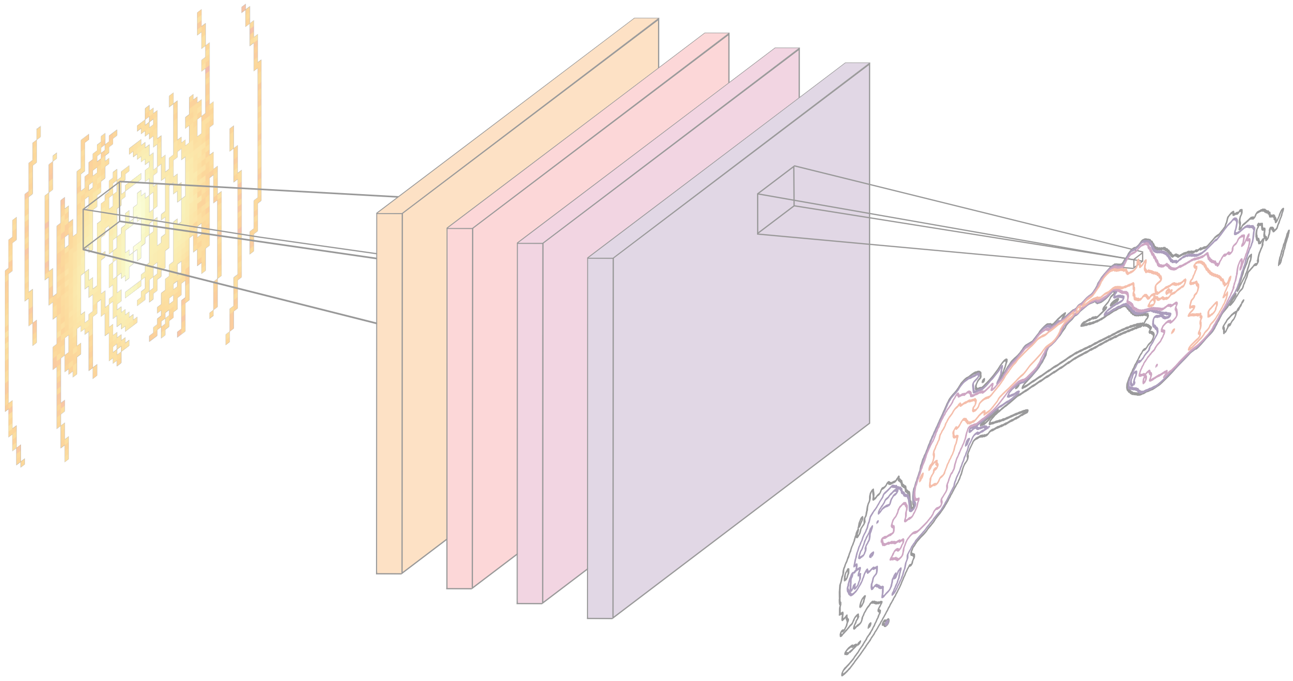
technische universität
dortmund

Another One Cleans the Dust

A Deep Learning-based Interpretation of an All-time Classic

Kevin Schmidt

2022



A document submitted in partial fulfillment of the requirements for the degree of
Doctor rerum naturalium
at
Faculty of Physics, TU Dortmund University

Supervised by
Prof. Dr. Dr. Wolfgang Rhode and Prof. Dr. Dominik J. Bomans

This thesis is set in Latin Modern and Fira Code,
typeset using LaTeX with LuaTeX from TeX-Live 2022.
Source contours on the title page are taken from Ramatsoku et al. (2020) [99].

For Ela,
you will always be part of my universe.

Abstract

Radio interferometers achieve the highest resolutions at the cost of sparse data coverage. Incompletely sampled sky distributions in Fourier space result in noise artifacts in the source reconstructions. Established cleaning software is often time-consuming and lacks reproducibility.

In this work, I propose a novel cleaning strategy for radio interferometer data based on convolutional neural networks to adjust current analysis strategies to the new telescope standards. This deep learning-based approach will allow a straightforward application that generates reproducible results with short reconstruction times.

The newly developed simulation chain enables the simulation of Gaussian radio galaxies and mimics observations by radio interferometers. By iterative adjustments, complexity is increased, ending up with a simulated data set comparable to MOJAVE archive data. In parallel, the deep learning framework radionets, capable of uncertainty estimates, is built to analyze large data samples with comparable characteristics. The improved reconstruction technique will allow scientists to focus more on their scientific analysis and omit a vast workload on data cleaning tasks.

Various evaluation techniques are created to quantify the trained deep learning models' reconstruction quality. Furthermore, the reconstruction performance is assessed on input data with different noise levels by comparing the resulting predictions with the simulated source distributions. Source orientations and sizes are well reproduced, while the recovered intensities show substantial scatter, albeit not worse than existing methods without fine-tuning.

Finally, all improvements are combined to train a deep learning model suitable to evaluate MOJAVE observations.

Zusammenfassung

Radiointerferometer erreichen die höchsten Auflösungen unter dem Preis einer geringen Datenabdeckung. Unvollständig gemessene Himmelsverteilungen im Fourier-Raum führen zu Rauschartefakten in den Quellenrekonstruktionen. Etablierte Bereinigungssoftware ist oft zeitaufwändig und liefert schlecht reproduzierbare Ergebnisse.

In dieser Arbeit schlage ich eine neuartige Bereinigungsstrategie für Radiointerferometerdaten vor, die auf „Convolutional Neural Network“ basiert, mit dem Ziel aktuelle Analysestrategien an die neuen Teleskopstandards anzupassen. Dieser auf „Deep Learning“ basierende Ansatz ermöglicht eine unkomplizierte Anwendung, die reproduzierbare Ergebnisse bei kurzen Anwendezeiten erzeugt.

Die neu entwickelte Simulationskette ermöglicht die Simulation von Gaußschen Radiogalaxien und ahmt Beobachtungen von Radiointerferometern nach. Durch iterative Anpassungen wird die Komplexität erhöht, so dass am Ende ein simulierter Datensatz vorliegt, der mit MOJAVE-Archivdaten vergleichbar ist. Parallel dazu wird das „Deep-Learning-Framework“ radionets entwickelt. Neben der Fähigkeit große Datenmengen mit vergleichbaren Eigenschaften zu analysieren, ermöglicht der entwickelte Ansatz auch Unsicherheitsabschätzungen für die Rekonstruktionen anzugeben. Die verbesserte Rekonstruktionstechnik führt zu Zeitersparnissen, da ein Großteil des Arbeitsaufwandes für die Datenbereinigung vermieden wird. Dadurch bleibt mehr Zeit für wissenschaftliche Analyse der Radiodaten.

Mit Hilfe verschiedener Bewertungskriterien wird die Rekonstruktionsfähigkeit der trainierten „Deep-Learning-Modelle“ quantifiziert. Die Rekonstruktionsqualität wird anhand von Daten mit unterschiedlichen Rauschpegeln bewertet, indem die resultierenden Vorhersagen mit den simulierten Quellenverteilungen verglichen werden. Quellenausrichtungen und -größen werden gut rekonstruiert, während die geschätzten Intensitäten erhebliche Streuungen aufweisen, wenn auch nicht schlechter als bei bestehenden Methoden ohne Feinabstimmung der Bereinigungsparameter. Abschließend werden alle Erkenntnisse kombiniert um ein „Deep-Learning-Modell“ zu trainieren, das zur Auswertung von MOJAVE-Beobachtungen geeignet ist.

Contents

1 Radio Interferometry in the Context of Modern Astronomy

"So much universe, and so little time."
— Terry Pratchett

The observable universe is enormous. There are no comparisons that can reasonably illustrate its size. As humans, we think on an everyday scale. We have a good idea of the distance between our house and the supermarket. The understanding of distance significantly decreases when we explain how far our home is from the Very Long Baseline Array (VLBA) [83] in North America. For Europeans, it is far away. In terms of the universe, far away means something completely different. With modern telescopes, we can observe objects at distances of kiloparsec in the Milky Way, our home galaxy [104]. We study galaxies light-years away [1], the clusters in which these galaxies interact [47, 90], and even the remnants of the Big Bang [69, 95] in which the universe was formed.

Nowadays, digitization is also taking place in observatories. State-of-the-art radio telescopes like the Low Frequency Array (LOFAR) measure terabytes of data during one observation [40], and the next generation of telescopes is expected to increase the data rates to petabytes, as planned for the Square Kilometre Array (SKA) [24, 35]. In conjunction with maximizing observing time, the astronomer's community has to adjust current analysis strategies to the new telescope standards. Otherwise, we will have too little time to get the most out of the observations.

