LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-858524

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

W. Thompson, S. M. Ammons, G. GPIES Team

December 19, 2023

The Astronomical Journal

Draft version May 10, 2023
Typeset using LaTeX twocolumn style in AASTeX631

Octofitter: fast, flexible, and accurate orbit modelling to detect exoplanets

William Thompson [1] Jensen Lawrence [2] Dori Blakely [1, 2] Christian Marois [2] Jason Wang [3] Mosè Giordano [4] Timothy Brandt [5] Jean-Baptiste Ruffio [6] Doug Johnstone [2, 1] and GPI opt in

[1] *Department of Physics and Astronomy, University of Victoria, 3800 Finnerty Rd, Victoria, BC V8P 5C2, Canada*
[2] *National Research Council of Canada Herzberg, 5071 West Saanich Rd, Victoria, BC, V9E 2E7, Canada*
[3] *Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) and Department of Physics and Astronomy, Northwestern University, Evanston, IL 60208, USA*
[4] *University College London, Gower St, London WC1E 6BT, United Kingdom*
[5] *Department of Physics, University of California, Santa Barbara, USA*
[6] *Department of Astronomy, California Institute of Technology, Pasadena, CA 91125, USA*

ABSTRACT

As next generation imaging instruments and interferometers search for planets closer to their stars, they must contend with both 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 radial velocity and proper motion anomaly, even without a confirmed detection in any single epoch. In this paper, we present a fast and differentiable multi-method orbit modelling and planet detection code called Octofitter. 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 broad application of multi-epoch and multi-method exoplanet detection, which could improve how future targeted ground- and space-based surveys are performed.

1. INTRODUCTION

Instruments for directly studying exoplanets are steadily improving in sensitivity. Current facilities are now accessing planets closer than 10 AU from their stars. At these separations, orbital motion can become apparent over a single month. This will be especially true for facilities with high angular resolving power thanks to having larger apertures or by operating at shorter 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 image to image, naive images stacking causes their signals to blur out and fade away. Reaching a sufficient 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 sufficient 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), VLTI-GRAVITY (Collaboration et al. 2017), and even ALMA in the case of accreting protoplanets.

A number of projects have sough to solve this challenge for images alone 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 (Dallant et al, submitted), the search for ε Eri (Mawet et al. 2019; Llop-Sayson et al. 2021), and the search for additional HR 8799 planets 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. Moving the analysis of direct imaging data into the orbital domain enables a further extension: joint models of both images or interferometric observables with indirect exoplanet detection techniques including radial velocity, astrometric motion, and transit (not directly considered in this paper). These have previously been explored in Mawet et al. (2019) and Llop-Sayson et al. (2021). Moving our analysis into the orbital domain places it on equal footing with indirect methods and opens up many possible scenarios. In addition to combining images to search for planets despite orbital motion, it would allow one to freely combine velocimetry with image data. This could be used to constrain orbits using images with tentative detections or non-detections, 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.

These scenarios are possible since 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 mass (m) or $m \sin(i)$ where i is the inclination; and imaging / inteferometry constrains all orbital parameters up to a $\pm$ ambiguity on the longitude of the ascending node (Ω). These connections could allow information from all methods to flow into a single orbit model and ultimately the detection of a new planet.

To apply these ideas broadly, the community will need a tool capable of modelling all different types of exoplanet data. The Orbitize! Blunt et al. (2020) and Orvara (Brandt, T. D. et al. 2021) packages come close: they support Bayesian modelling of relative astrometry, RV, and PMA. We needed a package that goes further to directly model image and inteferometer data with or without independent detections at each epoch. Further barriers are that image and interferometer data can present much more challenging posteriors to traverse, and because any inaccuracies in the orbit posteriors can lead to errors in mass and/or photometry and a spurious detection.

Additionally, it is challenging to compute orbital posteriors accurately even without considering ambiguous image and inteferometery data. The traditional Campbell orbital elements $(a, e, i, \omega, \Omega, t_{\mathrm{peri}})$ posses complex co-dependencies and degeneracies (e.g. when $e = 0$ or $i = 0$). Short orbital arcs do not have sufficient constraining power to independently determine all parameters, so orbit posteriors typically are both complex and very sensitive to their priors (e.g. O'Neil et al. 2019).

These challenges motivate the development of our new orbit- and data- modelling framework, "Octofitter." [1] Named for 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 modelling and experimentation while providing the highest levels of numerical sampling performance. These advances are thanks to our implementation of a pure-Julia (Bezanson et al. 2012), fully differentiable, and non-allocating modelling 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.

As a publicly available code with these advances, Octofitter will enable the wide application of multi-epoch, multi-instrument, and multi-method exoplanet detection and modelling. These approaches have the ability to improve how direct surveys are completed and the science 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. METHODS

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, pluggable components that can be freely combined to solve a wide range of orbit modelling problems. This section describes both the mathematical and computational structure of Octofitter and how it can be used.

Support for each kind of observation in Octofitter is provided by a separate Julia type inheritting from a super type called `AbstractLikelihood`. Each observation type wraps a data table of observations with a preset list of required columns. The observations themselves can then be provided directly in the code, loaded from a local CSV or Arrow file, or from a remote SQL database. For each observation type, a method of the `Octofitter.ln_like` function is provided that computes the likelihood of that data given a specific set of

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

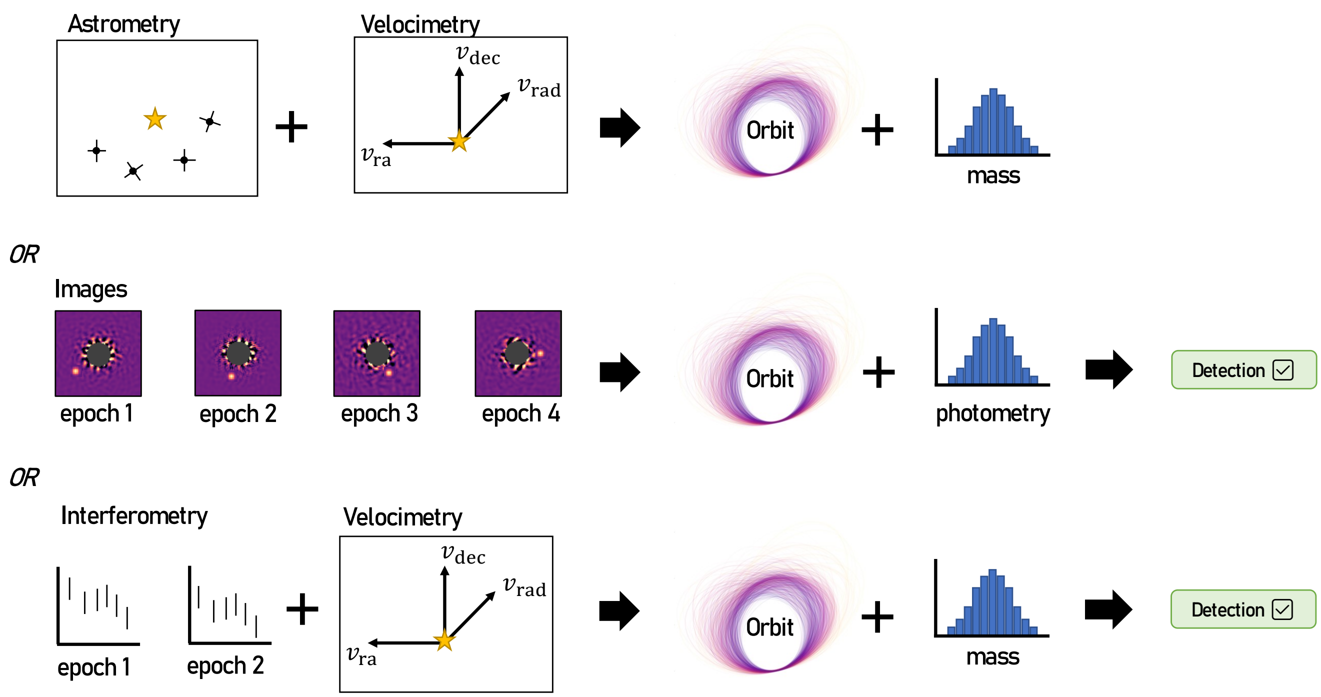

Figure 1. Conceptual schematic of three different ways to use Octofitter. Other possibilities, like combining images with velocimetry or using images without detections to constrain orbits are also possible..

parameters. We now descibe 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 a separation in milliarcseconds and position angle in degrees, or as offsets in milliarcseconds in the RA-DEC tangent plane.

Relative astrometry measurements can be provided to Octofitter using the `AstrometryLikelihood` type 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`, `dec`, `σ_ra`, and `σ_dec` can be substituted if preferred.

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. In the case where the correlation between uncertainties is zero this is

$$\begin{aligned}\log \mathcal{L}_{\rm astrom} = &-\frac{1}{2}\frac{({\rm ra}-\tilde{\rm ra})^2}{\sigma^2_{\tilde{\rm ra}}} - \log\sqrt{2\pi\sigma^2_{\tilde{\rm ra}}} \\ &-\frac{1}{2}\frac{({\rm dec}-\tilde{\rm dec})^2}{\sigma^2_{\tilde{\rm dec}}} - \log\sqrt{2\pi\sigma^2_{\tilde{\rm dec}}}\end{aligned} \tag{1}$$

where the tilde distinguishes measured quantities from values calculated from model parameters.

