Regularized Maximum Likelihood Techniques for ALMA Observations

Brianna Zawadzki 1, 2, 3, Ian Czekala 2, 3, 4, 5, Ryan A. Loomis 1, Tyler Quinn 1, Hannah Grzybowski 1, Robert C. Frazier 1, and Yina Jian 5

1 Department of Astronomy and Astrophysics, 525 Davey Laboratory, The Pennsylvania State University, University Park, PA 16802, USA
2 Center for Exoplanets and Habitable Worlds, 525 Davey Laboratory, The Pennsylvania State University, University Park, PA 16802, USA
3 Institute for Computational & Data Sciences, The Pennsylvania State University, University Park, PA 16802, USA
4 Center for Astrostatistics, 525 Davey Laboratory, The Pennsylvania State University, University Park, PA 16802, USA
5 National Radio Astronomy Observatory, 520 Edgemont Rd., Charlottesville, VA 22903, USA

ABSTRACT

Regularized Maximum Likelihood (RML) techniques are a class of image synthesis methods that have the potential to achieve improved angular resolution and image fidelity compared to traditional image synthesis methods like CLEAN when applied to sub-mm interferometric observations. We used the GPU-accelerated open source Python package MPoL to explore the influence of various RML prior distributions (maximum entropy, sparsity, total variation, and total squared variation) on images reconstructed from ALMA continuum observations of the protoplanetary disk hosted by HD 143006. We developed a K-fold process for the image validation procedure cross-validation (CV) and explored both uniform and “dartboard” styles of visibility sampling within the validation process. Using CV to find optimal hyperparameter values for the test case of total squared variation regularization, we discovered that a wide range of hyperparameter values (spanning roughly an order of magnitude) correspond to models with strong predictive power for visibilities across unsampled or sparsely sampled spatial frequencies. We also provide a comparison of RML and CLEAN images for the protoplanetary disk around HD 143006, finding that RML imaging improves the spatial resolution of the image by about a factor of 3. Lastly, we distill general recommendations for building an RML workflow for image synthesis of ALMA protoplanetary disk observations, including recommendations for incorporating CV most effectively. Using RML methods to improve the resolution of protoplanetary disk observations will enable new science requiring high resolution images, including the detection of protoplanets embedded within disks.

Keywords: protoplanetary disks — submillimeter astronomy — radio interferometry — deconvolution

1. INTRODUCTION

Protoplanetary disk observations provide critical insight into disk properties like temperatures and densities which can be used to better understand the planet formation process as a whole. Observations of such disks have supported theoretical models of grain growth, planetesimal and planet formation, and the emergence of disk substructures (e.g. Isella et al. 2010, 2016; Pérez et al. 2012; ALMA Partnership et al. 2015; Pérez et al. 2015; Andrews et al. 2016; Cieza et al. 2016, 2017; Zhang et al. 2016; Pinte et al. 2018; Tripathi et al. 2018; Casassus et al. 2021; Bae et al. 2022). In recent years, interferometric observations that achieve high angular resolution (some down to scales of only a few AU, such as DSHARP; see Andrews et al. 2018; Huang et al. 2018) have contributed to an ever-growing catalog of highly resolved protoplanetary disks. Making further progress requires disk images at smaller spatial scales, with corresponding advancements in image fidelity. As these smaller spatial scales are resolved, our ability to characterize the dust and gas substructures of protoplanetary disks will improve, including the ability to detect signatures of forming planets embedded within their disks.

The Atacama Large Millimeter/submillimeter Array (ALMA) is capable of observing sources at high angular and spectral resolution. With its 66 12-m antennas and baselines of up to 16 km, ALMA can achieve 20 mas
resolution at 230 GHz. In general, interferometers are composed of a number of individual antennas, with each pair of antennas defining a baseline. Because of practical limitations in the number and placement of antennas as well as observing time, every possible baseline length will not be represented during an observation. As a result, interferometers incompletely and noisily sample the visibility function of an astronomical source, given by

\[ V(u, v) = \int \int I(l, m) \exp\{-2\pi i(ul + vm)\} \, dl \, dm \]  

(1)

Here, \( V(u, v) \) is the visibility function parameterized by spatial frequencies \( u \) and \( v \), and \( I(l, m) \) is the sky brightness distribution, where \( l = \sin(\Delta \alpha \cos \delta) \) and \( m = \sin(\Delta \delta) \) for right ascension \( \alpha \) and declination \( \delta \). The visibility function \( V \) and the sky brightness distribution \( I \) are related by Fourier transform. Any interferometric observation has an inherently incomplete sampling of spatial frequencies \( u, v \) regardless of the array configuration, with the primary data product from the interferometer being a set of visibility measurements measured at coordinates \( (u_k, v_k) \).

Interferometric images are synthesized from the observed visibility data, and the final image product depends on which assumptions are made about the unsampled spatial frequencies. If all unsampled spatial frequencies are set to zero power, the inverse Fourier transform of the visibilities delivers what radio astronomers call the dirty image. The dirty image can be thought of as a convolution of the true sky brightness distribution and the instrument point spread function (PSF), or dirty beam (Högbohm 1974). Because beam sidelobes add image artifacts that are not representative of the true source sky brightness, dirty images require processing to better reconstruct the sky brightness. A more detailed overview of the general imaging process is provided in Thompson et al. (Ch. 10.11; 2017).

CLEAN is currently one of the most popular and well-supported imaging methods in the community (McMullin et al. 2007). CLEAN begins with the dirty image and iteratively deconvolves beam sidelobes while building up a model representation of the sky brightness (Högbohm 1974). The model is built from CLEAN components, usually Dirac \( \delta \)-functions or Gaussians, which are placed at the location of the brightest pixel in the dirty image. Deconvolution occurs when the CLEAN component is convolved with the dirty beam and subtracted from the dirty image. This process repeats until either a certain number of iterations has been reached, or the dirty image reaches some noise threshold. There are two products at the end of the CLEANing process: the residual image (originally the dirty image, but now contains only residuals after deconvolution) and the CLEAN model (composed of CLEAN components). The CLEAN model is then convolved with the CLEAN beam, which is usually a Gaussian fit to the main lobe of the dirty beam. Finally, the residual image is added to obtain the final CLEANed image (Ch. 11.1; Thompson et al. 2017).

Although CLEAN has long been a reliable way to process images, it does have limitations. CLEAN components are typically simplistic (e.g. a Gaussian), which may not be suitable for capturing certain morphologies, such as sharp edges or rings. Adaptations to CLEAN, such as a multi-scale approach that uses different component sizes, can yield an image that is a more realistic representation of an extended source (Cornwell 2008).

Regardless of the chosen variant of CLEAN, it is common practice to convolve the CLEAN model with a final restoring beam. This convolution makes the CLEANed image more visually pleasing, but generally acts as a low pass filter, spatially broadening any high resolution information in the CLEAN model. In other words, convolution with the CLEAN beam imposes a resolution limit on the final image. Other drawbacks include the computational speed; CLEANing even a small image can take several hours, while other image synthesis procedures developed with more modern computational infrastructure in mind can often synthesize an image at least an order of magnitude faster. Lastly, CLEAN is a nonlinear procedural method of image restoration rather than a true optimization algorithm. There are many user-set algorithm parameters that affect the outcome of the CLEANing process (e.g. in the Common Astronomy Software Applications (CASA) tclean implementation (McMullin et al. 2007): stopping criteria, masks that limit where CLEAN components may be placed), many of which do not have a clear best choice that corresponds with the image qualities needed or desired for a given science case.

Regularized maximum likelihood (RML) methods are an alternative class of imaging techniques that cast image synthesis as a forward-modeling problem. The goal is to maximize the likelihood of measuring a set of visibility data, given a set of predicted model visibility values. Regularization refers to the inclusion of additional information in the model that influences the values of the model visibilities, effectively making assumptions about the source. RML techniques have been well-established in the field for decades (e.g. Ponsonby 1973; Ables 1974; Cornell & Evans 1985; Narayan & Nityananda 1986), and have been shown to achieve higher angular resolution than CLEAN. With modern computing abilities, RML techniques for imaging have
become powerful alternatives to more traditional deconvolution procedures like CLEAN for sub-mm continuum observations (e.g. Cárcamo et al. 2018; Event Horizon Telescope Collaboration et al. 2019).

