maximised for a given number of samples when the sample weights are equal. Empirical tests of dynamic nested sampling show that increasing $n$ wherever points have the highest $p_i \propto L_i w_i$ works well at increasing parameter estimation accuracy for most calculations.

The optimum allocation of live points is different for different parameter estimation problems involving different functions of parameters $f(\theta)$ as the contribution of each sample $i$ is dependent on $f(\theta_i)$. In most cases the relative weight $p_i$ of samples is a good
approximation for their influence on a calculation, but for some problems much of the error may come from sampling log $X$ regions containing a small fraction of the posterior mass but with extreme parameter values‡. Estimating the importance of points to a specific parameter estimation calculation is discussed in Appendix C.

4. Dynamic nested sampling

We now present an algorithm for performing nested sampling calculations with a dynamically varying number of live points to optimise the allocation of samples.

Since the distribution of posterior mass as a function of the likelihood is a priori unknown, we first approximate it by performing a standard nested sampling run with some small constant number of live points $n_{\text{init}}$. The algorithm then proceeds by iteratively calculating the range of likelihoods where increasing the number of live points will have the greatest effect on calculation accuracy, and generating an additional thread§ running over these likelihoods.

From the discussion in Section 3.1 we define functions to measure the relative importance of a sample $i$ for evidence calculation¶ and parameter estimation respectively as

$$I_Z(i) \propto \frac{E[Z > i]}{n_i},$$

$$I_{\text{param}}(i) \propto L_i w_i.$$  

Modifying (9) to optimise for estimation of a specific parameter is discussed in Appendix C.

The user specifies how to divide computational resources between evidence calculation and parameter estimation through an input goal $G \in [0,1]$, where $G = 0$ corresponds to optimising for evidence calculation and $G = 1$ optimises for parameter estimation. The dynamic nested sampling algorithm calculates points’ importance as a weighted sum

$$I(G,i) = (1 - G) \frac{I_Z(i)}{\sum_j I_Z(j)} + G \frac{I_{\text{param}}(i)}{\sum_j I_{\text{param}}(j)}.$$  

The likelihood range in which to run an additional thread is chosen by finding all points with importance greater than some fraction (we use 90%) of the largest importance. To ensure any steep or discontinuous increases in the likelihood $L(X)$ are captured we find the first point $j$ and last point $k$ which meet this condition, then generate an additional thread starting at $L_{j-1}$ and ending when a point is sampled with likelihood greater than $L_{k+1}$. If $j$ is the first dead point, threads which initially sample the whole

‡See Higson et al. (2017, Section 3.1) for diagrams illustrating sampling errors and the distributions of parameter values as a function of log $X$ for nested sampling runs.

§If required some $n_{\text{batch}}$ additional threads can be generated at each step to reduce the number of importance calculations. We find in empirical tests that this has little effect on efficiency gains from dynamic nested sampling when the number of samples taken in each batch is small compared to the total number of samples in the run.

¶Alternatively (8) can be replaced with a more complex expression using the result derived in Appendix B (23), although we find this typically makes little difference to results.
prior are generated. If \( k \) is the final dead point then the thread will stop when a sample with likelihood greater than \( L_k \) is found. This allows the new thread to continue beyond \( L_k \), meaning dynamic nested sampling iteratively explores higher likelihoods when this is the most effective use of samples.

Unlike in standard nested sampling, more accurate dynamic nested sampling results can be obtained simply by continuing the calculation for longer. The user must specify a condition at which to stop dynamically adding threads, such as when fixed number of samples has been taken or some desired level of accuracy has been achieved. Sampling errors on evidence and parameter estimation calculations can be estimated from the dead points at any stage using the method described in Higson et al. (2017). We term these *dynamic termination conditions* to distinguish them from the type of termination conditions used in standard nested sampling.

More formally our dynamic nested sampling algorithm is:

**Output**: Samples and live points information \( n \).

**Input**: Goal \( G \), \( n_{\text{init}} \), dynamic termination condition.

Generate a nested sampling run with a constant number of live points \( n_{\text{init}} \);

**while** dynamic termination condition not satisfied **do**

  recalculate importance \( I(G, i) \) of all points;

  find first point \( j \) and last point \( k \) with importance of \( > 90\% \) of the largest importance;

  generate a additional thread\(^5\) starting at \( L_{j-1} \) and ending with the first sample taken with likelihood greater than \( L_{k+1} \);

**end**

**Algorithm 1**: Dynamic nested sampling.

Sampling errors on dynamic nested sampling parameter estimation calculations can be estimated using the bootstrap resampling technique introduced in Higson et al. (2017); see Appendix D for more details.

### 5. Numerical tests

In the manner described by Keeton (2011) we use spherically symmetric analytical cases for our empirical tests; here one can perform *perfect nested sampling* as perfectly uncorrelated samples from the prior space within some iso-likelihood contour can be found using standard techniques. The tests in this paper were run using our PERFECTNS package\(^6\). Software used for practical problems may cause additional effects from imperfect sampling within a likelihood contour that are specific to a given implementation — we discuss these in Section 6.

