

Challenging interferometric imaging: machine learning based source localization from uv-plane observations

O. Taran¹, O. Bait², M. Dessauges-Zavadsky², T. Holotyak¹, D. Schaerer², and S. Voloshynovskiy^{1,*}

¹ Department of Computer Science, University of Geneva, Switzerland
e-mail: olga.taran@unige.ch, svolos@unige.ch

² Observatoire de Genève, Université de Genève, 51 Chemin Pegasi, 1290 Versoix, Switzerland
e-mail: daniel.schaerer@unige.ch

May 2, 2023

ABSTRACT

Context. Increasing interest in radio astronomy and upcoming projects will produce petabytes of data per day questioning the applicability of traditional radio astronomy data analysis approaches under new large scale conditions and require new, intelligent, fast and efficient methods with potentially less domain-expert's input.

Aims. In our work we examine, for the first time to the best of our knowledge, the possibility of fast and efficient source localization directly from the uv-observations omitting the recovering of the dirty or clean images.

Methods. We propose a deep neural network based framework that takes as an input a low-dimensional vector of sampled uv-data and outputs source positions on the sky. We investigate a representation of the complex-valued input uv-data via the real and imaginary and magnitude and phase components. We provide a comparison of the efficiency of the proposed framework with the traditional source localization pipeline based on the state-of-the-art Python Blob Detection and Source Finder (PyBDSF) method. The investigation is performed on a data set of 9164 sky models simulated using the Common Astronomy Software Applications (CASA) tool for the Atacama Large Millimeter Array (ALMA) Cycle 5.3 antenna configuration.

Results. We investigate two scenarios: (i) noiseless as an ideal case and (ii) sky simulations including noise representative of typical extra-galactic millimeter observations. In the noiseless case, the proposed localization framework demonstrates the same high performance as the state-of-the-art PyBDSF method. For noisy data, however, our new method performs significantly better, achieving a ~ 3 times higher completeness for sources with uniform signal-to-noise (SNR) ratios between 1 and 10, and a large increase of completeness in the low SNR regime. Furthermore, the execution time of the proposed framework is significantly reduced (by factors ~ 30) compared to traditional methods which include image reconstructions from the uv-plane and subsequent source detection.

Conclusions. The proposed framework for fast and efficient source localization directly from uv-plane observations shows very encouraging results, which could open new horizons for interferometric imaging with existing and future facilities.

Key words. Techniques: interferometric – Methods: data analysis – Sub-millimeter: general – Radio continuum: general

1. Introduction

Radio astronomy is at the forefront of a revolution in terms of the sensitivity that can be achieved at centimeter and meter wavelengths. The various radio-astronomical interferometers like the LOw-Frequency ARray (LOFAR; van Haarlem et al. 2013), MeerKAT radio telescope (Jonas & MeerKAT Team 2016), Australian square kilometer array pathfinder (ASKAP; Hotan et al. 2021) and Murchison Widefield Array (MWA; Tingay et al. 2013) are already producing promising results. At the same time, radio astronomy is an extremely data-intensive science. It is expected that the upcoming projects will produce data volumes on an exabyte scale (Scaife 2020). Thus, it will be very challenging for astronomers to undertake standard radio data analysis tasks like calibration, imaging and source localization using traditional techniques. Hence, it is vital to design fast and efficient techniques to replace the traditional data analysis approaches in radio astronomy.

At millimeter (mm) wavelengths the Atacama Large Millimeter Array (ALMA) owing to its excellent sensitivity has led to several large imaging and spectroscopic programs. Of par-

ticular interest in the context of the current work are the various large programs targeting extra-galactic deep fields. This includes for example the Reionization Era Bright Emission Line Survey (REBELS; Bouwens et al. 2022), the ALMA SPECTroscopic Survey in the Hubble Ultra-Deep Field (ASPECS; Walter et al. 2016), the ALMA Large Program to INvestigate [CII] at Early times (ALPINE Le Fèvre et al. 2020; Béthermin et al. 2020; Faisst et al. 2020) and the GOODS-ALMA survey at 1.1 mm (Franco et al. 2018). Another rich ALMA data set on which our current study is focused is the Automated Mining of the ALMA Archive in the COSMOS Field (A³COSMOS) data set¹ (Liu et al. 2019). These surveys are studying the gas and dust properties of galaxies at high redshifts.

A key technical component while analyzing these data is to accurately identify sources (their positions, fluxes and sizes) in noise-limited images produced by radio interferometers. This is important for example to calculate the number density and luminosity function of astronomical sources. It has important implications for constraining various physical models in astrophysics and cosmology. Traditionally radio source properties were mea-

¹ <https://sites.google.com/view/a3cosmos/home?authuser=0>

* S. Voloshynovskiy is a corresponding author

sured using a two-dimensional Gaussian fit to the light profile (Condon 1997) and usually required some level of manual intervention. Such manual interventions become increasingly difficult when dealing with large-area ALMA surveys mentioned previously. Also with the all-sky radio surveys like the Faint Images of the Radio Sky at Twenty-cm (FIRST) survey (Becker et al. 1995) and NRAO Very Large Array (VLA) Sky Survey (NVSS; Condon et al. 1998) which contain millions of sources, manual source detection is difficult. Future large-area radio surveys (Norris et al. 2013) are expected to detect an order of magnitude more number of sources (e.g., the Evolutionary Map of the Universe (EMU) survey (Norris et al. 2011)). Radio surveys of extra-galactic deep fields using upgraded radio facilities, such as, the VLA-COSMOS 3 GHz survey (Smolčić et al. 2017) and the MeerKAT International GHz Tiered Extra-galactic Exploration (MIGHTEE) survey (Jarvis et al. 2016; Heywood et al. 2022), are already detecting several thousands of sources in a single field.

Thus, it is essential to build an automatic source detection algorithm. Such an algorithm should accurately detect sources above the noise level with high completeness and should simultaneously have a low number of false detection. There are several automatic source detection tools available in the literature for example, Search And Destroy (SAD)², Source-Extractor (SExtractor) (Bertin & Arnouts 1996), AEGEAN (Hancock et al. 2012, 2018) and Python Blob Detection and Source Finder (PyBDSF; Mohan & Rafferty 2015), PySE (Carbone et al. 2018), CAESAR (Riggi et al. 2019, 2016), PROFOUND (Robotham et al. 2018; Hale et al. 2019), SOFIA (Serra et al. 2015; Westmeier et al. 2021). Hopkins et al. (2015) provide a detailed comparison between different source detection algorithms available in the literature and their limitations. This works very well for bright sources. However, it is much more interesting to detect faint sources typically with signal-to-noise ratio (SNR) below 5.0 with high completeness and purity. This not only can probe the sources at even higher redshifts than currently intended in a given survey. It also has the potential to detect new kinds of astronomical sources lying close to the noise level which may have been missed by studies using traditional source detection techniques.

Recent advances in machine learning and in particular deep neural networks in the form of convolutional neural networks (CNNs) have led to a lot of success in radio astronomy. In particular, CNNs have been extensively used to classify radio galaxy morphologies, for example, (Aniyan & Thorat 2017; Lukic et al. 2019a; Ma et al. 2019; Tang et al. 2019; Bowles et al. 2021; Riggi et al. 2022). Riggi et al. (2022) use Mask R-CNN object detection framework on the ASKAP EMU survey data to perform both object detection and classification. Schmidt et al. (2022) use CNNs designed for super-resolution applications directly on the UV data to up-sample features in the case of sparse sampling, for instance, Very Long Baseline Interferometry. The images produced from these UV data show good recovery of source properties. CNNs have also shown great promise in point source detection as demonstrated in ConvoSource (Lukic et al. 2019b) and DeepSource (Vafaei Sadr et al. 2019). DeepSource is shown to be perfect in terms of purity and completeness down to SNR of 4 and outperforms the current state-of-the-art source detection algorithm PyBDSF in several metrics. Recently, it was shown that an encoder-decoder based neural network DECORAS (Rezaei et al. 2022) can perform source detection even on dirty images down to SNR of 5.5 and can also recover var-

ious source properties like fluxes and sizes quite accurately. Delli Veneri et al. (2022) further shows that deep learning-based source detection can also be performed on dirty spectral data cubes with good recovery of source properties like morphology, flux density and projection angle.

A common problem in most of the source finding algorithms in the literature is that they are performed on the image plane, and mostly on CLEAN images. CLEAN images are computationally expensive to produce and scale poorly with the data volumes. Thus even if these source finding algorithms are made efficient, they will still be limited by the time taken to produce the CLEAN images. It is well known that CLEAN leads to several imaging artifacts which can affect the purity of these source finding algorithms. And despite automating several steps in source finding, these approaches still require some amount of manual intervention to exclude imaging artifacts. In this work, we circumvent these problems by designing a novel direct uv-plane based source localization algorithm using recent advances in deep neural networks.

Motivated by the achievements of the deep neural networks (DNN) made in many domains, we propose a fast and efficient DNN-based framework for source localization. In general, this framework takes as input a low-dimension vector of sampled uv-observations and outputs the source position on sky in the form of a binary map. Our proposed framework is targeted towards source detection in ALMA continuum images at mm wavelengths, particularly to detect low SNR sources and to speed up the overall source detection process. We have trained and tested our proposed framework on simulated ALMA data. We investigate the impact of different factors on the performance of the proposed framework, namely, the input complex-valued data representation via real and imaginary, or magnitude and phase components, the impact of receiver noise and atmospheric noise due to the presence of water vapor, the impact of SNR of the sources and the impact of the number of sources in the field of view. We also provide a detailed analysis of the execution time of the proposed framework and compare it with those of the traditional source localization pipeline.

The paper is organized as follows. The ALMA data simulation procedure, the chosen parameters and the analysis are given in Sect. 2; the traditional pipeline of source localization is described in Sect. 3; the framework proposed in this work is explained in Sect. 4; Sect. 5 is dedicated to the analysis of the obtained results and Sect. 6 provides the comparisons with the literature. Finally, Sect. 7 concludes the paper.

2. Data

We train the proposed DNN-based framework on synthetic data with known source localization. All reported experiments have been performed exclusively on synthetic data. In the future, we plan to apply the proposed frameworks to the real observations taken from the (A³COSMOS) data set (Liu et al. 2019). Thus our simulated ALMA observations are designed to somewhat match the A³COSMOS observations.

For all our simulations we use the Common Astronomy Software Applications (CASA) data processing software v6.2 (McMullin et al. 2007; THE CASA TEAM et al. 2022). We produce simulations for the 12-m ALMA array with a fixed configuration from the ALMA cycle 5.3. All our simulation pointings are randomly distributed within a radius of 1 deg around the COSMOS field centered on: J2000 10h00m28.6s +02d12m21.0s. For simplicity, we fix the ALMA observing band to Band-6 centered at 230 GHz split in 240 channels with a channel width of 7.8

² <http://www.aips.nrao.edu/cgi-bin/ZXHLP2.PL?SAD>

Table 1: Summary of the data simulation parameters using CASA.