I propose a novel cleaning strategy for radio interferometer data based on convolutional neural networks in this work. This deep learning-based approach will allow a straightforward application without lengthy parameter fine-tuning and extensive user input. Therefore, the approach will be more accessible to junior scientists and even usable for astronomers from other subfields. Applying machine learning techniques in astronomy and astrophysics becomes more common with increasing computing power. For years, machine learning models have been used to improve analysis results, and scientific output in gamma-ray and neutrino astronomy [132, 31]. Recently, more publications have arisen where deep learning techniques are applied in all astronomy fields [53, 71]. Convolutional neural networks are known for their rapid image data analysis, which makes them superior to old-fashioned and time-intensive approaches. First attempts of application to data from radio interferometers have already been successfully tried by Morningstar et al. [79, 80]. The following analysis will further investigate the prospects of bringing deep learning to radio interferometry. I will start with simple simulations and tests followed by iterative adjustments, increasing the complexity. The goal is to create a fast and easily applicable tool capable of analyzing large data samples with comparable quality and observation settings. These conditions are valid for sky survey data [6, 114] or data from a regular monitoring program like the Monitoring Of Jets in Active galactic nuclei with VLBA Experiments (MOJAVE) program [66]. The improved reconstruction techniques will allow scientists to focus more on their scientific analysis and omit a vast workload on data cleaning tasks. Ideas and results presented in this thesis have been published in Astronomy and Astrophysics [106].

Section 1.1, Section 1.2, and Section 1.3 gives an overview of the opportunities of modern astronomy, the frequently observed targets, and an introduction to the concepts of radio interfer-

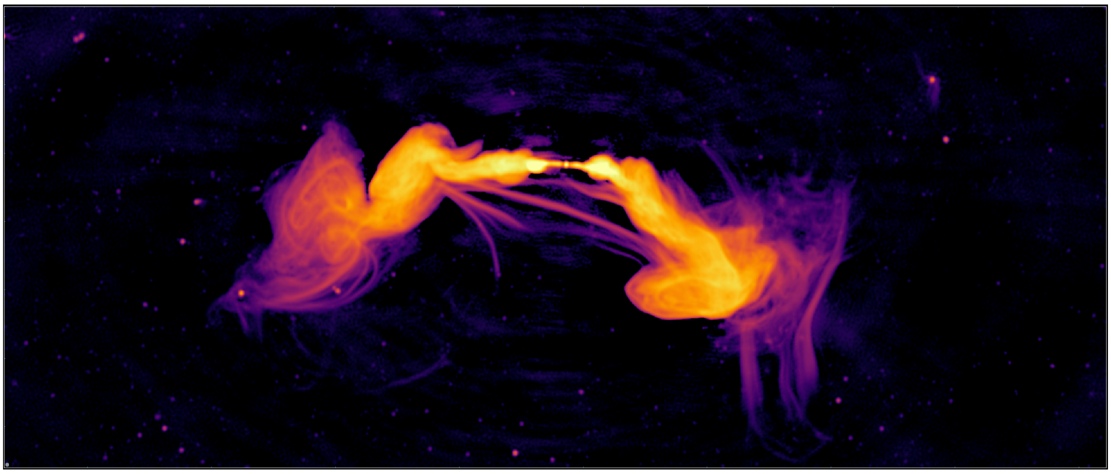


Figure 1.1: MeerKAT observation of the galaxy ESO 137-006 showing collimated synchrotron threads linking the radio lobes. The high resolution achieved by utilizing radio interferometry allows a detailed analysis of the source morphology. The image is taken from Ramatsoku et al. (2020) [99].

ometry, respectively. Additionally, observations using the VLBA are explained in Section 1.4, and standard processing steps for radio interferometer data are introduced in Section 1.5. After establishing the physical foundation, Chapter 2 introduces basic deep learning concepts focusing on convolutional neural networks. Afterward, the developed simulation chain is illustrated in Chapter 3. Besides radio galaxy simulations, the mimicking of radio interferometer observations with the help of sampling masks is explained. Various noise influences are added iteratively to create more realistic simulations. Furthermore, simulations using the Radio Interferometer Measurement Equation (RIME) [116] are introduced, creating a data set for comparing the deep learning approach with standard software. While Chapter 4 introduces the developed deep learning model, including the architecture and loss function, a first assessment of the reconstruction results is given in Chapter 5. Chapter 6 presents different techniques to evaluate the trained deep learning models on dedicated test sets. Additionally, a comparison with w-Stacking Clean (WSCLEAN) [89] is performed, and execution times are investigated. Advanced analysis approaches presented in Chapter 7 lead the way to apply the models to actual observation data. Model improvements that allow for uncertainty estimation and improved source simulations are being tested. Subsequently, Chapter 8 describes a simulation chain to train a deep learning model designed to reconstruct MOJAVE observations. In this chapter, the knowledge gained from the previous chapters is combined, probing the reconstruction of actual observational data. Finally, concluding remarks and an outlook are given in Chapter 9.

1.1 Opportunities of Modern Astronomy

In modern astronomy, sources are not only observed in a single wavelength but simultaneously in multiple ones [64, 33]. These wavelengths range from radio, optical, and X-ray to gamma-ray and cover most parts of the electromagnetic spectrum. While radio and optical telescopes often operate ground-based, direct detection of X-rays and gamma rays is performed utilizing satellites [18]. Ground-based detection of gamma rays is possible with imaging air Cherenkov telescopes like the Major Atmospheric Gamma-Ray Imaging Cherenkov (MAGIC) telescopes [3] and the Cherenkov Telescope Array (CTA) [21]. Additionally, the IceCube Neutrino Observatory (IceCube) [2] provides information about astrophysical neutrinos. Combining measurements from electromagnetic observatories with those from astroparticle experiments opened the field of multi-messenger astronomy, which played a decisive role in the recent astronomical achievements [52].

Radio astronomy, especially radio interferometry, plays a unique role as it acquires the highest resolutions. By combining measurements from different telescopes through cross-correlation, radio interferometry enables imaging of astrophysical objects with milliarcsecond resolution at the cost of sparse data coverages [72]. Figure 1.1 visualizes an observation of the Karoo Array Telescope in the Meerkat National Park (MeerKAT) [55], which is a pathfinder experiment for the upcoming SKA [24, 35]. Here, detailed structures of the source morphology are visible, and excellent signal-to-noise ratios are achieved. In this example, radio interferometry enables the study of the collimated synchrotron threads linking the radio lobes of the source ESO 137-006 [99]. These detailed images help study astrophysical sources' morphology, magnetic fields, and interaction with the surrounding matter. Such measurements contribute to studying the acceleration mechanisms of extragalactic sources like Active Galactic Nuclei (AGN) [7], which will be explained in the following Section 1.2. Including radio data is essential in multi-messenger analyses; therefore, a straightforward analysis that astronomers of other subfields can apply is crucial to push the opportunities of modern astronomy.

1.2 Observation Targets

The radio waves that can be measured on Earth have different origins. The most famous radio signal is the neutral hydrogen line (HI line), also known as 21 cm line, whose origin is the emission caused by the spin-flip of neutral hydrogen [82]. As hydrogen is the most common element in the universe [5], its emission is observable from nearby sources but also distant galaxies. Nonetheless, there are many extragalactic radio sources, for example, AGN, whose radio emission is caused by another mechanism.

AGN are large particle accelerators with a central supermassive black hole as their engine, which form large outflows, so-called jets [7]. The black hole accretes matter from the host galaxy forming a hot accretion disk around the inner region surrounded by gas and dust. AGNs are not only among the brightest objects in the universe, but they also emit in the entire electromagnetic spectrum. This complete coverage becomes evident when analyzing their Spectral Energy Distribution (SED) [56, 97] The core and the jet are the most prominent emission regions in radio wavelengths, expressed in the classification scheme by Fanaroff and Riley. While Fanaroff-Riley Class I (FR I) galaxies show bright cores but fainter jet structures, the opposite is true for Fanaroff-Riley Class II (FR II) galaxies [27]. More subclasses of AGN have been established depending on the jet's orientation regarding the observer's line of sight following the grant unification model [127, 85]. Effects where the sources appear with one- or two-sided jets are additionally explainable with the jet's orientation. Sources pointing close into the observer's direction have a strong relativistic boosting in that direction, which causes very-high emissions. At the same time, the counter jet's emission is boosted away from the observer, resulting in an apparently one-sided jet structure [14].

The common explanation for the radio emission of AGN is synchrotron radiation of high-energy electrons. Electrons are accelerated along the jet structure inside the high magnetic fields of AGN. When these electrons reach relativistic speeds and get distracted inside the magnetic field, they emit electromagnetic synchrotron radiation perpendicular to their velocity [25, 8]. Jet structures remain stable for long periods on kiloparsec scales. Models developed by Blandford and Königl [9, 60] describe these extended structures as narrow conical jets with continuous plasma inflows. Shock fronts marked by higher luminosities form along the jet axis [117], which can be identified as individual components. As electrons are accelerated in all parts of an AGN, the morphology of the complete source is visible in radio wavelengths, which becomes apparent when looking at the radio galaxy shown in Figure 1.1.