2.2. *Images*

Directly modelling point sources in images is one of the key features of Octofitter. The approach described in Ruffio et al. (2018) has been used 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 relative astrometry and photometry and then modelling those measurements. Where it goes beyond this 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 non-detection. 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

Table 1. Relative astrometry input sample.

`epoch`	`pa`	`sep`	σ`_sep`	σ`_pa`	`cor`
(mjd)	(°)	(mas)	(mas)	(°)	
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

planet passes too close to a star to detect. In this case, it's not possible to extract astrometry. By directly modelling 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 planets future location.

In Octofitter, image data can be modelled across any number of photometric bands, instruments, and epochs. Data can be provided using the `ImageLikelihood` observation type which accepts a data table with the columns `epoch`, `band`, `platescale`, `image`, and `contrast`. The platescale combined with the parallax system parameter (which may be fixed or fit to the data) allows a given orbit to be solved for position in an image in pixels, where the stellar position is assumed to be the centre pixel. The contrast entry for each image allows one to pass

Table 2. Images input sample

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

an arbitrary function giving 1σ contrast as a function of position. 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 that separation. Extended emission from disks is not currently considered.

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

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

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, 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 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 images and relative astrometry, we do not currently consider uncertainty in the instrument's North-angle or platescale. 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. 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 modelling removes the need to detect a companion in a single observation. We implement a model assuming

Table 3. Visibilities input sample

`epoch`	`band`	`pa`	`sep`	`contrast`
(mjd)	(symbol)	(°)	(mas)	
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

an unresolved point source primary and N unresolved point source companions. The complex visibilities of this model are given by

$$\mathrm{V_{bin}} = \frac{1 + \sum_{i=1}^{N} f_i \exp\Big(-2\pi \mathrm{i} \Big(\Delta \mathrm{RA}_i u + \Delta \mathrm{DEC}_i v \Big) \Big)}{1 + \sum_{i=1}^{N} f_i}, \tag{3}$$

where f_i is the (companion/primary) contrast of the ith companion, $\Delta\mathrm{RA}_i$ and $\Delta\mathrm{DEC}_i$ are the right ascension and declination 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).

The squared visibilities are calculated from the squared modulus of the complex visibilites 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 Thiébaut & 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

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

where CP_i is the ith closure phase, V_i^2 is the ith squared visibility and $\sigma_{CP,i}$, and $\sigma_{\mathrm{V}_i^2}$ 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 radial velocity data in combination with other observation types. This includes radial velocity measurements of the star or of the planet directly. If combined with relative astrometry, proper motion anomaly, or image data, this allows one to directly model the dynamical mass of a planet.

Additionally, when combining radial velocity data with direct imaging modelling, 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 of a planet is well-determined by radial velocity but has not yet been detected with direct imaging.

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

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

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

Octofitter supports multiple instruments, each with their 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 helper functions to load RV curves from two publicly available archives: HARPS RV Bank (Trifonov et al. 2020) and HiRES (Butler et al. 2017). Accessing either of these via will prompt the user to accept their licenses and will 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 and Hipparcos missions. The Hipparcos-Gaia Catalog of Accelerations (HGCA Brandt, T. D. 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 photocentre at the Hipparcos epoch, the Gaia epoch, and between the two.

Table 4. Radial velocity input sample

epoch	rv	σ_rv	inst_idx
(mjd)	(m/s)	(m/s)	
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

In Octofitter, proper motion anomaly must be loaded directly from the HGCA by specifying the host star's GAIA DR3 unique identifier. As with the radial velocity loaders, this will prompt the user to accept a license for the catalog and automatically download the HGCA.

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

$$\begin{aligned}\log \mathcal{L}_{\rm pma} = &- \frac{1}{2}\frac{(v_{\alpha*} - \tilde{v}_{\alpha*})^2}{\sigma^2_{\tilde{v}_{\alpha*}}} - \log 2\pi\sigma^2_{\tilde{v}_{\alpha*}} \\ &- \frac{1}{2}\frac{(v_{\delta} - \tilde{v}_{\delta})^2}{\sigma^2_{\tilde{v}_{\delta}}} - \log 2\pi\sigma^2_{\tilde{v}_{\delta}}\end{aligned} \quad (7)$$

Compared with Brandt, T. D. et al. (2021) and Brandt, G. M. et al. (2021) we adopt a simpler model of the GAIA and Hipparcos missions. Rather than use the exact epochs each mission scanned the star (or estimate using the Gaia Observation Forecast Tool[2]), 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) and Hipparcos (≈ 1227), this approximates the smearing effect caused by the moving photocentre during observations. Future work could refine this model further following their more sophisticated approach.

2.6. *Assessing Detections*

This section describes how we choose to interpret results of a model fit with Octofitter, though it should be noted that the modelling 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), upper limits either in photometry or mass units are the values of the cumulative posterior distribution equal to some cutoff probability. 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 $f_{\rm lim}$. This value will be driven by the brightest speckles close to the star, so a much more useful metric is to provide $f_{\rm lim}$ as a function of orbital parameters.

This model does presuppose that a planet exists and follows a Kelperian orbit around the star; however, as long as the photometry prior is broad, the model is free to drive the photometry towards zero. We can therefore use the photometry posterior to calculate a signal to noise ratio (SNR) analogous to the typical metric used in direct imaging. This has a benefit of familiarity with how direct images are typically evaluated 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 SNR as the mean, divided by the standard deviation of the marginal photometry posterior,
4. and then compare this SNR with a pre-selected threshold based on a tolerable false positive rate, e.g. 5σ.

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

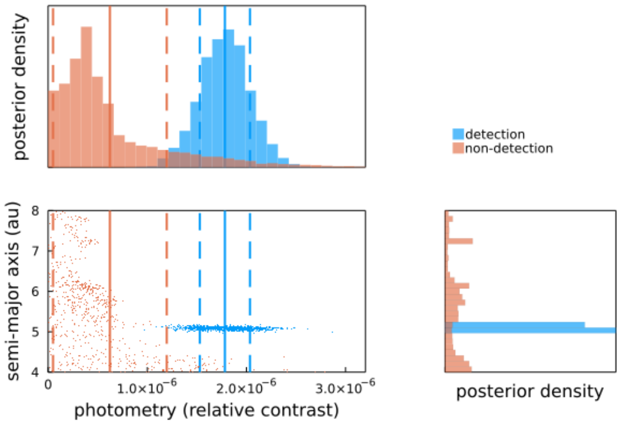

Figure 2. Posterior density of photometry versus semi-major axis for a simulated detection and non-detection. The solid lines marks the mean of the marginal photometry posterior, and the dashed lines mark $\pm 1\sigma$. We consider a planet detected when the SNR based on the photometry marginalized over all other parameters is greater than some chosen threshold, e.g. 5σ.

This gives a single SNR value for a planet matching those priors in that dataset. This is slightly different than the typical SNR detection thresholds used because this SNR 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 SNR, rather than a weighted combination of all positions.

In general, SNRs obtained using this approach assume that

1. the noise is approximately Gaussian distributed,
2. 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 (1-2) may increase the false positive rate, while deviations from (3-4) may lower the recovered photometry and SNR.

Multi-planet 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 contain a combination of 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 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 since it doesn't require a uniform prior—any prior that includes zero mass or photometry would suffice—and because it doesn't presume that the marginal posterior is Gaussian distributed.

2.7. *Orbital Bases and Priors*

Octofitter supports a wide range of different orbital bases for use in different situations. These include traditional Campbell elements $(a, e, i, \omega, \Omega)$, Thiele-Innes elements $(e, A, B, F, G$, Hartkopf et al. 1989; Wright & Howard 2009; O'Neil et al. 2019), Cartesian elements $(z, y, z, v_x, v_y, v_z$, Ferrer-Chávez et al. 2021a), and a reduced basis set for modelling radial velocity only $(a, e, m \sin i, \omega)$. Users can specify priors using arbitrary distributions from `Distributions.jl` 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 semi-major axis, a uniform prior on eccentricity, a sine prior on inclination, a Gaussian prior on host mass, and uniform circular priors on the remaining angular parameters. When using Thiele-Innes elements, we adopt a log uniform prior on a "scale" parameter and multiply this with standard Gaussian priors on the constants A, B, F, and G. This maintains a log-uniform prior on semi-major axis and sine prior on inclination. For both cases, we replace the parameter τ, which gives the position of a planet (see Blunt et al. 2020, for details) along its orbit, with θ, the position angle at the average epoch of the observations. This improves convergence relative to sampling from the τ parameter directly since it 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.

2.8. *Modelling Language*

To specify the structure of a system model, Octofitter provides a domain specific modelling language. This simple language allows one to independently specify for each planet and the host star what parameterizations should be used and what observations correspond to what object (e.g. attach radial velocity measurements to a planet and/or to the star). Orbital parameters can be drawn from arbitrary prior distributions, fixed to particular values, or computed from combinations of other system parameters.

This modelling language makes it convenient to fit simple systems like the orbit of one planet to relative astrometry measurements and more complex systems up to hierarchical models and multi-planet 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 Visual{KepOrbit} 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 `UniformCicular` 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 freely combine them. The only requirements are that they are differentiable, smooth, and return a finite value at all points in the domain given by the priors.