Parameter	Value	Notes
Antenna configuration	ALMA Cycle 5.3	
Number of antennas	50	
Field	COSMOS	J2000 10h00m28.6s +02d12m21.0s
Number of pointings/simulations	10166	Every pointing is randomly chosen within a radius of 1 deg around the field.
Central Frequency	Band 6 (230 GHz)	
Number of channels	240	
Channel Width	7.8 MHz	
Sampling time	10 secs	
Total Integration time	20 mins	
Hour Angle	Transit	
Sky model dimensions	512 × 512 pixel with 240 channels	
Pixel size	0.1''	
Source type	Gaussian	The size of the major/minor axis and position angle is varied randomly and is chosen to be below the synthesized beam.
Major (minor) axis	0.4'' to 0.8''	Typical resolution is 0.89'' × 0.82''
Position angle (PA)	0 to 360 deg	Chosen randomly
Number of sources	0 – 5	Randomly chosen between 1 and 5. We add a few source free simulations
Source positions	random	Sources are randomly distributed within the primary beam.
Flux range	0.05 mJy to 0.5 mJy	The flux of each source is randomly chosen from this range assuming a uniform distribution. This roughly keeps a flat SNR range for our data set.
Spectral index	0	We set this parameter to zero since the fractional bandwidth is quite small (0.8%).
Primary beam	22.86 ''	
Synthesized beam	0.89'' × 0.82''	
Noise (pwv)	1.796	This parameter adds the receiver and atmospheric noise due to water vapor to the visibilities.
Weighting	Natural	
Robust	0.5	
RMS noise (in images)	~50 μJy	

MHz. Our simulation pipeline follows the standard approach of first simulating a true sky model at a known phasecentre with known source positions and fluxes. The sky model consists only of sources with a Gaussian light profile with sizes close to or below the synthesized beam. In each pointing the source position, size, that is major and minor axis of the Gaussian, position angle, SNR and total number of sources that is in the range 1 to 5, are kept at random and drawn from an uniform distribution. This range in the number of sources per pointing and their sizes are chosen from true data from the A³COSMOS data in

Band-6 for the ALMA compact configuration (similar to ours). Instead of drawing our simulation parameters exactly from the observed distribution in A³COSMOS data, we use a uniform distribution but the range of these parameters is motivated by the A³COSMOS data set (see discussion in the following paragraph). The sky model contains a 512 × 512 pixel image with a pixel size of 0.1''. This sky model is used to generate the noise free uv-data using the SIMALMA task (McMullin et al. 2007). We sample each visibility every 10 secs for a total of 20 mins of integration on-source and choose the hour angle range such that

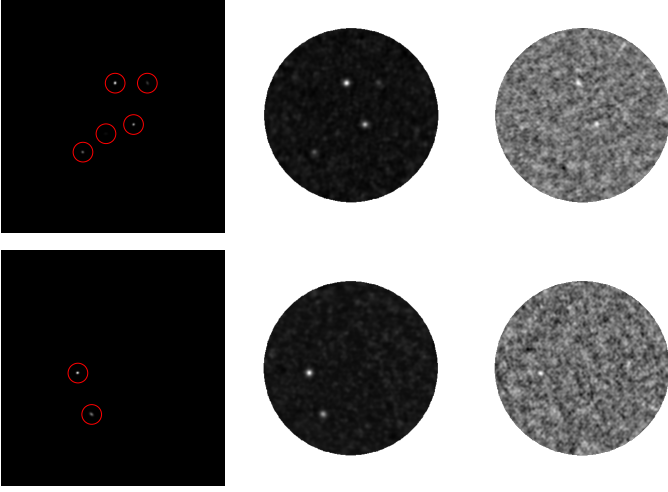


Fig. 1: Examples of the simulated data: top left - true sky model; top middle - noiseless; top right - noisy; bottom left - true sky model; bottom middle - noiseless; bottom right - noisy. For the noiseless and noisy cases, the CLEAN representations are visualized. The sources are highlighted by red circles.



Fig. 2: Example of closely located sources of different intensity. For better visibility the cropped and zoomed parts of the true sky models are shown.

the observing field is at transit. Appendix A shows a typical UV sampling for one of the simulations and the corresponding dirty beam. We then add the ALMA receiver and atmospheric noise due to water vapor to the uv-data using the pwv parameter value of 1.796 (McMullin et al. 2007). This is a typical value in Band-6 as mentioned in the ALMA technical handbook³. We then average the visibilities, both noisy and noise-free, in time and frequency by gridding the visibilities on a uniform grid using the msuvbin task (McMullin et al. 2007). Finally, we produce the dirty and CLEAN images from both the noisy and noise-free visibilities using the standard tclean task (McMullin et al. 2007). For our simulation setup, we typically reach a rms noise of 50 μ Jy in our dirty and CLEAN images. The size of the primary and synthesized beam is 22.86'' and 0.82'' respectively. Table 1 summarizes the various simulation parameters.

Although geared towards, our simulations are intentionally chosen to not match exactly the real observations from A³COSMOS. In particular, our sources are assigned random on-sky positions, fluxes and sizes as opposed to choosing directly from the distribution of sources from true observations or those motivated by cosmological simulations. The phasecenter and number of sources in each simulation pointing are also randomly distributed around the COSMOS field. This ensures that during training we do not end up in the region of latent space

Table 2: The number of sky models with different numbers of sources in the test subsets.

	Number of sources					Total
	1	2	3	4	5	
Test subset 1	161	177	196	177	205	916
Test subset 2	197	191	163	182	183	916
Test subset 3	165	175	181	180	215	916
Average	174	181	180	180	201	916

of our deep learning model which is only trained to work for a particular data set. And thus it does not learn additional patterns which might exist in real observations or cosmological simulations such as, source clustering, the luminosity function, size distribution etc. Instead, our model is trained to be flexible enough to be applied to any other new ALMA data. On the other hand, for our current work, we fix the ALMA observing band, channel width and total integration time. This is to reduce the number of free parameters in ALMA simulations. In the future, we will test the effect of changing these two parameters especially when we apply our model to real data.

We produce a total of 9164 independent simulations/pointings with the sky model containing between 1 to 5 sources. In total, we have 27632 sources across all the pointings. Each simulation produces ~ 3.5 GB of data which includes the uv-data, and dirty and CLEAN images that is ~ 35 TB for the entire data set⁴. The simulations are produced on the LESTA computing cluster. The more details are given in the Appendix B.

Fig. 1 and 2 show examples of the simulated data. The red circles highlight the positions of the sources in the true sky models. It is important to mention that the generated data set includes the same challenges encountered in real data, for example, the presence of blended sources as in Fig. 2, and sources hardly distinguishable from the background noise as in Fig. 1 (top middle and bottom right).

For the training of our framework, we perform the train-validation-test splitting 3 times with different seeds. Table 2 shows the number of sky models with different numbers of sources in the test subsets.

The notation we use is defined as follows: $\mathbf{x} \in \mathbb{R}^{N \times M}$ denotes the true sky model, where N and M are the image size, set to 512 in our simulations. $\hat{\mathbf{x}}_{dirty} \in \mathbb{R}^{N \times M}$ and $\hat{\mathbf{x}}_{clean} \in \mathbb{R}^{N \times M}$ denote dirty and CLEAN image respectively. $\mathbf{y} \in \mathbb{C}^K$ stands for the sampled uv-data with $K = 1400$ ⁵. $\mathbf{m} \in \{0, 1\}^{N \times M}$ is a binary source map, in which source pixels are set to 1, and background pixels are set to 0. We calculated SNR as:

$$\text{SNR} = \frac{\text{total flux}}{\sigma_{\text{noise}}}, \quad (1)$$

where total flux is the true source intensity and σ_{noise} is the noise RMS. As it is explained in Table 1, $\sigma_{\text{noise}} \sim 50 \mu\text{Jy}$.

³ <https://almascience.nrao.edu/proposing/technical-handbook>

⁴ The produced data set will be available by request upon the paper acceptance.

⁵ The technical details are given in Appendix C.

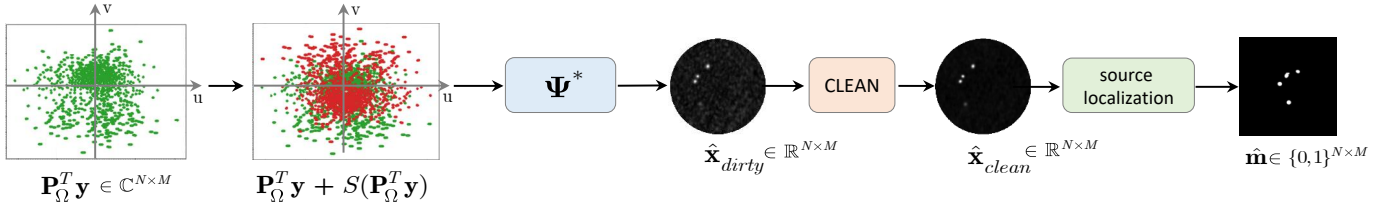


Fig. 3: Schematic representation of the traditional pipeline.

3. Traditional pipeline of source localization

The traditional pipeline of source localization is based on the reconstruction of the dirty and CLEAN images. The schematic representation of the traditional pipeline is given in Fig. 3.

3.1. Dirty image recovering

The recovering of the dirty image might be formulated as a recovering of a high dimension sky model⁶ $\mathbf{x} \in \mathbb{R}^{N \times M}$ from the corresponding low-dimensional sampled uv-visibility $\mathbf{y} \in \mathbb{C}^K$ ($K \ll N \cdot M$) corrupted by noise \mathbf{e} :

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{e}, \quad (2)$$

where $\mathbf{W} = \mathbf{P}_\Omega \Psi \in \mathbb{C}^{K \times N \cdot M}$ is a measurement sub-sampling matrices with an orthonormal basis $\Psi \in \mathbb{C}^{N \cdot M \times N \cdot M}$ and sampling operator $\mathbf{P}_\Omega : \mathbb{C}^{N \cdot M} \rightarrow \mathbb{C}^K$ and $|\Omega| = K$.

It should be pointed out that due to the physical imaging constraints in the radio-astronomy, Ψ corresponds to the Fourier operator and \mathbf{P}_Ω is determined by the antennas configuration, measured frequencies, Earth's movement, sampling and integration time.

