The Astronomical Journal, 166:164 (20pp), 2023 October https://doi.org/10.3847/1538-3881/acf5cc

Octofitter: Fast, Flexible, and Accurate Orbit Modeling to Detect Exoplanets

William Thompson[1], Jensen Lawrence[2], Dori Blakely[2,3], Christian Marois[2], Jason Wang[4], Mosé Giordano[5], Timothy Brandt[6], Doug Johnstone[1,7], Jean-Baptiste Ruffio[8], S. Mark Ammons[9], Katie A. Crotts[3], Clarissa R. Do Ó[10], Eileen C. Gonzales[11,13], and Malena Rice[12]

[1] Department of Physics and Astronomy, University of Victoria, Victoria, BC, V8P 5C2, Canada; wthompson@uvic.ca
[2] National Research Council of Canada Herzberg, Victoria, BC, V9E 2E7, Canada
[3] Department of Physics and Astronomy, University of Victoria, 3800 Finnerty Road, Victoria, BC, V8P 5C2, Canada
[4] Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) and Department of Physics and Astronomy, Northwestern University, Evanston, IL 60208, USA
[5] University College London, Gower Street, London WC1E 6BT, UK
[6] Department of Physics, University of California, Santa Barbara, CA, USA
[7] National Research Council of Canada Herzberg, 5071 West Saanich Road, Victoria, BC, V9E 2E7, Canada
[8] Department of Astronomy, California Institute of Technology, Pasadena, CA 91125, USA
[9] Lawrence Livermore National Laboratory, USA
[10] Center for Astrophysics and Space Sciences, University of California, San Diego, La Jolla, CA 92093, USA
[11] Department of Astronomy and Carl Sagan Institute, Cornell University, 122 Sciences Drive, Ithaca, NY 14853, USA
[12] Department of Astronomy, Yale University, New Haven, CT 06511, USA

Received 2023 June 2; revised 2023 August 2; accepted 2023 August 29; published 2023 September 20

Abstract

As next-generation imaging instruments and interferometers search for planets closer to their stars, they must contend with increasing orbital motion and longer integration times. These compounding effects make it difficult to detect faint planets but also present an opportunity. Increased orbital motion makes it possible to move the search for planets into the orbital domain, where direct images can be freely combined with the radial velocity and proper motion anomaly, even without a confirmed detection in any single epoch. In this paper, we present a fast and differentiable multimethod orbit-modeling and planet detection code called Octofitter. This code is designed to be highly modular and allows users to easily adjust priors, change parameterizations, and specify arbitrary function relations between the parameters of one or more planets. Octofitter further supplies tools for examining model outputs including prior and posterior predictive checks and simulation-based calibration. We demonstrate the capabilities of Octofitter on real and simulated data from different instruments and methods, including HD 91312, simulated JWST/NIRISS aperture masking interferometry observations, radial velocity curves, and grids of images from the Gemini Planet Imager. We show that Octofitter can reliably recover faint planets in long sequences of images with arbitrary orbital motion. This publicly available tool will enable the broad application of multiepoch and multimethod exoplanet detection, which could improve how future targeted ground- and space-based surveys are performed. Finally, its rapid convergence makes it a useful addition to the existing ecosystem of tools for modeling the orbits of directly imaged planets.

Unified Astronomy Thesaurus concepts: Exoplanet detection methods (489); Bayesian statistics (1900); Direct imaging (387)

1. Introduction

Instruments for directly studying exoplanets are steadily improving in sensitivity. Current facilities are now accessing planets less than 10 au from their stars. Below these separations, orbital motion can become significant over mere months. This will be especially true for facilities with high angular resolving power thanks to their larger apertures and/or shorter operating wavelengths. This is an advantage for those who wish to determine the orbits of already-known planets but presents a significant challenge when searching for new companions.

When planets move from observation to observation, naive image stacking causes their signals to blur out and fade away. Reaching a necessary integration in a single epoch hits practical scheduling constraints and eventually physical limitations—for a sufficiently faint planet, it would not be possible to detect a significant number of photons before it moves by a full-resolution element. These constraints apply equally to images as they do to integral field units and interferometers, including aperture masking interferometry (AMI) on JWST (Sivaramakrishnan et al. 2023), Very Large Telescope Interferometer GRAVITY (Collaboration et al. 2017), and even Atacama Large Millimeter/submillimeter Array (Wootten & Thompson 2009) in the case of accreting protoplanets.

A number of projects have sought to solve this challenge for image data only by compensating for orbital motion between epochs. These include a search for planets around Sirius B (Skemer & Close 2011), K-Stacker (Nowak et al. 2018; Le Coroller et al. 2020), PACOME (J. Dallant et al. 2023, submitted), the search for planets around ε Eri (Mawet et al. 2019; Llop-Sayson et al. 2021), and the search for additional HR 8799 planets by Thompson et al. (2023). These have now led to promising evidence for α Cen AB b (Le Coroller et al. 2022) and HR 8799 f (Thompson et al. 2023). Moving the analysis of direct imaging data into the orbital domain enables a

[13] 51 Pegasi b Fellow.

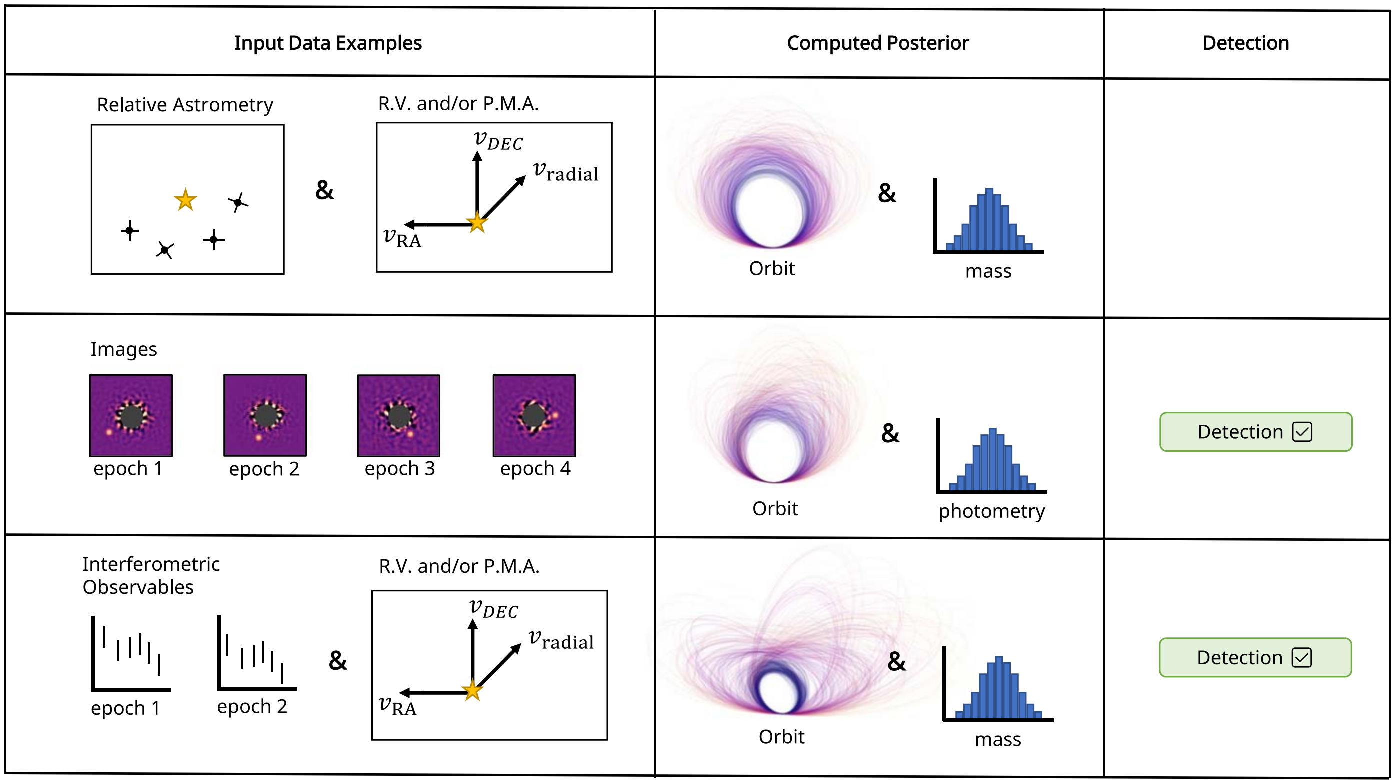

Figure 1. Conceptual schematic of three different ways to use Octofitter. Other possibilities, like combining images with Doppler or astrometric velocimetry or using a mix of relative astrometry and images without detections to constrain orbits, are also possible.

further extension: joint models of both images or interferometric observables with indirect exoplanet detection techniques, including the radial velocity (RV), astrometric motion, and transit (not directly considered in this paper). These have been previously explored by Mawet et al. (2019) and Llop-Sayson et al. (2021). This opens many possible scenarios. In addition to combining images to search for planets despite orbital motion, this allows one to freely combine Doppler or astrometric velocimetry with image data. This can then be used to constrain orbits using images with tentative detections or nondetections, improve photometry accuracy or limits, better constrain a planet's mass, and/or detect planets where any individual kind of data fails to reach significance (e.g., Llop-Sayson et al. 2021).

These scenarios are possible as all exoplanet detection methods provide orbital constraints that at least partially overlap. Imaging, RV, and transit all provide the orbital period (P); proper motion anomaly (PMA) and RV constrain eccentricity (e), the argument of periapsis (ω), and either the mass (m) or $m\,\sin(i)$, where i is the orbital inclination; and multiepoch imaging/interferometry constrains all orbital parameters up to a ± ambiguity on the longitude of the ascending node (Ω). These connections could, in principle, allow information from all methods to flow into a single orbit model and, ultimately, result in the detection of a new planet.

To apply these ideas broadly, the community will need a tool capable of modeling all different types of exoplanet data. The `orbitize!` (Blunt et al. 2020) and `orvara` (Brandt et al. 2021) packages come close: they support Bayesian modeling of relative astrometry, RV, and PMA. We needed a publicly available and generally applicable package that goes further to directly model image and interferometer data with or without independent detections at each epoch.

It is, generally, challenging to accurately compute orbital posteriors as the traditional Campbell orbital elements (a, e, i, ω, Ω, t_{peri}) possess complex codependencies and degeneracies (e.g., when $e = 0$ or $i = 0$). This task becomes even more challenging when working with short orbital arcs because they lack the constraining power to independently determine each Campbell element, meaning orbit posteriors are typically complex and very sensitive to their priors (e.g., O'Neil et al. 2019). Introducing image and interferometer data to the model exacerbates issues further, as they produce multimodal posteriors that are challenging to traverse. Any inaccuracies in the calculation of an orbit posterior can lead to errors in mass and/or photometry and spurious detection.

These challenges motivate the development of our new orbit- and data-modeling framework, "Octofitter."[14] Named after the eight types of data through which it aims to grasp new planets, Octofitter is designed from the ground up to be a flexible platform for modeling and experimentation while providing very high computational performance. These advances are thanks to our implementation of a pure-Julia (Bezanson et al. 2012), fully differentiable, and nonallocating modeling language, and our use of the higher-order No-U-Turn Hamiltonian Monte Carlo sampler (Xu et al. 2020). We further present the use of simulation-based calibration (SBC; Talts et al. 2020) to confirm the accuracy of orbital posteriors, which is made practical by Octofitter's speed. Figure 1 shows a schematic of a few of the ways Octofitter can be applied to exoplanet data.

[14] Octofitter is publicly available on the Julia General registry, and extensive usage examples are provided at https://sefffal.github.io/Octofitter.jl/dev/.

Table 1
Relative Astrometry Input Sample

epoch (mjd)	pa (°)	sep (mas)	σ_sep (mas)	σ_pa (°)	cor
58849.0	224.93	615.2	30.0	0.8	0.0
58879.0	228.53	606.4	30.0	0.8	0.0
58909.0	229.66	663.3	30.0	0.8	0.0
58939.0	232.48	635.9	30.0	0.8	0.0
58969.0	233.25	610.3	30.0	0.8	0.0
58999.0	235.22	669.7	30.0	0.8	0.0
59029.0	236.67	666.8	30.0	0.8	0.0
59059.0	237.85	654.3	30.0	0.8	0.0
59089.0	239.68	713.6	30.0	0.8	0.0
59215.0	245.67	747.3	30.0	0.8	0.0
59245.0	245.54	736.4	30.0	0.8	0.0
59275.0	247.94	709.7	30.0	0.8	0.0
59305.0	248.13	791.4	30.0	0.8	0.0
59335.0	251.17	777.8	30.0	0.8	0.0
59365.0	251.43	773.8	30.0	0.8	0.0
59395.0	250.44	866.5	30.0	0.8	0.0
59425.0	253.49	789.5	30.0	0.8	0.0
59455.0	253.98	839.1	30.0	0.8	0.0
59945.0	268.72	986.3	30.0	0.8	0.0
59975.0	268.89	941.6	30.0	0.8	0.0
60005.0	269.96	959.2	30.0	0.8	0.0
60035.0	270.0	928.9	30.0	0.8	0.0
60065.0	270.67	952.9	30.0	0.8	0.0
60095.0	272.02	977.5	30.0	0.8	0.0
60125.0	270.99	950.1	30.0	0.8	0.0
60155.0	272.15	953.9	30.0	0.8	0.0
60185.0	274.33	985.2	30.0	0.8	0.0

As a publicly available code with these advances, Octofitter will enable the wide application of multiepoch, multi-instrument, and multimethod exoplanet detection and modeling. These approaches can improve how direct surveys are completed and improve the yield of the upcoming Nancy Grace Roman Space Telescope Coronagraphic Instrument (Kasdin et al. 2020), the Habitable Worlds Observatory (HWO), and future facilities for extremely large telescopes.

2. Data Models

This section describes how the eight kinds of exoplanet data can be modeled by Octofitter. These data are the relative astrometry, images, interferometric observables, star and planet RVs (absolute and relative), and proper motion anomaly.

The Octofitter framework is structured around three concepts: likelihood functions for observations, system models to tie observations to parameters, and generative functions to create synthetic observations. These break the problem down into orthogonal components that can be freely combined to solve a wide range of orbit modeling problems.

Each kind of observation in Octofitter is supported by its own Julia data type. Every observation type is a wrapper for a data table of observations with a preset list of required columns. The observational data, in turn, can be provided to its associated type directly in the code, loaded from a local CSV or Arrow file, or obtained from a remote SQL database. For each observation type, a method of the `Octofitter.ln_like` function is provided that computes the likelihood of the data it contains given a specific set of parameters. We now describe the observation types and likelihood functions included in Octofitter.