This modelling approach, by being declarative rather than imperative, as in `exoplanet.py`, allows us to transform and evaluate the model in several ways. One key restriction is that each prior is proper, meaning that it is a true probability distribution that integrates to unity. This is in contrast to some of the defaults used by `orvara` which adopt 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 2.12.

2.9. *Numerical Methods*

From a model definition, Octofitter can generate efficient machine code using runtime 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 runtime overhead for features that aren't 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 semi-major axes are not possible to construct and wont be hit by a sampler. This transformation is performed by the Julia package `Bijectors.jl`[3].

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 the 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 pre-calculated 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 $2\times$ compared to $10.2\times$ 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 MCMC sampler packages including `AdvancedMH`, `AdvancedHMC`, and `MCMCTempering`. This allows users to select the best sampler for their

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

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 3.

When using the default NUTS sampler, we use a dense mass matrix, a jittered leapfrog integrator, a multinomial trajectory sampler, and allow the user to specify a maximum tree depth. We initialize the sampler by first 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 this procedure was more robust than trying to determine the maximum a-posteriori value with an optimizer for multi-modal posteriors like those found when modelling 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 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 on the order of 1000 to 10,000.

2.10. *Kepler Solver*

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

Kepler's equation is transcendental and can't 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. The approach of Octofitter is to wrap many different pluggable Kepler solvers that can be useful in different scenarios, and use a fast non-iterative method as a default.

A non-exhaustive 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`[4] julia package.

Since these solvers are all implemented in pure Julia, there is no overhead from calling between Python and a C / Cython[5] 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 default choice `Markley` 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 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 iterative Kepler equation solver would lead to poor performance. Using implicit differentiation, it can be shown that the derivative of eccentric anomaly with respect to mean anomaly or eccentricity has an analytic expression provided that the primal value of eccentric is already known (eqs. A5 and A6); that is, if we have already solved Kepler's equation for eccentric anomaly, we can calculate its gradient very inexpensively. We supply these equations as a manual rule to the automatic differentiation library.

2.11. *Analysis and Visualization*

Once sampling is complete, Octofitter supports the user in testing the convergence of their chain or chains. The results of sampling are returned in the form of a `Chains` object 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 compute time used. All variables are returned including those that were fixed to a constant or

[4] https://juliamath.github.io/Roots.jl/

[5] https://cython.org/

calculated deterministically from combinations of other variables. An automatic summary is output that includes plausible intervals, expectation values, effective sample sizes (ESS), and $\hat{R}$ statistics for each parameter. Users can test the convergence of their chain or chains by assessing those ESS and $\hat{R}$ values, creating a trace plot, and or by 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 velocimetry).

When visualizing an orbital posterior, a common challenge is ensuring 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 apoastron 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 the greatest curvature, creating smooth arcs with fewer points.

Octofitter includes functions to generating plots that visualize the posterior and compare it to the input data. This includes astrometry, separation, position angle, radial velocity, 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, beginning with Figures 4 and 7.

2.12. *Simulation Based Calibration*

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 modelling 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 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 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. An input of real observations are used by Octofitter in this procedure. The epochs and uncertainties of the observations are kept the same, but the observed values themselves are calculated from the orbit drawn from the priors. This should in general be applied to each model to confirm that it 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 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 AB and FG from log-uniform prior distributions rather than Gaussian distributions centred 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 in 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 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, that users of Octofitter can interpret their results with an appropriate level of skepticism in accordance with the level of bias detected.

2.13. *Other Packages*

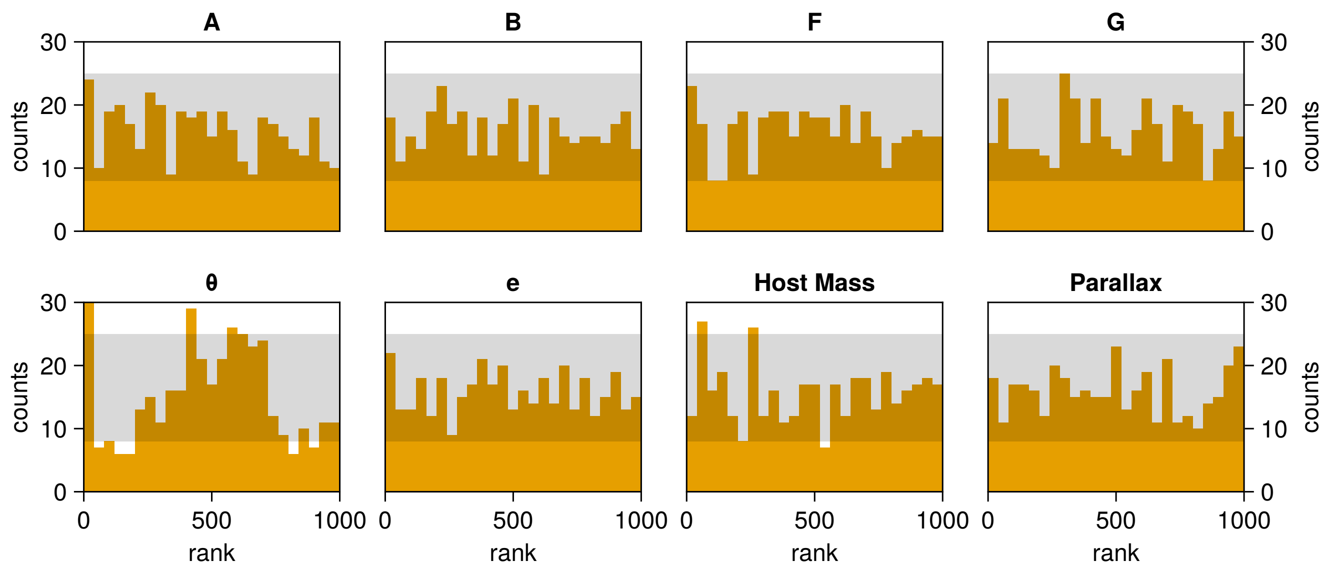

Figure 3. Sample results of running the simulation-based-calibration procedure on a model consisting of a single planet parameterized with Thiele-Innes elements (A, B, F, and G) and position angle θ. Each count represents a model fit to a different simulated system. The horizontal band gives a 99% range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

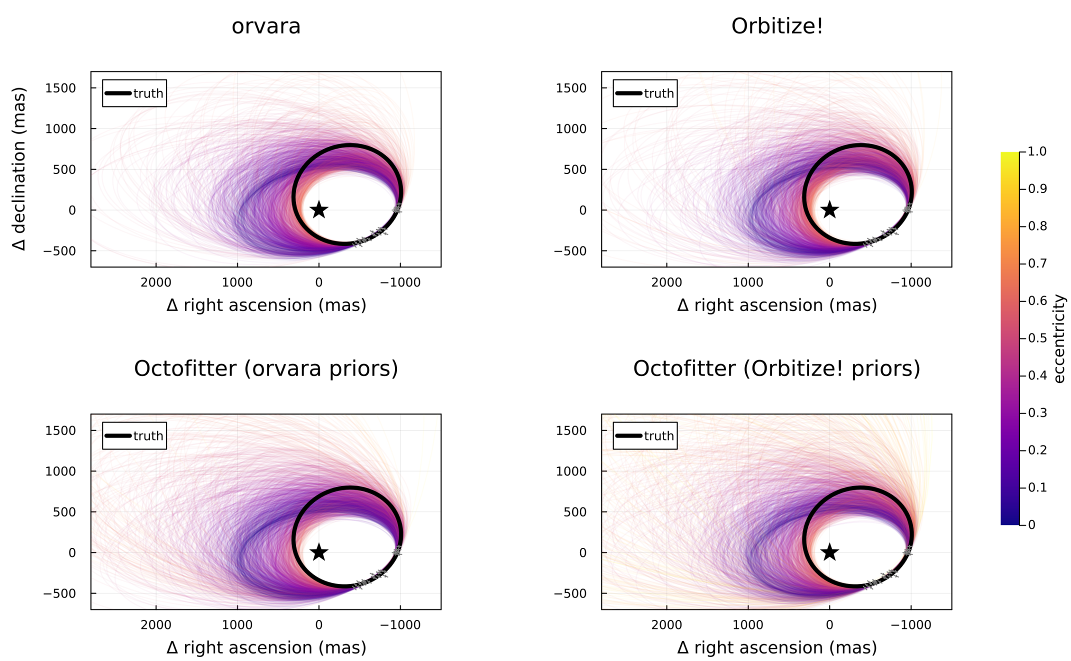

Figure 4. Orbit fitting posteriors visualized in the plane of the sky, compared between three packages: orvara, Orbitize!, and Octofitter.

