

Strategies for accurate effective point spread function (ePSF) modelling on undersampled images

Emma Godden^{*} and Katherine M. Blundell

Department of Physics, University of Oxford, Keble Road, Oxford, OX1 3RH, UK

Accepted 2025 December 18. Received 2025 December 18; in original form 2025 July 2

ABSTRACT

Accurate modelling of the effective point spread function (ePSF) is essential for high-precision photometry and astrometry, particularly in undersampled imaging regimes. In this work, we build on a well-established ePSF modelling framework and its commonly used open-source Python implementation and demonstrate that several simple but effective modifications to existing ePSF modelling routines can significantly improve model accuracy. We use synthetic ePSFs to generate simulated data sets of stellar images, allowing us to evaluate the accuracy of ePSF models and determine the scale of the pixel-phase errors in resulting flux and position measurements. We systematically investigate how specific modelling choices affect ePSF accuracy, and evaluate the influence of oversampling, interpolation, gridpoint estimation, smoothing, star-sample distribution and dithering on photometric precision. We apply our refined ePSF modelling routine to images from the Global Jet Watch observatories, demonstrating its improved ability to recover an accurate ePSF for real astronomical images. Our findings highlight the importance of tailoring the modelling approach to the specific characteristics of the instrument and detector, as well as to the nature of the available imaging data used to construct the ePSF model. These results provide practical guidance for optimising ePSF construction, thereby improving the reliability of photometric and astrometric measurements.

Key words: software – data methods – algorithms – effective point spread function (ePSF) – photometry and astrometry.

1 INTRODUCTION

Accurate measurements within observational astronomy rely on knowledge of the instrumental point spread function (PSF). For astronomical images, in order to measure the brightness of a star, or its exact position, we need to understand how the light from a point source is spread out and blurred by the instrument that produced the image. Different scientific goals make different demands on the images taken by an instrument. For single-object photometric analysis, it is advantageous for the light from the target object to be spread across a large number of pixels on the detector, giving us a very well-sampled PSF. For imaging surveys, the aim is to maximise the number of objects we can have in an image by having a large field of view and a compact PSF to avoid blending between objects. For some ground-based telescopes, it is often possible to still achieve a moderately sampled PSF in wide-field images, however for other ground-based and most space-based telescopes, wide-field images come with the challenge of having an undersampled PSF.

There are significant challenges in obtaining accurate photometry and astrometry for images with an undersampled PSF. Aperture photometry methods are commonly used to measure the fluxes of point sources (stars) in wide-field images. These methods assume that it does not matter where exactly the light from a star falls, we only need to know within which pixels the light falls. Then we can estimate the flux of the star by simply taking the sum of the flux measurements in these pixels. The simplistic approach has some limitations. First, even for images with well-sampled PSFs, aperture photometry is unable to measure accurate fluxes for bright stars where the central pixels are saturated,¹ or stars in crowded fields where individual pixels may contain flux from multiple stars. Second, complications in photometry and astrometry

¹This excludes detectors with a well-characterized saturation spill, where excess charge from saturated pixels is redistributed into neighbouring pixels in a predictable manner (as in the case of *Kepler*; R. L. Morris et al. 2020). For such detectors, aperture photometry of bright saturated stars is still feasible, provided that all pixels containing the spilled flux are included in the aperture.

* E-mail: emma.godden@physics.ox.ac.uk

can arise due to the non-uniform nature of the sensitivity response across an individual pixel (T. R. Lauer 1999). Undersampled images can be particularly affected by this, causing a dependence on the flux reported by a pixel with where exactly within the pixel the peak of the PSF falls. This can lead to significant ‘pixel-phase’ systematic errors in the aperture-measured flux and in centroid estimates that rely directly on the recorded pixel fluxes (i.e. flux-weighted, or centre-of-mass, centroids).

One possibility to improve the flux and position measurements in these cases is to use PSF-fitting photometry. PSF-fitting photometry is able to measure the flux and position of saturated stars by fitting the wings of the PSF and masking the saturated pixels. PSF-fitting photometry is also able to measure fluxes and positions of stars in crowded images by using iterative fitting procedures. In the case of undersampled images, provided that not all of the detectable light from a star falls within just one pixel, it is possible to obtain measurements of stars using PSF-fitting methods. A slight change in the centroid position of the PSF changes the relative flux that falls between adjacent pixels, so with knowledge of the PSF we can measure the position and the flux of the star. If the intra-pixel sensitivity variations are properly accounted for in PSF-fitting photometry, it is possible to minimise pixel-phase error systematics in the resulting measurements of stars even in very undersampled images. However, this is only possible with an accurate model of the instrumental PSF and the intra-pixel sensitivity variations.

A variety of PSF-fitting photometry methods have been developed and refined over the past few decades in an attempt to obtain accurate positions and fluxes of stars for a range of different instruments. Here, we focus our attention on the effective PSF (ePSF) photometry method developed by J. Anderson & I. R. King (2000). The ePSF photometry method utilizes the fact that each image contains many samples of the PSF (i.e. via different stars in the field) that can be used to construct an empirical model of the PSF without any assumptions about its analytical form. J. Anderson & I. R. King (2000) make an important distinction between the instrumental PSF and the observed ‘effective PSF’, the latter being the result of integrating the instrumental PSF over the individual detector pixels. This pixel-convolved version, the ePSF, is essential for accurate photometric and astrometric measurements. This version of the PSF is particularly useful for images with an undersampled PSF where pixel-phase systematic errors often appear in the light curves and positions of stars, as described in T. R. Lauer (1999). Intra-pixel sensitivity variations are accounted for within the ePSF itself, hence avoiding pixel-phase bias systematics from occurring.

The purely empirical modelling of the ePSF makes this method adaptable to different instruments since no assumptions need to be made about the true shape of the ePSF, which is why we have chosen to focus on it here. It was initially developed for the WFPC2 camera of the *Hubble Space Telescope* (HST) (J. Anderson & I. R. King 2000) and has since been used on a wide range of space-based telescopes [ACS/HRC: J. Anderson & I. R. King (2004), ACS/WFC: J. Anderson & I. R. King (2006); A. Bellini et al. (2019), WFC3/UVIS: J. Anderson et al. (2015); J. Anderson & L. R. Bedin (2017), WFC3/IR: J. Anderson (2016), *Kepler*/K2: M. Libralato et al. (2016a, b), *James Webb Space Telescope* (JWST) NIRISS: M. Libralato et al. (2023), JWST MIRI M. Libralato et al. (2024a), JWST NIRCAM: D. Nardiello et al. (2022), *Euclid*: M. Libralato et al. (2024b), *TESS*: D. Nardiello et al. (2019), and *CSST* (Y. Wu et al. 2025)]. The ePSF method has also been adapted for use on ground-based telescopes [WFI@2.2m: J. Anderson et al. (2006) and HAWK-I@VLT: M. Libralato et al. (2014)].

When being deployed for such a wide range of instruments, each new implementation of the ePSF routine requires careful adjustments to produce accurate results. This can be a challenging and time-consuming task for newcomers to the field who are wanting to use ePSF photometry for their observations. It is important to understand the accuracy and limitations of the ePSF modelling since using an inaccurate ePSF model can result in significant pixel-phase systematic errors in photometric and astrometric measurements. This work aims to elucidate generalised strategies for how to build an accurate ePSF model for any instrument, with a focus on undersampled images. In Section 2 we give a summary of the ePSF method described in J. Anderson & I. R. King (2000), and in Section 3 we discuss its implementation in the `photutils.psf` subpackage of `Astropy` (L. Bradley et al. 2024), highlighting differences and limitations relative to the original formulation. Section 4 describes how we use synthetic analytical ePSF models to generate simulated images to use in the ePSF modelling tests. Section 5 investigates how different parts of the ePSF routine impact the accuracy of the ePSF model and resulting photometric and astrometric measurements. We discuss how some simple adaptations to the modelling routines discussed in J. Anderson & I. R. King (2000) and implemented in `photutils.psf` can significantly improve the accuracy of the ePSF modelling. Finally, in Section 6, we demonstrate the application of the improved ePSF modelling routine to images from the Global Jet Watch observatories.

2 A REMINDER ON THE EFFECTIVE PSF

This section aims to summarise the ePSF formalism and modelling routine as described in J. Anderson & I. R. King (2000), however the reader is recommended to refer back to this work for further details.

2.1 ePSF definition

Each star in an astronomical image is a sample of the instrumental PSF scaled by the star’s flux. However, the image of the star is not this instrumental PSF directly, but a collection of discrete pixels where the flux value of each individual pixel is the integral over the portion of the instrumental PSF which overlaps the pixel combined with the intra-pixel sensitivity profile. The intra-pixel sensitivity profile accounts for the fact that the response of a pixel is not uniform and there is a dependence on where precisely within a pixel a photon hits (T. R. Lauer 1999; S. B. Mahato et al. 2018). We can write down the flux value of a pixel centred at (i, j) in the vicinity of a star centred at (x_*, y_*) with flux f_* as:

$$P_{ij} = f_* \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R(x-i, y-j) \Psi_I(x-x_*, y-y_*) dx dy + s_{ij}, \quad (1)$$

where R is the intra-pixel sensitivity profile, Ψ_I is the instrumental PSF, and s_{ij} is the background value in that pixel.

The most important lesson from J. Anderson & I. R. King (2000) is that it is not necessary to try to deconvolve the instrumental PSF from the intrapixel sensitivity profile. The instrumental PSF is never directly observed, so photometric measurements of stars in an image with the instrumental PSF would require re-doing the integration undone by the deconvolution. Instead, we should consider the ‘effective PSF’ (ePSF). We can re-write equation (1)

as:

$$P_{ij} = f_* \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R(x, y) \Psi_I(x - \Delta x, y - \Delta y) dx dy + s_{ij}, \quad (2)$$

where we have changed coordinates to be in the reference frame of the pixel, $(x, y) \rightarrow (x - i, y - i)$, and defined the displacement of the pixel from the centroid of the star as $(\Delta x, \Delta y) = (x_* - i, y_* - j)$. We can then define the ePSF as a function of $(\Delta x, \Delta y)$:

$$\Psi_E(\Delta x, \Delta y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R(x, y) \Psi_I(x - \Delta x, y - \Delta y) dx dy. \quad (3)$$

Using the ePSF, the flux value of a pixel in the vicinity of a star simply becomes:

$$P_{ij} = f_* \Psi_E(\Delta x, \Delta y) + s_{ij}. \quad (4)$$

Therefore, if we have a model for the ePSF we can easily fit the pixels in an image of a star to provide a measurement of the flux and centroid position of the star. In comparison to other PSF photometry approaches which model the instrumental PSF directly, no time-consuming integration of the ePSF model is required in this fitting process. We simply look up the ePSF values at different $(\Delta x, \Delta y)$ to find the best-fitting parameters.

2.2 ePSF modelling

To perform ePSF fitting photometry, we first need an accurate model of the ePSF. Fortunately, each pixel within each star on the image is a sample of the ePSF at a specific $(\Delta x, \Delta y)$. Re-writing equation (4), the ePSF for a given pixel at its corresponding $(\Delta x, \Delta y)$ is

$$\Psi_E(\Delta x, \Delta y) = \frac{P_{ij} - s_{ij}}{f_*}. \quad (5)$$

If we have a set of stars with known fluxes and centroid positions, their pixel values can be used as samples to construct the ePSF. However, measuring stellar fluxes and positions requires an ePSF model in the first place. In order to accurately model the ePSF we need accurate measurements of stars, but accurate measurements of stars require an accurate model of the ePSF. J. Anderson & I. R. King (2000) resolved this degeneracy by iterating between measuring the ePSF and measuring the stars. Each new measurement of the ePSF improves the measurement of the stars and *visa versa*. They take an empirical approach in which the ePSF is represented as a table of values over a uniform grid of $(\Delta x, \Delta y)$ points, referred to as gridpoints. The ePSF at arbitrary $(\Delta x, \Delta y)$, which may be in between gridpoints, is found by interpolation between the tabulated values. The main steps used to generate this empirical ePSF model are summarised below.

(1) From a set of dithered images, select a sample of isolated stars with high signal-to-noise ratios. Make initial estimates of the fluxes and positions using alternative methods (e.g. centre-of-mass centroid and aperture photometry).

(2) Construct a table of ePSF values defined on a uniform grid of $(\Delta x, \Delta y)$ positions. Initially, all ePSF values should be set to zero.

(3) For each sample star, convert each pixel value into a sample of the ePSF at its corresponding $(\Delta x, \Delta y)$ position using equation (5).

(4) For each ePSF sample, calculate the difference between the measured sample and the current ePSF model. In the initial iteration, the ePSF model is null so the residuals are simply the sample values.

(5) For each gridpoint, take all the residual samples within a 0.25-pixel square and calculate the average residual. Adjust the corresponding ePSF table value by this average residual.

(6) The assumption is made that the ePSF model must be smooth. Enforce this by convolving the tabulated ePSF values with a pre-defined smoothing kernel.

(7) Ensure that the centre of the ePSF lies at the centre of the empirical model. The definition of the centre of the ePSF is somewhat arbitrary, but must be kept consistent.

(8) Iterate through steps 4 to 7 several times. Five iterations tend to be sufficient.

(9) Use the improved ePSF model to re-measure the fluxes and centroids of stars. Evaluate the pixel-phase error in the flux and centroid measurements, if it is sufficiently small then the ePSF model is complete. If the pixel-phase error is still significant, continue with steps 10 and 11.

(10) Calculate the averaged fluxes and positions of the same stars observed in the dithered images to be used as un-biased measurements. The averaged positions must be calculated in a common reference frame and transformed back to the individual frame coordinates.

(11) Repeat steps 3 to 9.

3 EPSF MODELLING CODE IMPLEMENTATION

An open-source version of the ePSF modelling and fitting routine is currently maintained in the `photutils.psf` subpackage of Astropy in the `EPSFBuilder` and `EPSFFitter` classes (L. Bradley et al. 2024). This implementation has some differences in comparison to the original routine described in J. Anderson & I. R. King (2000). These differences and their implications for ePSF modelling and fitting accuracy are summarised below.

Iterations. In J. Anderson & I. R. King (2000) there are two levels of iteration in the ePSF-building routine. There is an overall iteration between measuring the ePSF and measuring the fluxes and positions of stars. Within each ePSF measurement there is an additional iteration between the adjustment of ePSF values based on the residuals, and the smoothing, re-centring, and normalization of the ePSF. This internal iteration step only iterates over the re-centring step in `EPSFBuilder`. However, this difference is not expected to affect the final ePSF model, since both approaches iteratively refine the ePSF and stellar parameters toward the same self-consistent solution. The `EPSFBuilder` routine may simply require a larger number of outer iterations for the ePSF model to converge to an equivalent result.

Re-centring In J. Anderson & I. R. King (2000), the ePSF is defined to be symmetric about its centre, and any offset is corrected by shifting the sampling grid so that the ePSF values at ± 0.5 pixel along each axis become equal. In `EPSFBuilder`, a peak-based centring algorithm is used instead.² In `EPSFBuilder`, this re-centring step is iterated over a small number of iterations. In practice, the definition of the ePSF centre is arbitrary provided it is applied consistently throughout the ePSF construction and fitting processes.

Normalisation In J. Anderson & I. R. King (2000), the ePSF is normalised such that a star with unit flux, centred exactly on the

²By default, this uses the `photutils.centroids.centroid_com` function, which determines the centroid as the intensity-weighted centre of mass.

middle of a detector pixel, produces an image whose summed pixel values equal one. This definition ensures that the ePSF represents the fraction of a star’s total light falling into each pixel as a function of its sub-pixel position. In contrast, there is no normalisation applied to the ePSF in `EPSFBuilder`.³ The normalisation of the ePSF is, in principle, arbitrary and only affects the overall flux scale. For applications using a single ePSF model, the lack of explicit normalisation simply results in a constant scaling of all measured fluxes, which has no effect on relative photometry or astrometry. However, when multiple ePSF models are used – for example, to capture the spatial variation in the ePSF across the image – the relative scaling of each ePSF model will differ. In such cases, normalisation is necessary to ensure consistent scaling of flux measurements between different ePSF models.

End-of-iteration requirement In J. Anderson & I. R. King (2000), the iterative process continues until the measured stellar fluxes and positions no longer exhibit significant pixel-phase-dependent errors. In contrast, in `EPSFBuilder` the iteration proceeds until either the user-specified maximum number of iterations is reached, or the measured source positions change by less than a defined tolerance between successive iterations, indicating that the ePSF model has converged. With this requirement, it is recommended to validate that the final ePSF removes pixel-phase biases in the final measured fluxes and positions.

Degeneracy-breaking In J. Anderson & I. R. King (2000), both the fluxes and positions of stars observed across multiple frames are averaged in order to break the degeneracy between the intrinsic stellar flux and the apparent flux variations caused by non-uniform intra-pixel sensitivity. In contrast, in `EPSFBuilder` only the stellar positions are averaged between iterations, while the fluxes are treated independently for each image. This difference in the degeneracy-breaking step reduces the accuracy of the resulting ePSF model, as the pixel-phase-dependent flux variations are not fully corrected. This is particularly significant for undersampled images which are more affected by intra-pixel sensitivity variations, as discussed further in Section 5.6.

In this work, we make a number of modifications to the `photutils.psf` subpackage to bring it more closely into alignment with the ePSF modelling routine described by J. Anderson & I. R. King (2000). We also implement several extensions that allow the ePSF modelling to be performed using different interpolation functions, gridpoint estimation methods, smoothing kernels, and ePSF fitting functions. These modifications provide additional flexibility to the ePSF modelling and fitting routines which allow for more accurate ePSFs to be generated, as discussed in Section 5. The changes and additions are described in detail in Appendix B. The modified code is available from the authors upon request and will subsequently be submitted for integration into the `photutils` package.

4 SIMULATING THE EPSF

This work uses simulated ePSFs and images to investigate the accuracy of the ePSF modelling described in Section 2.2, and of the resulting photometric and astrometric measurements. Synthetic

ePSFs are derived by integrating instrumental PSFs over a model for the intra-pixel sensitivity profile.

We model each detector pixel as a square collecting area with uniform sensitivity, separated by narrow gaps of zero response between adjacent pixels (a *flat with gaps* intra-pixel sensitivity profile). This geometry can be represented by a 2D top-hat function. We quantify the size of the gaps by the fill factor (FF), which is defined as the ratio of a pixel’s light sensitive area to its total area. With this model we can recover a uniform intra-pixel sensitivity profile (*flat and flush*) by using a full fill factor (FF = 1). This work aims to demonstrate the ability of the ePSF modelling procedure to fully account for a given non-uniform intra-pixel sensitivity profile in the ePSF model. Although simplified, the *flat with gaps* model is sufficient to demonstrate the principal effects of non-uniform intra-pixel sensitivity on the ePSF.

For simplicity, we adopted Gaussian forms for the instrumental PSF and explored three representative cases:

- (1) A simple circular Gaussian,
- (2) A rotated elliptical Gaussian, and
- (3) A skewed Gaussian.

³Although a normalisation method exists within the underlying `_LegacyEPSFModel` class used, it is never called in the `EPSFBuilder` class since the `_LegacyEPSFModel` class is implemented using its default parameter `normalize=False`.

The derivations of the synthetic ePSFs used to generate the simulated images are provided in Appendix A.

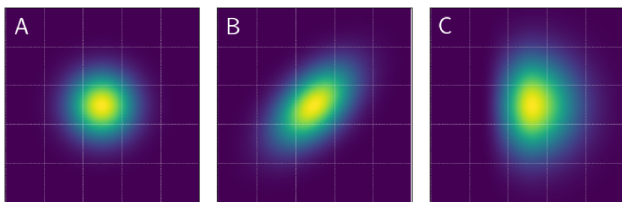


Figure 1. Synthetic ePSFs constructed by integrating a PSF model over ‘flush and flat’ intrapixel sensitivity profile. A: Circular Gaussian PSF with $\sigma = 0.5$ pixels. B: Rotated elliptical Gaussian PSF with $\sigma_x = 0.8$ pixels and $\sigma_y = 0.4$ pixels rotated by 45° . C: Skewed Gaussian with $\sigma_x = \sigma_y = 0.8$ and skew parameter $\eta = 5$. Dashed lines show pixel boundaries.