2.1. Relative Astrometry

The position measured between a directly imaged planet and its host star is one of the most fundamental measurements gathered from direct observations. It can be extracted from any image where the planet is robustly detected in a single epoch. Relative astrometry can be expressed either as separation in milliarcseconds and position angle in degrees or as offsets in milliarcseconds in the R.A.–decl. tangent plane.

Relative astrometry measurements can be provided to Octofitter using `AstrometryLikelihood`, which accepts a table with columns for `epoch`, the date of the measurement in units of modified Julian days; `sep`, the separation between primary and secondary in mas; `pa`, the position angle of the secondary measured East from North; σ`_pa`, σ`_sep`, and `cor` to specify the measurement uncertainties and correlation, respectively. Alternatively, `ra` (relative), `dec` (relative), σ`_ra`, and σ`_dec` can be substituted if preferred. A sample relative astrometry input table is presented in Table 1.

Based on these quantities, the likelihood function is simply taken as the Gaussian likelihood that the residual between the model and measurement for each parameter would be seen, given the provided measurement uncertainty. For relative positions measured along the R.A. and decl. axes, this is

$$\log \mathcal{L}_{\mathrm{astrom}} = -\frac{1}{2}\frac{(\Delta\mathrm{R.A.} - \Delta\tilde{\mathrm{R.A.}})^2}{\sigma^2_{\Delta\tilde{\mathrm{R.A.}}}} - \log\sqrt{2\pi\sigma^2_{\Delta\tilde{\mathrm{R.A.}}}} - \frac{1}{2}\frac{(\Delta\mathrm{decl.} - \tilde{\mathrm{decl.}})^2}{\sigma^2_{\Delta\tilde{\mathrm{decl.}}}} - \log\sqrt{2\pi\sigma^2_{\Delta\tilde{\mathrm{decl.}}}}, \quad (1)$$

where ΔR.A. and Δdecl. are the separation from the star along the R.A. and decl. axes, $\sigma_{\Delta\mathrm{R.A.}}$ and $\sigma_{\mathrm{decl.}}$ are the corresponding uncertainties, and where the tilde distinguishes measured quantities from values calculated from model parameters. A more complex expression is used when the correlations between uncertainties are nonzero.

2.2. Images

Directly modeling point sources in images is one of the key features of Octofitter. This is accomplished with the same approach described in Ruffio et al. (2018) and applied in Mawet et al. (2019), Llop-Sayson et al. (2021), and Thompson et al. (2023). In the case where a planet is robustly detected in each image and the uncertainties are well-approximated by Gaussian uncertainties, it is mathematically equivalent to extracting the relative astrometry and photometry and then modeling those measurements. Where this approach goes beyond the two-step process, however, is when a planet is not detectable in a single epoch or when one wants to constrain the orbit of a planet based on a nondetection. The first case may arise when searching for planets that are too faint to detect before they exhibit orbital motion. The second case may arise any time a monitored planet passes too close to its star to detect. In this case, it is not possible to extract the astrometry. By directly modeling the images, large swathes of the orbital parameter space may still be ruled out. This can improve constraints on orbital parameters and improve predictions of the planet's future location.

In Octofitter, image data can be modeled across any number of photometric bands, instruments, and epochs. Data can be provided using the `ImageLikelihood` observation type,

Table 2
Images Input Sample

epoch (mjd)	band (symbol)	platescale (mas/px)	image (matrix)	contrast (function)
59976.0	`:H`	10.2	⋯	⋯
59976.0	`:J`	10.2	⋯	⋯
60576.0	`:H`	10.2	⋯	⋯

which accepts a data table with the columns `epoch`, `band`, `platescale`, `image`, and `contrast`. A sample input table is presented in Table 2. The plate scale and parallax system parameters (which may be fixed or fit to the data) are used to map an orbit to a pixel location in the image. The `contrast` entry for each image allows one to pass an arbitrary function of position that gives 1σ contrast. If not provided, Octofitter calculates a contrast curve automatically from the image itself. This is sufficient for images with no clear detection but should be avoided if the image contains a bright planet as the contrast curve will overestimate the noise at the planet's separation. Extended emission from disks is not currently considered.

Given these data, the likelihood function used by Octofitter for each image by Octofitter is that of Ruffio et al. (2018):

$$\log \mathcal{L}_{\mathrm{img}} = \frac{1}{2\sigma^2_{B,x,i}}(f_B^2 - 2f_B\,\tilde{f}_{B,x,i}), \tag{2}$$

where f_B is the model flux parameter for the photometric band B, x is the position in the image determined from orbital parameters, i is the epoch, σ is the uncertainty, and $\tilde{f}_B$ is the measured flux at that same location.

This likelihood function assumes that flux is constant from epoch to epoch, but could easily be adapted to use an orbital phase function for planets imaged in reflected light (e.g., Pogorelyuk et al. 2022). Taken to the extreme, the flux can be fit independently at each epoch (e.g., Nowak et al. 2018) at the expense of reduced constraining power. For both the images and relative astrometry, we do not currently consider uncertainty in the instrument's north angle or plate scale. These might need to be considered when combining data from multiple instruments, in which case they could be added straightforwardly. We also note that this likelihood function assumes Gaussian-distributed noise in each image. Where this is not a good assumption, standard techniques for inflating uncertainties could be used before applying Octofitter. Furthermore, spatial correlations between pixels are not currently handled. Finally, this likelihood function is only specified up to a constant and is not suitable for techniques like nested sampling.

2.3. Interferometric Observables

Just as with imaging, combining interferometric observations across multiple epochs using orbital modeling removes the need to detect a companion in a single observation. We implement a model assuming an unresolved point-source primary and N unresolved point-source companions. The complex visibilities of this model are given by

$$\mathrm{V}_{\mathrm{bin}} = \frac{1 + \sum_{i=1}^{N} f_i \exp(-2\pi \mathrm{i}(\Delta\mathrm{R.A.}_{i} u + \Delta\mathrm{decl.}_{i} v))}{1 + \sum_{i=1}^{N} f_i}, \tag{3}$$

Table 3
Visibilities Input Sample

epoch (mjd)	band (symbol)	pa (°)	sep (mas)	contrast
60096.0	`:F480M`	−92.7	180.5	0.00036
60171.0	`:F480M`	−61.5	159.1	0.00036
60462.0	`:F480M`	56.9	213.9	0.00036

where f_i is the (companion/primary) contrast of the ith companion, ΔR.A.$_i$ and Δdecl.$_i$ are the R.A. and decl. of the ith companion, and u and v are the Fourier domain coordinates, with magnitudes given by the interferometer baseline lengths divided by the observing wavelength (e.g., Berger 2003; Kammerer et al. 2023). A sample input table is presented in Table 3.

The squared visibilities are calculated from the squared moduli of the complex visibilities, and the closure phases are calculated by summing the phases of the three complex visibilities calculated from triangles of stations in the interferometer. We construct likelihood functions for the squared visibilities and the closure phases separately, assuming Gaussian noise statistics and diagonal covariance matrices. We also assume that there is at least a moderate contrast between the primary and any companions such that we can neglect phase wrapping in the closure phase (Éric & Young 2017). These likelihood functions are given by

$$\log \mathcal{L}_{\mathrm{CP}} = -\frac{1}{2}\sum_{i} \frac{(\mathrm{CP}_i - \tilde{\mathrm{CP}}_i)^2}{\sigma^2_{\mathrm{CP},i}} - \frac{1}{2}\log 2\pi\sigma^2_{\mathrm{CP},i} \tag{4}$$

and

$$\log \mathcal{L}_{\mathrm{V}^2} = -\frac{1}{2}\sum_{i} \frac{(\mathrm{V}_i^2 - \tilde{\mathrm{V}}^2{}_i)^2}{\sigma^2_{\mathrm{V}^2,i}} - \frac{1}{2}\log 2\pi\sigma^2_{\mathrm{V}^2,i}, \tag{5}$$

where CP$_i$ is the ith closure phase, V_i^2 is the ith squared visibility, and $\sigma_{\mathrm{CP},i}$, and $\sigma_{\mathrm{V}^2,i}$ are the uncertainties in the ith squared visibility and closure phase, respectively.

2.4. Radial Velocity

Similarly to other packages, Octofitter allows one to model RV data in combination with other observation types. If combined with relative astrometry, proper motion anomaly, or image data, it allows one to directly model the dynamical mass of a planet.

RV data can be specified for the star or the planet using `RadialVelocityLikelihood`, which accepts a table with columns for `epoch` in MJD, `rv` in m s^{-1}, `σ_rv`, the uncertainty in `rv` in the same units, and `inst_idx`. `inst_idx` is an integer between 1 and 4 used to specify which instrument the measurement corresponds to. The zero-points `rv0_i` and jitter `jitter_i` must be specified as variables in the model, where `i` corresponds to the instrument index.

A sample input table is presented in Table 4. The input format is the same for both cases. Depending on how the zero-point is modeled, it is possible to use either relative or absolute RVs. The zero-point can also be modeled as an arbitrary function of other variables, allowing one to fit, for example, linear trends.

When combining RV data with direct imaging modeling, it is possible (though not required) to connect the planet's photometry with its dynamical mass by using a user-supplied model. This model can either map a mass and/or system age to photometry in each band, or vice versa. Connecting these two variables may be useful in cases such as when the orbit is determined by the RV up to the inclination degeneracy but has not yet been detected with direct imaging.

In a similar manner to relative astrometry, we define an RV likelihood function that allows us to fit orbital parameters to RV data. This function is

$$\log \mathcal{L}_{\mathrm{rv}} = -\frac{1}{2}\sum_{i} \frac{(v_{\mathrm{r},i} - \tilde{v}_{\mathrm{r},i})^2}{\sigma^2_{v_{\mathrm{r}},i}} - \log 2\pi\sigma^2_{v_{\mathrm{r}},i}, \tag{6}$$

where $v_{\mathrm{r},i}$ is the measured RV, $\tilde{v}_{\mathrm{r},i}$ is the maximum likelihood estimate of the RV, and $\sigma_{v_{\mathrm{r}},i}$ is the uncertainty in the RV, all at the epoch i.

Octofitter supports multiple instruments, each with its own RV zero-point and jitter term and all with independently selectable priors. This is accomplished via the `inst_idx` column, which associates each RV measurement to a particular instrument, zero-point, and jitter. As a convenience, Octofitter includes a helper function to load RV curves from the publicly available High Accuracy Radial Velocity Planet Searcher RV Bank (Trifonov et al. 2020). Accessing these data will prompt the user to accept its license and then automatically fetch the RV curves.

2.5. Proper Motion Anomaly

Many systems that are candidates for direct imaging have their positions and proper motions measured accurately by both the Gaia (GAIA-Collaboration et al. 2021) and Hipparcos (van Leeuwen 2007) missions. The Hipparcos–Gaia Catalog of Accelerations (HGCA Brandt 2021) cross-calibrates measurements from these two satellites and inflates uncertainties to cover most known systematics. This results in projected velocity measurements of the system's photo center at the Hipparcos epoch, the Gaia epoch, and between the two.

In Octofitter, proper motion anomaly must be loaded directly from the HGCA by specifying the host star's Gaia identifier (the Data Release 3 version at the time of writing). As with the RV loaders, this will prompt the user to accept a license for the catalog and automatically download the HGCA.

In order to connect the proper motion anomaly with the orbital image modeling described above, we define a likelihood function based on the HGCA data and the same orbital parameters:

$$\log \mathcal{L}_{\mathrm{pma}} = -\frac{1}{2}\frac{(v_{\mathrm{R.A.}} - \tilde{v}_{\mathrm{R.A.}})^2}{\sigma^2_{\tilde{v}_{\mathrm{R.A.}}}} - \log 2\pi\sigma^2_{\tilde{v}_{\mathrm{R.A.}}} - \frac{1}{2}\frac{(v_{\mathrm{decl.}} - \tilde{v}_{\mathrm{decl.}})^2}{\sigma^2_{\tilde{v}_{\mathrm{decl.}}}} - \log 2\pi\sigma^2_{\tilde{v}_{\mathrm{decl.}}}, \tag{7}$$

where $v_{\mathrm{R.A.}}$ and $\tilde{v}_{\mathrm{decl.}}$ are the R.A. and decl. proper motion at each epoch and $\sigma_{\mathrm{R.A.}}$ and $\sigma_{\tilde{v}_{\mathrm{decl.}}}$ are the associated uncertainties. The input table format for this observation type matches Table 4 of Brandt (2021).

Compared with Brandt et al. (2021) and Brandt et al. (2021) we adopt a simpler model of the Gaia and Hipparcos data. Rather than use the exact epochs in which each mission scanned the star (or estimate using the Gaia Observation Forecast Tool[15]), we presume that the position and proper motion of the star were measured independently 25 times over the duration of each mission. For orbits with periods longer than the observation windows of Gaia (≈668 days) and Hipparcos (≈1227 days), using these 25 epochs approximates the smearing effect caused by the moving photo center during observations. Future work could refine this model further following their more sophisticated approach by re-implementing the calculations of HTOF (Brandt et al. 2021) in Julia. Once it is available, it should also be possible to use the full intermediate GAIA astrometry catalog in Octofitter.

Table 4
Radial Velocity Input Sample

`epoch` (mjd)	`rv` (m s^{-1})	`σ_rv` (m s^{-1})	`inst_idx`
58849.0	47.2342	5.0	1
58879.0	14.4347	5.0	1
58909.0	12.9645	5.0	1
58939.0	26.2633	5.0	1
58969.0	−8.27905	5.0	1
58999.0	4.94685	5.0	1
59029.0	0.863664	5.0	1
59059.0	10.9524	5.0	1
59089.0	3.92389	5.0	1
59215.0	21.0509	5.0	1
59245.0	29.2009	5.0	1
59275.0	16.77	5.0	1
59305.0	19.0421	5.0	1
59335.0	52.9403	5.0	2
59365.0	21.8173	5.0	2
59395.0	55.0851	5.0	2
59425.0	20.444	5.0	2
59455.0	0.915145	5.0	2
59945.0	21.6418	5.0	2
59975.0	−0.54618	5.0	2
60005.0	11.4391	5.0	2
60035.0	15.3402	5.0	2
60065.0	17.7835	5.0	2
60095.0	5.35458	5.0	2
60125.0	60.3179	5.0	2
60155.0	43.5441	5.0	2
60185.0	11.4066	5.0	2

3. Methods

3.1. Assessing Detections

This section describes how we choose to interpret the results of a model fit with Octofitter, though it should be noted that the modeling code makes no such prescriptions. For evaluating upper limits and detections from image or visibility data alone, we follow the conventions of Ruffio et al. (2018).