Perfect nested sampling calculations depend on the likelihood \( L(\theta) \) and prior \( \pi(\theta) \) only through the distribution of posterior mass \( L(X) \) and the distribution of parameters on iso-likelihood contours \( P(f(\theta)|X) \), both of which are functions of \( L(\theta) \) and \( \pi(\theta) \) (Higson

\(^5\)PERFECTNS is python package for performing perfect dynamic and standard nested sampling for spherically symmetric likelihoods and priors, and analysing the resulting runs. It can be downloaded at [https://github.com/ejhigson/PerfectNS](https://github.com/ejhigson/PerfectNS) — see the documentation of more details.
et al., 2017). We therefore empirically test our method using likelihoods and priors with a wide range of distributions of posterior mass, and consider several functions of parameters \( f(\theta) \) in each case.

We first examine perfect nested sampling of \( d \)-dimensional spherical unit Gaussian likelihoods

\[
L(\theta) = (2\pi)^{-d/2} e^{-|\theta|^2/2},
\]

with \( d \)-dimensional co-centred spherical Gaussian priors

\[
\pi(\theta) = (2\pi\sigma_\pi^2)^{-d/2} e^{-|\theta|^2/2\sigma_\pi^2}.
\]

For additional tests using distributions with lighter and heavier tails than Gaussians we use \( d \)-dimensional exponential power likelihoods

\[
L(\theta) = \frac{d \Gamma(\frac{d}{2})}{\pi^{\frac{d}{2}} 2^{1+\frac{d}{2}} \Gamma(1 + \frac{b}{2})} e^{-|\theta|^b/2},
\]

where \( b = 1 \) corresponds to a \( d \)-dimensional Gaussian (11). The different distributions of posterior mass in log \( X \) for (11) and (13) with dimensions \( d \) are illustrated in Figure 3.

In tests of parameter estimation we denote the first component of the \( \theta \) vector as \( \theta_1^* \), although by symmetry the results will be the same for any component. \( \bar{\theta}_1 \) is the mean of the posterior distribution of \( \theta_1 \), and the one-tailed \( Y\% \) upper credible interval \( C.I.Y\%(\bar{\theta}_1) \) is the value \( \theta_1^* \) for which

\[
P(\theta_1 < \theta_1^* | L, \pi) = Y/100.
\]

Tests of dynamic nested sampling use \( n_{\text{init}} = 5 \) and terminate after a fixed number of samples, which is set such that they use similar or slightly fewer samples than standard nested sampling runs we compare them to. Standard nested sampling runs use the termination conditions described by Handley et al. (2015b, Section 3.4), stopping when the estimated evidence contained in the live points is less than \( 10^{-3} \) times the evidence contained in dead points (the default value used in POLYCHORD). This is an appropriate
termination condition for nested sampling parameter estimation (Higson et al., 2017),
but if only the evidence is of interest then stopping with a larger fraction of the posterior
mass remaining will have little effect on evidence calculation accuracy.

We begin by testing dynamic nested sampling on a 10-dimensional Gaussian likelihood (11)
with a Gaussian prior (11) and \( \sigma_\pi = 10 \). Figure 4 shows the relative allocation of live
points as a function of log \( X \) for standard and dynamic nested sampling runs. The dynamic
nested sampling algorithm (Algorithm 1) can accurately and consistently allocate live points,
as can be seen by comparison with the analytically calculated distribution of posterior mass and posterior mass remaining**. Similar diagrams for exponential power likelihoods (13) with \( b = 2 \) and \( b = \frac{3}{4} \) are presented in Appendix A (Figures 9 and 10), and show the allocation of live points is also accurate in these cases.

The distribution of results from repeated standard and dynamic nested sampling calculations
with a similar number of samples is shown in Table 1 and Figure 5. Dynamic
nested sampling optimised for evidence calculation (\( G = 0 \)) and at parameter estimation
(\( G = 1 \)) produce significantly more accurate results than standard nested sampling. In
addition results for dynamic nested sampling with \( G = 0.25 \) show that both evidence
calculation and parameter estimation accuracy can be improved simultaneously. Equivalent
results tables for 10-dimensional exponential power likelihoods (13) with \( b = 2 \) and \( b = \frac{3}{4} \) are presented in Appendix A (Tables 2 and 3).

As nested sampling calculation errors are typically proportional to the square root
of the computational effort applied (Skilling, 2006; Higson et al., 2017), the increase in
efficiency (computational speedup) from dynamic nested sampling over standard nested
sampling can be estimated as

\[
\text{efficiency gain} = \frac{\text{Var}[\text{dynamic nested sampling calculation results}]}{\text{Var}[\text{standard nested sampling calculation results}]},
\]

where the calculation results from dynamic and standard nested sampling use approximately the same number of samples. The reduction in evidence errors for \( G = 0 \) and parameter estimation errors for \( G = 1 \) in Table 1 correspond to increasing efficiency by factors of 1.45 and up to 3.9 respectively.