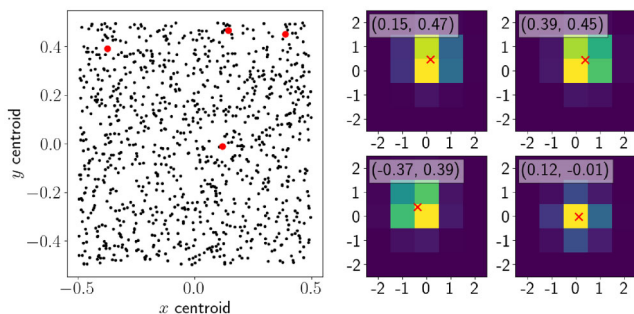


Figure 2. Left: Sub-pixel positions of 1000 random centroid coordinate samplings used to generate star images. Right: Examples of 5×5 pixel star images generated using synthetic ePSF A in Fig. 1, with the numbers indicating their input sub-pixel centroid positions.

Fig. 1 shows three synthetic ePSFs corresponding to the cases above, computed for pixels with a *flat and flush* sensitivity profile. Later in this work, we use synthetic ePSFs computed for pixels with small gaps between them in order to investigate the effect of non-uniform intra-pixel sensitivity profiles.

The synthetic ePSF is evaluated over a grid of pixels with a given centroid position and flux value to generate a star image. This is repeated for a random sample of centroid positions and fluxes to produce a simulated data set of star images as shown in Fig. 2. We can use these simulated stars to generate an empirical ePSF model and compare against the known input ePSF. In addition, we can then perform ePSF photometry on these simulated stars using the generated empirical ePSF model and compare the results with their input positions and fluxes. This simulated framework forms the basis for our analysis in Section 5.

5 ACCURACY OF EPSF MODELLING

For ePSF fitting photometry, the accuracy of the photometric measurements depends directly on how well the ePSF itself is modelled. Any imperfections in the model can introduce systematic pixel-phase errors in the measured fluxes and positions of stars. Constructing an accurate ePSF model, however, is a complex process influenced by multiple instrumental and (for ground-based telescopes) atmospheric factors, summarised below.

(1) **PSF sampling.** When the image is undersampled, meaning the light from a star falls on only a small number of pixels, each star provides limited information about the PSF shape. This makes the ePSF harder to constrain and increases systematic errors.

(2) **Intra-pixel sensitivity.** Non-uniform pixel response causes the measured fluxes and positions of stars to depend on their

sub-pixel location, introducing pixel-phase systematics in aperture photometry. These effects must be accounted for during ePSF modelling to prevent them from being imprinted on the ePSF model.

(3) **Geometric distortion.** This distortion alters the mapping between the true sky positions of stars and their measured detector coordinates. If uncorrected, it introduces systematic errors into the ePSF model. Accurate distortion correction is therefore essential for precise ePSF modelling (J. Anderson & I. R. King 2003, 2004).

(4) **Position dependence.** Optical effects can cause the PSF to vary across the detector. If these spatial variations are not included in the ePSF modelling, the resulting model will be inaccurate in different regions of the image. This can be mitigated by constructing a grid of ePSFs (J. Anderson & I. R. King 2000).

(5) **Brightness dependence.** In some detectors, particularly CCDs, charge from bright sources can repel subsequent charge, causing the ePSF to broaden for brighter stars (P. Antilogus et al. 2014). This ‘brighter-fatter’ effect means that a single ePSF model cannot accurately represent stars across all flux levels. This could be corrected by modelling the flux dependence of the ePSF or by generating separate ePSFs for different flux ranges.

(6) **Temporal variations.** The PSF can change over time due to telescope focus shifts, temperature fluctuations, mechanical flexure, or varying atmospheric conditions. If not tracked and modelled, these variations introduce inconsistencies into the ePSF, particularly in time-domain surveys or long observing campaigns. Temporal changes can be accounted for by applying small perturbations to the averaged ePSF model (J. Anderson & I. R. King 2006) or by using a phylogram-based technique (J. Anderson & L. R. Bedin 2017; A. Bellini et al. 2019).

In addition to instrumental and observational factors, the accuracy of the ePSF model depends on several user-defined parameters in the modelling routine that must be tuned to the specific implementation. The effect of each parameter on the accuracy of the ePSF model is examined in the following subsections.

In this work, we limit our analysis to the modelling of a *stationary* ePSF – one that does not vary with position, brightness, or time. While position-, flux-, and time-dependent variations are important to characterise for a complete description of an instrument’s behaviour, doing so first requires an accurate stationary ePSF. We therefore continue with this simplified assumption.

5.1 Oversampling

The oversampling parameter defines the number of gridpoints per pixel in the ePSF model. For an oversampling o , there will be o^2 gridpoints per pixel. The optimal oversampling to use is dependent on the shape and size of the ePSF, and on the sample data available to construct the model. If the oversampling is too small, fine structure details are lost and interpolation errors increase; if it is too high, there may be insufficient sample sources to accurately constrain the ePSF model. An oversampling of four is commonly used in the literature, following its use for the WFPC2 camera of the *HST* in J. Anderson & I. R. King (2000). This choice for the oversampling has been widely used across both space-based and ground-based instruments with diverse PSF characteristics (J. Anderson & I. R. King 2004, 2006; J. Anderson et al. 2006, 2015; M. Libralato et al. 2014; J. Anderson 2016; M. Libralato et al. 2016a, b, 2023, 2024a, b; D. Nardiello et al. 2019, 2022). However, it is unlikely that a fixed oversampling of four is optimal for all instruments, as the best oversampling should depend on the true ePSF morphology. The repeated use of this default value may be due to the difficulty

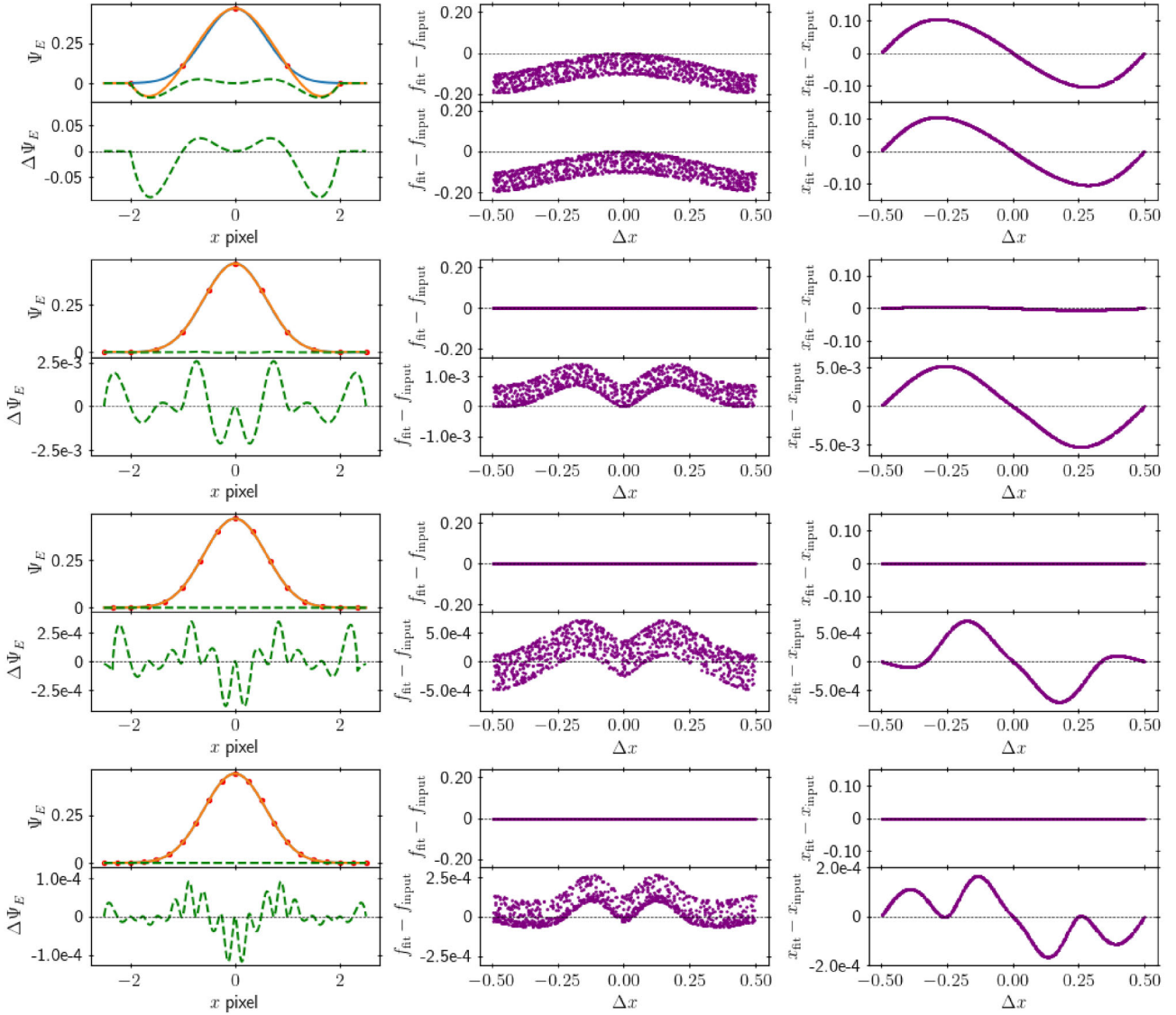


Figure 3. ePSF modelling and fitting accuracy of synthetic ePSF A, comparing different oversampling parameters. Rows top to bottom are: oversampling = 1, 2, 3, and 4. Left: True ePSF (blue) and interpolated ePSF model (orange) at $y = 0$; residuals ($\Delta\Psi_E$ = interpolated – true ePSF) are shown as green dashed line with a magnified view in the lower panel. Red points mark the tabulated ePSF values used in the interpolation. Middle: Pixel-phase bias in fitted fluxes of 1000 simulated stars. Δx is the x -displacement of the centroid of the star from the centre of a pixel. Since $f_{\text{input}} = 1$, the flux residual $\Delta f = f_{\text{fit}} - f_{\text{input}}$ represents the fractional flux error. Right: Pixel-phase bias of the fitted x centroid positions, with residuals $x_{\text{fit}} - x_{\text{input}}$ in pixel units.

of knowing the optimal oversampling without already knowing the detailed ePSF structure.

To isolate the effect of oversampling on ePSF accuracy, we assume perfect knowledge of the ePSF values at the gridpoints. Rather than estimating them from simulated star samples, we evaluate the ePSF directly at the $(\Delta x, \Delta y)$ of the gridpoints from the synthetic input ePSFs described in Section 4. The resulting ePSF model, reconstructed from these tabulated values, can then be compared directly to the true synthetic ePSF to assess the interpolation error. The ePSF model is then used to fit the fluxes and positions of 1000 simulated stars generated with the same synthetic input ePSF. Comparing the measured and true fluxes and positions allows us to quantify ePSF fitting accuracy and the scale of pixel-phase errors. We fix input fluxes to be unity, so the flux residual $f_{\text{fit}} - f_{\text{input}}$ is also the fractional error; position residuals are given in pixels as $x_{\text{fit}} - x_{\text{input}}$.

We first consider the simplest case – a circular Gaussian PSF and a *flat and flush* intrapixel sensitivity profile (ePSF Model A in Fig. 1). Fig. 3 compares oversampling parameters of $o = 1 - 4$. As anticipated, the interpolated model overfits for low oversampling parameters. In the lowest sampling case of $o = 1$, this leads to large pixel-phase errors, reaching amplitudes of up to ~ 20 per cent in flux and ~ 0.3 pixels in centroid position. Increasing the oversampling significantly reduces both interpolation artefacts and pixel-phase errors. We repeat these experiments for the other two synthetic ePSFs shown in Fig. 1 and for a wider range of oversampling parameters. The scale of the pixel-phase errors in the flux measurements is quantified using a 2D pixel-phase map of the median fractional flux residuals in $(\Delta x, \Delta y)$. Each cell of this pixel-phase map represents the median residual for stars whose position centroids fall within that sub-pixel region. The amplitude of the pixel-phase effect is then defined as the difference between

the maximum and minimum cell medians, providing a measure of the peak-to-peak systematic variation in the measured fluxes. The resulting pixel-phase amplitudes for the different oversampling parameters are shown in Fig. 4. Since we have assumed perfect ePSF values at the gridpoints, these measurements represent a lower limit to the pixel-phase errors expected in real data, where uncertainties in the gridpoint estimation introduce additional scatter and bias.

In the ePSF-building routine, the oversampling factor defines the number of gridpoints per detector pixel and is therefore independent of the intrinsic size of the ePSF. Consequently, for the same oversampling, a well-sampled ePSF that spans many pixels will be modelled more accurately than an undersampled one whose light is concentrated in fewer pixels. In practice, one should use the highest oversampling that the available sample of stars can reliably constrain (see Section 5.5 for the dependence on sample size); when the oversampling is limited by the number of star samples, a minimum acceptable oversampling can be estimated from the requirement to keep pixel-phase errors negligible. Our simulations show that for the approximately Gaussian synthetic ePSF (Model A), an oversampling factor of three results in photometric and astrometric pixel-phase errors on the order of 0.1 per cent. We measure the full-width at half-maximum (FWHM) of this ePSF to be about 1.35 pixels.⁴ This corresponds to roughly four gridpoints per FWHM. As a rule of thumb, we therefore recommend choosing an oversampling that provides at least four gridpoints per FWHM for near-Gaussian ePSFs. For non-circular ePSFs, the FWHM should be measured along the narrowest dimension, and for strongly asymmetric ePSFs, it should be estimated from the region with the steepest spatial gradient.

5.2 Interpolation

To fit the empirical ePSF model to a star image, the ePSF must be evaluated at positions between the tabulated gridpoints. This is typically achieved using a third-degree bivariate spline interpolator to estimate the ePSF value at arbitrary $(\Delta x, \Delta y)$ coordinates. As shown in the previous section, interpolation errors can introduce significant pixel-phase biases in the measured fluxes and positions, particularly when the ePSF is sampled with only a few gridpoints per FWHM. The interpolation method must therefore be flexible enough to capture genuine variations in the ePSF between gridpoints, but not so flexible that it overfits the tabulated values. We repeat the experiment from Fig. 4 for a range of alternative interpolation methods, summarised in Table 1. Fig. 5 shows a comparison of the amplitude of the pixel-phase errors on the flux measurements, estimated from the peak-to-peak flux residuals, for each interpolation method.

The two best-performing interpolation methods across all three synthetic ePSF models (A–C) are the third-degree bivariate spline (RBS $k = 3$) and the cubic radial basis function (RBF cubic). At low oversampling factors ($o = 1$ – 3), the Gaussian RBF also performs well, though its accuracy plateaus and ultimately degrades relative to other methods at higher oversampling ($o \geq 4$). For Models A and B, the RBS $k = 3$ interpolator performs poorly in the most undersampled case ($o = 1$), producing the largest pixel-phase errors of any method. In nearly all cases, the RBF cubic interpolator yields slightly smaller residuals than the RBS $k = 3$ interpolator, maintaining superior accuracy across a range of ePSF shapes and sampling levels. We therefore adopt the RBF cubic

interpolator for all subsequent analysis. This is implemented using the `RBFInterpolator` class from the SciPy library (P. Virtanen et al. 2020).

5.3 Gridpoint value estimation

In the previous section, we assumed perfect knowledge of the ePSF values at each gridpoint. In practice, these values are estimated from the ePSF samples within small regions around each gridpoint, hereafter referred to as *gridsections*. The standard approach is to take the sigma-clipped median of all ePSF samples that fall within each gridsection. The size of these regions is determined by the oversampling factor: for example, an oversampling of four divides each pixel into a 4×4 grid of sub-regions, with gridpoints located at their centres. Fig. 6 illustrates this geometry for a 3×3 pixel ePSF model with an oversampling of three.

The accuracy of the sigma-clipped median for estimating ePSF gridpoint values depends strongly on the distribution of samples within each gridsection and on the shape of the ePSF. The sigma-clipped median will result in a bad estimate for the gridpoint values if the samples are clustered in a particular region of the gridsection. Furthermore, this method will always underestimate the gridpoint values for a gridsection which contains a local extrema. This is the case for the central gridpoint which is placed at the peak of the ePSF, where the sigma-clipped median estimation will always be smaller than the true value.

Alternative methods are investigated for estimating the gridpoint values from the samples within the gridsections: sigma-clipped mean, sigma-clipped weighted mean, and a 2D polynomial surface fitting. For the weighted mean, samples are weighted by their distance from the gridpoint position. For the polynomial method, we fit a 2D polynomial surface over the samples within a gridsection. The value of the gridpoint is then estimated by evaluating the fitted surface at the gridpoint location. The polynomial used is either linear or quadratic depending on the number of sources within the gridsection. We generate 1000 simulated star images using the synthetic ePSF Model A to construct ePSF models with each of the gridpoint estimation methods. The known fluxes and centroid positions of the simulated stars are used so that differences in the resulting models arise solely from the gridpoint estimation technique. A second, independent set of 1000 simulated star images is then used to test the photometric accuracy of each ePSF model (Fig. 7). We repeat the experiment using only 100 sample stars for the ePSF modelling to examine how inhomogeneous sample distributions affects each method (Fig. 8). In both cases, the 2D polynomial surface-fitting method produces residuals more than an order of magnitude smaller than the alternatives, yielding significantly more accurate photometry. This approach estimates the central gridpoint value more accurately and is less sensitive to uneven sample distributions within the gridsections.

We extend this analysis to a range of oversampling factors and to all three synthetic ePSFs shown in Fig. 1. Fig. 9 compares the amplitude of the pixel-phase errors in the flux measurements of 1000 simulated stars for each input model, oversampling value, and gridpoint estimation method. Across all input ePSF models (A–C), the 2D polynomial surface-fitting method consistently yields the smallest pixel-phase errors for nearly all oversampling factors. The amplitude of these errors decreases by several orders of magnitude at higher oversampling, largely due to the polynomial method's ability to accurately estimate the ePSF at the central (peak) gridpoint, where other methods tend to underestimate the true value. At oversampling factors of $o \geq 3$, the residuals from all estimation methods exceed

⁴The FWHM of the ePSF is broader than that of the instrumental PSF due to the integration over pixel areas.

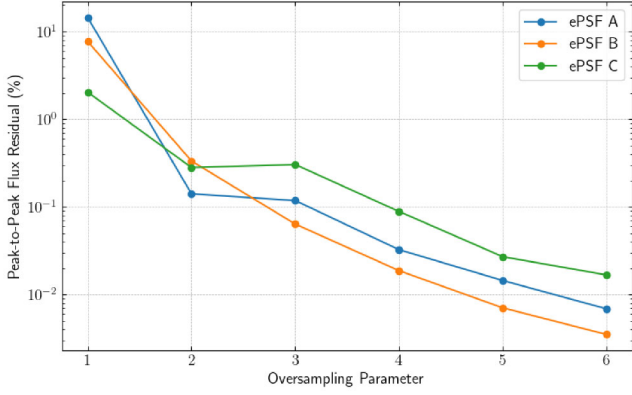


Figure 4. Pixel-phase error amplitude in flux measurements as a function of oversampling for the three synthetic ePSFs in Fig. 1. The amplitude is derived from the peak-to-peak variation in the median flux residuals across sub-pixel phase.

those expected from interpolation errors alone (cf. Fig. 5), indicating that gridpoint estimation uncertainty becomes the dominant source of error at these sampling levels. For the remainder of this work, the 2D polynomial surface-fitting method is used to determine the gridpoint values in the ePSF modelling.

