

Research Paper

LITMUS: Bayesian Lag Recovery in Reverberation Mapping with Fast Differentiable Models

Hugh McDougall¹, Benjamin J. S. Pope^{2,1}, Tamara M. Davis¹

¹School of Mathematics and Physics, University of Queensland, St Lucia, QLD 4072, Australia, ² School of Mathematical & Physical Sciences, 12 Wally's Walk, Macquarie University, Macquarie Park, NSW 2113

Abstract

Reverberation mapping is a technique in which the mass of a Seyfert I galaxy's central supermassive black hole is estimated, along with the system's physical scale, from the timescale at which variations in brightness propagate through the galactic nucleus. This mapping allows for a long baseline of time measurements to extract spatial information beyond the angular resolution of our telescopes, and is the main means of constraining supermassive black hole masses at high redshift. The most recent generation of multi-year reverberation mapping campaigns for large numbers of active galactic nuclei (AGN) (e.g. OzDES) have had to deal with persistent complications of identifying false positives, such as those arising from aliasing due to seasonal gaps in time-series data. We introduce LITMUS (Lag Inference Through the Mixed Use of Samplers), a modern lag recovery tool built on the “damped random walk” model of quasar variability, built in the automatic differentiation framework `jax`. LITMUS is purpose-built to handle the multimodal aliasing of seasonal observation windows and provides Bayesian evidence integrals for model comparison and null hypothesis testing, a more quantified alternative to existing post-fit selection methods. LITMUS also offers a flexible and modular framework for using more expressive high dimensional models for the AGN variability, and includes `jax`-enabled implementations of other popular lag recovery methods like nested sampling and the interpolated cross correlation function. We test LITMUS on a number of mock light curves modelled after the OzDES sample and find that it recovers their lags with high precision and successfully identifies spurious lag recoveries, reducing its false positive rate to drastically outperform the state of the art program `JAVELIN`. LITMUS's high performance is accomplished by an algorithm for mapping the Bayesian posterior density which both constrains the lag and provides Bayesian evidences for model comparison and null hypothesis testing while outperforming nested sampling in computational cost by an order of magnitude. 

Keywords: galaxies: active – galaxies: nuclei – quasars: general methods: data analysis – gravitational lensing: strong

(Received xx xx xxxx; revised xx xx xxxx; accepted xx xx xxxx)

1. Introduction

Reverberation mapping (RM) is a technique in which we measure the way that fluctuations of a source's brightness propagate through its physical components, using the delays in the signal stimulated by this flux to estimate the physical scale of the system. In this way, RM substitutes temporal resolution for angular resolution in our observations, allowing us to tease out spatial information about in-sky point-sources. The technique was first introduced (Blandford & McKee 1982; Peterson 1993) in measuring the scale of the broad line region (BLR) of active galactic nuclei (AGN), measuring the lag between variations in the broad-band photometry, dominated by the AGN's central accretion disk, and the emission lines excited in the BLR, observed via spectroscopy. Assuming a virialized orbit of the BLR, this radius (together with velocity measurements from Doppler

broadening of the emission lines), allows us to measure the mass of the AGN's central supermassive black hole.

Though uncertainty in the geometry and kinematics of the system mean that BLR RM requires a low-redshift anchor to measure masses, it nevertheless forms a powerful tool for measuring quasar masses to cosmological distances. Recent years have seen the close of the first generation “industrial-scale” RM surveys, namely the Sloan Digital Sky Survey (SDSS; Shen et al. 2015) and the Australian Dark Energy Survey (OzDES; King 2015; Lidman et al. 2020), with large numbers of AGN observed out to high redshifts. Such surveys measure AGN masses directly, but also use their measurements to constrain the “radius-luminosity”, or $R - L$, scaling relationship for AGN: an observed power-law between the scale of the BLR and the mean photometric luminosity of the quasar (Kaspi et al. 2000).

Though these large scale surveys collect hundreds to thousands of RM lags, they are impacted by the problem of aliasing: the emergence of multiple peaks in the lag posterior distribution that emerges from the annual six month seasonal gaps in observations. Aliasing has proven to be a major issue in long-baseline RM, with significant efforts being directed towards either suppressing spurious lags in the posterior distribution (e.g. Grier et al. 2017; Grier et al. 2019) or

Author for correspondence: Hugh McDougall, Email: hughmcdougall@mail@gmail.com

Cite this article: McDougall et al., litmus: Bayesian Lag Recovery in Multi-Reverberation Mapping with Fast Differentiable Models *Publications of the Astronomical Society of Australia* 00, 1–12. <https://doi.org/10.1017/pasa.xxxx.xx>

filtering out suspect lags through quality cuts (e.g. Penton et al. 2021; Yu et al. 2023; Shen et al. 2023; Penton et al. 2025). Such methods often disagree about how many sources to keep, and the most stringent of them can lead to as many as 90% of sources being discarded to ensure the purity of the remaining 10% (see McDougall 2025, for a comparison and discussion).

As well as extending into higher redshifts and farther sources, RM has also evolved to probe both smaller and larger physical scales of the AGN, with RM being applied to both the large dusty torus that orbits beyond the BLR (e.g. Suganuma et al. 2006; Koshida et al. 2014; Minezaki et al. 2019), and also to smaller scales (roughly light-days) of the accretion disk itself (Fausnaugh et al. 2016; Yu et al. 2020a). RM of the accretion disk typically measures lags between observations in different photometric filters, comparing the bluer light from the hot interior of the disk to the redder outer edge, allowing both the size and temperature profile of the disk to be constrained. Where X-ray observations are available, we can also probe the emission from the black hole’s corona that is believed to be the driving signal of the AGN variability (Cackett et al. 2007). Though RM alone only directly measures scale, it can also be used to infer properties in other dimensions. For example, in recent years intensive broad-band RM has revealed possible evidence of viscous motion of the disk material, identifying a “long” timescale of possibly negative lag associated with the inwards migration of hot material from the disk’s cooler outer edge to its hot interior (e.g. Hernández Santisteban et al. 2020; Secunda et al. 2023). Such measurements imply a disk that is significantly thicker than the traditional “optically thick, geometrically thin” disk assumed by the famous thin-disk model of Shakura & Sunyaev (1973). For a review of these techniques, we direct the reader to the excellent review of RM at different scales by Cackett et al. (2021).

High cadence of X-ray reverberation allows for complex Fourier analysis that infers multiple lags from different parts of the signal (for a more thorough review, see Uttley et al. 2014). A similar increase in complexity allows for “velocity resolved” reverberation mapping, which aims to constrain the geometry of the BLR in addition its characteristic scale. All other RM falls under the roof of time delay estimation, most commonly between two light curves at a time, in which we aim to constrain a single characteristic delay between the two signals. For quasars we have a convenient statistical description of AGN variability: though they lack deterministic light curves, their stochastic variations are found to follow closely to the power spectral density of the “damped random walk” (DRW), a first order continuous auto-regressive process (i.e. one that acts like red noise at short timescales). The DRW is an example of a Gaussian process (GP), a class of stochastic signals that are well suited to modelling AGN variability well in general (we give a brief overview in this paper, but for a full review of GP’s in astronomy see Aigrain & Foreman-Mackey 2023).

Since its inception, RM has seen a steadily evolving bank of tools for this time delay estimation / lag recovery task. These broadly fall into two categories: those that model the AGN light curves as a Gaussian Process and are agnostic about the underlying statistical behaviour of the light curve and parameters, and those that use our understanding of

the statistical properties of the AGN’s stochastic variations to fit lags with a Bayesian generative model.

Despite its conceptual simplicity, this is far from a trivial task. Aside from the ever growing tool-chest of AGN RM, the adjacent field of estimating quasar time delays from gravitational lensing, essentially the same statistical problem, encountered enough issues to warrant the Time Delay Lens Modelling Challenge (Ding et al. 2020, 2021), an open competition amongst a number of teams to find even a single robust and reliable method for inferring lags. Though the more rigorous GP models have been in place for well over a decade, non-GP methods are still often used thanks to their low numerical cost and apparent lower sensitivity to aliasing. In this paper, we show that the aliasing problem of the GP methods is in fact overstated, and a result not of the fundamental statistical properties of RM or seasonal observations but rather a numerical artefact owing to a choice of sampling method incorrect for this problem. In this paper we provide a new code, LITMUS, which solves many of the challenges of aliasing while putting rigorous statistical constraints on its impact, discussed in detail in Section 4. We describe the statistical modelling and numerical fitting of this new method, and demonstrate that this approach can both recover lags in mock data better than existing tools and that it can distinguish between true lags and spurious false positives.

2. Principles of Reverberation Mapping & Complications Therein

In the most general sense, RM is a technique of measuring the radial scale of a system, however in the context of surveys like OzDES the goal is to constrain the mass of the central black hole via the size of the BLR. This is done by simplifying the BLR to be a thin disk with a single characteristic radius/ reverberation lag $R = c \cdot \Delta t$, where c is the speed of light. Assuming the BLR is in a virialized orbit about the SMBH such, the virial mass can be estimated with a line-of-sight velocity dispersion from the doppler broadening of the emission line profile. The unknown geometry and kinematics are captured in the “virial factor”, calibrated with local anchors of known mass (e.g. from the $M_{\text{SMBH}} - M_*$ relationship; Woo et al. 2015; Grier et al. 2013). This factor has high population dispersion, imposing on RM mass measurements an uncertainty floor of roughly a quarter. RM lags are also used to constrain the power-law scaling relationship between BLR radius and AGN luminosity (the $R - L$ relationship). Accurate measurement of lags is crucial not only for inferring AGN masses, but also for constraining the slope, offset and scatter parameters in this relationship (e.g. McDougall 2025).

There has been a broad spectrum of approaches in attempting to consistently and reliably measure Δt across all AGN (see Section 3), and for LITMUS we adopt the approach of Bayesian forward modelling, i.e. one in which the AGNs signals are characterised by some set of model parameters θ which we have constrained to some prior probability distribution $\pi(\theta)$, and for which any θ has a likelihood $\mathcal{L}(D|\theta)$ of reproducing our observational data D such that there is a

joint distribution $P(\theta|D)$ ^a for parameters and observations:

$$\mathcal{P}(\theta|D) = \mathcal{L}(D|\theta)\pi(\theta) = Z \times \mathcal{P}(\theta). \quad (1)$$

We describe a few existing methods for recovering lags within this Bayesian GP framework in Section 3.1. This is the approach taken by LITMUS. Unique to LITMUS is the calculation of Bayesian model evidence, the total “probability mass” of the posterior distribution:

$$Z = \int \mathcal{P}(\theta|D)d\theta. \quad (2)$$

This evidence is a total “goodness of fit” for the model to the data, and comparing the evidence for two models (their ratio being the “Bayes Factor”) is a means of comparing which model is more supported by observation. This is the approach taken by LITMUS in determining whether lag recoveries are statistically significant.

As our understanding of the AGN variability has evolved we have come to describe their stochasticity as a GP, wherein the time series observations y are Gaussian-correlated observations, such that the model likelihood is:

$$\mathcal{L}(\theta|D) = \frac{1}{\sqrt{(2\pi)^N \det(C)}} \exp\left(-\frac{1}{2}y^T C^{-1} y\right), \quad (3)$$

where C is the ‘covariance matrix’ constructed for all observations over all light curves, and y is the vector of observations after subtracting off the signal means (these means also being model parameters). This includes both the intrinsic variance / covariance of the GP in the matrix S , and the approximately uncorrelated white noise measurement uncertainty for each observation (E_i for measurement i) in the diagonal matrix N :

$$C = S + N, \quad N = \delta_{ij} E_i E_j \quad (4)$$

If the underlying continuum light curve is described by a GP with a covariance function:

$$\phi_c(t) = \langle y_c(t - t'), y_c(t') \rangle, \quad (5)$$

and some the response function is “shifted, scaled and blurred”, i.e. multiplied by some amplitude, mean offset, mean lag and then blurred by smoothing kernel $\psi(t)$, then the auto-covariance of the response signal is:

$$\phi_r(t) = \int_{-\infty}^{\infty} \phi_c(t') \psi(t') dt', \quad (6)$$

and the covariance between the response signal and continuum is:

$$\phi_{rc}(t) = \int_{-\infty}^{\infty} \phi_c(t') \psi(t') \psi(t'') dt' dt''. \quad (7)$$

If the parameters are known, this covariance function can also be used to reconstruct confidence intervals for the behaviour of the light curve between observations (e.g. Figure 1).