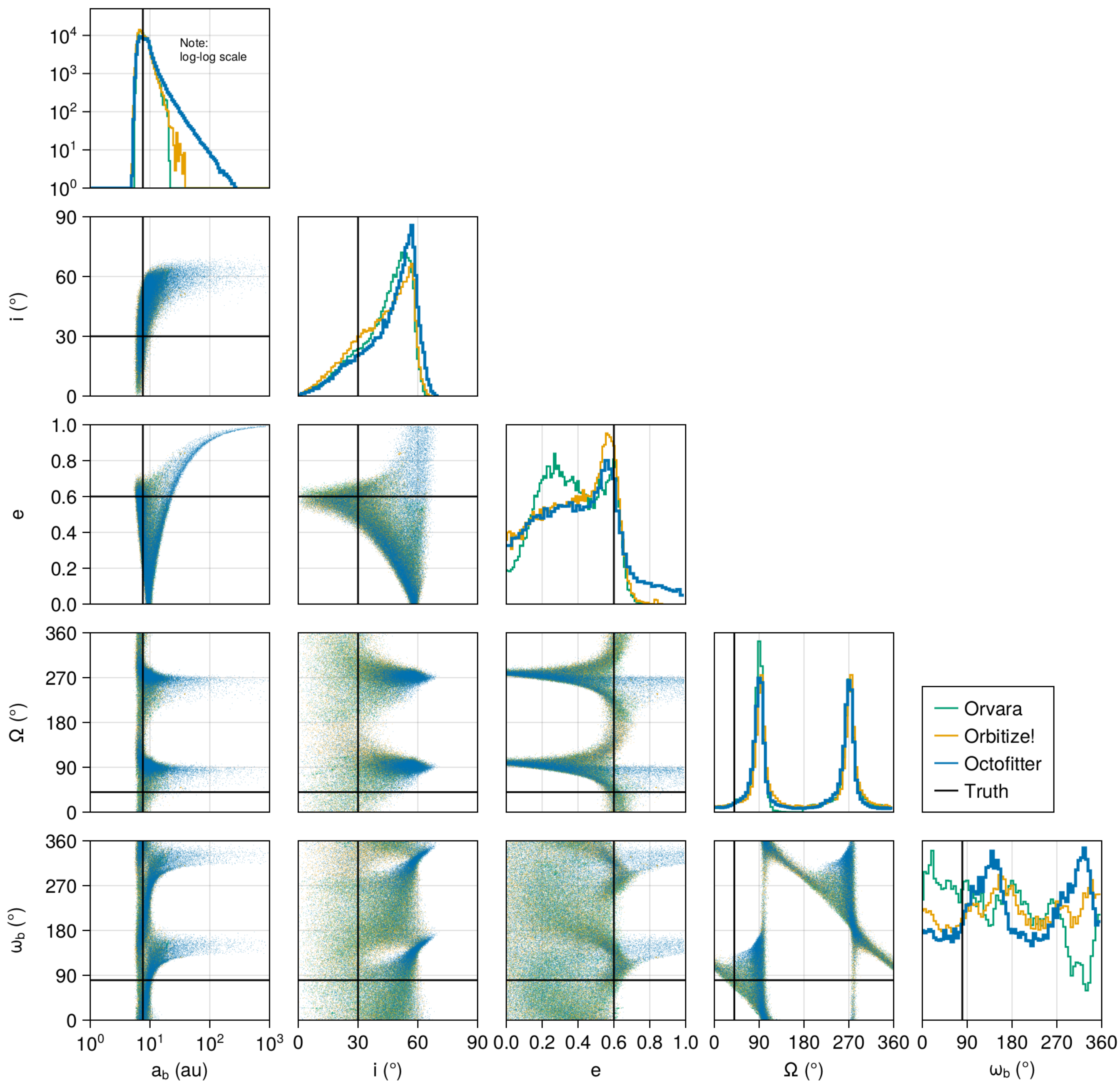

Figure 5. Corner plot of key orbital parameters for the case shown in Figure 4 compared between three packages: Orvara, Orbitize!, and Octofitter. Semi-major axis is plotted on a log scale to reveal how the sampler behaviour differs in the long tail. Though we expect all packages to eventually converge 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.

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 exoplanet.py (Foreman-Mackey et al. 2021). Most relevant to image- and visibility-based modelling are packages for fitting relative astrometry, radial velocity, and proper motion anomaly like Orbitize! (Blunt et al. 2020), orvara (Brandt, T. D. 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 (Dallant et al, 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 Markov Chain Monte Carlo (orvara and Orbitize!), Hamiltonian Monte Carlo (Hoffman & Gelman 2014) (exoplanet.py), and nested sampling (Skilling 2004) (efit). With the exception of exoplanet.py, the majority of these codes have used first order samplers. That is to say, the rely only on the evaluating the posterior density and do not calculate or make use of gradient information.

3. DEMONSTRATION WITH RELATIVE ASTROMETRY

We begin by demonstrating Octofitter by fitting orbits to simulated relative astrometry measurements which is one of the simplest usecases 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 the same fit using two other popular orbit fitting packages: orvara and Orbitize!.

We followed the best practices laid out in 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. We therefore 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 semi-major axis between 0.1 and 1000 AU. We ran the orvara and Orbitize! packages with settings recommended by their authors (4 temperatures and 100 walkers for Orvara, 20 temperatures and 1000 walkers for Orbitize!). 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. For Octofitter on the other hand, we initialized it automatically according using our procedure of drawing from the priors and selecting the parameters with the highest posterior density.

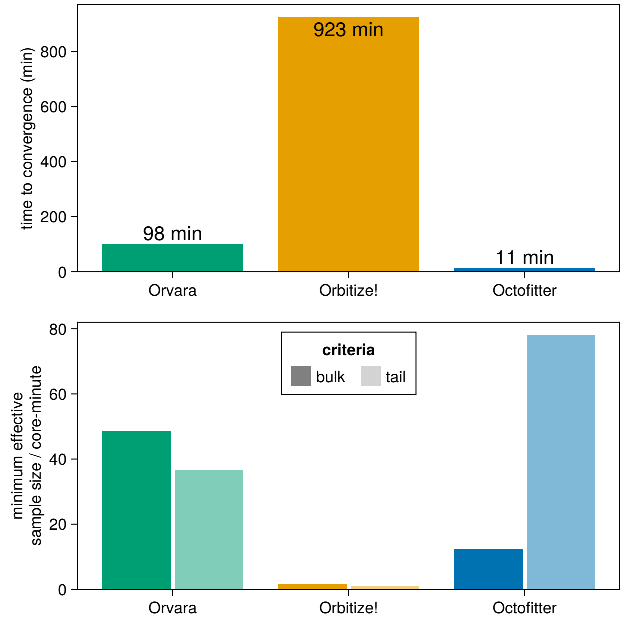

Figure 6. Comparison between packages of the average time until chains converged to a stationary distribution and the rate at which independent posterior samples are generated. The effective sample size rate was measured separately using the bulk and tail methods of MCMCDiagnostics.jl. These results are based on the astrometry presented in Table 1. These results are expected to depend strongly on hardware, the input data, and choice of priors.

For Orvara and Orbitize!, we drew 2,000,000 samples from each walker to ensure that the results were converged. For Octofitter , we ran adaptation for 5,000 iterations and then drew 20,000 samples. The resulting posteriors are compared in Figure 4.

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 10 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[6] to evaluate how well the samplers were converged at the tails of the distribution. We considered it converged once $\hat{R}$ became less than 1 ± 0.001 for

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

all variables. We evaluated this on all chains 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 which include the true orbit (Figure 5). This should serve to further improve confidence in the results of all packages.

The one exception is that the orbit posterior from Octofitter contains more samples from a high-eccentricity, high semi-major 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 dataset (Figure 3) reveals than Octofitter is acceptably calibrated and is not over-estimating the spread of the posterior. It is not computationally feasible to perform the same SBC experiment with the other packages, but this likely indicates 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 to converge (Figure 6). Since assessing convergence of MCMC chains can be difficult, we suggest that 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 Markov-Chain Monte Carlo, all packages would approach the same distribution given infinite time. This comparison therefore is meant only to illustrate the typical efficiency one might expect with Octofitter on similar problems and reasonable run times.

4. DEMONSTRATION WITH RELATIVE ASTROMETRY, RV, AND PMA

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

The HD91312 system consists of a $1.6 M_\odot$ star orbited by a binary companion with a mass of approximately $0.3 M_\odot$ and separation of roughly 10 AU (Chilcote et al. 2021). The companion was discovered by direct imaging by a search targeting accelerating stars (Currie et al. 2021) from the Hipparcos-GAIA catalogue of accelerations (HGCA; Brandt, T. D. 2021). We now reproduce the orbital analysis of the discovery paper in order to demonstrate Octofitter's modelling capabilities when applied to a combination of relative astrometry, radial velocity, 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 radial velocity data originally from Borgniet et al. (2019). As the original data was not forthcoming, we measured it from the PDF plots submitted to the 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 their analysis for the purpose of comparing codes and demonstrating Octofitter rather than any physical motivation on our part.