5.4 Smoothing

As described in the previous section, the empirical ePSF model consists of estimated ePSF values at discrete gridpoints. In practice, these estimates are derived from noisy sample star images, introduc-

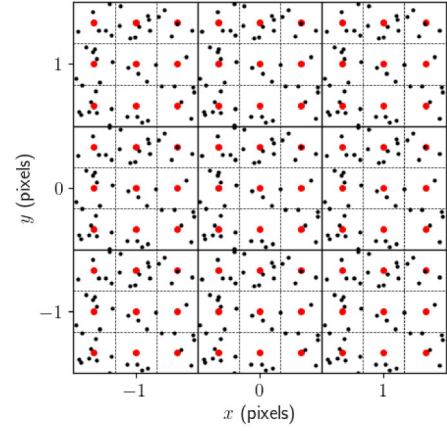


Figure 6. Demonstration of gridpoints and gridsections in the ePSF modelling for an oversampling factor of three. Solid black lines mark pixel boundaries, and dashed lines show the gridsection boundaries. Red points at the centres of gridsections indicate the gridpoints, and black points show the locations of the ePSF samples from 50 simulated star images (each 3×3 pixels) with randomly distributed sub-pixel centroids.

ing noise into the gridpoint values. To mitigate this, J. Anderson & I. R. King (2000) imposed a smoothness constraint on the ePSF model by convolving it with a smoothing kernel. In their implementation, with an oversampling factor of four, the kernel is a 5×5 least-

Table 1. Labels and descriptions of interpolator methods tested for the interpolation of the ePSF model between gridpoint values. These are implemented in Python using the SciPy library (P. Virtanen et al. 2020).

Interpolator label	Description	Kernel function
RBS $k = 2$	Second degree bivariate spline approximation.	–
RBS $k = 3$	Third degree bivariate spline approximation.	–
RBF thin_plate_spline	Radial basis function interpolation with a thin plate spline kernel.	$\phi(r) = r^2 \ln(r)$
RBF cubic	Radial basis function interpolation with a cubic kernel.	$\phi(r) = r^3$
RBF multiquadric $\epsilon = 1.5$	Radial basis function interpolation with a multiquadric kernel with $\epsilon = 1.5$.	$\phi(r) = \sqrt{(r/\epsilon)^2 + 1}$
RBF inverse_multiquadric $\epsilon = 1.0$	Radial basis function interpolation with an inverse multiquadric kernel with $\epsilon = 1.0$.	$\phi(r) = \frac{1}{\sqrt{(r/\epsilon)^2 + 1}}$
RBF inverse_quadratic $\epsilon = 1.0$	Radial basis function interpolation with an inverse quadratic kernel with $\epsilon = 1.0$.	$\phi(r) = \frac{1}{(r/\epsilon)^2 + 1}$
RBF Gaussian $\epsilon = 1.0$	Radial basis function interpolation with a Gaussian kernel with $\epsilon = 1.0$.	$\phi(r) = \exp(-(r/\epsilon)^2)$

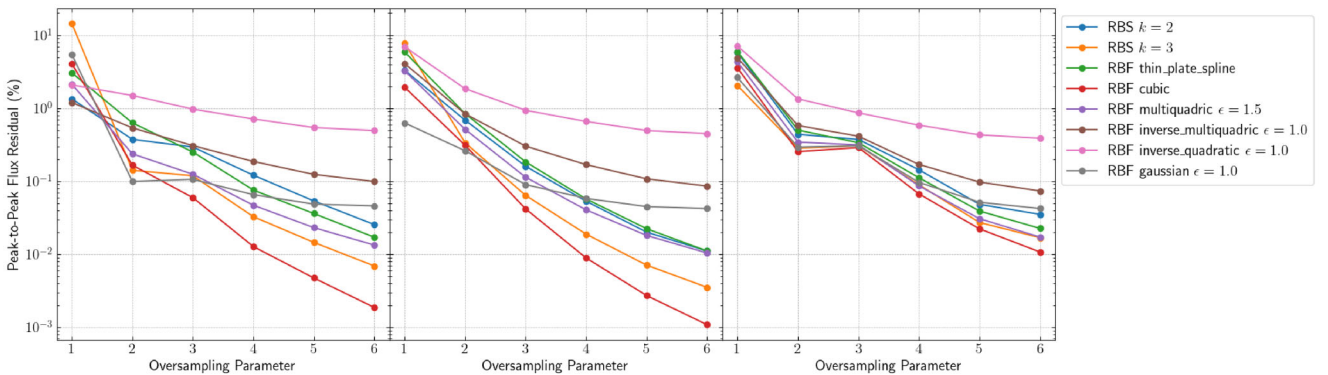


Figure 5. Comparison of different interpolation methods for ePSF fitting photometry with the synthetic ePSF models A (left), B (middle), and C (right) shown in Fig. 1, across a range of oversampling values. Each circular point represents the pixel-phase error amplitude in the flux measurements of 1000 simulated sources, for the given oversampling, interpolator, and input model.

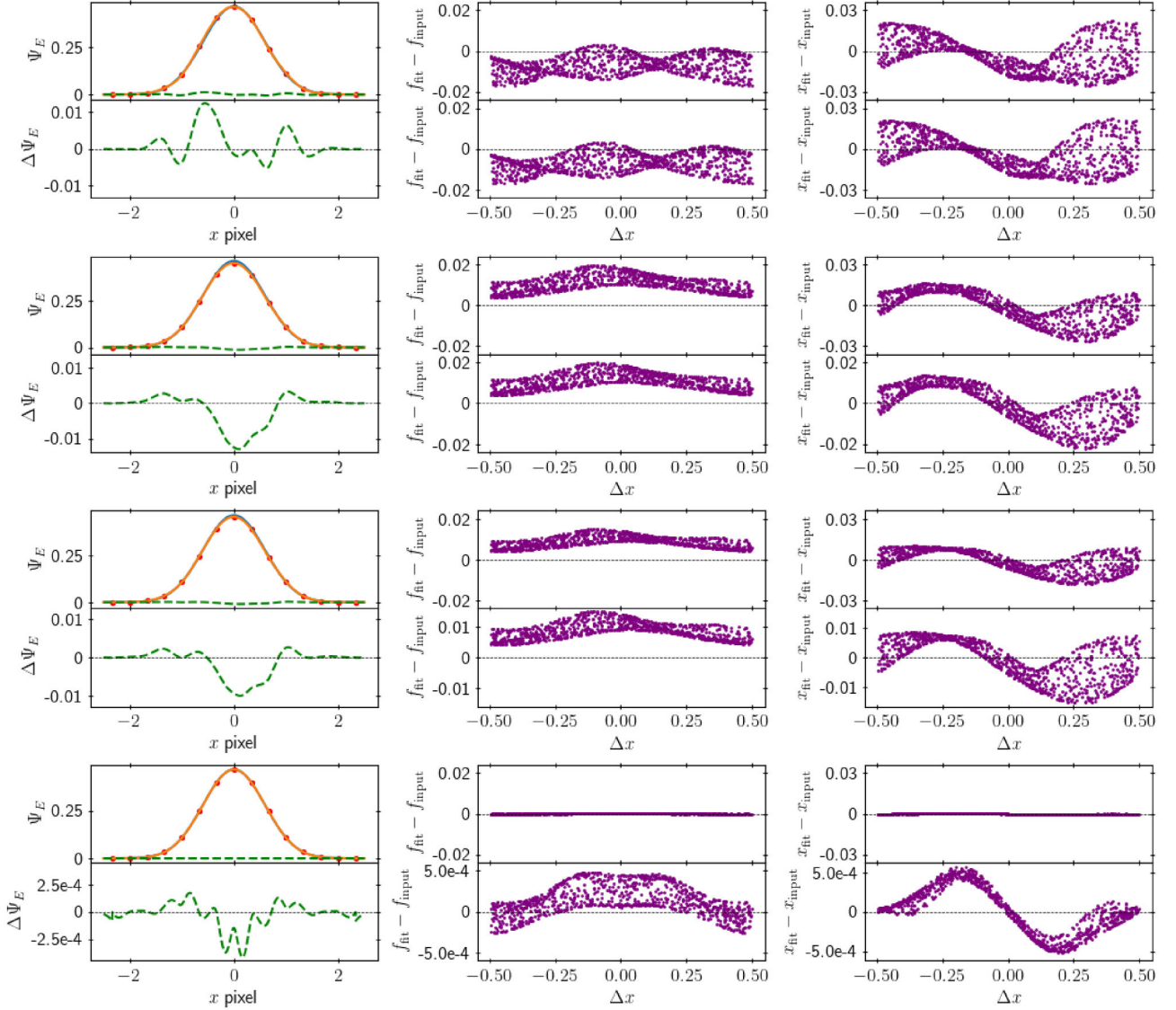


Figure 7. ePSF modelling and fitting accuracy of input ePSF Model A analogous to Fig. 3. We model the ePSF with 1000 sample stars, use an oversampling parameter of 3, and use the RBF cubic interpolator method. Here we compare different methods for the gridpoint value estimation, where rows from top to bottom are: sigma-clipped median, sigma-clipped mean, sigma-clipped weighted mean, and 2D polynomial surface fitting.

squares quartic kernel with coefficients:

$$\begin{bmatrix}
 0.041632 & -0.080816 & -0.078368 & -0.081816 & 0.041632 \\
 -0.080816 & -0.019592 & 0.200816 & -0.019592 & -0.080816 \\
 0.078368 & 0.200816 & 0.441632 & 0.200816 & 0.078368 \\
 -0.080816 & -0.019592 & 0.200816 & -0.019592 & -0.080816 \\
 0.041632 & -0.080816 & -0.078368 & -0.081816 & 0.041632
 \end{bmatrix}. \quad (6)$$

The appropriate smoothing kernel depends on the oversampling factor (number of gridpoints per pixel), the intrinsic size and shape of the ePSF, and the level of noise in the gridpoint values. The kernel adopted by J. Anderson & I. R. King (2000) may not be optimal for all instruments or data qualities, so alternative kernels should be explored. In this section, we investigate how kernels of different sizes and polynomial degrees affect the accuracy of the ePSF modelling and the resulting ePSF-fitting photometry.

We construct smoothing kernels by fitting a 2D polynomial to an array of the desired kernel size, where the central element is

set to one and all others are set to zero. To ensure a well-defined centre, the kernel dimensions are required to be odd. The resulting kernel is convolved with the empirical ePSF model, which is defined in oversampled coordinates. Consequently, the physical scale of the kernel relative to the detector pixels depends on the ePSF oversampling factor – for example, a 7×7 kernel applied to an ePSF with an oversampling of seven corresponds to a kernel scale of one pixel, however the same kernel applied to an ePSF with an oversampling of three corresponds to a kernel scale of 2.33 pixels.

We generate ePSF models for the synthetic ePSF Model A from simulated star images following the procedure described in the previous section. To simulate realistic observational conditions, photon shot noise is added to the star images. For each pixel, the observed value is drawn from a Poisson distribution with a mean equal to the noiseless model intensity, representing the statistical uncertainty in photon counting. For bright sources, this is well approximated by adding Gaussian noise with a standard deviation equal to the square root of the pixel value. We set the noiseless

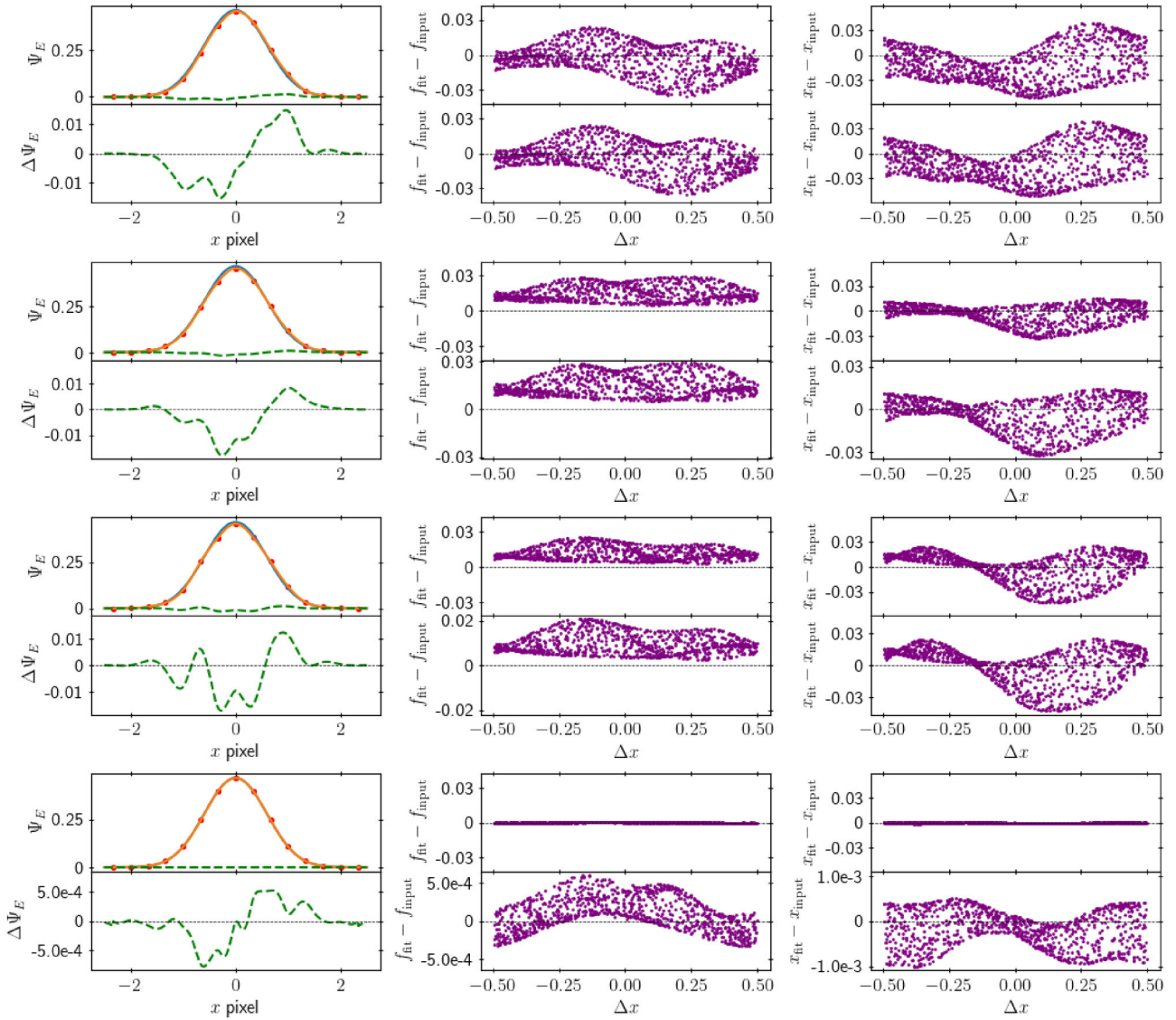


Figure 8. ePSF modelling and fitting accuracy of input ePSF Model A analogous to Fig. 3. We model the ePSF with 100 sample stars, use an oversampling parameter of 3, and use the RBF cubic interpolator method. Here we compare different methods for the gridpoint value estimation, where rows from top to bottom are: sigma-clipped median, sigma-clipped mean, sigma-clipped weighted mean, and 2D polynomial surface fitting.

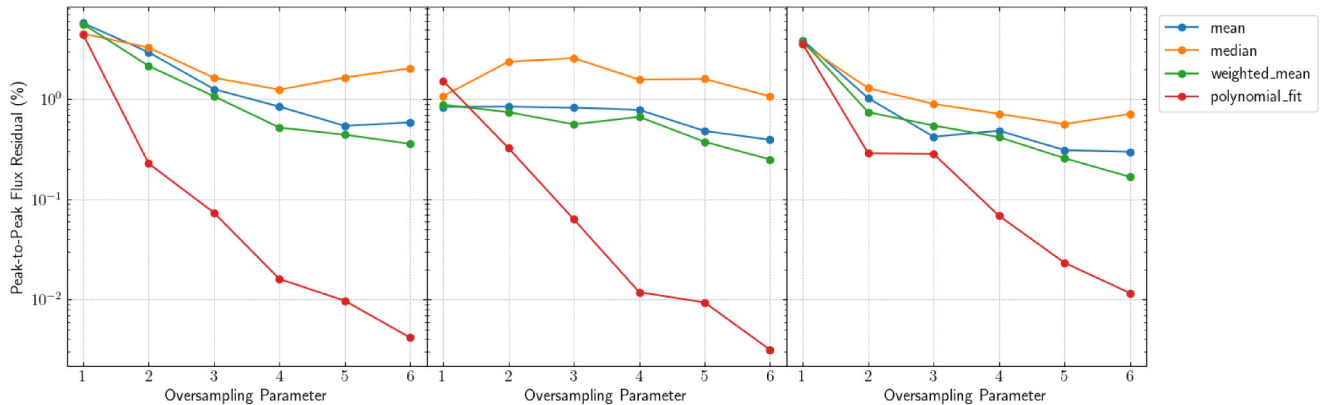


Figure 9. Comparison of different gridpoint estimation methods for ePSF fitting photometry for synthetic ePSF models A (left), B (middle), and C (right) as shown in Fig. 1, across a range of oversampling values. Each circular point represents the pixel-phase error amplitude in the flux measurements of 1000 simulated sources, for the given oversampling, gridpoint estimation method, and input model.

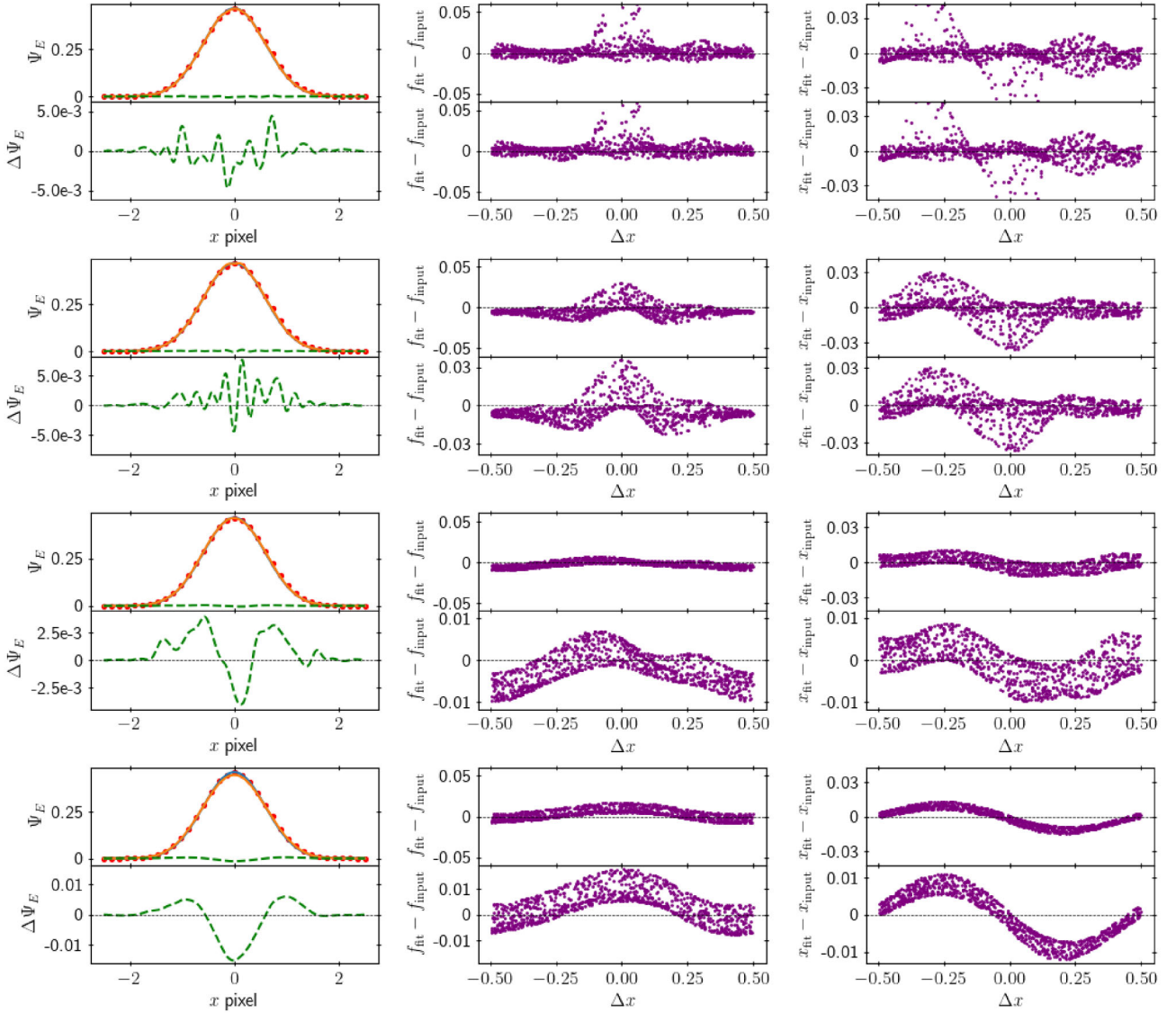


Figure 10. ePSF modelling and fitting accuracy of input ePSF Model A analogous to Fig. 3. Here we fix the oversampling to seven samples per pixel and compare the ePSF models with different smoothing kernels. Rows from top to bottom: no smoothing kernel, 5×5 quartic kernel, 9×9 quartic kernel, and 13×13 quartic kernel.

intensity of each star to be 10^5 counts. A constant background of 100 counts per pixel and Gaussian read noise with an RMS of 5 counts were also included. We use a smaller sample of 250 star images in this experiment to reproduce conditions where the noise on star images results in noisy gridpoint estimations. The resulting noisy star images are then used to construct the empirical ePSF models using the different smoothing kernels tested in this section. The independent set of 1000 simulated stars used to evaluate the ePSF-fitting accuracy is kept noise-free, ensuring that any differences in the recovered fluxes arise solely from imperfections in the ePSF model rather than noise in the fitting sample.