The recovering of the $\hat{\mathbf{x}}_{dirty} \in \mathbb{R}^{N \times M}$ consists in: (i) $\mathbf{P}_\Omega^T \mathbf{y}$ - expanding the observation \mathbf{y} to a $N \times M$ representation by placing zeros in the entries corresponding to Ω^C that is the complementary support set of Ω , (ii) adding the corresponding symmetrical signal via $S(\cdot)$ related to the symmetry property of Fourier, and then (iii) applying the inverse Fourier Ψ^* :

$$\hat{\mathbf{x}}_{dirty} = \Psi^* (\mathbf{P}_\Omega^T \mathbf{y} + S(\mathbf{P}_\Omega^T \mathbf{y})). \quad (3)$$

3.2. Clean image recovering

Due to the presence of missing Fourier frequencies in \mathbf{P}_Ω^C , the dirty image contains a lot of artifacts leading to the increase of the false detected sources. In order to remove these artifacts the CLEAN algorithm (Högbom 1974) is applied to the dirty image. The CLEAN is a fundamental method in radio astronomy and it consists of the following steps:

1. Find the intensity and position of the peak that is of the greatest absolute intensity, in the dirty image $\hat{\mathbf{x}}_{dirty}$.
2. Generate at this position a spike of an intensity equal to the product of a damping factor and the intensity at that position. Usually, the damping factor is ≤ 1 and is termed the loop gain.
3. Convolve this spike with the instrumental point source function (PSF).
4. Subtract this instrumental response from the dirty image $\hat{\mathbf{x}}_{dirty}$.

5. The steps 1-4 are repeated unless any remaining peak is below some user-specified level. The search for peaks may be constrained to specified areas of the image, called CLEAN windows.
6. Then the accumulated point sources are convolved with an idealized CLEAN beam that, usually, is an elliptical Gaussian fitted to the central lobe of the dirty beam.
7. Finally, the remaining residual of the dirty image is added to the result of step 6. The obtained reconstruction is called CLEAN image $\hat{\mathbf{x}}_{clean}$.

In terms of fast processing of large amounts of data, CLEAN, and also its accelerated versions like w -Stacking Clean (WSCLEAN; Offringa et al. 2014), is the main bottleneck. The reconstruction remains time-consuming since several parameters have to be adjusted in iterative runs.

3.3. Source localization

Traditionally the source localization is applied to the CLEAN images. Depending on the antenna configuration and the corresponding amount of artifacts, it might be also applied to the dirty images. Nowadays, there exists a broad range of different source localization approaches used in the traditional pipeline. Hopkins et al. (2015) provide a good overview of these methods. In our work we consider the PyBDSF (Mohan & Rafferty 2015).

4. Proposed framework

We propose a DNN-based framework that performs the source localization in the form of a binary map $\mathbf{m} \in \{0,1\}^{N \times M}$ directly from the uv-data by taking as an input only sampled visibility data without reconstruction of dirty or CLEAN images. The general scheme of the basic framework is illustrated in Fig. 4 and it consists of 3 steps:

1. Input data pre-processing (i.e., normalization);
2. DNN-processing: stage 1 and 2;
3. Post-binarization and source localization.

It should be pointed out that the sampled visibility data $\mathbf{y} \in \mathbb{C}^K$ are complex values. However, modern DNNs are not designed to work with complex values. To deal with this, we decompose the complex-valued uv-samples $\mathbf{y} \in \mathbb{C}^K$ into real-valued real and imaginary or magnitude and phase representation $\tilde{\mathbf{y}} \in \mathbb{R}^{2 \times K}$. Due to the nature of the Fourier transform, in the case of real and imaginary representation, the trained model g_θ at the stage 1 is additive in nature as shown in left Fig. 5. While in case of the magnitude and phase, it is multiplicative in nature as shown in right Fig. 5. The other steps are universal and remain unchanged⁷.

⁶ The sky model \mathbf{x} is of size $N \times M = 512 \times 512$. For the simplicity of notations we use the vectorized representation $N \cdot M$.

⁷ The python implementation of the proposed framework is publicly available at https://github.com/taran0/ml-based-source_localization_from_uv-plane.

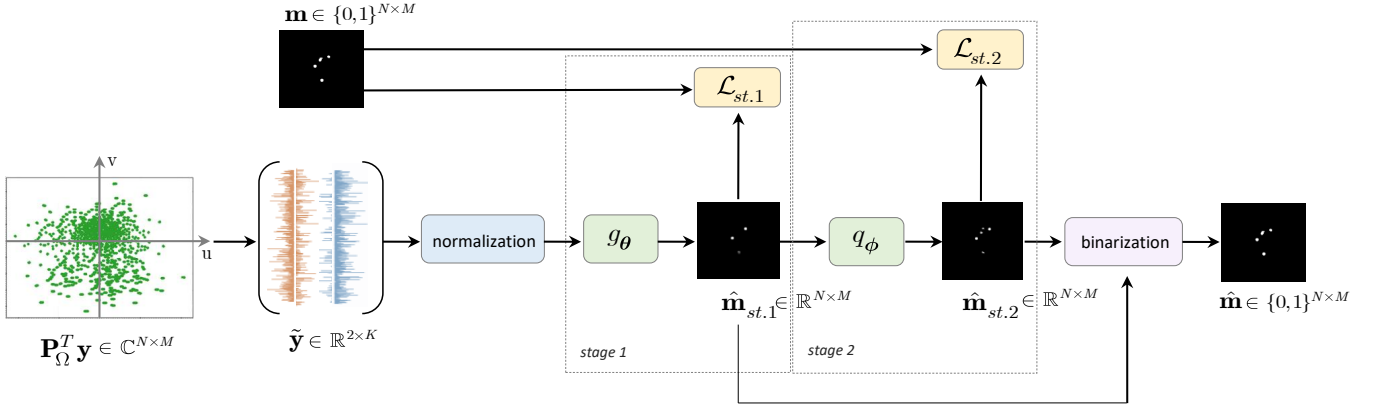
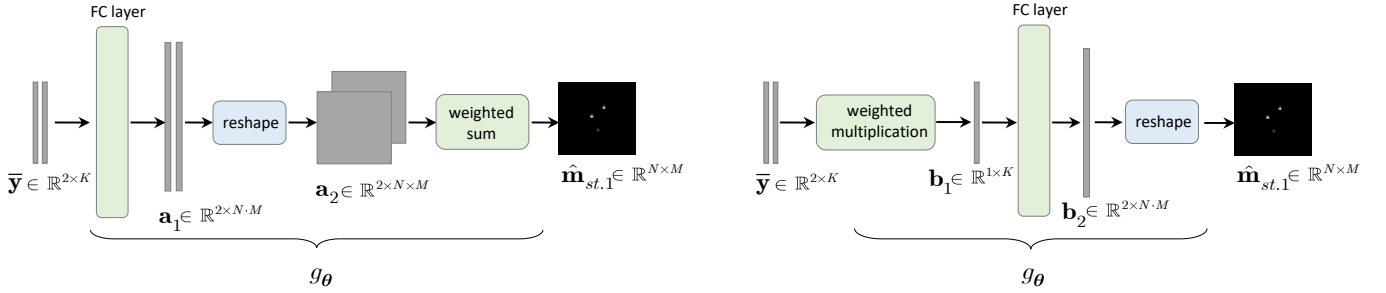


Fig. 4: Schematic representation of the proposed framework.

Fig. 5: Detailed architecture of the g_θ model, where $\bar{\mathbf{y}}$ denotes the normalized input data $\tilde{\mathbf{y}}$: in left panel $\bar{\mathbf{y}} \in \mathbb{R}^{2 \times K}$ is represented by the real and imaginary components; in right panel $\bar{\mathbf{y}} \in \mathbb{R}^{2 \times K}$ is represented by the magnitude and phase components.

4.1. Input data pre-processing

Examples of the raw noiseless and noisy input data $\tilde{\mathbf{y}} \in \mathbb{R}^{2 \times K}$ are given in Fig. 6. As can be seen from the noiseless case, the real and imaginary components have the same dynamic range of power 10^{-4} (Fig. 6 top left), while the magnitude and phase have different dynamic range (Fig. 6 top right): the magnitude is of power 10^{-4} similarly to the real and imaginary components and the phase is in a much wider range from -3 to 3 . Regarding the noisy case, it is important to note that in the real, imaginary and magnitude components the noise dominates and increases the dynamic range of data by two orders of magnitude, while the phase component is less affected by noise and preserves the same dynamic range from -3 to 3 . To guarantee stable DNN training and to avoid the vanishing of gradients, the real and imaginary components are normalized by multiplying by 1000 and clipped in the range $[-10, 10]$. The clipping allows to decrease the impact of strong outliers. The magnitude component is multiplied by 100 and clipped in the range $[0, 1]$ ⁸. The phase component is processed without any normalization. Empirically, such a type of normalization was found optimal during the proposed model training. The normalized vector is denoted as $\bar{\mathbf{y}} \in \mathbb{R}^{2 \times K}$.

4.2. DNN-processing

It is important to highlight that our goal is to perform the source localization, and we are not interested, for the time being, in the prediction of the sources fluxes and other parameters such as,

sizes etc. In this respect stages 1 and 2 are trained to minimize the similarity score with respect to the true binary source map $\mathbf{m} \in \{0, 1\}^{N \times M}$.

4.2.1. Stage 1: Real and imaginary representation

For the real and imaginary representation of the input data the schematic architecture of the model g_θ is shown in left Fig. 5⁹. At first, the given input low-dimensional representation $\bar{\mathbf{y}} \in \mathbb{R}^{2 \times K}$ is mapped into a higher dimension representation $\mathbf{a}_1 \in \mathbb{R}^{2 \times N \times M}$ via a fully connected layer. Then the obtained high dimension representation is reshaped into a square representation $\mathbf{a}_2 \in \mathbb{R}^{2 \times N \times M}$. Finally, the weighted sum of the obtained components produces the output $\hat{\mathbf{m}}_{st,1} \in \mathbb{R}^{N \times M}$.

4.2.2. Stage 1: Magnitude and phase representation

For the magnitude and phase representation of the input data the schematic architecture of the model g_θ is shown in right Fig. 5¹⁰. First, we perform the weighted element-wise multiplication of the two low-dimensional components in $\bar{\mathbf{y}} \in \mathbb{R}^{2 \times K}$. Then the obtained low-dimensional representation $\mathbf{b}_1 \in \mathbb{R}^{1 \times K}$ is mapped into a higher dimensional representation $\mathbf{b}_2 \in \mathbb{R}^{1 \times N \times M}$ via a fully connected layer. Finally, this high dimensional representation is reshaped to a square representation that corresponds to the output $\hat{\mathbf{m}}_{st,1} \in \mathbb{R}^{N \times M}$ of the stage 1, which is a real valued estimation of the source map.