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

5. DEMONSTRATION WITH IMAGES

We now present a series of simulations showing how this framework allows us to detect planets using multi-epoch direct images. We selected 50 sequences from the Gemini Planet Imager Exoplanet Survey (Nielsen et al. 2020) processed using a forward model matched filter (FFMF, Ruffio et al. 2017) that 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

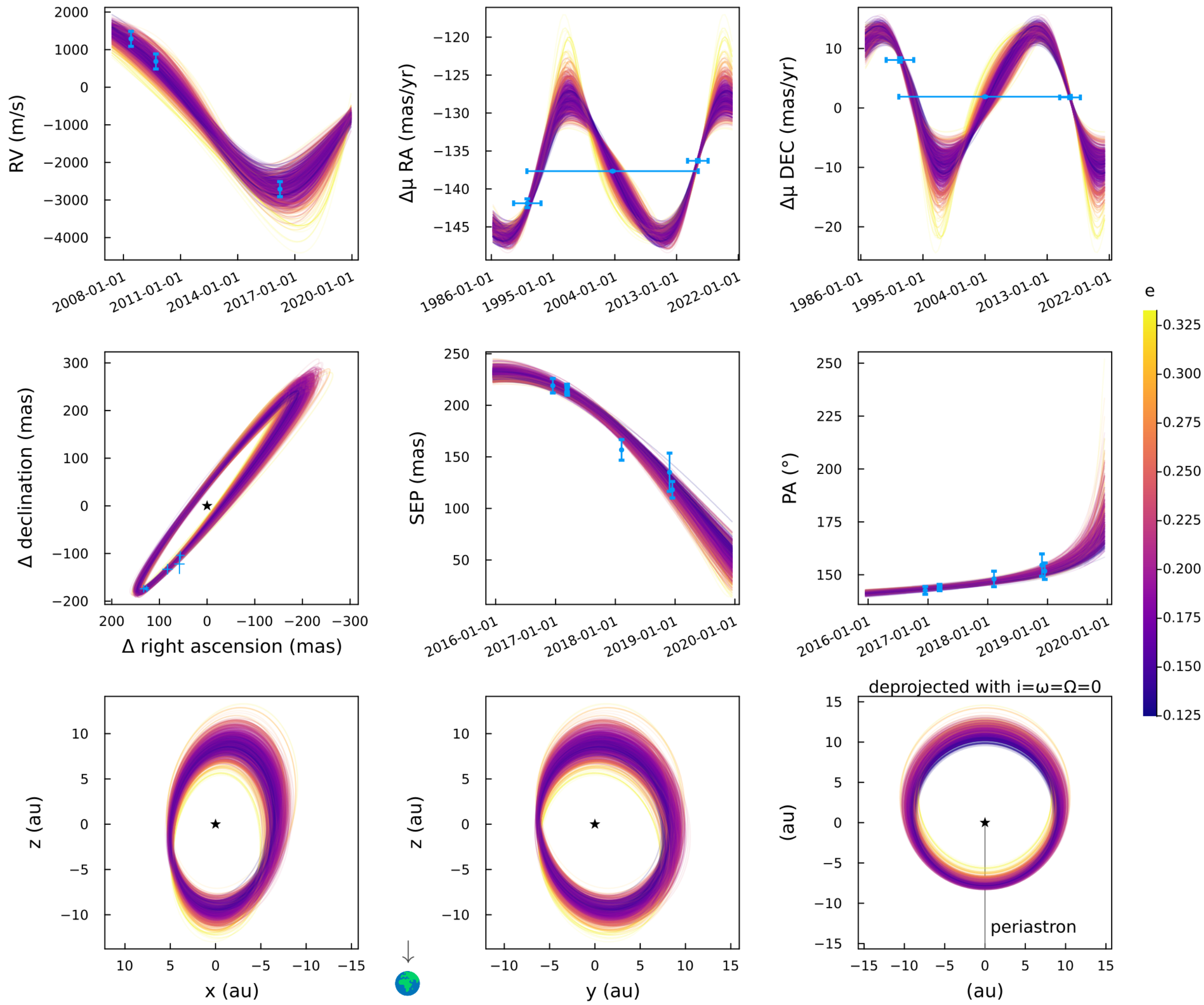

Figure 7. Sample plot output from Octofitter using data on the HD 91312 system. The top row visualizes the orbital posterior compared with velocity measurements of the host. The middle row shows the posterior compared with astrometry measurements in the plane of the sky and in separation and position 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.

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 $1\mathrm{M}_\odot$ 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.

For all models, we adopted a $1 \pm 0.1\mathrm{M}_\odot$ prior on M, a uniform $0 - 30\mathrm{M}_{\mathrm{jup}}$ prior on m, Gaussian prior on parallax, uniform prior on a, and 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 SNR ranging between 0 and 5 in each image. Finally, we applied our model to each of these simulated datasets to arrive at a grid of recovered SNR values as a function of number of epochs and SNR per epoch (Figure 9). We find that we are able to recover planet detections with arbitrarily low SNR per epoch despite orbital motion, provided with have a sufficient number of observations.

Figure 10 shows how the recovered SNR compares with the SNR we would expect for combining images

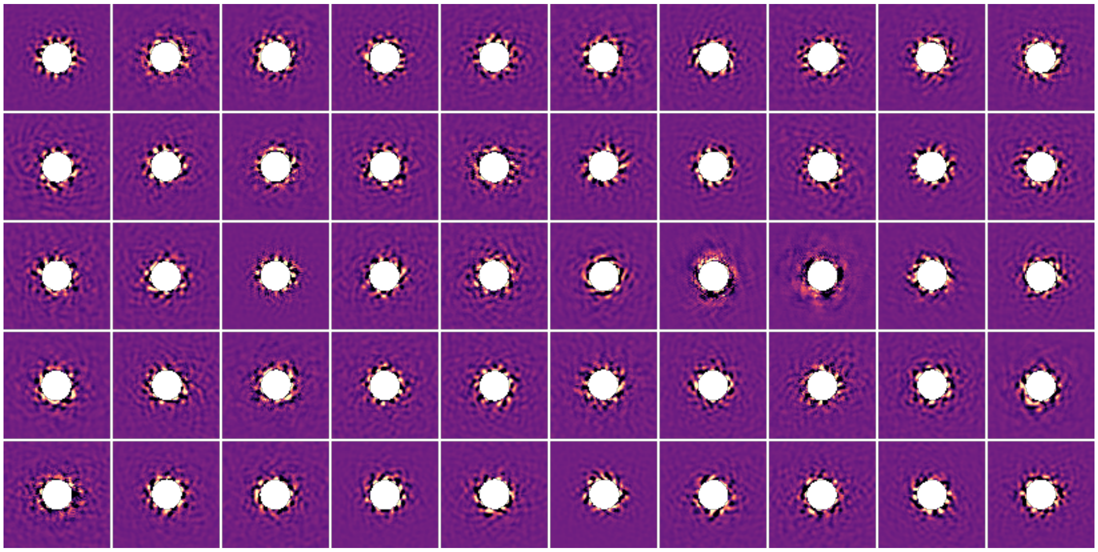

Figure 8. GPI sequences used for simulations in this section. Each image was normalized to have the same average contrast at 200 mas separation from the star (just outside the edge of the mask). The images displayed above contain a simulated planet orbiting CCW at an average of just SNR 1 per epoch,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

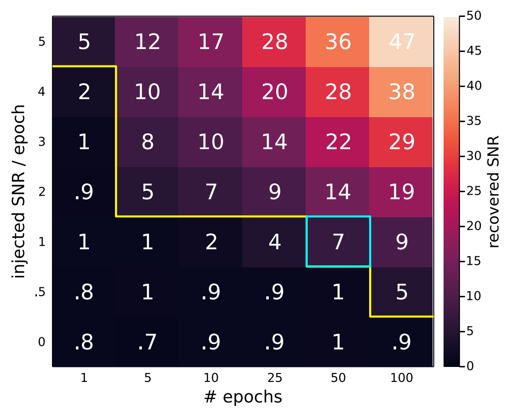

Figure 9. Posterior SNR for circular face-on simulations with different numbers of epochs, and injected SNR per epoch. Cells above the yellow line would be detected with a 5σ threshold. The simulated data used for cell highlighted in cyan is shown in Figure 8 and photometry posterior in Figure 2.

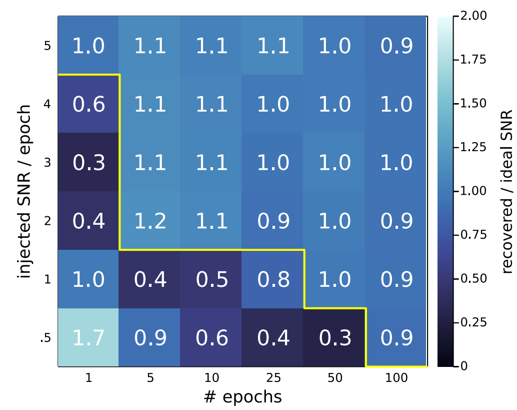

Figure 10. Posterior SNR for circular, face-on orbits compared to what we expect for adding signals in the presence of uncorrelated Gaussian noise. SNR grows according to the square root of the number of epochs when the final combined SNR of the planet $\gtrsim 5$. Otherwise, there are many possibilities to consider and the SNR drops.

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 SNR is greater than ~ 5. For instance, a planet injected into 100 epochs spaced one month apart, at an SNR per epoch of just 0.5 (below the noise) is still robustly detected at a final SNR of 5. This ideal scaling indicates that there is little to no correlation in the speckle noise between sequences when there is orbital motion present. We do note that since each image was taken of a different star, this is a best case scenario. It is possible that repeated observations of the same target could lead to correlated residuals that reduce this scaling, though orbital motion of the planet should also act to mitigate this.

The left-hand column of Figures 9 and 10 are of particular note. They show that the model fails to detect a point source injected into a single image at an SNR of 4. The failure to detect planets with expected significance below SNR ≈ 5 can be understood by contrasting our detection criteria with the standard used in the field. In these models, we consider the overall SNR marginalized over all locations in the image (or orbits through a sequence), whereas typically, one looks at the maximum SNR at any given location. For example, in any SNR 5 detection of a planet, numerous other SNR 2 and 3 peaks exist. When looking at the SNR marginalized over all locations, these other peaks serve to reduce the final SNR. This makes the SNR calculated from the marginal photometry posterior a more stringent planet detection test.