It has been empirically observed that AGN variability matches closely to the damped random walk (DRW Kelly et al. 2009; Kozlowski et al. 2010; MacLeod et al. 2010), in which covariance function obeys the double exponential Laplace distribution:

$$\phi_c(t) = \sigma_c^2 \exp\left(\frac{-|t|}{\tau}\right). \quad (8)$$

^aIn this paper, we use $P(\theta)$ for the normalised probability distribution, and $\mathcal{P}(\theta)$ to represent the un-normalised joint distribution, i.e. $P(\theta) = \frac{1}{Z} \mathcal{P}(\theta)$, where Z is the model evidence per Equation 2.

Though actual AGN variability differs slightly from a DRW (Zu et al. 2013), it has been found that the exact choice of GP / covariance function has little impact on lag recovery (Yu et al. 2019).

Although LITMUS works only with single lags / response signals in its initial release, this model also extends to the multi-lag case by having response-response covariance for two response signals with transfer functions $\psi_1(t)$ and $\psi_2(t)$:

$$\phi_{rr}(t) = \iint_{-\infty}^{\infty} \phi(t) \psi_1(t') \psi_2(t'') dt' dt''. \quad (9)$$

This multi-lag RM sees considerable use in mapping the temperature profile of accretion disks (e.g. Yu et al. 2020b), where different colour temperatures occur at different annular radii.

2.1. The Aliasing Problem

As reverberation mapping first pressed into the regime of industrial scale RM, the endeavour soon ran afoul of the confounding effects of ‘aliasing’: a suite of problems emerging from the combination of low precision measurements (shorter exposures, looser cadence) and the half-yearly seasonal gaps in the observations. Aliasing manifests as a tendency for lag recovery methods to over-report lags at and around the “aliasing peaks”, which correspond to minimal data overlap between the continuum and response curves (≈ 180 d, 540 d etc.). This aliasing problem is a major confounding factor in modern RM, with the arising false positive lags dominating the mass measurements and $R - L$ constraints when false positives are not screened.

The traditional approach to the aliasing problem has been to take a frequentist approach of quality control: drawing a statistical cordon around the entire lag recovery method, inclusive of the statistical model, numerical model and lag recovery software within a single ‘black box’ and determining some fit quality measure for determining how likely a recovery is to be erroneous. The OzDES team has adopted a consistent, if steadily evolving, framework for this approach: using JAVELIN as its primary lag recovery method, in conjunction with PyCCF as a validation method, and using simulation-based measures of the the likelihood of a false positive to winnow down to a high purity final data set with $< 10\%$ contamination by spurious lag recoveries. Across its various data releases SDSS have trialled multiple different anti-aliasing regimes. In earlier releases (Grier et al. 2017; Grier et al. 2019), they use CREAM (Starkey et al. 2015) for their lag posterior and validate through a combination of down-weighting lags that lie near aliasing peaks and rejecting recoveries that allow for non-physical negative lags. In their final data release they adopt a less stringent selection criteria, using PyROA (Donnan 2021) as their primary lag recovery method and testing for significance with the r^2 recovered by PyCCF. For a full review and comparison of these selection criteria, we direct the reader to the OzDES RM wrap-up paper (McDougall 2025).

In the abstract sense, aliasing occurs because testing the half-yearly “off-season” lags coincides with little to no overlap in data between light curves. Lag recovery is as much about rejecting poor fits as identifying good ones, and such a lack of overlap means we have fewer chances to identify tensions that indicate such bad fits. Between seasonal gaps, over-loose

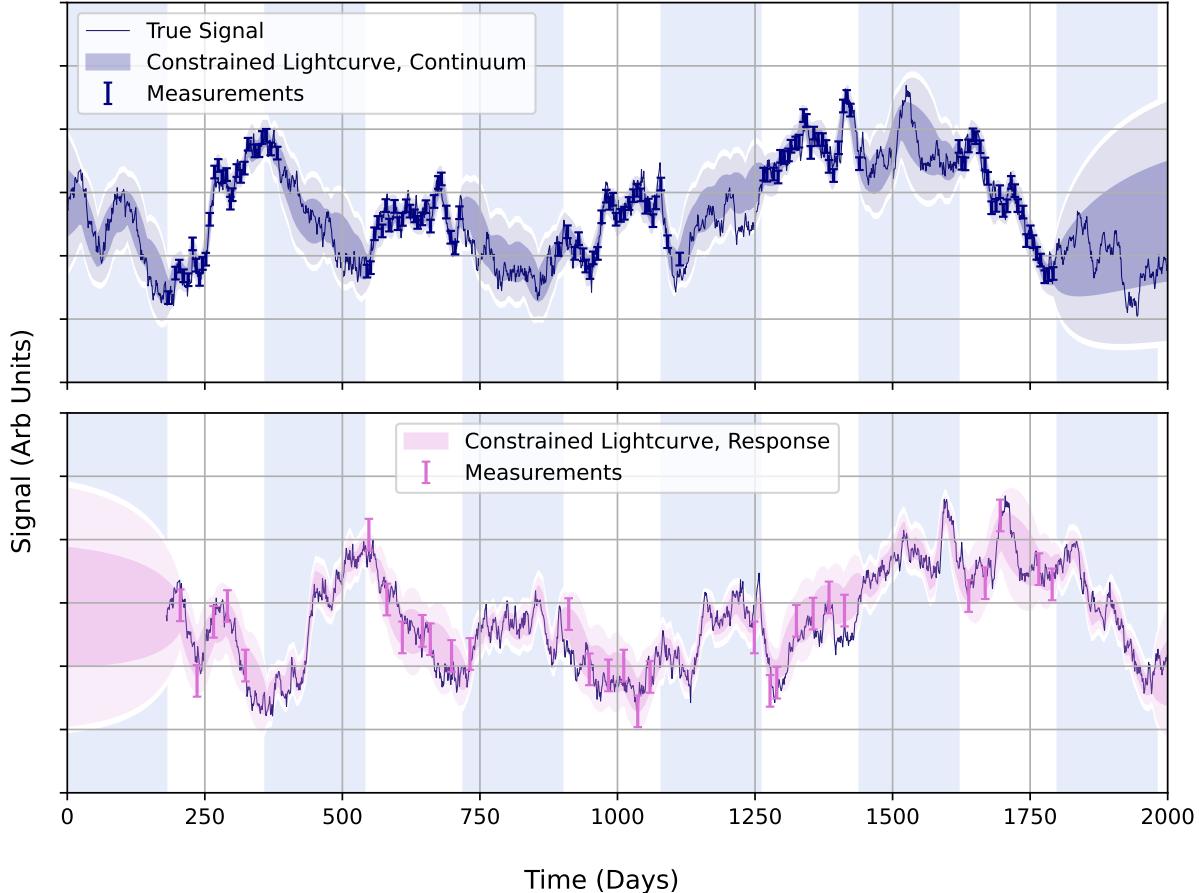


Figure 1. A demonstration of the sort of light curves that GP modelling can reconstruct from observations. For some time-series observations (error bars) a particular GP models the entire family of underlying light curves that exhibit the power spectral density of the GP, conditioned on how well they fit the observations. In this example the light curves is fit as a DRW with $\tau = 200$ d and $\sigma = 1$, both in arbitrary units for this demonstrative example. The shaded regions represent the 1 and 2σ contours of the distribution of all such walks.

interpolations are vague and weakly constraining, while overgenerous interpolations are at the mercy of coincidence and can produce spurious structure. In either case, failure-modes of the light curve reconstruction / lag testing are at their worst at these difficult lags. In Figure 2 we demonstrate this for mock data, and in particular show the multimodality that occurs in the lag posterior for the specific case of a parametric GP model. Aliasing becomes worse when the true lag is near an aliasing peak, as these are more ambiguously observed and produce a shallower peak in the posterior. In such cases we can only observe tension / discontinuity in the reconstructed light curve in the joins between observation seasons, and so aliasing becomes more pronounced for more rapid timescales of variability (smaller τ in Equation 8) and for fewer observed seasons.

In the example shown in Figure 2, the lag is well constrained to its true value as the aliasing peaks are several orders of magnitude shallower. However some Bayesian modelling tools can fail even in these cases due to the secondary numerical challenges that aliasing poses. JAVELIN fits lags by sampling the posterior with `emcee`, an implementation of the Affine Invariant Ensemble Sampler (Goodman & Weare 2010). Though `emcee` has a well earned reputation as a robust and reliable sampler, it is decidedly not suited to such multimodal distributions. Its proposal method, which

involves drawing a chord between two samples and performing a “stretch-move” to propose a new sample, cannot easily mix between well separated modes. The result is the ensemble of live points becomes partially ‘pinned’ at the aliasing modes, over-sampling them and giving the false appearance of a significant bulk of posterior density, sometimes multiple orders of magnitude over the true height of the peak. In Figure 3 we see the dangerous failure mode that can result: the mirage appearance of an aliasing peak in the reported posterior where none truly exists.

In addition to aliasing, there is a less discussed but still important matter of the sharp dips and valleys in the posterior that give it a “rough” geometry, particularly in the low log-density regions (more commonly around the on-season lags, between the aliasing peaks). The existence of these “furrows” does not obscure the statistical results, but it can make the posterior extremely difficult to navigate for some algorithms. The sharp gradients and potential energy “walls” can cause samplers, particularly gradient based samplers like Hamiltonian Monte Carlo (Duane et al. 1987; Neal 1996; Betancourt 2018) to become stuck, and the cases where these dips go to extreme negative values can lead to unstable computational overflow. These furrows arise from the same source as aliasing, but on a shorter timescale. At lags

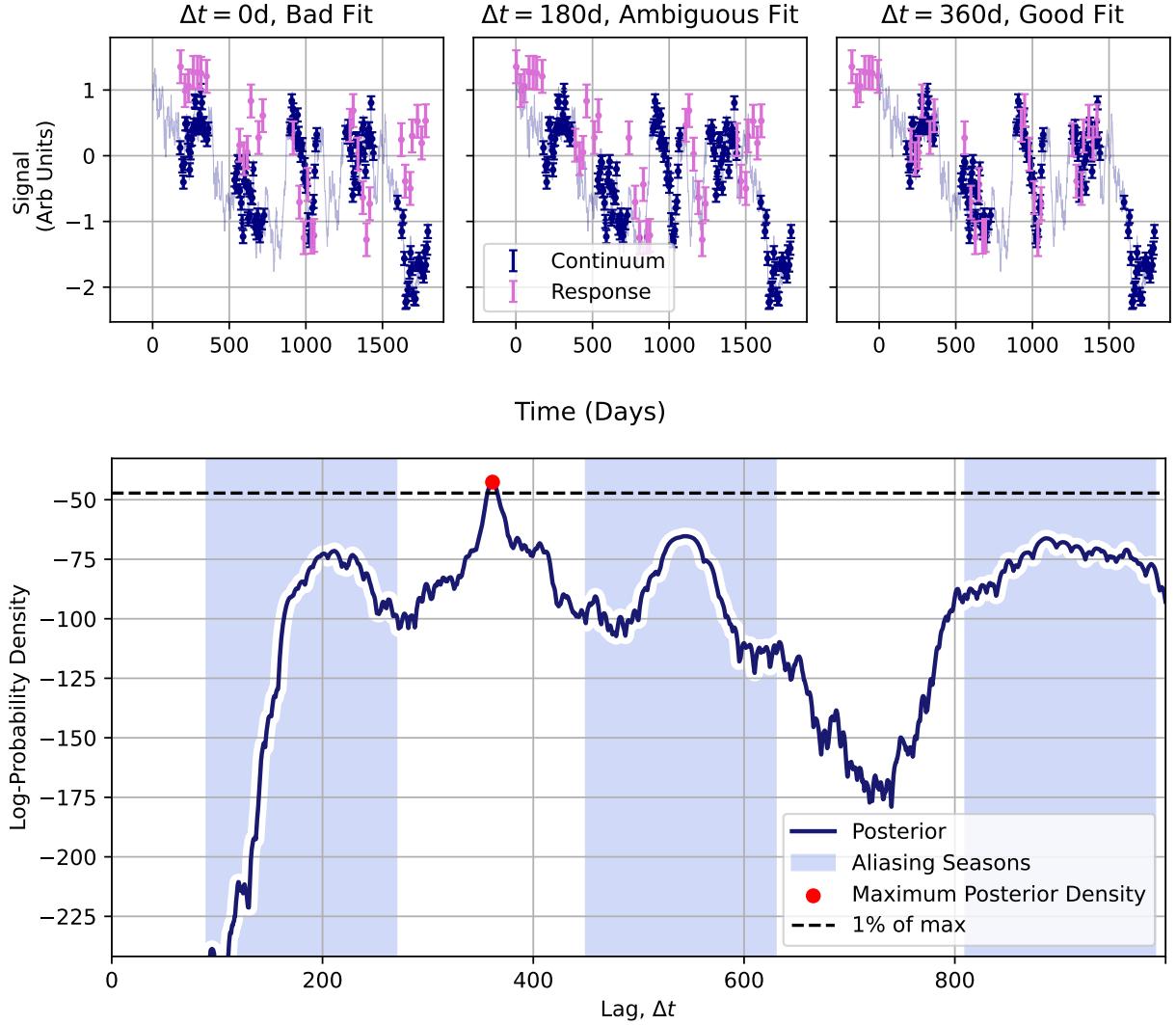


Figure 2. A demonstration of the source of the aliasing problem, specifically in the context of a parametric GP model. Top shows mock data with cadence, measurement uncertainty and baseline similar to OzDES with a DRW timescale of $\tau = 200$ d and a true lag of $\Delta t = 360$ d. From left to right the sub-panels show lags being tested at $\Delta t = 0$ d, 180d and 360d. The left panel is clearly a bad fit as near simultaneous observations are in clear tension, and the right panel is a clear good fit as we see very little tension. The middle panel, corresponding to the first aliasing peak, is an ambiguously good fit; the lack of overlap means we cannot observe clear tensions between the light curves. The bottom panel shows the (un-normalised) log-natural of the posterior distribution, with all non-lag parameters fixed at their true values. At “on-season” lags (un-shaded) we can easily reject bad fits, and so the posterior is extremely low. During the off-season lags (blue shading) there are local optima arising from the ambiguity. The mode associated with the true lag (red dot) is clearly defined and dominates over aliasing modes, with the rest of the posterior being $< 1\%$ of the maximum posterior density in this well behaved, high SNR example. Even so, the posterior still suffers from the rough geometry and multimodality that introduces numerical challenges in navigating it.