Fig. 10 shows the effect of the smoothing kernel size on the ePSF modelling, compared with an un-smoothed model. Each kernel is constructed using a quartic polynomial. The 5×5 kernel produces residuals of similar amplitude to the un-smoothed case, indicating that a kernel of this size is insufficient to suppress the random noise in the gridpoints. Conversely excessively large kernels (13×13) oversmooth the ePSF, erasing real small-scale structure in the model

which results in large errors on the ePSF model and large pixel-phase errors.

Next, we investigate the effect of the polynomial degree used to generate the smoothing kernel. Fig. 11 shows results from a similar experiment to that in Fig. 10, but with the kernel size fixed at 7×7 (equivalent to 1×1 pixels for $o = 7$). We compare quadratic, cubic, and quartic polynomial kernels. In this case, the three kernels produce ePSF models with comparable residuals and pixel-phase errors. Both the quadratic and cubic models show evidence of oversmoothing, whereas the quartic kernel introduces oscillatory artefacts.

When the smoothing kernel is too large, it flattens the ePSF model and introduces significant pixel-phase errors in the photometry and astrometry. This effect is most pronounced for models with small oversampling factors, where the kernel spans a larger fraction of a pixel (e.g. a 7×7 kernel at $o = 3$ corresponds to 2.3×2.3 pixels). These experiments demonstrate that smoothing must be chosen with care. The optimal kernel depends on the oversampling factor, and on

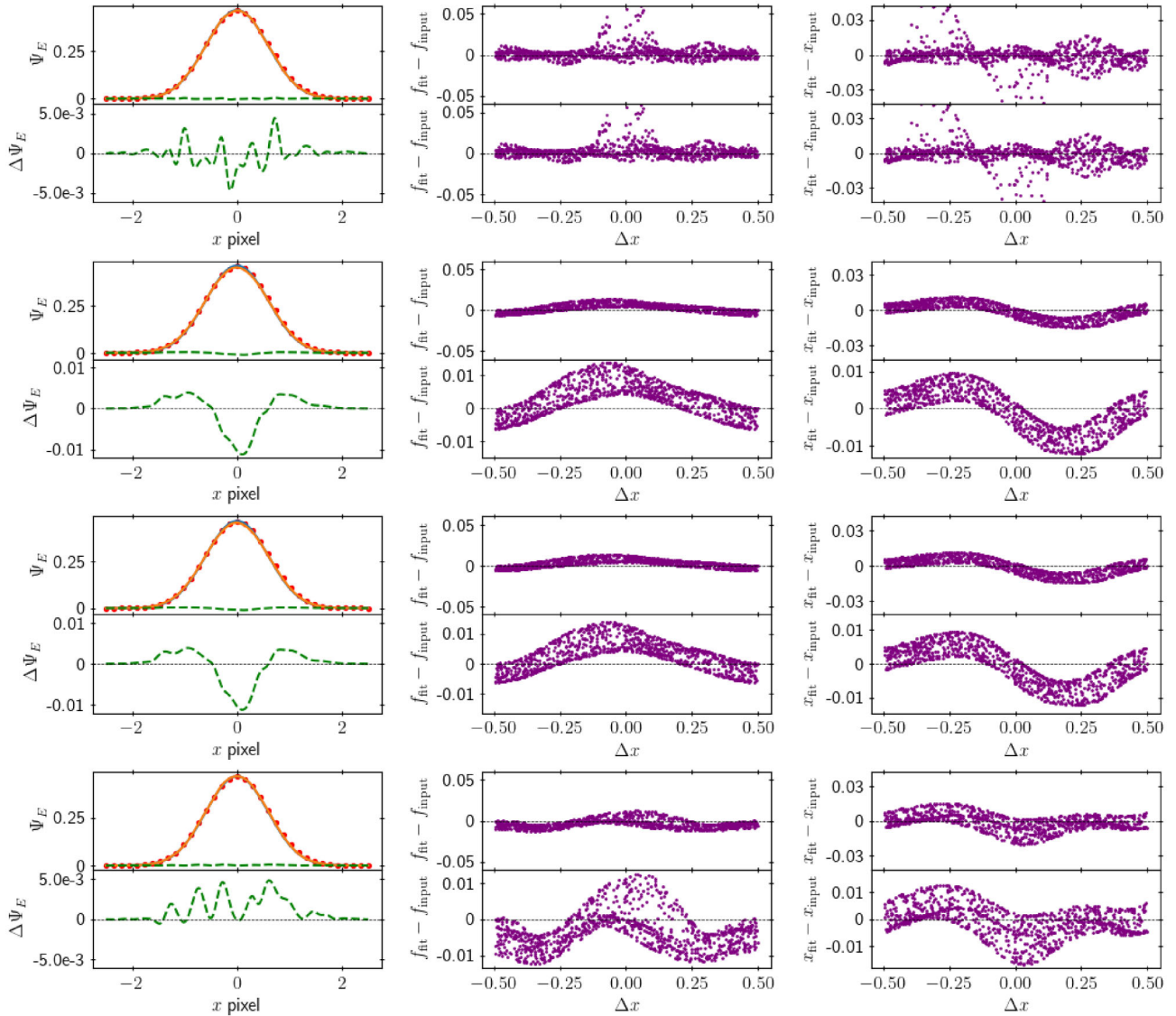


Figure 11. ePSF modelling and fitting accuracy of input ePSF Model A analogous to Fig. 3. Here we fix the oversampling to seven and compare the ePSF models with different smoothing kernels. Rows from top to bottom: no smoothing kernel, 7×7 quadratic kernel, 7×7 cubic kernel, and 7×7 quartic kernel.

the ePSF’s intrinsic size and structure. For a moderate oversampling value (≥ 3), a range of smoothing kernel sizes should be tested in order to find the minimum smoothing kernel which removes the smoothing noise but does not cause oversmoothing problems. In the case of undersampled ePSF with smaller oversampling values, one should avoid smoothing unless there are large amounts of noise in the ePSF gridpoints.

5.5 Number of star samples

In practice, one of the main limitations in ePSF modelling is the number of stellar samples available to constrain the model. The required sample size depends on how finely we want to supersample the ePSF (the oversampling factor), since a higher oversampling – and therefore a larger number of gridpoints – requires more samples. For the WFPC2 ePSF, J. Anderson & I. R. King (2000) used 45 000 star samples (3000 per image across 15 images). To account for spatial variation across the field, they constructed nine independent ePSF models at evenly spaced detector positions, each based on

roughly 5000 stars. These 5000 stars were plenty sufficient to estimate the ePSF at the 16 gridpoints per pixel for an oversampling of four. In contrast, ground-based applications of the ePSF routine use far fewer stars – typically 50–150 for the same oversampling factor (J. Anderson et al. 2006). There is currently no clear guideline for how many star samples are required to achieve a given level of ePSF accuracy for a specific oversampling value.

It is clear that a sufficient number and distribution of stars is required to ensure that each gridsection contains enough ePSF samples to estimate the gridpoint values using the methods described in Section 5.3. One way to achieve a well-distributed sampling is through a precisely dithered set of exposures, in which the sub-pixel centroids of stars are systematically shifted to cover all gridsections. Dithered images are also essential for mitigating the effects of intrapixel sensitivity variations on the ePSF model, as discussed further in Section 5.7. However, in many practical cases such a data set may not be available, and we must instead rely on the random distribution of stars to provide an adequate set of ePSF samples.

Table 2. Minimum number of star samples, N , required to achieve the given probability of having M samples per gridsection. Number of gridsections is determined by the oversampling as o^2 .

Oversampling	Probability	$M = 1$	$M = 3$	$M = 6$	$M = 10$
2	70 per cent	8	21	36	56
	80 per cent	10	22	39	59
	90 per cent	12	25	42	63
	95 per cent	15	29	48	68
3	70 per cent	28	56	97	142
	80 per cent	32	63	102	150
	90 per cent	39	72	112	164
	95 per cent	44	78	121	172
4	70 per cent	60	116	187	274
	80 per cent	68	125	199	288
	90 per cent	77	139	219	311
	95 per cent	88	156	238	327
5	70 per cent	105	196	311	454
	80 per cent	119	210	328	472
	90 per cent	136	237	358	501
	95 per cent	163	265	383	539

We perform a series of Monte Carlo simulations to determine how many randomly distributed star samples are required so that the probability of having at least M ePSF samples per gridsection exceeds 70, 80, 90, or 95 per cent for a given oversampling factor. The results for various M and oversampling values are summarised in Table 2. For the commonly used oversampling of four, we find that about 240 star samples provide a 95 per cent probability of having at least six ePSF samples per gridsection – the minimum needed to fit a 2D quadratic surface in the gridpoint estimation. Conversely, using fewer than about 60 stars reduces the probability of having even a single ePSF sample per gridsection below 70 per cent. When no ePSF samples fall within a gridsection, the gridpoint value is estimated from surrounding points using either 2D cubic spline or nearest-neighbour interpolation. This will add an additional interpolation error to the ePSF modelling which should be taken into account when using the model for photometric measurements.

We test the effect of the star sample number, N , on the accuracy of the ePSF model generated following the experiments in Section 5.3. Fig. 12 shows the results for the synthetic ePSF Model A, using an oversampling of three, with randomly distributed star samples of 10, 50, 100, and 500. The corresponding sub-pixel distributions of centroid positions of the stars are shown in Fig. 13. The distribution of the 10 samples is very inhomogeneous across the sub-pixel space, meaning some gridpoints are evaluated by just one sample, or must be interpolated from neighbouring points. If a gridpoint is evaluated by just one sample, the single sample could be from anywhere within the $1/3 \times 1/3$ pixel gridsection. The estimated ePSF values often deviate substantially from the true values, producing large residuals and significant pixel-phase errors in both flux and position. Increasing the sample to 50 stars reduces these errors but still leaves some undersampled regions, as evident in Fig. 13 (e.g. the middle-left gridsection contains only two samples). For $N = 100$ and $N = 500$, the sample distribution becomes more uniform, resulting in much more accurate ePSF reconstructions. The corresponding photometric and astrometric pixel-phase errors are several orders of magnitude smaller than in the cases with fewer star sampling, confirming that an adequate and homogeneous sample is essential for reliable ePSF modelling.

We perform an experiment to estimate the minimum number of star samples required to accurately model the ePSF for a given

oversampling factor. For the synthetic ePSF model A in Fig. 1, we generate ePSF models from simulated star samples while varying the number of stars between 10 and 550, repeating this for oversampling values of $o = 2-5$. As in previous sections, we calculate the amplitude of the pixel-phase errors in the flux measurements from ePSF-fitting photometry of 1000 simulated stars, as shown in Fig. 14. For each oversampling factor, the pixel-phase error amplitude converges to a minimum when the number of stars approximately matches that predicted for a 95 per cent probability of having at least $M = 6$ samples per gridsection (Table 2). For larger oversampling factors than those tested here, a similar Monte Carlo procedure can be used to estimate the number of stars needed for accurate ePSF modelling. If precisely dithered images are available, this will reduce the total number of stars required to generate an accurate ePSF model.

5.6 Degeneracy breaking

So far, we have not been considering the effect of intra-pixel sensitivity variations on the ePSF since we have been using synthetic ePSF models with a *flat and flush* intra-pixel sensitivity profile. As shown previous sections, pixel-phase bias errors arise from having errors in the ePSF model, regardless of whether intra-pixel sensitivity variations are present or not. However, if intra-pixel sensitivity variations present, we will always end up with an incorrect estimate of the ePSF if we do not properly consider the degeneracy between the flux of stars and their sub-pixel centroid position.

The response of a detector pixel to a photon depends on the exact location within the pixel where photon falls, for both CCD (T. R. Lauer 1999) and CMOS (S. B. Mahato et al. 2018) sensors. This means the measured pixel values in a stellar image are dependent on both the true flux of the star and the positions of the pixels relative to the centroid of the star. To demonstrate this, Fig. 15 shows simulated star images with identical input fluxes but different sub-pixel centroid positions. The input synthetic ePSF model used here has a circular Gaussian PSF ($\sigma = 0.4$ pixels) and a *flat with gaps* intra-pixel sensitivity profile (FF = 0.64) which produces a non-uniform sensitivity that varies with sub-pixel position. The image generated for a star centred at the middle of a pixel has a higher aperture sum flux than the image of the star centred at the corner of a pixel, despite both images being generated with identical input fluxes. This difference in observed flux is entirely due to the non-uniform intra-pixel sensitivity.

If the stellar position and flux are measured directly from the pixel values (e.g. with centre-of-mass centroiding and aperture photometry), the results will depend on the star’s sub-pixel position. To demonstrate this, we generate 1000 star simulated images with the same input flux but at random sub-pixel positions for ePSFs with both *flat with gaps* (FF = 0.64) and *flat and flush* (FF = 1) intra-pixel sensitivity profiles. The centroid and flux of the star images are then measured with centre-of-mass centroiding and aperture photometry respectively. Fig. 16 shows the resulting pixel-phase bias in the centroid and flux measurements of the simulated star images. A clear systematic trend is seen for the *flat with gaps* model in both flux and centroid position. The *flat and flush* model shows no flux bias, but does show a trend in the centroid position measurements due to inaccuracies in the centre-of-mass method for undersampled images, however this is smaller than for the *flat with gaps* model.

Intra-pixel sensitivity variations affect our ability to construct the ePSF using the routine presented in J. Anderson & I. R. King

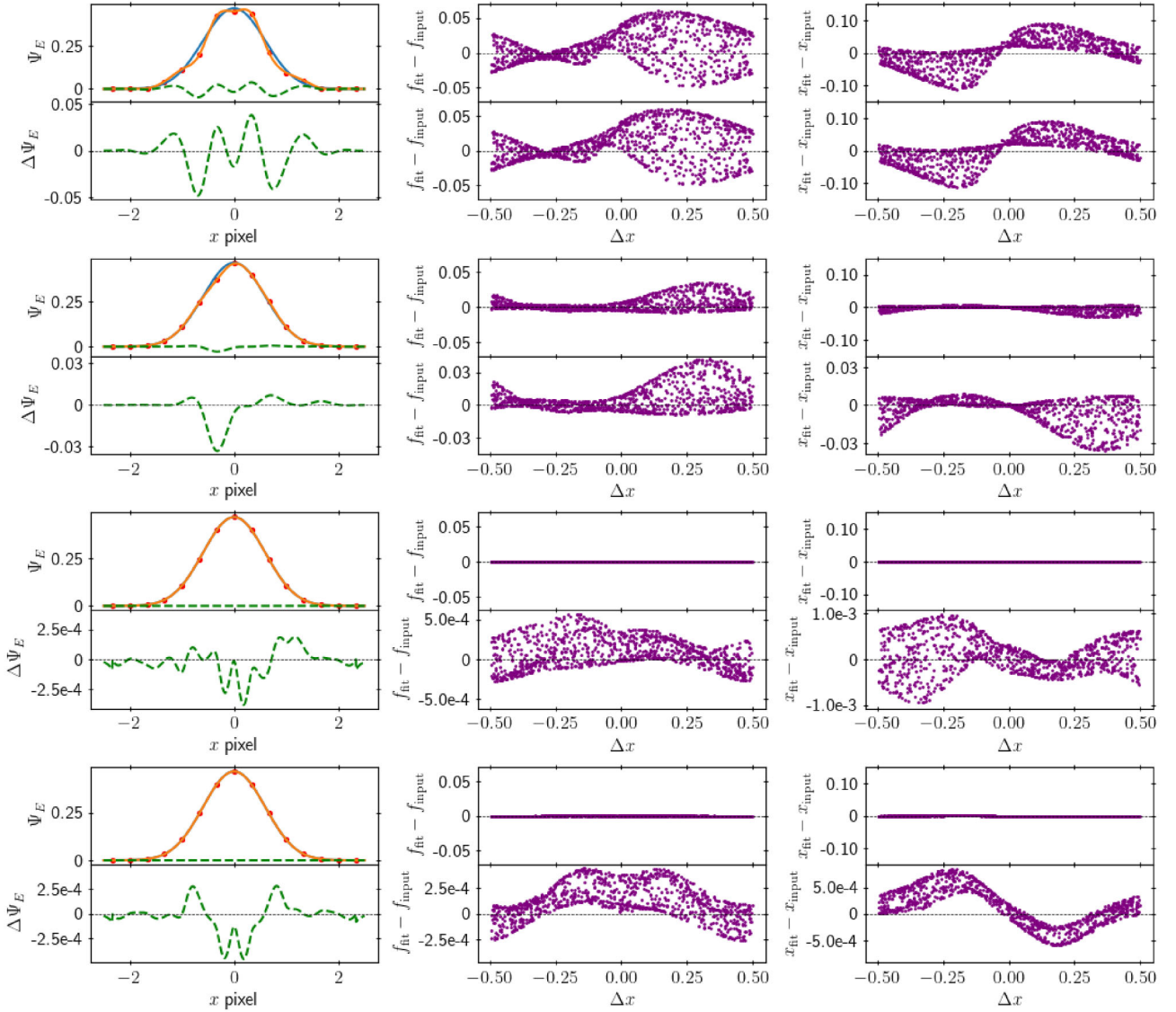


Figure 12. ePSF modelling and fitting accuracy of input ePSF Model A analogous to Fig. 3. We fix the oversampling parameter to $o = 3$ and test the effect of the number of star samples. Rows top to bottom are: 10, 50, 100, and 500 samples.

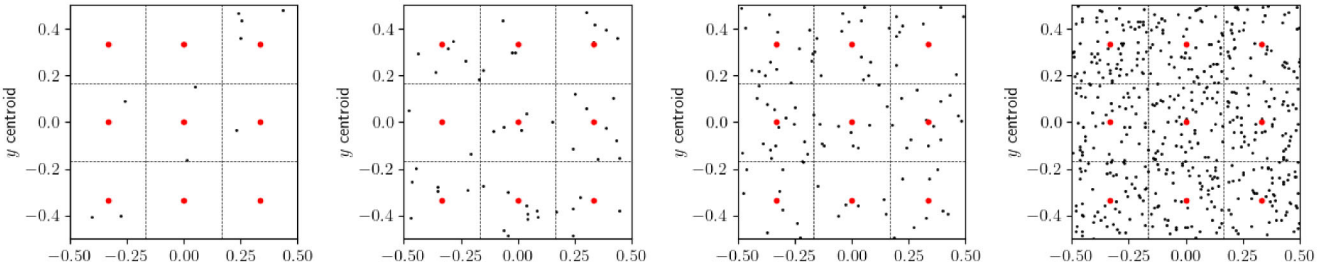


Figure 13. Distribution of sub-pixel centroid positions for different numbers of randomly distributed stars. Black points show the sub-pixel centroid positions. From left to right: 10, 50, 100 and 500 stars. Red points show the positions of gridpoints and dashed lines show the gridsection boundaries for an oversampling parameter of $o = 3$.