6. DEMONSTRATION WITH APERTURE MASKING INTERFEROMETRY AND RADIAL VELOCITY

We now demonstrate Octofitter using a combination of simulated radial velocity data and aperture masking inteferometry (AMI) visibilities. We considered a plausible scenario where a planet had been detected and orbit characterized using radial velocity, and is then followed up with a series of three observations with JWST NIRISS AMI (Doyon et al. 2012; Sivaramakrishnan et al. 2023). For this experiment we connected the mass of the planet to its photometry in Ms band 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 semi-major 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.

The radial velocity data was simulated by adding zero mean Gaussian noise with a standard deviation of 25 m/s 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 was generated using Equation 3 for a single companion following the given orbit. We added zero mean Gaussian noise to the calculated closure phases, with a closing triangle dependant 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. For this example we used only the closure phase and did not include the squared visibilities.

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 11.

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 modelling across epochs can be used to increase the significance of AMI and other similar interferometric observations.

7. CONCLUSION

We have presented a new code, Octofitter, for modelling exoplanet relative astrometry, radial velocity, and proper motion anomaly, as well as performing non-traditional tasks like detecting moving point sources across images and interferometric observables.

- 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.

- We compared the results of Octofitter to the popular packages `orvara` and `Orbitize!` and found that all three arrive at similar posterior distributions.

- 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 datasets.

- We demonstrated a combined fit to relative astrometry, radial velocity, and proper motion anomaly of the HD 91312 system and found a planet mass that agreed with previous results.

- We demonstrated the ability to detect arbitrarily faint companions despite orbital motion using simulations and data from the Gemini Planet Imager Exoplanet Survey.

- We demonstrated a combined model of simulated radial velocity and multi-epoch JWST/NIRISS aperture masking interferometry.

Octofitter is a powerful new tool that will allow the community to broadly apply joint multi-epoch, multi-instrument, and multi-band direct imaging, interferometry, and velocimetry. 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!, exoplanet.py, and orvara packages. Their open

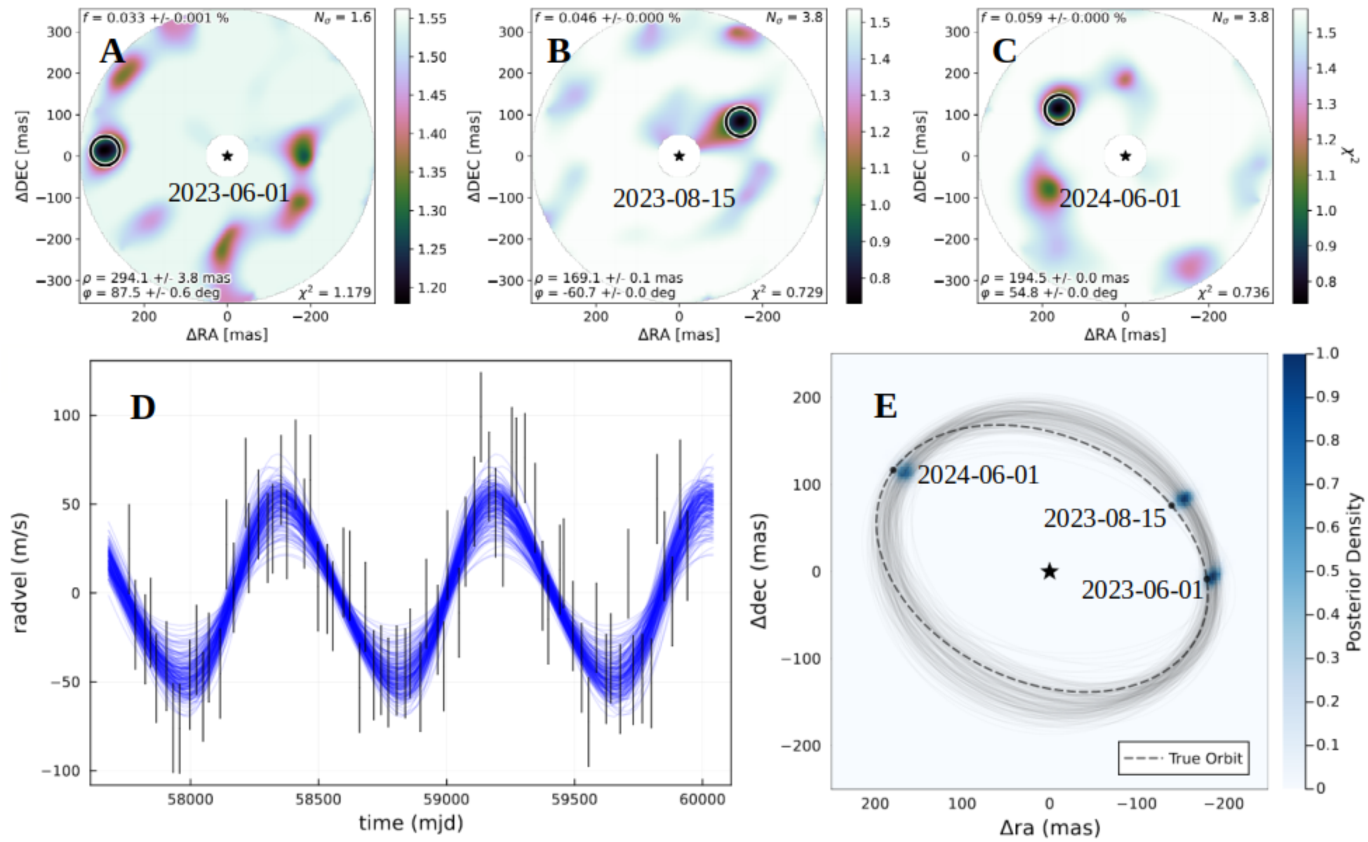

Figure 11. Joint modelling of AMI and RV data. **A-C**: Recovered χ^2 maps at each independent AMI epoch created using Fouriever. **D and E:** joint radial velocity and AMI posteriors visualized as an RV time series (D) and in the plane of the sky (E).

source software served as valuable references that greatly aided in the development of this package.

WT and CM 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 (GPI), Gaia, Hipparcos

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

APPENDIX

A. DERIVATION OF CARTESIAN AND CELESTIAL POSITION, VELOCITY, AND ACCELERATION

To properly fit to astrometry, proper motion anomaly, angular acceleration, and radial velocity 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 semi-major 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

$$M(t) = \frac{2\pi(t-t_0)}{T}. \tag{A4}$$

Here, t_0 is the time of periastron passage and T is the orbital period. Since Eq. 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

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

However, the frame of the secondary and 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), i (the inclination), and Ω (the longitude of the ascending node). The rotations associated with these angles are given by the rotation operators

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

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

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

Explicitly, we have

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

Inspecting this equation, we see that $\mathbf{r}_{\rm obs}(t)$ is dependent on a, e, i, ω, and Ω. In addition, there is dependence on t_0 and T because of Eqs. 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 $(a, e, i, \omega, \Omega, t_0)$; these are referred to as the Campbell elements. Note that in this paper, we replace t_0 with

$$\tau = \left(\frac{t_0 - t_{\rm ref}}{T}\right) \mod 1, \tag{A11}$$

where we choose $t_0 = 58849.0$ MJD (January 1, 2020) (Blunt et al. 2020). We now differentiate $\mathbf{r}_{\rm obs}(t)$ to obtain $\mathbf{v}_{\rm obs}(t)$. 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 $\mathbf{r}_{\rm obs}(t)$, we get that $\mathbf{v}_{\rm obs}(t)$ is given by

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

We determine $\mathbf{a}_{\rm obs}(t)$ 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 $\mathbf{v}_{\rm obs}(t)$ yields

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

Having expressions for $\mathbf{r}_{\rm obs}(t)$, $\mathbf{v}_{\rm obs}(t)$, and $\mathbf{a}_{\rm obs}(t)$ is quite useful, since they allow us to determine the corresponding quantities in celestial coordinates with ease. Furthermore, the radial velocity of the secondary planet with respect to its primary can be read off directly as the z-component of Eq. 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 get the primary's radial velocity with respect to the secondary, we simply employ conservation of momentum and multiply the previous expression by $-M_{\rm s}/(M_{\rm s}+M_{\rm p})$. To determine the radial velocities measured by an external observer, we add the barycentre radial velocity γ. Additional linear, 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. Doing so, we get that the observed radial velocities of the primary and secondary are

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

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

where $M_{\rm s}$ is the mass of the secondary, $M_{\rm p}$ is the mass of the primary, and $t_{\rm i}$ is the time of the initial observation. Using these expressions, orbital and physical parameters can be fit to radial velocity data.

A.2. *Celestial Coordinates*

Now that we know $\mathbf{r}_{\rm obs}(t)$, $\mathbf{v}_{\rm obs}(t)$, and $\mathbf{a}_{\rm obs}(t)$, 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 declination (upwards on the sky), while the positive y-axis points in the direction of positive right ascension (leftwards on the sky). This choice of coordinates is important to note, since it differs from the right/up orientation of the x/y axes seen elsewhere.