1.3 Concepts of Radio Interferometry

When using telescopes, the limiting factor is often the resolution of the aperture. The resolution limit is given by the Rayleigh criterion [68], which is defined as

$$\mathit{\Theta} \approx 1.22 \cdot \frac{\lambda}{D}. \tag{1.1}$$

The angular resolution $\mathit{\Theta}$ depends on the observation wavelengths λ and the telescopes' diameter D. In the case of single-dish observations, an excellent effective collection area is achieved, as the dish reflects the incoming radiation onto the receiver. This effective area results in high sensitivities and high signal-to-noise ratios. Technically the dish size is limited to diameters around 100 m, which restricts the maximum resolution. The invention of radio interferometry enabled us to aim for higher resolutions again, especially the idea of Very Long Baseline Interferometry (VLBI) [16]. Several individual radio telescopes are thereby combined, creating one extended instrument.

The concepts of radio interferometry are known as complex and unintuitive, but adopting the descriptions by Smirnov [116], and Noordam [86] using the RIME gives straightforward access. Their descriptions and additional explanations in Thompson et al. [122] form the basis for the following section. Vectors are marked by $\vec{v}$, while bold symbols $\mathbf{V}$ indicate matrices.

The main assumptions in radio interferometry are quasi-monochromatic signals and linearity. In general, an electromagnetic signal of a single point source can be described by a complex vector $\vec{e}$ in a (x, y, z) coordinate frame, with

$$\vec{e} = \begin{pmatrix} e_\mathrm{x} \\ e_\mathrm{y} \end{pmatrix} \tag{1.2}$$

Here, z is pointing in the source direction. On its way toward the telescope and before leaving the receiver, the signal is affected by several linear effects, describable by a collection of Jones matrices $\mathbf{J}$ [41]. When the signal hits a usual two-feed antenna, the signal is converted into complex voltages $\vec{v}$,

$$\vec{v} = \begin{pmatrix} v_\mathrm{a} \\ v_\mathrm{b} \end{pmatrix} = \mathbf{J}\vec{e}. \tag{1.3}$$

Consequently, two antennas p, q measure the voltages $\vec{v}_\mathrm{p}$ and $\vec{v}_\mathrm{q}$. Signals of a radio interferometer are fed into a correlator, generating four pairwise correlations in the case of $\vec{v}_\mathrm{p}$ and $\vec{v}_\mathrm{q}$. At this point, it is possible to define the visibility matrix $\mathbf{V}_\mathrm{pq}$,

$$\mathbf{V}_\mathrm{pq} = 2 \begin{pmatrix} < v_\mathrm{pa} v^*_\mathrm{qa} > & < v_\mathrm{pa} v^*_\mathrm{qb} > \\ < v_\mathrm{pb} v^*_\mathrm{qa} > & < v_\mathrm{pb} v^*_\mathrm{qb} > \end{pmatrix} = 2 < \vec{v}_\mathrm{p} \vec{v}^H_\mathrm{q} >, \tag{1.4}$$

where $< . >$ denotes the averaging over time and frequency, v^* marks the complex conjugate of v, and $\vec{v}^H$ stands for conjugate transpose. The introduction of the brightness matrix $\mathbf{B}$, which is defined as

$$\mathbf{B} = 2 \begin{pmatrix} < e_\mathrm{x} e^*_\mathrm{x} > & < e_\mathrm{x} e^*_\mathrm{y} > \\ < e_\mathrm{y} e^*_\mathrm{x} > & < e_\mathrm{y} e^*_\mathrm{y} > \end{pmatrix} = \begin{pmatrix} I + Q & U + iV \\ U - iV & I - Q \end{pmatrix}, \tag{1.5}$$

already allows the formulation of the basic RIME:

$$\mathbf{V}_\mathrm{pq} = 2 < \mathbf{J}_\mathrm{p} (\vec{e}\,\vec{e}^H) \mathbf{J}^H_\mathrm{q} >= \mathbf{J}_\mathrm{p} \mathbf{B} \mathbf{J}^H_\mathrm{q}. \tag{1.6}$$

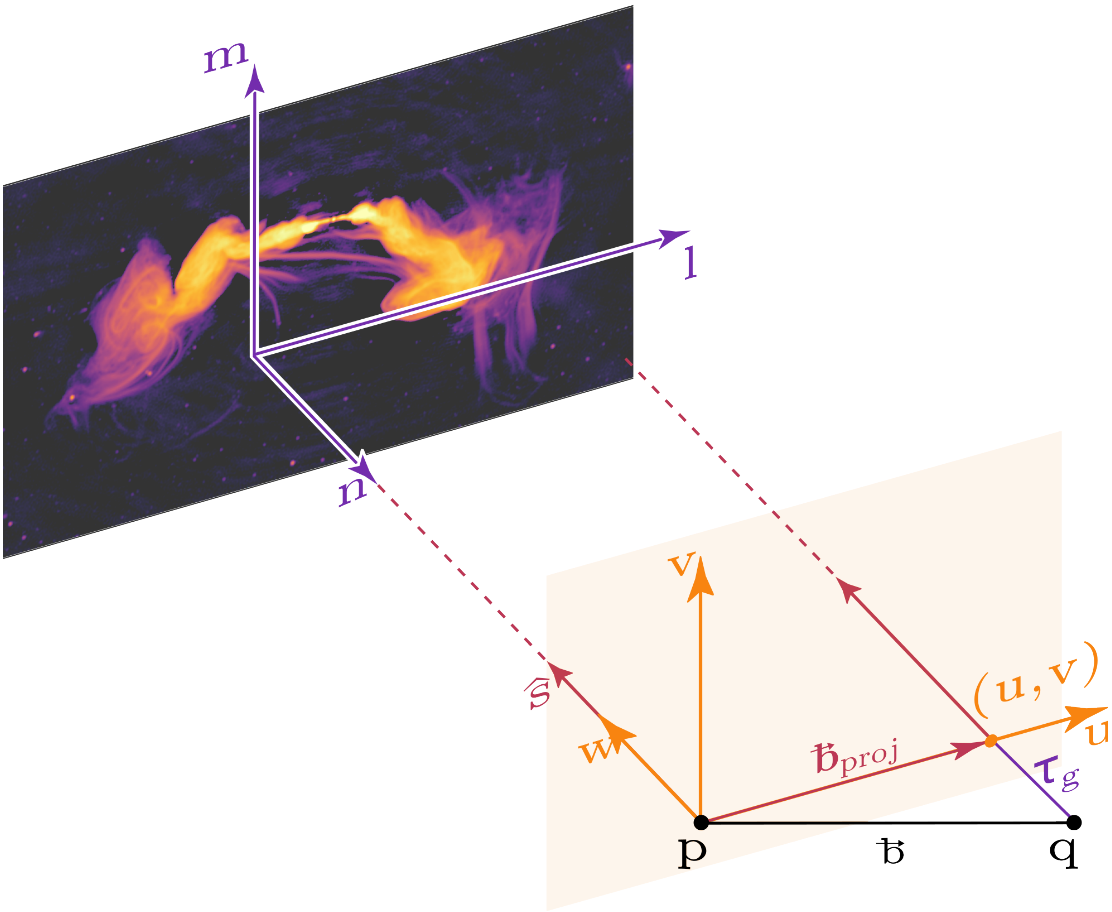


Figure 1.2: Coordinate relations for a radio interferometer observation explained in the example of a two-element interferometer. A sky with the brightness distribution $\mathbf{B}(\sigma)$ is observed by antennas p and q. As the source is very far away, the radio waves reaching the interferometer are quasi-monochromatic. The interferometer layout causes phase delays for the approaching signal by propagating different geometric path lengths. This geometric delay is described by τ_{g}. Correcting this delay creates projected baselines expressed in (u, v, w) coordinates. Here, w points in the observed direction, and (u, v) spans a tangential plane toward this direction. Because of the large distance, the sky is assumed as a plane described in (l, m, n) coordinates, where n points into the observer's direction. Both, (u, v, w) and (l, m, n) are direction cosines. Schematic after [102]. The source image is taken from Ramatsoku et al. (2020) [99].

$\mathbf{B}$ expresses the emission of the observed sky, which is commonly related to the Stokes parameters I, Q, U, and V [12].