A well-known example is the analysis of the Event Horizon Telescope images synthesized from observations of M87. Event Horizon Telescope Collaboration et al. (2019) successfully used two independently-developed RML pipelines to obtain high-resolution images of M87. In order to directly compare with CLEAN results, the RML images were restored with beams large enough to reduce the resolutions to that of the CLEAN image. The blurred RML images characterized the ring diameter and asymmetry consistently with the CLEAN image, showing that RML methods can produce higher resolution images at similar image fidelity requirements. Image fidelity was ensured by using the RML pipelines on synthetic data from models, for which the ground truth image was known. RML imaging techniques have also contributed to analyses of ALMA continuum observations of protoplanetary disks. For example, Cárcamo et al. (2018) and Pérez et al. (2019) used RML with entropy-based regularizers on ALMA continuum observations of protoplanetary disks HL Tau and HD 169142, respectively, finding that RML methods can not only achieve better resolution than the corresponding CASA tclean image, but also suppress background noise more effectively. Yamaguchi et al. (2020) applied RML imaging techniques with sparsity and total squared variation regularizers to ALMA observations of the protoplanetary disk around HD 142527, finding that RML techniques result in improved image fidelity and higher angular resolution compared to CLEAN images.

In this paper we explore the effects of different regularizers for the case of imaging the protoplanetary disk around HD 143006. This paper is organized as follows. We describe the data used in this study and how it was prepared for RML imaging in section 2. In section 3 we discuss the theory of RML imaging, including the mathematical forms of various regularizers, a technical overview of the RML imaging Python package MPoL, and image validation procedures. We discuss the behavior and attributes of different regularizing terms in section 4, followed by a discussion on characterizing the resolution of RML images and a more thorough examination of image validation procedures in section 5. We present our conclusions in section 6. Appendix 6 contains recommendations for developing a successful RML workflow for ALMA measurement sets.

Throughout this study, we used two reference ALMA visibility datasets. The first is a small, mock dataset created from the ALMA logo. Its small size means that it can be easily stored and processed on servers with limited computational means. A full accounting of the processing steps are available on the mpoldatasets repository, we briefly summarize them here. The logo was converted to a greyscale image, Fourier transformed, apodized with a Blackman Harris window function (to remove spatial frequencies substantially higher than will be sampled by the target array), and saved as a FITS file. We then used simobserve (CASA version 6.1 McMullin et al. 2007) with the C43-7 reference ALMA configuration from simobserve (alma.cycle7.7.cfg), to “observe” the source as it transits zenith for 1 hour under median conditions.

The second dataset we used in this study is a real Band 6 ALMA dataset containing the observations of the protoplanetary disk hosted by HD 143006 obtained by the DSHARP survey (Andrews et al. 2018), achieving a resolution of 45 milliarcseconds. We chose this protoplanetary disk because of its public and well-known nature, as well as its potential for structures at small spatial scales, including azimuthal asymmetries. The visibilities were originally calibrated by the DSHARP team following the standardized CASA procedures described in Andrews et al. (2018). The full listing of archival observations can be found in Andrews et al. (2018, Table 3), and the calibrated visibilities can be downloaded from the DSHARP archive.

We performed one additional step of calibration beyond that of the DSHARP team. We found that the definition of the visibility weights was not consistent across all of the archival datasets, most likely because the treatment of statistical weights frequently changed in 4.x versions of CASA used to originally calibrate the visibilities. We found empirical weight scalings for each dataset by creating a tclean model, subtracting it from the visibilities, and examining the scatter in the visibility residuals compared to the Gaussian envelope expected from the thermal weights. Each spectral window was corrected individually, with a minimum scale factor of 1.47, a maximum of 1.47, and an average of 1.78 over all spectral windows. A walkthrough of this rescaling process is described as part of the MPoL doc-

1 Such as those used for continuous integration of code on GitHub, which is one reason why this dataset appears in many of the tutorials that accompany the MPoL documentation.
2 https://github.com/MPoL-dev/mpoldatasets
3 https://almascience.eso.org/almadata/lp/DSHARP/MSfiles/HD143006_continuum.ms.tgz
Before performing any imaging (either creating a dirty image or an RML image), we take the loose visibility data and average it to grid cells in the visibility domain. We specify the grid cells by first defining the spatial extent and desired number of pixels of the image. Then, we define a corresponding Fourier grid with the same number of grid cells as the image has pixels. The loose visibilities can be averaged using a simple weighted average, which is equivalent to uniform weighting. RML images always begin with uniformly weighted visibilities, as only uniform weighting retains the statistical properties of the data needed for forward modeling.

3. FORWARD MODELING WITH RML

The “maximum likelihood” part of RML refers to finding a set of visibilities that maximizes the likelihood function

$$\mathcal{L} = p(D | V).$$

The likelihood function expresses the likelihood of measuring a set of visibility data $D$, given a set of predicted model visibility values $V$. The model can be parameterized in a number of ways. It is possible to proceed with only a handful of parameters to describe the model, for example, a parametric model for a protoplanetary disk might be defined as a set of annular rings, each further specified by their radius and intensity (e.g. Zhang et al. 2016; Guzmán et al. 2018). Alternatively, a model could be parameterized by wavelet coefficients as in Carrillo et al. (2014).

However, we may not know enough about the source to make such model choices; in this case, a non-parametric approach can offer more flexibility during the imaging process. Consider an image with $N \times N$ pixels. Each pixel has some intensity $I_N$ such that the image is described by a set of pixel intensities,

$$I = \{I_1, I_2, \ldots, I_{N^2}\}.$$

We can use the intensity of each pixel in the image cube as model parameters, allowing each pixel to vary freely. In this case, the visibilities $V$ are deterministically calculated by using the Fourier transform of the image sampled at the range of baselines corresponding to the visibility data, such that $V \leftrightarrow I$. A non-parametric model introduces a great deal of flexibility into the imaging process, which can be harnessed to significantly improve image fidelity compared to a parametric fit (e.g. Jennings et al. 2020).

However, because interferometric data sets inherently present a missing data problem, simply maximizing the likelihood function is not enough to obtain a high fidelity image. There is a potentially infinite set of images that perfectly corresponds to the sampled visibilities, which includes the dirty image. This is where the “regularized” part of RML comes into play; in order to improve imaging outcomes, RML imaging aims to maximize the posterior distribution of the image, given the dataset and specified priors/regularizers. Starting with Bayes theorem and acknowledging that the model parameterization (e.g. number of pixels) will remain fixed, we obtain

$$p(V | D) \propto p(D | V) p(V)$$

where $p(V | D)$ is the posterior distribution to be maximized, $p(D | V)$ is the likelihood function, and $p(V)$ is the prior distribution.

We calculate the log likelihood for the sake of computational efficiency. Assuming that the noise is uncorrelated and follows a normal distribution with standard deviation $\sigma$, the natural logarithm of the likelihood function is

$$\ln \mathcal{L} = \ln p(D | V) = -N \ln(\sqrt{2\pi\sigma}) - \frac{1}{2} \sum_{i}^{N} \frac{(D_i - V_i)^2}{\sigma_i^2}.$$

Here, $D_i$ is a measured complex visibility at a $(u_i, v_i)$ point and $V_i$ is the predicted value of the model for the same $(u_i, v_i)$. Except for the factor of $1/2$, the rightmost term above is simply a $\chi^2$ distribution,

$$\chi^2(D | V) = \sum_{i}^{N} \frac{(D_i - V_i)^2}{\sigma_i^2},$$

and the log likelihood can be expressed as

$$\ln p(D | V) = -\frac{1}{2} \chi^2(D | V) + C.$$

We can now see that in order to maximize the log likelihood (which, if the priors are flat, will also maximize the posterior), it is necessary to minimize $\chi^2(D | V)$. Rather than maximizing the log likelihood, however, in

---

4 https://mpol-dev.github.io/visread/tutorials/rescale_AS209_weights.html
5 https://github.com/MPoL-dev/mpoldatasets/tree/main/products/HD143006-DSHARP-continuum
6 We commonly interchange the terms “prior” and “regularizer” depending on whether the presentation is primarily Bayesian or computational in nature, though both terms are practically synonymous.
computing it is more common to minimize the negative log likelihood, given by

\[ L_{\text{nlh}}(V) = -\ln p(D \mid V) = \frac{1}{2} \chi^2(D \mid V). \]  

(8)

In the machine learning community, it is common to focus on the optimization of some metric that can be described by a *loss function* (e.g. Bishop 2006; Hastie et al. 2009; Murphy 2012; Deisenroth et al. 2020). A loss function is some function which, when minimized, yields optimal parameter values. In the Bayesian framework we want to maximize the likelihood of our model visibilities, satisfying both the visibilities sampled by the interferometer as well as prior information included in the model. Because the machine learning framework is consistent with the Bayesian framework but more practical in how it relates to algorithmic implementation, we adopt the use of a loss function as the primary quantity to be minimized. It is well established that well calibrated data has Gaussian uncertainties, thus, we adopt the negative log likelihood as the first term in our loss function.