(2000) (as described in Section 2.2). The first step in this process is to estimate the fluxes and positions of stars, for this we use a centre-of-mass centroid and aperture photometry. The initial flux measurement (f_{obs}^1) for a star with a true centroid position of (x_*, y_*)

is given by:

$$f_{\text{obs}}^1 = f_* \times \sum_{i,j} \Psi_E^T(i - x_*, j - y_*), \quad (7)$$

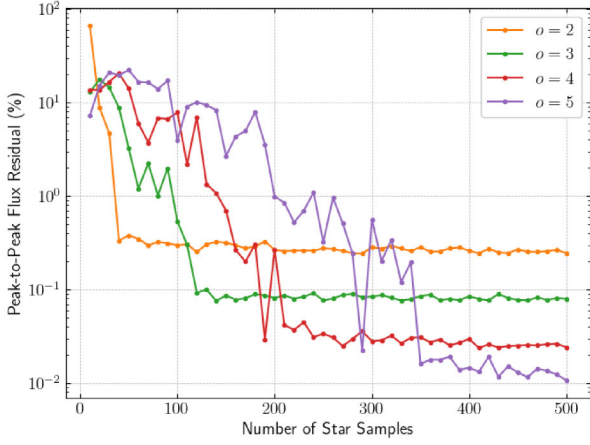


Figure 14. Comparison of the effect of the number of sample stars used in the ePSF modelling on the ePSF fitting photometry results for synthetic ePSF Model A shown in Fig. 1. Each circular point represents the pixel-phaser error amplitude in the flux measurements of 1000 simulated stars for the given oversampling and number of sample stars.

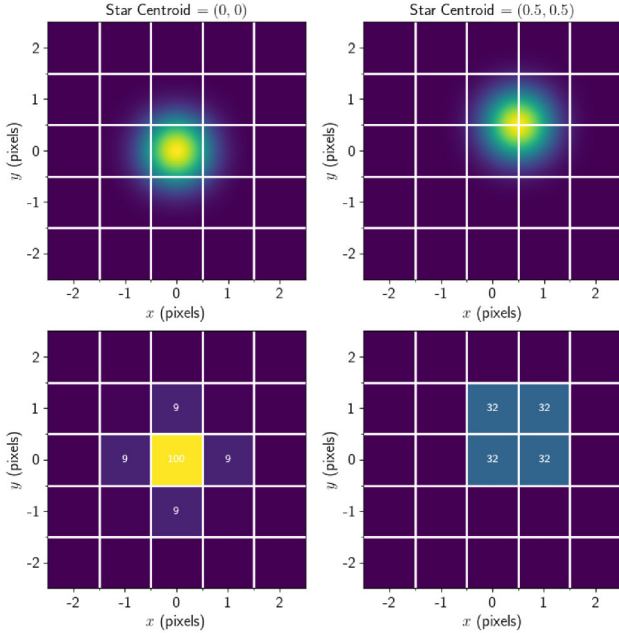


Figure 15. Top: The ePSF input model for a circular Gaussian PSF with $\sigma = 0.4$ and a *flat with gaps* intra-pixel sensitivity profile (FF = 0.64). The model is evaluated for a star centred on a pixel (left) and for one centred at a pixel corner (right), both with the same input stellar flux. Bottom: Simulated stellar images corresponding to the two cases above. Solid lines indicate pixel boundaries, and numbers within the pixels show the simulated counts.

where f_* is the true flux of the star, Ψ_E^T is the true ePSF of the instrument, and i, j are integer values denoting which pixels are included in the aperture sum. The centre-of-mass centroid position, $(x_{\text{obs}}^1, y_{\text{obs}}^1)$, is given by:

$$\begin{aligned} x_{\text{obs}}^1 &= \frac{\sum_{i,j} i \times \Psi_E^T(i - x_*, j - y_*)}{\sum_{i,j} \Psi_E^T(i - x_*, j - y_*)}, \\ y_{\text{obs}}^1 &= \frac{\sum_{i,j} j \times \Psi_E^T(i - x_*, j - y_*)}{\sum_{i,j} \Psi_E^T(i - x_*, j - y_*)}. \end{aligned} \quad (8)$$

The next step is to use the flux and centroid estimates of stars and the individual pixel values, P , within the star images to construct a first model of the ePSF (Ψ_E^1) using the following relation:

$$\Psi_E^1(i - x_{\text{obs}}^1, j - y_{\text{obs}}^1) = \frac{P(i - x_{\text{obs}}^1, j - y_{\text{obs}}^1)}{f_{\text{obs}}^1}. \quad (9)$$

The individual pixel values are determined by the true properties of the star and true ePSF, so we can re-write the above as:

$$\Psi_E^1(i - x_{\text{obs}}^1, j - y_{\text{obs}}^1) = \frac{f_* \Psi_E^T(i - x_*, j - y_*)}{f_{\text{obs}}^1}. \quad (10)$$

This ePSF model contains errors relating to both the errors on the centroid and flux estimates. Using equation (7) for f_{obs}^1 and simplifying the notations for $(i - x_*, j - y_*)$ and $(i - x_{\text{obs}}^1, j - y_{\text{obs}}^1)$ as $(\Delta x_*, \Delta y_*)$ and $(\Delta x_{\text{obs}}^1, \Delta y_{\text{obs}}^1)$ respectively, we can write down the ratio between the model and true ePSF as:

$$\frac{\Psi_E^1(\Delta x_{\text{obs}}^1, \Delta y_{\text{obs}}^1)}{\Psi_E^T(\Delta x_{\text{obs}}^1, \Delta y_{\text{obs}}^1)} = \frac{\Psi_E^T(\Delta x_*, \Delta y_*)}{\Psi_E^T(\Delta x_{\text{obs}}^1, \Delta y_{\text{obs}}^1) \sum_{i,j} \Psi_E^T(\Delta x_*, \Delta y_*)}. \quad (11)$$

The next step is to use this preliminary ePSF estimate to re-measure the stellar fluxes and positions through an iterative least-squares fitting procedure. Because this process is iterative, it is difficult to express the updated measurements of flux and position explicitly in terms of the true stellar values and the ePSF. To illustrate the effect of intra-pixel sensitivity variations on the ePSF modelling routine more clearly, we therefore simplify the problem by assuming that the true stellar positions are known, allowing equation (11) to be written as:

$$\frac{\Psi_E^1(\Delta x, \Delta y)}{\Psi_E^T(\Delta x, \Delta y)} = \frac{1}{\sum_{i,j} \Psi_E^T(\Delta x, \Delta y)}, \quad (12)$$

where we now have $\Delta x = \Delta x_* = \Delta x_{\text{obs}}^1$ and $\Delta y = \Delta y_* = \Delta y_{\text{obs}}^1$.

This simplification removes the need for a least-squares fitting procedure to measure the stellar flux with the ePSF. Instead, the flux is simply the scaling factor between the pixel values of the stellar image and the corresponding ePSF values at the same pixel displacements. The measured flux for a star at position (x_*, y_*) can therefore be written as:

$$f_{\text{obs}}^2 = \frac{P(\Delta x, \Delta y)}{\Psi_E^1(\Delta x, \Delta y)} = f_* \Psi_E^T(\Delta x, \Delta y) \times \frac{\sum_{i,j} \Psi_E^T(\Delta x, \Delta y)}{\Psi_E^T(\Delta x, \Delta y)} = f_{\text{obs}}^1, \quad (13)$$

where $(\Delta x, \Delta y)$ can be any $(i - x_*, j - y_*)$ for any (i, j) pixel position within the star's image. We find that the recovered fluxes are consistent with those measured by aperture photometry. Any pixel-phase errors present in the initial flux measurements are propagated into the ePSF-derived fluxes through the corresponding errors introduced into the ePSF model. If we had not assumed the stellar positions were known in equation (12), additional pixel-phase dependent errors would arise through inaccuracies in the position measurements, further amplifying the pixel-phase biases in both the ePSF-derived fluxes and centroids.

As described in J. Anderson & I. R. King (2000), the issue with the ePSF modelling is due to the degeneracy between the flux of a star and its sub-pixel centroid position. They demonstrate that this degeneracy can be broken using dithered observations of the same stars. Instead of using the measured fluxes and positions of each individual star image to measure the ePSF, the fluxes and positions of each star are fixed to be their average across all dithered images – this is the key degeneracy breaking step highlighted in J. Anderson & I. R. King (2000).

Continuing under the assumption the true positions of stars are known, the ePSF model constructed using stars with this fixed

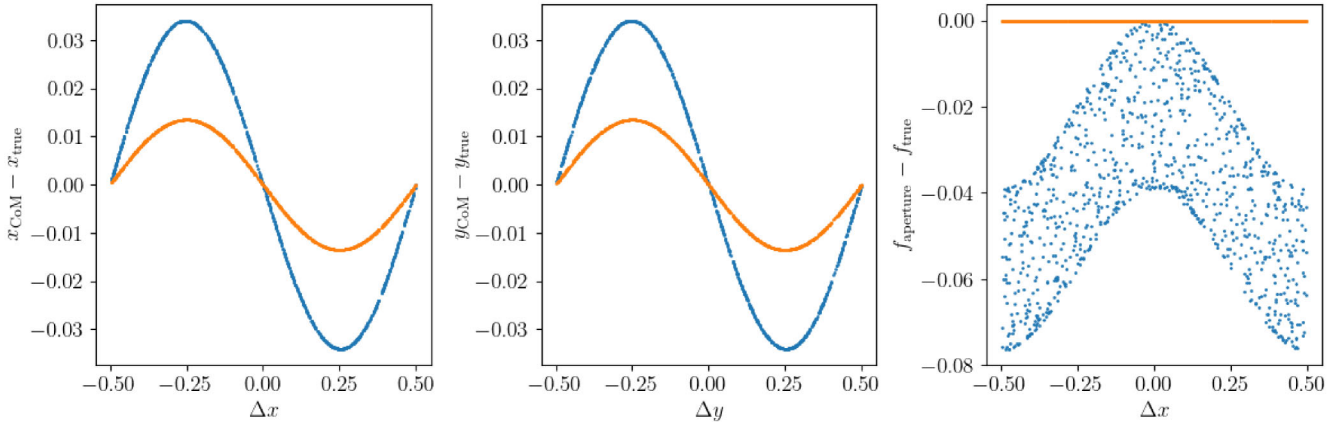


Figure 16. Centre-of-mass centroiding and aperture photometry measurements of 1000 simulated stellar images generated with an ePSF model for a circular Gaussian PSF ($\sigma = 0.4$) and two intra-pixel sensitivity profiles: *flat-with-gaps* (FF = 0.64, blue) and *flat-and-flush* (FF = 1, orange). Left: Pixel-phase errors in the x -centroid versus x sub-pixel position. Middle: Pixel-phase errors in the y -centroid versus y sub-pixel position. Right: Pixel-phase errors in flux versus x sub-pixel position.

average flux, $\overline{f_{\text{obs}}^2}$, is given by:

$$\Psi_E^2(\Delta x, \Delta y) = \frac{f_* \Psi_E^T(\Delta x, \Delta y)}{f_{\text{obs}}^2}. \quad (14)$$

If an infinite number of dithered frames were available such that the stellar centroids uniformly sampled the sub-pixel space in both x and y , the pixel-phase biases in the individual flux measurements would cancel out when averaged. In this ideal case, $\overline{f_{\text{obs}}^2}$ would equal the true stellar flux, f_* , the measured ePSF would be identical to the true ePSF, and subsequent flux measurements using this model would be free of pixel-phase errors.

In practice, however, the number of dithered frames is finite, so we cannot always assume that $\overline{f_{\text{obs}}^2} = f_*$. The value of $\overline{f_{\text{obs}}^2}$ depends on both the dither pattern and the positions of the stars within the dithered images, approaching f_* only for large numbers of uniformly distributed dithers. The effect of the dither pattern on breaking the flux–position degeneracy and producing accurate ePSF models is discussed further in Section 5.7. Equation (13) highlights the need to constrain both the position and the flux of stars in the dithered images to be their average values. We arrived at equation (13) under the assumption that we already knew the exact positions of stars, which has the same effect as if we had constrained the positions of the stars to their average value. However, using constrained star positions but not fluxes will still generate pixel-phase errors in the resulting flux measurements.

We demonstrate this with our simulation framework in Fig. 17. We evaluate the accuracy of the ePSF model and the resulting photometric and astrometric measurements on 1000 simulated star images under four different scenarios: neither flux nor position of stars are constrained, only position of stars are constrained, only flux of stars are constrained, both position and flux of stars are constrained. As expected, when no constraints are applied, strong pixel-phase errors appear in both the fitted fluxes and positions. Constraining only the positions eliminates most of the bias on the positions but leaves flux errors unchanged, while constraining only the fluxes has the opposite effect. When both quantities are constrained, pixel-phase errors in both flux and position are greatly reduced.

Users of the `photutils.psf` subpackage should note that by default, the `LinkedEPSFStar` class constrains only the positions of stars when linking dithered observations of the same source (L.

Bradley et al. 2024). As detailed in Appendix B, we modify the implementation of the `photutils.psf` subpackage to constrain both fluxes and positions in the ePSF modelling.

5.7 Dither pattern

As discussed in Section 5.6, dithered observations are essential for breaking the degeneracy between stellar flux and sub-pixel position introduced by intra-pixel sensitivity variations. However, the effectiveness of this approach depends not only on the number of dithered frames but also on how well their sub-pixel positions sample the detector’s pixel phase space. In this section, we investigate how different dither patterns influence the accuracy of the ePSF modelling and the resulting photometric and astrometric measurements.

In Figs 18 and 19 we test the effect of different dither patterns on the ePSF modelling for the same input ePSF model used in Figs 15 and 16. For each dither pattern tested, we begin by generating 100 simulated star images with random centroid positions. We then make several copies of each star image where the centroids are shifted by small amounts corresponding to each of the dither positions for the given dither pattern. These star images are then used to construct the ePSF model with an oversampling of 4. In Fig. 18, we test random dither patterns corresponding to 2, 4, 8, and 16 randomly dithered images. In Fig. 19, we construct dither patterns which uniformly sample the sub-pixel space as 2×2 , 3×3 , and 4×4 grid of dither positions. In both cases, we compare the accuracy of the ePSF modelling and the resulting flux and position measurements to that for the case of no dithering.

It is clear that a uniform dither pattern is preferable to random dithering. This is expected given the way that the ePSF is modelled as a set of values at specific gridpoints as discussed in Section 5.3. For example, say we have a dither pattern which matches the oversampling grid of the ePSF model. Each gridpoint is evaluated using the star ePSF samples which fall within the corresponding sub-pixel grid section. For such a dither pattern, each individual star will have its centre shifted in each image so that it falls within each gridsection in the central pixel. Therefore, each gridpoint in the ePSF model will be evaluated using samples from all of the stars, minimising the pixel-phase dependence on the ePSF model and resulting flux and position measurements of stars. Even with

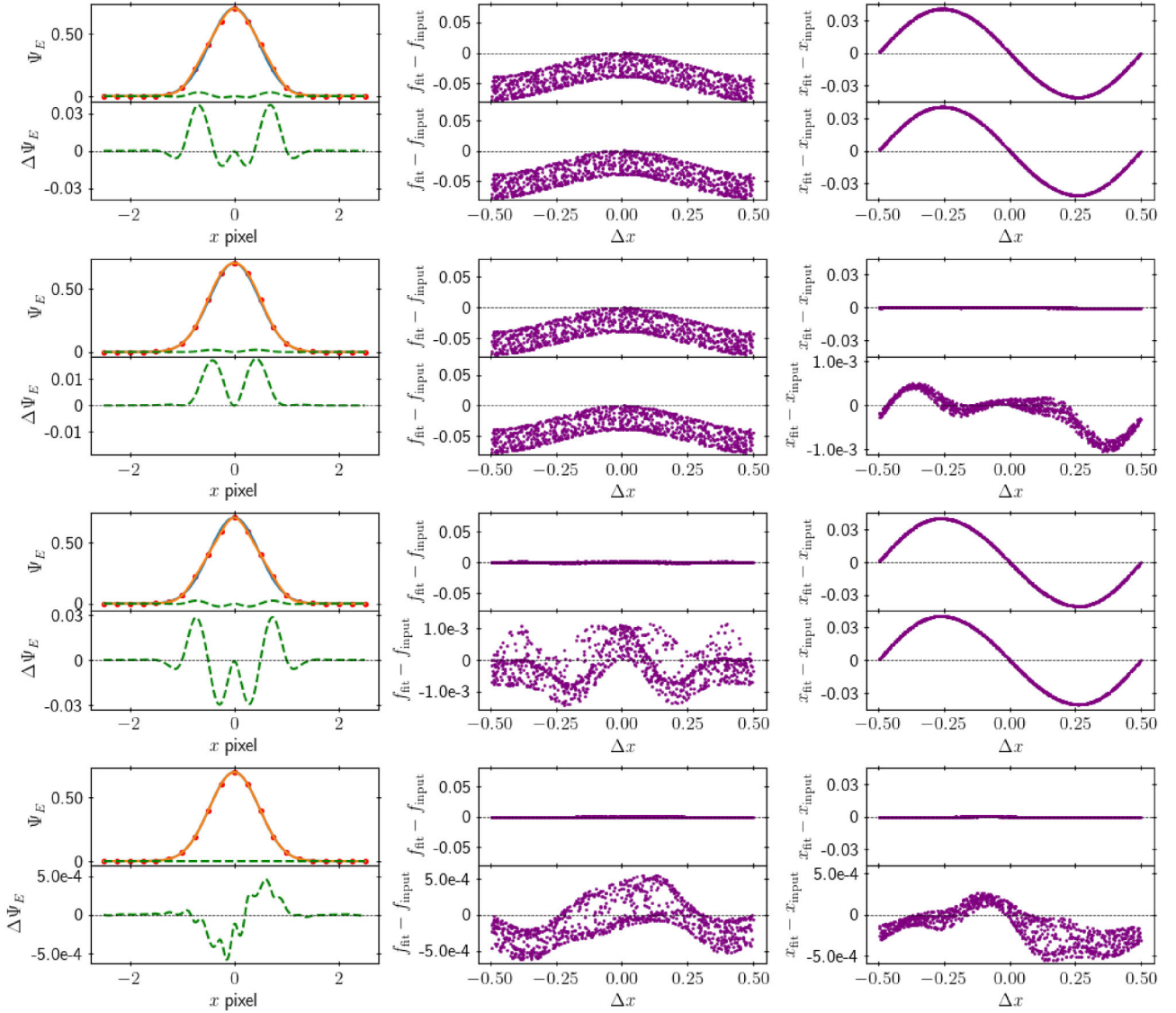


Figure 17. ePSF modelling and fitting accuracy using dithered images of input ePSF model for a circular Gaussian PSF with $\sigma = 0.4$ and a *flat with gaps* intra-pixel sensitivity profile (FF = 0.64). Columns are analogous to Fig. 3. The dither pattern used is a uniform dithering at 1/4 pixel spacings. Rows from top to bottom: Neither flux nor positions constrained, only position constrained, only flux constrained, both position and flux constrained.

a dither pattern which does not fully match the oversampling grid, we still achieve much better results with the uniform dither patterns than random dither patterns with many more dithered frames. Therefore, when either planning dithered measurements or selecting dithered sample images to use for ePSF modelling, the focus should be on obtaining ditherings which uniformly sample the sub-pixel space rather than the number of dithered images obtained.