⁸ As it can be seen from Fig. 6, the dynamic range of the magnitude is smaller than the dynamic range of the phase component. In this respect, we try to preserve this deviation and use the smaller normalization factor for the magnitude compared to the real and imaginary components.

⁹ The architecture details are given in Table D.1 in the Appendix D.

¹⁰ The architecture details are given in Table D.2 in the Appendix D.

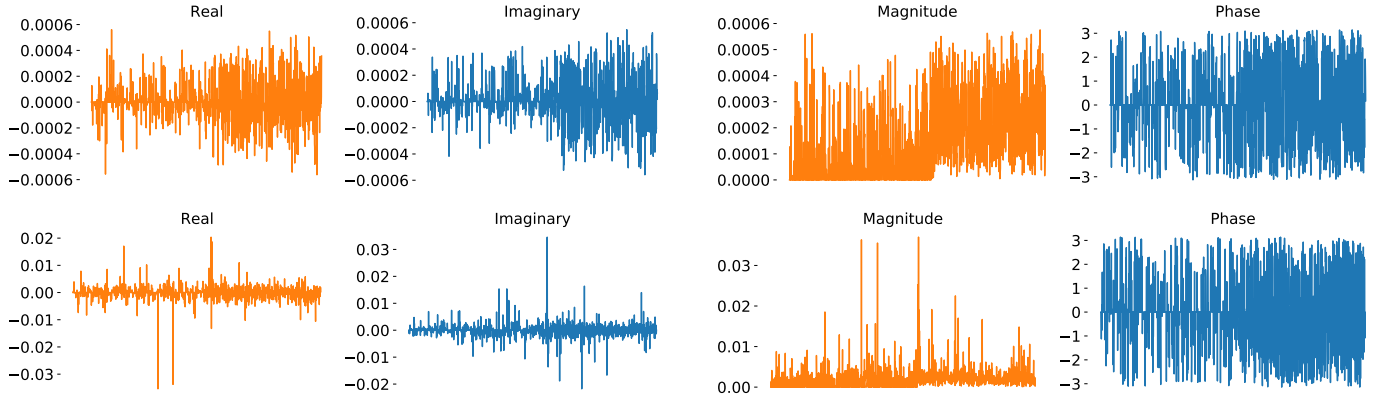


Fig. 6: Examples of input data: top left - the real and imaginary components in the noiseless case; top right - the magnitude and phase components in the noiseless case; bottom left - the real and imaginary components in the noisy case; bottom right - the magnitude and phase components in the noisy case.

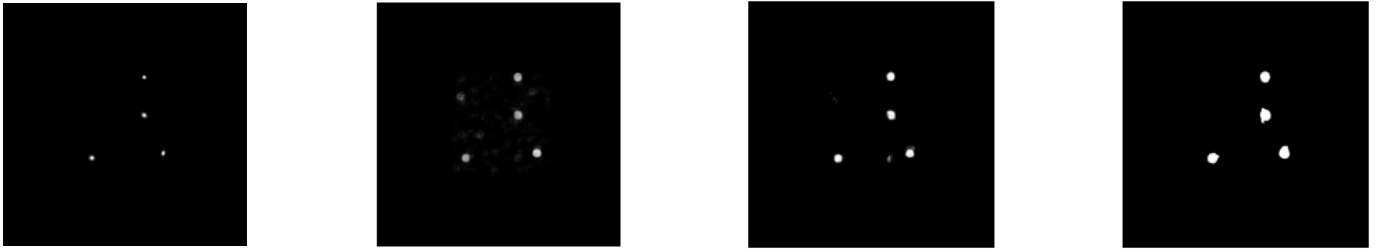


Fig. 7: From left to right: example of the true binary source map $\mathbf{m} \in \{0, 1\}^{N \times M}$; the proposed framework estimations $\hat{\mathbf{m}}_{st,1} \in \mathbb{R}^{N \times M}$ and $\hat{\mathbf{m}}_{st,2} \in \mathbb{R}^{N \times M}$ based on the real and imaginary input data representation (for the magnitude and phase representation the results look similar); the final estimation $\hat{\mathbf{m}} \in \{0, 1\}^{N \times M}$.

The estimation of the parameters θ of the trained model g_θ is based on solving the minimization problem:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}_{st,1}(\theta) = \mathcal{L}_{\text{mse}}(\mathbf{m}, \hat{\mathbf{m}}_{st,1}), \quad (4)$$

where $\hat{\mathbf{m}}_{st,1} = g_\theta(\bar{\mathbf{y}})$ and $\mathcal{L}_{\text{mse}}(\cdot, \cdot)$ denotes the mean square loss.

4.2.3. Stage 2

Due to the fact that the network prediction $\hat{\mathbf{m}}_{st,1} \in \mathbb{R}^{N \times M}$ is real valued, the model g_θ preserves some information about the true source intensity. Therefore, some predicted sources might be of low intensity as shown in Fig. 7 (the second sub-figure). As a result, after the binarization certain sources might be lost, especially, in the noisy case. In this respect, the stage 2 might be considered as a quality enhancement stage for the source detection where, as can be seen from Fig. 7 (the third sub-figure) the predicted source intensity is close to the binary representation. Taking into account the simple nature of the expected prediction in a form of a simple binary map without any complex shapes and textures, as a model q_ϕ we use a simple auto-encoder model¹¹. For the more complicated tasks like, for example, source intensity or any other physical parameters estimation, the model q_ϕ might be represented by more advanced models like, for example, UNet (Long et al. 2015) or Transformers (Vaswani et al. 2017).

¹¹ The architecture details of the used auto-encoder are given in Table D.3 in the Appendix D.

The estimation of the parameters ϕ of the trained model q_ϕ is based on solving the minimization problem:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{st,2}(\phi) = \mathcal{L}_{\text{mse}}(\mathbf{m}, \hat{\mathbf{m}}_{st,2}), \quad (5)$$

where $\hat{\mathbf{m}}_{st,2} = q_\phi(g_{\theta^*}(\bar{\mathbf{y}}))$, θ^* denotes fixed pre-trained model parameters and $\mathcal{L}_{\text{mse}}(\cdot, \cdot)$ denotes the mean square loss.

4.3. Post-binarization and source localization

Taking into account a need to satisfy the differentiability of DNN for the gradient propagation at the training, the DNN can not produce the binary outputs. In this respect, the post-binarization stage is necessary. However, instead of performing the hard thresholding that might produce source blobs of uncontrollable size, a morphological-based binarization is used.

The morphological binarization consists in determining the optimal threshold value¹² for every particular DNN output $\hat{\mathbf{m}} \in \mathbb{R}^{N \times M}$ on-the-fly. After the binarization, the connected neighborhoods are detected and too small or too big regions are rejected. Finally, the source position is estimated for each detected region by taking their centroids.

4.4. Conceptual advantages

Compared to the traditional pipeline, conceptually, the proposed framework has the following advantages:

¹² More details are given in Appendix D.

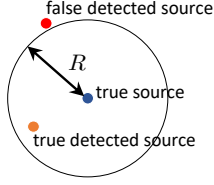


Fig. 8: Schematic representation of the acceptable distance R between the true and predicted sources.

- The input data dimensionality is much smaller: only the sampled uv-data are processed to produce a localization binary map \mathbf{m} , while the traditional pipeline works with the full size uv-plane where missing frequencies are filled with zeros.
- The proposed framework might be extended to any type of data sources and even for mixed types, while the CLEAN method used in the traditional pipeline has a poor reconstruction of regions of extended emission. Due to this, it is applied usually to the sky models that are only composed of point sources. For the regions of extended emission other methods are used. In this respect, the traditional pipeline is more demanding in terms of the expert knowledge.
- The CLEAN reconstruction as well as dirty image estimation are not required as such to do the source detection that represents a considerable computational advantage in practice.

5. Results and analysis

By default and unless specified, all results are given as an average over three test subsets mentioned in Sect. 2.

For simplicity we name the proposed framework based on the real and imaginary components input data representation as Re & Im and based on the magnitude and phase components as Mag & Phase.

5.1. Metrics

We cross-matched detected sources with true sources in each sky model using a match radius R equal to the beam size as shown in Fig. 8 to compute model performance metrics listed below. The impact of the matched radius in the results is discussed in Sect. 5.3.

To evaluate the performance of the traditional pipeline and the proposed framework we use the *purity* and *completeness* metrics defined as follows.

Purity shows the fraction of the true sources among all detected sources:

$$\text{Purity} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (6)$$

where TP and FP denote true positive and false positive sources, respectively. $1 - \text{purity}$ represents the fraction of false detections.

Completeness equals to the fraction of true sources that are successfully detected:

$$\text{Completeness} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

where FN is the false negative (missing) sources.

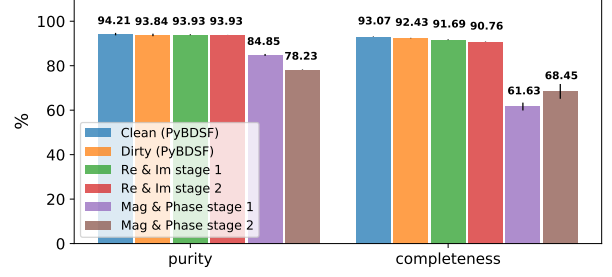


Fig. 9: The purity and completeness of the methods under investigation in the noiseless case. The small black lines on the top of each bar correspond to the standard deviation from the average.



Fig. 10: Difference between the noiseless (left) and noise (right) prediction of the proposed Re & Im stage 1 for a simulated sky model.

5.2. Results on noiseless data

We consider the noiseless case as an ideal condition measurement. In real observations, this reflects a hypothetical situation, since all measurements are corrupted by noise of different nature. However, we found it to be important to validate the performance of the investigated approaches under the assumption that there is no noise to understand their baseline performance.

Fig. 9 demonstrates the purity and completeness obtained for PyBDSF applied to the CLEAN and dirty images and the proposed framework in four different configurations. We choose the source detection parameters in such a way as to have the maximum completeness under the assumption that the acceptable purity should be about 94 %¹³. It is easy to see that PyBDSF on the CLEAN and dirty images and the proposed Re & Im framework have, in general, the same performance. There is no big difference in the performance of PyBDSF at the CLEAN and dirty images. The Re & Im stage 1 and 2 provide very close results. In the case of Mag & Phase, the performance is worse. As it is mentioned in Sect. 4.1, there is a big difference in the dynamic range of the magnitude and phase components. From one side, such a difference is natural and should be preserved. From the other side, it is a disadvantage and a challenge for the DNN-training. This explains the obtained non-optimal results.

