Contents

1. INTRODUCTION

Extensive research in radio astronomy has heightened the understanding of the universe, cosmology, galaxy evolution, and the extragalactic source population in recent times (Casey et al. (2014)). This is due to dust being transparent in radio bands, which leads to unbiased observations in radio continuum observations when it comes to tracing cosmic evolution (Mauch et al. (2020)). A valuable and important part in constraining radio cosmic evolution is characterizing the underlying confused source population, where its relation to radio source evolution is discussed later in this paper.

Confusion can be described as background intensity readings that cannot be distinguished as distinct sources. This is due to the fact that no amount of additional integration time past a certain point at a given resolution can differentiate individual sources. In other words, the only way to differentiate these blurred sources is to achieve a higher resolution image, which involves utilizing or building a larger effective dish for a radio telescope. Due to this obstacle, it is more effective to gather what information can be extracted from confusion than to minimize confusion altogether. The specific flux-density point at which these sources become 'confused' is called the confusion limit. A method of best constraining the source counts at these low flux-density levels is by using the probability of deflection, P(D), analysis (Scheuer (1957)). As a radio telescope scans the sky, it reads the low flux-density amplitudes, or deflections D, and P(D) describes the probability of how strong that deflection is. For a power-law universe where point sources obey the differential source counts:

$$\frac{dN}{dS} = N_0 S^{-\kappa} \tag{1}$$

the noise-free confusion probability distribution function (PDF) for an arbitrary pencil-beam antenna follow the scaling relation:

$$P[(N_0\Omega_e)^{1/(\kappa-1)}D] = (N_0\Omega_e)^{-1/(\kappa-1)}P(D) \tag{2}$$

where the effective solid angle Ω_e, reliant on the Gaussian beam solid angle Ω_b is:

$$\Omega_e = \frac{\Omega_b}{\kappa-1} \tag{3}$$

$$\Omega_b = \frac{\pi\theta_1\theta_2}{4\ln 2} \tag{4}$$

where the elliptical Gaussian beam has FWHM axes θ_1 and θ_2. For visual purposes of displaying analytic $P(D)$ distributions is $N_0\Omega_e = \eta_1^{-1}$, where:

$$\eta_1^{-1} = \frac{2\Gamma(\kappa/2)\Gamma[(\kappa+1)/2]\sin[\pi(\kappa-1)/2]}{\pi^{\kappa+1/2}} \tag{5}$$

where Γ is the factorial function. It is important to note that the functional shape of P(D) relies upon only κ. This further analytical analysis was derived by Condon (1974) off of work by Scheuer (1957).

The observed D at any point (or pixel), in units of flux-density per beam solid angle, is the sum of the image noise and the noise free-source confusion. These measurements are independent from each other, so the observed P(D) is the convolution of the source confusion and image noise (Condon (1974)). Since Ω_e is derived from the normalized point source function (PSF), $\Omega_e = \int G^{\kappa-1}d\Omega$, it is implied that the sidelobes of the PSF have a larger effect as κ decreases, therefore the confusion measurements for μJy and below levels at $\kappa - 1 \sim 0.5$ must have very low sidelobe levels (Condon et al. (2012)). The rms confusion variance σ_c^2 diverges for all $\frac{dN}{dS}$ (Mauch et al. (2020) and Condon (1974)), so P(D) must be truncated above a cutoff deflection D_c. Favoring $q = \frac{D_c}{\sigma_c}$ for a constant q representing the signal-to-confusion ratio, the rms confusion σ_c is

$$\sigma_c = (\frac{q^{3-\kappa}}{3-\kappa})^{1/(\kappa-1)}(N_0\Omega_e)^{1/(\kappa-1)} \tag{6}$$

and for a power-law source count, the number of beam solid angles per source stronger than D_c is

$$\beta = \frac{q^2}{3-\kappa} \tag{7}$$

As $q = 5$ is a typical and reliable source detection, Equation (7) limits the maximum density of reliably detectable sources to about 1 source per 25 beam areas around $\kappa \sim 2$ (Condon et al. (2012)).

To accurately extract the noise-free P(D), and therefore $\frac{dN}{dS}$ and σ_c, N_0 and κ have to be optimized as free parameters. From the determined or parameterized σ_n, the observed P(D) is extracted from the convolution of the image noise and the noise-free P(D). Equation (2) is a simplified relation of the previous statement, and is ultimately derived from the master P(D) equation,

$$P(D) = 2\int_0^\infty \exp\left(-N_0\Omega_e\eta_1\omega^{\kappa-1}\right)\cos\left(N_0\Omega_e\eta_2\omega^{\kappa-1} + 2\pi\omega D\right)d\omega \tag{8}$$

where η_1 was referenced in Equation (5) and η_2 is

$$\eta_2 = \pi^{\kappa-1/2}\Gamma[\frac{1}{2}(2-\kappa)]/\{2\Gamma[\frac{1}{2}(\kappa+1)]\} \tag{9}$$

where D in Equation(8) is now taken upon to represent the mean deflection $\overline{D}$ instead of absolute zero, unless $1 < \kappa < 2$, where the sky brightness no longer diverges and Olbers' paradox can be avoided so D can represent absolute zero (Mauch et al. (2020)).

2. METHODOLOGY

2.1. *The Lévy Distribution*

Due to the extreme difficult for computing realizations of P(D) derived by Condon (1974) as a result of it having no analytical form from the characteristic equation, we approached the problem with a more generic method, which Condon's P(D) is based off of. Described in Herranz, D. et al. (2004), the pixel distribution of the sub-μJy region resembles an extremely impulsive noise probability distribution function (PDF). This behavior can be partially described by a Lévy α-stable distribution, a statistics model popularly used for economics and engineering. The characteristic function for the Lévy distribution completely describes any PDF, such as Gaussian or Poisson distributions, and has no analytical form. Four parameters govern the behavior of the Lévy distribution:

1. α is the characteristic exponent that describes the impulsiveness of the function. In the domain $\alpha = (0, 2]$, $\alpha = 2$ describes a Gaussian, and as $\alpha \to 0$, the function becomes more impulsive.

2. β controls how skewed or symmetric the function is on the domain $\beta = [-1, 1]$. Examples of β are $\beta = 0$ resembles a symmetric distribution, and $\beta = 1$ completely left-skews the function.

3. γ is the scale parameter which controls how spread out the distribution is for $\gamma > 0$.

4. μ is the shift parameter, which offsets the distribution along the x-axis.

The Lévy distribution is then described in the Fourier space as:

$$\psi(w) = exp\{i\mu w - \gamma|w|^{\alpha} B_{w,\alpha}\} \tag{10}$$

$$B_{w,\alpha} = \begin{cases} [1 + i\beta\,\mathrm{sgn}\,(w)\tan{(\frac{\alpha\pi}{2})}] & \alpha \neq 0 \\ [1 + i\beta\,\mathrm{sgn}\,(w)\frac{2}{\pi}\log|w|] & \alpha = 0 \end{cases} \tag{11}$$

A new model for P(D) can then be constructed by convolving a parameter-optimized Lévy distribution with a Gaussian distribution, representing the image noise. In this sense, the Lévy α-stable distribution becomes a maximum-skewed ($\beta = 1$) distribution with no localization due to convolution (hence, μ is arbitrary).