As set out in Ruffio et al. (2018), we calculate the upper photometry or mass limits by finding where the cumulative distribution function equals some threshold, e.g., 97.7%. That is, $\mathrm{CDF}(f_{\mathrm{lim}}) = 97.7\%$. This can be stated as an overall value for the system to say, for example, that we believe with 97.7% confidence that there are no planets present at all with photometry above some f_{lim}. It can also be calculated over small ranges of orbital parameters, in order to, for example, calculate a flux upper limit as a function of the semimajor axis. This value will be driven by the brightest speckles close to the

[15] https://gaia.esac.esa.int/gost/

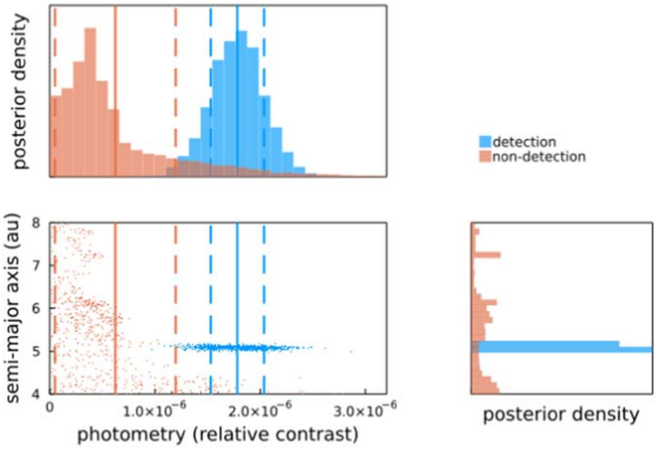

Figure 2. Posterior density of photometry vs. semimajor axis for a simulated detection and nondetection. The solid lines mark the mean of the marginal photometry posterior, and the dashed lines mark $\pm 1\sigma$. We consider a planet detected when the S/N based on the photometry marginalized over all other parameters is greater than some chosen threshold, e.g., 5σ.

star, so a much more useful metric is to provide $f_{\rm lim}$ as a function of the orbital parameters.

This model does presuppose that a planet exists and follows a Keplerian orbit around the star; however, as long as the photometry prior is broad, the model is free to drive the photometry toward zero. We can therefore use the photometry posterior to calculate a signal-to-noise ratio (S/N) analogous to the typical metric used in direct imaging. This approach has the benefit of being familiar to those with experience evaluating direct images for detections.

The procedure we adopt for assessing detections is to

1. sample from the posterior,
2. marginalize over all orbital parameters,
3. calculate the S/N as the mean of the marginal photometry posterior divided by its standard deviation,
4. and then compare this S/N with a preselected threshold based on a tolerable false-positive rate, e.g., 5σ.

The results of this procedure are visualized in Figure 2. This gives a single S/N value for the existence of any planet in that data set consistent with the user's choice of priors. This is somewhat different from the typical S/N detection thresholds used because this S/N is calculated by marginalizing over all credible orbits, or equivalently if applied to a single image, marginalizing over all credible positions. By contrast, the standard 5σ threshold is applied to the position in an image with the highest S/N, rather than a weighted combination of all positions.

In general, S/Ns obtained using this approach assume that

1. the noise is approximately Gaussian distributed,
2. the noise is uncorrelated along orbital tracks between images,
3. the images contain only a single planet,
4. and that planet closely follows a Keplerian orbit.

Deviations from assumptions 1 and 2 may increase the false-positive rate, while deviations from assumptions 3 and 4 may lower the recovered photometry and S/N.

Multiplanet systems could be accommodated easily by introducing additional planets to the model and/or restricting the priors to include only a subset of the parameter space.

For models that combine direct and indirect data, the relationship between mass and photometry variables may be complex, and this simple scheme may not be appropriate. In this case, it may be better to calculate a Bayes factor between planet and no-planet models. Such a Bayes factor can be treated as a direct measurement of our belief that a planet exists and follows a Keplerian orbit. This can be carried out in a limited fashion in Octofitter by calculating the Savage–Dickey density ratio of the mass or photometry marginal posterior (Dickey 1971; Koop 2003; for a pedagogical text, see Wagenmakers et al. 2010). This is more flexible as it does not require a uniform prior—any prior that includes zero mass or photometry would suffice—and because it does not presume that the marginal posterior is Gaussian distributed.

3.2. Orbital Bases and Priors

Octofitter supports a wide range of different orbital bases for use in different situations. These include traditional Campbell elements (a, e, i, ω, Ω), Thiele-Innes elements (e, A, B, F, G, Hartkopf et al. 1989; Wright & Howard 2009; O'Neil et al. 2019), Cartesian elements (x, y, z, v_x, v_y, v_z, Ferrer-Chávez et al. 2021a), and a reduced basis set for modeling the RV only (a, e, $m\sin(i)$, ω). Users can specify priors using arbitrary distributions from `Distributions.jl`[16] and functions of those distributions.

For the analyses presented in this paper, we adopt either Campbell elements or Thiele-Innes elements with the following priors and modifications. When using Campbell elements, we adopt a log-uniform prior on the semimajor axis, a uniform prior on eccentricity, a sine prior on inclination, a Gaussian prior on host mass, and uniform priors on the remaining angular parameters. Note that Octofitter reports the argument of periastron of the planet (and not the star) as ω and adopts $+z$ increasing away from the observer. These conventions match those used by `orvara` and `orbitize!` (Householder & Weiss 2023). The full conventions used in Octofitter are described in Appendix A. When using Thiele-Innes elements, we adopt a log-uniform prior on a "scale" parameter and multiply this with standard Gaussian priors on constants A, B, F, and G. This maintains a log-uniform prior on the semimajor axis and sine prior on the inclination. For both cases, we replace parameter τ, which gives the position of a planet along its orbit (Blunt et al. 2020), with θ, the position angle at the average epoch of the observations. This improves convergence relative to sampling from the τ parameter directly as θ is directly constrained by relative astrometry and is not sensitive to other orbital parameters. A derivation of τ from position angle θ is presented in Appendix B.

3.3. Modeling Language

To specify the structure of a system model, Octofitter provides a domain-specific modeling language. This simple language allows for the parameterizations and observations associated with each planet and the host star to be independently specified. For example, we could attach separate RV measurements to each object in a system. Orbital parameters can be drawn from arbitrary prior distributions, fixed to particular values, or computed from combinations of other system parameters.

[16] https://juliastats.org/Distributions.jl

This modeling language makes it convenient to work with simple systems, like fitting the orbit of one planet to relative astrometry measurements, as well as more complex multiplanet systems.

The following is an example of a planet model using traditional Campbell orbital parameters, user-specified priors, and relative astrometry data:

```
table = Table(CSV.read("astrom.csv"))
astrom = AstrometryLikelihood(table)
@planet b VisualKepOrbit begin
a ~ LogUniform(2.5, 25)
i ~ Sine()
e ~ truncated(
Normal(0.2, 0.2),
lower = 0,
upper = 1
)
 Ω~UniformCircular()
 ω~ UniformCircular()
 τ~UniformCircular(1.0)
end astrom
```

A similar syntax is used to specify stellar properties:

```
@system HD1234 begin
plx ~ truncated(
 Normal(41.123, 0.012), lower = 0)
M ~ truncated(
 Normal(1.5, 0.05), lower = 0)
age_Myr ~ Uniform(30, 300)
end b
```

The planet and system definition blocks contain pairs of variable names and values, which can be constants, prior distributions, or arbitrary functions of other variables. Variables drawn from priors are specified by ~, whereas variables defined as constants or calculated as a function of other variables are defined with =. In this example, the convenience function `UniformCircular` creates two independent variables with standard normal priors and computes the angle between them using the arctangent. This, in effect, creates a circular domain where samples can smoothly wrap past 0 and 2π. Users are free to create their own parameterizations and likelihoods and combine them arbitrarily. The only requirements are that they are differentiable and smooth, and return a finite value at all points in the domain given by the priors.

This modeling approach, by being declarative rather than imperative, as in `exoplanet.py` (Foreman-Mackey et al. 2021), allows us to transform and evaluate the model in several ways. One key restriction is that each prior is proper, meaning it is a true probability distribution that integrates to unity. This is in contrast to some of the defaults used by `orvara`, which adopts fully scale-independent, but improper (in the statistical sense of the word) log-uniform priors. Octofitter requires proper priors to support tasks that require sampling directly from the prior distributions, such as simulation-based calibration, as will be discussed in Section 3.7.

3.4. Numerical Methods

From a model definition, Octofitter can generate efficient machine code using run-time-generated functions and Julia's just-in-time compiler. This code generation step allows Octofitter to support a rich variety of observation types without paying any run-time overhead for features that are not used.

For the purposes of sampling from the posterior, Octofitter begins by remapping all variables from their possibly limited support (for example, eccentricity constrained between 0 and 1) into new variables defined across the entire real line. This makes it so that by construction, invalid parameter values like negative masses or semimajor axes are not possible to construct and will not be hit by a sampler. This transformation is performed by the Julia package `Bijectors.jl`.[17]

Next, a log-prior function is created for the model that simply evaluates the log-probability density of each prior distribution given a set of parameters. To preserve the user-specified prior distributions in place of the automatic bijections, a correction is applied based on the Jacobian of the transformation.

Similarly, a log-likelihood function is created based on the provided model and observations. Various constants are precalculated and reused between orbit solutions.

To enable the use of higher-order samplers, including Hamiltonian Monte Carlo, forward-mode automatic differentiation is used to differentiate through the generated log-prior and log-likelihood functions (Revels et al. 2016). This provides the gradient of the log-posterior density, in addition to the value itself, without the overhead of calculating finite differences. The overhead of calculating both the log-posterior density and its gradient using forward-mode automatic differentiation can be as low as just 2 × compared to 10.2 × for finite differences.

Special care was taken to remove all dynamic memory allocations from the generated log-density and gradient functions to prevent overhead from the Julia garbage collector.

Octofitter implements the Julia `LogDensityProblems` interface so that user models can be sampled from a wide variety of Julia-based Markov chain Monte Carlo (MCMC) sampler packages, including `AdvancedMH`, `AdvancedHMC`, and `MCMCTempering`. This allows users to select the best sampler for their particular problem and to compare results against samplers used by other popular packages.

The No-U-Turn Sampler (NUTS) variant of Hamiltonian Monte Carlo (Hoffman & Gelman 2014) provided by `AdvancedHMC` is the default used by Octofitter. It allows the code to explore complex posterior distributions with many fewer log-posterior density evaluations by simulating physical trajectories across the posterior landscape. The performance of NUTS in Octofitter will be discussed in Section 4.1.

When using the default NUTS sampler, we use a dense mass matrix, a jittered leapfrog integrator, and a multinomial trajectory sampler, and allow the user to specify a maximum tree depth. We initialize the sampler by drawing 250,000 samples from the priors and selecting the value with the highest posterior density as the starting point. We also initialize the diagonal elements of the mass matrix using the interquartile range of the priors. We found that this procedure was more robust than trying to determine the maximum a posteriori value with an optimizer for multimodal posteriors like those found when modeling images. In particular, this procedure is less likely than an optimization pass to get stuck in a local optimum or other pathological position. A downside of this approach is that it may be inefficient if the priors are narrow and far from the posterior. For this case, the user can supply a starting point

[17] https://turinglang.org/Bijectors.jl/

manually as in other tools. Next, we adapt the mass matrix and step size according to `AdvancedHMC`'s implementation of the Stan windowed adaptation strategy. Finally, sampling proceeds until a preset number of accepted proposals are found—typically chosen to be on the order of 1000 to 10,000.

3.5. Kepler Solver

To sample from planet models, one must map from the orbital parameters specified in the system model to simulated observations, such as a planet's position over time in the plane of the sky or the RV of the host star. In all cases, this requires solving Kepler's equation at every observation epoch and parameter draw. Kepler's equation (Equation (A3)) connects eccentricity and mean anomaly, a pseudo-angle somewhat analogous to the amount a planet has moved around its orbit, into an eccentric anomaly, which can be used to find the actual location of the planet in its orbit.

Kepler's equation is transcendental and cannot be solved analytically outside of special cases. The traditional approach to solving the equation is to use an iterative procedure like Newton's method with an initial guess chosen carefully from the mean anomaly and eccentricity. Octofitter's strategy is to wrap many different pluggable Kepler solvers that can be useful in different scenarios and use a fast noniterative method as a default.

A nonexhaustive list of solvers supported by Octofitter are `Markley` (Markley 1995), `Goat` (Philcox et al. 2021), and `Newton`. Newton's method and many other available root-finding algorithms are provided by the `Roots.jl`[18] Julia package.

As these solvers are all implemented in pure Julia, there is no overhead from calling between Python and a C/Cython[19] library and full performance is achieved with or without vectorization. Additionally, the solvers support changing the numerical precision between 16 bit, 32 bit, 64 bit, and arbitrary precision. Currently, no particular effort has been made to exploit hardware SIMD vectorization across epochs.

The `Markley` algorithm, the default choice, converges to nearly machine precision for all bound orbits when used with 64 bit floating point values. The `Newton` method can be combined with arbitrary precision floating point numbers to achieve arbitrarily tight tolerances if needed for niche applications. The default Markley (1995)-based method executes in just 90 ns on 64 bit floating point values for any valid eccentricity and mean anomaly (benchmarked on a Skylake Intel Xeon processor). This is slightly slower than the state-of-the-art results reported by Brandt et al. (2021); however, their results of ≈60 ns are only achieved when the solver is vectorized over many epochs.

Applying automatic differentiation through an iterative Kepler equation solver would lead to poor performance. Even though Kepler's equation has no closed solution, the derivatives of the eccentric anomaly with respect to mean anomaly and eccentricity can be found analytically using implicit differentiation (Equations (A5) and (A6)); that is, if we have already solved Kepler's equation for the eccentric anomaly, we can calculate its gradient inexpensively. We supply these equations as a manual rule to the automatic differentiation library.

3.6. Analysis and Visualization

Once sampling is complete, Octofitter supports the user in testing the convergence of their chain or chains. The sampling results are returned as `Chains` objects from `MCMCChains.jl`. This table-like structure includes entries for each accepted MCMC proposal as well as metadata about the sampling process, such as the computational time used. All variables are returned, including those that were fixed to a constant or calculated deterministically from combinations of other variables. The output is an automatic summary that includes plausible intervals, expectation values, effective sample sizes (ESS), and $\hat{R}$ statistics for each parameter. The user can test the convergence of their chain or chains by assessing those ESS and $\hat{R}$ values, creating a trace plot, and/or calculating figures like the Gelman, Rubin, and Brooks diagnostic (Gelman & Rubin 1992; Brooks & Gelman 1998) using tools provided by `MCMCChains.jl` and `MCMCDiagnosticTools.jl`.

Octofitter can be used to visualize orbit fits in several ways. The orbits represented by chains can be plotted in the plane of the sky, in physical system coordinates (i.e., au), or as a time series (e.g., for Doppler or astrometric velocimetry).