¹³ The used PyBDSF parameters are $\text{thresh_pix} = 7$ and $\text{thresh_isl} = 5$. In the proposed framework area_lim parameter was set to 125 in case of Re & Im at stage 1 and to 200 at stage 2. In Mag & Phase the same parameter was set to 300 and 310 at stage 1 and stage 2 correspondingly.

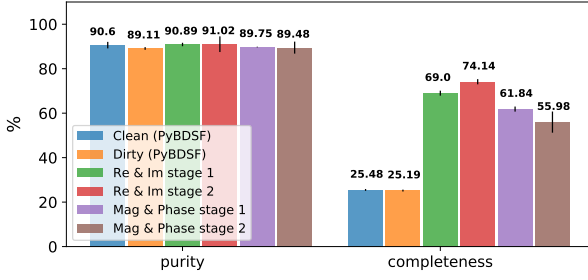


Fig. 11: The purity and completeness of the methods under investigation in the noisy case, averaged over all SNRs. The small black lines on the top of each bar correspond to the standard deviation from the average.

5.3. Results on noisy data

The noisy scenario is of a particular interest for our study of real observations. In this respect, we provide a more detailed analysis of the obtained results.

Before presenting the results of the proposed framework for the noisy case, we would like to underline the difference in the predictions between the noiseless and noisy case. It helps understand why the proposed framework is better than the traditional pipeline. The noiseless and noisy predictions of the Re & Im after stage 1 are shown in Fig. 10. It is important to highlight that, in contrast to the CLEAN or dirty noisy data used in the traditional pipeline (Fig. 1), the proposed framework predictions based on the noisy input data are free from the background noise which is very important for the efficient source localization.

In Fig. 11 we show the purity and completeness for PyBDSF and the proposed framework in four different configurations. For a fair comparison, the parameters of source localization for all methods under investigation are selected in such a way to have the purity about 90 %¹⁴. We fixed and used these parameters for all following experiments.

In comparison to the noiseless case, in the noisy case we have slightly smaller purity (about 3-4 % less), but the obtained completeness is significantly smaller for all methods under investigation. Secondly, it is important to note that PyBDSF completeness does not exceed 25 - 26 %, which is very small for practice. At the same time, it is interesting that the proposed Mag & Phase framework that has the worst performance in the noiseless case, outperforms the state-of-the-art PyBDSF on the noisy data. Its completeness is about 60 % after stage 1 and about 55 % after stage 2. The proposed Re & Im framework demonstrates the best completeness about 69 % after stage 1 and 74 % after stage 2. However, it should be pointed out that stage 1 is more stable as shown by the smaller standard deviation.

It is obvious that the performance of the source localization highly depends on SNR value. In Fig. 12 we plot the completeness obtained for each method under investigation with respect to SNR of the source. One can see that when $\text{SNR} \leq 3$ PyBDSF can not detect many sources (the completeness does not exceed 2-3 %). While for $\text{SNR} \sim 2$ the proposed Mag & Phase reaches 10 % and for $\text{SNR} \sim 3$ it already has 25 % completeness. The

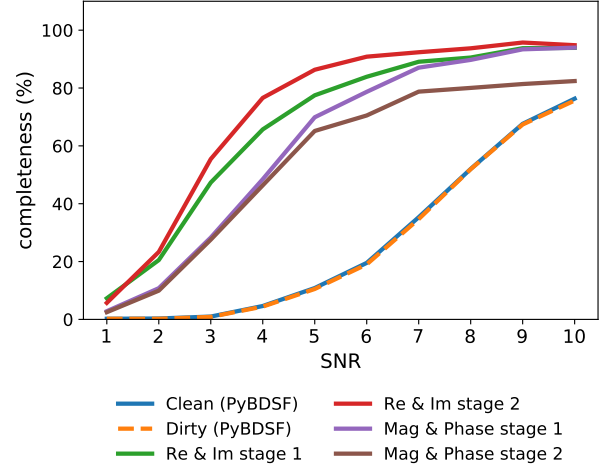


Fig. 12: Dependence of completeness on SNR of the sources.

performance of the proposed Re & Im is even better: for $\text{SNR} \sim 2$ the completeness is about 20 % and for $\text{SNR} \sim 3$ it is about 50 % after the stage 1 and about 55 % after the stage 2. With the increasing SNR the performance of all methods increases. However, for the maximum SNR the state-of-the-art PyBDSF can get the completeness only about 75 %, while the proposed Re & Im achieves 93 - 94 %. It is also important to note that the Re & Im stage 2 gives improvement for $\text{SNR} 2 - 5$ compared to the Re & Im stage 1. For the very small or high SNR the performance at both stages is similar.

As mentioned in Sect. 2 one of the challenges of our data set is closely located sources which are difficult to distinguish especially in the noisy case with increasing source densities. It is obvious when we have more sources in the sky model then the probability to have close sources is higher. We analyze the performance of PyBDSF and the proposed Re & Im with respect to the source SNR specifically for different numbers of sources in the sky models. The derived results are shown in Fig. 13. In general, the obtained behavior is similar to the one shown in Fig. 12. However, there are several interesting facts that one can notice. First of all, the behavior of PyBDSF is stable and does not depend too much on the number of sources in the sky model. On the other hand, when the number of sources increases the efficiency of the proposed framework decreases especially for the low SNR. For example, for $\text{SNR} \sim 2$ in the case with only 1 source (Fig. 13) the completeness is about 50 %, while in the case with 5 sources (Fig. 13) the completeness drops till about 15 - 20 %. With the SNR increase, the drop in efficiency slows down. This phenomenon is naturally connected with the increased entropy of multi-source spectra. Keeping the fixed number of samples in the uv-plane, the sky model with several sources is characterized by a lower entropy spectrum. Therefore, there is a fundamental accuracy limit of source detection under the restricted sparse sampling of uv-plane. This is one of the directions for our future investigation. At the same time, it is important to point out that for the setup under investigation and the chosen antennas configuration (for more details see Sect. 2), the expected number of sources ranges from 1 to 5. For the larger number of sources, the ambiguity in the under-sampled uv-data increases. For this reason the configuration with the larger number of antennas should be used. Under the chosen setup, even though the completeness of Re & Im drops, its general efficiency is much better than PyBDSF.

¹⁴ We set PyBDSF parameter *rms* to 4.2450E-05 for the CLEAN data and to 4.275E-05 for the dirty case. In the proposed framework *area_lim* parameter was set to 75 in case of Re & Im at stage 1 and to 240 at stage 2. In Mag & Phase the same parameter was set to 200 and 340 at stage 1 and stage 2 correspondingly.

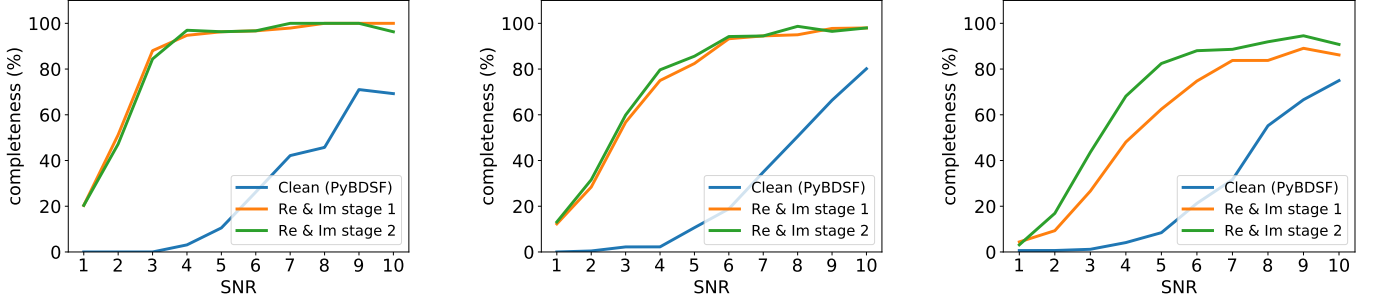


Fig. 13: Dependence on the efficiency of PyBDSF and the proposed Re & Im framework on the number of sources in the sky model: left - 1 source; middle - 3 sources; right - 5 sources.

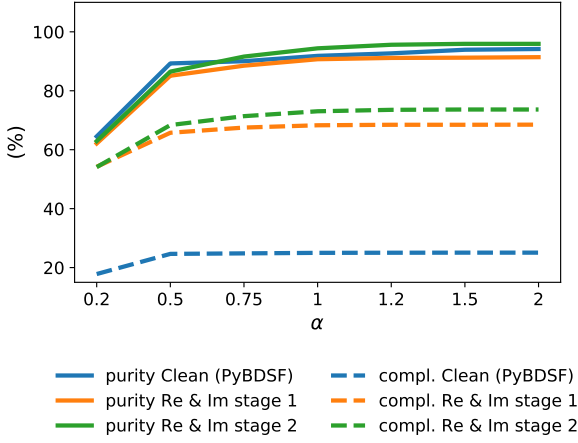


Fig. 14: Dependence on α , where $\alpha = \frac{R}{\text{beam size}}$ and R is the distance between the true and predicted sources, in the noisy case, averaged over all SNRs.

Another important question that we study is the dependency of source localization accuracy on the distance to the true sources. In Fig. 8 we schematically show the locations of the true source and the corresponding predicted source. We consider the predicted source as true if it is inside a given radius R around the true source. In the current experiments we set $R = \alpha \cdot \text{beam size}$ ¹⁵ where α is a scale factor. The results shown in Fig. 14 are obtained for the test subset 1¹⁶. It is important to mention that in all previous experiments $\alpha = 1$ or, in other words, the radius R equals to the beam size. In Fig. 14 we can observe some increase of the purity when $\alpha > 1$, but the saturation is achieved already for $\alpha = 1.2 - 1.5$. For the completeness, the saturation is achieved for $\alpha = 1$. In Fig. 15 we plot the mean distance in arcsec between the centers of true and detected sources. The gray bars show the relative number of detected sources. The relative means were divided by the total number of true sources for the given SNR. It should be pointed out, that in the case of the Re & Im framework the deviation from the mean value is quite stable and does not depend a lot on the number of detected sources or SNR. At the same time, with the increase of SNR and the number of detected sources the mean value converges to about 0.16'' after stage 1 and to about 0.19'' after stage 2. For PyBDSF

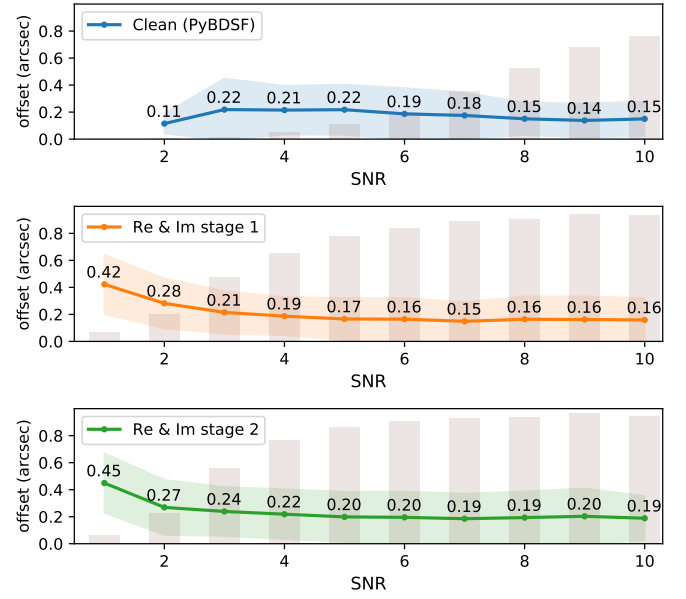


Fig. 15: The mean distance (the y-axis) between the true and predicted sources, in the noisy case. The coloured semi-transparent background shows the standard deviation from the mean value. The gray bars show the relative number of detected sources.

one can see quite small deviation from the mean values for SNR = 2 and the big deviation for SNR = 3. This can be explained by the very small number of detected sources and, as a consequence, the poor statistics. With the increase of SNR and the number of detected sources the deviation from the mean value decreases and the mean value converges to about 0.15''. It is important to note that Re & Im framework reaches the convergence at SNR = 5 while for PyBDSF we can observe the convergence only after SNR = 8.