5.1. Efficiency gains for different distributions of posterior mass

Efficiency gains (14) from dynamic nested sampling depend on the fraction of the log \( X \)
range explored which makes a significant contribution to calculation accuracy — if this
is small then most samples taken by standard nested sampling contain little information
and dynamic nested sampling can greatly improve performance. For parameter estimation (\( G = 1 \)), only log \( X \) regions containing significant posterior mass (\( \propto L(X)X \))

**Dynamic nested sampling live point allocations do not precisely match the distribution of posterior mass and posterior mass remaining in the \( G = 1 \) and \( G = 0 \) cases because they include the initial exploratory run with a constant \( n_{\text{init}} \) live points, and because each thread includes one sample taken after its termination condition is met. Furthermore as additional live points are added where samples’ importance is more than 90% of the maximum importance, the number of live points allocated by dynamic nested sampling is approximately constant for regions with importance of greater than \( \sim 90% \) of the maximum — this can be clearly seen in Figure 4 near the peak number of live points in the \( G = 1 \) case.
Dynamic nested sampling

Fig. 4. Live point allocation for a 10-dimensional Gaussian likelihood (11) with a Gaussian prior (12) and $\sigma_\pi = 10$. Solid lines show the number of live points as a function of $\log X$ for 10 standard nested sampling runs, and 10 dynamic nested sampling runs with a similar number of samples and different values of $G$. The dotted and dashed lines show the relative posterior mass $\propto L(X) X$ and the posterior mass remaining $\propto \int_{-\infty}^{X} L(X') X' \, dX'$ at each point in $\log X$; for comparison these lines are scaled to have the same area under them as the average of the number of live point lines. Standard nested sampling runs include the final set of live points at termination, which are modeled using a decreasing number of live points as discussed in Section 3. Similar diagrams for exponential power likelihoods (13) with $b = 2$ and $b = \frac{3}{4}$ are presented in Appendix A (Figures 9 and 10).

are important, whereas for evidence calculation ($G = 0$) all samples taken before the bulk of the posterior is reached are valuable. Both cases benefit from dynamic nested sampling using fewer samples to explore the region after most of the posterior mass has been passed but before termination.

We now test the efficiency gains (14) of dynamic nested sampling empirically for a wide range of distributions of posterior mass by considering Gaussian likelihoods (11) and exponential power likelihoods (13) of different dimensions $d$ and prior sizes $\sigma_\pi$. The results are presented in Figures 6 and 7, and show large efficiency gains from dynamic nested sampling for parameter estimation in all of these cases.

Increasing the dimension $d$ typically means the posterior mass is contained in a smaller fraction of the prior volume (Higson et al., 2017), as can be seen in Figure 3. In the spherically symmetric cases we consider the range of $\log X$ to be explored before significant posterior mass is reached increases approximately linearly with $d$. This increases the efficiency gain (14) from dynamic nested sampling for parameter estimation ($G = 1$) but reduces it for evidence calculation ($G = 0$). High dimensional problems often have the vast majority of the $\log X$ range explored lying before any significant posterior mass is reached; this results in very large efficiency gains for parameter estimation but almost
Table 1. Test of dynamic nested sampling for a 10-dimensional Gaussian likelihood (11) and a Gaussian prior (12) with $\sigma_\pi = 10$. The first four rows show the standard deviation of 5000 calculations for standard nested sampling with a constant number of live points $n = 200$. The next three rows show the standard deviations of 5000 dynamic nested sampling calculations with a similar number of samples, optimised for purely evidence calculation ($G = 0$), for both evidence and parameter estimation ($G = 0.25$) and purely for parameter estimation ($G = 1$). The final three rows show the computational efficiency gain (14) from dynamic nested sampling over standard nested sampling in each case. The first column shows the mean number of samples for the 5000 runs; the remaining columns show calculations of the log evidence and the mean, median and 84% one-tailed credible interval of a parameter $\theta_1$. Numbers in brackets show the error on the final digit.

| # samples | log $\mathcal{Z}$ | $\bar{\theta}_1$ | median($\theta_1$) | C.I.84%($\bar{\theta}_1$) |
|-----------|------------------|-----------------|------------------|------------------|
| St.Dev. standard | 6,076 | 0.304(3) | 0.0250(3) | 0.0312(3) | 0.0407(4) |
| St.Dev. $G = 0$ | 6,040 | 0.252(3) | 0.0283(3) | 0.0406(4) | 0.0469(5) |
| St.Dev. $G = 0.25$ | 6,044 | 0.275(3) | 0.0200(2) | 0.0262(3) | 0.0325(3) |
| St.Dev. $G = 1$ | 6,048 | 1.73(2) | 0.0130(1) | 0.0161(2) | 0.0206(2) |
| efficiency gain $G = 0$ | | 1.45(4) | 0.78(2) | 0.59(2) | 0.75(2) |
| efficiency gain $G = 0.25$ | | 1.22(3) | 1.56(4) | 1.42(4) | 1.58(4) |
| efficiency gain $G = 1$ | | 0.031(1) | 3.7(1) | 3.7(1) | 3.9(1) |