When visualizing an orbital posterior, a common challenge is ensuring that enough data points are used to create a smooth arc. This becomes especially challenging with eccentric orbits or ensembles of orbits with widely varying periods. Tracing out orbits in equal time steps will lead to most points clustering closer to apastron, where the planet is moving the most slowly. The strategy used by Octofitter is to trace orbits in equal steps of eccentric anomaly, as suggested in Berry & Healy (2002). This places points in regions of greater curvature, creating smooth arcs with fewer points.

Octofitter includes functions to generate plots that visualize the posterior and compare it to the input data. This includes astrometry, separation, position angle, RV, and proper motion anomaly. It also includes a function for visualizing orbits in spatial coordinates (units of au) in one, two, or three dimensions. Examples of these plots are shown throughout the text and documented with the online tutorials.

3.7. Simulation-based Calibration

SBC (Cook et al. 2006; Talts et al. 2020) is a technique that allows one to verify that the output of a Bayesian modeling procedure is unbiased. This includes any part of the computation, from model specification to the sampling procedure after the choice of priors. Verifying that this choice of priors is reasonable is the domain of other procedures like prior and posterior predictive tests.

A conceptually related procedure was carried out in Ferrer-Chávez et al. (2021b) in which the authors systematically tested the `orbitize!` package, parameterization, and default priors for biases. Our contribution here is to present a procedure that is well explored in the statistics literature, is specific to a given set of observation epochs and measurement uncertainties, and can be applied in an automated fashion.

The calibration procedure requires a generative model—that is, a way to take a given set of parameters and create a simulated observation. A familiar example from direct imaging is the injection of fake point sources to test image processing algorithms. To follow this procedure, one repeatedly draws a set of parameters from the priors, creates simulated observations based on those parameters, samples from the resulting

[18] https://juliamath.github.io/Roots.jl/
[19] https://cython.org/

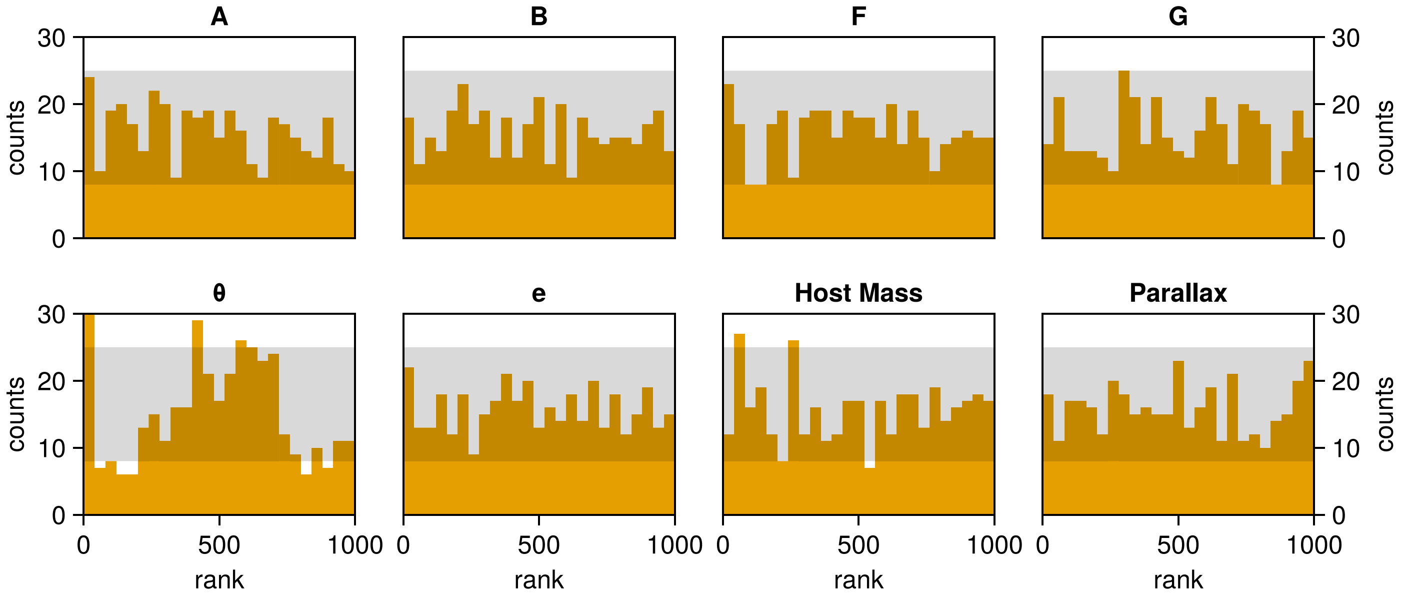

Figure 3. Sample results of running the simulation-based calibration procedure on a model consisting of a single planet parameterized with Thiele-Innes elements (A, B, F, G, etc.) and position angle θ. Each count represents a model fit to a different simulated system. The horizontal band gives a 99% range around the expected value for a perfect sampling procedure. The observation epochs and uncertainties are taken from Table 1.

posterior, and then compares the true parameter values (originally sampled from the priors) with the resulting posterior. By doing this many times, one creates a histogram of rank statistics that can reveal many sources of biases present in the model and sampling process. To be clear, this does not evaluate how the choice of priors impacts the posterior.

We implemented support for performing simulation-based calibration automatically in Octofitter. In this procedure, Octofitter takes a given model's observational data, discards the actual measurements, and keeps the epochs and uncertainties associated with each measurement. Octofitter then repeatedly creates simulated data at each epoch by drawing from the priors and performs SBC on these simulated data. SBC should, in general, be applied to each model to confirm that it is working as expected.

Figure 3 shows the results of the simulation-based calibration procedure applied to a model of a planet parameterized by Thiele-Innes elements and the position angle at the average epoch. For the most part, these histograms do not reveal any systematic biases in the Octofitter sampling procedure. One exception is the "bump" in the position angle histogram. This shape indicates that Octofitter is under-confident and the sampled posterior distribution is wider than the true posterior. By contrast, seemingly small tweaks to the choice of priors can result in histograms with strong biases. For example, drawing the Thiele-Innes constants (A,B) and (F,G) from log-uniform prior distributions rather than Gaussian distributions centered around a single scale, itself drawn from a log-uniform prior, results in noticeable issues with the SBC histograms. A guide to interpreting the results of the SBC procedure is available in Talts et al. (2020).

These tests illustrate the value of performing the simulation-based calibration procedure for each new model and data combination a user wishes to use. Given the complexity of orbit models and the difficulty of sampling from them, we do not expect our sampling procedure to be entirely unbiased. Rather, we hope that by creating an easy way to diagnose these biases, users of Octofitter can interpret their results with an appropriate level of skepticism in accordance with the level of bias detected.

3.8. Other Packages

Fitting observations of planets and binary stars to orbits has been widely addressed in the literature, dating back to Kepler's seminal work. More recently, a variety of software packages have been released following both frequentist and Bayesian approaches. Some of these packages include EXOFAST (Eastman et al. 2019), rvfit (Iglesias-Marzoa et al. 2015), radvel (Fulton et al. 2018), and the Python `exoplanet` package (Foreman-Mackey et al. 2021). Most relevant to image- and visibility-based modeling are packages for fitting relative astrometry, RV, and proper motion anomaly, like `orbitize!` (Blunt et al. 2020), `orvara` (Brandt et al. 2021), and `Efit` (presented in Meyer et al. 2012), as well as image searching codes like K-Stacker (Nowak et al. 2018) and PACOME (J. Dallant et al. 2023, submitted).

These tools have employed a variety of methods for approximating posteriors, including affine invariant (Foreman-Mackey et al. 2013) and parallel tempered affine invariant MCMC (Vousden et al. 2016) in `orvara` and `orbitize!`, Hamiltonian Monte Carlo (Hoffman & Gelman 2014) in `exoplanet`, and nested sampling (Skilling 2004) in `Efit`. With the exception of `exoplanet`, the majority of these codes have used first-order samplers. That is to say, they rely only on evaluating the posterior density and do not calculate or make use of gradient information.

4. Results

4.1. Demonstration with Relative Astrometry

We begin our demonstration of Octofitter by fitting orbits to the simulated relative astrometry measurements, which is one of the simplest use cases for the package. We considered a simulated orbit of a single planet and generated astrometry at 27 epochs (Table 1). For the sake of comparison, we performed

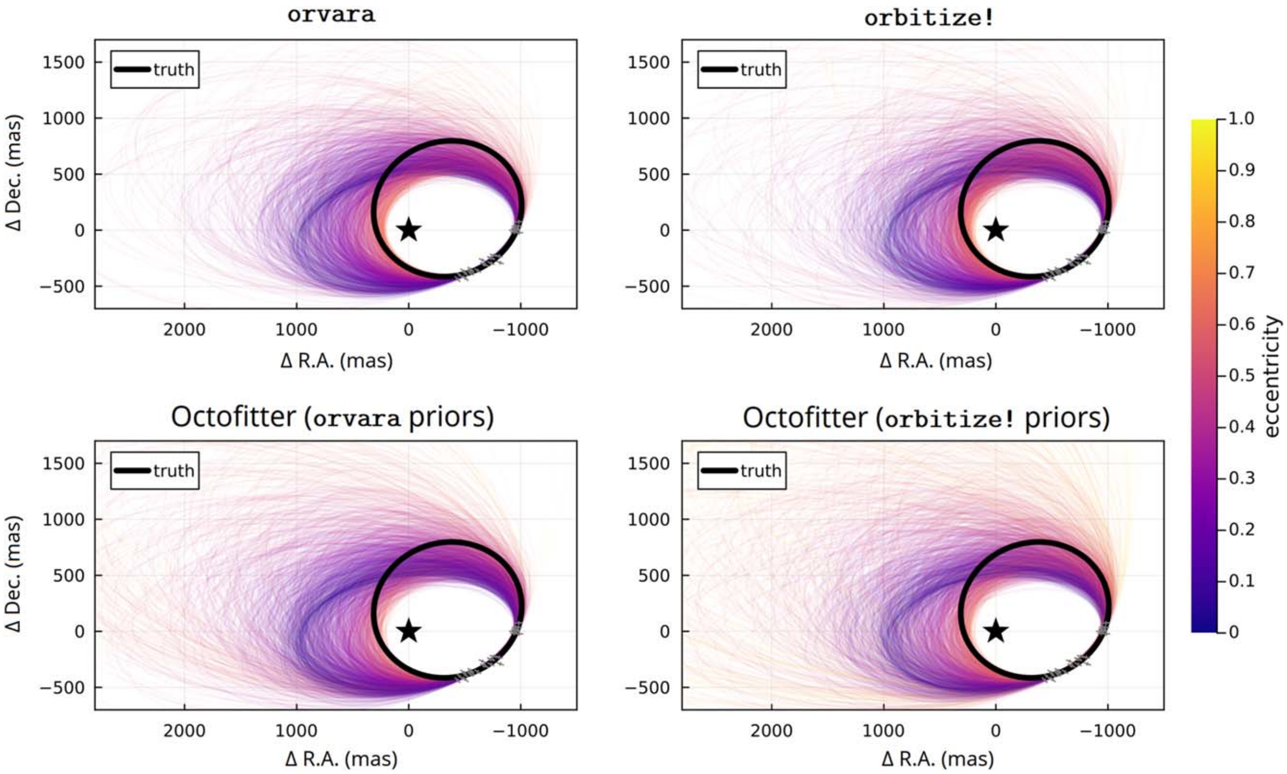

Figure 4. Orbit fitting posteriors visualized in the plane of the sky, compared between three packages: `orvara`, `orbitize!`, and Octofitter.

the same fit using two other popular orbit fitting packages: `orvara` and `orbitize!`.

We followed the best practices laid out in the tutorials provided with each package. `orvara` and `orbitize!` use slightly different priors by default and cannot be made to match each other exactly without code modifications. `Orvara` uses an $\sqrt{e}\,\sin(\omega)$ and $\sqrt{e}\,\cos(\omega)$ parameterization, while `orbitize!` uses separate uniform priors on both e and ω. Therefore, we ran our comparisons twice, first adopting priors similar to `orvara` and then priors similar to `orbitize!` In all cases, we used a log-uniform prior on the semimajor axis between 0.1 and 1000 au. We ran the `orvara` and `orbitize!` packages with settings recommended by their authors. These were four temperatures with 100 walkers for `orvara` and 20 temperatures with 1000 walkers for `orbitize!`. We ran Octofitter with 16 independent chains. We initialized the walkers used by `orvara` and `orbitize!` in a Gaussian ball around the true orbit. For `orbitize!`, we improved the convergence by randomly initializing half the walkers on the second mode of the ω marginal posterior. For Octofitter, on the other hand, we initialized it automatically using our procedure of drawing from the priors and selecting the parameters with the highest posterior density.

We drew 20,000,000 samples from each walker using `orvara` and 100,000 from each walker using `orbitize!`. For Octofitter, we adapted the step size and mass matrix for 5000 iterations and then drew a further 15,000 samples. The resulting posteriors are compared in Figure 4. To remove the burn-in phase, we discarded the first halves of the `orvara` and `orbitize!` chains and the first 5000 samples of the Octofitter chains (the adaptation phase).

To measure how long each package takes to converge to a steady distribution, we followed a similar procedure to Ferrer-Chávez et al. (2021a). We divided each chain into 50 segments and assumed that in the final segment, the chain is fully converged. We then calculated the $\hat{R}$ statistic between each segment and the final segment. We used the rank normalized and median folded version of the statistic as implemented in `MCMCDiagnosticTools.jl`[20] to evaluate how well the samplers converged in the bulk and in the tails of the distribution. We considered it converged once $\hat{R}$ became less than 1 ± 0.005 for all variables. We evaluated this on all walkers (`orvara` and `orbitize!`) and all chains (Octofitter) and took the median.

The most important result of this code comparison is that in the limit of large numbers of samples, all packages produce posterior distributions that largely agree with each other and that include the true orbit. The orbit paths in the plane of the sky are compared in Figure 4, and the parameters are compared in a corner plot in Figure 5. This should serve to further improve confidence in the results of all packages.

The sole exception is that the orbit posterior from Octofitter contains more samples from a high-eccentricity, high-semimajor axis branch of the posterior than the two other packages (Figure 5). Nonetheless, running SBC on this model with uncertainties and observation epochs from this data set (Figure 3) reveals that Octofitter is acceptably calibrated and is not overestimating the spread of the posterior. It is not computationally feasible to perform the same SBC experiment with the other packages, but this likely indicates that the additional samples from the long tail of the posterior are representative of the true posterior and would eventually appear in the outputs of the other packages.