5.4. Execution time

To investigate the question of the time complexity, we measure the execution time of the source localization for the methods under investigation. The obtained CPU time in sec¹⁷ is summarized

¹⁵ The size of the beam used in the simulation is about 0.82''.

¹⁶ In Fig. 11 one can see significant deviation from the average for Re & Im stage 2 in comparison to Re & Im stage 1. This explains why in Fig. 14 the performance of stage 2 is better than of stage 1.

¹⁷ The inference of the proposed framework might be efficiently run at GPU. However, taking into account that the traditional methods are usually run on CPU and that not all users have access to GPU, we performed the comparison with respect to CPU to have a fair comparison and to show that the proposed framework is easily runnable on CPU.

Table 3: Execution time for the source localisation per sky model (noisy data).

Approach	Execution CPU time, sec		
	reconstruction from uv	source localization	total
dirty + CLEAN + PyBDSF	17.29	1.48	18.77
dirty + PyBDSF	13.97	1.51	15.48
Re & Im stage 1	0.09	0.18	0.27
Re & Im stage 1 and 2	0.15	0.34	0.49
Mag & Phase stage 1	0.25	0.22	0.47
Mag & Phase stage 1 and 2	0.33	0.51	0.84

in Table 3. For the proposed framework the "reconstruction from uv" corresponds to the DNN-processing, Sect. 4.2. The column "source localization" corresponds to the procedure explained in Sect. 4.3. It should be pointed out that the recovery of the dirty image takes about 14 sec. The recovery of the CLEAN image takes about 17 sec. On the other hand, the proposed framework in the slowest case (Mag & Phase after two stages) estimates the real-valued source map only in 0.33 sec. The source localization by PyBDSF compared to the most efficient Re & Im stage 2 in about 4.3 times slower. In terms of the total time complexity, PyBDSF on the CLEAN data (the best traditional pipeline results) in ~ 38 times slower than the proposed Re & Im stage 2 (the best results) even on the CPU execution.

6. Discussion

6.1. Performance comparisons with the literature

In classical approaches to interferometric observations, fidelity and completeness are two important measures of the significance and "statistical importance" resulting from source detection and characterization, which are generally determined by empirical methods. For example, the fidelity is computed by comparing the number of sources detected with positive flux to those in the negative image (Fidelity = $1 - \frac{N_{neg}}{N_{pos}}$) to empirically determine a SNR threshold above which individual sources are considered "reliably detected" (see, e.g., Aravena et al. 2016; Béthermin et al. 2020).

And completeness is generally determined using Monte Carlo source injections in the observational data (e.g., in the image plane).

Since fidelity requires a flux measurement, which the proposed framework does not do, it is not possible to compare the performance of our method for this quantity, as already mentioned above (Sect. 5.1). However, we have compared our results to those of Béthermin et al. (2020), who present a detailed analysis of the completeness from extra-galactic ALMA continuum observations with properties similar to those of our sky simulations. Following Béthermin et al. (2020), we therefore determine the completeness as a function of the normalized injected flux:

$$f_{\text{norm}} = \frac{\text{total flux}}{\sigma_{\text{noise}}} \cdot \frac{b_{\text{min}} \cdot b_{\text{maj}}}{\sqrt{b_{\text{min}}^2 + s_{\text{min}}^2} \cdot \sqrt{b_{\text{maj}}^2 + s_{\text{maj}}^2}}, \quad (8)$$

where b_{min} and b_{maj} correspond to the minor and major axes of the beam, and s_{min} and s_{maj} denote the minor and major axes of the source. This quantity resembles an effective, normalized SNR, and it encapsulates in particular variations of sources sizes

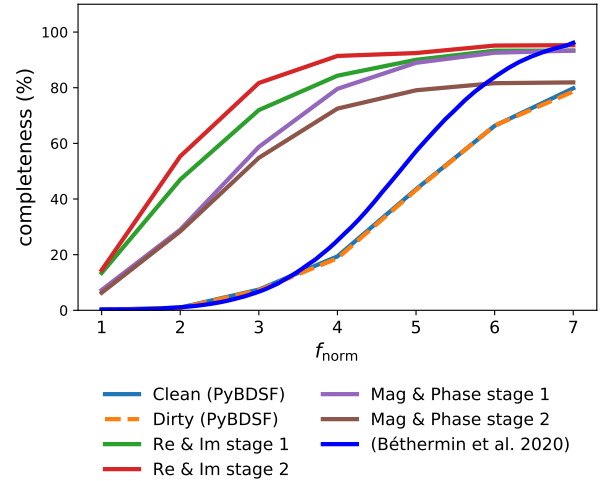


Fig. 16: Dependence of completeness on normalized SNR compared to Béthermin et al. (2020).

and provides a simple functional description of the completeness, as shown by Béthermin et al. (2020).

The results obtained on our simulated data are shown in Fig. 16. First we note that the behavior of PyBDSF on our CLEAN and dirty images is similar to the *find_peak* source detector from *astropy* (Robitaille et al. 2013) used by Béthermin et al. (2020), especially at $f_{\text{norm}} \lesssim 4$. Most importantly, for all normalized fluxes $f_{\text{norm}} \lesssim 5$ the proposed framework achieves significantly higher completeness than classical methods. Finally, for high normalized fluxes (and high SNR), the completeness obtained by the proposed framework and Béthermin et al. (2020) are similar. From this comparison we conclude that the proposed method should not only significantly speed up, but also strongly improve source detection in interferometric imaging.

6.2. Impact of the noise

As it is mentioned in Sect. 2, in the ALMA setup under investigation there are two sources of the noise. The first one is the ALMA receiver noise that is fixed in our case, since we fix the total integration time and observing band. The second one is the atmospheric noise related to the water vapor. To investigate the sensitivity of the proposed framework to the change of noise statistics we consider the change of the atmospheric noise by varying the *pwv* parameter. We choose 10 different *pwv* values from 0.472 to 5.186. These values are chosen from the ALMA

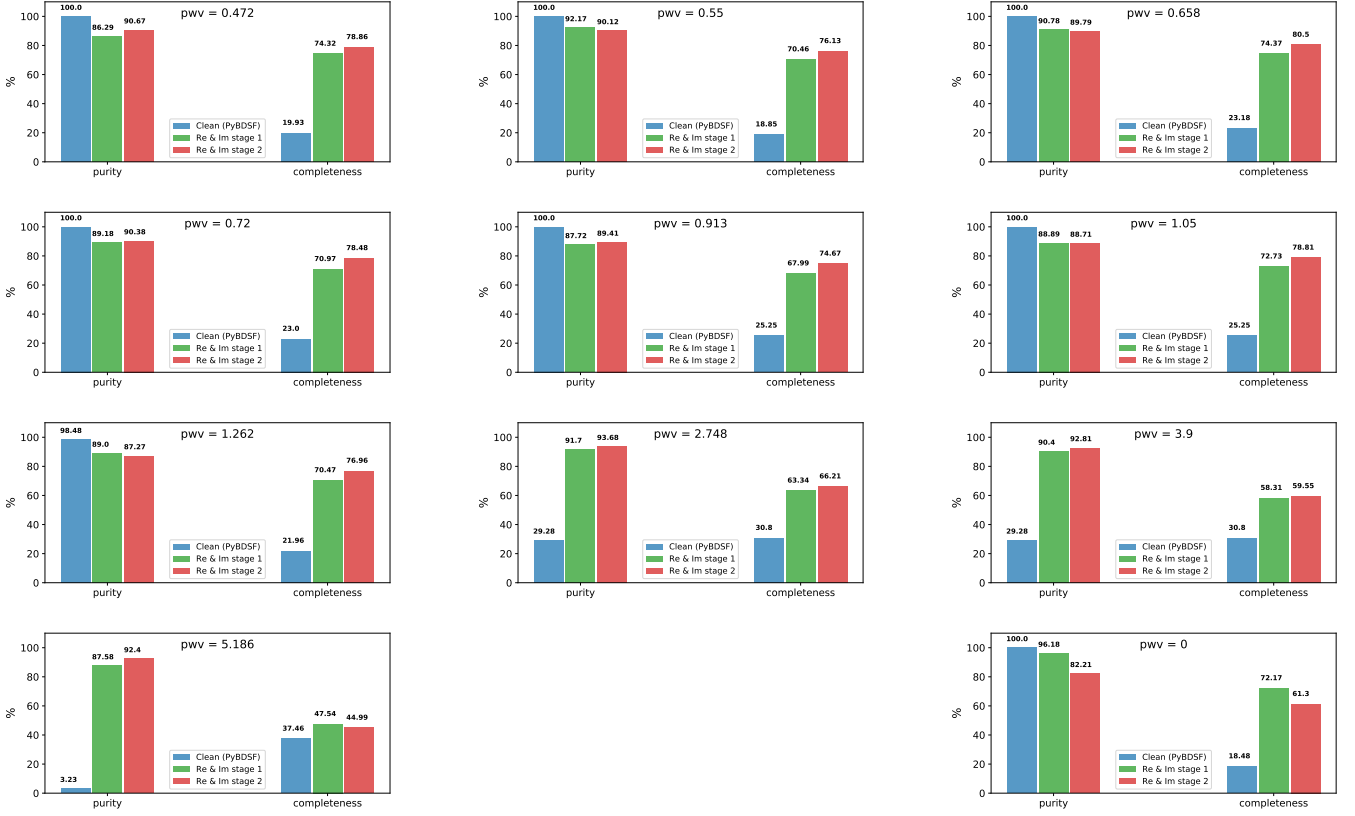


Fig. 17: The purity and completeness of PyBDSF on CLEAN data and the proposed Re & Im framework for the different noise levels, averaged over all SNRs. From top left to bottom right: noise (pwv) = 0.472; noise (pwv) = 0.55; noise (pwv) = 0.658; noise (pwv) = 0.72; noise (pwv) = 0.913; noise (pwv) = 1.05; noise (pwv) = 1.262; noise (pwv) = 2.748; noise (pwv) = 3.9; noise (pwv) = 5.186; noise (pwv) = 0.