The connection between measured visibility and the sky brightness becomes more accessible with the example of the two-element interferometer, illustrated in Figure 1.2. Here, the telescopes p and q are separated by the baseline $\vec{b}$. Both observe the same part of the sky described by the brightness distribution $\mathbf{B}(\sigma)$. Due to the enormous distance between telescopes and sources, the sky is assumed as a plane represented by (l, m, n) coordinates, where n points into the observer's direction. Electromagnetic waves emitted by the source reach the telescopes as quasi-monochromatic signals. The interferometer's layout causes phase delays for the approaching signal as the signal has to propagate different geometric path lengths to reach the antennas.

τ_g expresses this geometric delay. Correcting this delay so that the signal is detected by both antennas simultaneously creates projected baselines described by (u, v, w) coordinates. For this system, w points into the source direction, representing the same axis as n, but with an opposite sign. The new coordinate basis represents antenna positions in units of u and v onto the so-called (u, v) plane. Both (l, m, n) and (u, v, w) coordinates are direction cosines [123].

The phase delay associated with signal propagation even occurs in an entirely uncorrupted observation. For that reason, it has to be considered in every RIME formulation. Radio interferometers are only sensitive to the phase difference between $\vec{v}_\mathrm{p}$ and $\vec{v}_\mathrm{q}$. Phase delays are measured regarding a specific phase center, in this example, regarding $(u, v) = (0, 0)$. Taking the coordinates of antenna p as $\vec{u}_\mathrm{p} = (u_\mathrm{p}, v_\mathrm{p}, w_\mathrm{p})$, the phase difference for u_p relative to the phase center for a signal from $\vec{\sigma}$ can be expressed as

$$k_\mathrm{p} = 2\pi(u_\mathrm{p}l + v_\mathrm{p}m + w_\mathrm{p}n - 1). \tag{1.7}$$

In this expression, $(l, m, n = \sqrt{1 - l^2 - m^2})$ are direction cosines of σ, and u, v, and w are given in units of λ. Equation (1.7) shows that the geometric phase delay is another scalar effect on the source's signal. Again, this linear transformation is expressable as a Jones matrix

$$\mathbf{K}_\mathrm{p} = \exp(-ik_\mathrm{p}) = \exp(-2\pi i(u_\mathrm{p}l + v_\mathrm{p}m + w_\mathrm{p}n - 1)), \tag{1.8}$$

leading to the uncorrupted RIME

$$\mathbf{V}_\mathrm{pq} = \mathbf{K}_\mathrm{p}\ \mathbf{B}\ \mathbf{K}_\mathrm{q}^H. \tag{1.9}$$

Only a single point source was assumed in equation (1.6). As a realistic sky consists of many sources, it has to be extended to

$$\mathbf{V}_\mathrm{pq} = \sum_\mathrm{s} \mathbf{J}_\mathrm{sp}\ \mathbf{B}_\mathrm{s}\ \mathbf{J}_\mathrm{sq}^H, \tag{1.10}$$

summing over all point sources s. The contributions to $\mathbf{V}_\mathrm{pq}$ add up linearly. Now, some effects are the same for all sources, so-called source-independent effects $\mathbf{G}_\mathrm{p}$. Consequently, there are source-dependent effects $\mathbf{E}_\mathrm{sp}$, splitting up the Jones chain into $\mathbf{J}_\mathrm{sp} = \mathbf{G}_\mathrm{p}\ \mathbf{E}_\mathrm{sp}\ \mathbf{K}_\mathrm{sp}$, changing the visibility matrix to

$$\mathbf{V}_\mathrm{pq} = \mathbf{G}_\mathrm{p} \left(\sum_\mathrm{s} \mathbf{E}_\mathrm{sp}\ \mathbf{K}_\mathrm{sp}\ \mathbf{B}_\mathrm{s}\ \mathbf{K}_\mathrm{sq}^H\ \mathbf{E}_\mathrm{sq}^H \right) \mathbf{G}_\mathrm{q}^H. \tag{1.11}$$

At this point, the RIME is only a few assumptions away from the van Cittert-Zernike theorem [124], which is the fundamental theorem of radio interferometry. First, the sky has a continuous brightness distribution $\mathbf{B}(\sigma)$; the integration over all directions is necessary. Furthermore, a sine projection of the sphere onto the (l, m) plane tangential to the field center is performed. The non-coplanarity term with the n dependence is put into the source dependent effects $\mathbf{E}(l, m)$. The resulting RIME has the form of a two-dimensional Fourier transform:

$$\mathbf{V}_\mathrm{pq} = \mathbf{G}_\mathrm{p} \left(\int_l \int_m \mathbf{E}_\mathrm{p}(l, m)\ \mathbf{B}(l, m)\ \mathbf{E}_\mathrm{q}^H(l, m)\ \exp\left(-2\pi i(u_\mathrm{pq}l + v_\mathrm{pq}m)\right) \mathrm{d}l\mathrm{d}m \right) \mathbf{G}_\mathrm{q}^H. \tag{1.12}$$

Equation (1.12) means that the interferometer measures the two-dimensional Fourier transformation of the sky, but the sky is seen differently for every baseline. This dependency between the observed sky and the obtained signal is the main statement of the van Cittert-Zernike theorem [124].

1.4 Observations with the VLBA

The relation derived in the previous Section 1.3 is evident when illustrated with a vivid example using an actual interferometer layout. Figure 1.3 visualizes the power pattern of three antenna

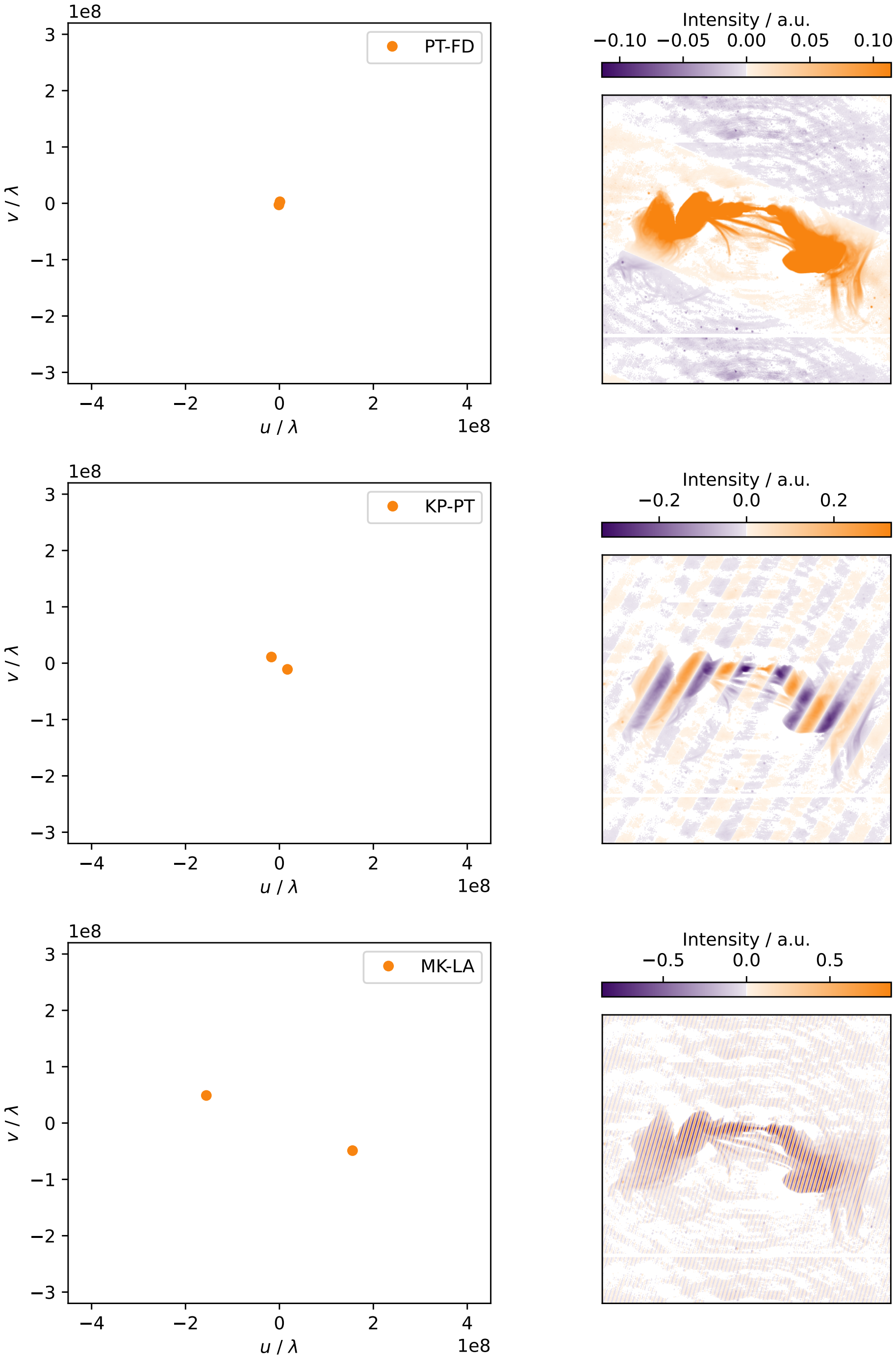


Figure 1.3: Power pattern of the baselines Pie Town–Fort Davis, Kitt Peak–Pie Town, and Mauna Kea–Los Alamos for a RIME simulated observation with the VLBA. Baseline lengths are visualized in (u, v) space (left column). The corresponding power pattern (right column) illustrates the sensitivity of the baseline for a specific spatial distance. The source image is taken from [99].