2.2. *P(D) Implementation*

P(D) is now described here as the convolution of the noise-free distribution due to the underlying confused source population and Gaussian-assumed image noise. The parameters that define the noise-free distribution, shown in (11), are α, β, γ, and μ. Since β and μ are not relevant to describing the confusion source density, only α and γ are used in calculations, where:

$$\alpha = \kappa - 1 \tag{12}$$

$$\gamma = \frac{\pi^{3/2} N_0 \Omega_e}{2^{\alpha+1}\Gamma(\frac{\alpha+1}{2})\Gamma(\frac{\alpha+2}{2})\sin(\frac{\alpha\pi}{2})} \tag{13}$$

Rearranging to solve for N_0 and κ, the source-defining parameters can be found by parameterizing and optimizing α and γ:

$$\kappa = \alpha + 1 \tag{14}$$

$$N_0 = \gamma\frac{2^{\alpha+1}\Gamma(\frac{\alpha+1}{2})\Gamma(\frac{\alpha+2}{2})\sin(\frac{\alpha\pi}{2})}{\pi^{3/2}\Omega_e} \tag{15}$$

The units of the differential source counts (1) are Jy^{-1} sr^{-1}, and the normalization choice affects the final units as well, where normalizations are discussed later. This is constant across all literature, so the optimized choice of κ dictates the units of the source density normalization, N_0, so that N_0 multiplied with $S^{-\kappa}$ always results in Jy^{-1} sr^{-1}. During parameter optimization, which is discussed later, the optimized N_0 has the correct units by default, but this fact becomes extremely important when working in any regime that's not strictly on the Jy scale. Through trial and error, it is apparent that performing the analysis in the μJy regime outputs far more accurate results, given the limitations of the specific python calculations when dealing with numbers on a tiny scale ($< 10^{-6}$). To combat this, P(D) analysis was done here in the μJy scaling, then converted to the appropriate Jy scaling while keeping the units in mind. Hence, the scaling of 10^{-6} also has to have the same powered-unit as N_0 to keep the multiplied units of N_0 with $S^{-\kappa}$ as Jy^{-1}. Therefore, the ultimate equation used to calculate the confusion source density normalization parameter is now:

$$N_0 = \gamma \frac{2^{\alpha+1}\Gamma(\frac{\alpha+1}{2})\Gamma(\frac{\alpha+2}{2})\sin(\frac{\alpha\pi}{2})}{\pi^{3/2}\Omega_e} * \left[(\frac{10^6\mu\text{Jy}}{\text{Jy}})^\alpha\right]^{-1} \tag{16}$$

where (16) is strictly used only when the parameter optimization is performed in the μJy regime to utilize $\frac{dN}{dS}$ (1) in the units of Jy^{-1} sr^{-1}. This power optimization is indirectly showcased in Section IV a) within Mitchell & Condon (1985), where they show the power of their k variable, synonymous with our N_0. Though never explicitly stated in any relevant literature, these standardized units are also apparent across all related cited papers here.

2.3. *Modeling Choice*

The most efficient way to model P(D) is to utilize the Markov Chain Monte Carlo (MCMC) method due to its effective ability to model complicated functions. The advantage of using MCMC methods over other optimization methods, such as the Nonlinear Least Squares (NLS) Regression optimizer, is that MCMC incorporates randomness to best optimize multi-parameter spaces with trial-and-error precision. The NLS Regression technique requires precise knowledge of the priors, is more strongly susceptible to outliers, and can also get stuck on local minimums rather than the absolute minimum, whereas using an MCMC diminishes these downsides.

A brief explanation of how MCMC works is it constructs a Markov chain, which is a sequence of states where each state depends only on the previous state. MCMC explores the space of possible states by making small, random changes to the current state. These changes are proposed based on a user-supplied proposal distribution. The acceptance of these proposed changes is determined by comparing the probabilities of the current state and the proposed state. By accepting or rejecting proposed states, the Markov chain efficiently explores the most probable, or highest likelihood, regions of the space. If the MCMC samples the space ergodiclally, the density of visits of the chain to each point in the space is proportional to that point's probability, and so the total MCMC run allows us to generate samples from the target distribution. With this we may estimate various quantities of interest, such as parameter probability means, variances, and ultimately in this use of MCMC, the best optimized parameter value. The power of MCMC lies in its ability to handle high-dimensional problems and provide reliable approximations even when explicit calculations are infeasible.

A downside of using an MCMC is computational intensity and time. Producing one realization of P(D) computationally takes a noticeable fraction of a second on a computer with a powerful CPU, which is much longer, in comparison, than computing the result for a much simpler function. Since optimizing P(D) sufficiently requires hundreds of iterations, producing a result can take hours, so a good thought into the resolution of the data and the number of iterations required to output reliable results in an efficient manner is recommended.

2.4. *emcee*

For convenience and reliability purposes, we utilized the python package `emcee`, described and documented in Foreman-Mackey et al. (2013). The source documentation website provides excellent resources and guidance on using MCMC and this python package to work in the most simplistic and efficient manner. We found this to be the most helpful of the MCMC options out there due to these reasons, and most importantly, it works great with our complex function to optimize.

2.5. *Method Explanation*

P(D) is defined to completely describe the probability of a pixel deflecting by a certain intensity for a large range of flux density bins, above and below a detection limit at which the sources are no longer individually resolvable. There have been many interpretations on how to utilize P(D) to estimate the faint source population (Scheuer (1957), Condon (1974), Vernstrom et al. (2014), Herranz, D. et al. (2004)), and our method combines the methods Condon (1974) and Herranz, D. et al. (2004) uses. Our reasoning is Condon et al. (2012), based off of Condon (1974), explains that source confusion is independent from noise contributions. The deflection, D, is the sum of the noise-free source confusion and the image noise in a given flux density bin, therefore it is appropriate to claim that the P(D) distribution is the convolution of the source confusion and image noise distributions. This statement is the basis of our method and model, where the later is discussed in the sequential subsections. The source confusion rms is defined as:

$$\sigma_c = (\sigma_o^2 - \sigma_n^2)^{1/2} \tag{17}$$

where we see we have to take the extracted source confusion rms with caution, since careful calculations of σ_o and σ_n are required to produce a confident σ_c.

Hewish (1961) showcases the practical usage and validity of the original P(D) methods, described by Scheuer (1957), by utilizing the Monte Carlo method and fitting a Rayleigh distribution. Herranz, D. et al. (2004) later expanded on this discussion by showcasing an expanded reasoning on analytic expressions of the characteristic function of P(D)'s PDF in the Fourier space. Their equations (1) - (6) exactly derive their state equation, (7), or (11) in this paper. By combining the noise-free source confusion model in the form of a Lévy α-stable distribution (Herranz, D. et al. (2004)) and the reasoning that the noise-free source confusion distribution convolved with the image noise distribution is equivalent to P(D) (Condon et al. (2012)), conveniently modeled by a Gaussian, we take the explanation literally and model the observed P(D) distribution, seen in radio data, as such. In summary, by modeling the observed P(D) where we best fit the image noise and the parameters that best fit the noise-free source confusion and convolve them, we obtain the best fitted model P(D).

Herranz, D. et al. (2004) and Condon et al. (2012) both show their methods on how to extract P(D), using logarithmic moments with expectation operators and mathematical scaling relationships, respectively. These methods performed are independent of the theory, as in they have no effect on the validity of the theory derived to produce P(D), and they simply use their own applications of the theory to best optimize the parameters defining P(D). We do the same by deriving this method we used to obtain the best fit P(D), as it is the most practical for us to use, due to the mathematical complexity in the other methods.

2.6. *The Differential Source Counts*

The term 'differential sources counts' has been thrown around without further detail on what it details, but it is now relevant to explain further on what it means. $\frac{dN}{dS}$ describes the number of sources with flux densities S to $S + dS$ per steradian, while taking the integral of $\frac{dN}{dS}$ gives the total integral counts within the desired range of flux densities. Hence, it can also be viewed as the statistical source density for sources beyond the confusion limit.

The best way to showcase the resulting optimized confusion source density is by utilizing the differential source count density plot. Two common displays of this plot are Euclidean and brightness-weighted normalizations, due to the fact that the slopes of the counts are too steep to analyze its features regularly. Euclidean normalization multiplies the counts by $S^{5/2}$, while brightness-weighted normalization multiplies the counts by S^2 (resulting in the density becoming $S^{5/2}\frac{dN}{dS}$ and $S^2\frac{dN}{dS}$, respectively). More about the fundamentals and explanations of this can be found in Kellermann & Verschuur (1988). Described in Mauch et al. (2020), especially in their Figure (18), a rudimentary analysis of the differential source count curve can be split up into to evolving groups: active galactic nuclei (AGNs), which dominate the bright radio sky, and star-forming galaxies (SFGs), which dominate the dim radio sky. The confusion limit for modern radio observations is well beyond the area where AGNs dominate, suggesting that SFGs almost entirely make up the underlying confused source population. The apparent trend of confused counts via P(D) and distinct brighter counts have a tight correlation with the evolution of the local radio luminosity function (Condon et al. (2012)), and can serve as a constraint on the general evolution of the radio source population, which is looked into later.