Fig. 5. Distributions of results for the standard nested sampling and dynamic nested sampling calculations in Table 1. Dynamic nested sampling with $G = 0$ and $G = 1$ significantly reduces evidence and parameter estimation sampling errors respectively compared to standard nested sampling. Dynamic nested with $G = 0.25$ reduces both evidence and parameter estimation sampling errors. The colour scale shows the fraction of the cumulative probability distribution lying between some region and the median.
Fig. 6. Efficiency gain (14) from dynamic nested sampling compared to standard nested sampling for likelihoods of different dimensions; each has a Gaussian prior (12) with $\sigma = 10$. Results are shown for calculations of the log evidence and the mean, second moment, median and 84% one-tailed credible interval of the posterior distribution of the parameter $\theta_1$. Each efficiency gain is calculated using 1000 standard nested sampling calculations with $n = 200$ and 1000 dynamic nested sampling calculations using a similar or slightly smaller number of samples.
no gains for evidence calculation as shown in Figure 6. For the 1000-dimensional exponential power likelihood with \( b = 2 \), dynamic nested sampling with \( G = 1 \) improves parameter estimation efficiency by a factor of up to 70 ± 4.

Increasing the size of the prior \( \sigma_\pi \) increases the fraction of the log \( X \) range explored before any significant posterior mass is reached, resulting in larger efficiency gains (14) from dynamic nested sampling for parameter estimation (\( G = 1 \)) but smaller gains for evidence calculation (\( G = 0 \)). However when \( \sigma_\pi \) is small the bulk of the posterior mass is reached after a small number of steps, and most of the log \( X \) range explored is after the majority of the posterior mass but before termination. Dynamic nested sampling places few samples in this region, leading to large efficiency gains for both parameter estimation and evidence calculation, as shown in Figure 7. When \( \sigma_\pi = 0.1 \), dynamic nested sampling evidence calculations with \( G = 0 \) improve efficiency over standard nested sampling by a factor of approximately 7. However we note that if only the evidence estimate is of interest then standard nested sampling can safely terminate with a higher fraction of the posterior mass remaining than \( 10^{-3} \), in which case efficiency gains would be lower.

### 6. Dynamic nested sampling with existing nested sampling software

Existing standard nested sampling software such as MultiNest and PolyChord can be easily adapted to perform dynamic nested sampling. We are working on incorpo-
rating dynamic nested sampling into POLYCHORD 2, which is currently in production. POLYCHORD 2 will be able to perform nested sampling in up to approximately 1000 dimensions — dynamic nested sampling can offer significant efficiency gains in such calculations.

We anticipate reloading a past iteration \( i \) of a POLYCHORD 2 nested sampling run in order to add additional threads will be less computationally expensive than a single likelihood call for many problems. Nevertheless it may prove most efficient for nested sampling software performing dynamic nested sampling to generate additional threads in selected likelihood regions in batches rather than one at a time; provided the number of samples taken in each batch is much less that the total number of samples in the run this will have little effect on efficiency gains from dynamic nested sampling.

Nested sampling software used for practical problems can only approximately sample randomly from the prior within iso-likelihood contours — this introduces additional complexities and errors which are specific to a given software and are not present in perfect nested sampling. MULTINEST and POLYCHORD use dead points to help them sample within iso-likelihood contours, so provided no significant modes are lost we expect the additional samples from dynamic nested sampling in the regions of \( \log X \) of most importance to the calculation will reduce implementation-specific errors.

6.1. Multimodal posteriors

The most important concern for implementing dynamic nested sampling is ensuring no modes or other regions of significant posterior mass are lost. Dynamic nested sampling aids the location of modes which become separated from the remainder of the posterior at likelihood values where it places more samples than standard nested sampling, but will make it harder to find modes where there are few samples, as illustrated in Figure 8. However if a mode is only discovered late in the dynamic nested sampling process it may still be undersampled due to not being present in threads calculated before it was found. If modes separate at likelihood values where dynamic nested sampling assigns few samples, \( n_{\text{init}} \) must be made large enough to ensure no significant modes are lost.

For highly multimodal posteriors, a safe approach is to set \( n_{\text{init}} \) high enough to find all significant modes, in which case dynamic nested sampling will use the remaining computational budget to increase calculation accuracy as much as possible. Even if half of the computational budget is used on the initial exploratory run, dynamic nested sampling will achieve over half of the efficiency gain compared to standard nested sampling that it could with a very small \( n_{\text{init}} \).

7. Conclusion

This paper began with an analysis of the effects of changing the number of live points on the accuracy of nested sampling parameter estimation and evidence calculations. We then presented dynamic nested sampling (Algorithm 1), which varies the number of live points to allocate posterior samples efficiently for a priori unknown likelihoods and priors.