6 APPLICATION TO THE GLOBAL JET WATCH

We now consider the application of the ePSF modelling routine described above to real astronomical images from the Global Jet Watch (GJW) commensal photometric cameras. The GJW is a network of observatories spread across the globe in longitude to enable round-the-clock spectroscopy and photometry. The GJW observatories were constructed with the goal of performing time-

domain astronomy, whereby we investigate the varying nature of objects on time-scales of hours and days. Each observatory is equipped with a 0.5 m main telescope which collects light for a dual-arm fibre-fed spectrograph. In addition, each observatory has three commensal photometry cameras which are attached to the frame of the main telescope, making use of the pointing, tracking, and guiding of the main telescope. The three cameras allow us to observe in three different photometric filters: H- α , H- β , and a continuum filter at 736 nm. There have been two generations of these commensal cameras. The first generation took wide field images of 5.4×4.3 degrees on CCD detectors of size 2749×2199 pixels, while the second generation take images of 4.5×4.5 degrees on CMOS detectors of size 3008×3008 pixels. Since they are attached to the main telescope, the fields observed by the commensal cameras are determined by the particular objects being observed with the GJW spectrographs. The main scientific goal of these commensal cameras is to identify new transient objects, for instance new microquasar candidates, which will be followed-up with the

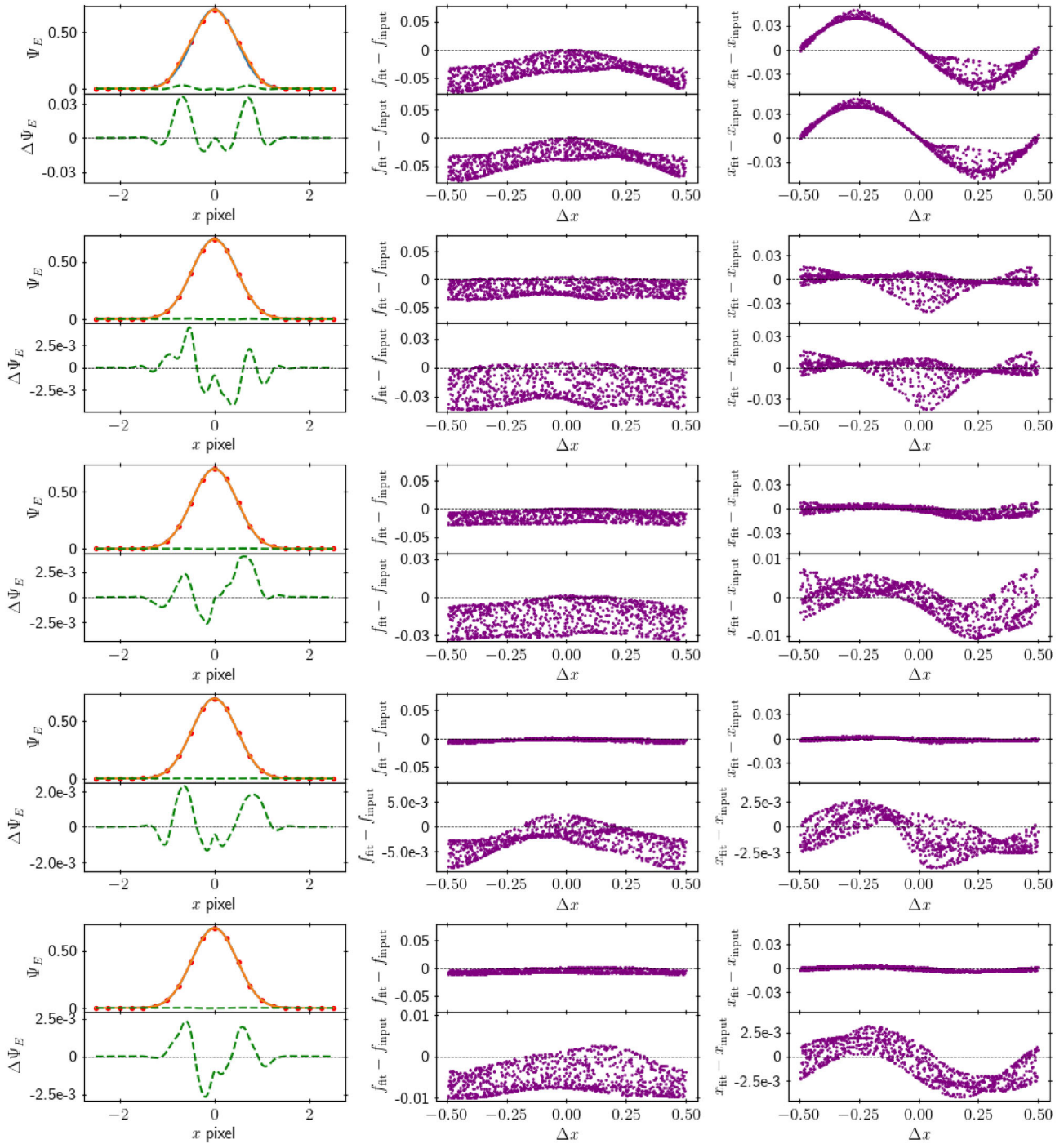


Figure 18. ePSF modelling and fitting accuracy of input ePSF model for a circular Gaussian PSF with $\sigma = 0.4$ and a *flat with gaps* intra-pixel sensitivity profile (FF = 0.64). Columns are analogous to Fig. 3. We investigate the effect of different dither patterns on the ePSF modelling. Rows from top to bottom: No dithering, 2 random ditherings, 4 random ditherings, 8 random ditherings, and 16 random ditherings.

GJW spectrographs. Full descriptions of the observatories and their instruments will be given in Blundell et al. (in preparation) and Lee et al. (in preparation).

The GJW commensal cameras have been collecting images since 2017, and our data set currently consists of over 300 000 images. These images are processed by a bespoke photometric pipeline (full description of this pipeline will be given in Godden et al., in preparation). This pipeline generates light curves for each of the

unique objects detected in the images. The current data base consists of over 500 000 unique objects with each light curve potentially having tens of thousands of data points. A major challenge in the construction of the light curves is accurately measuring the stellar fluxes and positions. The GJW images are very undersampled, with the FWHM of our images being under two pixels, leading to significant pixel-phase systematics in light curves generated using aperture photometry.

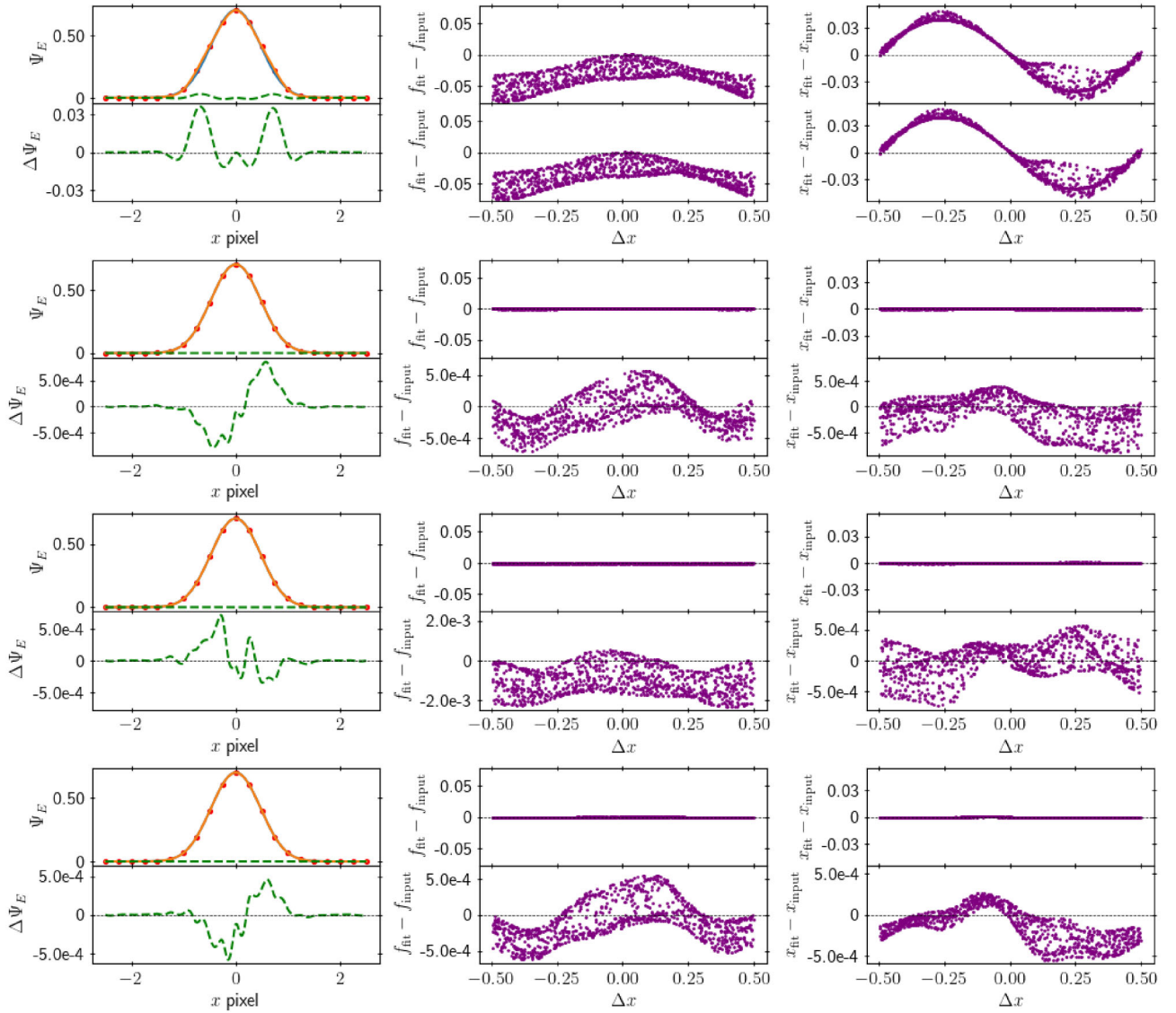


Figure 19. ePSF modelling and fitting accuracy of input ePSF model for a circular Gaussian PSF with $\sigma = 0.4$ and a *flat with gaps* intra-pixel sensitivity profile (FF = 0.64). Columns are analogous to Fig. 3. We investigate the effect of different dither patterns on the ePSF modelling. Rows from top to bottom: No dithering, uniform dithering at 1/2 pixel spacings, uniform dithering at 1/3 pixel spacings, and uniform dithering at 1/4 pixel spacings.

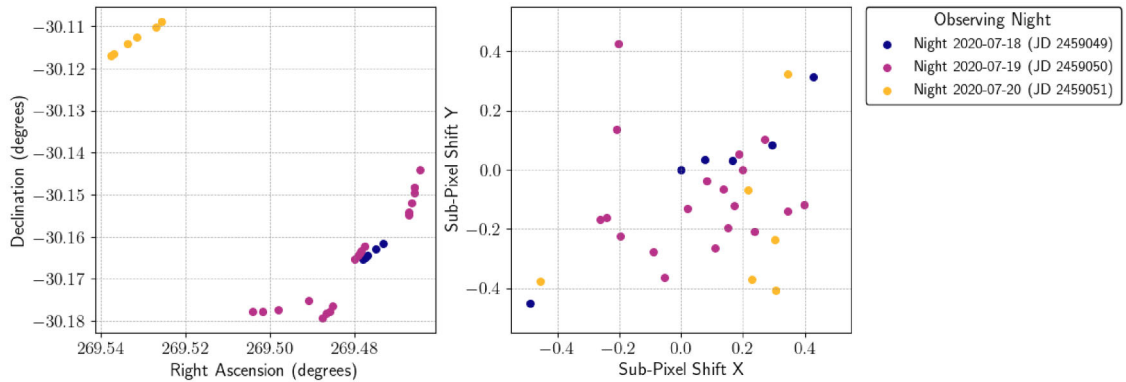


Figure 20. Dithering of sample images used in ePSF construction. Left: Variation in the pointing right ascension and declination of the images. Right: Sub-pixel displacements of the images with respect to the first image in the sample.

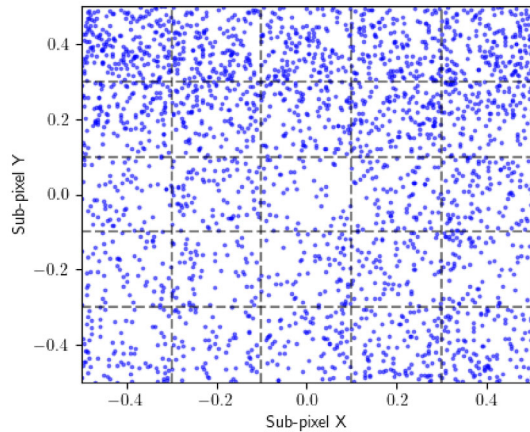


Figure 21. The sub-pixel distribution of the centroids of stars within the sample used to construct the ePSF model for the GJW-SA H- α camera. Black dashed lines show gridsection boundaries for an oversampling of five.

In this section, we investigate the use of the ePSF modelling and fitting routine described in this work to assess whether it is possible to reduce these pixel-phase systematics in the GJW light curves. We use the knowledge from previous sections to determine how to best tune the ePSF modelling routine to the GJW images, implementing this with our modified version of the `photutils.psf` subpackage. We compare our results with those from using the current release of the `photutils.psf` subpackage, and with the results from aperture photometry. As discussed in Section 5, ePSF modelling and fitting with real data present additional challenges, including spatial variation of the ePSF across the detector, geometric distortion, the brighter-fatter effect, and temporal variations in atmospheric seeing. A full description of the ePSF modelling for the GJW cameras, accounting for these effects, will be presented in the second paper in this series, Godden et al. (in preparation). Here, we restrict our analysis to a small central region of the images, using stars within a narrow flux range and selecting frames with consistent atmospheric seeing, effectively treating the ePSF as spatially and temporally invariant. As a proof of concept, we present ePSF modelling results for the H- α -filter commensal camera (first generation CCD camera) at the South African GJW observatory (GJW-SA); the full set of ePSF models will be presented in Godden et al. (in preparation).

6.1 Sample data

To model the ePSF for the GJW-SA H- α camera, we use a set of images of the field around the classical nova V6595 Sgr. This nova was observed between the nights of the 17th July 2020 and the 20th July 2020 by the GJW spectroscopic system for an analysis on the precessing jets of classical novae (D. McLoughlin et al. 2021). During the observation campaign of V6595 Sgr, the commensal photometric cameras obtained images of the field around V6595 Sgr with 100 second exposure times. After removing unsuitable images (e.g. with large cloud cover), there are a total of 237 images within this data set to be used for ePSF modelling. The raw GJW-SA images were bias-, dark-, and flat-field corrected before being used for ePSF modelling. Details of the image calibration and background subtraction procedures are provided in Appendix C.

To construct a stationary ePSF for the GJW-SA H- α data, we restrict the analysis to regions and stars where the PSF shape is

approximately constant. The ePSF modelling is limited to the central 916×732 pixel region of the image, where the PSF is least distorted and geometric distortion effects are minimal. We further restrict the sample to stars with fluxes between 10^4 and 10^5 counts to mitigate brightness-dependent shape variations (the brighter-fatter effect). To minimise the impact of variable atmospheric seeing, only images with similar measured atmospheric seeing values are used, which limits our sample to 31 images.

Fig. 20 shows the dither pattern of this subset of the images. We show both the pointing right ascension and declination of the images (determined by the main telescope), and the sub-pixel displacements of the images. The sub-pixel displacements show how the sub-pixel centroids of each star in the images are varied compared to their position in the initial image of the sample. As discussed in Section 5.7, the optimum dither pattern would form a uniform grid of sub-pixel displacements. Although our sample does not exhibit a uniform pattern, the sub-pixel displacements are fairly well sampled in both the x and y directions. Given a sufficiently large set of sample of randomly distributed stars in each frame, this will be sufficient for ePSF modelling.

From these 31 images, we detect a total of 2767 star samples (123 unique stars) within the specified position and flux ranges. We then remove star samples unsuitable for ePSF modelling, including: blended sources or stars with close neighbours that prevent clean cutouts; stars with noisy backgrounds and low signal-to-noise ratios; sources detected in only a few images; and stars showing flux variations inconsistent with pixel-phase effects, suggesting intrinsic variability. After applying these cuts, 2515 star samples (99 unique stars) remain.

6.2 ePSF Modelling

In this section, we construct the ePSF model using the filtered GJW-SA data set described above. We first need to determine which oversampling parameter to use in the modelling. Fig. 21 shows the sub-pixel distribution of the centre-of-mass centroid of each of the stars. With an oversampling of five, there are an average of 100 samples per gridsection and the minimum number of samples in a single gridsection is 46. This number of samples will be sufficient to accurately estimate the ePSF value at each of the gridpoints with the polynomial fitting method described in Section 5.3. For our ePSF model, we use the RBF cubic interpolator from our investigations in Section 5.2 implemented with SciPy's `RectBivariateSpline` class (P. Virtanen et al. 2020). No smoothing is applied to the ePSF model generated. We run the ePSF modelling routine through 10 iterations, with each iteration updating the measured stellar parameters and the ePSF model.

Fig. 22(a) shows the resulting ePSF model. The residuals between this ePSF model and the stellar samples used to construct it are shown in Fig. 22(c). These residuals demonstrate how accurately our ePSF model represents the star samples. In an ideal case, the residuals would consist only of small random noise. However, we observe some larger residuals – up to approximately 10 per cent of the ePSF peak intensity – and a systematic pattern around the peak of the ePSF model, suggesting that the model slightly overestimates the stellar flux in this region. These residuals likely originate from imperfections in the star samples themselves. The flux recorded by individual pixels in astronomical images contain multiple sources of error, and it is unlikely that all of these have been perfectly accounted for. This means the ePSF samples from stellar image pixels have

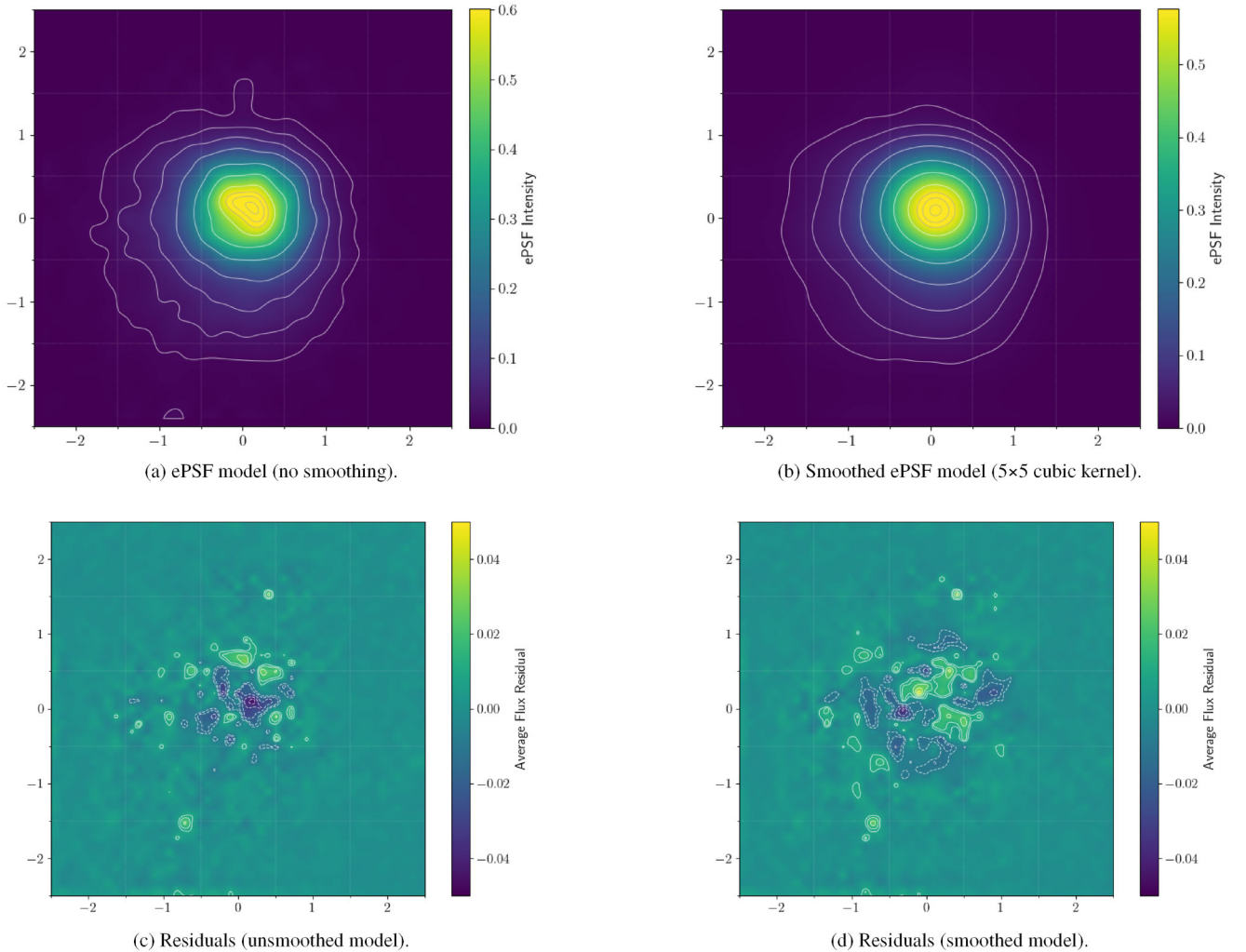


Figure 22. Comparison of ePSF models generated using our updated modelling routine presented in this paper, and their corresponding residuals. **(a)** The ePSF model generated with an oversampling of 5 and no smoothing. **(b)** The same ePSF model with a 5×5 cubic smoothing kernel applied. **(c)** Residuals between the unsmoothed model and the set of sample star images. **(d)** Residuals between the smoothed ePSF model and the same data set. In all panels, contour levels are shown in white and grey.

non-negligible uncertainties, which limit the precision of the ePSF model and produce the residuals seen in Fig. 22(c).