3. DATA AND SAMPLE

3.1. *The CHILES Con Pol Ultra Deep Radio Continuum Dataset*

The CHILES Continuum Polarization Survey (CCP) is a radio continuum deep-field with 1000 hours of integration time at 1.4 GHz taken by the Jansky Very Large Array (VLA) in New Mexico. It is a partner survey taken alongside the COSMOS HI Large Extragalactic Survey (CHILES). It also has 4.5" resolution, making it one of the most sensitive images in the L-band. More about the CHILES Con Pol can be read in Luber et al. (2023), in preparation.

Figure (1) shows the entire survey area of the CCP. Important details to note are the confusion is very apparent, especially in the distinguished primary beam area. Figure (1) shows a zoomed in image of the central 5' radius from the center of the beam, showcasing the heavy source confusion. Due to how confused the data is along with how deep the CCP goes, this is a great sample to perform P(D) analysis on to characterize the underlying population as low as we feasibly can.

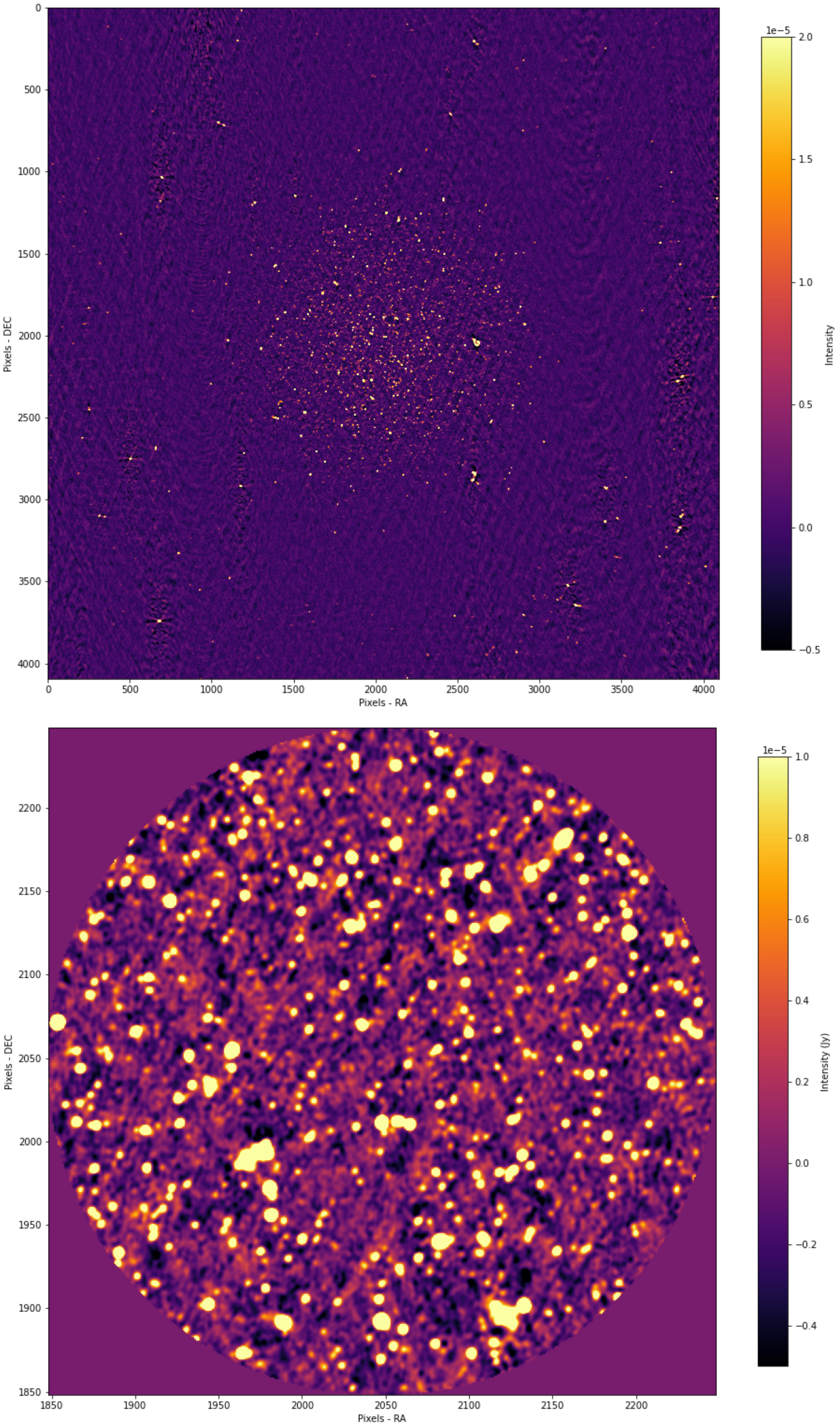

Figure 1. Top: 4096x4096 image of the CCP. Each pixel has an area of 4.16E-4 x 4.16E-4 deg^2. The noise on the outside of the primary beam is extremely noticeable. Streaks and patterns are also apparent. Bottom: Central 5' of the image and primary beam. Both images are scaled to a minimum and maximum of -5 μJy and 10 μJy to emphasize the source confusion.

3.2. *Selection*

To extract the most accurate calculation of the confused source population, we choose to select the region with the best readings of sources, confusion, and image noise that also has the least bias towards any one group. Therefore, we based our analysis on the central 5' of the image (1), which is also the center of the primary beam. One of the main assumptions of a power-law focused P(D) analysis is a Gaussian beam along with Gaussian image noise. Though it is not the reality of the dataset, and with this assumption ignoring the sidelobes of the beam, a Gaussian image noise well describes the pixel distribution and is a fair assumption, which will be described later. This means that the power-reading of true sources falls off away from the center of the beam, while image noise, including instrument noise, stays consistent throughout the image. The central 5' of pixels is then determined to be the most unbiased sample from the dataset, while also containing enough of a well-sampled area for the MCMC.

3.3. *Notes*

There are a couple things to note about the dataset regarding systematic effects. In Figure (1), there are clear patterns and artifacts visually present, especially away from the primary beam. Taking the Fourier transformation of the image also shows patterns, where it would normally be uniform. The best explanation of this is there were a couple days involving bad weather while taking data that were not removed from reduction and CLEANing. This directly affects the quality of the image noise, and one of the fundamental factors of this P(D) analysis is obtaining the most accurate reading of the image noise. It is unknown at the time how much of a factor it is in affecting the optimized results, if at all, but requires careful thought in how to analyze the data.

4. ANALYSIS AND RESULTS

4.1. *Execution*

We converted the central 5' of pixels into a distribution of pixel counts per intensity bin. In Figure (2), the resulting distribution has similar features as a Gaussian, but has a noticeable positive tail-end. The domain of this distribution is -10 μJy to 10 μJy, where the lower limit is where the distribution ends and the upper limit is set to 10 μJy to easily compare this analysis to those in other literature, such as Mauch et al. (2020), who used the same upper limit. Breaking it down, the distribution is made up of primarily 3 components: distinct sources and their point spread functions, image noise, and source confusion. Distinct, bright sources reside in the extreme positive tail, whereas the image noise and confusion primarily construct the body of the distribution. The image noise is assumed Gaussian with a mean of $\bar{S} = \mu_n = 0$ by definition, and is fitted later in the analysis. The source confusion, in theory, is an impulsive-like distribution that mainly follows a source population model, specifically Equation (1), and is non-negative. In realistic scenarios, there is a minimum and maximum intensity observed, therefore, the population model stands true for only

$$\frac{dN}{dS} = \begin{cases} 0 & S < S_{min} \\ N_0 S^{-\kappa} & S_{min} < S < S_{max} \\ 0 & S > S_{max} \end{cases} \tag{18}$$

Under these assumptions, it can be inferred that the mean of the entire distribution is slightly positive-skewed, but in practice, the mean turns out to be approximately 0 with the peak intensity being slightly negative. This is caused from the post-processing of the radio data, where a side-effect of it integrates the data to 0, shifting the observed distribution. Careful attention is required when carrying out the fitting to the data, discussed more in Section 4.2.4.