where two individual observations overlap between continuum and response, the sharp tension between them causes a massive penalty to the log-likelihood, forming a deep valley in the log-posterior. Furrows are then more severe for smaller measurement uncertainty and higher measurement cadence, paradoxically becoming more of an issue the better our measurements are. For signals with slower variations (larger τ in Equation 8), the longer correlation timescale smooths this effect out somewhat.

3. Existing Methods

Reverberation mapping has progressed markedly over its decades long history, both in our understanding of the physics and of the best-practice in constraining the lags.

Lag recovery methods can be broadly grouped into two categories: the Bayesian GP methods which make use of our understanding of the AGN signal statistics, and the non-GP methods which aim to be as agnostic as possible about the signal, using only flexible means of interpolation and measures of goodness of fit.

LITMUS’s main fitting methods belong to the GP-based class, and so in Section 3.1 we discuss this family and, in examine in particular its widely used exemplar JAVELIN and its limitations. We also discuss non-GP methods in Section 3.2, in particular the method used by PyCCF. In Section 3.3 we give a brief overview of methods that are not used by OzDES/ not directly compared against LITMUS in this paper.

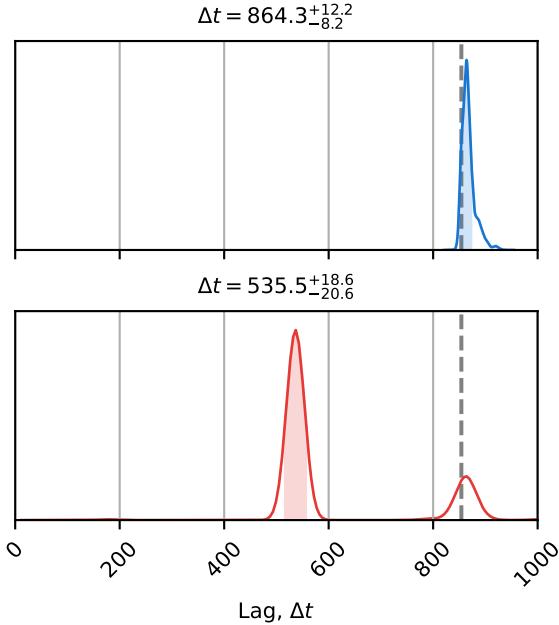


Figure 3. A demonstration of the failure mode of the Affine-Invariant Ensemble Sampler (AIES), the MCMC proposal algorithm used by `emcee`, in multi-modal distributions. Both top and bottom panels are posterior distributions generated from the same mock data with a true lag at $\Delta t = 854$ d (dashed line), with the bottom panel being the result from the AIES, the same MCMC sampler as JAVELIN, while the top is found from exhaustive sampling of the prior range. The AIES estimate for the posterior has produced an aliasing peak at $\Delta t = 540$ d where none truly exists due to its ensemble of live sampling points becoming pinned at this minor mode.

3.1. Gaussian Process Methods / JAVELIN

The main focus of this paper is the class of GP methods that leverage our understanding of AGN signal statistics to work within the framework of a full Bayesian generative model. The core principle of these methods is to take the description of the AGN variability as a Gaussian process (discussed in detail in Section 2), for which there is a closed form and (somewhat) easily evaluable likelihood function.

Different methods of this class differ in three ways:

1. How they construct their covariance matrix (what GP to describe the underlying AGN variability with, how to describe the transfer function of the response).
2. What statistical methods they use to map the Bayesian posterior distribution.
3. What parameter-space they define their priors over, i.e. a purely phenomenological set of parameters to describe the signals (signal mean, amplitude etc, in the style of JAVELIN), or a more physically motivated set of parameters (SBMH mass, accretion rate etc, in the style of CREAM).

JAVELIN is the longest standing of the GP fitting methods and has the distinction of being the most widely used. JAVELIN (Zu et al. 2010) is a successor to the FORTRAN-based SPEAR (Zu et al. 2011). It models the AGN light curves as a damped random walk (following the covariance functions outlined in Section 2) and models the response smoothing with a “top-hat” smoothing function:

$$\psi(t) = \frac{1}{b} \begin{cases} 1, & |t| \leq \frac{b}{2} \\ 0, & |t| > \frac{b}{2} \end{cases} . \quad (10)$$

Noting that the exact form of smoothing tends to have little impact on the lag recovery in BLR RM (Yu et al. 2019) it is common-place when using JAVELIN to fix the smoothing scale, b , to some fixed value shorter than the observation cadence (e.g. Penton et al. 2021), a choice found to have little impact on lag recovery in OzDES-like observational surveys (Yu et al. 2019).

JAVELIN has seen considerable use in the first industrial scale generation of RM, being the “primary” lag recovery method for all of OzDES (Hoormann et al. 2019; Malik et al. 2023; Penton et al. 2021; Yu et al. 2021, 2023), and some of the SDSS releases (e.g. Shen et al. 2015). For constraining its parameters, including the lag, JAVELIN uses the MCMC package `emcee` (Foreman-Mackey et al. 2013), though earlier versions used the classic Metropolis-Hastings Algorithm (Metropolis et al. 1953). As we discuss in Section 2.1, `emcee` fails to properly handle the multimodal log-probability distributions of seasonal light curves. This makes it a poor fit to the trials of aliasing, failing to converge in even high signal to noise cases.

Other GP-based lag recovery programs include CREAM (Starkey et al. 2017) and MICA (Li et al. 2016), discussed in more detail in Section 3.3. In this paper we focus our attention on JAVELIN’s AEIS fitting method, as JAVELIN is the primary OzDES lag recovery code and has a history as the most prevalent tool for BLR RM.

3.2. Non-Gaussian Process Methods / PyCCF

GP methods offer the most complete way to perform lag recovery, they are also more complicated and computationally arduous. Aside from costing sheer time to compute, this also opens room for numerical errors (see discussion of the aliasing mixing problem in Section 2.1). For this reason, the less rigorous but more exhaustive family of non-GP methods, which make as few assumptions about the underlying signal properties as possible, are still actively used in conjunction with or in preference over their more statistically precise GP cousins.

Such non-GP methods are valuable in that their lack of loose commitment towards a particular signal model lends them a flexibility and vagueness that absorbs our uncertainty about the AGN signal properties. Such models tend to also be numerically inexpensive, allowing exhaustive searches of their respective parameter spaces forcing their way past potential numerical challenges. Though LITMUS is primarily a GP method, we include these here for the sake of completeness and comparison.

Of the extant methods of lag recovery, the Interpolated Cross-Correlation Function (ICCF Gaskell & Peterson 1987), is conceptually the simplest. The ICCF describes the best fit lag as being the one that maximises the cross correlation, r , between the emission and response light curves:

$$r(\Delta t) = \frac{\langle y_1(t), y_2(t - \Delta t) \rangle}{\sqrt{\langle y_1(t), y_1(t) \rangle \langle y_2(t), y_2(t) \rangle}}, \quad (11)$$

where y_1 and y_2 represent the sets of photometric and spectroscopic amplitudes respectively, and angled brackets indicate an inner product. Because the measurements are not simultaneous, the ICCF reconstructs one or both of the light curves by linearly interpolating between observations. The uncertainty in the recovered lag is estimated from ‘bootstrapping’: repeating the lag recovery over multiple realisations generated by randomly sub-sampling observations and re-sampling within their measurement uncertainties.

The ICCF method has been found to agree with more rigorous models like JAVELIN to within statistical bounds, though with higher reported uncertainties (Yu et al. 2019). Owing to its numerical robustness and low computational overhead, it is still used as a validation tool by both OzDES (Malik et al. 2023; Yu et al. 2023; Penton et al. 2025) and SDSS (Shen et al. 2023) to identify and remove poorly performing for lags recovered by more complex methods. For comparison, LITMUS includes a `jax` accelerated implementation of the ICCF method in its array of fitting algorithms.

In their final RM data release, Shen et al. (2023) make use of the non-GP method `PyROA` (Donnan 2021), discussed in more detail in Section 3.3. In this paper we focus on a comparison with the ICCF method, which OzDES uses as a secondary method as a part of its validation and quality cuts.

3.3. Other Methods

Here we give a brief overview of three lag recovery methods that have been widely used outside of OzDES’s RM program: `CREAM`, `MICA` and `PyROA`.

CREAM. The GP-based fitting method `CREAM` (Starkey et al. 2015) is similar to JAVELIN in its modelling of the AGN light curve as a stochastic process and using Bayesian fitting, but differs in two key respects. Firstly, instead of fitting for the parameters of the observed signals it instead fits for physical properties of the AGN (e.g. mass, accretion rate etc.). Secondly it models the light curves by fitting the phase and amplitude of a series of Fourier components and, unlike JAVELIN, does not include the covariance of the observations in its likelihood function, instead using a χ^2 goodness of fit loss function. A novel feature of `CREAM` is that it can be set to rescale the error bars of measurements from different telescopes as a part of the MCMC fitting in an internal calibration process (Starkey et al. 2017). `CREAM` is used as the primary lag recovery method by some SDSS releases (Grier et al. 2017; Grier et al. 2019), and is included in the reported results of their final data release (Shen et al. 2023).

MICA. Another GP-based fitting method from Li et al. (2016) is the code `MICA`. The main feature of this code is that it constructs the transfer function of Equation 5 from a sum of Gaussian profiles with positions and widths as free parameters that are marginalised over in the fitting process. In this way, instead of using a simplified approximate transfer function in the style of JAVELIN’s top-hat, `MICA` can allow for a more detailed accounting of the BLR’s geometry. `MICA` uses much of the same statistical approach as JAVELIN, but uses the Metropolis-Hastings algorithm (Metropolis et al. 1953) in place of `emcee`.

PyROA. Introduced by Donnan (2021), `PyROA` adopts a similar strategy to the ICCF but generalises beyond the linear interpolation kernel. It instead uses a “rolling average” to interpolate the light curves, i.e. taking the weighted average with weights following some kernel function, e.g. a Gaussian (multiple options are presented). The width of the kernel is optimised at each point in the reconstructed light curve to maximise the model Bayesian Information Criterion (BIC Schwarz 1978), and from these reconstructions the lag is estimated with their correlation function. `PyROA` is the main lag recovery method in the final SDSS data release of Shen et al. (2023), as they find it to be more precise than `PyCCF` and more reliable than JAVELIN. Like `CREAM`, `PyROA` includes a utility to re-calibrate measurement uncertainty of observations.

4. LITMUS Methodology

The aliasing problem presents challenges on two fronts. The first is numerical: the posterior distribution is very difficult to properly navigate and map the shape of thanks to its rough and multimodal shape. The second is statistical: how can we tell when a recovery is statistically significant when the aliasing peaks can appear to give meaningfully constrained peaks even in the case of a false positive. Here we present a methodologically consistent solution to both problems: firstly, to use novel numerical techniques to properly explore the parameter space of a proper Bayesian model, and secondly to use this robust mapping of the posterior distribution to make use of Bayesian model comparison tools to evaluate the significance of a lag recovery in a principled way. In brief: we do the statistics completely, and the numerics properly.