pairs of a simulated VLBA observation. Here, the RIME was utilized for the calculations. The left column illustrates the baseline lengths in (u, v) space. The corresponding power pattern

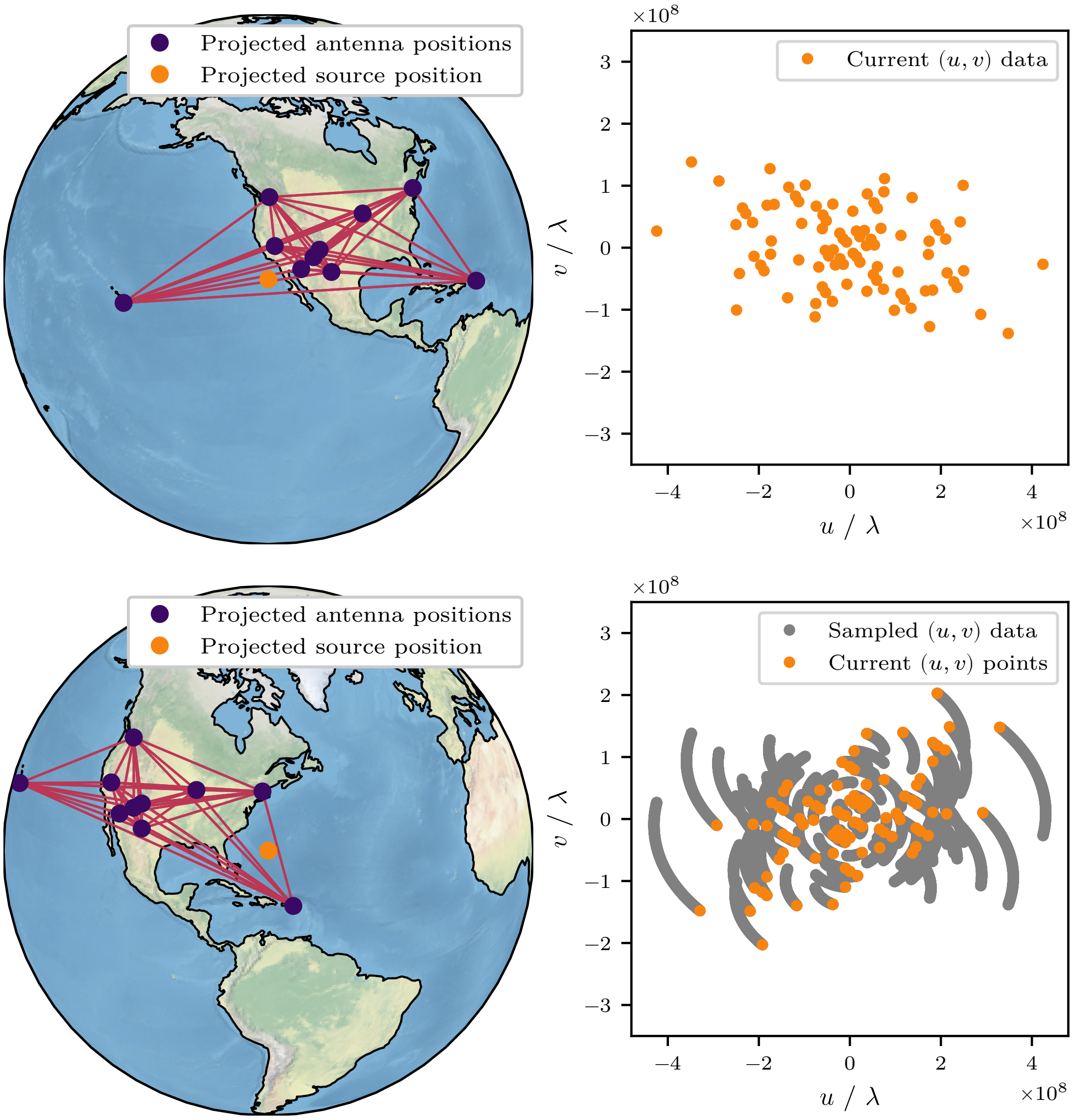


Figure 1.4: (u, v) coverage during a simulated VLBA observationat 15.21 GHz. While the projected antenna positions of the VLBA are shown in purple, the projected source position is marked in orange inside the Earth overviews (left column). The corresponding (u, v) coverages are illustrated inside the (u, v) plane (right column). The upper row represents the start of the observation. Here, the (u, v) coverage includes only 90 visibilities. With ongoing observation time, the Earth rotates, and the projection changes for every time step, filling the (u, v) plane. The lower row displays the end of the observation, where the (u, v) coverage increased significantly. Still, broad areas with missing information are visible due to VLBA's sparse antenna layout.

shows for which spatial distances the baseline is sensitive (right column). Small baselines measure the large-scale structures of the sky's brightness distribution. At the same time, the largest baselines measure the fine-scale structures. As the power pattern changes signs in a regular frequency, it is called the fringe pattern. These fringe patterns are different for every baseline. Maximizing the number of different-lengthed baselines is essential to collect as much information about the source as possible.

In this work, the VLBA [83] was chosen as the interferometer layout to perform the simulations for two reasons. First, it consists only of ten antennas, which makes it handy and accelerates calculations. Per snapshot, its 45 baselines create 90 visibilities. Second, the VLBA is used for the MOJAVE program [66], which has an extensive data archive suitable for deducing source shapes and other observation parameters. The numerous existing measurements are exploited to develop new analysis strategies and test the current simulation chain. Developed analysis setups are later easily adjustable for other telescopes like LOFAR [40], which consists of thousands of baselines per snapshot.

A disadvantage is the VLBA's small detection area. On the one hand, it achieves milliarcsecond resolution. On the other hand, it only obtains information at ten antenna positions. Therefore, the exceptionally high resolution comes with sparse data coverage.

The Earth's rotation is exploited for actual observations to increase the data coverage. Figure 1.4 illustrates the projected antenna positions at the start and the end of a simulated observation together with the (u, v) coverage. For each timestep, the projection is calculated, and the corresponding visibilities are evaluated. The projection changes through the Earth's rotation, and so do the projected baselines. As arcs start to form in (u, v) space, the coverage of the simulated observation increases.

1.5 Processing of Radio Interferometer Data

Complex values measured by the radio interferometer have to be further processed before they can be used to generate scientific output.

First, a calibration of the data is necessary. Calibration operations correct receiver signals and compensate for different antenna gains of the individual telescopes. Observing a well-known calibrator source that generates calibration solutions to correct the target field is common. Such calibration corrections were applied to the MOJAVE data used in Section 8.6. No calibration is performed for the simulated observations, as in the simulation chain, no corruption effects requiring a complete calibration were considered. Still, calibration is an important topic, which will be included in subsequent works. Detailed information about the different calibration approaches can be found in [101, 94, 100].

Second, it is necessary to put the irregularly distributed visibility data on a regular grid before applying the Fast Fourier Transform (FFT). Especially for images with sizes of 2^{n}, the FFT is superior to the Discrete Fourier Transform (DFT) in computing costs and time [87]. Creating this regularly sampled data is called gridding [15]. The analyst has to choose a reasonable grid and frequel size which fits the observation parameters. A frequel in (u, v) space is the equivalent of a pixel that suits to describe source distributions in image space. Different gridder implementations are available, for example, in WSCLEAN [126] or Difference Mapping (DIFMAP) [113]. These implementations use convolutional resampling [15], which was avoided for the analyses in this thesis for complexity reasons. The developed implementation inside the `pyvisgen` package [30] performs a two-dimensional binning for a fixed bin size and grid size.