Dynamic nested sampling can be optimised specifically for parameter estimation, showing increases in computational efficiency over standard nested sampling (14) by
Fig. 8. Dynamic nested sampling’s ability to discover modes is determined by the number of live points present at the likelihood \( L(X_{\text{split}}) \) at which a mode splits from the remainder of the posterior (illustrated on the left). In the schematic graph on the right we would expect dynamic nested sampling to be better at finding modes than standard nested sampling in region B (where it has a higher number of live points) but worse in regions A and C.

factors of up to 70 ± 4 in numerical tests. The algorithm can also increase evidence calculation accuracy, and can improve both evidence calculation and parameter estimation simultaneously. We discussed factors effecting the efficiency gain from dynamic nested sampling, including showing large improvements in parameter estimation are possible when the posterior mass is contained in a small region of the prior (as is often the case in high dimensional problems). Empirical tests show significant efficiency gains from dynamic nested sampling for a wide range likelihoods, priors, dimensions and estimators considered. Another advantage of dynamic nested sampling is that more accurate results can be obtained by continuing the run for longer, unlike in standard nested sampling.

We have implemented dynamic nested sampling in our PERFECTNS package, and it can be easily incorporated in existing standard nested sampling software for use on practical problems including degenerate and multimodal posteriors. We are currently working on including dynamic nested sampling in the forthcoming POLYCHORD 2 software package.

References

Brewer, B. J. and Foreman-Mackey, D. (2016) DNest4: Diffusive Nested Sampling in C++ and Python. preprint (arXiv:1606.03757).

Brewer, B. J., Pártay, L. B. and Csányi, G. (2011) Diffusive nested sampling. Statistics and Computing, 21, 649–656.

Cameron, E. and Pettitt, A. (2014) Recursive Pathways to Marginal Likelihood Estimation with Prior-Sensitivity Analysis. Statistical Science, 29, 397–419.

Chopin, N. and Robert, C. P. (2010) Properties of nested sampling. Biometrika, 97, 741–755.
DES Collaboration (2017) Dark Energy Survey Year 1 Results: Cosmological Constraints from Galaxy Clustering and Weak Lensing. *preprint (arXiv:1708.01530)*.

Feroz, F. and Hobson, M. P. (2008) Multimodal nested sampling: An efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society, 384*, 449–463.

Feroz, F., Hobson, M. P. and Bridges, M. (2008) MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society, 398*, 1601–1614.

Feroz, F., Hobson, M. P., Cameron, E. and Pettitt, A. N. (2013) Importance Nested Sampling and the MultiNest Algorithm. *preprint (arXiv:1306.2144)*.

Handley, W. J., Hobson, M. P. and Lasenby, A. N. (2015a) PolyChord: Nested sampling for cosmology. *Monthly Notices of the Royal Astronomical Society: Letters, 450*, L61–L65.

— (2015b) PolyChord: next-generation nested sampling. *Monthly Notices of the Royal Astronomical Society, 15*, 1–15.

Higson, E., Handley, W., Hobson, M. and Lasenby, A. (2017) Sampling errors in nested sampling parameter estimation. *preprint (arXiv:1703.09701)*.

Keeton, C. R. (2011) On statistical uncertainty in nested sampling. *Monthly Notices of the Royal Astronomical Society, 414*, 1418–1426.

Martiniani, S., Stevenson, J. D., Wales, D. J. and Frenkel, D. (2014) Superposition enhanced nested sampling. *Physical Review X, 4*.

Okamoto, Y. (2004) Generalized-ensemble algorithms: Enhanced sampling techniques for Monte Carlo and molecular dynamics simulations. In *Journal of Molecular Graphics and Modelling*, vol. 22, 425–439.

Skilling, J. (2006) Nested sampling for general Bayesian computation. *Bayesian Analysis, 1*, 833–860.

**Appendices**

**A. Additional numerical tests**

This section contains additional tests of dynamic nested sampling using 10-dimensional exponential power likelihoods (13) with $b = 2$ and $b = \frac{3}{4}$; compared to Gaussians (11) these respectively have lighter and heavier tails. Each uses a Gaussian prior (11) with $\sigma_\pi = 10$. Figures 9 and 10 show that the dynamic nested sampling algorithm can accurately and consistently allocate live points for these likelihoods.

Tables 2 and 3 show the errors in repeated calculations for standard and dynamic nested sampling in these two cases. The reduction in errors correspond to increasing efficiency of evidence calculation ($G = 0$) and parameter estimation ($G = 1$) by factors of up to 1.32 and 6.3 respectively in the $b = 2$ case, and by factors of up to 1.58 and 2.89 in the $b = \frac{3}{4}$ case.
Fig. 9. As in Figure 4 but with a 10-dimensional exponential power likelihood (13) with $b = 2$, and a Gaussian prior (12) with $\sigma_\pi = 10$.