In this section, we introduce LITMUS , a new lag recovery package, based in `python`, making use of modern computational tools like `jax` (Bradbury et al. 2018) and its ecosystem of flexible software like the statistical modelling framework `numpyro` (Phan et al. 2019) and the GP modelling package `tinyGP` (Foreman-Mackey et al. 2022). LITMUS’s core feature is an algorithm for exploring and integrating the posterior that we call the Laplace Quadrature, which efficiently explores the parameter space to lags while also finding an estimate of the model evidence such that proper Bayesian null hypothesis testing is possible.^b LITMUS also offers a flexible and modular statistical framework that allows it to be easily extended to new generative models, unlike existing tools which hard-code their statistical description of the AGN signals. We then present a new tool for RM that is more precise, mode correct and more robust than anything else in the literature while also being significantly faster and more broadly applicable.

4.1. The Laplace Quadrature

The core idea of the Laplace Quadrature is to side-step the rough geometry of the posterior along the lag-axis (e.g. see the bottom panel in Figure 2) by not trying to navigate over this geometry at all. Instead, a grid of test-lags are spaced

^bNote that this method bears some similarity to a special case of the Integrated Nested Laplace Approximation (INLA Rue et al. 2009; van Niekerk et al. 2022).

along the lag-axis, and the shape of the posterior along all non-lag parameters is estimated with the Laplace approximation (approximating a distribution as Gaussian by a second order expansion of its log-density). In contrast to MCMC-like strategies which need to explore each axis of parameter space to marginalise over them, this Laplace approximation approach relies only on optimisation, for which the complexity scales much less severely with dimensionality of the parameter space. As such, the Laplace Quadrature approach can be applied up to markedly higher model dimensions without incurring an exorbitant computational burden.

We begin with a set of I ordered lags, $\{\Delta t_i\}$, distributed, though not necessarily evenly, between the upper and lower ranges of the uniform lag prior:

$$\Delta t_{\text{Min}} \leq \Delta t_0 < \Delta t_1 < \Delta t_2 < \dots < \Delta t_I < \Delta t_{\text{Max}}. \quad (12)$$

At each of these lags, there is an (here, un-normalised) conditional joint distribution $P_i(\phi)$ for all of the un-fixed non-lag parameters ϕ , (i.e. the full set of model parameters in Equation 1 is $\theta = \{\Delta t\} \cup \phi$):

$$\mathcal{P}_i(\phi) = \pi(\phi|\Delta t) \mathcal{L}(\phi|\Delta t), \quad Z_i = \int \mathcal{P}_i(\phi) d\phi, \quad (13)$$

where Z_i is the evidence integral of the un-normalised conditional distribution such that $Z_i = \frac{dZ}{d\Delta t}|_{\Delta t=\Delta t_i}$. The Laplace Quadrature approximates this distribution and evidence via the Laplace approximation, i.e. supposing that the conditional distribution can be approximated by a multivariate Gaussian distribution:

$$P_i(\phi) \approx Q_i(\phi) = \mathcal{N}(\phi|\mu, \Sigma), \quad (14)$$

where the mean, μ , and covariance matrix, Σ , of this Gaussian can be estimated from the optimum of the distribution, $\hat{\phi}$, and the Hessian matrix, H , of the log density at that point:

$$\mu = \hat{\phi}^i, \Sigma = -H(\hat{\phi}^i)^{-1}. \quad (15)$$

Here, the Hessian matrix is the curvature of the log-density, i.e.

$$H_{i,jk}(\phi) = \frac{\partial^2 \ln |\mathcal{P}_i(\phi)|}{\partial \phi_j \partial \phi_k}. \quad (16)$$

As LITMUS is built in `jax` and `numpyro`, this Hessian matrix can be evaluated with the need for hand-calculated derivatives. Gaussian distributions have easily evaluable integrals, and so the Laplace approximation also lets us easily calculate Z_i for each slice:

$$\ln |Z_i| \approx \ln |\mathcal{P}_i(\hat{\phi}^i)| + \frac{1}{2} \ln |\det(\Sigma)| + \frac{\text{Dim}(\Sigma)}{2} \ln |2\pi|. \quad (17)$$

To summarise, the Laplace Quadrature procedure is:

1. Generate some set of ordered lags $\{\Delta t_i\}$.
2. At each lag Δt_i , optimise all other free parameters ϕ to find the conditional optimum $\hat{\phi}^i$.
3. Calculate the density $\mathcal{P}_i(\hat{\phi}^i)$ and Hessian matrix $H_i(\phi = \hat{\phi})$ of the conditional distribution.
4. Using these, form a normal distribution approximation for the conditional distribution of the non-lag parameters (the Gaussian slice) with mean and covariance from Equation 15 and with normalising evidence from Equation 17. This is done with `numpyro`'s autodiff capabilities.

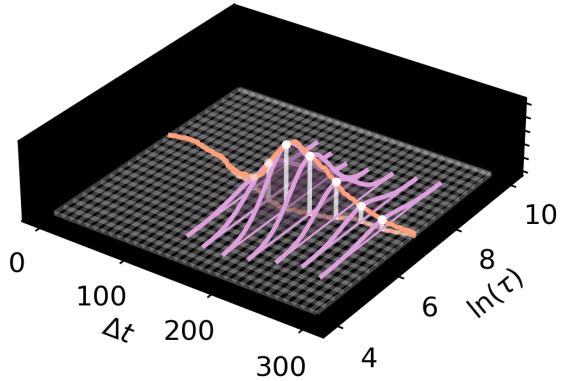


Figure 4. A simplified demonstration of the operating principle behind LITMUS's Laplace Quadrature for a case of only two free parameters (lag and DRW timescale). First, a 1D locus of conditional optima is traced out along the lag axis (orange line), finding the conditional optima at a discrete grid of lags (white points). At these points, the Laplace approximation is applied to divide the posterior up into a series of Gaussian slices (purple, shaded).

5. With the integral at each Gaussian slice, use finite element integration along the Δt axis to estimate the full integral / model evidence via Simpson's rule, the Trapezoidal rule, etc.

A simplified example of this for a case where $\phi = \{\tau\}$ and $I = 5$ is shown in Figure 4. Evenly spaced test lags are wasteful and error-prone, and so LITMUS uses a “grid smoothing” algorithm, outlined in Appendix B, to preferentially sample good-fit lags. For the optimisation to find $\hat{\phi}^i$ we use `jaxopt`'s implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970), using also a pre-conditioning scheme described in Appendix C. Assuming that $\hat{\phi}$ is a smooth and continuous function of Δt , we use the solution $\hat{\phi}^i$ as the starting point for finding $\hat{\phi}^{i+1}$. To find the Hessian matrix, we use `jax`'s autodiff tool-set on the probability log-density functions provided for a particular model by `numpyro`.

4.2. Statistical Modelling

In its statistical modelling, LITMUS adopts the same approach as JAVELIN in describing the AGN fluctuations as a damped random walk and using the Bayesian likelihood in Equation 3. For simplicity, we do not include any smoothing of the response function, i.e. we use a transfer function $\psi(t) = \delta(t)$, where δ is the Dirac delta function. Again following the lead of JAVELIN, we adopt uniform priors for all variables, except the damping timescale τ and continuum signal amplitude σ_c which are fit on a log-uniform scale owing to their strong correlation and the lack of a prior knowledge of the magnitude of the signal timescale. In this way, LITMUS's statistical models are one-to-one with JAVELIN's under the condition of no smoothing, i.e. $b = 0$ in Equation 10. We can define two simpler models to test the significance of our lag recovery against:

- The continuum and response being uncorrelated DRW’s, i.e. no lag encoded in the signal, and
- The continuum being a DRW, but the response being pure white noise.

The Bayes factors comparing these models allows us to test the significance of a lag recovery. For an example of how these ratios can be interpreted, see Section 4.4.

4.2.1. Alternative Statistical Models

A useful feature of LITMUS is that, unlike existing tools which hard-code their statistical models, its modular design means that LITMUS can also fit alternative statistical descriptions of the signals, e.g. the null hypothesis models used to test the significance of lag recoveries. As an example, LITMUS includes an alternative to the normal GP modelling but using an adjustable log-normal prior of the kind that would be recovered from an $R - L$ relationship, i.e. $\pi(\Delta t) \sim \mathcal{N}$. Similar extensions can be made to incorporate the flexible transfer functions of MICA or the physically motivated parameterization and error calibration of CREAM, features that LITMUS’s initial release lacks.

This modularity also allows for more expressive models to probe physical phenomena, for example:

- Using a summation of additional GP kernels to examine differences between the signal and the canonical DRW,
- Modelling the BLR holiday (Dehghanian et al. 2019), e.g. with a short-lived signal super-imposed on the GP signal, or
- Testing models with time-varying lags to investigate changes in the BLR shape or scale

are all implementable in LITMUS if they can be expressed with a `numpyro` generative model, and can be fit by a suitable choice of its existing fitting methods.

4.3. Additional Algorithms

As well as the Laplace Quadrature, LITMUS includes other lag recovery algorithms for comparison. Firstly, it includes a `jax` based high speed implementation of ICCF. Secondly, it includes a variation of the Laplace Quadrature that uses stochastic variational inference (SVI) in place of the Laplace approximation, and finally it has an interface with the nested sampling (Skilling 2006), specifically `JAXNS` (Albert 2020).

4.3.1. The SVI Quadrature

The Laplace Quadrature relies on Laplace approximation, which can be fragile against highly non-Gaussian posteriors. As an extension LITMUS also offers the SVI Quadrature, which has similar working but uses SVI to generate its Gaussian distribution approximation. While the Laplace approximation fits a Gaussian with a point-estimate based on the Hessian of the log-posterior at its maximum, SVI seeks to fit a Gaussian that is ‘most similar on average’ (see Figure 5 for a visual demonstration). This is codified by the Kullback–Leibler divergence (KL divergence), or “average log difference” between the Gaussian ‘surrogate distribution’

in Equation 14 and the true posterior density:

$$\text{KL}_{P_i(\phi)Q_i(\phi)} = \mathbb{E} \left[\ln \left| \frac{P_i(\phi)}{Q_i(\phi)} \right| \right]_{Q_i(\phi)}, \quad (18)$$

where $P_i(\phi)$ is again the un-normalised joint distribution conditioned at test lag Δt^i . The slice evidence Z_i is not known a priori, but can be factored out of the expression, such that the minimum value of KL can be found to within an additive constant:

$$\text{KL}_{P_i(\phi) \rightarrow Q_i(\phi)} = \mathbb{E} \left[\ln \left| \frac{\mathcal{P}_i(\phi)}{Q_i(\phi)} \right| \right]_{Q_i(\phi)} - \ln |Z_i|$$

It can be shown that the KL divergence is always positive, which means that $\ln |Z_i|$ is always greater than or equal to the expectation value in Equation 18, being equal at a perfect match between the surrogate distribution and the true posterior (i.e. when $\text{KL}=0$). As such, this is called the “Evidence Lower Bound” (ELBO) of the fit, and can be approximated by evaluating the log-difference at samples drawn from $Q_i(\phi)$:

$$\ln |Z_i| \geq \text{ELBO}_{P_i(\phi) \rightarrow Q_i(\phi)} \approx \frac{1}{M} \sum_{m=1}^M \ln \left| \frac{\mathcal{P}_i(\phi_m)}{Q_i(\phi_m)} \right|, \quad \phi_m \sim Q_i(\phi_m) \quad (19)$$

By maximising the ELBO we find both the highest (best) estimate of the evidence, and also the surrogate distribution that is the closest fit to the posterior. This is done by assuming some parametric ansatz form for the surrogate distribution and optimising these parameters. In our case we use the normal distribution in Equation 14 with parameters being the mean and covariance. In actuality the symmetry of the covariance matrix means that we optimise the elements of its Cholesky decomposition, L , where $\Sigma = LL^T$.

The exact value of the ELBO is not easily evaluable, and is instead estimated from the summation in Equation 19. This is a necessarily stochastic measure as it relies on a finite set of random samples. As such maximising the ELBO makes use of stochastic optimisation. In LITMUS’s case, use `numpyro`’s ready-built SVI tool set and the stochastic optimiser ‘adam’ (Kingma & Ba 2017). Aside from using SVI to estimate the Gaussian slices and the ELBO to estimate the slice evidence, the SVI Quadrature operates identically to the Laplace Quadrature in Section 4.1.