technical handbook¹⁸ and represent the typical values observed at the ALMA site. For each pwv value we simulated 100 test samples. Then we test the proposed Re & Im framework without retraining and without any change of test parameters and perform the comparison with PyBDSF that is also used without any change of parameters for a fair comparison. The obtained purity and completeness are shown in Fig. 17. It is important to notice that, on average, for the Re & Im framework one can observe the same dynamic as represented in Sect. 5.3, namely, stage 2 is slightly superior to stage 1. The performance of PyBDSF under the noise pwv smaller than in the main simulations (i.e., pwv = 1.796) is similar to the one reported in Sect. 5.3. With the increase of the noise we can observe a drastic decrease in purity from 98 - 100 % to 30 % and to 3 % for the pwv = 5.186. This shows the need in the expert knowledge to adapt the parameters. For the proposed Re & Im framework there is also decrease in the performance but not so drastic. On average, the purity remains about 87 - 90 %, while the completeness decreases from about 60 - 65 % to 45 - 47 % that is 10 % better than PyBDSF.

In addition we investigate an extreme case that is the performance obtained on the noiseless data (pwv = 0) with the models trained on the noisy data. The obtained results are shown in Fig. 17 (bottom right). In general, the behavior of the proposed Re & Im framework is quite good. One can observe a certain decrease of the performance at stage 2, while the purity and completeness obtained at stage 1 are high. The completeness obtained by

PyBDSF is smaller than in case of noisy data. This can be explained by the choice of parameters assuming the presence of noise.

6.3. Caveats

It should be pointed out that, despite the advantages discussed above, the proposed framework also has certain limitations.

Although it requires less expert knowledge, the amount of training data increases with increasing data complexity such as, for example, the complexity of the shape of the source and their variability, the increase in the number of sources in the field of view, the proximity of the sources with different SNR, etc.

Moreover, taking into account that the proposed framework shows high sensitivity to the low SNR sources, it might lead to false sources in the source free sky models (e.g., pure background noise). And although such a scenario usually is not considered in the state-of-the-art, we tested the proposed Re & Im framework on 1000 source free noisy sky models. After stage 1, the false sources are detected in about 40 % of the sky models, while after stage 2, only in 27 % of cases. The results obtained can be explained by the fact that the source free sky models were not taken into account during the framework training.

6.4. Future developments and applications

The proposed framework can be applied to real data, and will be tested with available data from the ALMA archive, such as data

¹⁸ <https://almascience.nrao.edu/proposing/technical-handbook>

from A³COSMOS (Liu et al. 2019). This will in particular also allow us to examine the behavior with data taken in different conditions and accounting for all noise sources present in the system.

The next tasks to be tackled from uv-data alone will include source characterization that are measurements of fluxes, source morphologies and others, and the treatment of extended sources, possibly with complex morphologies. New machine learning based approaches on visibilities should also be able to handle spectral lines and thus be applicable to an even broader range of astrophysical questions. In parallel, our framework should also be tested and generalized to cover a wide range of interferometric data taken with very different facilities and across a wide spectral range from the millimeter to the radio domain. If proven successful, machine learning based methods could have a strong impact on our future handling of interferometric data and help push the discovery space of upcoming observatories even further.

7. Conclusions

Radio astronomy is at a historic moment in its development. Innovative DNN-based methods may drastically change the way measurements are processed and thus also improve our capacities to detect faint and complex signals. In this context, we have taken a new ML-based approach to solve the problem of source localization, directly working on the natural measurement sets (visibilities) and without reconstructing the dirty or CLEAN image. The proposed framework consists of two stages generating source localization maps (see Fig. 4).

To train the network, validate and test the proposed framework, we have used synthetic data generated with the CASA data processing software (McMullin et al. 2007), the software tool for interferometric data/observations with ALMA and similar observatories, and PyBDSF (Mohan & Rafferty 2015) for the classical source detection. The sky simulations generated in this work were chosen to represent typical extra-galactic (sub)-millimeter continuum survey observations undertaken with ALMA. The simulations included both ideal noiseless and realistic noisy simulations. The comparisons between the proposed DNN-based approach and existing state-of-the-art source localization pipelines can be summarized as follows:

- In ideal noiseless cases, the proposed framework has overall the same purity and completeness as the traditional pipeline using PyBDSF source localization.
- For noisy data, the proposed DNN-based method fares significantly better than classical source localization algorithms. For example, the Re & Im framework shows an almost 3 times higher completeness for the same purity as the traditional PyBDSF-based pipeline for all sky models, considering a uniform, random distribution of sources with SNR = 1 – 10 (Fig. 12).
- In the low SNR-regime ($\text{SNR} \lesssim 5$) the performance gain of the proposed method is very high: whereas the traditional pipeline achieves a completeness of 2 – 5 % for $\text{SNR} < 3$, the proposed framework detects sources with a completeness of ~ 20 (45 – 55) % for $\text{SNR} = 2$ (3).
- The new source detection method represents an important gain in execution time, with total execution times which are more than 30 times faster than the traditional pipeline involving image reconstruction from the uv-plane and source detection (Table 3).

We have investigated the impact of different factors on the efficiency of the proposed framework, in particular:

- *Input data representation*: While traditional DNNs are designed to work with real-valued data, we have tested representations of the complex-valued uv-data by real-valued real and imaginary or magnitude and phase components. The obtained results show that the real and imaginary components suit better for the proposed framework, since they have the same dynamic range in contrast to the magnitude-phase case.
- *Source density*: Traditional source detection in the image plane demonstrates a stable behavior that does not depend much on the number of sources. However, in the proposed framework the completeness decreases with increasing numbers of sources (Fig. 13). Despite this, in tested conditions which represent typical ALMA extra-galactic continuum surveys, the proposed framework reaches a higher completeness than traditional source detection algorithms.

In short, we have developed a DNN-based method of source detection using only uv-plane observations and shown that it provides strong improvements in detecting sources in the low-to-intermediate SNR regime. The new approach can already be applied to existing interferometric observations, and it opens many new possibilities which will be explored in the near future. Machine learning based methods have the potential to significantly alter our approach to interferometric data, as we enter the era of new facilities like the Square Kilometer Array¹⁹ (SKA).

Acknowledgements. We acknowledge the referee for their comments. O. Taran and O. Bait are supported by the *AstroSignals* Sinergia Project funded by the Swiss National Science Foundation. We also thank B. Magnelli and M. Béthermin, and the nordic arc node <https://nordic-alma.se>.

References

- Aniyan, A. K. & Thorat, K. 2017, *ApJS*, 230, 20
- Aravena, M., Decarli, R., Walter, F., et al. 2016, *ApJ*, 833, 71
- Becker, R. H., White, R. L., & Helfand, D. J. 1995, *ApJ*, 450, 559
- Bertin, E. & Arnouts, S. 1996, *A&AS*, 117, 393
- Béthermin, M., Fudamoto, Y., Ginolfi, M., et al. 2020, *Astronomy & Astrophysics*, 643, A2
- Béthermin, M., Fudamoto, Y., Ginolfi, M., et al. 2020, *A&A*, 643, A2
- Bouwens, R. J., Smit, R., Schouws, S., et al. 2022, *ApJ*, 931, 160
- Bowles, M., Scaife, A. M. M., Porter, F., Tang, H., & Bastien, D. J. 2021, *MNRAS*, 501, 4579
- Carbone, D., Garsden, H., Spreeuw, H., et al. 2018, *Astronomy and Computing*, 23, 92
- Condon, J. J. 1997, *PASP*, 109, 166
- Condon, J. J., Cotton, W. D., Greisen, E. W., et al. 1998, *AJ*, 115, 1693
- Delli Veneri, M., Tychoniec, L., Guglielmetti, F., Longo, G., & Villard, E. 2022, *MNRAS[arXiv:2211.11462]*
- Faisst, A. L., Schaerer, D., Lemaux, B. C., et al. 2020, *ApJS*, 247, 61
- Franco, M., Elbaz, D., Béthermin, M., et al. 2018, *A&A*, 620, A152
- Hale, C., Robotham, A., Davies, L., et al. 2019, *Monthly Notices of the Royal Astronomical Society*, 487, 3971
- Hancock, P. J., Murphy, T., Gaensler, B. M., Hopkins, A., & Curran, J. R. 2012, *MNRAS*, 422, 1812
- Hancock, P. J., Trott, C. M., & Hurley-Walker, N. 2018, *PASA*, 35, e011
- Heywood, I., Jarvis, M. J., Hale, C. L., et al. 2022, *MNRAS*, 509, 2150
- Högbom, J. 1974, *Astronomy and Astrophysics Supplement Series*, 15, 417
- Hopkins, A. M., Whiting, M. T., Seymour, N., et al. 2015, *PASA*, 32, e037
- Hotan, A. W., Buntton, J. D., Chippendale, A. P., et al. 2021, *PASA*, 38, e009
- Jarvis, M., Taylor, R., Agudo, I., et al. 2016, in *MeerKAT Science: On the Pathway to the SKA*, 6
- Jonas, J. & MeerKAT Team. 2016, in *MeerKAT Science: On the Pathway to the SKA*, 1
- Le Fèvre, O., Béthermin, M., Faisst, A., et al. 2020, *A&A*, 643, A1
- Liu, D., Lang, P., Magnelli, B., et al. 2019, *ApJS*, 244, 40