Keeping this coordinate convention in mind, we can now 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 right ascension and declination offsets of the secondary from its primary are

$$\Delta\alpha = \frac{y_{\rm obs}}{d}, \quad \Delta\delta = \frac{x_{\rm obs}}{d}, \tag{A21}$$

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

$$\Delta\dot{\alpha} = \frac{v_{y,\rm obs}}{d}, \quad \Delta\dot{\delta} = \frac{v_{x,\rm obs}}{d}, \tag{A22}$$

where $v_{x,\rm obs}$ and $v_{y,\rm obs}$ are the x and y-components of Eq. A14, and that

$$\Delta\ddot{\alpha} = \frac{a_{y,\rm obs}}{d}, \quad \Delta\ddot{\delta} = \frac{a_{x,\rm obs}}{d}, \tag{A23}$$

where $a_{x,\rm obs}$ and $a_{y,\rm obs}$ are the x and y-components of Eq. A16. The equations for $\Delta\alpha$ and $\Delta\delta$ can be used to fit orbital and physical parameters to astrometry data, while the equations for $\Delta\dot{\alpha}$ and $\Delta\dot{\delta}$ can be used to fit orbital and physical parameters to proper motion anomaly data. $\Delta\ddot{\alpha}$ and $\Delta\ddot{\delta}$ may find application to angular acceleration data in the future, and in calculating model derivatives for use in higher order samplers.

B. DERIVATION OF PARAMETER τ FROM POSITION ANGLE θ

The parameter τ used in Orbitize! and Octofitter gives the fraction of the orbit completed by a planet after periastron, at some reference epoch $t_{\rm ref}$. This is a useful parameterization since 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 pathalogical behaviour as the other orbital parameters change. One solution is to adjust the reference epoch to each given dataset, but a further improvement for many cases is to instead adopt the observed position angle θ at epoch t as an independent variable. This is an improvement 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 $\bar{\omega}$, the host mass M, the eccentricity e, and the Thiele-Innes constants $A, B, F,$ and G given in Eqs. C29 – C32. From these, we write the transformation matrix

$$T = \begin{bmatrix} A & B \\ F & G \end{bmatrix} \tag{B24}$$

and find the unit vectors

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = T^{-1} \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}. \tag{B25}$$

We can then calculate true anomaly ν as

$$\nu = \arctan(\hat{y}, \hat{x}), \tag{B26}$$

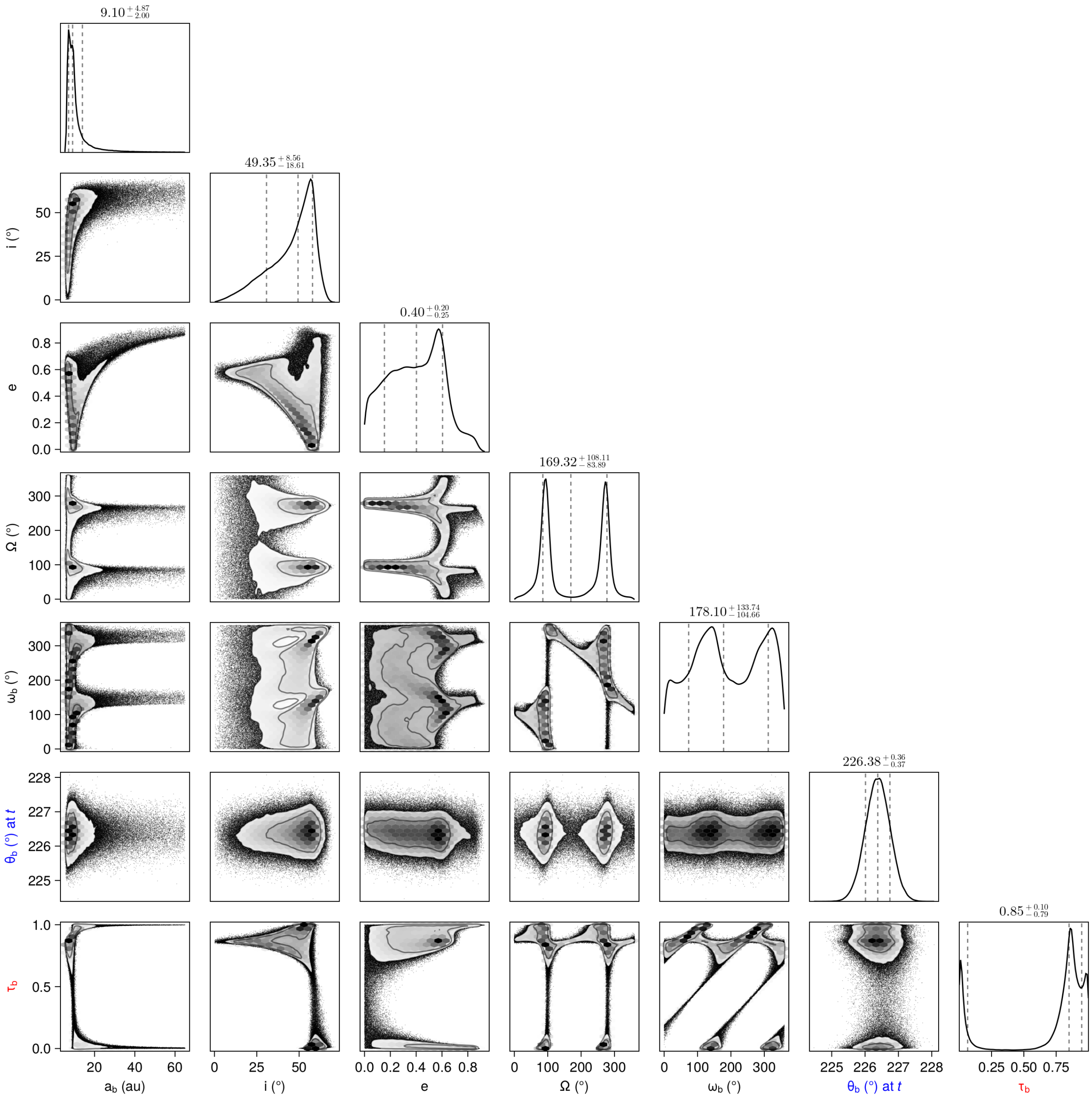

Figure 12. Comparison of θ (blue, second last row) and τ (red, last row) paramterizations of the same orbital posterior. The θ marginal posterior is nearly Gaussian, and has simple relationships with all other orbital parameters. By contrast, the τ marginal posterior is less informative and has complex relationships to ω and Ω. Complex structures in a posterior decrease sampling efficiency and could in some cases lead to biased results.

and mean anomaly as

$$\mathrm{MA} = \arctan(-\sin(\nu)\sqrt{1-e^2}, -\cos(\nu) - e) + \pi - \frac{e\sqrt{1-e^2}\sin(\nu)}{1+e\cos(\nu)}. \tag{B27}$$

From this, the parameter τ can be calculated using the period as follows. First, calculate the semi-major axis a using Eq. C35. The period is then $P = \sqrt{a^3/M}$, mean motion is $n = 2\pi/P$, and the epoch of periastron passage is

$t_{\rm peri} = t - {\rm MA}/n - t_{\rm ref}$. Finally,

$$\tau = \frac{t_{\rm peri}}{P}. \tag{B28}$$

The corner plot in Figure 12 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.

C. THIELE-INNES ELEMENTS

In Section A.1, we saw that Keplerian orbits can be uniquely characterized by the Campbell elements $(a, e, i, \omega, \Omega, \tau)$. However, Keplerian orbits can be equivalently characterized by (e, A, B, F, G, θ), where e is the orbital eccentricity, A, B, F, and G are the Thiele-Innes elements, and θ is the position angle. The Thiele-Innes elements are defined as

$$A = a\bar{\omega}(\cos(\Omega)\cos(\omega) - \sin(\Omega)\sin(\omega)\cos(i)), \tag{C29}$$

$$B = a\bar{\omega}(\sin(\Omega)\cos(\omega) + \cos(\Omega)\sin(\omega)\cos(i)), \tag{C30}$$

$$F = a\bar{\omega}(-\cos(\Omega)\sin(\omega) - \sin(\Omega)\cos(\omega)\cos(i)), \tag{C31}$$

$$G = a\bar{\omega}(-\sin(\Omega)\sin(\omega) + \cos(\Omega)\cos(\omega)\cos(i)), \tag{C32}$$

where $\bar{\omega}$ is the parallax distance to the system.