4.3.2. Nested Sampling

Nested sampling (Skilling 2006) operates by taking a uniformly distributed ensemble of ‘live points’ along with a statistical estimate of the volume of parameter space that they subtend. Its iterations slowly shrink and split the volume subtended by this ensemble, at each step keeping track of changes to this volume. The result is an ordered series of Lebesgue integral elements describing the nominal posterior density and volume of a series of contours, the summation of which estimates the evidence integral.

In LITMUS we include nested sampling by way of `JAXNS`, a `jax`-accelerated implementation that combines the Gaussian shell approach of `MultiNEST` (Feroz et al. 2009) with the Slice sampling of `PolyChord` (Handley et al. 2015). Unlike the Laplace and SVI quadrature methods, no assumptions of Gaussianity are made, and so this approach is unconditionally convergent in the limit of large sampling. For the specifics

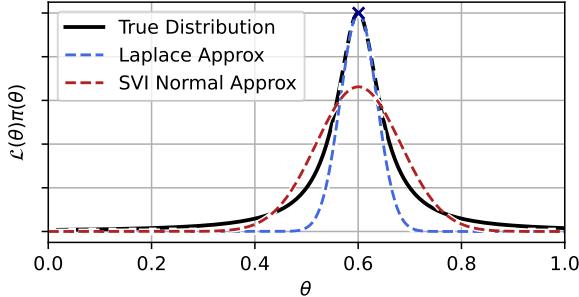


Figure 5. A demonstration of the difference in the Laplace and SVI approximations, both attempting to emulate a Cauchy distribution (black solid line). The Laplace approximation (blue dotted line) creates a Gaussian that matches the curvature at the MAP of the true distribution, and in this case under-estimates the distribution everywhere else. The SVI approximation, here also fitting a Gaussian, instead tries to get as close as possible to the true distribution ‘on average’, and so under-estimates in the core region while balancing the impact on the evidence integral with over-estimates in the distribution’s tails.

of Nested Sampling, including its convergence diagnostics and uncertainty estimates, we direct the reader to Ashton et al. (2022) for a more detailed review.

4.4. Hypothesis Testing

LITMUS’s biggest improvement over existing methods is its ability to not only constrain the best fit lag, but determine the significance of there being a lag at all. We can demonstrate this feature on mock data, using models from Section 4.2 and Bayesian evidence using the algorithms in Section 4.1 and Section 4.3. As an example we can use the prickly $\Delta t = 540$ d case, shown in Figure 6, which falls into one of the aliasing gaps and so is maximally ambiguous. At this lag we generate six mock cases, shown in the top panel, broken into high SNR and low SNR examples:

- A continuum and response that actually encodes a lag.
- That same continuum, but with a response from a different mock such that it is still a DRW with the same timescale, but encoding no meaningful lag response.
- The same continuum again but with a response signal that is pure white noise, encoding no signal structure whatsoever.

In this way we can use Bayes factors (ratios of the models’ Bayesian evidences) to test null hypotheses via model comparison. Here we formalise this process as asking two questions: is there a signal in the response, and if so does this signal demonstrate a significant lag? We consider a “strong” result to be a Bayes factor of $100\times$ or more, and a moderate result to be a factor of $10\times$ or more. Using the Laplace Quadrature we find that we successfully identify the true lag where it exists, and our evidence calculations allow us to identify and exclude the spurious recoveries for the two false mocks at high SNR.

Figure 6 contains a demonstration of LITMUS’s ability to use Bayes factors to distinguish if a signal is lag-carrying and/or contains structure. We simulate mock light curves emulating the cadence and uncertainty of OzDES in two cases: mocks with a low measurement uncertainty and easily observable slow fluctuations with a variational timescale

signal of $\tau = 200$ d, and mocks with a higher measurement uncertainty and a more rapid variational timescale of $\tau = 50$ d that is difficult to distinguish from white noise with such a coarse observational cadence. For each case we simulate a single continuum but three different response signals: a simulated reverberation response with a lag of $\Delta t = 540$ d, a response that follows the same DRW but is decoupled such that there is no observable lag (equivalent to $\Delta t \rightarrow \infty$) and a response that is drawn purely from white noise (equivalent to a variational timescale of $\tau = 0$). Using the Laplace Quadrature’s evidence integrals, LITMUS can successfully recover the truth for all tests at strong significance (all true positives and true negatives) in the high SNR case, and returns no false positives even when the constraints are weak.

5. Validation of Results

The example in Section 4.4 demonstrates LITMUS’s success on a small number of mocks, but it is necessary to also show that it performs reliably at scale. In this section, we test and compare LITMUS’s performance against existing methods on three sets of mocks:

1. A set of 440 mock AGN with lags distributed uniformly in $\Delta t \in [0, 1000]$ d and timescales drawn log-uniformly in $100 - 1000$ d.
2. A similar sample of 490 mocks with lags drawn from a log-normal distribution, $\Delta t \sim \mathcal{N}(2, 0.4)$, roughly emulating the spread of the OzDES MgII sample.
3. The same set of 490 mocks in set 2 but with the response curves randomly regenerated to create lag-free mocks, similar to the decoupled responses in Fig 6.

Mock set 1 allows us to interrogate LITMUS’s performance in lag recovery and its resistance to aliasing, particularly in contrast to JAVELIN’s AEIS method, while mock sets 2 and 3 allow us to examine how the Bayes factor allows us to screen for false positive lags. The decoupled mocks are of physical interest as they correspond to the case of the lag being longer upper limit detectable in a particular survey, as can happen for longer observer-frame CIV lags (Penton et al. 2025). Mocks are all generated with continuum and response amplitudes of unity and zero mean, with weekly cadence and $\approx 1\%$ uncertainty for the continuum observations and monthly cadence with $\approx 10\%$ uncertainty for the response light curves.

For each mock in each set, we recover the lag with LITMUS’s Laplace Quadrature, SVI Quadrature and Nested Sampling fitting methods, as well as its in-built ICCF and JAVELIN-like AEIS fitting methods.^c The fitting parameters for each method are listed in Appendix C. For a fair comparison we only include mocks for which all five fitting methods are have fully converged. For the Nested Sampling method we use 10^4 live points, enough to ensure good performance, so that this well established algorithm can act as a benchmark to test the other four fitting methods against. In this benchmarking, use the rough metric of a lag being correct or

^cThe AEIS method does not map exactly to a true JAVELIN fit as no smoothing is applied from a transfer function and fitting is performed in the unconstrained domain (See Appendix A).

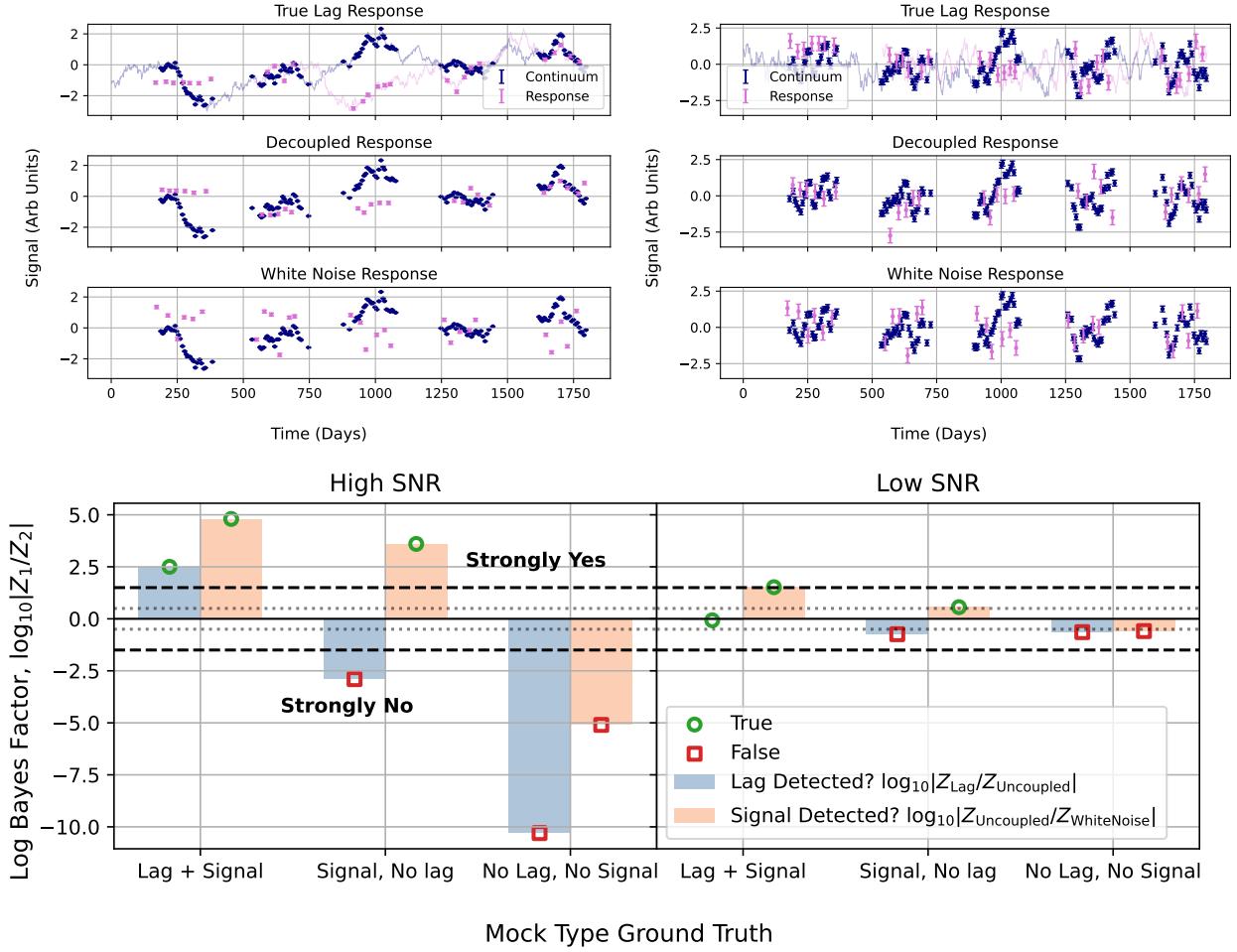


Figure 6. Mocks and evidence ratios for the demonstrative mock signals in the body of the text. The top panels shows the mock light curves for continuum (blue) and response (pink) signals. The left and right columns correspond to high and low SNR, while the rows from top to bottom show the mocks for a coupled response at a lag of $\Delta t = 540$ d, a decoupled response and a pure white noise response. The bottom panel shows the differences in log-Bayes factors for each of the hypotheses in Table 1, with orange bars being how strongly we can see structure in the response signals and blue bars being how well we can confirm the existence of a lag from this structure, with dotted lines indicating different significance levels in favour of accepting or rejecting these hypotheses. Red and green markers dots indicate the ground truth for each question: green circles mean true, red squares mean false.

Mock Type	$\log_{10} Z/Z_{\text{WhiteNoise}} $					
	Low SNR Mocks			High SNR Mocks		
	Lag + Structure	Structure, No Lag	No Lag or Structure	Lag + Structure	Structure, No Lag	No Lag or Structure
Lag Response	7.3	4.8	0.0	1.5	1.5	0.0
Uncoupled Response	0.7	3.6	0.0	0.2	0.6	0.0
White Noise Response	-15.4	-5.1	0.0	1.2	0.6	0.0

Table 1. Bayes factors (log scale) of model evidences when the different mocks in Figure 6 are fit with a model that encodes a lag response or that encodes an uncoupled but still structured response signal as compared to a model in which the response is unstructured noise. The bottom panel of Figure 6 shows the Bayes factors from these evidences that are used to test different hypotheses / compare the relative strength of the different models.

incorrect if it agrees / differs from the true lag by a threshold of 30 d.

As can be seen in Figure 7, LITMUS's aliasing friendly fitting methods produce posteriors that map closely to the true mock lags save a few outliers that can be mostly screened with significance tests (see Table 2). We can also see that the Laplace Quadrature's assumption of Gaussianity does not significantly disrupt this result as compared to Nested Sampling. Conversely, the results from the JAVELIN-like AIES are heavily obscured by the impacts of aliasing, with

more than half the posterior samples sitting in the seasonal striations of the aliasing bands. Even before screening spurious lags with Bayes factor tests, LITMUS's various fitters are significantly better than the AEIS at identifying correct lags.