The data distribution was then used for the model fitting, using `emcee` to utilize the MCMC method. The MCMC code takes initial parameters in, and so therefore, we used `scipy.optimize.curve_fit` to obtain a rough initial guess. The functionally equivalent computation that was implemented into the model for the MCMC is:

$$P(D) = \left[\frac{1}{\sigma_n\sqrt{2\pi}} \exp{(-\frac{1}{2}(\frac{D}{\sigma_n})^2)}\right] * \left[\frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} \exp{(-iwD)}\psi(w)dw\right] \tag{19}$$

where D is the deflection and $D = S$, σ_n is the image noise, $*$ is the convolution sign, $\psi(w)$ was defined in Equation (10), and w is the characteristic function argument.

We utilized the python package `scipy.stats.levy_stable` to make the code more efficient as a result of extremely slow test trials when calculating the Lévy function manually. We also implemented a normalization for Equation (19) equal to 1 because the Lévy package did not round perfectly to a normalization of 1, where the convolution requires inputs to be PDFs to output a PDF.

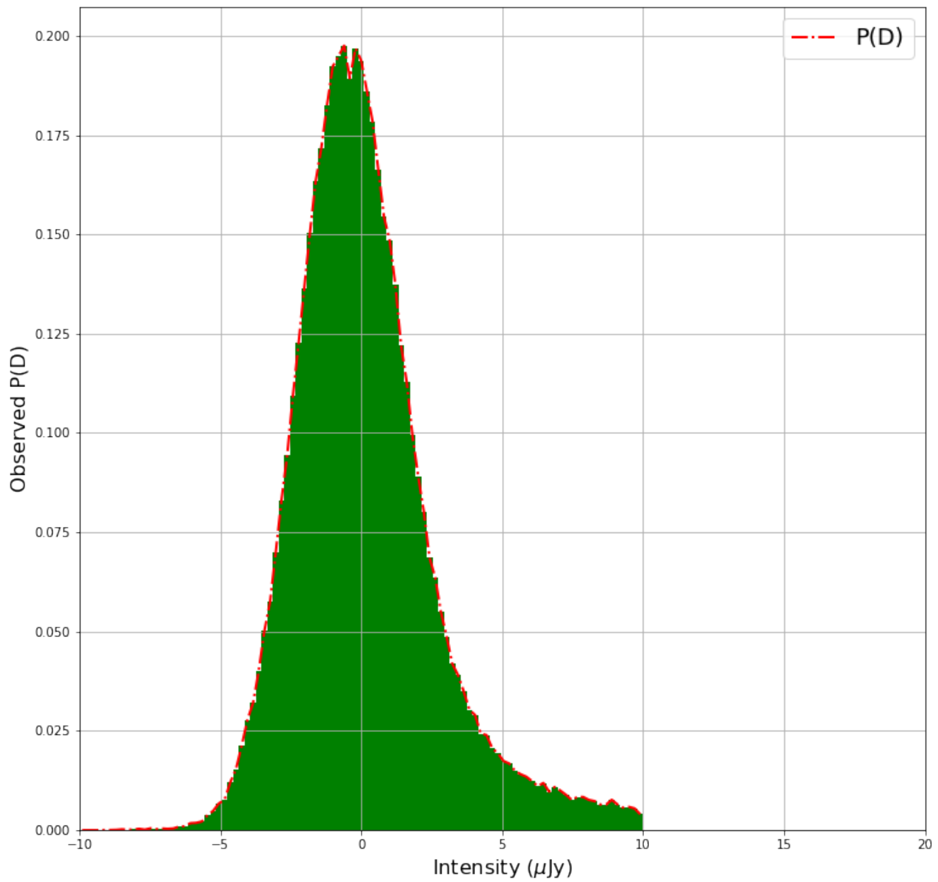

Figure 2. The distribution of the central 5' of pixels of the CCP dataset. The red dotted line represents the counts at each bin height and average bin step, which is then used as the data for the model fitting. The domain of the observed P(D) used for this analysis is -10 μJy to 10 μJy.

The MCMC takes in 4 arguments: 1. The modeled function, 2. The likelihood function, 3. The priors, 4. The probability check. As stated, the function that is used to fit to the data is Equation (19). The likelihood function is implemented as a χ^2 logarithm function, where it being a logarithm explained in the documentation for `emcee` (Foreman-Mackey et al. (2013)). The likelihood function used can be referred to in the appendix, within the included code. The priors are simply the designated ranges we allow the 'walkers' to explore for each parameter. We allowed four free parameters, α, γ, μ, and σ_n to be explored. Though it was stated prior that μ was arbitrary in extracting the source density, it is ultimately allowed to be a free parameter which is briefly explained in Section 4.2.4. For reasons explained in Section 2, we allowed $0 < \alpha < 2$ and $\gamma > 0$. μ was allowed to shift in the range $-10 < \mu < 10$, and σ_n was allowed to explore $.8 < \sigma_n < 1.8$, which is the realistic range of expected image noise values. Finally, the probability check is used to check if the current iteration of the MCMC follows the priors. By running the MCMC, we obtained results and fined tuned the code and the amount of iterations to obtain the most accurate results while avoiding issues, which is the basis of the next section.

4.2. *Arising Problems and Solutions*

Writing code is no easy task, especially one with a complicated function. Many issues we ran into encouraged better solutions to obtain more accurate results. Some of the issues that we shortly discuss here involve complications with the Lévy function, convolution, and more. It is documented here to clearly address the major issues that can be avoided in the future with this research or related applications.

4.2.1. *Lévy Selection*

We first attempted to use the functionally equivalent form of the Lévy function, but we ran into issues where Python would not output the desired form. Due to the Lévy function not having any analytical form from its characteristic function, Python had trouble manually integrating the numerical integral with any computational method. We sought to use a package that cleanly addressed it, first trying out `pylevy`. Whether it was a client error or package error, the package could not be successfully convolved with a Gaussian for the model. Since we utilized `scipy` to render the Gaussian curve, we sought to use `scipy.stats.levy_stable`, ultimately solving this issue.

4.2.2. *Working in the μJy Regime*

When attempting to create the observed P(D) to fit the model function to, Python had issues running such small numbers ($< 10^{-6}$) in a histogram. The simple solution to this was to work in the μJy regime, effectively multiplying the entire dataset by 10^6, hence rendering Equation (16) relevant to convert the final output back to the Jy regime.

4.2.3. *Convolution Choice and Range*

Within the MCMC, the domain used to fit the observed data with the model function is defined when calling the domain of the observed data. The primary data we use has a domain of -10 μJy to 10 μJy, but the emphasis on this is that the domain is strictly set to $+/-10$, or the mean of the domain always has to be 0. This is the result of the definition of convolution, where the best analogy is the rolling average of the two functions results in the domain of the output becoming the merged domains of the input functions. Without the mean of the domain being 0, the mean of the domain of the resulting convolved function will not be based around 0 either, allowing extreme values of α, γ, and σ_n to be explored that may be extremely off the truth.