We attempt to mitigate this issue by applying smoothing to the ePSF model. We generate the ePSF with the same configuration as previously, but with the addition of using a 5×5 cubic smoothing kernel. Fig. 22(b) shows the resulting ePSF model and Fig. 22(d) show the residuals between this model and the star samples. The smoothing removes some of the fine scale structure in the ePSF model, and generally makes the model more circularly symmetric. There is no significant change in the scale of the residuals for this smoothed model, and there is not obvious evidence of over-smoothing, for which we would expect a very structured residual.

We compare this to the ePSF model generated from the same star samples but using the current implementation of the `photutils.psf` subpackage (L. Bradley et al. 2024). As previously, the ePSF model is generated with an oversampling of five and with a 5×5 cubic smoothing kernel. The default third degree bivariate spline interpolator is used for this model. We show the ePSF model after 10 iterations in Fig. 23(a), and its residuals with the star samples in Fig. 23(b). For this ePSF model, we observe residuals

with similar scale to those in Fig. 22(d), however they are much tightly clustered around the peak of the ePSF. This means there are more significant residuals at parts of the ePSF model with relatively lower flux. This indicates that the ePSF modelling using the standard implementation of the code is less capable of accurately constraining our ePSF. We measure the widths of the two smoothed ePSF models at 90 per cent of their peak values along the $x = 0$ and $y = 0$ axes. We find that the `photutils` ePSF is 4 per cent wider along $x = 0$ and 17 per cent wider along $y = 0$ than our ePSF model in Fig. 22(b). This, combined with the shape of its residuals, indicates the `photutils` ePSF modelling may be overestimating the width of the ePSF.

We next evaluate the accuracy of ePSF-fitting photometry using the models generated above and compare it to aperture photometry to assess whether the ePSF modelling removes pixel-phase systematics. We use the same subset of 31 images that was used to construct the ePSF models and apply the ePSF fitting to all stars within the region and flux range for which the ePSF models were created. We first consider the pixel-phase errors on the measured centroid positions of the stars. In Fig. 21, we see that the sub-pixel

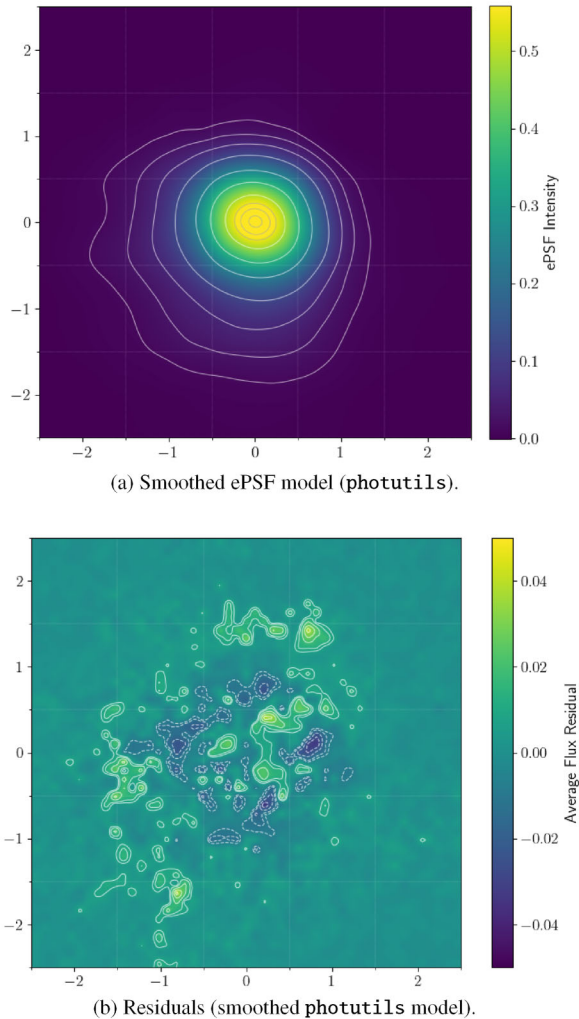


Figure 23. (a) The ePSF model generated using `photutils.psf` with an oversampling of 5 and a 5×5 cubic smoothing kernel applied. (b) The corresponding residuals between this ePSF model and the set of star sample images used to generate it. In both panels, contour levels are shown in white and grey.

positions of stars measured with a centre-of-mass centroid are not uniformly distributed, indicating the presence of pixel-phase errors. In Fig. 24 we show the distribution of the sub-pixel centroids of the same stars as measured with ePSF-fitting, using the improved ePSF from this work and the ePSF constructed with the default `photutils.psf` subpackage. For both cases, we see that the sub-pixel centroids are much more uniformly distributed than the centre-of-mass centroid case, showing that the pixel-phase errors in the position measurements are reduced. There remain some non-uniformities in these distributions, for example a clustering in the bottom left corner in Fig. 24(b). This indicates some remaining errors in the ePSF models, which result in small pixel-phase systematics in the measured positions of stars.

To visualise the pixel-phase dependencies in the measured fluxes of stars, we produce 2D maps of the mean flux residuals as a function of sub-pixel centroid position. Systematic patterns in these maps indicate residual inaccuracies in the ePSF model. Fig. 25 compares the flux pixel-phase errors from our ePSF model, the default `photutils.psf` implementation, and aperture photometry. The pixel-phase maps in this figure have been slightly smoothed to

highlight evidence of systematic trends. For the aperture photometry fluxes, we observe that the flux residuals are more negative towards the centre of a pixel and more positive close to the corners of a pixel. This indicates the presence of some non-uniform intra-pixel sensitivity profile for the pixels of this camera. We see similar scale residuals in the pixel-phase maps for our ePSF model, however we no longer see the same pattern in the residuals. There is still some structure in the pixel-phase residual map which is due to the inaccuracies remaining in the ePSF model. The flux residuals are strongest for the `photutils` ePSF, and show large patches of clear positive and negative systematics. This indicates that this ePSF model has some large deviations from the true ePSF, which cause systematic errors in recorded fluxes which can be even worse than those for aperture photometry.

The ePSF modelling for the GJW commensal cameras is a particularly challenging application, due to the variations in the instrumental PSF (position dependence, brighter-fatter effect, atmospheric seeing fluctuations) and many sources of error on the recorded pixel values of stars within the images. Although we have not been able to generate a perfect ePSF model in this work, our investigations into the ePSF modelling routine and updates to the Python implementation have enabled us to improve on the accuracy of the ePSF model compared with the original modelling routine in the `photutils` package. Work into modelling the ePSF for each of the GJW commensal cameras, accounting for instrumental PSF variations, is ongoing and will be presented in Godden et al. (in preparation).

7 CONCLUSIONS

Accurate modelling of the ePSF is essential for achieving reliable photometric and astrometric precision, especially in undersampled imaging regimes. In this work, we have conducted a systematic investigation into the accuracy of ePSF modelling, following the method of J. Anderson & I. R. King (2000) and its implementation in the open-source `photutils.psf` subpackage of `Astropy` (L. Bradley et al. 2024). Using synthetic ePSFs to generate simulated data sets, we quantified how each component of the ePSF-building routine – oversampling, interpolation, gridpoint estimation, smoothing, star sample distribution, degeneracy breaking, and dither patterns – affect the accuracy of the ePSF model and resulting photometric measurements.

Through controlled experiments, we identified several practical strategies for optimising ePSF construction.

(1) **Oversampling.** The oversampling factor strongly affects model and photometric accuracy. Use the highest oversampling permitted by the available star samples; for near-Gaussian ePSFs we recommend at least ~ 4 gridpoints per FWHM.

(2) **Interpolation.** A radial-basis-function interpolator with a cubic kernel (RBF cubic) gives the most accurate interpolation between gridpoints, especially for undersampled cases.

(3) **Gridpoint estimation.** 2D polynomial surface fits outperform sigma-clipped mean/median estimators, particularly when ePSF samples are sparse or unevenly distributed.

(4) **Smoothing.** Apply smoothing cautiously to suppress noise in empirical gridpoint values; oversmoothing flattens real structure and induces pixel-phase biases.

(5) **Sample size.** Monte Carlo tests quantify how many randomly distributed star samples are required to accurately estimate the gridpoints of the ePSF model; in practice, always inspect the sub-pixel distribution of your samples before fixing the oversampling.

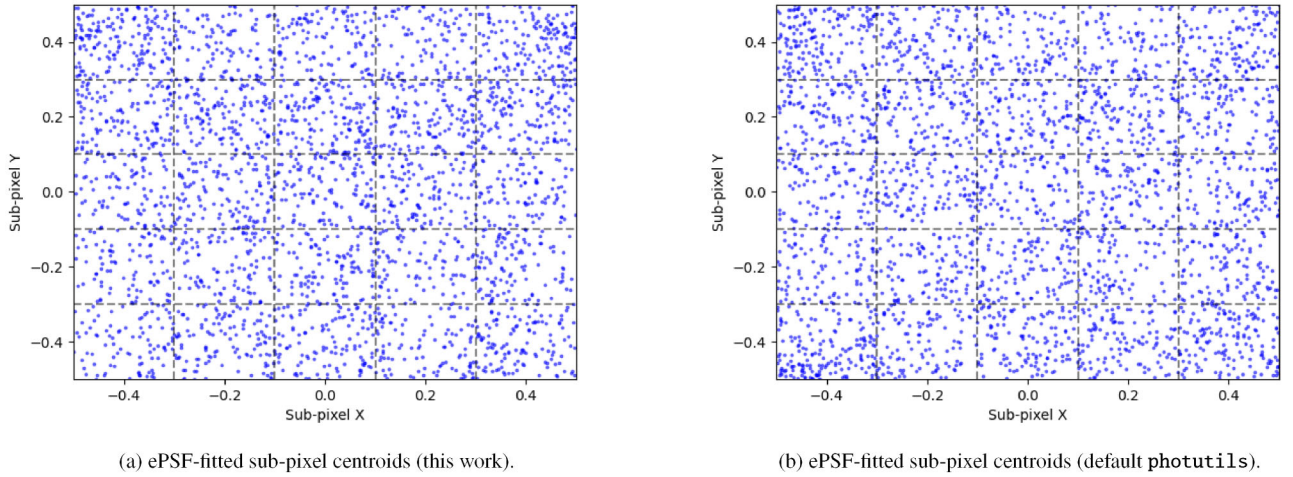


Figure 24. The sub-pixel distribution of the centroids of stars as measured with ePSF-fitting photometry. (a) Centroids measured with our improved ePSF model (Fig. 22b). (b) Centroids measured with the ePSF model using the default `photutils.psf` subpackage (Fig. 23a). Black dashed lines show gridsection boundaries for an oversampling of five.

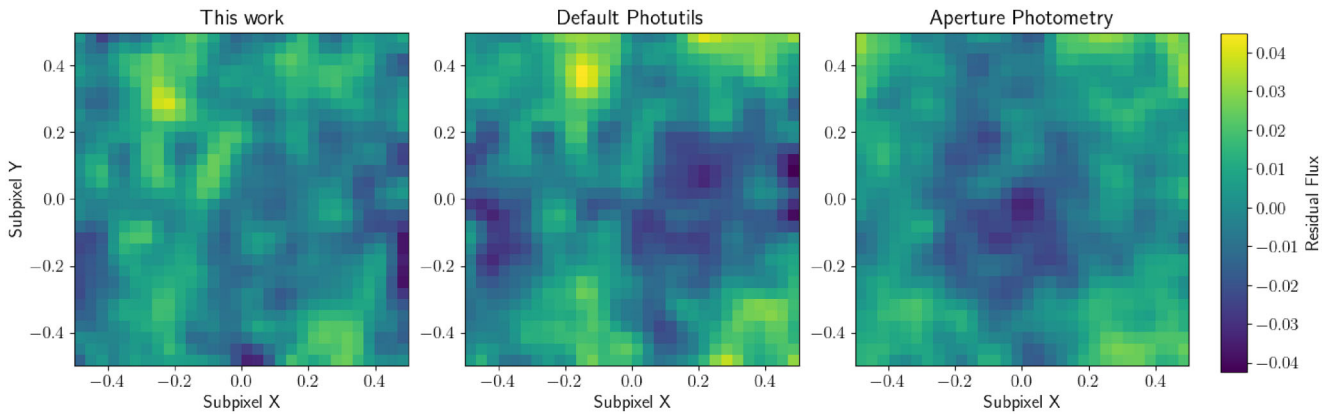


Figure 25. Comparison of flux pixel-phase error (PPE) plots for ePSF-fitting photometry using our improved ePSF model, the default `photutils.psf` model, and aperture photometry. The plots show 2D maps of the average flux residual with respect to sub-pixel centroid position of a star.

(6) **Dithering.** For dedicated dithered observations, uniform sampling of the sub-pixel space (ideally matching the oversampling grid) is far more effective than random dithers.

Importantly, we demonstrate that degeneracy breaking must constrain *both* positions *and* fluxes of stars across dithered frames. Constraining positions alone, the current default in `photutils.psf`, does not remove the flux–position degeneracy introduced by intra-pixel sensitivity. Constraining both the fluxes and positions is necessary to prevent intra-pixel sensitivity variations from introducing systematic errors in the ePSF model and in subsequent photometric and astrometric measurements. This insight provides a clear direction for improving future implementations of `photutils.psf` and similar ePSF frameworks.

We further applied our refined ePSF-modelling framework to real observational data from the GJW commensal cameras. Despite the challenges of undersampled and noisy images, the ePSF models constructed with our modified routine produced photometry with reduced pixel-phase systematics compared to the default `photutils.psf` implementation. While residual scatter remains due to the highly undersampled nature of the data, these results demonstrate the feasibility and benefits of accurate ePSF modelling for real ground-based observations.

Overall, this work provides both theoretical and practical guidance for optimising ePSF modelling routines. The simulation framework, code improvements, and recommendations presented here form a foundation for robust ePSF modelling across diverse instruments. This is a step towards understanding how the ePSF modelling could be fully automated so that minimal human interaction is required to generate an ePSF model for a new instrument.

ACKNOWLEDGEMENTS

The authors acknowledge support from the Science and Technology Facilities Council (STFC), and from St Cross College, University of Oxford, through the St Cross Astrophysics scholarship. This research made use of Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration 2013, 2018, 2022), and SciPy, an open-source Python library for scientific computing (P. Virtanen et al. 2020).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY

The code used to generate our simulated data and the code used to run the experiments in this paper can be provided upon reasonable request to the authors. Our modified version of the `photutils.psf` subpackage of `Astropy` will be submitted for inclusion in the official open-source release.

REFERENCES

- Anderson J., 2016, Instrument Science Report WFC3 2016–12. Space Telescope Science Institute, Baltimore, MD
- Anderson J., Bedin L. R., 2017, *MNRAS*, 470, 948
- Anderson J., King I. R., 2000, *PASP*, 112, 1360
- Anderson J., King I. R., 2003, *PASP*, 115, 113
- Anderson J., King I. R., 2004, Instrument Science Report ACS 2004–15. Space Telescope Science Institute, Baltimore, MD
- Anderson J., King I. R., 2006, Instrument Science Report ACS 2006–01. Space Telescope Science Institute, Baltimore, MD
- Anderson J., Bedin L. R., Piotto G., Yadar R. S., Bellini A., 2006, *A&A*, 454, 1029
- Anderson J., Bourque M., Sahu K., Sabbit E., Viana A., 2015, Instrument Science Report WFC3 2015–08. Space Telescope Science Institute, Baltimore, MD
- Antilogus P., Astier P., Doherty P., Guyonnet A., Regnault N., 2014, *J. Instrum.*, 9, C03048
- Astropy Collaboration, 2013, *A&A*, 558, A33
- Astropy Collaboration, 2018, *AJ*, 156, 123
- Astropy Collaboration, 2022, *ApJ*, 935, 167
- Bellini A., Anderson J., Grogin N. A., Miles N., 2019, American Astronomical Society Meeting Abstracts #234, 301.10
- Bradley L. et al., 2024, *astropy/photutils: 2.0.2*. Zenodo, Available at: <https://doi.org/10.5281/zenodo.13989456>
- Lang D., Hogg D. W., Mierle K., Blanton M., Roweis S., 2010, *AJ*, 139, 1782
- Lauer T. R., 1999, *PASP*, 111, 1434
- Libralato M., Bellini A., Bedin L. R., Piotto G., Platais I., Kissler-Patig M., Milone A. P., 2014, *A&A*, 563, A80
- Libralato M., Bedin L. R., Nardiello D., Piotto G., 2016a, *MNRAS*, 456, 1137
- Libralato M. et al., 2016b, *MNRAS*, 463, 1780
- Libralato M. et al., 2023, *ApJ*, 950, 101
- Libralato M. et al., 2024a, *PASP*, 136, 034502
- Libralato M. et al., 2024b, *A&A*, 692, A96
- Mahato S. B., de Ridder J., Meynants G., Raskin G., van Winckel H., 2018, *IEEE Sensors J.*, 18, 2722
- McLoughlin D., Blundell K. M., Lee S., McCowage C., 2021, *MNRAS*, 503, 704
- Morris R. L., Twicken J. D., Smith J. C., Clarke B. D., Jenkins J. M., Bryson S. T., Girouard F., Klaus T. C., 2020, in Jenkins J. M., ed., *Photometric Analysis*, KSCI-19081-003, p. 87
- Nardiello D. et al., 2019, *MNRAS*, 490, 3806
- Nardiello D., Bedin L. R., Burgasser A., Salaris M., Cassisi S., Griggio M., Scalco M., 2022, *MNRAS*, 517, 484
- Virtanen P. et al., 2020, *Nature Methods*, 17, 261
- Wu Y. et al., 2025, *Res. Astron. Astrophys.*, 25, 125003

APPENDIX A: SYNTHETIC EPSF DERIVATIONS

In our simulations, we use a *flat with gaps* intra-pixel sensitivity profile. This corresponds to a 2D top-hat function of half-width w (so the sensitive side-length is $l = 2w$)

$$R(x, y) = \begin{cases} 1, & \text{if } |x| \leq w \text{ and } |y| \leq w, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A1})$$

The half-width w relates to the fill factor (the ratio of a pixel's light-sensitive area to its total area) as $\text{FF} = 4w^2$. A *flat and flush* pixel model (no gaps, $\text{FF} = 1$) corresponds to $w = 0.5$, whereas a 9:1 ratio between collecting-area width and inter-pixel gap corresponds to $w = 0.45$ ($\text{FF} = 0.81$).

Below we give derivations of the ePSF, $\Psi_E(i, j)$, for instrumental PSF models of interest using this intra-pixel sensitivity profile. These provide the basis for the simulated ePSFs used throughout this work.