A second result is that for this problem, Octofitter converges to a steady distribution almost immediately using a single computer core, while other packages take many hours

[20] http://turinglang.org/MCMCDiagnosticTools.jl/

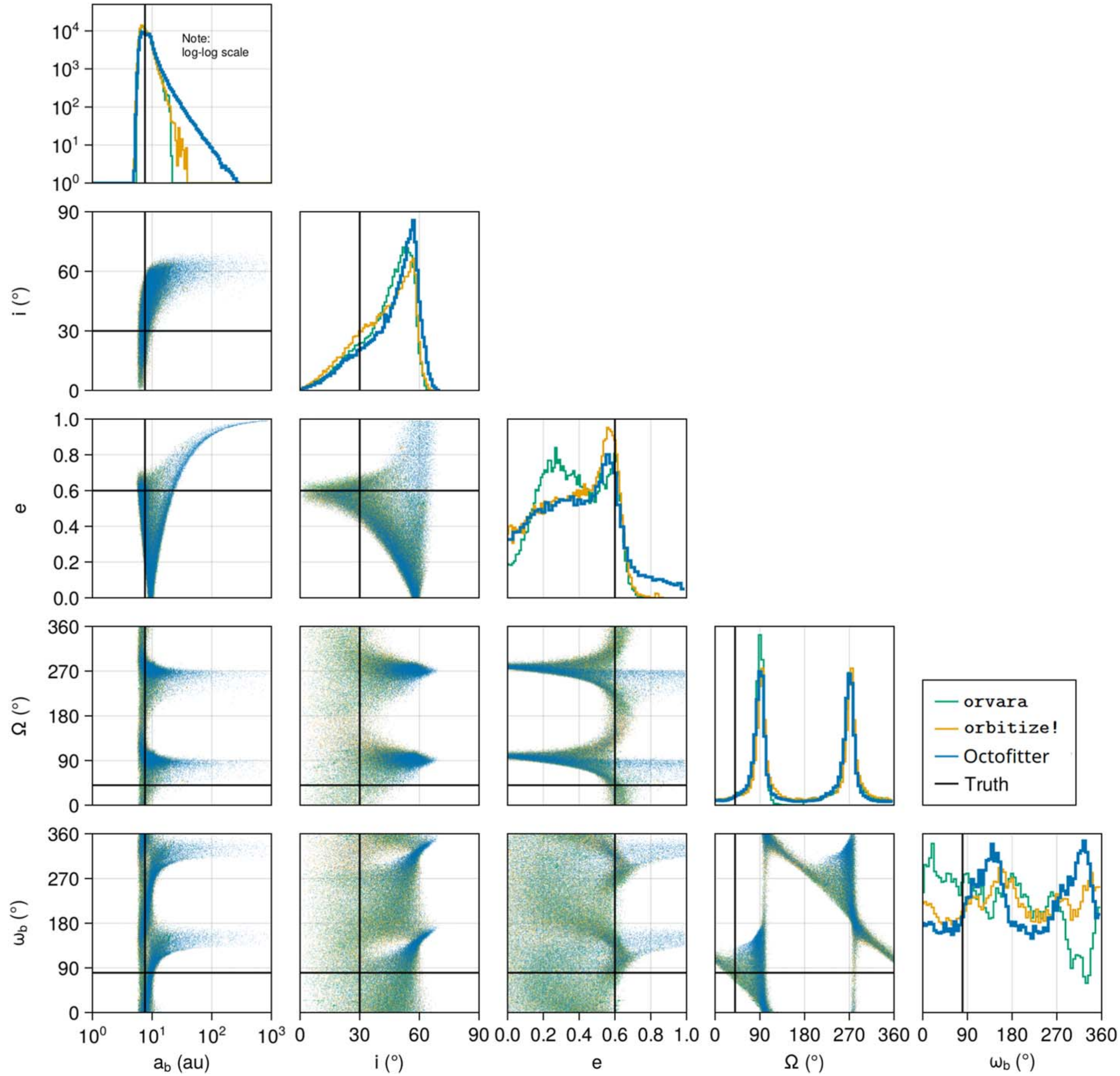

Figure 5. Corner plot of key orbital parameters for the case shown in Figure 4 compared between three packages: orvara, orbitize!, and Octofitter. The semimajor axis is plotted on a log scale to reveal how the sampler behavior differs in the long tail. Though we expect all packages to eventually converge to near-identical results, we find that there are small differences in the ω, a, and e marginal posteriors that persist even after running for multiple days.

to converge (Figure 6). As assessing the convergence of MCMC can be difficult, we suggest that with Octofitter, unsophisticated users are therefore less likely to use insufficiently converged results in their analyses. Of course, in addition to needing the chain to be converged, one must also generate a sufficient number of samples for the problem at hand. Interestingly, despite the much faster convergence, Octofitter is no faster than the other packages at producing independent samples. In most workflows, the bottleneck is ensuring convergence rather than a need to produce tens of thousands of samples, so Octofitter should still offer a decisive improvement in computation time.

The exact results are hardware-specific, sensitive to the data, and depend on the choice of priors and parameterization. It is not feasible to fully account for all small differences between software packages, and it is likely that different approaches will perform better or worse depending on the problem at hand. An additional caveat is that the ptemcee sampler used by orvara and orbitize! scales across many cores, which reduces the total sampling time. Octofitter supports running multiple chains in parallel but is not configured to use multiple cores to accelerate a single chain. Finally, given the convergence guarantees of MCMC, all packages would approach the same distribution given infinite time. This

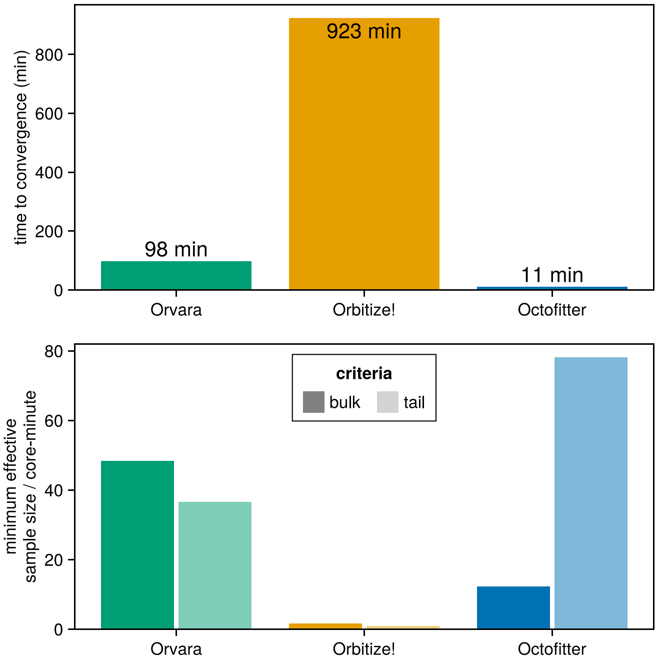

Figure 6. Comparison between packages of the average time until chains converged to a stationary distribution and the rate at which independent posterior samples are generated. The effective sample size (ESS) rate was measured separately using the bulk and tail methods of `MCMCDiagnostics.jl`. Note that the $\hat{R}$ and ESS of the slowest variable for a given sampler are used as this is what ultimately limits sampling performance. These results are based on the astrometry presented in Table 1 and are expected to depend strongly on hardware, input data, and choice of priors.

comparison, therefore, is meant only to illustrate the typical efficiency one might expect with Octofitter on similar problems and reasonable run times.

4.2. Demonstration with Relative Astrometry, RV, and PMA

We now demonstrate Octofitter on a real system with RV, proper motion anomaly, and relative astrometry data.

The HD 91312 system consists of a $1.6M_\odot$ star orbited by a binary companion with a mass of approximately $0.3M_\odot$ and separation of roughly 10 au (Chilcote et al. 2021). The companion was discovered by a direct imaging search targeting accelerating stars (Currie et al. 2021) from the HGCA. We now reproduce the orbital analysis of the discovery paper in order to demonstrate Octofitter's modeling capabilities when applied to a combination of relative astrometry, RV, and proper motion anomaly data.

For this analysis, we used the astrometry data from Chilcote et al. (2021), proper motion anomaly data from the HGCA (Brandt 2021), and limited RV data originally from Borgniet et al. (2019). As the original RV data were not forthcoming, we measured the RVs from the probability density function plots submitted to arXiv alongside the 2021 manuscript. We used similar priors as those described in the discovery paper, namely log-normal priors on primary and companion mass and a uniform prior on the square root of eccentricity. These were chosen to match the analysis of the discovery paper for the purpose of comparing codes and demonstrating Octofitter rather than any physical motivation on our part.

The results of this orbit modeling are presented in Figure 7. Octofitter recovers the orbit of the companion with similar results to those presented by Chilcote et al. (2021). The RV, proper motion anomaly, and relative astrometry are all consistent with the secondary companion having a mass of approximately 300 $M_{\rm jup}$, given the choice of priors described in the original discovery paper.

4.3. Demonstration with Images

We now present a series of simulations showing how this framework allows us to detect planets using multiepoch direct images. We selected 50 sequences from the Gemini Planet Imager Exoplanet Survey (Nielsen et al. 2020), processed using a forward model matched filter (FMMF; Ruffio et al. 2017), which have a stellar I-band magnitude less than or equal to 6 and have no previously detected point sources. To be clear, we do not search these sequences for real companions but merely use them as realistic noise maps for our simulations. We normalized the contrasts of each sequence to the median at 200 mas separation. This retains the true noise distribution but removes the effects of sequence-to-sequence variation on our results. We consider a hypothetical system at 20 pc with a $1M_\odot$ star that is observed once per month for between 1 and 100 months (a little over eight years). Using these sequences and parameters, we generated simulated observations of a planet by injecting a synthetic PSF into the correct positions in each epoch. These input data are presented in Figure 8.

For all models, we adopted a $1 \pm 0.1M_\odot$ prior on M, a uniform 0–$30M_{\rm jup}$ prior on m, a Gaussian prior on parallax, a uniform prior on a, and a uniform circular prior on τ.

We test this model's ability to detect planets in sequences of direct images for the most straightforward case: circular, face-on orbits. We injected the planet into 1, 5, 10, 25, 50, and 100 images with an average S/N ranging between 0 and 5 in each image. Finally, we applied our model to each of these simulated data sets to arrive at a grid of recovered S/N values as a function of number of epochs and S/N per epoch (Figure 9). We find that we are able to recover planet detections with arbitrarily low S/N per epoch despite orbital motion, provided we have a sufficient number of observations.

Figure 9 also shows how the recovered S/N compares with the S/N we would expect for combining images without orbital motion in the presence of uncorrelated Gaussian noise. We find that the model detects the injected planets with near-perfect $\sqrt{N}$ scaling when the final, combined S/N is greater than ~5. For instance, a planet injected into 100 epochs spaced one month apart, at an S/N per epoch of just 0.5 (below the noise), is still robustly detected at a final S/N of 5. We do note that as each image was taken of a different star, this ideal scaling is the best-case scenario. It is possible that repeated observations of the same target could lead to correlated residuals that reduce the final S/N, though the orbital motion of the planet should mitigate this in much the same way as angular differential imaging (Males et al. 2015; Marois et al. 2006).

The left-hand columns of Figure 9 are of particular note. They show that the model fails to detect a point source injected into a single image at an S/N of 4. The failure to detect planets with expected significance below S/N ≈ 5 can be understood by contrasting our detection criteria with the standard used in the field. In these models, we consider the overall S/N marginalized over all locations in the image (or orbits through a sequence), whereas typically, one looks at the maximum S/N at any given location. For example, in any S/N 5 detection of a planet, numerous other S/N 2 and 3 peaks exist. When looking at the S/N marginalized over all locations, these other peaks

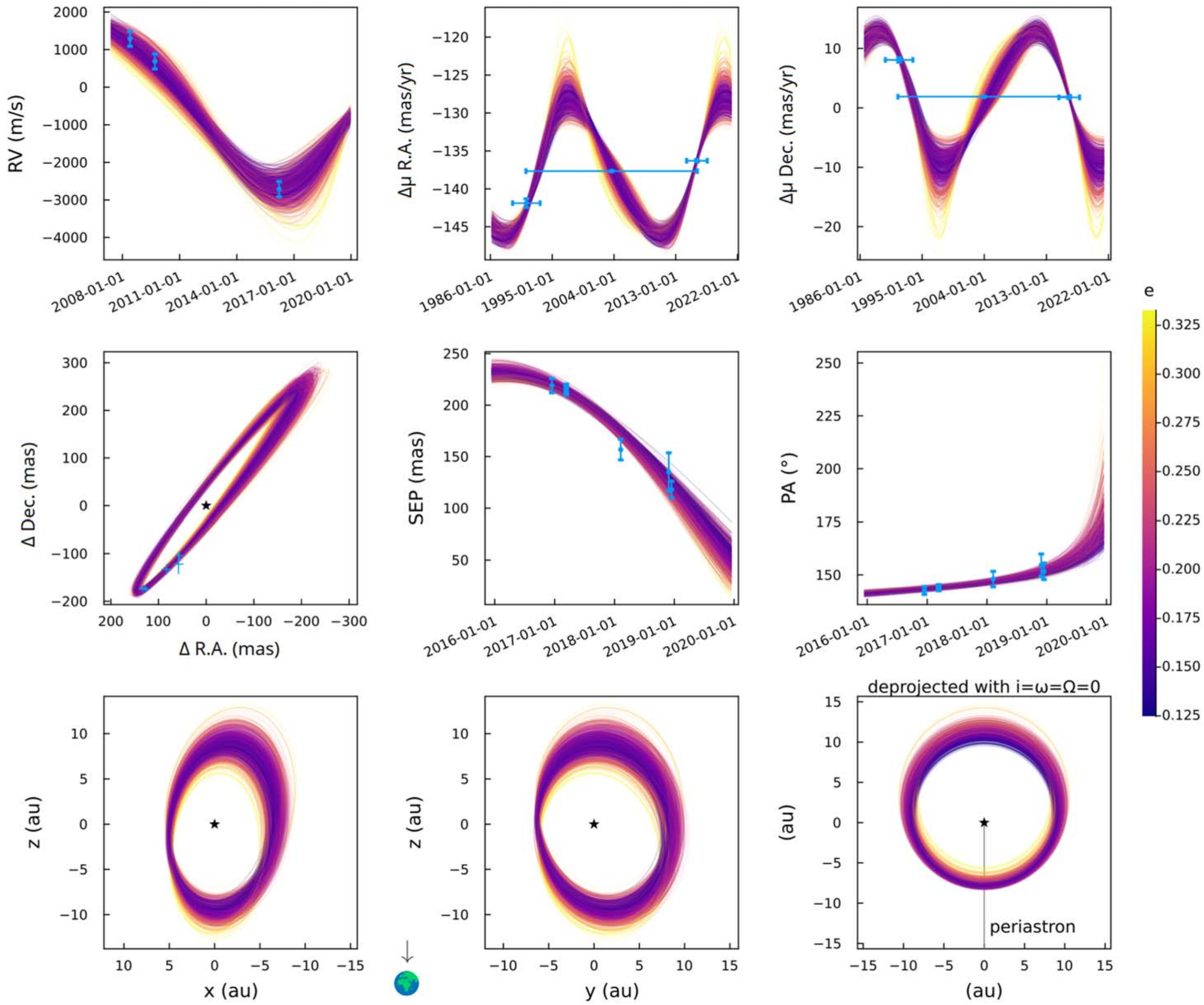

Figure 7. Sample plot output from Octofitter using data from the HD 91312 system. The top row visualizes the orbital posterior compared with velocity measurements of the host. The horizontal bars in the proper motion panels show the time spans over which the average velocity was measured. The middle row shows the posterior compared with astrometry measurements in the plane of the sky and in the separation and position angle over time. The bottom row shows the orbital posterior in physical coordinates to compliment the astrometry plot. The rightmost panel shows a deprojected view of the system where orbits have been rotated face on and to place periastron at the bottom. The conventions used by Octofitter are described in Appendix A.

serve to reduce the final S/N. This makes the S/N calculated from the marginal photometry posterior a more stringent planet detection test.