Though the negative log likelihood can function independently as a loss function, it provides no direct constraints on the image, yielding an unregularized fit. In radio interferometry, minimizing the negative log likelihood of the data alone often results in an undesirable image. This is due to the incomplete sampling of the visibility function at certain spatial frequencies; if the visibility function has significant power in \( u,v \) space that is unsampled or only sparsely sampled, a loss function with no regularization is not particularly useful because there exist many images with the same minimum loss value. As a result, the (dirty) image product is unlikely to be the best representation of the true sky brightness distribution.

Figure 1 shows the \( u,v \) sampling of the HD 143006 dataset alongside the dirty images made by gridding visibilities with both uniform and Briggs weighting. Uniform weighting yields constant weights within a grid cell, and usually results in an image with high resolution at the cost of sensitivity. Briggs weighting has an adjustable robust parameter which determines the balance between resolution and sensitivity, making it a popular choice for making a visually pleasing dirty image (Briggs 1995).

The visibility function likely has power at some of the unsampled spatial frequencies. While setting these unsampled but presumably nonzero visibilities to zero is a conventional and conservative imaging procedure, the resulting dirty images contain artifacts such as blotchy emission or a noisy background. One can mitigate the effects of incomplete visibility sampling by incorporating regularization into the loss function. Additional terms added to the loss function directly regularize the image, for example incorporating smoothness. The inclusion of these terms can greatly influence model visibilities at spatial frequencies not sampled by the interferometer itself, reducing the number of images that could correspond to the set of visibilities and lessening the inherently ill-conditioned missing data problem posed by the interferometer. For example, a loss function that includes regularization could be of form

\[ L(I) = L_{\text{nlh}}(I) + \lambda_A L_A(I) + \lambda_B L_B(I) + \ldots, \]  

(9)

with various loss functions depending on the needs of the specific problem. Each \( \lambda \) value allows the strength...
of each regularizer to be tuned. Tuning this parameter is important in order to prevent over-regularizing the model, as a poorly-weighted regularizing term will result in an image that is a poor match to the data. However, a loss function can include terms with a wide variety of functional forms, and whether a $\lambda$ prefactor is required simply depends on the functional form of each regularizer. Note that these regularizing terms correspond to priors in the Bayesian framework, as they both impose some existing knowledge or expectation about the source on the model (Sivia & Skilling 2006).

3.1. Regularizers

A regularizer can be thought of as a stabilizing force on the image parameters, for example, penalizing pixels that poorly match the expected sky brightness. Implementing regularizers in the imaging process effectively allows us to make assumptions about unsampled and noisily sampled frequencies based on our prior knowledge of the source, in many cases changing the loss function space to become convex and have one clear minimum corresponding to a specific image rather than many minima (and thus many images) that perfectly fit the sampled data. In practice, one may need to use a combination of several regularizers to obtain an image that best represents the true sky brightness.

Regularizers vary in their implementations and their potential effects on the image. Some regularizers can be imposed by construction. For example, certain parameterizations of $I$ may disallow negative surface brightness values. Other regularizers can be imposed via loss terms, computed directly as a function of the image pixels themselves or via some additional property derived from the image (e.g., the power spectrum). These additional loss terms will require their own strength prefactors ($\lambda_j$), which can be adjusted to balance the relative impact of each regularizer. Here we discuss the functional form and motivation for each regularizer we tested.

3.1.1. Image Positivity

The true surface brightness distribution of any astrophysical source will be strictly greater than or equal to zero intensity. This constraint is frequently violated by CLEAN-based imaging procedures, with many synthesized images containing negative pixels in noisy background regions. The physical constraint on image positivity can be naturally incorporated into an RML imaging framework via construction of the image parameterization $I$.

Rather than directly parameterizing $I$ using the set of pixel values $\{I_1, I_2, \ldots, I_N\}$, instead, we parameterize the pixel values using latent variables $\{X_1, X_2, \ldots\}$ which are then mapped $X_i \rightarrow I_i$ using a function with a strictly positive range. We chose the Softplus function

$$I_i = f_{\text{Softplus}}(X_i) = \log(1 + \exp(X_i)).$$

The Softplus function maps negative input values to small but positive nonzero output while leaving positive input values largely unchanged. $I_i = \exp(X_i)$ is another potential mapping function, however, we found the Softplus function hastened model optimization.

3.1.2. Maximum Entropy

Maximum entropy is one of the best-established regularizers for radio interferometric imaging, and has been shown to deliver images with better spatial resolutions than the CLEAN algorithm (Cornwell & Evans 1985; Narayan & Nityananda 1986). Maximum entropy regularization aims to find an image that 1) is consistent with all testable information (here, the visibilities sampled by the interferometer) and 2) is maximally non-committal to untestable parameter space (Ables 1974; Sivia & Skilling 2006).

Several different functional forms of the maximum entropy regularizers have historically been used, usually similar to either $\log I$ or $-I \log I$ (where the base of the logarithm could be any value including $e$). The latter is similar in form to statistical mechanics equations of entropy, but repurposed for information entropy (Shannon 1948). We follow the definition in Event Horizon Telescope Collaboration et al. (2019) and define maximum entropy loss as

$$L = \frac{1}{\sum_i I_i} \sum_i I_i \ln \frac{I_i}{p_i},$$

where $p_i$ is a reference pixel value against which other pixels are compared. The reference pixel values could be as simple as a “blank” image of uniform intensity (e.g. Cárcamo et al. 2018), or they could take additional knowledge about the source into account. For example, Event Horizon Telescope Collaboration et al. (2019) used circular Gaussian images for the sets of $p_i$.

Maximum entropy regularization inherently promotes image positivity because of the logarithm built into the functional form of the regularizer; only positive non-zero values $I_i$ result in a real and defined $\ln I_i$, so the regularizer favors positive pixel values (Narayan & Nityananda 1986; Högbom 1979). In addition, maximum entropy generally encourages uniformity in the image and in the errors, making it a useful regularizer for identifying the presence of features in the image (Högbom 1979; Gull & Daniell 1978).

Maximum entropy regularization also introduces the potential to achieve some degree of superresolution in
the RML image. Superresolution refers to an image resolution that is improved compared to another resolution standard, such as a Gaussian fit to the main lobe of the dirty beam (which is also usually the CLEAN beam). The potential for superresolution exists in maximum entropy regularization because the features of the chosen entropy function (e.g. concavity, change in slope) result in an image with sharpened peaks and flattened baseline oscillations (Narayan & Nityananda 1986). Sharper baseline oscillations dampen the blotchy imaging artifacts that stem from incomplete sampling of spatial frequencies, such as those seen in Figure 1.

3.1.3. Sparsity

Sparsity regularization uses the $L_1$ norm to promote an image that is a sparse collection of nonzero pixels. Derived from the least absolute shrinkage and selection operator (lasso, see Tibshirani (1996)), sparsity is a pixel-based regularizer that has successfully been applied to radio interferometric imaging to achieve high-resolution images around black holes and protoplanetary disks (e.g Honma et al. 2014; Akiyama et al. 2017a; Kuramochi et al. 2018; Event Horizon Telescope Collaboration et al. 2019; Yamaguchi et al. 2020).

We formulate the sparsity loss as

$$L = \sum_i |I_i|.$$  \hspace{1cm} (12)

Sparsity regularization reduces the amplitudes of unneeded pixels (i.e., promoting an image that is a sparse collection of nonzero pixels), making it a useful regularizer when the true sky brightness distribution of a source is likely to be sparse.