¹⁹ <https://www.skao.int/>

- Long, J., Shelhamer, E., & Darrell, T. 2015, in Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440
- Lukic, V., Brüggen, M., Mingo, B., et al. 2019a, MNRAS, 487, 1729
- Lukic, V., de Gasperin, F., & Brüggen, M. 2019b, Galaxies, 8, 3
- Ma, Z., Xu, H., Zhu, J., et al. 2019, ApJS, 240, 34
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in Astronomical data analysis software and systems XVI, Vol. 376, 127
- Mohan, N. & Rafferty, D. 2015, Astrophysics Source Code Library, ascl
- Norris, R. P., Afonso, J., Bacon, D., et al. 2013, PASA, 30, e020
- Norris, R. P., Hopkins, A. M., Afonso, J., et al. 2011, PASA, 28, 215
- Offringa, A., McKinley, B., Hurley-Walker, N., et al. 2014, Monthly Notices of the Royal Astronomical Society, 444, 606
- Rezaei, S., McKean, J. P., Biehl, M., & Javadpour, A. 2022, MNRAS, 510, 5891
- Riggi, S., Ingallinera, A., Leto, P., et al. 2016, Monthly Notices of the Royal Astronomical Society, 460, 1486
- Riggi, S., Magro, D., Sortino, R., et al. 2022, arXiv e-prints, arXiv:2212.02538
- Riggi, S., Vitello, F., Becciani, U., et al. 2019, Publications of the Astronomical Society of Australia, 36, e037
- Robitaille, T. P., Tollerud, E. J., Greenfield, P., et al. 2013, Astronomy & Astrophysics, 558, A33
- Robotham, A., Davies, L., Driver, S., et al. 2018, Monthly Notices of the Royal Astronomical Society, 476, 3137
- Scaife, A. M. M. 2020, Philosophical Transactions of the Royal Society of London Series A, 378, 20190060
- Schmidt, K., Geyer, F., Fröse, S., et al. 2022, A&A, 664, A134
- Serra, P., Westmeier, T., Giese, N., et al. 2015, Monthly Notices of the Royal Astronomical Society, 448, 1922
- Smolčić, V., Novak, M., Bondi, M., et al. 2017, A&A, 602, A1
- Tang, H., Scaife, A. M. M., & Leahy, J. P. 2019, MNRAS, 488, 3358
- THE CASA TEAM, Bean, B., Bhatnagar, S., et al. 2022, arXiv e-prints, arXiv:2210.02276
- Tingay, S. J., Goeke, R., Bowman, J. D., et al. 2013, PASA, 30, e007
- Vafaei Sadr, A., Vos, E. E., Bassett, B. A., et al. 2019, MNRAS, 484, 2793
- van Haarlem, M. P., Wise, M. W., Gunst, A. W., et al. 2013, A&A, 556, A2
- Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, Advances in neural information processing systems, 30
- Walter, F., Decarli, R., Aravena, M., et al. 2016, ApJ, 833, 67
- Westmeier, T., Kitaëff, S., Pallot, D., et al. 2021, Monthly Notices of the Royal Astronomical Society, 506, 3962

Appendix A: UV Coverage and the dirty beam

Fig. A.1 shows the typical UV coverage for our simulations and the corresponding dirty beam. Note that the dirty beam has several sidelobes but of smaller amplitude owing to the relatively filled UV.

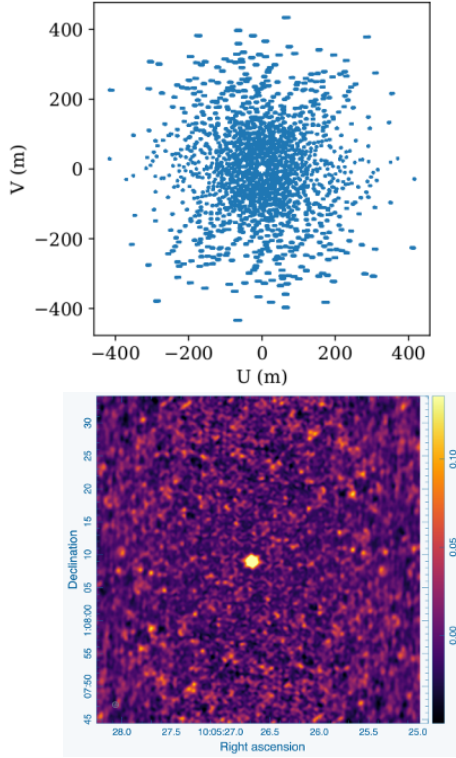


Fig. A.1: Top panel: An example of the total UV coverage for one of our ALMA 12-m array simulation pointing lasting 20 mins with 240 channels. Bottom panel: Shows the corresponding dirty beam.

Appendix B: The LESTA computing cluster and simulation details

The simulations presented in this work are produced on the LESTA computing cluster hosted at the University of Geneva Department of Astronomy. The details of the used resources:

- 16 Intel Xeon (E5-2640 v3, 2.6GHz) with 128GB (2133 MHz) memory (per node)
 - number of nodes: 32;
 - number of cores: 512
- 32 Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with 256GB (2666 MHz) memory (per node):
 - number of nodes: 8;
 - number of cores: 256

Each simulation/pointing is run in parallel using slurm job arrays on both resources, depending on availability. Each simulation takes approximately 12 minutes.

Appendix C: Sub-sampling

It is important to note that networks are quite demanding for input data stability such as, the data size, the data dynamic range,

etc. In the considered antennas configuration there are 50 antennas. That gives 1225 measurements. Taking into account Earth movement during the integration time, source positions and the following gridding the final number of measured frequencies (i.e., dimensionality of the input data) increases and is variable from about 1300 until about 1500. To solve the issue with the variable size of the input data we chose 1400 most frequently measured uv-positions. To do it, at first, we took at random 100 gridded data and converted them to the binary maps, where un-measured frequencies were set to 0 and the measured ones to 1, disregarding the real intensity. Then we summed up these 100 binary maps and chose 1400 positions with the highest values sorted in the decreasing order. That, in its turn, leads to the final binary map with 1400 values equal to 1 and the others equal to 0. Then every gridded data was multiplied by this final binary map. It is clear that in the cases when the gridded data contains more than 1400 measured frequencies it leads to the loss of information. But, we sacrifice it in order to have the fixed size input data for the network.

Appendix D: Proposed framework training

The python implementation of the proposed framework is publicly available at https://github.com/taran0/ml-based_source_localization_from_uv-plane.

For the training of the proposed framework, the data set was split into the train (80%), validation (10%) and test subsets (10%). The training was done 3 times under 3 different seeds.

Stage 1

Table D.1: Architecture of the g_θ model for $\bar{y} \in \mathbb{R}^{2 \times K}$ represented by the real and imaginary components.

Size	Layer	Param #
2×1400	Input	
2×262144	Dense, ReLU	367263744
512×512	Reshape	
512×512	Weighted sum, Tanh	524288

Table D.2: Architecture of the g_θ model for $\bar{y} \in \mathbb{R}^{2 \times K}$ represented by the magnitude and phase components.

Size	Layer	Param #
2×1400	Input	
512×512	Weighted mult.	2800
2×262144	Dense, ReLU	367263744
512×512	Reshape	

Training of the proposed framework was performed on the NVIDIA Titan RTX GPU with 24GB memory during 1000 epochs (the training time is about 3 hours) with the learning rate $1e-4$ and the batch size of 312.

Stage 2 training was performed on the NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory during 500 epochs (the training time is about 8 hours) with the learning rate $1e-3$ and the batch size of 32.

Table D.3: Architecture of the q_ϕ model.

Size	Layer	Param #
$512 \times 512 \times 1$	Input	
$256 \times 256 \times 8$	Conv2D, ReLU	80
$256 \times 256 \times 8$	BN	32
$128 \times 128 \times 16$	Conv2D, ReLU	1168
$256 \times 256 \times 16$	BN	64
$256 \times 256 \times 8$	Conv2DTran, ReLU	1160
$256 \times 256 \times 8$	BN	32
$512 \times 512 \times 1$	Conv2DTran, Sigmoid	73
$512 \times 512 \times 1$	BN	4

Post-binarization: we do not apply the fixed threshold t to binarize the DNN outputs $\hat{\mathbf{m}}^{N \times M}$. Instead, the binarization threshold t is chosen on the fly for each particular $\hat{\mathbf{m}}$. From the prior knowledge, we know that there are no more than 5 sources in each simulation. To determine the optimal threshold t , the proposed method performs an iterative thresholding until the number of detected sources p , both true and false, is not bigger than 15. This value was chosen empirically based on the trade-off between the corresponding purity and completeness as shown in Fig. D.1.

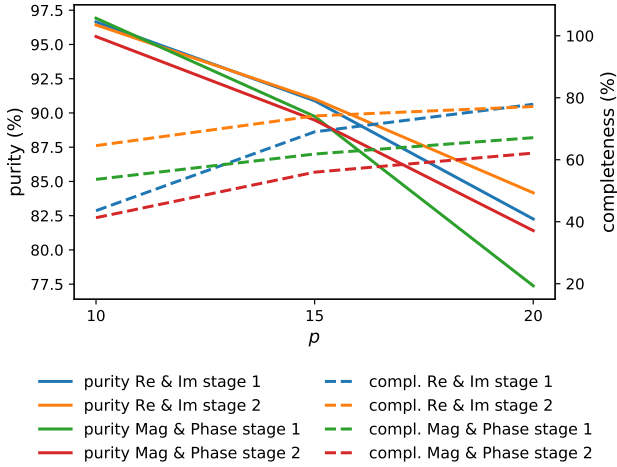


Fig. D.1: Dependence of the purity and completeness on the limit on the number of detected sources p .

Appendix E: State-of-the-art source detectors

E.1. Noiseless data

Table E.1: Performance (in %) of the state-of-the-art source detectors on the CLEAN images in the noiseless case.

	PyBDSF	AEGEAN	PySE
Purity	94.21 (± 0.55)	94.41 (± 0.45)	94.23 (± 0.8)
Completeness	93.07 (± 0.11)	93.62 (± 0.13)	91.49 (± 0.25)

The comparison of the performance of the state-of-the-art PyBDSF, AEGEAN (Hancock et al. 2012, 2018) and PySE (Carbone et al. 2018) methods on our data set is given in Table E.1. The parameters of the detectors were chosen to provide the purity of about 94 %. It is easy to see that the purity and completeness obtained for PyBDSF and AEGEAN are very similar. Under the same purity PySE has slightly smaller completeness.

E.2. Noisy data

Table E.2: Performance (in %) of the state-of-the-art source detectors on the CLEAN images in the noisy case, averaged over all SNRs.

	PyBDSF	AEGEAN	PySE
Purity	90.60 (± 1.06)	48.49 (± 1.16)	65.73 (± 1.56)
Completeness	25.48 (± 0.42)	25.57 (± 0.60)	25.64 (± 0.29)

Similarly to the noiseless case, we compare the performance of the state-of-the-art detectors PyBDSF, AEGEAN and PySE. The obtained purity and completeness are given in Table E.2. The parameters of the detectors were chosen to provide the maximum purity under the assumption that the completeness should be no less than 25 %. In contrast to the noiseless case, in case of noisy data under the same completeness PySE outperforms AEGEAN in terms of the purity. AEGEAN detector has almost two times smaller purity compared to PyBDSF. PyBDSF demonstrates the best performance.