A1 Circular Gaussian PSF

We begin with a circular Gaussian PSF with standard deviation σ :

$$\Psi_1(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right]. \quad (\text{A2})$$

Following equation (3), the ePSF can be written as:

$$\Psi_E(i, j) = \frac{1}{2\pi\sigma^2} \int_{i-w}^{i+w} \int_{j-w}^{j+w} \exp\left[-\frac{(x-x_*)^2 + (y-y_*)^2}{2\sigma^2}\right] dx dy, \quad (\text{A3})$$

where (i, j) are pixel coordinates and (x_*, y_*) is the centroid of the star. Because the Gaussian is separable in x and y , the ePSF integral factorises and can be evaluated in closed form using the error function,

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt. \quad (\text{A4})$$

We obtain the ePSF

$$\Psi_E(i, j) = \frac{1}{4} \left[\text{erf}\left(\frac{i+w-x_*}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{i-w-x_*}{\sqrt{2}\sigma}\right) \right] \times \left[\text{erf}\left(\frac{j+w-y_*}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{j-w-y_*}{\sqrt{2}\sigma}\right) \right]. \quad (\text{A5})$$

In our simulations, the error function terms are evaluated using the `scipy.special.erf` implementation.

A2 Elliptical Gaussian PSF

The same approach can be applied to an elliptical Gaussian PSF with different widths σ_x and σ_y :

$$\Psi_E(i, j) = \frac{1}{4} \left[\text{erf}\left(\frac{i+w-x_*}{\sqrt{2}\sigma_x}\right) - \text{erf}\left(\frac{i-w-x_*}{\sqrt{2}\sigma_x}\right) \right] \times \left[\text{erf}\left(\frac{j+w-y_*}{\sqrt{2}\sigma_y}\right) - \text{erf}\left(\frac{j-w-y_*}{\sqrt{2}\sigma_y}\right) \right]. \quad (\text{A6})$$

A3 Multicomponent Gaussian PSF

For an instrumental PSF represented as a superposition of k Gaussian components with relative amplitudes a_k , the ePSF becomes:

$$\Psi_E(i, j) = \frac{1}{4} \frac{\sum_k g_k(i, j)}{\sum_k a_k}, \quad (\text{A7})$$

where each component g_k is given by:

$$g_k(i, j) = a_k \left[\text{erf}\left(\frac{i+w-(x_*+x_k)}{\sqrt{2}\sigma_{k,x}}\right) - \text{erf}\left(\frac{i-w-(x_*+x_k)}{\sqrt{2}\sigma_{k,x}}\right) \right] \times \left[\text{erf}\left(\frac{j+w-(y_*+y_k)}{\sqrt{2}\sigma_{k,y}}\right) - \text{erf}\left(\frac{j-w-(y_*+y_k)}{\sqrt{2}\sigma_{k,y}}\right) \right]. \quad (\text{A8})$$

A4 Rotated elliptical Gaussian PSF

To represent a PSF whose principal axes are not aligned with the detector pixel grid, we use a rotated elliptical Gaussian model,

$$\Psi_{\mathbf{R}}(x, y) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{r} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{r} - \boldsymbol{\mu})\right],$$

where $\mathbf{r} = (x, y)^T$ and $\boldsymbol{\mu} = (x_*, y_*)^T$ are the position and centroid vectors, and Σ is the 2×2 covariance matrix describing the PSF shape and orientation. The covariance matrix can be written as

$$\Sigma = R(\theta) \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} R(\theta)^T,$$

where σ_1 and σ_2 are the standard deviations along the PSF's principal axes, and $R(\theta)$ is the rotation matrix

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

This formulation reduces to the axis-aligned elliptical Gaussian when $\theta = 0$ and to the circular Gaussian when $\sigma_1 = \sigma_2$.

The ePSF is obtained by integrating the bivariate Gaussian over the rectangular pixel area. This integral can be expressed as a combination of bivariate normal cumulative distribution functions (CDFs) evaluated at the pixel boundaries:

$$\Psi_{\mathbf{E},\mathbf{R}}(i, j) = F(i+w, j+w) - F(i-w, j+w) \\ - F(i+w, j-w) + F(i-w, j-w),$$

where $F(x, y)$ denotes the bivariate normal CDF for mean $\boldsymbol{\mu}$ and covariance Σ ,

$$F(x, y) = \frac{1}{2\pi\sqrt{|\Sigma|}} \int_{-\infty}^x \int_{-\infty}^y \exp\left[-\frac{1}{2}(\mathbf{r}' - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{r}' - \boldsymbol{\mu})\right] dy' dx'.$$

In practice, the bivariate normal CDF can be evaluated efficiently using standard numerical routines (e.g. `scipy.stats.multivariate_normal.cdf`).

A5 Skewed Gaussian

To represent a PSF with a tail preferentially on one side we use a skew-normal-type model that multiplies a 2D Gaussian by a 1D skew factor in x . We can represent this skew factor using the standard normal CDF

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt.$$

The skewed Gaussian PSF can then be written as

$$\Psi_{\text{skew}}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{(x-x_*)^2}{2\sigma_x^2} - \frac{(y-y_*)^2}{2\sigma_y^2}\right] \Phi\left(\frac{\eta x}{\sigma_x}\right), \quad (\text{A9})$$

where η is the skew parameter ($\eta > 0$ produces a tail on the $+x$ side). Since the skew factor depends only on x , the ePSF integral factorises:

$$\Psi_{\mathbf{E}}(i, j) = I_x(i) I_y(j).$$

The y -part can be written analytically as

$$I_y(j) = \int_{j-w}^{j+w} \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left[-\frac{(y-y_*)^2}{2\sigma_y^2}\right] dy \quad (\text{A10}) \\ = \frac{1}{2} \left[\text{erf}\left(\frac{j+w-y_*}{\sqrt{2}\sigma_y}\right) - \text{erf}\left(\frac{j-w-y_*}{\sqrt{2}\sigma_y}\right) \right].$$

This leaves the x -part as a 1D integral

$$I_x(i) = \int_{i-w}^{i+w} \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(x-x_*)^2}{2\sigma_x^2}\right] \Phi\left(\eta \frac{x-x_0}{\sigma_x}\right) dx. \quad (\text{A11})$$

For our simulations, we evaluate $I_x(i)$ with a robust 1D integrator (e.g. `scipy.integrate.quad`).

APPENDIX B: MODIFICATIONS TO PHOTUTILS.PSF SUBPACKAGE

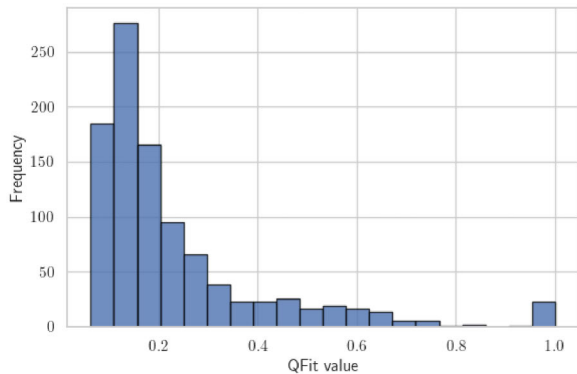
This appendix describes the modifications and extensions made to the `photutils.psf` subpackage in order to align the ePSF modelling routine implemented in the `EPSSFBuilder` class more closely with that described by J. Anderson & I. R. King (2000), and to enable additional flexibility in the ePSF modelling and fitting processes.

B1 Core structural changes

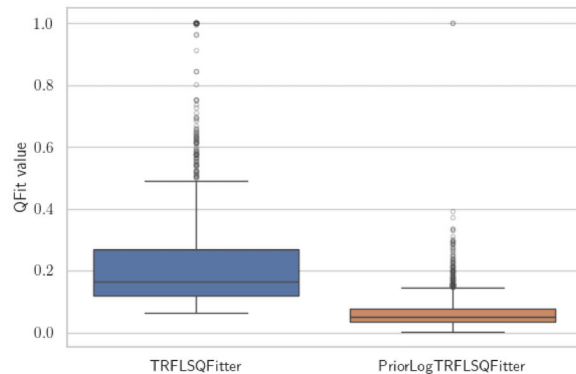
Replacement of legacy ePSF model Both the `EPSSFitter` and `EPSSFBuilder` classes currently implement the `_LegacyEPSFModel` class to represent an ePSF model. This class is a copy of the deprecated `EPSFModel` class, which has since been superseded by the `ImagePSF` class. In our modifications, the `EPSSFBuilder` and `EPSSFitter` classes were refactored to use the `ImagePSF` class, or its subclasses, to represent the ePSF model. During this process, we identified an error in the `_create_initial_epsf` method of `EPSSFBuilder`, in which the ePSF centre was incorrectly calculated due to a misunderstanding of Python's zero-based indexing. This error caused numerical instabilities in the ePSF modelling routine once the `ImagePSF` implementation was adopted. After correcting the centre calculation, the routine produced stable, convergent ePSF models.

Introduction of interpolator subclasses We define a set of subclasses of the `ImagePSF` class that implement alternative interpolation methods for evaluating the ePSF at arbitrary, including sub-pixel, positions. In addition to the default third degree `RectBivariateSpline` interpolator used in `ImagePSF`, we provide subclasses that use `SmoothBivariateSpline` for smoothed spline interpolation, `RectBivariateSpline` with user-defined polynomial degree and smoothing, and `RBFInterpolator` for radial basis function interpolation with a user-defined kernel (i.e. 'cubic').

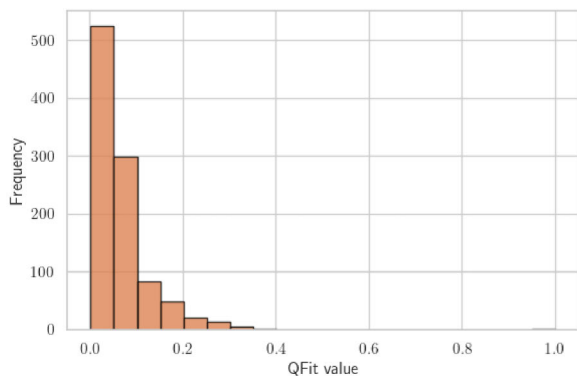
Introduction of new fitter class The `EPSSFitter` class accepts a `fitter` parameter to specify which `astropy.modeling.fitting.Fitter` object to use when fitting the ePSF model to a star image. The default is the `TRFLSQFitter` (Trust Region Reflective Least Squares) from `scipy.optimize.least_squares`. However, we found that this fitter can produce ill-conditioned covariance matrices when fitting parameters such as flux and centroid position that differ by several orders of magnitude, leading to numerical instability. If we fit for the natural logarithm of the flux rather than the linear flux, giving more stable and consistent parameter uncertainties without introducing bias in the recovered flux. This adjustment is particularly important for undersampled images, where only a few pixels constrain each stellar profile and the fit is especially sensitive to numerical issues.



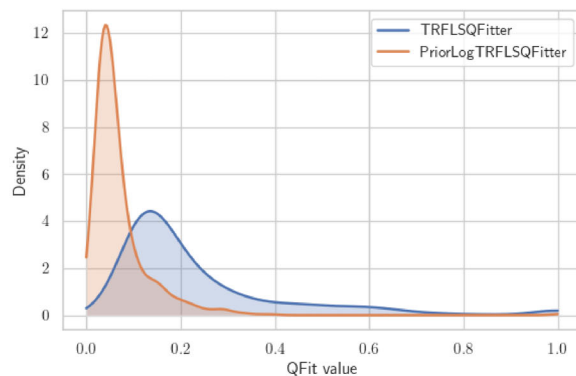
(a) Histogram of QFit results using TRFLSQFitter



(b) Boxplots of QFit results comparing TRFLSQFitter and PriorLogTRFLSQFitter. Grey circles show outliers.



(c) Histogram of QFit results using PriorLogTRFLSQFitter



(d) KDE / density comparison of QFit results using TRFLSQFitter and PriorLogTRFLSQFitter.

Figure B1. Comparison of QFit distributions from fitting of 1000 simulated stars with their known ePSF using different fitting methods.

We define a new TRF-like fitter class, `PriorLogTRFLSQFitter`, which fits the $\log(\text{flux})$ internally, supports Gaussian priors, and returns covariances in both internal ($\log(\text{flux})$, x_0 , y_0) and physical (flux , x_0 , y_0) spaces. Figs B1(a)–(c) show validation tests comparing this fitter with the standard `TRFLSQFitter`. Using simulated stars generated from ePSF Model A (Fig. 1) with fluxes randomly drawn between 10 000 and 100 000 counts, we fit each star with both methods and evaluate the results using the QFit statistic (the sum of absolute residuals between the data and fitted model). The `PriorLogTRFLSQFitter` consistently produces smaller QFit values, indicating more accurate and robust fits.

Introduction of synthetic ePSF models The synthetic ePSF models described in Appendix A are implemented as subclasses of the `Fittable2DModel` class of the `astropy.modelling` subpackage, following the convention of the ‘PRF’ models defined in the `photutils.psf` subpackage. The term ‘PRF’ is equivalent to ‘ePSF’ in this usage. These new classes make it possible to simulate ePSFs with a *flat with gaps* intra-pixel sensitivity profile, making it possible to perform investigations into the effect of non-uniform pixel response, as we have done in Sections 5.6 and 5.7.

B2 Modifications to ePSF modelling routine

We have introduced several modifications to the `EPSPBuilder` class to provide greater flexibility in configuring the ePSF modelling

routine. The original behaviour and settings remain accessible by overriding the default parameters during class initialisation.

Alternative gridpoint estimation methods The gridpoint estimation methods tested in Section 5.3 are implemented in the `EPSPBuilder` class. The desired method can be specified with the `gridpoint_estimation` parameter on class initialisation.

Implementation of ePSF normalisation We implement normalisation of the ePSF model on each iteration in the modelling routine. The ePSF is normalised such that a star with a flux of 1 centred at the middle of a pixel will have an aperture sum of its pixel values of 1. This normalisation can be skipped by setting the parameter `normalise_epsf = False` on the `EPSPBuilder` class initialisation.

Constraining fluxes and positions for degeneracy breaking We define a new method of the `LinkedEPSPStars` class, `constrain_fluxes`, to constrain the fluxes of linked stars to be their average value. This method is analogous to that for the stellar positions, `constrain_centers`. We remove the call of the `constrain_centers` method in the initialisation of the `LinkedEPSPStars` class, since there may be occasions where we do not want to automatically constrain parameters of linked stars. We instead constrain both the fluxes and positions of linked

stars from within the `EPSFBuilder` class during the modelling of the ePSF.

APPENDIX C: GJW IMAGE PRE-PROCESSING

Prior to images being used in ePSF modelling, we perform some image processing steps:

Instrumental calibration The raw images are instrumentally calibrated using bias, dark, and flat-field images. For each of these calibrations, we observe a set of calibration images and produce an averaged *master* calibration image. The raw images are then calibrated using the standard approach:

$$I_{\text{cal}} = \frac{I_{\text{raw}} - B - D}{F}, \quad (\text{C1})$$

where B is the master bias image, D is the master dark image

(bias-subtracted and calibrated for exposure time), and F is the normalised master flat-field image (bias and dark calibrated).

Background subtraction The raw images in this work are taken with an H- α filter, so we need to take care to remove the background due to diffuse nebular light which appears in the images. We find that the background estimation method in the `Background2D` class of the `photutils.background` Python subpackage does not accurately model the background and as a result some structured background light remains in the images. Instead, we decide to estimate the background using Gaussian filtering.

Fig. C1 shows a comparison between the background modelling using this method and with the `photutils.background.Background2D` method. In our background estimation, the large-scale background was estimated via a Gaussian convolution whose width was chosen to be large enough to smooth out flux from stars and high-frequency noise, yet small enough to preserve the smaller scale variations of the nebulous background light. Masks were placed over each star to ensure that stellar flux did not contribute to the background estimate. A PSF-scale smoothed version of the background-subtracted image was then subtracted to obtain a high-frequency residual, from which the local RMS map was derived. To capture the smaller scale structure in the background, we used a Gaussian smoothing kernel with a width of $\sigma \approx 6.8$ px (corresponding to $\text{FWHM} \approx 16$ px). This relatively narrow filter causes a small amount of flux from bright stars that are not fully masked to be included in the background model. This could be mitigated by identifying these bright stars and applying larger mask radii. However, since the ePSF modelling only uses stars within a specific flux range, the brightest stars are excluded from that analysis, and therefore the inclusion of a small fraction of their flux in the background model has no impact on our results in this work.

Astrometric calibration After background calibration, we then calculate the World Coordinate System (WCS) transformation of the images using the `Astrometry.net` software (D. Lang et al. 2010). This is essential for being able to cross-match detections of the same stars across different frames.

Source detection We use the `photutils.detect_sources` function in order to identify stars above a specified threshold in our images (L. Bradley et al. 2024). Our threshold for detection is set such that there must be four connected pixels at a value of five standard deviations above the local background value.

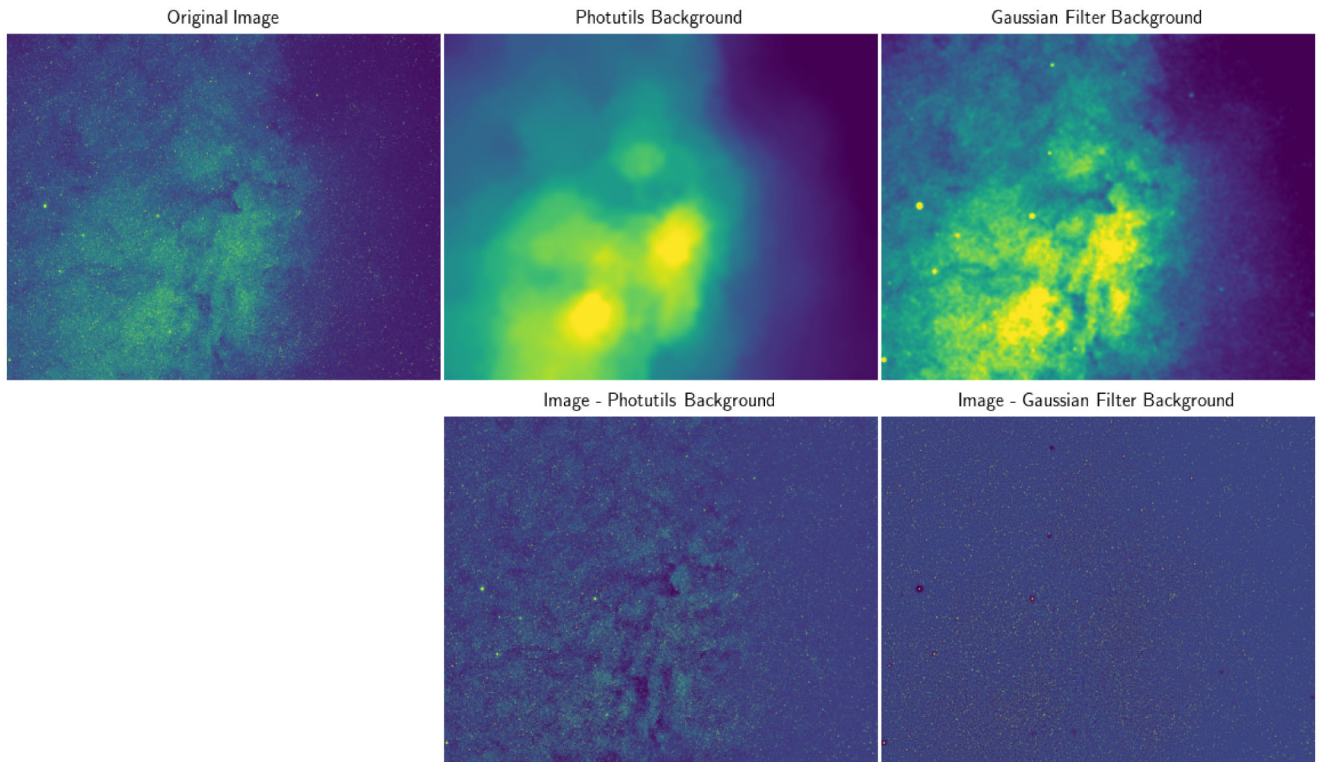


Figure C1. Comparison of background estimation methods using a single image of the V6595 Sgr field with the GJW-SA H- α camera. Left: Raw image. Middle: Background and subtracted image with `photutils Background2D` estimation. Right: Background and subtracted image with Gaussian filtering estimation.

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.