3.1.4. Total Variation

Total variation (TV) regularization promotes images with sharp edges at areas with significant changes in intensity, and relatively smooth areas in between. TV regularization applies the $L_1$ norm to the gradient image, that is, the changes in adjacent pixel intensities in the image. As a result, an image made with a strong TV term is likely to exhibit sparsity in the gradient image, as the sharp edges in the source morphology are the main contributing components. In other words, the TV regularizer promotes similarity (or smoothness) between adjacent pixels unless the difference is significant, in which case a change in intensity is clearly needed. TV regularization has been used with success on its own and in combination with other regularizers for astronomical interferometric imaging (e.g. Wiaux et al. 2010; Akiyama et al. 2017b,a).

Following Rudin et al. (1992), we define the TV loss as

$$L = \sum_{l,m} \sqrt{(I_{l+1,m} - I_{l,m})^2 + (I_{l,m+1} - I_{l,m})^2 + \epsilon}.$$ \hspace{1cm} (13)

The image has dimensions where $l$ corresponds to right ascension and $m$ corresponds to declination. The $\epsilon$ term is an optional softening parameter which determines how pixel-to-pixel variations within the image slice will be penalized. If adjacent pixels vary more than $\epsilon$ the total loss will greatly increase, so TV regularization favors minimal variation between adjacent pixels.

3.1.5. Total Squared Variation

The total squared variation (TSV) regularizer is a variant of the TV regularizer, still summing the brightness differences between adjacent pixels. However, by not taking the square root of the differences, the TSV prior results in images with smoother edges (Kuramochi et al. 2018). The TSV regularizer,

$$L = \sum_{l,m} (I_{l+1,m} - I_{l,m})^2 + (I_{l,m+1} - I_{l,m})^2,$$ \hspace{1cm} (14)

is functionally similar to the TV prior, except the expression inside of the summation has been squared.

3.2. Minimizing the Loss Function

Minimizing the loss function maximizes the likelihood function, giving a “best fit” image that can change based on what kind of regularization is implemented. There are a variety of optimization methods that can be used for this minimization problem, such as those that require computing first- or second-order derivatives (e.g. gradient descent algorithms, Newton’s method) or those that attempt to minimize a function without computing gradients. We use gradient descent methods, which are iterative processes with several components. First, the gradient of the loss function is computed with respect to model parameters,

$$\nabla L(\mathbf{I}) = \left\{ \frac{\partial L(I_1)}{\partial I_1}, \frac{\partial L(I_2)}{\partial I_2}, \cdots, \frac{\partial L(I_{N^2})}{\partial I_{N^2}} \right\}.$$ \hspace{1cm} (15)

Here, the set of model parameters is equivalent to the set of pixel intensities. In order to begin optimization, it is necessary to select an initial set of pixel intensities to be evaluated against the loss function for the first iteration. The simplest starting point is a constant value image. However, a faster alternative is to initialize the model with some approximation of the true sky brightness. In most cases, the dirty image itself is already a decent approximation of the sky brightness distribution,
Figure 2. Visualizing the regularization process using gradient descent optimization to minimize the loss function for different initial pixel values. A combination of entropy ($\lambda = 0.5$), sparsity ($\lambda = 5 \times 10^{-6}$), and total squared variation ($\lambda = 10^{-4}$) regularizing terms were used on a mock dataset made from the ALMA logo with added noise. Each row shows the state of the model and gradient image at a different number of iterations during optimization. The leftmost columns show the sky brightness and gradient images of a model initialized with the dirty image, the middle columns show the same for a model initialized with a blank image, and the rightmost columns show the same for a model initialized with a custom image (in this case, an image of a dog). From left to right, this can be read as the best to worst guess for the true sky brightness distribution. In each gradient image, positive (negative) values are shown in red (blue), and $\nabla$ denotes the magnitude of the gradient vector (i.e. the gradient value of each pixel added in quadrature). All sky brightness and gradient images are plotted on the same color scale. A good set of initial pixel values like the dirty image quickly converges to the final image, while a poor guess is more computationally expensive but achieves the same result.
enabling the optimization process to converge in fewer iterations than if the initial state of the parameters had been uniform (or in any other configuration that is unlikely to represent the true sky brightness, as shown in the last two columns of Figure 2). If the loss function space is convex (i.e. has only a single global minimum), the model will converge to the same result regardless of the initial state of the parameters. The loss surface is convex for the regularizing terms presented here (e.g. see Akiyama et al. 2017a; Chael et al. 2018; Yamaguchi et al. 2020), so a poor choice of initial pixel intensities comes only at the cost of requiring more iterations to converge on a minimum loss value.

Figure 2 shows how different sets of initial pixel intensities impact the speed of convergence while regularizing a sky brightness projection of the ALMA logo with added noise (described in Section 2). We apply entropy, sparsity, and total squared variation regularizers to the loss function. The dirty image converges first, the blank image second, and the custom image last. We use a custom image of a dog, intentionally selecting a set of pixel intensities with no similarity to the true image. Though the custom set of initial pixel intensities takes significantly longer to converge, it ultimately does converge on the same result as the initial dirty and uniform images, showing that the final result is not sensitive to the initial state of the model. Figure 2 also shows how the model image is updated during optimization: after each iteration, the gradient of the image is added to the model parameters, creating a new model image. This process repeats until the loss function converges on a minimum, and the gradient is zero or approximately zero.

One important consideration with the gradient descent method is step size, also called the learning rate. Steps that are too large could overshoot the minimum, causing the algorithm to diverge. The smaller the step size, the more iterations will be required for the loss function to converge on a final value, meaning that steps that are too small can quickly become too computationally expensive to reach the minimum (Ch. 7.1, Deisenroth et al. 2020; Ch. 8.4, Murphy 2022). It is worthwhile to check that the loss function has actually converged on a value; an image that has not been fully optimized can be misleading as it is not actually the maximum likelihood solution. For example, in Figure 2 all of the model images at 50 and 150 iterations look quite similar. However, the magnitude of the gradient image is reduced by several orders of magnitude at 150 iterations. We can also see that the loss function has not yet been minimized at 50 iterations, especially when the model was initialized with a blank or custom image. Though it may be tempting to run fewer iterations in the interest of computational speed, it is essential to use enough iterations so that the loss function fully converges on a maximum likelihood solution.

3.3. Cross-Validation

Regularizers can be tuned by trial and error, testing new λ values until a seemingly reasonable value is found. This method has historically been used with success (e.g. Casassus et al. 2006), however, modern computational resources enable a more systematic way of determining λ prefactor values and optimally tuning regularizers. One way to determine whether the regularization (whether it be the strength of the λ prefactors or the functional form of the regularizer itself) is appropriately tuned is by using cross-validation (CV). CV aims to find optimal parameter values by determining how consistently the model performs given variations in the data set, working on the concepts of training data and testing data (Ch. 7.10, Hastie et al. 2009; Ch. 8, Deisenroth et al. 2020).

Training data is used to find the model which minimizes the specified loss function (including regularizers) and yields the best-fit image. Testing data is used for comparison against the model optimized with the training data. If some range of spatial frequencies is not covered by the training data, but is covered by the testing data, then comparing the trained model to the testing data effectively measures the predictive power of the model with respect to that range of spatial frequencies. In other words, testing data allows us to see how well the model predicts new data.

In principle, one would like to have a large enough pool of data such that partitioning it into a training set and a testing set would not compromise the utility of either subset. When dealing with costly observational data, however, using enough data to train the model typically leaves only a small amount for testing, resulting in a noisy estimate of the predictive performance of the model (Bishop 2006). CV partially circumvents this limitation by partitioning all the measured visibility data \( \mathbf{D} \) into subsets such that \( \mathbf{D}_i(\{u,v\}) \subset \mathbf{D} \), fitting the model on one or more subsets, and testing the model on the remaining subsets. One popular method is K-fold CV, which performs this process in multiple rounds and rotates which subsets are used for testing in each round (e.g. Akiyama et al. 2017a,b; Yamaguchi et al. 2020). In this case, data are partitioned into \( K \) subsets.

\[
\mathbf{D} = \begin{cases} 
\mathbf{D}_1(\{u,v\}) \\
\mathbf{D}_2(\{u,v\}) \\
\vdots \\
\mathbf{D}_K(\{u,v\}) 
\end{cases}
\] (16)
After partitioning, $K - 1$ subsets are combined to form the training data $D_{\text{train}}$ and the remaining subset $D_{\text{test}}$ is used for testing. Using only $D_{\text{train}}$, a set of model visibilities $V_{\text{train}}$ is obtained which covers a more comprehensive set of $u, v$ locations beyond those sampled by $D_{\text{train}}$. Then, $V_{\text{train}}$ and the withheld test set $D_{\text{test}}$ are compared within the same $u, v$ space sampled by $D_{\text{test}}$. Applying Equation 7, we obtain

$$p(V_{\text{train}} | D_{\text{test}}) \propto \exp \left[-\frac{1}{2} \chi^2(V_{\text{train}} | D_{\text{test}})\right],$$

which indicates that a smaller $\chi^2$ value corresponds to a better match between the trained visibilities and the testing data. In other words, the lower the $\chi^2$ value, the higher the probability of $V_{\text{train}}$ accurately modeling visibilities in $u, v$ space not included in the data set.