C.1. *Converting Thiele-Innes to Campbell elements*

As we can see in Eqs. C29 – C32, 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

$$u = \frac{1}{2}(A^2 + B^2 + C^2 + D^2), \tag{C33}$$

$$v = AG - BF. \tag{C34}$$

We can recover a using u and v as follows:

$$a = \frac{\sqrt{u + \sqrt{(u+v)(u-v)}}}{\bar{\omega}}. \tag{C35}$$

Once we have a, i immediately follows from

$$i = \arccos\left(\frac{v}{a^2\bar{\omega}^2}\right). \tag{C36}$$

Note that since $\cos(\theta) = \cos(-\theta)$, and Eqs. C29 – C32 involve only $\cos(i)$, this is technically an equation for $|i|$. Next, we define

$$j = \arctan\left(A + G, B - F\right), \tag{C37}$$

$$k = \arctan\left(A - G, B + F\right), \tag{C38}$$

where $\arctan(y, x)$ is a quadrant-sensitive version of $\arctan(y/x)$ (sometimes written as $\arctan2(y, x)$ or $\mathrm{atan2}(y, x)$). From these quantities, ω and Ω are given by

$$\omega = \frac{1}{2}(j - k), \tag{C39}$$

$$\Omega = \frac{2}{2}(j + k). \tag{C40}$$

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

REFERENCES

Berger, J. P. 2003, in EAS Publications Series, Vol. 6, EAS Publications Series, ed. G. Perrin & F. Malbet, 23, doi: 10.1051/eas:2003002

Berry, M., & Healy, L. 2002, The Generalized Sundman Transformation for Propagation of High-Eccentricity Elliptical Orbits, Tech. rep., Naval Research Lab Washington DC

Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. 2012, arXiv e-prints, doi: 10.48550/arXiv.1209.5145

Blunt, S., Wang, J. J., Angelo, I., et al. 2020, The Astronomical Journal, 159, 89, doi: 10.3847/1538-3881/ab6663

Borgniet, S., Lagrange, A.-M., Meunier, N., et al. 2019, A&A, 621, A87, doi: 10.1051/0004-6361/201833431

Brandt, T. D. 2021, ApJS, 254, 42, doi: 10.3847/1538-4365/abf93c

Brandt, T. D., Dupuy, T. J., Li, Y., et al. 2021, arXiv:2105.11671 [astro-ph]. http://ascl.net/2105.11671

Brandt, G. M., Michalik, D., Brandt, T. D., et al. 2021, The Astronomical Journal, 162, 230, doi: 10.3847/1538-3881/ac12d0

Brandt, T. D. 2021, The Astrophysical Journal Supplement Series, 254, 42, doi: 10.3847/1538-4365/abf93c

Brandt, T. D., Dupuy, T. J., Li, Y., et al. 2021, arXiv:2105.11671 [astro-ph]. http://arxiv.org/abs/2105.11671

Brooks, S. P., & Gelman, A. 1998, Journal of Computational and Graphical Statistics, 7, 434, doi: 10.1080/10618600.1998.10474787

Butler, R. P., Vogt, S. S., Laughlin, G., et al. 2017, VizieR Online Data Catalog, J/AJ/153/208. https://ui.adsabs.harvard.edu/abs/2017yCat..51530208B/abstract

Chilcote, J., Tobin, T., Currie, T., et al. 2021, The Astronomical Journal, 162, 251, doi: 10.3847/1538-3881/ac29ba

Collaboration, G., Abuter, R., Accardo, M., et al. 2017, Astronomy and Astrophysics, 602, A94, doi: 10.1051/0004-6361/201730838

Cook, S. R., Gelman, A., & Rubin, D. B. 2006, Journal of Computational and Graphical Statistics, 15, 675, doi: 10.1198/106186006X136976

Currie, T., Brandt, T., Kuzuhara, M., et al. 2021, A New Type of Exoplanet Direct Imaging Search: The SCExAO/CHARIS Survey of Accelerating Stars, arXiv, doi: 10.48550/arXiv.2109.09745

Danisch, S., & Krumbiegel, J. 2021, Journal of Open Source Software, 6, 3349, doi: 10.21105/joss.03349

Dickey, J. M. 1971, The Annals of Mathematical Statistics, 42, 204, doi: 10.1214/aoms/1177693507

Doyon, R., Hutchings, J. B., Beaulieu, M., et al. 2012, in Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, Vol. 8442 (SPIE), 1005–1017

Eastman, J. D., Rodriguez, J. E., Agol, E., et al. 2019, EXOFASTv2: A Public, Generalized, Publication-Quality Exoplanet Modeling Code, arXiv, doi: 10.48550/arXiv.1907.09480

Éric Thiébaut, & Young, J. 2017, J. Opt. Soc. Am. A, 34, 904, doi: 10.1364/JOSAA.34.000904

Ferrer-Chávez, R., Blunt, S., & Wang, J. J. 2021a, Res. Notes AAS, 5, 162, doi: 10.3847/2515-5172/ac151d

Ferrer-Chávez, R., Wang, J. J., & Blunt, S. 2021b, The Astronomical Journal, 161, 241, doi: 10.3847/1538-3881/abf0a8

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306, doi: 10.1086/670067

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, Publications of the Astronomical Society of the Pacific, 130, 044504, doi: 10.1088/1538-3873/aaaaa8

Gelman, A., & Rubin, D. B. 1992, Statistical Science, 7, 457, doi: 10.1214/ss/1177011136

Goldstein, H., Poole, C. P., & Safko, J. L. 2008, Classical Mechanics, 3rd edn. (San Francisco Munich: Addison Wesley)

Hartkopf, W. I., McAlister, H. A., & Franz, O. G. 1989, The Astronomical Journal, 98, 1014, doi: 10.1086/115193

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

Hoffman, M. D., & Gelman, A. 2014, Journal of Machine Learning Research, 15, 1593. http://jmlr.org/papers/v15/hoffman14a.html

Iglesias-Marzoa, R., López-Morales, M., & Jesús Arévalo Morales, M. 2015, Publications of the Astronomical Society of the Pacific, 127, 567, doi: 10.1086/682056

Kammerer, J., Cooper, R. A., Vandal, T., et al. 2023, PASP, 135, 014502, doi: 10.1088/1538-3873/ac9a74

Kasdin, N. J., Bailey, V., Mennesson, B., et al. 2020, in Space Telescopes and Instrumentation 2020: Optical, Infrared, and Millimeter Wave, ed. M. Lystrup, N. Batalha, E. C. Tong, N. Siegler, & M. D. Perrin (Online Only, United States: SPIE), 194, doi: 10.1117/12.2562997

Koop, G. 2003, Bayesian Econometrics (Chichester ; Hoboken, N.J: J. Wiley)

Le Coroller, H., Nowak, M., Wagner, K., et al. 2022, A&A, doi: 10.1051/0004-6361/202243576

Le Coroller, H., Nowak, M., Delorme, P., et al. 2020, Astronomy and Astrophysics, 639, A113, doi: 10.1051/0004-6361/202037605

Llop-Sayson, J., Wang, J. J., Ruffio, J.-B., et al. 2021, The Astronomical Journal, 162, 181, doi: 10.3847/1538-3881/ac134a

Markley, F. L. 1995, Celestial Mech Dyn Astr, 63, 101, doi: 10.1007/BF00691917

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

Mawet, D., Hirsch, L., Lee, E. J., et al. 2019, The Astronomical Journal, 157, 33, doi: 10.3847/1538-3881/aaef8a

Meyer, L., Ghez, A. M., Schödel, R., et al. 2012, Science, 338, 84, doi: 10.1126/science.1225506

Nielsen, E. L., Rosa, R. J. D., Wang, J. J., et al. 2020, AJ, 159, 71, doi: 10.3847/1538-3881/ab5b92

Nowak, M., Coroller, H. L., Arnold, L., et al. 2018, A&A, 615, A144, doi: 10.1051/0004-6361/201629531

O'Neil, K. K., Martinez, G. D., Hees, A., et al. 2019, AJ, 158, 4, doi: 10.3847/1538-3881/ab1d66

Philcox, O. H. E., Goodman, J., & Slepian, Z. 2021, Monthly Notices of the Royal Astronomical Society, 506, 6111, doi: 10.1093/mnras/stab1296

Pogorelyuk, L., Fitzgerald, R., Vlahakis, S., Morgan, R., & Cahoy, K. 2022, ApJ, 937, 66, doi: 10.3847/1538-4357/ac8d56

Revels, J., Lubin, M., & Papamarkou, T. 2016, arXiv:1607.07892 [cs]. http://ascl.net/1607.07892

Ruffio, J.-B., Macintosh, B., Wang, J. J., et al. 2017, ApJ, 842, 14, doi: 10.3847/1538-4357/aa72dd

Ruffio, J.-B., Mawet, D., Czekala, I., et al. 2018, The Astronomical Journal, 156, 196, doi: 10.3847/1538-3881/aade95

Sivaramakrishnan, A., Tuthill, P., Lloyd, J. P., et al. 2023, PASP, 135, 015003, doi: 10.1088/1538-3873/acaebd

Skemer, A. J., & Close, L. M. 2011, The Astrophysical Journal, 730, 53, doi: 10.1088/0004-637X/730/1/53

Skilling, J. 2004, AIP Conference Proceedings, 735, 395, doi: 10.1063/1.1835238

Soulain, A., Sivaramakrishnan, A., Tuthill, P., et al. 2020, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 11446, Optical and Infrared Interferometry and Imaging VII, ed. P. G. Tuthill, A. Mérand, & S. Sallum, 1144611, doi: 10.1117/12.2560804

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. 2020, arXiv:1804.06788 [stat]. http://ascl.net/1804.06788

Thompson, W., Marois, C., Do Ó, C. R., et al. 2023, AJ, 165, 29, doi: 10.3847/1538-3881/aca1af

Trifonov, T., Tal-Or, L., Zechmeister, M., et al. 2020, A&A, 636, A74, doi: 10.1051/0004-6361/201936686

Wagenmakers, E.-J., Lodewyckx, T., Kuriyal, H., & Grasman, R. 2010, Cognitive Psychology, 60, 158, doi: 10.1016/j.cogpsych.2009.12.001

Wright, J. T., & Howard, A. W. 2009, The Astrophysical Journal Supplement Series, 182, 205, doi: 10.1088/0067-0049/182/1/205

Xu, K., Ge, H., Tebbutt, W., et al. 2020, in Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference (PMLR), 1–10. https://proceedings.mlr.press/v118/xu20a.html

Part of this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.