In Figure 8, bottom panel we can see how the Bayes factor acts as a measure of the reliability of lag recoveries. As the Bayes factor between the models of lag-bearing and decoupled response becomes smaller, the error in the recovered lag (measured from the posterior median) rapidly increases.

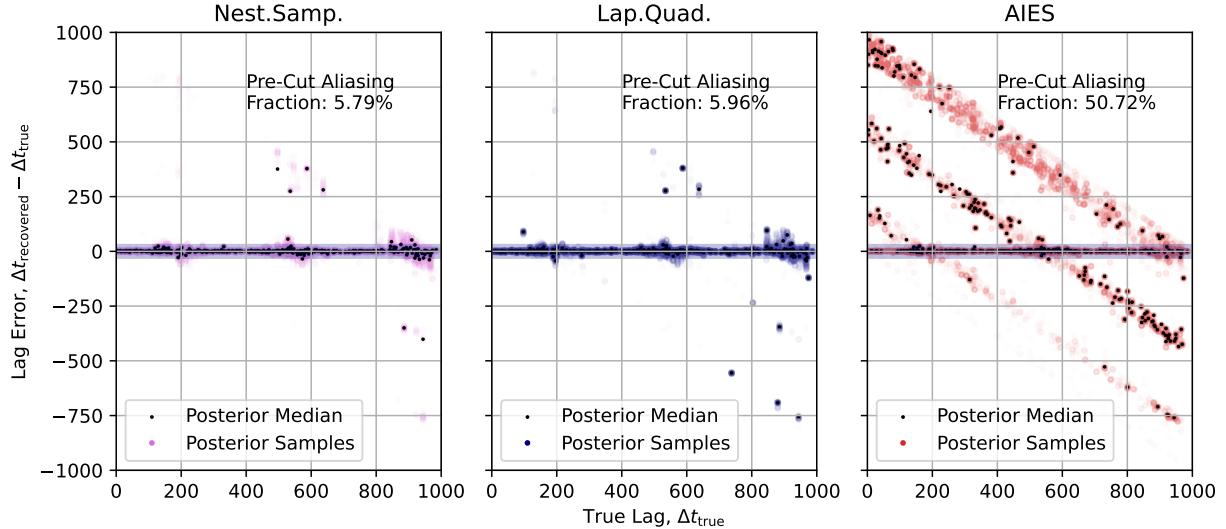


Figure 7. A comparison of the posterior distributions for the lag error, i.e. the difference between true and recovered lag, comparing some of LITMUS’s aliasing-friendly methods, namely Nested Sampling (left panel) and the Laplace Quadrature (middle panel), to the JAVELIN-like AEIS (right panel). These plots are for mock sample 1 which has 440 mocks with true lags distributed uniformly over the range $\Delta t \in [0, 1000]$ d. The aliasing fraction is the fraction of samples / posterior density that sits more than 30 d from the true value for these mocks. The Laplace Quadrature and Nested Sampling results adhere extremely closely to the true lags save for a single errant false positive, and the similarity between the two validates the Laplace Quadrature’s recovery of the true posterior shape. Conversely, more than half the AEIS samples are incorrect, and the posterior median (black dots) is often far from the true value.

Figure 8 also shows in its top panel how the Bayes factor helps to separate signals that do and do not encode lags within our prior range, with the decoupled population (mock set 3) sitting almost entirely at Bayes factors of $Z_2/Z_1 < 1$. This shows how the Bayes factor offers a direct and easily implementable means screening low confidence lag recoveries, and how this quality cuts can be tuned to a nominal level of reliability.

We compare all of LITMUS’s fitting methods, in concert with a quality cut of enforcing $Z_2/Z_1 \geq 10$, to the existing methods of lag recovery (JAVELIN-like AEIS and PyCCF-like ICCF) in Table 2. To roughly emulate the quality cuts of the MgII OzDES sample, we employ the first 2 of Yu et al. (2023)’s quality cuts on the AEIS results, namely that the width of the AEIS posterior, as measured between the 16th and 84th percentiles, is less than 110 d, and that the AEIS and ICCF methods agree to within 2σ , where σ is the standard deviation of the AEIS posterior. For the ICCF results we retain only results that agree with the AEIS to within 100 d, and for which the standard deviation of the bootstrapped lag recoveries are also less than 100 d.

Prior to applying cuts, LITMUS’s alias-friendly methods identify the correct lag in both the uniform and log-normal samples with a False Positive Rate (FPR) of < 5%, while the ICCF and AEIS methods recover the incorrect lag at an FPR of $\approx 40\%$ or higher in these mock sets. After discarding sources with a Bayes factor $Z_2/Z_1 < 10$, the LITMUS results improve even further, and importantly retain the vast majority of true recoveries. By contrast the AEIS and ICCF methods, while still seeing a decrease in FPR, do so at the cost of discarding much of the sample. Similarly in the entirely decoupled mock set 3, which contains only false positives, the Bayes factor test removes $\approx 98 - 99\%$ of these sources while the ICCF and AEIS cuts retain almost ten

times more. In short, LITMUS’s methods identify the correct lag more often and performs much better at discarding incorrect lags.

Note that these result should not be considered indicative of the actual FPR in the published OzDES H β , MgII and CIV lags, as these mocks samples do not fully emulate the physical OzDES sample, nor do they apply all of OzDES’s stringent selection criteria. With that being said, the improved performance of LITMUS’s methods in fitting lags over ICCF and AEIS are clear.

6. Discussion & Future Work

LITMUS’s improved handling of the aliasing problems of multi-year RM surveys means that, by being applied to existing data, it stands to improve the statistical power of our RM results without the need for new observations. McDougall (2025) found that the collective constraining power of the entire literature of RM measurements is sufficient to constrain the high redshift $R - L$ relationships such that the statistical uncertainty is subdominant compared to the inherent population noise. However, it is not confirmed whether a single $R - L$ relationship is sufficient to explain all AGN behaviour over all time. Current constraints are not enough to test the possibility of time-varying $R - L$ relationships, and the existing sample is restricted to only a narrow window of redshift-luminosity space. Through LITMUS’s hypothesis testing we can not only successfully recover more lags, but also expand their breadth in parameter space, and so will be equipped to begin using RM as a probe of more expressive physical models.

One such question is McDougall (2025)’s finding that the lags associated with the MgII line were systematically and significantly larger than those of H β , indicating that the MgII emission region of the BLR may be exterior rather than cospatial with that of H β , contrary to existing observations

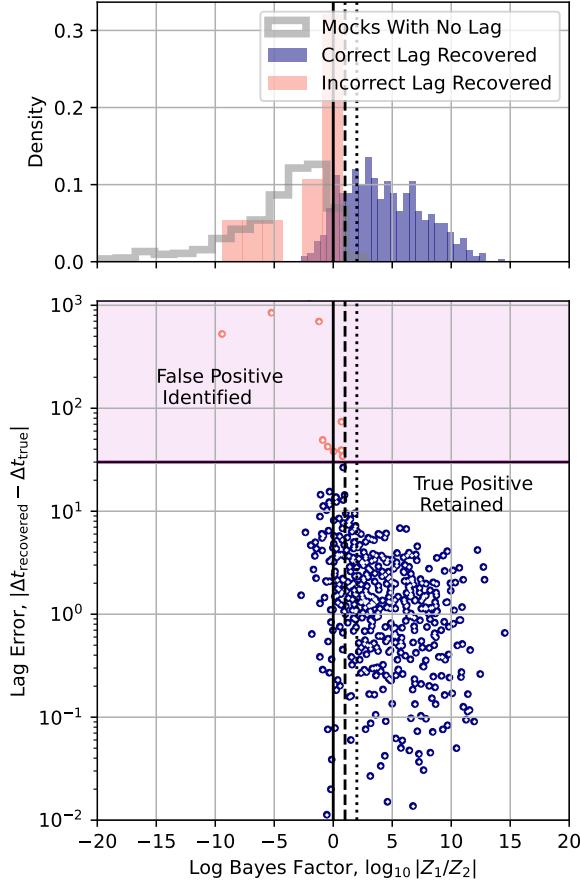


Figure 8. A demonstration of how the Bayes factor acts as a measure of lag reliability. The top panel shows histograms of the Bayes factor evidence ratios for the decoupled mocks with no lag (grey), mocks with a lag that was successfully recovered (navy) and mocks that had an underlying lag but for which the posterior median of the recovery was more than 30 d from the ground truth. The bottom panel shows how the error in the lag (here, the deviation between ground truth and posterior median) decreases for strong Bayes factors. As the evidence ratio for the lag and decoupled models lowers, the error in the median recovered lag rapidly increases, and above some reasonable threshold (e.g. $Z_2/Z_1 > 10^2$, the results become significantly more reliable. The solid, dashed and dotted lines represent evidence ratio thresholds of 1:1, 1:10 and 1:100 in favour of a lag. The correct and incorrect lags are for the 490 realistic mocks in mock set 2, while the mocks with no lag are from the decoupled mock set 3.

(e.g. Shen et al. 2019). This result was significant at barely 2σ , meaning even a small increase in the number of MgII lags would could confirm or falsify this finding. OzDES and SDSS have both completed their final RM data releases, meaning a sample of nearly 2000 AGN light curves are available for LITMUS to be applied to.

LITMUS is a general purpose single-lag RM tool, and so can be applied beyond this BLR RM domain. The same GP statistical model can be applied to lensed quasar time delay without much alteration (e.g. see the GP based approaches in Ding et al. 2021), as well as any other single-lag estimation applications. Because LITMUS’s statistical models can be extended or swapped out, it can also be aimed at answering some of the stickier questions of BLR RM, such as means of accounting for the BLR holiday (Dehghanian et al. 2019) or descriptions of outlier measurements.

	FPR Before Cuts	FPR After Cuts	Retained Sources
Mock Set 1 - Uniform Lags			
Laplace Quadrature	4.32%	0.57%	80.23%
SVI Quadrature	4.09%	0.59%	77.27%
Nest. Samp.	3.18%	0.00%	52.05%
JAVELIN-Like AIES	41.59%	12.90%	14.09%
ICCF	43.41%	30.34%	20.23%
Mock Set 2 - Log-Normal Lags			
Laplace Quadrature	4.29%	1.30%	94.08%
SVI Quadrature	2.45%	1.07%	95.10%
Nest. Samp.	2.24%	0.00%	79.39%
JAVELIN-Like AIES	66.73%	61.54%	79.39%
ICCF	39.39%	18.75%	9.80%
Mock Set 3 - Decoupled Response			
Laplace Quadrature	-	-	0.98%
SVI Quadrature	-	-	0.78%
Nest. Samp.	-	-	1.37%
JAVELIN-Like AIES	-	-	13.28%
ICCF	-	-	8.01%

Table 2. Summary of the performance of LITMUS’s three fitting methods, the Laplace Quadrature, SVI Quadrature and Nested Sampling approaches, as compared the AEIS fitting method used by JAVELIN and the ICCF method use by PyCCF, as tested on the three sets of mock light-curves (uniform in lag, log-normal in lag to emulate the OzDES MgII sample, and mocks with the response light curve decoupled such that no true lag is present). Listed are the fraction of false positive lags before and after quality cuts (a lag here being considered incorrect if it differs by more than 30 d from the ground truth), as well as the total number of retained sources after cuts. In general, the LITMUS fitting methods perform significantly better at identifying a true lag where it exists, with a pre-cut FPR < 5% in all cases, reducing by a factor of a few when removing sources with a lag recovery evidence ratio $Z_2/Z_1 < 10$. Overall, LITMUS yields significantly more and and significantly more accurate lags, while also retaining $10 - 20 \times$ fewer spurious lags from the decoupled sample.

Acknowledgements

Parts of this research were conducted by the Australian Research Council Centre of Excellence for Gravitational Wave Discovery (OzGrav), through project number CE230100016. HMG & BJSP acknowledge the support of the DECRA fellowship DE210101639, funded by the Australian Government through the Australian Research Council.

The authors would like to thank Louis Desdoigts for his experience and advice in writing LITMUS, and Rahma Alfarsy of the DESI collaboration for aiding in testing its initial stages. We thank also Madeline L. Cross-Parkin for her assistance in formatting the plots in this paper and LITMUS’s visual style. All software was made using python (Van Rossum & Drake 2009) and with the aid of numpy (Harris et al. 2020). Plots and figures were generated with the aid of matplotlib (Hunter 2007) and chainconsumer (Hinton 2016).

We acknowledge and pay respect to the traditional owners of the land on which the University of Queensland is situated, upon whose unceded, sovereign, ancestral lands we work. We pay respects to their Ancestors and descendants, who continue cultural and spiritual connections to Country.