This process is repeated $K$ times such that each subset functions as the testing data exactly once. We obtain a final CV score by summing the $\chi^2$ values calculated for each of the $K$ CV rounds,

$$CV = \sum_{k}^{K} \chi^2(V_{\text{train}} | D_k),$$

where $D_k$ is the test dataset, and $V_{\text{train}}$ is the set of model visibilities that minimize the loss function for visibility data $D - D_k$. A low CV score indicates that the model (consisting of the choices of image parameterization, regularizers, and regularizer strengths), when trained on the training data, does a good job at predicting the withheld training data. If the regularizers and their strengths are poorly chosen, however, at least two failure modes arise. In the first, the model may simply fail to fit the training data adequately. This can happen if the model is over-regularized (not sufficiently flexible). When this happens, it is not surprising that the model also fails to predict the withheld test data accurately. The second failure mode arises when the model fits the training data accurately but fails to predict the withheld test data. This can happen if the model is under-regularized. In this situation, the model would be said to be over-fit.

Visibility datasets acquired by ALMA have many unique characteristics, such as their number, variable density of $u, v$ sampling, and varying signal-to-noise ratio, when compared to simpler datasets (e.g., data points in a polynomial regression). This presents many opportunities and challenges for how to partition data for K-fold CV. We explored CV using two methods of partitioning, which we dub “uniform” and “dartboard.” Uniform partitioning utilizes $K$ subsets that are composed of randomly-selected visibility grid cells. Grid cells are randomly drawn without replacement, so each grid cell is only used in a single subset. Dartboard partitioning uses polar grid lines to create a new layer of azimuthal and radial bins, each of which contains many visibility grid cells. Each of the $K$ subsets consists of randomly-drawn dartboard cells without replacement. Figure 3 shows an example of dartboard partitioning.

3.4. The MPoL Package
Million Points of Light (MPoL)

4. RESULTS

4.1. Performance of Maximum Entropy

The first column of Figure 4 shows the effect of different \( \lambda \) prefactors for maximum entropy regularization of HD 143006 with a positive and non-zero uniform set of reference pixels (\( I = 10^{-7} \)). Even at high \( \lambda \) values, maximum entropy regularization can retain high-resolution features in the image. However, because maximum entropy regularization generally promotes uniformity in the image, the image tends to a model image that appears “faded” at excessive values of \( \lambda \), making emission appear fainter across the entire source. The bottom-left panel of Figure 4 shows an example of such an image.

The primary indication of over-regularization with maximum entropy is an image that appears faint or slightly blurred compared to images made from different entropy \( \lambda \) values, suppressing bright peaks in the image. Another way to check for over-regularization is by examining the residual image. In the bottom-left panel of Figure 5, ringed structure is evident in the residual image created by maximum entropy regularization with \( \lambda = 5 \). In some cases, these effects may be mitigated by using a non-uniform set of reference pixels, such as a circular Gaussian (e.g. Event Horizon Telescope Collaboration et al. 2019) or a uniform ring. This should be done with caution, as maximum entropy regularization favors similarity with the reference image, and the reference image may not capture enough characteristics of the true source. This could have unintended consequences such as regularizing out small-scale structures or unexpected asymmetries that are present in the data but not in the set of reference pixels.

4.2. Performance of Sparsity

Sparsity regularization promotes mostly blank images, with only the most impactful pixels having nonzero values. Only a \( \lambda \) prefactor determines how strongly sparsity should be imposed during optimization; unlike maximum entropy regularization, no reference image is needed. Column 2 of Figure 4 shows images of HD 143006 with 5 different sparsity \( \lambda \) values. A small \( \lambda \) can effectively suppress noisy background pixels in the image without changing much, if anything, about the source emission.

Sparsity regularization alone does not introduce any sort of “smoothing” effects that may be desirable for a resolved source; the image will ultimately be a sparse collection of nonzero pixels, which can make it difficult or impossible to identify small-scale structures within the image. In the case of HD 143006, sparsity regularization is most effectively used in combination with other regularizing terms.

7 https://mpol-dev.github.io/MPoL/
Figure 4. The effect of individual regularizers on images of HD 143006. From left to right, each column shows entropy, sparsity, total variation, and total squared variation regularizers at varying strengths. The top row uses the least regularization (i.e. a small λ prefactor on the regularizing term) and the bottom row shows extremely high regularization. A range of λ values were selected in order to show the full range of possible images, but the optimal λ values can be found using CV methods. All images are displayed on a color scale normalized to the same maximum value.
Figure 5. Residuals for the images presented in Figure 4. Regularizer strengths are lowest in the top row, and highest in the bottom row. Images generated with strong regularizing terms show structure in the residual images, indicating that the model is underfitting the data.
Figure 6. Dirty image (left), tclean (center), and RML (right) images of HD 143006. The loss function for the RML image included maximum entropy ($\lambda = 0.5$), sparsity ($\lambda = 10^{-3}$), and TSV ($\lambda = 2 \times 10^{-3}$) terms. In this case, maximum entropy contributes high resolution features to the image, sparsity removes background noise, and TSV helps better define ring structure in the disk.

Over-regularization with sparsity can have a significant negative impact on image fidelity, yielding an image that is not representative of the entire source. Because the sparsity regularizer encourages mostly blank images, one potential drawback is the risk that astrophysically real but faint emission may not appear in the synthesized image. This has the effect of neglecting more diffuse emission in the synthesized image, as diffuse emission does not have bright peaks for the sparsity regularizer to identify. In the case of HD 143006, sparsity regularization with a high $\lambda$ value removed some of the more diffuse emission, and in extreme cases removed some rings entirely. In Figure 4, the bottom panel of column 2 of shows the result of imaging HD 143006 with sparsity over-regularization. Here, the outer ring has been regularized away, but the bright azimuthal asymmetry that normally coincides with the outer ring is still present, completely misrepresenting the morphology of the source.

Over-regularization is very evident in the residual image (column 2, Figure 5). Because only the most prominent features remain in the model image, any diffuse or generally lower-intensity emission will instead be apparent in the residuals. In the model image of a resolved source, things to look out for that may indicate sparsity over-regularization include features that appear ‘incomplete’ such as having partial rings, an unexpected bright standalone feature, or unexpectedly sharp changes in intensity.

The sparsity regularizer does not use any information on the contiguity of blank regions, therefore including a sparsity term will not necessarily favor adjacent bright pixels that would often be expected in a resolved source. However, even if the source is unlikely to be sparse in the image domain (e.g. extended sources like galaxies), sparse regularization has previously been shown to successfully reconstruct these images by transforming the data such that it is sparse in some other domain like wavelet coefficients (Li et al. 2011; Carrillo et al. 2012).

4.3. Performance of Total Variation

TV regularization promotes sharp edges between areas of different intensities, with smoothness in areas of similar intensity. Column 3 of Figure 4 shows the effect of different $\lambda$ prefactors for TV regularization of HD 143006. Because TV promotes similarity between adjacent pixels unless there is a large change in intensity (i.e. sparsity in the spatial gradient of the image), TV regularization can result in a model image composed of many nearly uniform cells, each containing several pixels. This can create an optical illusion where the image appears to have larger pixels than the true pixel size (e.g. see the image in the third column and third row of Figure 4, where $\lambda = 5 \times 10^{-5}$).