4.2.4. *Localization*

Other than what was stated in Section 4.2.3, we allowed μ to be a free parameter, despite it being an arbitrary variable. This is to counteract the integration effects of the dataset being CLEANed, and therefore, shifted artificially. This P(D) method strongly relies on the distribution being where it theoretically should lie: the Gaussian image noise should theoretically have a mean of 0, and the Lévy function describing the underlying source population should be non-negative. In reality, the observed data is shifted to the left, which can cause unwanted effects on the fitting. Therefore, by letting μ be a free parameter, the distribution has another degree of freedom so that the shape of the modeled distribution, defined strictly by α, γ, and σ_n, have more opportunities to have a better matching fit with each iteration's 'walkers' in the MCMC.

4.2.5. *NaNs*

Another things added was an additional domain check within the probability check in the MCMC. A full run of the MCMC would take up to 24 hours, and it would often crash halfway through. Through debugging, some values of potential priors within the correct domain would not work with the model and return an array full of 'not a number' (NaNs). By passing a NaN check in the probability check to see if the iteration's model was valid, the MCMC ended up running smoothly.

4.3. *Results*

Figure (3) shows the resulting sample chain from the best trial, meaning the best P(D) fit. The MCMC automatically discarded the first 50 iterations and started to record the chain from then, adding in a soft burn-in discard. A further 150 iterations were discarded to where the walkers seemed to be random regardless of the initial starting value. The rest of the iterations were then plotted on a corner plot, shown in Figure (3), highlighting the densities of the parameter space. The 50th quantile values from the chain are displayed at the top of each column, with the 34th and 66th quantiles as errors. The 50th quantile was chosen as the best display for each parameter because, due to the nature of randomness within the MCMC, the walkers should, in theory, walk around the truth, with it being the mean given enough walkers and iterations.

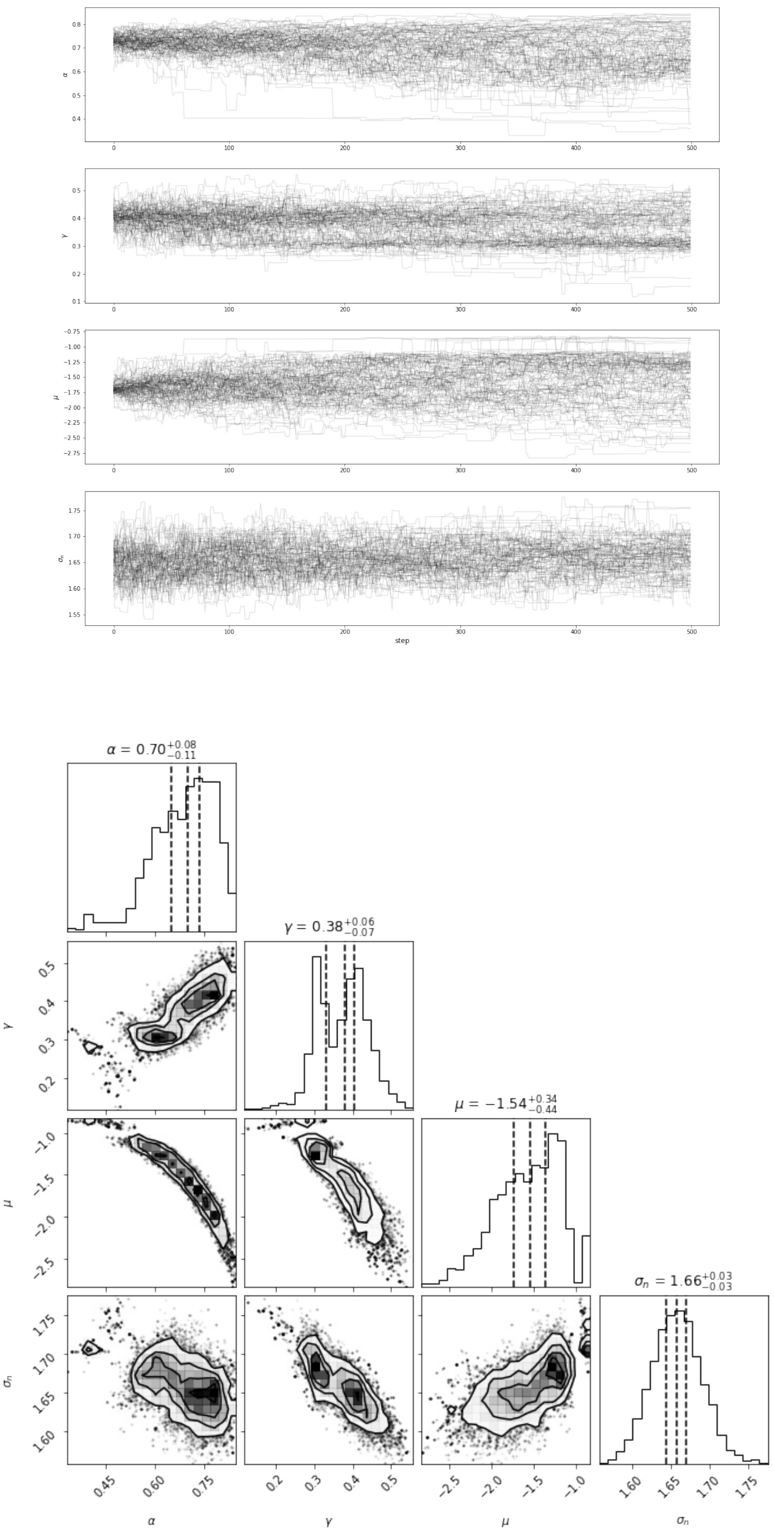

Figure 3. Top: Sample chain of this MCMC run with 500 iterations of 100 walkers. Upon visual inspection, we discarded the first 150 iterations as the burn-in. Bottom: Corner plot of the free parameters. $\alpha + \gamma$ and $\gamma + \sigma_n$ have apparent bimodality, and $\alpha + \mu$ have a very tight relationship. The optimized parameters, determined by selecting the 50th quantile values, are listed at the top. The errors are determined by the 34th and 66th quantiles.