Last, cleaning of the gridded data has to be performed. In this final processing step, the remnants of the telescope's Point Spread Function (PSF), also called beam in radio interferometry, have to be removed from the sky image [125]. The gridded visibility data's direct FFT creates a dirty image dominated by noise artifacts. This image corruption originates from incomplete sampling during the observation. A common approach to get an uncorrupted source image is the `CLEAN` algorithm [110]. `CLEAN` performs an iterative point source subtraction, creating a source model convolved with the observation's clean beam. The clean beam represents the innermost part of the interferometer PSF. Over the last 50 years, different adjustments and improvements to match new standards were made. Cleaning techniques started with Högbom's implementation [48] going over Clark's [20] and Schwab's [109], reaching solutions suitable for wide-field arrays as developed by Cornwell [22] and Offringa et al. [89].

This work proposes an alternative deep learning-based imaging approach, which will be described in Chapter 4 and Chapter 5.

An Introduction to Deep Learning 2

> *"He'd have been a little bit happier if there'd been a demon or some sort of magic. Something simple and understandable."*
>
> — Terry Pratchett, *Soul Music*

Depending on the person asked, people still see deep learning models as black boxes that magically solve problems. However, there is no demon sitting inside the graphical unit of the computer, and even though some results look like it, there is no magic involved. Instead, deep learning is a mixture of precise simulations, linear algebra, and the minimization of a pre-defined criterion. In the modern world, many successful applications are visible in all fields, for example, DeepL for text translation [23], or object detection for autonomous driving [81]. Steadily, the acceptance of deep learning applications is increasing in science. There is still immense potential to improve current analysis strategies to maximize the scientific results with the help of deep learning techniques. Therefore, this work explores opportunities for utilization in radio interferometry. For wider acceptance, it is essential to make deep learning approaches more understandable and clear up existing prejudices. Through cooperations such as the Sonderforschungsbereich 876 (SFB 876) [77], the educational work of the Kaggle community [57], and workshops organized by the ErUM-Data-Hub [26], deep learning becomes more understandable and usable for the whole scientific community.

2.1 General Concepts

Deep learning algorithms are data-based algorithms designated to learn a mapping of the input data x to the output y. The mapping function $f(x;\Theta)$ describes the relationship between input and output. Here, Θ are weighting parameters learned by the deep learning model. A pre-defined measure allows for quantifying the mapping and adjusting the learned parameters based on the mapping quality. This work applies supervised learning, meaning a set of observations with known ground truths is utilized. For every example in the input data, a target is specified. Subsequently, a loss function is used to evaluate the deviation between the model's prediction and the target. The goal is the minimization of this loss function. In deep learning, this minimization is mainly performed by applying stochastic gradient decent-like algorithms [13]. The smaller the loss value, the better the prediction and the target match. Small loss values are desired for the whole data set so that the model generalizes the learned mapping and no overtraining occurs. Overtraining is when the model learns the training data set by heart and thus fails on unseen data. Different regularization techniques help to prevent overtraining by adding penalty terms to the loss function or adding more variation to the training data set. [34]

First, the scientific question has to be defined. For a robust model creation, an exact problem definition is crucial. Additionally, a reproducible and easy-to-understand approach is desirable. The "no free lunch" theorem states that no universal best machine learning algorithms are available for all possible distributions [131]. Generally, assumptions and specifications must be made to minimize the generalization error. Only then the trained deep learning model performs well on new unseen data. [78]

2.2 Neural Network Models

Neural networks are directed graphs, which consist of nodes organized in different layers. The input layer takes the input features. These features are propagated to the output layer, delivering the result. During the propagation, they pass several hidden layers, only accessed by the learning algorithm during the training. The hidden layers make the neural network model deep. Activation functions at the end of each layer enable the representation of non-linear relations. In this work, the Parameter Rectified Linear Unit (PReLU) function [46] is applied as non-linearity. The PReLU is a modified Rectified Linear Unit (ReLU) [54] with an additional learnable parameter defining the function's slope in the negative part:

$$\mathrm{PReLU}(x) = \begin{cases} x, & \text{if x} \geq 0 \\ ax, & \text{otherwise} \end{cases}, \tag{2.1}$$

which is visualized in Figure 2.1. Without activation functions, neural networks can only represent linear dependencies, as only linear matrix multiplication is applied.

Each layer of the network has weights, which are optimized during the training to achieve the optimal mapping between input and target. The optimization is done using backpropagation [103]. If the prediction does not fit the target, the error is backpropagated from the last layer until the first one is reached. The derivatives of the mapping function concerning all variables are calculated. Backpropagation is a fast technique as it exploits the chain rule. At the end of a backpropagation step, the model's weights are optimized regarding the calculated derivatives. Here, Stochastic Gradient Descent (SGD) is utilized to find the best possible parameters [44]. Nowadays, enhanced SGD methods are favored. Adam is a common alternative, as it uses momentum to achieve an improved model stability [59].

A series of layers, called architecture, is necessary to develop a well-performing deep learning algorithm. Different layers are available; convolutional layers are most commonly used in this work. Neural networks made from these layers are called Convolutional Neural Network (CNN). A convolutional layer consists of a kernel K that is usually smaller than the input data. The kernel is iteratively moved over the input I, generating the output S by convolving input values and kernel weights:

$$S(i,j) = (K * I)(i,j) = \sum_u \sum_v I(i-u, j-v) K(u,v). \tag{2.2}$$

The parameters are shared, as the kernel has the same weights for all input data. This sharing makes convolutional layers extraordinarily fast on image data. [34]

Furthermore, the convolution of two functions is well-suited for pattern recognition. In Chapter 4, this feature will be used to design an architecture capable of finding patterns in incomplete Fourier data. The adjustment of the architecture is essential to achieve the best-possible training outcome. Changing the kernel sizes allows for regulating what scales in the input image influence the calculated output. Adding more layers to the architecture increases the number of kernels and, therefore, the number of trainable parameters. Expanding the number of channels, which increases the depth of the convolution matrices, further enhances the number of trainable parameters. The number of parameters is a vital feature to fine-tune, as the complexity of the model usually depends on its depth [120]. Deeper networks have an increased representation capacity, allowing them to learn more complex relations. Simultaneously, they are harder to train, and more training data is needed for robust training. Here, residual networks improved the robustness significantly [45]. Often additional data augmentation methods help, such as slightly variating the input data so that the neural network does not simply learn the training data by heart. These aspects enable the creation of a deep learning model which neither

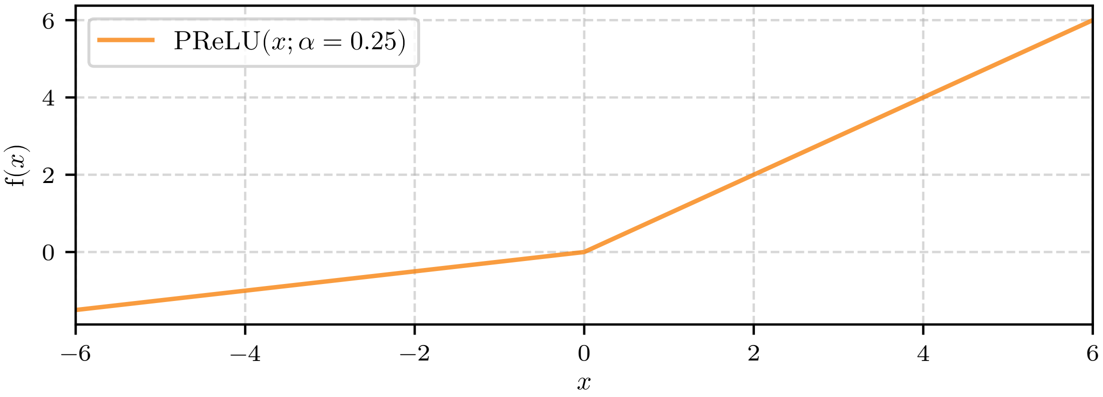


Figure 2.1: Visualization of the PReLU activation function. The learnable parameter α is set to 0.25.

underfits nor overfits the learning task. However, there is more to consider. Neural networks are often fragile. Small changes in the architecture or slightly different data sets can significantly impact the learning result. Furthermore, weights should not explode during the training process. Batchnorm layers help to keep everything in a small parameter range by normalizing incoming values using their mean and standard deviation [105]. One goal of making neural network models useable in science is to make them more robust and explainable [38]. [78]

2.3 Model Training

Training a neural network starts with precise data preparation. This step is often the most time-consuming in the whole training process. Especially when dealing with physics data, the problem must be well-defined to keep the learning process understandable. Another crucial aspect is data representation. The data set and format must be appropriate for the selected algorithm. Otherwise, the model is unable to learn the underlying problem.