This effect changes as the $\lambda$ value increases, with model images being reminiscent of a watercolor painting or a photo that has been posterized. Because TV regularization does not favor gradual changes in intensity, instead preferring sharp changes, a smooth change in source intensity is likely to become a set of sharply defined layers in the image. In the case of extreme over-regularization, this can remove most detail from the image, resulting in an image that appears blotchy or smeared. However, in a source like HD 143006 which exhibits ringed emission, the smearing is mostly azimuthal rather than radial, retaining the large scale ring struc-
ture of the source in the image while losing or minimizing finer details like gaps or azimuthal asymmetries.

For these reasons, TV regularization may be a poor choice if the source is likely to have small scale features or gradual changes in intensity, as many astronomical sources do. Tuning the $\lambda$ prefactor for TV regularization is finicky compared to other regularizers, as increasing $\lambda$ can quickly take the image from essentially unregularized to the large pixel illusion to completely smeared emission. Though over-regularization can be evident due to the presence of structure in residual images (see the bottom panel of column 3, Figure 5), it may not be evident from inspection of residuals alone until well beyond the $\lambda$ value at which morphological details are regularized out of the image.

4.4. **Performance of Total Squared Variation**

Like TV regularization, TSV regularization promotes sharp edges between areas of different intensities. However, the TSV regularizer is less rigid with this condition, allowing for larger differences between adjacent pixels — while TV regularization applies sparsity (or the L1 norm) to the gradient of the image, TSV regularization applies the L2 norm to the gradient of the image. This makes TSV a strong performer for sources with clearly defined but not perfectly sharp features, such as ringed emission. The rightmost column of Figure 4 shows images of HD 143006 with 5 different TSV $\lambda$ values. Well-tuned TSV regularization performs comparably to maximum entropy regularization, retaining high-resolution features in the model image.

The primary sign of over-regularization with TSV is a blurred image. If the TSV-regularized image appears to have no sharp features at all, as if it had been put through a low pass filter, it is likely over-regularized. This is also evident in the residual images (rightmost column, Figure 5), where sharp structures will appear if they have been regularized out of the sky brightness image.

5. **DISCUSSION**

5.1. **Best Practices for Cross-Validation**

Cross-validation (CV) is a useful tool for determining the settings of the regularization parameters that yield an image model with the best predictive power for new data. The use of CV methods on interferometric data is an active area of research, and there are many aspects which have not yet been fully explored. First, there are both exhaustive and non-exhaustive CV methods; exhaustive methods use all possible ways to partition data, while non-exhaustive methods use only a subset of possible partitions. An example of an exhaustive method is leave-one-out CV (LOOCV), which takes all but one data point as the training set and uses the remaining data to test. These methods can be extremely computationally expensive, especially in the case of interferometric data containing millions of visibility measurements. In addition, LOOCV performs poorly compared to other methods of CV (Breiman & Spector 1992).

Non-exhaustive CV methods like K-fold CV greatly reduce the total computational burden, and are thus a more practical CV method for interferometric data. K-fold CV requires a choice of $K$ that balances bias and variance, with a high $K$ yielding a low bias, high variance estimation and a low $K$ yielding a high bias, low variance estimation (Hastie et al. 2009). Studies in statistics and informatics have consistently found $K = 10$ to provide the best bias-variance trade-off (e.g. Breiman & Spector 1992; Kohavi 1995; Molinaro et al. 2005). While no studies have specifically examined the optimal $K$ for K-fold CV of interferometric data, the standard $K = 10$ has been used with success for such applications (Akiyama et al. 2017b,a; Yamaguchi et al. 2020).

Aside from the choice of $K$, one must also decide how to partition the data set. Uniform partitioning randomly selects visibility grid cells without replacement for each subset of testing data. With such a large number of visibility grid cells relative to the $K$ number of subsets, it is likely that each subset will have similar $u,v$ coverage. Therefore, CV with uniform partitioning effectively tests how well a trained model predicts new data with similar $u,v$ coverage. Uniform partitioning has been used to tune regularizer hyperparameters in previous studies of RML for interferometry (e.g. Akiyama et al. 2017b,a; Yamaguchi et al. 2020). We also explore dartboard partitioning, which generates testing data from radial and azimuthal bins of gridded visibility cells (see Figure 3). CV with dartboard partitioning tests how well the model extrapolates to $u,v$ space notably different than the training data. Dartboard partitioning aims to simulate the irregular $u,v$ sampling common to most ALMA interferometric observations (e.g., across execution blocks or array configurations), approximating how the model might fit with data obtained from a variety of array configurations. As the number of dartboard bins increases, the dartboard partitioning scheme begins to test the predictive power of the model in comparable u-v space (as with uniform partitioning). The trade off between uniform and dartboard partitioning is still an active area of research, along with the potential development of other CV partitioning schemes. However, we find that in general, both uniform and dartboard partitioning are viable options for hyperparameter tuning.
Regardless of the choice of $K$ and data partitioning scheme, it is vital to ensure that the model has fully converged for each training set. We find that the number of iterations needed to optimize the model during K-fold CV is often greater than the number of iterations needed to optimize the model with the full dataset, as training on fewer data points can require more iterations before the loss function is minimized. This is especially true for the K-fold that does not contain the visibilities at the lowest spatial frequencies (i.e. $u, v$ close to zero), as the omission of this data can cause a slow initial decline of the total loss. We find that for both uniform and dartboard partitioning, the training set without the lowest spatial frequency visibilities exhibits delayed convergence.

Because the final CV score is the sum of the $\chi^2$ for all K-folds, a K-fold that has not fully converged can result in a spuriously high CV score. We recommend inspecting each K-fold for convergence after training, as well as inspecting the final $\chi^2$ value for each K-fold. An unusually high $\chi^2$ value for a single K-fold compared to the other $K-1$ K-folds can indicate a lack of convergence. This usually coincides with the K-fold for which the training set lacks data at the lowest spatial frequencies (e.g. K-fold 5 in Figure 3). If each K-fold does not reach convergence, the final CV score is invalid and the entire CV process must be repeated with enough iterations to ensure full convergence.

While CV scores can be used to find optimal regularization parameters, it is important to take care when comparing CV scores. First, as mentioned above, any CV score obtained from training without allowing the loss function to reach a minimum (i.e. one or more K-folds do not reach convergence) cannot be used. CV scores may only be compared across the same dataset, model specification, and CV setup. The CV setup includes the value of $K$ and the choice and implementation of partitioning scheme. For example, a CV workflow could be set up with $K=5$ and dartboard partitioning with 12 radial bins and 16 azimuthal bins, as shown in Figure 3.

It is possible to compare CV scores when using different regularizers, as long as the regularizer has a tuneable prefactor (e.g. $\lambda$ values) that can be set to zero. For example, the CV score for a model with only an entropy regularizer can be compared to that of a model with only a TV regularizer, because this is effectively comparing different ways to tune the $\lambda$ prefactors, including $\lambda_{\text{entropy}}=0$ and $\lambda_{\text{TV}}=0$. In addition to these edge cases which “turn off” certain regularizers, CV scores can be compared for any values of $\lambda_{\text{entropy}}$ and $\lambda_{\text{TV}}$, as long as the dataset and CV setup remain consistent.

Figure 7 shows an example of using CV score comparison to tune the hyperparameter for an image made with only TSV regularization. The CV score with respect to $\lambda_{\text{TSV}}$ is loosely U-shaped, and there are a range of $\lambda_{\text{TSV}}$ values that produce comparably low CV scores. We show three images over nearly a full order of magnitude in $\lambda_{\text{TSV}}$, but all corresponding to CV scores within 10% of the minimum CV score. Thus, rather than requiring strict minimization of the CV score, there can be a range of hyperparameters that result in similar images. This means that one can obtain optimal hyperparameter values using CV with a fairly coarse grid of hyperparameter values.

This may initially be counterintuitive, because the CV score measures the predictive power of the model and lower CV scores correspond to better predictive power. However, the exact value of the CV scores will depend on the CV setup, including the partitioning scheme and number of K-folds. In MPOI, visibility cells are partitioned randomly, and it is possible to store the random seed in order to reproduce the exact CV set up for the purpose of comparing different values. For example, in Figure 7 the CV score is minimized with $\lambda_{\text{TSV}}=3\times10^{-4}$ (solid black curve), but when we repeated the CV score analysis with a different random seed for the CV partitioning, the CV score was minimized with $\lambda_{\text{TSV}}=1\times10^{-4}$ (blue dashed curve). The curves have roughly the same shape, but do not track each other exactly and do not minimize the CV score at the same $\lambda_{\text{TSV}}$. We find that in practice, aiming to strictly minimize the CV score is unlikely to be meaningful due to there existing a range of $\lambda$ values for a given regularizer yielding low total CV scores, especially considering variations stemming from randomness in the data partitioning scheme.