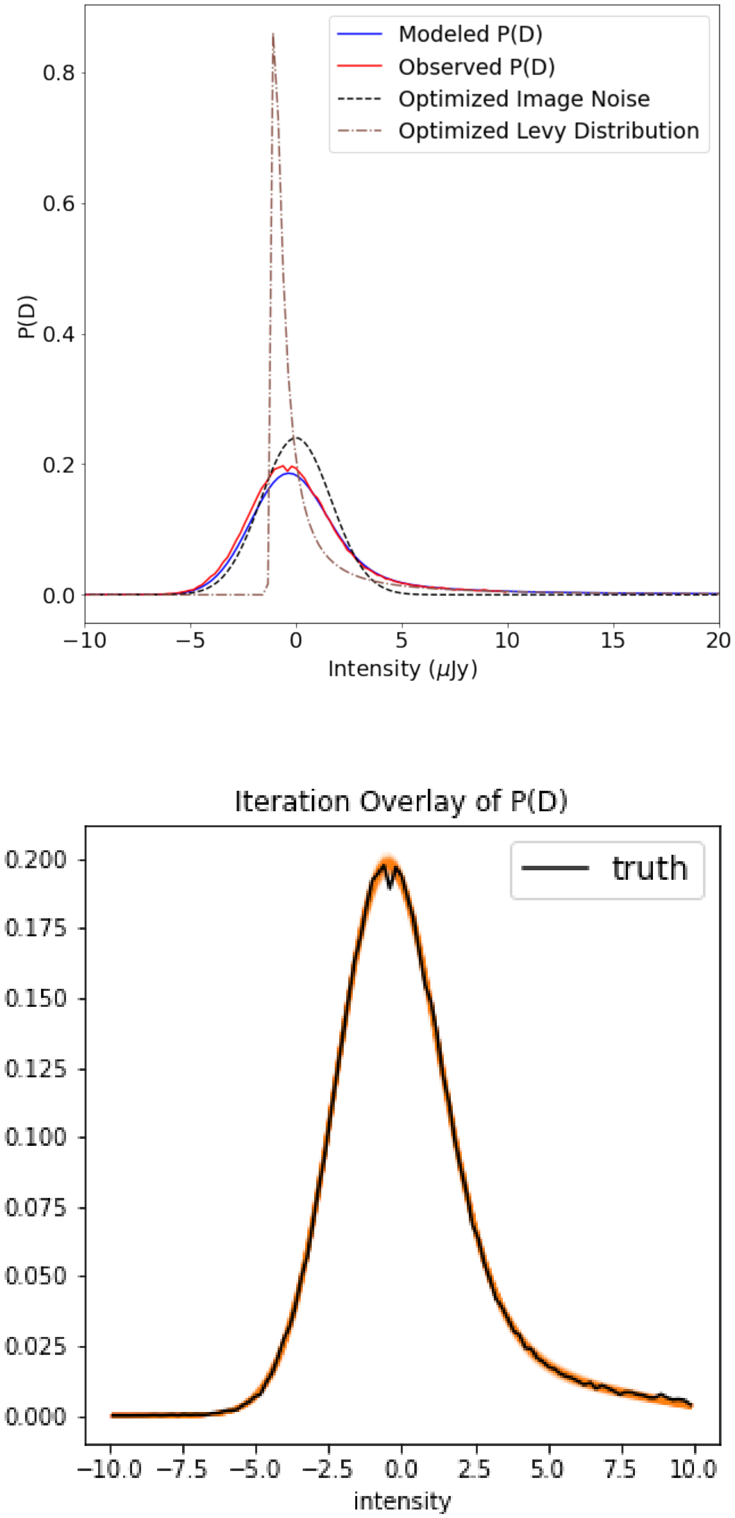

Figure 4. Top: The optimized image noise, Lévy distribution, and P(D) distribution compared to the observed P(D) (data). The modeled P(D) has a pretty fair fit with the data, but has room to improve. Bottom: 100 randomly selected iterations from the chain and overlaid over the observed P(D). It is clear that some of the iterations match extremely well with the data.

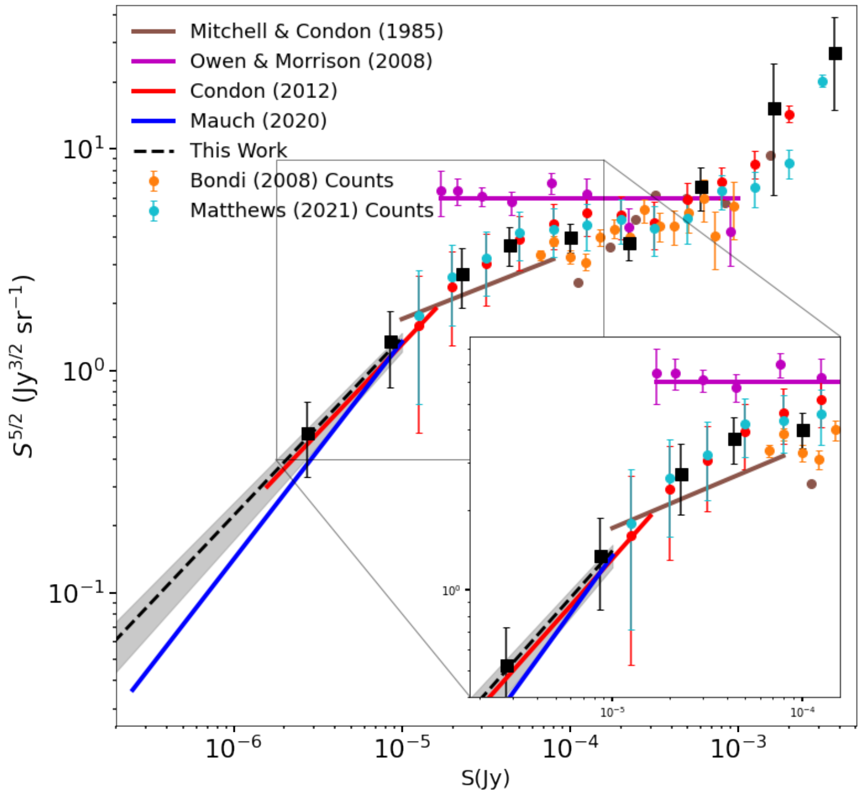

Figure 5. The 1.4 GHz differential source count population. The zoomed in overlay highlights the merge between P(D) analyses and distinct counts, as well as the minute density shifts of the counts.

Given the nature of the relationships between each parameter, namely α, γ, and σ_n, we sought to obtain errors based on the pairing of parameters in each iteration, rather than using each parameter's individual error obtain in Figure (3). By flattening the chain and obtaining the 34th, 50th, and 66th quantiles overall, we finally obtain the optimized parameters and the error range: $\alpha = 0.699^{0.036}_{-0.048}$, $\gamma = 0.379^{0.027}_{-0.047}$, $\mu = -1.536^{0.180}_{-0.196}$, and $\sigma_n = 1.658^{0.013}_{-0.013}\mu$Jy. Now using Equations (14) and (16), we obtain $N_0 = 14221$ Jy$^{0.699}$ and $\kappa = -1.699$. We now have our optimized underlying population for a power-law universe: $\frac{dN}{dS} = 14000S^{-1.7}\text{Jy}^{-1}\text{sr}^{-1}$.

5. DISCUSSION

5.1. *P(D) Analysis*

Figure (4) shows the optimized P(D), along with the iteration evolution of the P(D) fitting. The optimized Lévy distribution is shown as the brown dashed line, and is noticeably shifted by the μ parameter. The optimized image noise is shown as the black dashed line, and is centered with a mean of 0, following the definition of noise. Convolving the two functions together created the optimized P(D) (the blue solid line), which nicely overlays with the data, or the observed P(D) (the red solid line). The fitting is not perfect, due to a couple of factors. First off, by limiting the range to 10μJy, there are fewer bins and less of the positive right tail shown to allow the χ^2 to better optimize the modeled function. This is discussed in more depth in Section 5.3. To combat the shift of the data by post-processing and CLEANing, we allowed μ to be a free parameter to allow more freedom to shape the model, but this shift was not accounted for when optimizing the image noise parameter. This let the optimization of the Lévy distribution to overcompensate for the skewness and the shift in the observed data in order to better match it, hence causing an error. This systematic error in the slight shift is further discussed in Section 5.4. The bottom figure in Figure (4) shows 100 randomly picked iterations from the flatted chain (containing 35,000 samples after discarding the burn-in) and plots it along with the P(D) truth. It's clear that almost every iteration present well match the observed P(D), contrary to the modeled P(D), which leaves room for massive improvement and tweaking.

5.2. *Radio Population*

Figure (5) shows the 1.4 GHz differential source counts with the static Euclidean normalization ($S^{5/2}\frac{dN}{dS}$). Other literature's counts are referenced for comparison as well, including Mitchell & Condon (1985) counts and power law $\frac{dN}{dS} = 54S^{-2.2}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$, Owen & Morrison (2008) counts and power law $\frac{dN}{dS} = 6S^{-2.5}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$, Condon et al. (2012) counts and power law $\frac{dN}{dS} = 9000S^{-1.7}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$ converted from 3.02 GHz to 1.4 GHz, Mauch et al. (2020) power law $\frac{dN}{dS} = 107000S^{-1.52}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$, Bondi et al. (2008) counts, and Matthews et al. (2021a) counts. Power laws that were converted to $\nu = 1.4$ GHz for comparison utilized:

$$\left(\frac{N_{0,1}}{N_{0,2}}\right) = \left(\frac{\nu_1}{\nu_2}\right)^{\langle\alpha\rangle(\kappa-1)} \tag{20}$$

where $\langle\alpha\rangle$ is the spectral index, which is approximately equal to $\langle\alpha\rangle = -0.7$ (Condon et al. (2012)). All of the counts that were translated to 1.4 GHz sourced from differing frequencies and from brightness-weighted to Euclidean-weighted were converted using:

$$\log\left[S^{5/2}n(S)\right]_{1.4GHz} = 0.5\log(S) + \log\left[S^2 n(S)\right]_\nu + \langle\alpha\rangle\log\frac{1.4}{\nu} \tag{21}$$

where ν is the original counts' frequency and $\langle\alpha\rangle$ is, again, the spectral index of $\langle\alpha\rangle = 0.7$ (Matthews et al. (2021b)).