We acknowledge the traditional custodians of the Macquarie University land, the Wallumattagal clan of the Dharug nation, whose cultures and customs have nurtured and continue to nurture this land since the Dreamtime. We pay our respects to Elders past and present.

Appendix A. The Constrained & Unconstrained Domain

Many Bayesian fitting algorithms encounter difficulties with hard boundaries on model priors, and perform best when parameters are “unconstrained”, i.e. with a domain of $\theta \in \mathbb{R}$, $-\infty < \theta < \infty$. Such hard boundaries have infinite gradients in log-probability and other pathologies that can lead to artefacts or non-convergence with some fitting methods. This is a particular concern in RM, where convention is to use bounded uniform priors on all parameters. `numpyro` obviates these difficulties with a statistical slight of hand. Prior to fitting, all parameters with constrained priors are shifted with a change of variables into a new coordinate space in which they are unconstrained, e.g. for a Uniform distribution on $\theta \sim U_{a,b}(\theta)$ there’s a mapping $f : (a, b) \rightarrow \mathbb{R}$, $\theta' = f(\theta)$.

KL divergences and evidences are preserved across this transformation, which changes the posterior density like:

$$P(\theta) = P(\theta) \times J(\theta), \quad (20)$$

where $J(\theta)$ is the Jacobian of the transformation $\theta \rightarrow \theta'$. `LITMUS`’s Laplace Quadrature and SVI Quadrature fitting methods (see Section 4.3) are all performed in this coordinate system by default, i.e. they assume that the posterior is approximately Gaussian in this unconstrained domain. Strictly speaking all densities should be written in terms of ϕ' and θ' in the description of these algorithms, but we exclude this for simplicity of reading.

Appendix B. The Grid Smoothing Algorithm

The Laplace Quadrature and SVI Quadrature algorithms divide the lag domain into an uneven grid of Gaussian slices. If these are spaced too sparsely we can miss important information about the posterior peaks, particular in the narrowly constrained modes of high SNR cases, and if they are packed too densely we can lose information about the long tails of the distribution, which can impact the accuracy of the evidence estimate (see Figure B-1). In the the idealised case, the two extremes are to have the points evenly spaced across the entire domain (Δt is constant) and having them distributed with spacing inversely proportional to the posterior density ($\Delta t \propto P(\Delta t)$).

In `LITMUS` we describe a sliding scale between these with a “grid bunching” parameter, α , where $\alpha = 0$ is an evenly spaced grid and $\alpha = 1$ is bunched proportional to the posterior. Values in between describe spacings whose cumulative distribution function (CDF) is a weighted average of these two extremes. First, we find some estimate of the best-fit parameters for the entire model, $\hat{\theta}$, and from this point forward fix all non-lag parameters at these values, $\hat{\phi}$, to produce a conditional distribution $\propto P(\Delta t | \hat{\phi})$. Estimating the entire marginal distribution $P(\Delta t)$ would be equivalent to solving the entire distribution, and so we form the grid based only on estimates of this much simpler conditional distribution.

For brevity, define

$$x = \frac{\Delta t - \Delta t_{\text{Min}}}{\Delta t_{\text{Max}} - \Delta t_{\text{Min}}} \quad (21)$$

and function $y(x) = \mathcal{P}(\Delta t(x) | \hat{\phi})$. The grid smoothing algorithm is as follows:

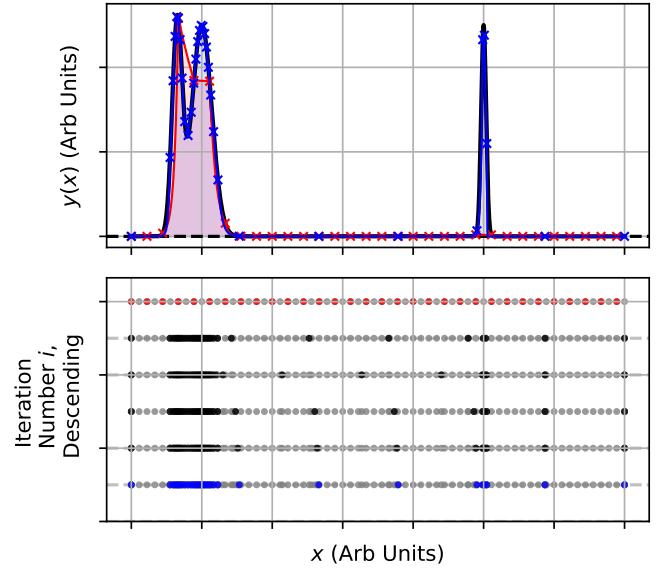


Figure 1. An exaggerated demonstration of the grid smoothing algorithm for a simple multimodal function using $\alpha = 0.8$ up to $j = 5$ iterations with 32 points. The top panel shows the true distribution (black) with its estimate from the first evenly spaced grid (red) and the final smoothed grid (blue). The bottom panel shows how the spacing of the grid updates over each iteration, progressing from top to bottom, with the first and last iterations coloured for emphasis, and gray dots representing samples from previous iterations. The initial spacing is so coarse that it misses much of the detail of the left mode, and cuts off the right-mode entirely. By final iteration, the estimate of the mode is significantly more accurate.

1. Start with a grid of I evenly spaced lags along $x \in [0, 1]$ and calculate $y(x)$ for each of these. These obey a cumulative distribution function (CDF) $Y^0(t) = x$, the CDF for initial iteration $j = 0$.
2. Linearly interpolate between all points and from this estimate the CDF of y : $Y = \int_0^y y(x') dx' / \int_0^1 y(x') dx'$.
3. Define a ‘half-step CDF’, $Y^{j+\frac{1}{2}}(x)$, that is the weighted average of this and a uniform distribution: $Y^{j+1}(x) = \alpha Y^j(x) + (1 - \alpha)x$
4. Draw a new set of I points whose spacings obey $Y^{j+1}(x)$, append these to the set of test lags.
5. Repeat steps (ii)→(iv) until converged, usually only a few steps.

Once converged (in Figure B this is after 5 iterations), we construct a grid of I ordered x values from the final $Y(x)$ and convert to the desired set of lags Δt^i with $\Delta t^i = \Delta t_{\text{Min}} + x_i^{j-1}(\Delta t_{\text{Max}} - \Delta t_{\text{Min}})$.

We find that $\alpha \in [0.5, 0.8]$ tends to give the best results over a wide range of lag posteriors.

Appendix C. Optimisation, Preconditioning & Convergence Tests

The Laplace Quadrature algorithm’s speed comes from its shifting of the integral problem into an optimisation problem. Good performance then relies on this optimisation being as robust and efficient as possible. To aid in the speed and reliability of this convergence, we use three tricks in the course of this optimisation:

1. Using point estimators of the model parameters for a good ‘seed’ location for the optimisation.
2. Using the optimum of each slice as the start for the following slice, and ordering these to be as smooth as possible, and
3. Preconditioning the function to be optimised to make the posterior as close as possible to a unit Gaussian.

Estimating Seed Parameters

To initialise the Laplace and SVI quadrature methods and the grid generation, we attempt to find the maximum a posteriori estimate (MAP), i.e. the highest point in the posterior density. This optimisation requires an initial ‘seed’ location. If not specified, this is calculated assuming stationary signal statistics to estimate the mean and amplitude of the continuum and response signals, i.e.:

$$\begin{aligned}\mu_{\text{cont/resp}} &= \left(\sum_i E_i^{-2} \right)^{-1} \sum_i E_i^{-2} y_i, \\ \sigma_{\text{cont/resp}} &= \left(\sum_i E_i^{-2} \right)^{-1} \sum_i E_i^{-2} (y_i - \mu)^2.\end{aligned}\quad (22)$$

To find a good starting value for τ , we use the ICCF method outlined in Section 3.2 to estimate the auto-covariance function of the continuum. This estimate becomes increasingly less accurate at high autocorrelation times δt , but drawing some window $|\delta t| < a$ it should follow close to the exponential form in Equation 8. Knowing this, we can say that $\ln |\text{ACF}(\delta t)| \approx -|\delta t|/\tau + 2 \ln |\sigma|$. Drawing some window $|\delta t| < a$ and inverting the $\delta t > 0$ side, this logarithm becomes roughly linear with slope $\frac{\partial y}{\partial \delta t}$. Taking a standard linear regression of this function, we can then estimate τ . Precision is not too important here, as the model parameter is $\ln |\tau|$ in GP fitting and so is insensitive to small deviations.

Moving from Slice to Slice

Under the assumption that the optimum is a smooth function of lag (see Figure 4 for an illustration), we use the solution for each slice as the starting location for the next slice. This is only done if this solution is accepted as being a valid representative of the marginal posterior’s local behaviour. Solutions are rejected if they:

1. Appear to be diverging, i.e. have undefined computation results, or
2. Do not have a positive definite covariance matrix (i.e. the Laplace approximation fails), or
3. Exhibit a severe drop in the peak posterior density compared to the previous slice, which would suggest the slice is in one of the furrows shown in Figure 2.

If an optimisation diverges, a second attempt is made by resetting the start location to the seed parameters. If this second attempt fails, the slice is discarded for all further calculations / processing.

Preconditioning

For finding the peak density, LITMUS uses the BFGS algorithm. This algorithm performs best in unit-quadratic functions, and we gain significant improvement in the convergence rate by linearly preconditioning the density function to warp it into this shape, i.e. rather than optimising $\ln |P_i(\phi)|$ we optimise $\ln |P_i(\phi(x))|$, $x = A(\phi - \phi_0)$, where ϕ_0 is an a priori estimate of ϕ_i and the starting location for optimisation, and A is a preconditioning matrix encoding the local principle axes and their widths. Assuming $P_i(\phi)$ is roughly Gaussian, then we can construct suitable A ’s from the Hessian evaluated at ϕ_0 in three ways:

1. From the Cholesky decomposition of the inverse Hessian, $AA^T = H^{-1}$.
2. Performing a PDP decomposition of H to find the covariance axes and corresponding Gaussian widths, $A = PD^{1/2}P$.
3. An approximate form of (2) where we take only the diagonal elements of H , i.e. no skewing / rotating of the axes.

For stability purposes, any negative or zero eigenvalues are fixed to unity in approaches (2) and (3).

Convergence & Uncertainty Estimates

Though the Laplace Quadrature’s approximation of the slice distribution log-densities as quadratic functions means that its convergent value is not exactly equal to the true model evidence, it is still necessary to constrain the deviation of our numerical calculation from this convergence point. There are two sources of numerical uncertainty in the calculation of Z : convergence of the optimiser to the true peak and the truncation error in the integral.

In Equation 17, errors in $\ln |Z_i|$ are determined to first order by the estimate of the peak density $\ln |P_i(\hat{\phi})|$. Under a simple Newtonian optimisation, the next required ‘uphill’ step is $\Delta\phi = H^{-1}\nabla f$, and the associated uncertainty in the peak density is $\nabla f^T \Delta\phi$, i.e.:

$$\Delta \ln |Z_i| = -\nabla f^T H^{-1} \nabla f. \quad (23)$$

Conveniently, this also has a natural interpretation in terms of the ‘closeness to the peak’. Noting that the Hessian is the negative inverse of the covariance matrix, this may also be written $-\Delta\phi^T H \Delta\phi = \Delta\phi^T C^{-1} \Delta\phi$, i.e. the square ‘number of standard deviations’ that the estimated solution is from the peak under the Gaussian approximation. This gives both a measure of the numerical uncertainty in $\ln |Z_i|$ and also a natural measure of ‘closeness’ to the optimum.

The second source of numerical uncertainty is the integration error due to the finite number of slices in the quadrature. To estimate this, we treat the integral as having $E_{\text{int}} \propto \mathcal{O}(h^2)$ error scaling and sub-sample the slices, using only every second value to get a second less precise estimate $Z_{1/2} \pm 4 \times E_{\text{int}}$. Approximating the two estimates as having uncorrelated errors, their difference is $Z - Z_{1/2} = 0 \pm \sqrt{E_{\text{int}}^2 + (4 \times E_{\text{int}}^2)}$, from which we can estimate the integral uncertainty E as:

$$E_{\text{int}} \approx \frac{1}{\sqrt{17}} |Z - Z_{1/2}| \quad (24)$$

The convergence uncertainty always under-estimates Z_i at each slice, $\Delta \ln |Z_i| < 0$, and so we assume a worst case sum for the total convergence uncertainty. These two sources are then added in quadrature for a total evidence error estimate:

$$\Delta Z^2 = E_{\text{int}}^2 + \left(\sum_i Z_i \times \exp(\Delta \ln |Z_i|) \right)^2. \quad (25)$$

For the uncertainty in the SVI Quadrature method, we use Equation 25 but with the $\Delta \ln |Z_i|$ of Equation 23 instead estimated from the standard deviation of the ELBO over multiple SVI iterations assuming a $1/\sqrt{N}$ scaling of the uncertainty.