We recommend using CV methods as a guide towards finding a range of acceptable hyperparameter values rather than using absolute minimum CV score to identify a single “best” hyperparameter value. Repeating CV with different random seeds and/or different partitioning schemes may be useful for further constraining this range and determining where the model most consistently exhibits strong predictive power. Note that while the CV scores themselves cannot be directly compared across different CV setups, the range of hyperparameter values that yield low CV scores can be compared.

5.2. Determining Image Resolution with RML

Imaging with CLEAN typically involves building up a model of CLEAN components and then convolving that model with the CLEAN beam. A CLEAN component may be as simple as a Dirac $\delta$-function, which is useful for fields with many point sources, but is unlikely to
Figure 7. An example of hyperparameter tuning with CV scores. Here, we use dartboard partitioning with 30 log-linearly spaced radial bins and 10 equally spaced azimuthal bins from 0 to $\pi$. Panels a, b, and c show images made with different values of $\lambda_{TSV}$ and their corresponding CV scores. Panel b shows the image corresponding to the minimum CV score, while panels a and c show images with CV scores within 10% of the minimum. Panel d shows how the CV score varies with $\lambda_{TSV}$. The gray lines show the $\chi^2$ contribution from each of the 10 K-folds used during CV. The solid black line shows the sum of these $\chi^2$ contributions, which yields the total CV score. The blue dashed line shows the total CV score for a different random seed used when partitioning data for CV, all else held constant. Because the CV score with respect to $\lambda_{TSV}$ generally makes a U shape, there are a range of $\lambda_{TSV}$ values that produce comparably low CV scores. All three images presented here fall into this range which spans nearly a full order of magnitude in $\lambda_{TSV}$. The K-fold denoted by the dash-dot line represents the K-fold in which the training data did not include the centermost visibilities (but the testing data did), which dominates the total CV score at low $\lambda_{TSV}$. 
Figure 8. The effect of a restoring beam on an RML image of HD 143006. As in Figure 6, the loss function for the RML image included maximum entropy ($\lambda = 0.5$), sparsity ($\lambda = 10^{-3}$), and TSV ($\lambda = 2 \times 10^{-3}$) terms. The top left panel shows the RML image with no restoring beam. All other panels show the same image after being convolved with a circular Gaussian restoring beam, with beam sizes shown in the bottom left corner of each panel. The beam sizes shown are based off the resolution of the CLEAN image made from the DSHARP data of HD 143006, which has a synthesized beam FWHM of 45 $\times$ 46 mas (Andrews et al. 2018; Pérez et al. 2018). We use this as a reference resolution, and show the RML image restored to 46 mas in the bottom center panel. We also show images convolved with beams of 1/4 (top center), 1/3 (top right), 1/2 (bottom left), and 3/2 (bottom right) the size of the 46 mas reference.
be a poor representation of a spatially resolved source. Beam convolution effectively spreads these components over the size of the beam, making the image more representative of the true source morphology at the cost of resolution. Because the size of the CLEAN beam is a known quantity, characterizing the resolution of a CLEANed image is relatively straightforward.

RML images, on the other hand, are not generated from a set of individual components and thus do not require beam convolution in order to obtain a smoother image product. The most obvious benefit to this is that a strict limit on resolution is not baked into the imaging workflow. However, the lack of restoring beam does make characterizing the resolution of an RML image more ambiguous than characterizing the resolution of a CLEAN image. Chael et al. (2016) find that restoring beams can still be useful for RML methods, as false high-frequency features can sometimes be injected into the image. While a restoring beam can suppress this issue, using too large of a beam could remove real features in the image.

The optimal restoring beam size can be computed given knowledge of the true source; the beam size that minimizes the NRMSE

\[
\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{N^2} (|I'_i| - |I_i|)^2}{\sum_{i=1}^{N^2} |I_i|^2}}, \tag{19}
\]

where \( I'_i \) is an RML image pixel and \( I_i \) is some corresponding reference image pixel for an image with \( N^2 \) total pixels, is commonly adopted as the primary metric for evaluating the quality of reconstructed interferometric images (e.g. Chael et al. 2016; Akiyama et al. 2017a,b; Kuramochi et al. 2018; Yamaguchi et al. 2020). Notably, Chael et al. (2016) show that when trying to recover a “true” reference model input image of a compact source, the NRMSE is minimized at a considerably smaller beam size with RML techniques compared to CLEAN. If the ground truth is known (e.g. when making an RML image from simulated data based off of a model image), then convolving the RML image with a beam size that minimizes the NRMSE can maintain the highest degree of superresolution in the image while removing any potential spurious high-frequency features.

Given that perfect knowledge of the true sky brightness does not accompany observational data, this approach will not always be possible. Even though the optimal restoring beam size cannot always be constrained, image fidelity is only mildly worsened by selecting a beam size smaller than the optimal value. As a result, the use of a restoring beam in an RML imaging workflow can only result in small fidelity gains in the best case, and can result in significant loss of resolution in the worst case.

RML methods are known to be capable of generating images superresolved to 1/4 of the nominal resolution of the interferometer \( R_{\text{min}} = \lambda/b_{\text{max}} \), where \( b_{\text{max}} \) is the length of the longest baseline in the array (e.g. Narayan & Nityananda 1986; Honma et al. 2014). This is typically treated as an upper resolution limit for RML methods, as the derivation of this factor assumes that the data have high signal-to-noise and thoroughly sample the visibility function (Holdaway 1990). In practice, superresolution factors ranging from roughly 1/3 to 1/2 of the nominal resolution are the most common outcome from RML imaging methods (e.g. Chael et al. 2016; Akiyama et al. 2017a,b; Cieza et al. 2017; Casassus et al. 2018; Kuramochi et al. 2018; Casassus et al. 2019, 2021).

Figure 8 shows RML images of HD 143006 both without any restoring beam, and with restoring beams equal to 1/4, 1/3, 1/2, 1, and 3/2 times the synthesized beam size of the DSHARP continuum tclean image of HD 143006 (11.5, 15.3, 23.0, 46.0, and 69.0 mas, respectively). We used astropy.convolution in order to convolve the model with circular Gaussians directly in the image plane. There are no significant qualitative differences in the base RML image (top left panel) and the RML images convolved with beams equal to 1/4 (top center), 1/3 (top right), and 1/2 (bottom left) the size of the synthesized beam of the tclean image. In this case, we do not observe spurious high-frequency features in the base RML image, so convolution with modest restoring beams has little impact on the final image. We emphasize that while RML methods do not require a restoring beam, convolving an RML image with a modest restoring beam (i.e. 1/3-1/2 the nominal resolution of the observations) yields a more conservative final image while still benefitting from some degree of superresolution.

6. CONCLUSION

We have developed a GPU-accelerated RML imaging package called MPoL designed for synthesis of complex visibilities from ALMA measurement sets. We provide a mathematical description of RML imaging methods and several regularizers, and developed an RML imaging framework with MPoL. We explored how maximum entropy, sparsity, TV, and TSV regularizers can be incorporated into the imaging process, and how each of these regularizers impacts image synthesis of ALMA data of protoplanetary disk HD 143006. We found that a combination of entropy, sparsity, and TSV regularization works well for this particular imaging case. For the thin
ringed structure present in HD 143006, we found that TV regularization does not adequately retain fine details in the image.

In addition, we explored CV methods, a robust procedure for hyperparameter tuning, for image validation in order to maximize image fidelity and resolution. We used K-fold CV with novel dartboard partitioning, finding that it is a viable CV partitioning method that can be used alongside or in place of uniform partitioning. We found that a range of hyperparameter values can result in comparably low CV scores, suggesting that it is not necessary to finely tune hyperparameters according to the CV score (which can impose a computational burden). Rather, using CV across a coarse grid of hyperparameter values is a more efficient way to guide the tuning process.