4.4. Demonstration with Aperture Masking Interferometry and Radial Velocity

We now demonstrate Octofitter using a combination of simulated RV data and AMI visibilities. We considered a plausible scenario where a planet has been detected using the RV, had its orbit characterized, and is then followed up with a series of three observations with JWST/Near InfraRed Imager and Slitless Spectrograph (NIRISS) AMI (Doyon et al. 2012; Sivaramakrishnan et al. 2023). For this experiment, we connected the mass of the planet to its *Ms*-band photometry using Sonora Bobcat models (Marley et al. 2021) for a fixed system age of 10 Myr. The simulated system had a true orbit with a semimajor axis of 2 au, an eccentricity of 0.1, an inclination of 45°, a parallax distance of 100 mas, and a mass of 5 M_{jup}. The star had a stellar absolute *Ms*-band magnitude of 2.6.

We simulated the RV by adding Gaussian noise with a standard deviation of 25 m s^{-1} to 74 epochs of RV data generated from the above system. The data points were spaced by ∼30 days between 2017 and 2023.

The AMI data were generated using Equation (3) for a single companion following the given orbit. For this example, we used only the closure phase and did not include the squared visibilities. We again added Gaussian noise to the calculated closure phases, with a closing triangle-dependent standard deviation taken from the closure phase uncertainties calculated with AMICAL (Soulain et al. 2020) from NIRISS AMI F480M data of (the presumed single stars) HD 101531 and HD 123991 calibrated against each other.

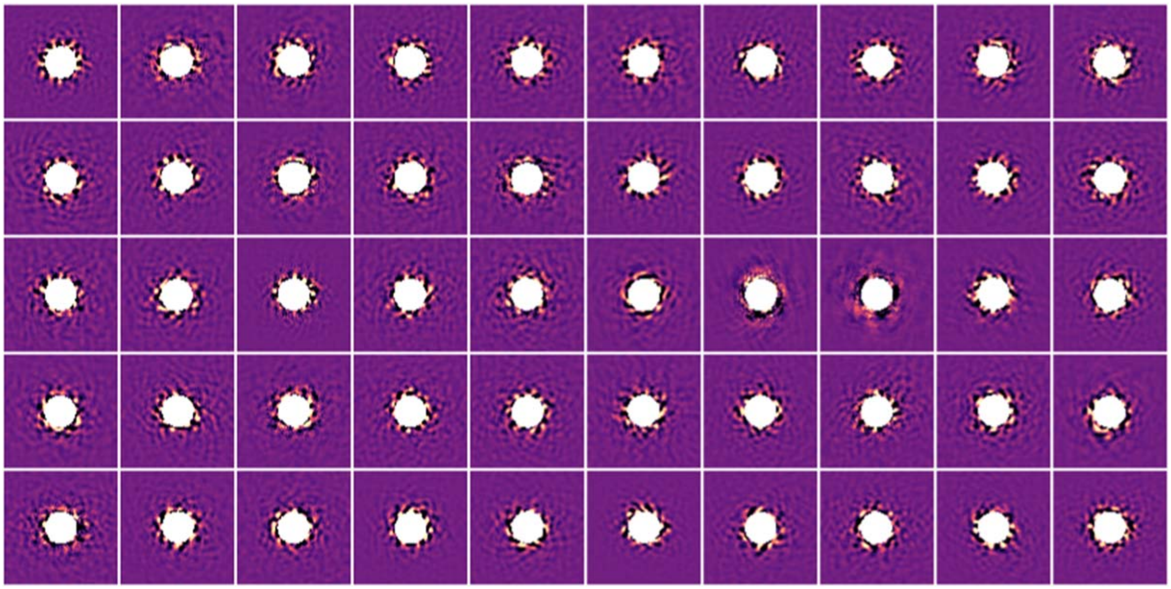

Figure 8. GPI sequences used for simulations in this section. Each image was normalized to have the same average contrast at 200 mas separation from the star (just outside the edge of the mask). The images displayed above contain a simulated planet orbiting CCW at an average of just S/N 1 per epoch, spaced one month apart. Given these data, the model recovers the simulated planet at S/N 7.

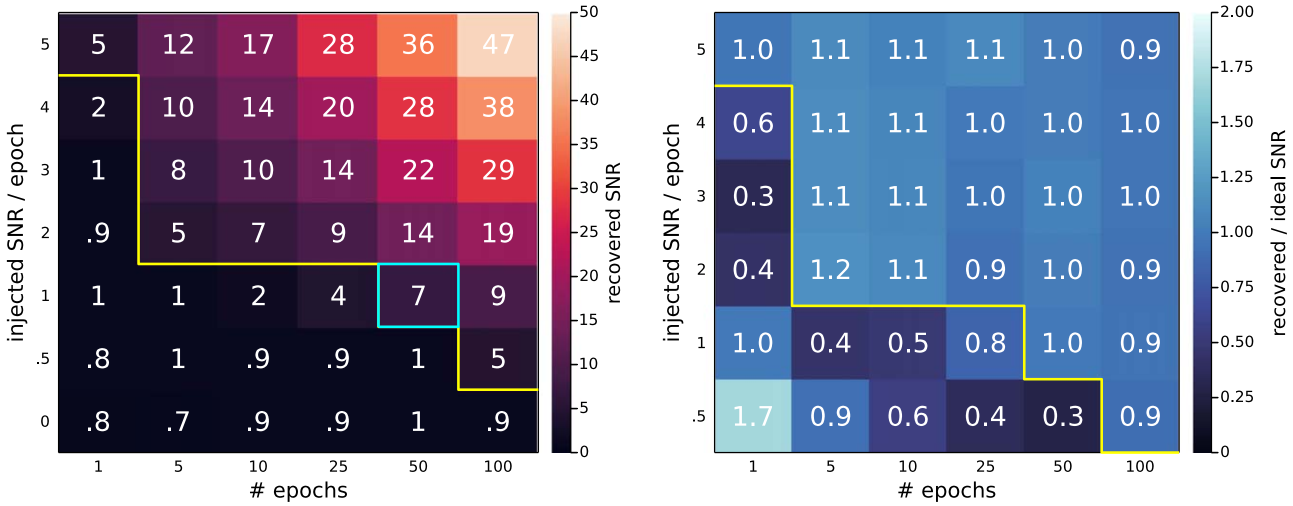

Figure 9. Octofitter's ability to recover planets from simulations of circular, face-on orbits as a function of the S/N per epoch and the number of epochs. Left: Recovered S/N. Cells above the yellow line would be detected with a 5σ threshold. The cell outlined in cyan corresponds to the data shown in Figure 2. Right: The same S/Ns relative to an ideal $\sqrt{N}$ improvement with the number of epochs. We find that the S/N grows as expected unless the final, combined S/N is below ≈5. The recovered S/N falls off quickly below this value and levels off at approximately 1. This explains why the recovered S/N of 1 exceeds the injected value of 0.5 in the bottom left corner.

We applied Octofitter with a joint model of the RV data and three AMI epochs. We also compared this to χ^2 maps from the `Fouriever` framework (presented in Kammerer et al. 2023) of each individual epoch. The results of this experiment are shown in Figure 10.

Using the AMI data, the Octofitter model is able to constrain the inclination and, therefore, the mass of the planet. Due to the faint signal and presence of noise, the `Fouriever` results show spurious signals at each epoch. In comparison, Octofitter is able to connect the three epochs by a single higher-significance mode in orbit-parameter space.

This example illustrates how joint modeling across epochs can be used to increase the significance of AMI and other similar interferometric observations.

5. Conclusion

We have presented a new code, Octofitter, for modeling exoplanet relative astrometry, RV, and proper motion anomaly, as well as performing nontraditional tasks like detecting moving point sources across images and interferometric observables.

1. We demonstrated the simulation-based calibration procedure on a hypothetical orbit fitting task and found that for orbits parameterized with Thiele-Innes constants, Octofitter is acceptably calibrated.
2. We compared the results of Octofitter to the popular packages `orvara` and `orbitize!` and found that all three arrive at similar posterior distributions.

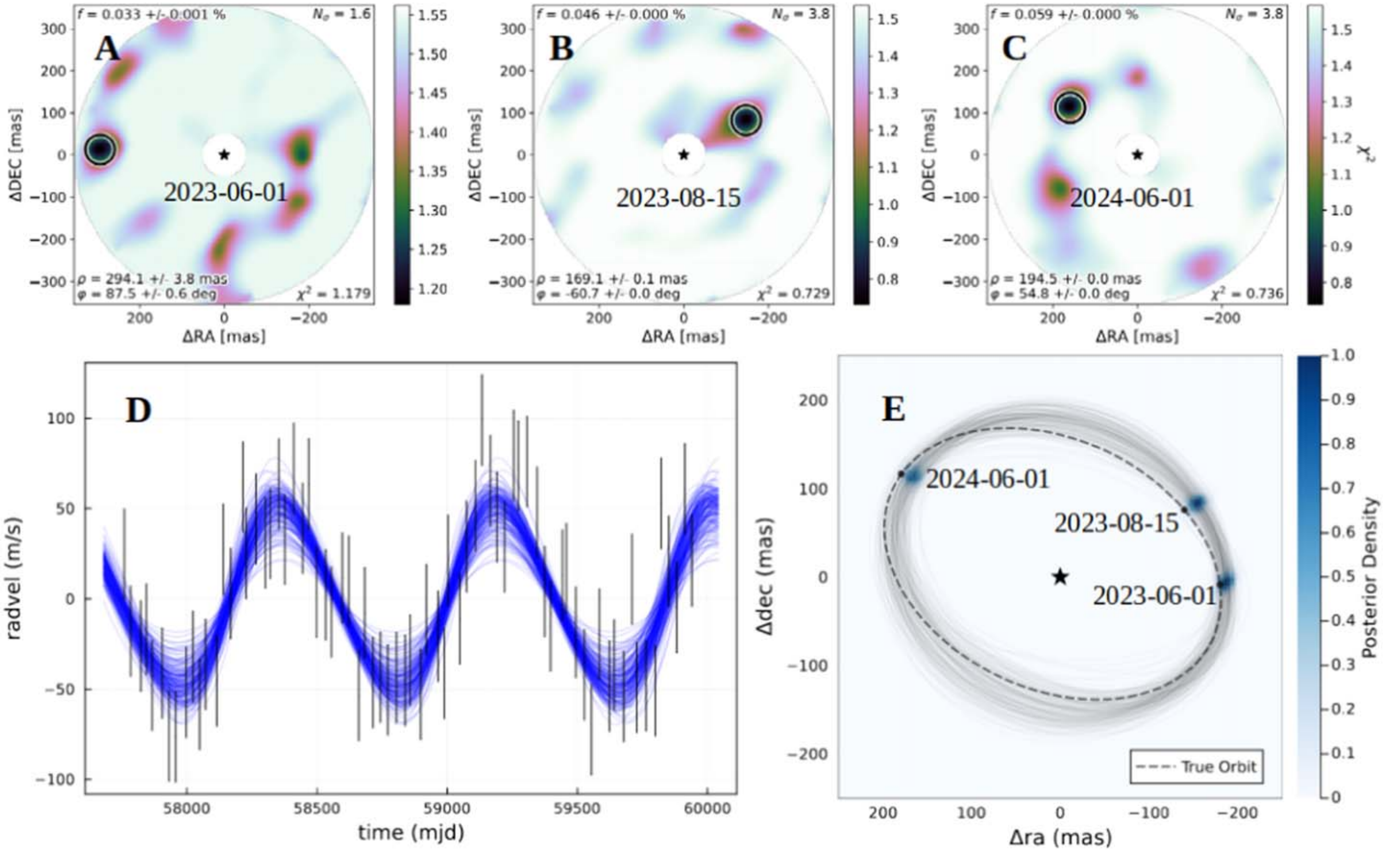

Figure 10. Joint modeling of AMI and RV data. A–C: Recovered χ^2 maps at each independent AMI epoch created using `Fouriever` (presented in Kammerer et al. 2023). D and E: joint RV and AMI posteriors visualized as an RV time series (D) and orbit posterior in the plane of the sky (E).

3. We showed that Octofitter can converge 1 to 2 orders of magnitude faster than other popular packages, making it computationally feasible to perform a variety of statistical checks like simulation-based calibration for individual data sets.
4. We demonstrated a combined fit to relative astrometry, RV, and proper motion anomaly of the HD 91312 system and found a companion mass that agreed with previous results.
5. We demonstrated the ability to detect arbitrarily faint companions despite orbital motion using simulations and data from the Gemini Planet Imager Exoplanet Survey.
6. We demonstrated a combined model of simulated RV and multiepoch JWST/NIRISS aperture masking interferometry.

Octofitter is a powerful new tool that will enable the community to broadly apply joint multiepoch, multi-instrument, and multiband direct imaging, interferometry, Doppler RV, and astrometric motion. This will allow for the detection and characterization of fainter and lower-mass planets more similar to those in our solar system.

Acknowledgments

The authors wish to thank the contributors of the orbitize!, orvara, and exoplanet python packages. Their open source software served as valuable references that greatly aided in the development of this package.

W.T., C.M., D.B., and D.J. acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. Based on observations obtained at the international Gemini Observatory, a program of NSF's NOIRLab, which is managed by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the National Science Foundation. on behalf of the Gemini Observatory partnership: the National Science Foundation (United States), National Research Council (Canada), Agencia Nacional de Investigación y Desarrollo (Chile), Ministerio de Ciencia, Tecnología e Innovación (Argentina), Ministério da Ciência, Tecnologia, Inovações e Comunicações (Brazil), and Korea Astronomy and Space Science Institute (Republic of Korea).