Fig. 10. As in Figure 4 but with a 10-dimensional exponential power likelihood (13) with $b = \frac{3}{4}$, and a Gaussian prior (12) with $\sigma_\pi = 10$. 
Dynamic nested sampling

Table 2. As in Table 1 but with a 10-dimensional exponential power likelihood (13) with \( b = 2 \), and a Gaussian prior (12) with \( \sigma_x = 10 \).

|              | # samples | \( \log \mathcal{Z} \) | \( \bar{\theta}_1 \) | median(\( \theta_1 \)) | C.I.84\%(\( \bar{\theta}_1 \)) |
|--------------|-----------|---------------------|-----------------|----------------------|---------------------|
| St. Dev. standard | 7,239 | 0.366(4) | 0.0135(1) | 0.0175(2) | 0.0210(2) |
| St. Dev. \( G = 0 \) | 7,199 | 0.319(3) | 0.0169(2) | 0.0238(2) | 0.0267(3) |
| St. Dev. \( G = 0.25 \) | 7,202 | 0.363(4) | 0.00945(9) | 0.0126(1) | 0.0144(1) |
| St. Dev. \( G = 1 \) | 7,205 | 2.20(2) | 0.00544(5) | 0.00708(7) | 0.00838(8) |
| efficiency gain \( G = 0 \) | | 1.32(4) | 0.64(2) | 0.54(2) | 0.62(2) |
| efficiency gain \( G = 0.25 \) | | 1.02(3) | 2.06(6) | 1.95(6) | 2.12(6) |
| efficiency gain \( G = 1 \) | | 0.028(1) | 6.2(2) | 6.1(2) | 6.3(2) |

Table 3. As in Table 1 but with a 10-dimensional exponential power likelihood (13) with \( b = \frac{3}{4} \), and a Gaussian prior (12) with \( \sigma_x = 10 \).

|              | # samples | \( \log \mathcal{Z} \) | \( \bar{\theta}_1 \) | median(\( \theta_1 \)) | C.I.84\%(\( \bar{\theta}_1 \)) |
|--------------|-----------|---------------------|-----------------|----------------------|---------------------|
| St. Dev. standard | 5,143 | 0.249(2) | 0.0413(4) | 0.0509(5) | 0.0693(7) |
| St. Dev. \( G = 0 \) | 5,112 | 0.198(2) | 0.0440(4) | 0.0625(6) | 0.0752(8) |
| St. Dev. \( G = 0.25 \) | 5,115 | 0.217(2) | 0.0342(3) | 0.0451(5) | 0.0596(6) |
| St. Dev. \( G = 1 \) | 5,121 | 1.29(1) | 0.0246(2) | 0.0306(3) | 0.0408(4) |
| efficiency gain \( G = 0 \) | | 1.58(4) | 0.88(2) | 0.66(2) | 0.85(2) |
| efficiency gain \( G = 0.25 \) | | 1.32(4) | 1.46(4) | 1.27(4) | 1.35(4) |
| efficiency gain \( G = 1 \) | | 0.037(1) | 2.82(8) | 2.76(8) | 2.89(8) |

B. Effect of varying \( n \) on evidence calculation accuracy

Nested sampling estimates the evidence as the expectation of (5), with the dominant sampling errors from the unknown shrinkage ratios \( t_i \) which are independently distributed according to (4). We now investigate the effect of increasing the number of live points \( n_i \) across some shrinkage \( t_i \) by considering (5) with all \( t_j \neq i \) marginalised out and conditioned on \( t_i \), defining

\[
\mathcal{Z}(t_i) \equiv \int \left( \sum_j w_j \mathcal{L}_j \right) \prod_{j \neq i} P(t_j) \, dt_j. \tag{15}
\]

For simplicity instead of using the trapezium rule we calculate point weight as \( w_i = X_i - X_{i-1} = (1 - t_i) \prod_{k<i} t_k \). In this case uncertainty in \( t_i \) causes sampling errors in the weight of point \( i \) and all subsequent points†† and

\[
\sum_j w_j(t) \mathcal{L}_j = \left[ \sum_{j<i} w_j \mathcal{L}_j \right] + (1 - t_i) \left[ \frac{w_i \mathcal{L}_i}{1 - t_i} \right] + t_i \left[ \sum_{j>i} \frac{w_j \mathcal{L}_j}{t_i} \right]. \tag{16}
\]

††If the trapezium rule is used \( t_i \) also affects the weight of the previous point \( i - 1 \), but this has little effect on the results.
where the terms in square brackets are independent of $t_i$. Substituting (16) into (15) and integrating gives

$$Z(t_i) = E[Z_{<i}] + (1 - t_i) \frac{E[L_i w_i]}{1 - E[t_i]} + t_i \frac{E[Z_{>i}]}{E[t_i]},$$

(17)

where we have defined $Z_{>i} \equiv \sum_{k>i} L_k w_k$ and $Z_{<i} \equiv \sum_{k<i} L_k w_k$. From the distribution of the shrinkage ratios (4)

$$E[t_i] = \frac{n_i}{1 + n_i}, \quad \text{St.Dev.}[t_i] = \frac{n_i^{1/2}}{(n_i + 1)(n_i + 2)^{1/2}},$$

(18)

so (17) can be written

$$Z(t_i) = \left( E[Z_{<i}] + E[L_i w_i](n_i + 1) \right) + t_i \left( \frac{n_i}{n_i + 1} E[Z_{>i}] - (1 + n_i) E[L_i w_i] \right),$$

(19)

where terms in large brackets are independent of $t_i$. Using the expression for St.Dev.$[t_i]$ from (18), the standard deviation of $Z(t_i)$ is

$$\text{St.Dev.}[Z(t_i)] = \frac{n_i^{3/2}}{(n_i + 2)^{1/2}(n_i + 1)^2} E[Z_{>i}] - \frac{n_i^{1/2}}{(n_i + 2)^{1/2}} E[L_i w_i],$$