Overall, RML techniques provide flexible imaging processes that are well-suited for applications to ALMA measurement sets, particularly observations of protoplanetary disks. The use of RML techniques can improve image fidelity on small scales, with the potential to aid the detection and characterization of kinematic disturbances within protoplanetary disks (such as the discovery of the circumplanetary disk candidate in the disk around AS 209 presented in Bae et al. (2022)). Exploring regularization on a wider range of source morphologies observed by ALMA, including image cubes with many channels, will broaden our understanding of how image synthesis with RML techniques can benefit different science cases.

We acknowledge funding from an ALMA Development Cycle 8 grant number AST-1519126. This paper makes use of the following ALMA data: ADS/JAO.ALMA#2016.1.00484.L. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), MOST and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc. Yina Jian is a summer student at the National Radio Astronomy Observatory. Computations for this research were performed on the Pennsylvania State University’s Institute for Computational and Data Sciences’ Roar supercomputer. The Center for Exoplanets and Habitable Worlds is supported by the Pennsylvania State University, the Eberly College of Science, and the Pennsylvania Space Grant Consortium. This research has made use of NASA’s Astrophysics Data System Bibliographic Services.

**Software:** Astropy (Astropy Collaboration et al. 2013, 2018), PyTorch (Paszke et al. 2019), MPoL (Czekala et al. 2021), CASA (McMullin et al. 2007)

APPENDIX

A. RECOMMENDATIONS FOR RML WITH ALMA

The MPoL software used in this study is open source and designed with ALMA measurement sets in mind. We recommend that anyone wishing to use MPoL for RML imaging of ALMA data follow these general steps.

1. **Obtain arrays of complex visibility data.** MPoL is designed to work directly with arrays of complex visibilities in order to avoid potential issues with installation and version control of CASA and related tools. Visibilities can be obtained from CASA measurement sets using CASA tools. There are also open source software packages like visread\(^8\) that can aid in this process.

2. **Select pixel size and number of pixels in the image.** These pixels will serve as the model parameterization, and allow loose visibilities to be gridded. It is important to be mindful of the choice of pixel size, as pixels that are too large will place an intrinsic resolution limit on the final image, and using too many small pixels will introduce an unnecessary computational burden during the imaging process. At this point, it is also a good idea to examine the dirty image in MPoL to make sure that everything has been loaded and initialized as expected.

\(^8\) https://mpol-dev.github.io/visread/
3. **Set the initial state of the model and determine which regularizers to include in the loss function.** We recommend initializing the model with the dirty image. Dirty image initialization leads to faster convergence of the loss function. The loss function may include any number of regularizing terms.

4. **Define a range of hyperparameter values to be tested with CV.** We recommend beginning with a coarse grid of values. The hyperparameter values that yield the highest fidelity images can change with the dataset, so starting with a wide range of values (i.e. spanning several orders of magnitude) helps quickly hone in on a narrower range of potential values. For example, while one dataset might minimize the CV scores with an entropy prior with $\lambda = 0.1$, another dataset might minimize the CV score with $\lambda = 10$, so a coarse round of CV might include $\lambda = [0.01, 0.1, 1, 10]$.

5. **Set up CV process by defining the number of K-folds and data partitioning scheme.** We recommend using $K = 10$, which has been shown to balance bias and variance in the estimation and has already been used with success in interferometric imaging.

6. **Perform CV (perhaps in a coarsely-defined round and a fine-tuning round) to obtain optimal imaging hyperparameters.** The set of hyperparameters that minimizes the CV score correspond to the model with the best predictive performance. Ensure that each K-fold has reached convergence during the CV process; if each K-fold did not fully converge, the CV score is invalid and CV must be restarted with enough iterations to allow full convergence.

7. **Image using the full dataset with the hyperparameters that minimized the CV score.** Ensure that the model has fully converged. If it has, the result is a fully regularized image.

8. **Optionally, convolve the image with a restoring beam.** While beam convolution is not required in an RML imaging workflow, some users may find it helpful for characterizing the resolution of the image and filtering out potential spurious high-resolution features in the image. We recommend restoring the image to no more than 1/3 to 1/2 of the nominal resolution in order to retain the superresolution benefits of RML.

While this is a rough outline of a functional RML workflow, in practice the RML imaging process need not be so linear. For some imaging cases, one may find it beneficial to try individual regularizers before combining them, or to produce some preliminary images before running the full CV process. When selecting regularizers, we recommend considering the qualitative features the source is likely to have. Maximum entropy and sparsity regularization tend to produce high-resolution features, and sparsity has the added benefit of effectively removing background noise. TV and TSV regularizers tend to emphasize sharp edges in the image, with TV enforcing sharp edges more rigidly. These general characteristics can help inform which regularizers to include in the loss function.

**REFERENCES**

Ables, J. G. 1974, A&AS, 15, 383

Akiyama, K., Ikeda, S., Pleau, M., et al. 2017a, AJ, 153, 159, doi: 10.3847/1538-3881/aa6302

Akiyama, K., Kuramochi, K., Ikeda, S., et al. 2017b, ApJ, 838, 1, doi: 10.3847/1538-4357/aa6305

ALMA Partnership, Brogan, C. L., Pérez, L. M., et al. 2015, ApJL, 808, L3, doi: 10.1088/2041-8205/808/1/L3

Andrews, S. M., Wilner, D. J., Zhu, Z., et al. 2016, ApJL, 820, L40, doi: 10.3847/2041-8205/820/2/L40

Andrews, S. M., Huang, J., Pérez, L. M., et al. 2018, ApJL, 869, L41, doi: 10.3847/2041-8213/aaf741

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123, doi: 10.3847/1538-3881/aabc4f

Bae, J., Teague, R., Andrews, S. M., et al. 2022, ApJL, 934, L20, doi: 10.3847/2041-8213/ac7fa3

Bishop, C. M. 2006, Pattern Recognition and Machine Learning, ed. M. Jordan, J. Kleinberg, & Schölkopf, Bernhard (Springer-Verlag New York).

https://www.springer.com/gb/book/9780387310732

Breiman, L., & Spector, P. 1992, International Statistical Review / Revue Internationale de Statistique, 60, 291.

http://www.jstor.org/stable/1403680

Briggs, D. S. 1995, PhD thesis, New Mexico Institute of Mining and Technology
Pérez, L. M., Benisty, M., Andrews, S. M., et al. 2018, ApJL, 869, L50, doi: 10.3847/2041-8213/aaf745

Pérez, S., Casassus, S., Baruteau, C., et al. 2019, AJ, 158, 15, doi: 10.3847/1538-3881/ab1f88

Pinte, C., Price, D. J., Ménard, F., et al. 2018, ApJL, 860, L13, doi: 10.3847/2041-8213/aac6dc

Ponsonby, J. E. B. 1973, MNRAS, 163, 369, doi: 10.1093/mnras/163.4.369

Rudin, L. I., Osher, S., & Fatemi, E. 1992, Physica D Nonlinear Phenomena, 60, 259, doi: 10.1016/0167-2789(92)90242-F

Shannon, C. E. 1948, The Bell System Technical Journal, 27, 379, doi: 10.1002/j.1538-7305.1948.tb01338.x

Sivia, D. S., & Skilling, J. 2006, Data Analysis - A Bayesian Tutorial, 2nd edn., Oxford Science Publications (Oxford University Press)

Thompson, A. R., Moran, J. M., & Swenson, George W., J. 2017, Interferometry and Synthesis in Radio Astronomy, 3rd Edition, doi: 10.1007/978-3-319-44431-4

Tibshirani, R. 1996, Journal of the Royal Statistical Society. Series B (Methodological), 58, 267. http://www.jstor.org/stable/2346178

Tripathi, A., Andrews, S. M., Birnstiel, T., et al. 2018, ApJ, 861, 64, doi: 10.3847/1538-4357/aac5d6

Wiaux, Y., Puy, G., & Vanderghynst, P. 2010, MNRAS, 402, 2626, doi: 10.1111/j.1365-2966.2009.16079.x

Yamaguchi, M., Akiyama, K., Tsukagoshi, T., et al. 2020, ApJ, 895, 84, doi: 10.3847/1538-4357/ab899f

Zhang, K., Bergin, E. A., Blake, G. A., et al. 2016, ApJL, 818, L16, doi: 10.3847/2041-8205/818/1/L16