Facilities: Gemini:South, Gaia, Hipparcos.

Software: Julia (Bezanson et al. 2012), Makie (Danisch & Krumbiegel 2021), GR (Heinen et al. 1985–2020).

Appendix A
Derivation of the Cartesian and Celestial Positions, Velocity, and Acceleration

To properly fit relative astrometry, proper motion anomaly, and RV data, expressions for the position, velocity, and acceleration of a secondary companion (such as a planet) about a primary mass (such as a star) in Cartesian and celestial coordinates are required. It is well-known that the solution to the Keplerian two-body problem (for example

Goldstein et al. 2008) is

$$r(t) = \frac{a(1 - e^2)}{1 + e\cos(\nu(t))}, \tag{A1}$$

where r is the radial separation of the two bodies, a is the orbital semimajor axis, e is the orbital eccentricity, and $\nu(t)$ is the true anomaly. The true anomaly describes the angular position of the orbit with respect to periastron, and is the solution to the equation

$$\tan\left(\frac{\nu(t)}{2}\right) = \sqrt{\frac{1 + e}{1 - e}}\tan\left(\frac{E(t)}{2}\right). \tag{A2}$$

The eccentric anomaly $E(t)$ is in turn the solution to

$$M(t) = E(t) - e\sin(E(t)), \tag{A3}$$

with $M(t)$ being the mean anomaly

$$M(t) = \frac{2\pi(t - t_{\rm peri})}{T}. \tag{A4}$$

Here, $t_{\rm peri}$ is the time of periastron passage and T is the orbital period. As Equation (A3) is a transcendental equation, $E(t)$ must be determined numerically. Once calculated, though, the derivatives of the eccentric anomaly with respect to the mean anomaly and eccentricity have analytic expressions:

$$\frac{\partial E}{\partial M} = \frac{1}{1 - e\cos E(t)} \tag{A5}$$

and

$$\frac{\partial E}{\partial e} = \frac{\sin E(t)}{1 - e\cos E(t)}. \tag{A6}$$

These expressions can be determined using implicit differentiation.

A.1. Cartesian Coordinates

Using the well-known relation between polar coordinates (r, ν) and Cartesian coordinates (x, y, z), we see that in the frame of the secondary, its position with respect to the primary is

$$\boldsymbol{r}(t) = \begin{bmatrix} r(t)\cos(\nu(t)) \\ r(t)\sin(\nu(t)) \\ 0 \end{bmatrix} = \frac{a(1 - e^2)}{1 + e\cos(\nu(t))}\begin{bmatrix} \cos(\nu(t)) \\ \sin(\nu(t)) \\ 0 \end{bmatrix}. \tag{A7}$$

However, the frame of the secondary and the frame of an external observer are unlikely to be coincident. Instead, an external observer is likely to be rotated with respect to the frame of the secondary. This rotation is quantified by ω (the argument of periastron of the secondary), i (the inclination of the orbit), and Ω (the longitude of the orbit's ascending node). The rotations associated with these angles are given by the rotation operators

$$R_z(\omega) = \begin{bmatrix} \cos(\omega) & -\sin(\omega) & 0 \\ \sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$R_x(i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & -\sin(i) \\ 0 & \sin(i) & \cos(i) \end{bmatrix},$$
$$R_z(\Omega) = \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A8}$$

Thus, the position of the secondary with respect to its primary as seen by an external observer is

$$\begin{aligned}\boldsymbol{r}_{\rm obs}(t) &= R_z(\Omega)R_x(i)R_z(\omega)\boldsymbol{r}(t) \\ &= r(t)\begin{bmatrix} \cos(\nu(t) + \omega)\cos(\Omega) - \sin(\nu(t) + \omega)\cos(i)\sin(\Omega) \\ \cos(\nu(t) + \omega)\sin(\Omega) + \sin(\nu(t) + \omega)\cos(i)\cos(\Omega) \\ \sin(\nu(t) + \omega)\sin(i) \end{bmatrix}.\end{aligned} \tag{A9}$$

Explicitly, we have

$$\begin{aligned}\boldsymbol{r}_{\rm obs}(t) &= \frac{a(1 - e^2)}{1 + e\cos(\nu(t))} \\ &\times \begin{bmatrix} \cos(\nu(t) + \omega)\cos(\Omega) - \sin(\nu(t) + \omega)\cos(i)\sin(\Omega) \\ \cos(\nu(t) + \omega)\sin(\Omega) + \sin(\nu(t) + \omega)\cos(i)\cos(\Omega) \\ \sin(\nu(t) + \omega)\sin(i) \end{bmatrix}\end{aligned} \tag{A10}$$

Inspecting this equation, we see that $\boldsymbol{r}_{\rm obs}(t)$ is dependent on a, e, i, ω, and Ω. In addition, there is dependence on $t_{\rm peri}$ and T because of Equations (A2) and (A3). However, a and T are related by Kepler's third law, meaning dependence on T can be reformulated as dependence on a. Thus, a Keplerian orbit is uniquely described by the parameters $(a, e, i, \omega, \Omega, t_{\rm peri})$; these are referred to as the Campbell elements. Note that in Octofitter, we replace $t_{\rm peri}$ with either τ or θ. τ is given by

$$\tau = \left(\frac{t_{\rm peri} - t_{\rm ref}}{T}\right) \bmod 1, \tag{A11}$$

where we choose $t_{\rm ref} = 58849.0$ MJD (2020 January 1; Blunt et al. 2020), and θ is discussed in Appendix B. We now differentiate $\boldsymbol{r}_{\rm obs}(t)$ to obtain $\boldsymbol{v}_{\rm obs}(t)$. Kepler's second law implies that

$$\frac{r(t)^2\dot{\nu}(t)}{2} = \frac{\pi a^2\sqrt{1 - e^2}}{T}. \tag{A12}$$

From this, we obtain

$$\dot{r}(t) = \frac{2\pi a}{T}\frac{e\sin(\nu(t))}{\sqrt{1 - e^2}}, \quad r(t)\dot{\nu}(t) = \frac{2\pi a}{T}\frac{1 + e\cos(\nu(t))}{\sqrt{1 - e^2}}. \tag{A13}$$

Using these identities to differentiate $\boldsymbol{r}_{\rm obs}(t)$ produces

$$\boldsymbol{v}_{\rm obs}(t) = \frac{2\pi a}{T}\frac{1}{\sqrt{1-e^2}} \times \begin{bmatrix} -\cos(i)\sin(\Omega)[\cos(\nu(t)+\omega)+e\cos(\omega)) + \cos(\Omega)(\sin(\nu(t)+\omega)+e\sin(\omega)] \\ \cos(i)\cos(\Omega)[\cos(\nu(t)+\omega)+e\cos(\omega)) - \sin(\Omega)(\sin(\nu(t)+\omega)+e\sin(\omega)] \\ \sin(i)[\cos(\nu(t)+\omega)+e\cos(\omega)] \end{bmatrix}. \tag{A14}$$

We determine $\boldsymbol{a}_{\rm obs}(t)$ in an identical manner. Observing that

$$\dot{\nu}(t) = \frac{2\pi}{T}\frac{(1+e\cos(\nu(t)))^2}{(1-e^2)^{3/2}}, \tag{A15}$$

differentiating $\boldsymbol{v}_{\rm obs}(t)$ yields

$$\boldsymbol{a}_{\rm obs}(t) = \frac{4\pi^2 a}{T^2}\frac{1}{(1-e^2)^2} \times \begin{bmatrix} (1+e\cos(\nu(t)))^2\,[\cos(i)\sin(\Omega)\sin(\nu(t)+\omega) - \cos(\Omega)\cos(\nu(t)+\omega)] \\ -(1+e\cos(\nu(t)))^2\,[\cos(i)\cos(\Omega)\sin(\nu(t)+\omega) + \sin(\Omega)\cos(\nu(t)+\omega)] \\ -(1+e\cos(\nu(t)))^2\,\sin(i)\sin(\nu(t)+\omega) \end{bmatrix}. \tag{A16}$$

Having expressions for $\boldsymbol{r}_{\rm obs}(t)$, $\boldsymbol{v}_{\rm obs}(t)$, and $\boldsymbol{a}_{\rm obs}(t)$ is quite useful, as they allow us to determine the corresponding quantities in celestial coordinates with ease. Furthermore, the RV of the secondary with respect to its primary can be read off directly as the z-component of Equation (A14), giving us

$$v_{z,\rm obs} = \frac{2\pi a}{T}\frac{\sin(i)[\cos(\nu(t)+\omega)+e\cos(\omega)]}{\sqrt{1-e^2}}. \tag{A17}$$

To obtain the primary's RV with respect to the secondary, we simply employ conservation of momentum and multiply the previous expression by $-M_{\rm s}/(M_{\rm s}+M_{\rm p})$, with $M_{\rm p}$ and $M_{\rm s}$ being the masses of the primary and secondary, respectively. To determine the RVs measured by an external observer, we add the barycentre RV γ. Additional linear trends, quadratic trends, or other trends can be accounted for in Octofitter by making the instrument zero-points a function of other parameters in the fit. Overall, we get that the observed RVs of the primary and secondary are

$$v_{\rm r,s} = \frac{2\pi a}{T}\frac{\sin(i)[\cos(\nu(t)+\omega)+e\cos(\omega)]}{\sqrt{1-e^2}} + \gamma + \xi, \tag{A18}$$

$$v_{\rm r,p} = -\frac{M_{\rm s}}{M_{\rm s}+M_{\rm p}}\frac{2\pi a}{T}\frac{\sin(i)[\cos(\nu(t)+\omega)+e\cos(\omega)]}{\sqrt{1-e^2}} + \gamma + \xi, \tag{A19}$$

where ξ is the combined effect of all relevant trends. Using these expressions, orbital and physical parameters can be fit to RV data.

A.2. Celestial Coordinates

Now that we know $\boldsymbol{r}_{\rm obs}(t)$, $\boldsymbol{v}_{\rm obs}(t)$, and $\boldsymbol{a}_{\rm obs}(t)$, we can calculate the corresponding expressions in celestial coordinates. Before doing so, a brief discussion of coordinate conventions is required. For this paper, we measure with reference to the celestial north pole. This means that, in reference to the previous section, the positive x-axis points in the direction of positive decl. (upwards on the sky), while the positive y-axis points in the direction of positive R.A. (leftwards on the sky). This choice of coordinates is important to note, as it differs from the right/up orientation of the x/y axes commonly seen elsewhere. Furthermore, the positive z-axis points away from the observer.

Keeping this coordinate convention in mind, we can determine position, velocity, and acceleration in celestial coordinates. Normally, an observer a distance d away from an object of size r measures the object to have an angular size of

$$\Delta\theta = \arctan\left(\frac{r}{d}\right). \tag{A20}$$

However, for scenarios relevant to this work, r is on the order of astronomical units and d is on the order of parsecs, meaning the small angle approximation $\Delta\theta \approx r/d$ can be used with negligible error. Applying this approximation, we get that the R.A. and decl. offsets of the secondary from its primary are

$$\Delta\mathrm{R.A.} = \frac{y_{\rm obs}}{d}, \quad \Delta\mathrm{decl.} = \frac{x_{\rm obs}}{d}, \tag{A21}$$

where d is the distance to the system, and $x_{\rm obs}$ and $y_{\rm obs}$ are the x- and y-components of Equation (A10). From this, it immediately follows that

$$\Delta\dot{\mathrm{R.A.}} = \frac{v_{y,\rm obs}}{d}, \quad \Delta\dot{\mathrm{decl.}} = \frac{v_{x,\rm obs}}{d}, \tag{A22}$$

where $v_{x,\rm obs}$ and $v_{y,\rm obs}$ are the x- and y-components of Equation (A14), and that

$$\Delta\ddot{\mathrm{R.A.}} = \frac{a_{y,\rm obs}}{d}, \quad \Delta\ddot{\mathrm{decl.}} = \frac{a_{x,\rm obs}}{d}, \tag{A23}$$

where $a_{x,\mathrm{obs}}$ and $a_{y,\mathrm{obs}}$ are the x- and y-components of Equation (A16), respectively. The equations for ΔR.A. and Δdecl. can be used to fit orbital and physical parameters to relative astrometry data, while the equations for $\Delta\dot{\mathrm{R.A.}}$ and $\Delta\dot{\mathrm{decl.}}$ can be used to fit orbital and physical parameters to proper motion anomaly data. $\Delta\ddot{\mathrm{R.A.}}$ and $\Delta\ddot{\mathrm{decl.}}$ are used for calculating model derivatives for higher-order samplers and may be applied to angular acceleration data in the future.

Appendix B Derivation of Parameter τ from Position Angle θ

Parameter τ used in `orbitize!` and Octofitter is the fraction of the orbit completed by a planet after periastron, at some reference epoch t_{ref}. This is a useful parameterization as it lies between 0 and 1 for all orbits; however, if the reference epoch is not similar to the epochs of the observations, it can exhibit pathological behavior as the other orbital parameters change. One solution is to adjust the reference epoch for each data set, but a further improvement for many cases is to instead adopt the observed position angle θ at epoch t as an independent variable. θ is an improvement over τ because it is insensitive to changes in the other parameters and has a straightforward interpretation. A derivation of τ from θ is given here.

We begin with the parallax distance to the system ϖ, the host mass M, the eccentricity e, and the Thiele-Innes constants A, B, F, and G given in Equations (C1)–(C4). From these, we write the transformation matrix

$$T = \begin{bmatrix} A & B \\ F & G \end{bmatrix}, \tag{B1}$$

from which we find the unit vectors

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = T^{-1} \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}. \tag{B2}$$

We can then calculate the true anomaly ν as

$$\nu = \arctan(\hat{y}, \hat{x}), \tag{B3}$$

and mean anomaly as

$$\mathrm{MA} = \arctan(-\sin(\nu)\sqrt{1-e^2}, -\cos(\nu) - e) + \pi - \frac{e\sqrt{1-e^2}\sin(\nu)}{1 + e\cos(\nu)}. \tag{B4}$$

From this, parameter τ can be calculated using the period as follows. First, we calculate the semimajor axis a using Equation (C7). Next, combining Kepler's third law and the value of a gives us the period $P = 2\pi\sqrt{a^3/(GM)}$. The mean motion is then $n = 2\pi/P$, and the epoch of periastron passage follows from this as $t_{\mathrm{peri}} = t - \mathrm{MA}/n - t_{\mathrm{ref}}$. Finally,

$$\tau = \frac{t_{\mathrm{peri}}}{P}. \tag{B5}$$

The corner plot in Figure 11 demonstrates the benefit of this parameterization. The bottom rows show the same orbits parameterized with τ and with θ. The θ marginal posterior is nearly Gaussian and has simple relationships with all other orbital parameters. By contrast, the τ marginal posterior is less informative and has complex relationships to ω and Ω.