μJy	$S^{5/2}\frac{dN}{dS}$	$S^{5/2}\frac{dN}{dS}$ Err.
2.740	0.5273	0.1971
8.560	1.3547	0.5102
22.829	2.7352	0.8130
43.580	3.6960	0.7351
99.758	3.9985	0.6025
222.009	3.7611	0.6209
597.058	6.7524	1.4818
1619.599	15.2272	9.0361
3700.841	27.2310	12.1991

Table 1. Table of discrete counts from the CCP dataset. Obtained from the CCP Collaboration (Luber et al. (2023), in preparation).

The distinct counts of the CCP are also shown in Figure (5) as black squares, and the specific values are provided in Table (1). These counts were calculated from the CCP Collaborative Group (Luber et al. 2023, in preparation) through a separate process, and they line up extremely well with the extracted underlying $\frac{dN}{dS}$, providing a strong argument to the validity and accuracy of this alternative P(D) approach. Our $\frac{dN}{dS}$ also lines up and fairly agrees with Mauch et al. (2020) and Condon et al. (2012), where our optimized source density is a little denser than both works. The density also lines up well for the static evolution of flux density along the brighter end, specifically towards the 'knee'. An extrapolative thought is, since it is very apparent that no part of the differential source count trends is strictly linear, a power-law model for underlying source population for confused sources is not the best approximation, and P(D) analysis would well benefit from a varying model, such as a broken power law (Vernstrom et al. (2014)). This suggestion could provide an answer to the varying power-laws shown, which could be dependent on the limited range for which P(D) is applied to.

5.3. *Limitations of the Range*

We performed an exploratory test to see the impact of changing the upper limit for P(D), due to how power-law approximations can only describe these confusion-limited sources to a certain extent. Figure (6) shows the 1.4 GHz differential sources counts with adjusted upper limits of 60μJy and 100μJy. The P(D) analysis for the 60μJy cutoff has power-law $\frac{dN}{dS} = 5360S^{-1.8}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$, and the 100$\mu$Jy cutoff has power-law $\frac{dN}{dS} = 6415S^{-1.794}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$. These are quite different from our original extracted power-law of $\frac{dN}{dS} = 14000S^{-1.7}\mathrm{Jy}^{-1}\mathrm{sr}^{-1}$, analytically and visually. The

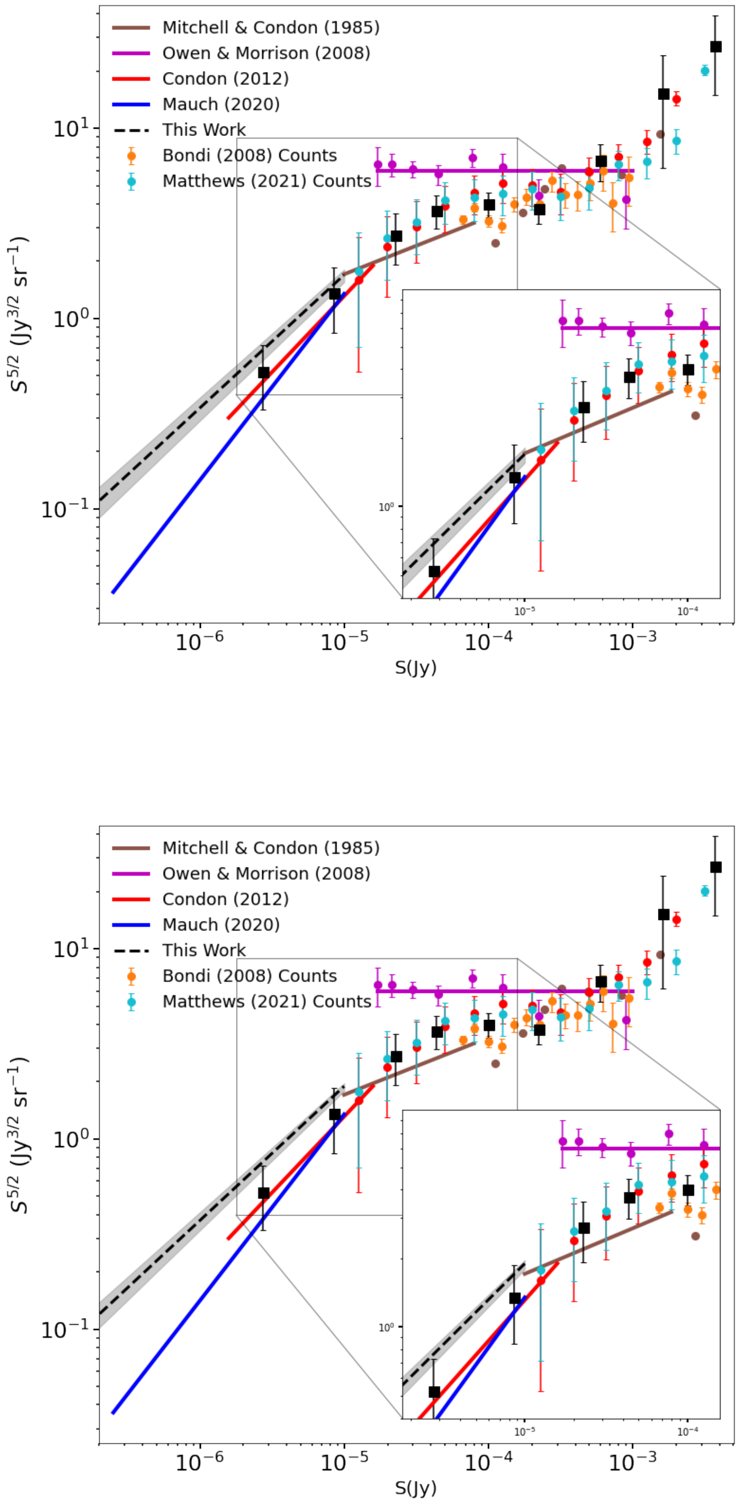

Figure 6. The 1.4 GHz differential source count population with varying upper limits on the range. Top: Instead of performing P(D) analysis with a 10μJy upper limit, it is shown here with a 60μJy limit (as the black dotted line), though it is not physically extended here. Bottom: $\frac{dN}{dS}$ is shown with a 100μJy upper limit.