(20)

The expected number of samples (computational work) needed to increase the number of live points over some interval $(L_a, L_b)$ is proportional to the log prior shrinkage $\log X(L_a) - \log X(L_b)$. Hence the expected extra samples $\Delta N_s$ required to increase the local number of live points $n_i$ is proportional to the interval log $t_i$, which has an expected size of $1/n_i$. The change in the error on the evidence with extra samples is therefore

$$\frac{d}{dN_s} \text{St.Dev.}[Z(t_i)] = \frac{dn_i}{dN_s} \text{St.Dev.}[Z(t_i)]$$

(21)

$$\propto n_i \frac{d}{dn_i} \text{St.Dev.}[Z(t_i)]$$

(22)

$$\propto - \frac{n_i^{3/2}(n_i^2 - 3)}{(n_i + 1)^3(n_i + 2)^{3/2}} E[Z_{>i}] - \frac{n_i^{1/2}}{(n_i + 2)^{3/2}} E[L_i w_i].$$

(23)

This quantity can be easily calculated for a set of dead points with little computational cost. Typically $n_i \gg 2$ and $E[L_i w_i] \ll E[Z_{>i}]$, in which case

$$\frac{d}{dN_s} \text{St.Dev.}[Z(t_i)] \propto - \frac{E[Z_{>i}]}{n_i}.$$  

(24)

Thus the accuracy gained from taking additional samples is approximately proportional to the evidence contained in subsequent dead points. This makes sense as the dominant evidence errors are from statistically estimating shrinkages $t_i$ which affect all subsequent points $j \geq i$. 


C. Tuning for a specific parameter estimation problem

Dynamic nested sampling improves parameter estimation efficiency by placing more samples in log $X$ regions with significant posterior mass and fewer in regions with little posterior mass. However for some likelihoods and parameter estimation problems a large contribution may come from samples in log $X$ regions containing extreme parameter values or very high variability of parameter values but little posterior weight. In this case the expression for sample importances (9) can be modified to favour points with parameter values which will have a large effect on the calculation.

For example estimating the global mean of some parameter or function of parameters $E[f(\theta)] = \sum_i f(\theta_i) L_i w_i$, one could place additional weight on regions with parameter values that have a large effect on the calculation by calculating importances as

$$I_{\text{param}}(i) \propto |f(\theta_i) - E[f(\theta)]| L_i w_i. \quad (25)$$

This expression is highly variable as each point $i$ is a single sample from an iso-likelihood contour $L(\theta) = L_i$ which may cover a wide range of parameters. However dynamic nested sampling (Algorithm 1) uses only the first and last points of high importance in allocating new threads, so using (25) captures log $X$ regions in which some samples have extreme or highly variable parameter values. For tuning dynamic nested sampling for calculating the mean of a parameter $\hat{\theta}_1$, (25) becomes

$$I_{\text{param}}(i) \propto |\theta_i - \bar{\theta}_1| L_i w_i \quad (26)$$

where $\bar{\theta}_1$ is the global mean of $\theta_1$.

We illustrate tuning for a specific parameter by using dynamic nested sampling with a $d$-dimensional spherical unit Cauchy likelihood

$$L(\theta) = \frac{\Gamma \left( \frac{1+d}{2} \right)}{\pi^{(d+1)/2}} \left( 1 + |\theta|^2 \right)^{-\left( \frac{d+1}{2} \right)}. \quad (27)$$

The Cauchy likelihoods have extremely heavy tails and (except in high dimensions) have significant posterior mass present across almost the entire range of log $X$ explored, as shown in Figure 11. We therefore expect relatively low efficiency gains for dynamic parameter estimation ($G = 1$) in this case, but use it for a proof of principle.

Figure 12 shows the allocation of live points by dynamic nested sampling with and without tuning. In the tuned case the live points allocation is consistent with the analytical expectation of (26)‡‡, showing live points can be allocated correctly.

Table 4 shows the efficiency gain for dynamic nested sampling for a 10-dimensional Cauchy likelihood (27) with a Gaussian prior (12) and $\sigma_\pi = 10$. When estimating $\hat{\theta}_1$ the

‡‡For a Cauchy likelihood (27) with a co-centred spherically symmetric uniform prior, the analytic value of $E[|\theta_i|]$ is 0 and each iso-likelihood contour $L(\theta) = L(X)$ is a spherically symmetric surface with radius $|\theta|$. The expectation of $|\theta_i|$ on such an iso-likelihood contour is $|\theta|/\sqrt{d}$, so the analytical expectation of the importance (26) is

$$I_{\text{param}}(X) \propto |\theta| X L(X)/\sqrt{d}. \quad (28)$$
Fig. 11. Relative posterior mass ($\propto L(X)X$) as a function of $\log X$ for Cauchy likelihoods (27), with Gaussian likelihoods (11) shown for comparison. Each has a Gaussian prior (12) with $\sigma = 10$. The lines are scaled so that the area under each of them is equal.