It is worth emphasising that this represents only the numerical uncertainty arising from convergence of the Laplace Quadrature and SVI Quadrature algorithms, not the uncertainty in the evidence that arises from the Gaussian approximation of the posterior density.

Computational Speed & Tuning

A major feature of LITMUS is that it offers complete Bayesian integrals without exorbitant computational cost. As a part of the scale-testing in Section 5, we measure also the full runtime of the different algorithms, inclusive of null hypothesis model fitting for those methods that can do evidence integrals. The examples here are deliberately under-tuned, with default values for most fitting parameters (see Table 1) but with the number of iterations in the SVI Quadrature ELBO calculations at each slice deliberately set high to ensure convergence.

Shown in Figure C-1, we can see that LITMUS’s various fitting methods are of comparable speed to the JAVELIN-like AEIS with default parameters while still being significantly more reliable (see

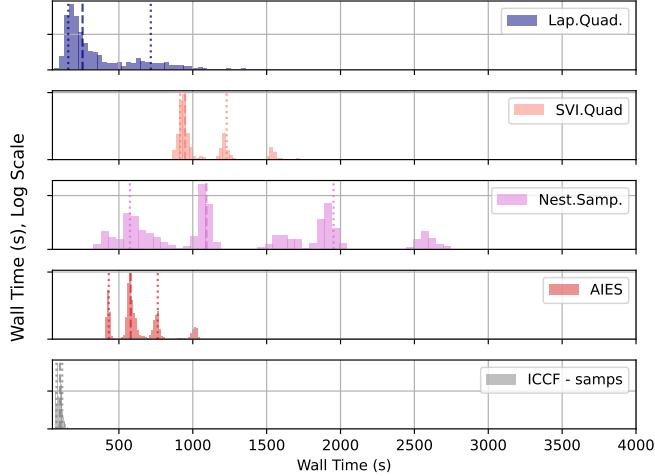


Figure 1. Histogram of the run-times for the five fitting methods in `LITMUS` over all mocks, using the fitting parameters described in Table 1. The ICCF method, which requires no matrix inversion owing to its absence of GP fitting, is consistently the fastest. The Laplace Quadrature can perform very fast except for cases where it get stuck optimising at a new test lag when the local optimum changes quickly over the lag axis. The SVI Quadrature has a similar issue, but runs overall somewhat slower.

Section 5). The multimodality in these distributions comes from periods in the fitting of mocks with inconvenient posteriors during which proposals / new live points are difficult to find.

It is difficult to fairly compare the run times of different algorithms due to the long list of tuning parameters for each method. With proper tuning, the SVI Quadrature and Laplace Quadrature algorithms can run markedly faster. Note also that this is not a direct comparison with `JAVELIN`, only with `LITMUS`'s example implementation of its fitting method, which is still built in `jax` and `numpyro`. In this way Figure C-1 is a comparison of the speed of the algorithms and not the codes themselves.

Laplace Quadrature Parameters	
Number of Test lags	128
Grid Bunching Parameter	0.5
Slice Optimisation Tolerance	0.01
Slice Optimisation BFGS Step Size	0.001
Max Optimisation Steps	1000
Slice Discard Log-Density Threshold	10000
SVI Quadrature Parameters	
Grid Bunching Parameter	0.5
SVI ELBO Optimisation Steps	1000
SVI Slice Discard ELBO Threshold	1000
ELBO adam Step Size	0.005
Number of ELBO Particles	128
Nest. Samp. Parameters	
Number of live points	1000
Max samples	10000
Target evidence uncertainty	0.001
AEIS Parameters	
Number of Walkers in Ensemble	256
Total Number of Samples	200000
Warmup Samples	5000
ICCF Parameters	
Number of Lags in Grid	512
Number of Bootstraps	512
Number of Times for Light Curve Interpolation	1024

Table 1. Tuning parameters for the fitting methods used in Section 5.

References

Aigrain, S., & Foreman-Mackey, D. 2023, *ARA&A*, 61, 329

Albert, J. G. 2020, JAXNS: a high-performance nested sampling package based on JAX, *arXiv:2012.15286*

Ashton, G., Bernstein, N., Buchner, J., et al. 2022, *Nature Reviews Methods Primers*, 2, doi:10.1038/s43586-022-00121-x

Betancourt, M. 2018, A Conceptual Introduction to Hamiltonian Monte Carlo, *arXiv:1701.02434*

Blandford, R. D., & McKee, C. F. 1982, *ApJ*, 255, 419

Bradbury, J., Frostig, R., Hawkins, P., et al. 2018, JAX: composable transformations of Python+NumPy programs

Broyden, C. G. 1970, *IMA Journal of Applied Mathematics*, 6, 222

Cackett, E. M., Bentz, M. C., & Kara, E. 2021, *iScience*, 24, 102557

Cackett, E. M., Horne, K., & Winkler, H. 2007, *Monthly Notices of the Royal Astronomical Society*, 380, 669–682

Dehghanian, M., Ferland, G. J., Kriss, G. A., et al. 2019, *The Astrophysical Journal*, 877, 119

Ding, X., Treu, T., Shajib, A. J., et al. 2020, Time Delay Lens Modeling Challenge: I. Experimental Design, *arXiv:1801.01506*

Ding, X., Treu, T., Birrer, S., et al. 2021, *Monthly Notices of the Royal Astronomical Society*, 503, 1096–1123

Donnan, F. 2021, PyROA: Modeling quasar light curves, *Astrophysics Source Code Library*, record ascl:2107.012

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. 1987, *Physics Letters B*, 195, 216

Fausnaugh, M. M., Denney, K. D., Barth, A. J., et al. 2016, *ApJ*, 821, 56

Feroz, F., Hobson, M. P., & Bridges, M. 2009, *MNRAS*, 398, 1601

Fletcher, R. 1970, *The Computer Journal*, 13, 317

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, *Publications of the Astronomical Society of the Pacific*, 125, 306

Foreman-Mackey, D., Yadav, S., Tronsgaard, R., Schmerler, S., & theorashid. 2022, dfm/tinygp: tinygp v0.2.2, doi:10.5281/zenodo.6473662

Gaskell, C. M., & Peterson, B. M. 1987, *ApJS*, 65, 1

Goldfarb, D. 1970, *Mathematics of Computation*, 24, 23

Goodman, J., & Weare, J. 2010, *Communications in Applied Mathematics and Computational Science*, 5, 65

Grier, C. J., Martini, P., Watson, L. C., et al. 2013, *ApJ*, 773, 90

Grier, C. J., Trump, J. R., Shen, Y., et al. 2017, *ApJ*, 851, 21

Grier, C. J., Shen, Y., Horne, K., et al. 2019, *The Astrophysical Journal*, 887, 38

Handley, W. J., Hobson, M. P., & Lasenby, A. N. 2015, *MNRAS*, 453, 4384

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, 585, 357

Hernández Santisteban, J. V., Edelson, R., Horne, K., et al. 2020, *MNRAS*, 498, 5399

Hinton, S. R. 2016, *The Journal of Open Source Software*, 1, 00045

Hoormann, J. K., Martini, P., Davis, T. M., et al. 2019, *Monthly Notices of the Royal Astronomical Society*, 487, 3650–3663

Hunter, J. D. 2007, *Computing in Science & Engineering*, 9, 90

Kaspi, S., Smith, P. S., Netzer, H., et al. 2000, *The Astrophysical Journal*, 533, 631–649

Kelly, B. C., Bechtold, J., & Siemiginowska, A. 2009, *ApJ*, 698, 895

King, A. 2015, *Monthly Notices of the Royal Astronomical Society: Letters*, 456, L109–L112

Kingma, D. P., & Ba, J. 2017, Adam: A Method for Stochastic Optimization, *arXiv:1412.6980*

Koshida, S., Minezaki, T., Yoshii, Y., et al. 2014, *The Astrophysical Journal*, 788, 159

Kozłowski, S., Kochanek, C. S., Udalski, A., et al. 2010, *ApJ*, 708, 927

Li, Y.-R., Wang, J.-M., & Bai, J.-M. 2016, *The Astrophysical Journal*, 831, 206

Lidman, C., Tucker, B. E., Davis, T. M., et al. 2020, *Monthly Notices of the Royal Astronomical Society*, 496, 19–35

MacLeod, C. L., Ivezić, Ž., Kochanek, C. S., et al. 2010, *The Astrophysical Journal*, 721, 1014

Malik, U., Sharp, R., Penton, A., et al. 2023, *MNRAS*, 520, 2009

McDougall, H. e. a. 2025, IN PREPARATION - TO BE UPDATED UPON SUBMISSION

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, *The Journal of Chemical Physics*, 21, 1087

Minezaki, T., Yoshii, Y., Kobayashi, Y., et al. 2019, *The Astrophysical Journal*, 886, 150

Neal, R. M. 1996, *Monte Carlo Implementation* (New York, NY: Springer New York), 55–98

Penton, A., Malik, U., Davis, T. M., et al. 2021, *Monthly Notices of the Royal Astronomical Society*, 509, 4008–4023

Penton, A., McDougall, H. G., Davis, T. M., et al. 2025, IN PREPARATION - TO BE UPDATED UPON SUBMISSION, 000–000

Peterson, B. M. 1993, *PASP*, 105, 247

Phan, D., Pradhan, N., & Jankowiak, M. 2019, Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro, doi:10.48550/ARXIV.1912.11554

Rue, H., Martino, S., & Chopin, N. 2009, *Journal of the Royal Statistical Society Series B*, 71, 319

Schwarz, G. 1978, *The Annals of Statistics*, 6, 461

Secunda, A., Greene, J. E., Jiang, Y.-F., Yao, P. Z., & Zoghbi, A. 2023, *ApJ*, 956, 81

Shakura, N. I., & Sunyaev, R. A. 1973, *A&A*, 24, 337

Shanno, D. F. 1970, *Mathematics of Computation*, 24, 647

Shen, Y., Brandt, W. N., Dawson, K. S., et al. 2015, *ApJS*, 216, 4

Shen, Y., Hall, P. B., Horne, K., et al. 2019, *The Astrophysical Journal Supplement Series*, 241, 34

Shen, Y., Grier, C. J., Horne, K., et al. 2023, The Sloan Digital Sky Survey Reverberation Mapping Project: Key Results, *arXiv:2305.01014*

Skilling, J. 2006, *Bayesian Analysis*, 1, 833

Starkey, D., Horne, K., Fausnaugh, M. M., et al. 2017, *ApJ*, 835, 65

Starkey, D. A., Horne, K., & Villforth, C. 2015, *Monthly Notices of the Royal Astronomical Society*, 456, 1960–1973

Suganuma, M., Yoshii, Y., Kobayashi, Y., et al. 2006, *ApJ*, 639, 46

Uttley, P., Cackett, E. M., Fabian, A. C., Kara, E., & Wilkins, D. R. 2014, *The Astronomy and Astrophysics Review*, 22, doi:10.1007/s00159-014-0072-0

van Niekerk, J., Krainski, E., Rustand, D., & Rue, H. 2022, A new avenue for Bayesian inference with INLA, *arXiv:2204.06797*

Van Rossum, G., & Drake, F. L. 2009, *Python 3 Reference Manual* (Scotts Valley, CA: CreateSpace)

Woo, J.-H., Yoon, Y., Park, S., Park, D., & Kim, S. C. 2015, *The Astrophysical Journal*, 801, 38

Yu, Z., Kochanek, C. S., Peterson, B. M., et al. 2019, *Monthly Notices of the Royal Astronomical Society*, 491, 6045–6064

Yu, Z., Martini, P., Davis, T. M., et al. 2020a, *ApJS*, 246, 16

—. 2020b, *ApJS*, 246, 16

Yu, Z., Martini, P., Penton, A., et al. 2021, *Monthly Notices of the Royal Astronomical Society*, 507, 3771–3788

—. 2023, *Monthly Notices of the Royal Astronomical Society*, 522, 4132–4147

Zu, Y., Kochanek, C. S., Kozłowski, S., & Udalski, A. 2013, *The Astrophysical Journal*, 765, 106

Zu, Y., Kochanek, C. S., & Peterson, B. M. 2010, JAVELIN: Just Another Vehicle for Estimating Lags In Nuclei, *Astrophysics Source Code Library*, record ascl:1010.007, ascl:1010.007

—. 2011, *ApJ*, 735, 80