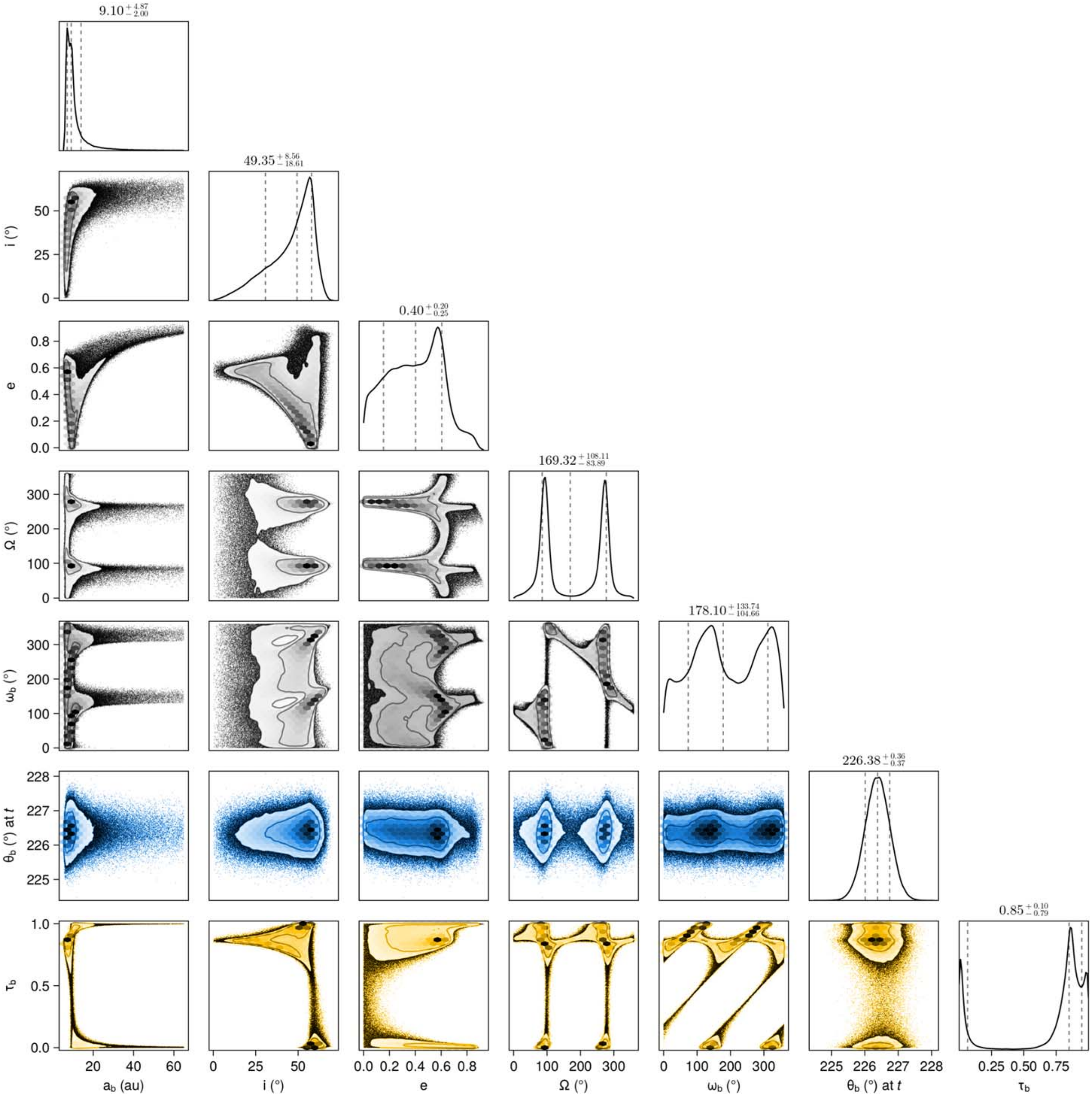

Figure 11. Comparison of θ (blue, second last row) and τ (yellow, last row) parameterizations of the same orbital posterior. The θ marginal posterior is nearly Gaussian and has simple relationships with all other orbital parameters. By contrast, the τ marginal posterior is less informative and has complex relationships to ω and Ω. Complex structures in a posterior decrease sampling efficiency and could in some cases lead to biased results.

Appendix C
Thiele-Innes Elements

In Appendix A.1, we saw that Keplerian orbits can be uniquely characterized by the Campbell elements (a, e, i, ω, Ω), in addition to an orbit position parameter (t_{peri}, τ, or θ). However, Keplerian orbits can be equivalently characterized by the Thiele-Innes elements (e, A, B, F, G), where e is the orbital eccentricity and A, B, F, and G are the Thiele-Innes constants, along with an orbit position parameter. The Thiele-Innes constants are defined as

$$A = a\varpi(\cos(\Omega)\cos(\omega) - \sin(\Omega)\sin(\omega)\cos(i)), \tag{C1}$$

$$B = a\varpi(\sin(\Omega)\cos(\omega) + \cos(\Omega)\sin(\omega)\cos(i)), \tag{C2}$$

$$F = a\varpi(-\cos(\Omega)\sin(\omega) - \sin(\Omega)\cos(\omega)\cos(i)), \tag{C3}$$

$$G = a\varpi(-\sin(\Omega)\sin(\omega) + \cos(\Omega)\cos(\omega)\cos(i)), \tag{C4}$$

where ϖ is the parallax distance to the system. As we can see, it is straightforward to calculate A, B, F, and G from a, i, ω, and

Ω. Inverting this relationship, although less straightforward, is still doable. We begin by defining

$$u = \frac{1}{2}(A^2 + B^2 + C^2 + D^2), \tag{C5}$$

$$v = AG - BF. \tag{C6}$$

We can recover a using u and v like so:

$$a = \frac{\sqrt{u + \sqrt{(u+v)(u-v)}}}{\varpi}. \tag{C7}$$

Once we have a, i immediately follows from

$$i = \arccos\left(\frac{v}{a^2\varpi^2}\right). \tag{C8}$$

As Equations (C1)–(C4) involve only $\cos(i)$, and $\cos(\theta) = \cos(-\theta)$, this is technically an equation for $|i|$. Next, we define

$$j = \arctan(A + G, B - F), \tag{C9}$$

$$k = \arctan(A - G, B + F), \tag{C10}$$

where $\arctan(y, x)$ is a quadrant-sensitive version of $\arctan(y/x)$ (sometimes written as $\arctan 2(y, x)$ or $\mathrm{atan2}(y, x)$). From these quantities, ω and Ω are given by

$$\omega = \frac{1}{2}(j - k), \tag{C11}$$

$$\Omega = \frac{1}{2}(j + k). \tag{C12}$$

Thus, we have successfully recovered a, i, ω, and Ω from the Thiele-Innes constants.

ORCID iDs

William Thompson https://orcid.org/0000-0001-5684-4593
Jensen Lawrence https://orcid.org/0009-0007-5371-3548
Dori Blakely https://orcid.org/0000-0001-9582-4261
Christian Marois https://orcid.org/0000-0002-4164-4182
Jason Wang https://orcid.org/0000-0003-0774-6502
Mosé Giordano https://orcid.org/0000-0002-7218-2873
Timothy Brandt https://orcid.org/0000-0003-2630-8073
Doug Johnstone https://orcid.org/0000-0002-6773-459X
Jean-Baptiste Ruffio https://orcid.org/0000-0003-2233-4821
S. Mark Ammons https://orcid.org/0000-0001-5172-7902
Katie A. Crotts https://orcid.org/0000-0003-4909-256X
Clarissa R. Do Ó https://orcid.org/0000-0001-5173-2947
Eileen C. Gonzales https://orcid.org/0000-0003-4636-6676
Malena Rice https://orcid.org/0000-0002-7670-670X

References

Berger, J. P. 2003, in An Introduction to Visibility Modeling, ed. G. Perrin & F. Malbet, Vol. 6 (Les Ulis: EDP Sciences), 23

Berry, M., & Healy, L. 2002, The Generalized Sundman Transformation for Propagation of High-Eccentricity Elliptical Orbits, Technical Report, Naval Research Lab

Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. 2012, arXiv:1209.5145

Blunt, S., Wang, J. J., Angelo, I., et al. 2020, AJ, 159, 89

Borgniet, S., Lagrange, A.-M., Meunier, N., et al. 2019, A&A, 621, A87

Brandt, G. M., Michalik, D., Brandt, T. D., et al. 2021, AJ, 162, 230

Brandt, T. D. 2021, ApJS, 254, 42

Brandt, T. D., Dupuy, T. J., Li, Y., et al. 2021, AJ, 162, 186

Brooks, S. P., & Gelman, A. 1998, J. Comput. Graph. Stat., 7, 434

Chilcote, J., Tobin, T., Currie, T., et al. 2021, AJ, 162, 251

Collaboration, G., Abuter, R., Accardo, M., et al. 2017, A&A, 602, A94

Cook, S. R., Gelman, A., & Rubin, D. B. 2006, J. Comput. Graph. Stat., 15, 675

Currie, T., Brandt, T., Kuzuhara, M., et al. 2021, Proc. SPIE, 11823, 1182304

Danisch, S., & Krumbiegel, J. 2021, JOSS, 6, 3349

Dickey, J. M. 1971, Ann. Math. Stat., 42, 204

Doyon, R., Hutchings, J. B., Beaulieu, M., et al. 2012, Proc. SPIE, 8442, 84422R

Eastman, J. D., Rodriguez, J. E., Agol, E., et al. 2019, arXiv:1907.09480

Éric, T., & Young, J. 2017, JOSAA, 34, 904

Ferrer-Chávez, R., Blunt, S., & Wang, J. J. 2021a, RNAAS, 5, 162

Ferrer-Chávez, R., Wang, J. J., & Blunt, S. 2021b, AJ, 161, 241

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306

Foreman-Mackey, D., Luger, R., Agol, E., et al. 2021, exoplanet: Gradient-based probabilistic inference for exoplanet data & other astronomical time series, 0.5.1, doi:10.5281/zenodo.1998447

Fulton, B. J., Petigura, E. A., Blunt, S., & Sinukoff, E. 2018, PASP, 130, 044504

GAIA-Collaboration, Brown, A. G. A., Vallenari, A., et al. 2021, A&A, 649, A1

Gelman, A., & Rubin, D. B. 1992, StaSc, 7, 457

Goldstein, H., Poole, C. P., & Safko, J. L. 2008, Classical Mechanics (3rd edn.; Boston, MA: Addison Wesley)

Hartkopf, W. I., McAlister, H. A., & Franz, O. G. 1989, AJ, 98, 1014

Heinen, J., et al. 1985–2020, GR Framework, https://gr-framework.org/

Hoffman, M. D., & Gelman, A. 2014, JMLR, 15, 1593

Householder, A., & Weiss, L. 2023, arXiv:2212.06966

Iglesias-Marzoa, R., López-Morales, M., & Jesús Arévalo Morales, M. 2015, PASP, 127, 567

Kammerer, J., Cooper, R. A., Vandal, T., et al. 2023, PASP, 135, 014502

Kasdin, N. J., Bailey, V., Mennesson, B., et al. 2020, Proc. SPIE, 11443, 114431U

Koop, G. 2003, Bayesian Econometrics (Hoboken, NJ: J. Wiley)

Le Coroller, H., Nowak, M., Delorme, P., et al. 2020, A&A, 639, A113

Le Coroller, H., Nowak, M., Wagner, K., et al. 2022, A&A, 667, A142

Llop-Sayson, J., Wang, J. J., Ruffio, J.-B., et al. 2021, AJ, 162, 181

Males, J. R., Belikov, R., & Bendek, E. 2015, Proc. SPIE, 9605, 414

Markley, F. L. 1995, CeMDA, 63, 101

Marley, M., Saumon, D., Morley, C., et al. 2021, Sonora Bobcat: Cloud-Free, Substellar Atmosphere Models, Spectra, Photometry, Evolution, and Chemistry, Zenodo, doi:10.5281/zenodo.5063476

Marois, C., Lafrenière, D., Doyon, R., Macintosh, B., & Nadeau, D. 2006, ApJ, 641, 556

Mawet, D., Hirsch, L., Lee, E. J., et al. 2019, AJ, 157, 33

Meyer, L., Ghez, A. M., Schödel, R., et al. 2012, Sci, 338, 84

Nielsen, E. L., Rosa, R. J. D., Wang, J. J., et al. 2020, AJ, 159, 71

Nowak, M., Coroller, H. L., Arnold, L., et al. 2018, A&A, 615, A144

O'Neil, K. K., Martinez, G. D., Hees, A., et al. 2019, AJ, 158, 4

Philcox, O. H. E., Goodman, J., & Slepian, Z. 2021, MNRAS, 506, 6111

Pogorelyuk, L., Fitzgerald, R., Vlahakis, S., Morgan, R., & Cahoy, K. 2022, ApJ, 937, 66

Revels, J., Lubin, M., & Papamarkou, T. 2016, arXiv:1607.07892

Ruffio, J.-B., Macintosh, B., Wang, J. J., et al. 2017, ApJ, 842, 14

Ruffio, J.-B., Mawet, D., Czekala, I., et al. 2018, AJ, 156, 196

Sivaramakrishnan, A., Tuthill, P., Lloyd, J. P., et al. 2023, PASP, 135, 015003

Skemer, A. J., & Close, L. M. 2011, ApJ, 730, 53

Skilling, J. 2004, in AIP Conf. Proc. 735, Bayesian Inference and Maximum Entropy Methods in Science and Engineering (Melville, NY: AIP), 395

Soulain, A., Sivaramakrishnan, A., Tuthill, P., et al. 2020, Proc. SPIE, 11446, 1144611

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. 2020, arXiv:1804.06788

Thompson, W., Marois, C., Doó, C. R., et al. 2023, AJ, 165, 29

Trifonov, T., Tal-Or, L., Zechmeister, M., et al. 2020, A&A, 636, A74

van Leeuwen, F. 2007, A&A, 474, 653

Vousden, W. D., Farr, W. M., & Mandel, I. 2016, MNRAS, 455, 1919

Wagenmakers, E.-J., Lodewyckx, T., Kuriyal, H., & Grasman, R. 2010, Cognitive Psychology, 60, 158

Wootten, A., & Thompson, A. R. 2009, IEEEP, 97, 1463

Wright, J. T., & Howard, A. W. 2009, ApJS, 182, 205

Xu, K., Ge, H., Tebbutt, W., et al. 2020, Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference (PMLR), 1, https://proceedings.mlr.press/v118/xu20a.html