Table 4. Test of tuned dynamic nested sampling with a 10-dimensional Cauchy likelihood (27), and a Gaussian prior (12) with $\sigma = 10$. The first four rows show the standard deviation of 5000 calculations for standard nested sampling and dynamic nested sampling with $G = 0$, $G = 1$ and with a tuned importance function (26) and $G = 1$. The final three rows show the computational efficiency gain (14) from dynamic nested sampling over standard nested sampling in each case. The first column shows the mean number of samples for the 5000 runs; the remaining columns show calculations of the log evidence and the mean, second moment and 84% one-tailed credible interval of the parameter $\theta_1$. Numbers in brackets show the error on the final digit.

|                   | # samples | $\log Z$  | $\bar{\theta}_1$ | $\bar{\theta}_1^2$ | C.I.84%($\bar{\theta}_1$) |
|-------------------|-----------|-----------|-------------------|---------------------|---------------------------|
| St.Dev. standard  | 7,283     | 0.260(3)  | 0.0206(2)         | 0.378(4)            | 0.0893(9)                 |
| St.Dev. $G = 0$   | 7,239     | 0.212(2)  | 0.0190(2)         | 0.348(3)            | 0.0924(9)                 |
| St.Dev. $G = 1$   | 7,256     | 0.673(7)  | 0.0200(2)         | 0.372(4)            | 0.0710(7)                 |
| St.Dev. $G = 1$ tuned | 7,255     | 0.514(5)  | 0.0166(2)         | 0.293(3)            | 0.0692(7)                 |
| efficiency gain $G = 0$ | 1.51(4)  | 1.17(3)  | 1.18(3)           | 0.93(3)             |
| efficiency gain $G = 1$ | 0.150(4) | 1.06(3)  | 1.04(3)           | 1.58(4)             |
| efficiency gain $G = 1$ tuned | 0.257(7) | 1.55(4)  | 1.66(5)           | 1.67(5)             |

calculation is dominated by samples in the tails of the distribution at low likelihoods and dynamic nested sampling therefore gives only a small efficiency gain (14) over standard nested sampling. Tuned dynamic nested sampling is able to improve efficiency gain, as shown in the final row of Table 4. Using the tuned importance function affects the performance gain for other calculations — for example in this case it significantly improves estimates of the second moment of the distribution $\bar{\theta}_1^2$ in comparison to the $G = 1$ case without tuning.

D. Estimating sampling errors in dynamic nested sampling

The technique for estimating sampling errors by resampling threads introduced in Higson et al. (2017) can be applied to dynamic nested sampling. Table 5 shows numerical tests
of the bootstrap error estimates for dynamic nested sampling parameter estimation with the goal $G = 1$ (the most challenging case). The bootstrap error estimates match the sampling errors observed when the calculation is repeated many times, in line with the results for standard nested sampling presented in Higson et al. (2017).

Dynamic nested sampling bootstrap sampling error estimates also work for calculations of the log evidence $\log Z$, but can become slightly inaccurate when $n_{\text{init}}$ is low and $G$ is close to 1 — in this case the simulated weights method may perform better (see Higson et al. (2017) for more details).
Table 5. Sampling errors for dynamic nested sampling of a 3-dimensional Gaussian likelihood (11), a Gaussian prior (12), $G = 1$ and $n = 200$. The first row shows the standard deviation of 5000 nested sampling calculations and the second shows the mean of 5000 error estimates from resampling (using 200 replications) as a ratio to the error observed from repeated calculations. The third row shows the standard deviations of bootstrap sampling error estimates for single runs as a percentage of the mean estimate. The fourth row shows the mean of 500 bootstrap estimates of the one-tailed 95% credible interval on the calculation result given the sampling error, each using 1000 bootstrap replications. The final two rows show the empirical coverage of the bootstrap standard error and 95% credible interval from the 5000 repeated calculations. Numbers in brackets show the error on the final digit.

|                        | $\bar{\theta}_1$ | $\bar{\theta}_1^2$ | median($\bar{\theta}_1$) | C.I.95%($\bar{\theta}_1$) |
|------------------------|-------------------|----------------------|---------------------------|---------------------------|
| Repeated runs St.Dev.   | 0.0185(2)         | 0.0310(3)            | 0.0232(2)                 | 0.0295(3)                 |
| Bootstrap St.Dev. / Repeats St.Dev. | 0.99(1)   | 1.01(1)              | 1.00(1)                   | 1.00(1)                   |
| Bootstrap St.Dev. estimate variation | 6.8(1)%  | 8.3(1)%              | 12.1(1)%                  | 13.4(1)%                  |
| Bootstrap C.I.95%      | 0.0305(6)         | 1.043(1)             | 0.0371(8)                 | 1.038(1)                  |
| Bootstrap Mean±1St.Dev. coverage | 68.2%   | 69.0%                | 68.2%                     | 68.7%                     |
| Bootstrap C.I.95% coverage | 94.9%   | 94.9%                | 95.1%                     | 95.0%                     |