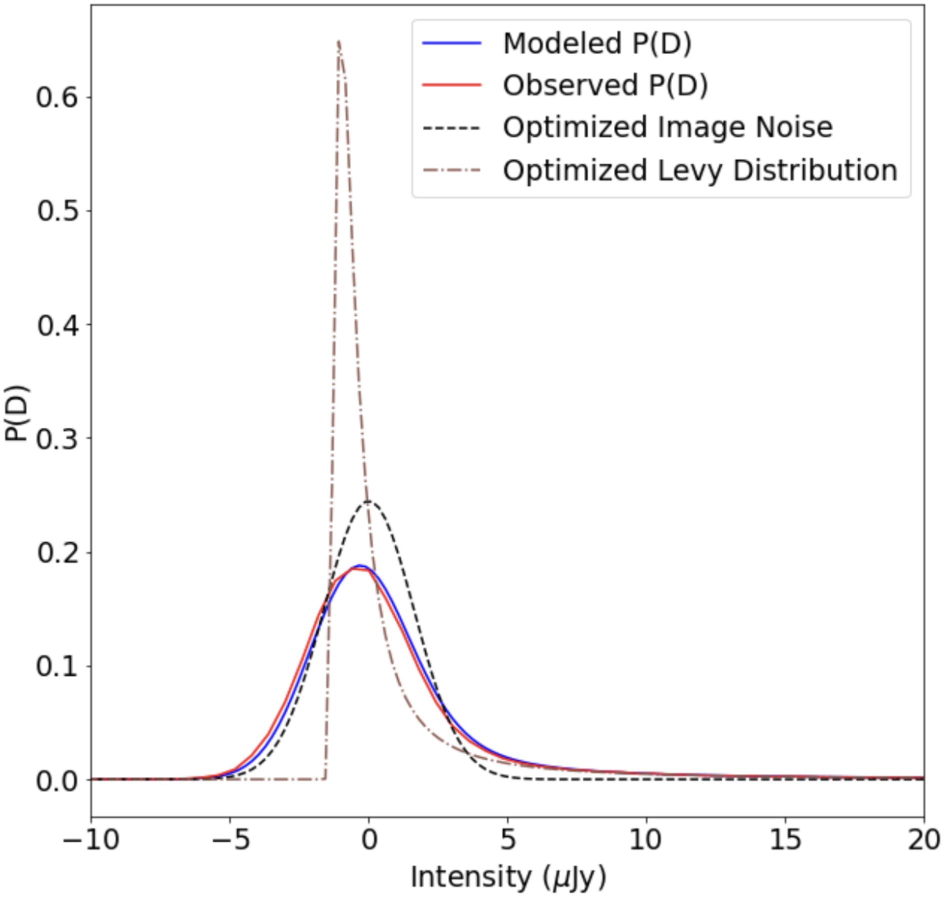

Figure 7. Optimized P(D) with a 60 μJy cutoff. Compared to Figure (4), the P(D) fitting (in blue) are arguably better shaped distribution, with a minor phase difference being the only issue with a perfect fit.

60μJy cutoff is significantly more dense than the 10μJy approximation, and the 100μJy cutoff is slight more dense than that of the 60μJy cutoff. Generally speaking, the population became less steep with Euclidean-weights as the limit was raised, and also apparently got more dense, comparatively. This generally makes sense, as the higher the upper limit on the range becomes, the more sources and their point source function (PSF) effects on the pixels are included.

Although an intriguing result, this light takeaway should be taken with a grain of salt. Referring back to Figures (2) and (4), and the method which MCMC performs the optimization, limiting the range isn't so intuitive at first. In a brief sense, the allowed domain of the data that is sent to the MCMC is utilized only for optimizing a matching shape for the convolved function. Since the Lévy distribution is responsible for the heavy positive-tail, cutting off more of the domain on the right leaves less room for comparison for the MCMC by use of its χ^2, and vice versa, allowing a larger domain to be fit would give more definition to the Lévy part of the function, as opposed to the Gaussian noise fitting. In a sense, this is what we see here in Figure (7). Here, the optimization for P(D) is much better for the 60μJy cutoff than the original fitting for the 10μJy cutoff. In this sense, we ask, is using the same range as other recent literature (specifically, the same cutoff as Mauch et al. (2020)) while having a little 'worse' of a P(D) fitting more feasible to use, or is a better pure fit of P(D) something to prioritize more? With the current understanding of how the MCMC performs the optimization, regardless of the final outcome of how well it fits in comparing to the literature's results, we cannot confidently say which one is preferred and what this science means, and deserves more careful thought and research to understand the impact of this.

5.4. *Future Implementations*

A number of improvements can be made to better this method and the results. One such example is integrating better analytical reasoning behind choosing how many iterations to cut off for the MCMC burn-in, rather than visually deciding and approximating. Within the documentation for emcee, they decide to quantify it using the estimated integrated autocorrelation time, meaning that statistically, it will take 'x' amount of steps to 'forget' where the MCMC started due to the priors. In other words, the iterations past this calculated specific iteration are not statistically affected by the initial values for each parameter. This would ensure that there are no biased iterations passed to the final sampling when choosing the best parameter values.

Another improvement that can be made also relates to selecting a better choice for the final chosen values: developing a better method to extract parameters that leads to a better P(D) fitting. In Figure (4), it is visually apparent that all 100 randomly chosen samples from the flat chain match up with the data near perfectly. Despite this, the 50th quantile value, which statistically would be the best choice given enough samples and iterations, resulted in a relatively poor P(D) fitting. This calls for two actions: allowing longer runtime and iterations (which was already strenuous on a personal laptop; it took 25 hours on a 2020 M1 MacBook Pro), and/or better method to extract the best parameter values.

Another issue that can be address is the shift of the data distribution due it being CLEANed. It was softly counted by allowing μ as a free shift parameter for the Lévy distribution, but since the Gaussian was not shifted, the shift in the Lévy distribution could lead to unwanted parameter alterations. To combat this, allowing the same exact shift in the Gaussian would allow more accurate and realistic values to extract a better optimized descriptor for the underlying confused source population.

6. CONCLUSION

This new method of approaching P(D) is promising and likely a very reliable method of extracting the differential source count population. Due to this method being new, there are many ways to improve it, and this first realization of applying this method is no exception. Applying methods incorporated from Condon (1974), Hewish (1961), and Herranz, D. et al. (2004), it is no coincidence that this method resulted in very accurately describing the radio differential confused source counts, along with matching up quite well with the CCP distinct counts.

For the next steps in research within radio galaxy evolution and cosmology, the next clear step is to implement the findings from P(D) to methodically determine the evolving local radio luminosity function (LF). Shown in Mauch et al. (2020), Béthermin et al. (2012), and Béthermin, M. et al. (2011), the local radio LF can be constrained utilizing the counts from $\frac{dN}{dS}$ in order to best determine the factor of redshift z evolution. For more information on extracting the counts from an evolutionary model reliant on z, refer to Mauch et al. (2020) and Madau & Dickinson (2014).

REFERENCES

Béthermin, M., Daddi, E., Magdis, G., et al. 2012, The Astrophysical Journal, 757, L23, doi: 10.1088/2041-8205/757/2/l23

Béthermin, M., Dole, H., Lagache, G., Le Borgne, D., & Penin, A. 2011, A&A, 529, A4, doi: 10.1051/0004-6361/201015841

Bondi, M., Ciliegi, P., Schinnerer, E., et al. 2008, The Astrophysical Journal, 681, 1129, doi: 10.1086/589324

Casey, C. M., Narayanan, D., & Cooray, A. 2014, Physics Reports, 541, 45, doi: 10.1016/j.physrep.2014.02.009

Condon, J. J. 1974, ApJ, 188, 279, doi: 10.1086/152714

Condon, J. J., Cotton, W. D., Fomalont, E. B., et al. 2012, ApJ, 758, 23, doi: 10.1088/0004-637X/758/1/23

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

Herranz, D., Kuruoglu, E. E., & Toffolatti, L. 2004, A&A, 424, 1081, doi: 10.1051/0004-6361:20035858

Hewish, A. 1961, MNRAS, 123, 167, doi: 10.1093/mnras/123.2.167

Kellermann, K. I., & Verschuur, G. L. 1988, Galactic and Extragalactic Radio Astronomy

Madau, P., & Dickinson, M. 2014, Annual Review of Astronomy and Astrophysics, 52, 415, doi: 10.1146/annurev-astro-081811-125615

Matthews, A. M., Condon, J. J., Cotton, W. D., & Mauch, T. 2021a, The Astrophysical Journal, 914, 126, doi: 10.3847/1538-4357/abfaf6

—. 2021b, The Astrophysical Journal, 909, 193, doi: 10.3847/1538-4357/abdd37

Mauch, T., Cotton, W. D., Condon, J. J., et al. 2020, The Astrophysical Journal, 888, 61, doi: 10.3847/1538-4357/ab5d2d

Mitchell, K. J., & Condon, J. J. 1985, AJ, 90, 1957, doi: 10.1086/113899

Owen, F. N., & Morrison, G. E. 2008, The Astronomical Journal, 136, 1889, doi: 10.1088/0004-6256/136/5/1889

Scheuer, P. A. G. 1957, Mathematical Proceedings of the Cambridge Philosophical Society, 53, 764–773, doi: 10.1017/S0305004100032825

Vernstrom, T., Scott, D., Wall, J. V., et al. 2014, Monthly Notices of the Royal Astronomical Society, 440, 2791, doi: 10.1093/mnras/stu470