Second, the model setup influences the training process substantially. A wise selection of architecture components speeds up the model's convergence. Sometimes wrong choices even prevent a convergence at all. The best fitting combination of activations, loss function, and minimizer for the problem has to be found. During the training, the data flows batch-wise through the architecture. After each batch, the loss function is evaluated, the derivatives for backpropagation are determined, and the weights get adjusted. Here, training batches are created randomly from the complete training data set. When an entire cycle of batches for the training data is completed, the validation data is processed batch-wise. For the validation data, only the loss is calculated, but no backpropagation is performed, as this data is meant to test the model's generalization grade. Furthermore, it is a measure of overtraining when training and validation loss do not decrease simultaneously. Lastly, independent test data sets help perform advanced evaluations after finishing the training. Learned lessons from these tests help to adjust the neural network model, retrain it and perform another evaluation round.

2.4 Software and Computing Setup

In this work, calculations, simulations, and deep learning tasks are done in `Python` [98]. `PyTorch` [92] has been chosen as the fundamental deep learning framework because of its flexibility and the ability to develop algorithms in `Python` directly. On top of `PyTorch`, the deep learning library fast.ai is utilized [50], which supplies high-level components to build customized deep learning algorithms quickly and efficiently. For simulations and data analysis, the Python packages NumPy [42], SciPy [128], Astropy [121, 96], Cartopy [88], scikit-image [129], and Pandas [91, 70] were used. The results were illustrated using the visualization package Matplotlib [51]. A complete overview of the used libraries and the developed open-source packages are listed on GitHub: radionets [107], radiosim [108], pyvisgen [30].

The simulations and the deep learning models were developed on a Ubuntu Linux 20.04 LTS desktop system with

- CPU: Intel® Core i7-8700K CPU @ 3.70GHz, 12 logical cores,
- GPU: NVIDIA GeForce GTX 1080 with 8 GB GPU memory,
- RAM: 64 GB.

For the RIME simulations, the high-Performance computing cluster Linux Cluster an der TU Dortmund (LiDO3) was used [1]. The deep learning models, which include uncertainty estimates, were trained on the Competence Center Machine Learning Rhine-Ruhr (ML2R) computing cluster [2] equipped with NVIDIA A100 graphics processing units.

[1] LiDO3: https://www.lido.tu-dortmund.de/cms/de/home/index.html
[2] ML2R: https://www.ml2r.de/en/about/

3 Basic Simulations in Radio Interferometry

> *"They were learning fast, or at least collecting data, which they considered to be the same as learning."*
> — Terry Pratchett, *Thief of Time*

In the age of machine learning, especially deep learning, data to train the models have become more critical than ever before. With the correct data, training is possible, and the models are meaningful. For this reason, data and learning are closely connected.

When training supervised machine learning models, data with known ground truths are necessary [43]. This work simulates a collection of data sets to achieve two goals. First, the complexity of the data is increased steadily, which allows an iteratively adjusting of the model. Second, besides the input data, the truths which are unknown when using measurement data can be simulated at the same time. This chapter summarizes the creation of a complete simulation chain, starting from generating radio galaxies and completing with mimicking observations of radio interferometers. In the beginning, everything is kept simple so that the single parts of the simulations are easy to follow. Simulations of radio galaxies follow the AGN theory introduced in Section 1.2, while radio interferometer simulations are related to the ones described in [106].

3.1 Extended Gaussian Radio Galaxies

The first approach simulating a data set of radio galaxies follows the description of AGN jets by Blandford and Königl [9, 60]. They note that active galactic nuclei host a narrow conical jet that is fueled by plasma inflows. Hotspots with increased intensity, called jet components, form inside these jet structures. Following analysis strategies from the MOJAVE team [66], using jet components for source representation is a common simplification for VLBI data [65, 67]. For this reason, Gaussian distributions form the basis of radio galaxies simulated in this chapter.

Figure 3.1 visualizes a radio galaxy built from two-dimensional Gaussian distributions. Several components placed next to each other mimic its jet structure. The brightest component in the center of the image represents the core of the active galactic nucleus. For a structured distribution, the centers of jet components get placed on a straight line emerging from the core and moving toward the image's edge. The number of components is randomly drawn between three and six, guaranteeing that the jets do not expand outside the image borders. The distance between individual components is 5 pixels and fixed for all simulated radio galaxies. In addition to one-sided jets, also two-sided jets are simulated. In the two-sided cases, jet components are mirrored at the core component leading to a total number of source components between four and thirteen. Jet and counter-jet components have the same intensity, but the intensity of different radio galaxies is varied. The core component is initialized for each source with a value between 10^{-3} and 10^{1}. These values are drawn from a uniform distribution. Afterward, the intensity of jet components is logarithmically reduced by the factor

$$A(c_\mathrm{n}) = \frac{A_\mathrm{core}}{\exp(c_\mathrm{n})}, \tag{3.1}$$

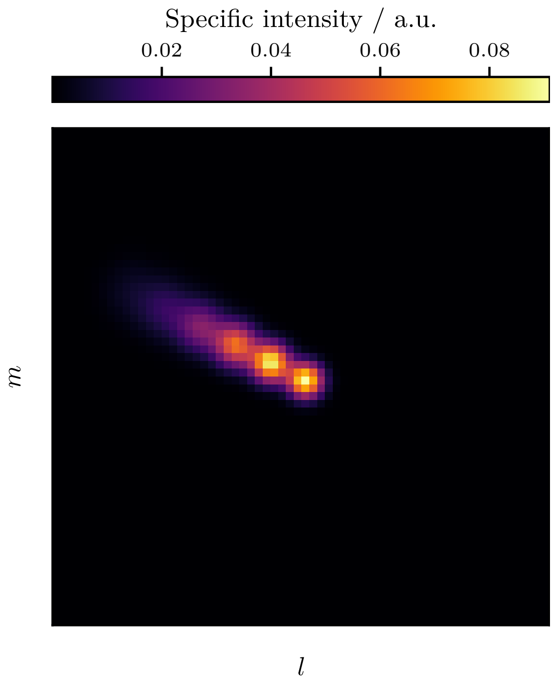


Figure 3.1: Source distribution of a simulated radio galaxy with a one-sided jet. The source is built from several Gaussian components.

where A is the peak intensity of the component and c_{n} denotes the number of the jet component. The components' standard deviations are initialized randomly and increase for components further outside the jet. Furthermore, jet orientations are varied by drawing their value uniformly between 0° and 360°. Appendix 2 summarizes all available simulation parameters and the default simulation configuration.

3.2 Radio Interferometer Simulations Using Sampling Masks

When observing a source with a radio interferometer, the analyzer of the data has to deal with different specific characteristics. As described in Section 1.3, a radio interferometer collects information in Fourier space. During an observation, they gather complex values called visibilities in (u, v) space. Depending on the array layout, the data coverage in (u, v) space varies but always remains sparse due to the interferometer's limited amount of antennas. In this thesis, the antenna layout of the VLBA is used, see Section 1.4, which results in 90 visibilities per snapshot during an observation. The utilization of the Earth's rotation introduced in Section 1.3 helps improve the data coverage by increasing observation time. Still, the characteristic holes in the (u, v) coverages remain.

The developed simulation chain utilizes the radio interferometer array layout and calculates the (u, v) coverage for an observation depending on source position and observation length. Simulated (u, v) coverages of single-channel observations form the basis for sampling masks that create incomplete visibility samples. Figure 3.2 visualizes the transformation of the simulated (u, v) coverage (upper left) to a binned sampling mask (upper right) by sorting the (u, v) points into a two-dimensional histogram. Afterward, these two-dimensional masks get applied to the Fourier transform of the simulated radio galaxies described in Section 3.1. As complex values are difficult to interpret by the human eye, amplitude and phase distributions are calculated with the help of Euler's formula [84]:

$$z = x + i \cdot y = |z| \cdot (\cos(\varphi) + i \cdot \sin(\varphi)) \,. \tag{3.2}$$

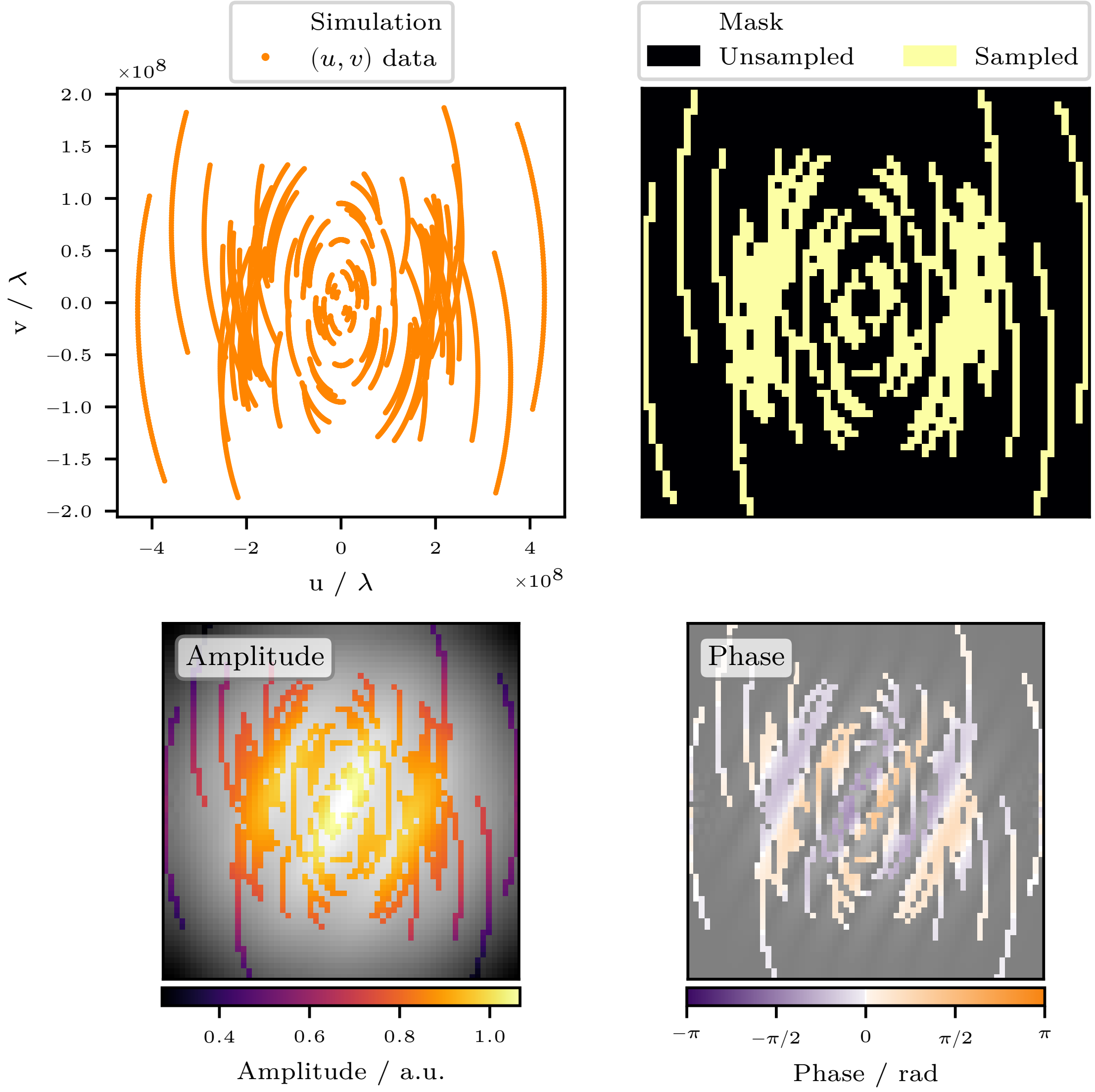


Figure 3.2: (u, v) coverage simulation based on VLBA's array layout. A simulated single-channel observation (upper left) serves as a basis to create a sampling mask by binning all (u, v) points into a two-dimensional histogram (upper right). When applied to the Fourier transform of a radio galaxy, an incomplete gridded visibility sample is generated. These incomplete visibility maps can be split into amplitude (lower left) and phase (lower right) maps utilizing Euler's formula. The remaining information is shown colored, while all dropped information is grayed out.

Here, x is the real part, and y is the imaginary part of the complex number z. Furthermore, $|z|$ represents the amplitude and φ the phase. Figure 3.2 illustrates the amplitude and phase distribution of the source shown in Figure 3.1 in the lower part. Information lost during sampling is grayed out and set to zero for the following data processing steps.

During the mask sampling, the information about physical scales gets lost. Therefore, distances inside the shown images do not refer to absolute angles, and all distances must be given in 64-pixel image size units. Furthermore, no instrumental effects like single antenna characteristics or receiver noise are considered. The resulting simulated data corresponds to already gridded (u, v) data. In (u, v) space, it is common to use the name frequel to address gridded data points.

A frequel is the equivalent of a pixel that suits to describe source distributions in image space.

Applying the inverse Fourier transformation to incomplete data samples creates a so-called ”dirty image”, see Section 1.3. In the case of these simulations, dirty images look like the one shown in Figure 3.3 (left). Artifacts dominate these images, and noisy structures cover source morphologies. Cleaning is mandatory beforehand to use these images for scientific analyses. Instead of conventional cleaning approaches, this thesis deals with training neural networks to reconstruct incomplete data before generating a source image in local space by applying the inverse Fourier transformation.

3.3 Noise Influences

Introducing uncorrelated noise effects at two stages inside the simulation chain allows to some extent, to consider noise influences. After the radio galaxy creation, newly drawn offsets from a standard normal distribution,

$$g(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{3.3}$$

are added to each pixel in the image space to mimic noise corruption from other sources in the universe. Offsets get scaled before being added by 5 % of the source’s peak flux, which results in noisy source distributions described by

$$I_{\text{noisy}}(l,m) = I(l,m) + 0.05 \cdot I_{\text{max}} \cdot g(x|\mu,\sigma)_{lm}. \tag{3.4}$$

$I(l,m)$ describes the source intensity in local space, I_{max} represents the peak intensity, and $g(x|\mu,\sigma)_{lm}$ denotes a random number drawn for the pixel at coordinate (l,m). When changing to (u,v) space via Fourier transform and performing the pixel-wise application of the sampling mask S to simulate an observation with a radio interferometer, this procedure results in corrupted visibilities

$$V_{\text{noisy}} = I_{\text{noisy}} \cdot S_{\text{mask}}. \tag{3.5}$$

Figure 3.3 visualizes the dirty image generated with noisy visibilities V_{noisy} in the middle.

The complexity of the noise simulation is further increased by adding additional white noise to the data in (u,v) space. For this purpose, random values are drawn again from a standard normal distribution. This procedure results in corrupted visibilities $V_{\text{white noise}}$, which can be described by

$$V_{\text{white noise}}(u,v) = V_{\text{noisy}}(u,v) + g(x|\mu,\sigma)_{uv}. \tag{3.6}$$

Again, $g(x|\mu,\sigma)_{uv}$ denotes a newly drawn number for every frequel. A dirty image generated with noisy visibilities plus additional white noise $V_{\text{white noise}}$ is shown in Figure 3.3 on the right. Further noise corruption leads to additional artifacts in the initial source reconstruction, which occur more randomly. Furthermore, the jet structure is hidden, and only the core component is visible.

A realistic description of the noise is necessary to analyze actual measurement data with neural networks. Often training data represents ideal conditions, and some noise influences are neglected or even unknown. For example, correlated noise is not considered in the simulations up to this step. The aim is to improve the simulations steadily. A possible solution is using the RIME formalism introduced in Section 1.3 to describe the noise influences occurring during an observation. While the RIME formalism is applied in a basic version to create a comparison data set in the next section, Chapter 8 will give a more detailed insight into the possibilities enabled by utilizing the RIME.

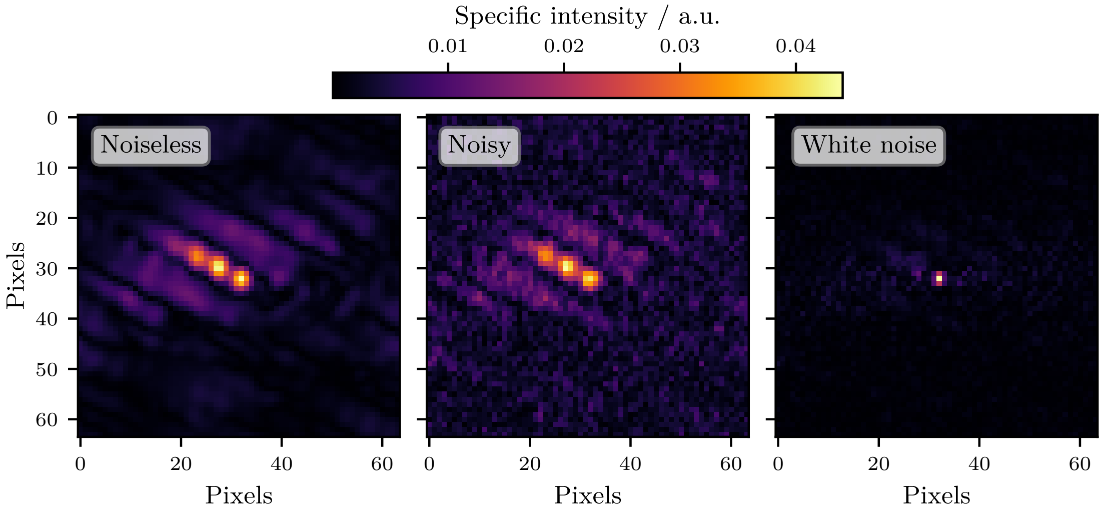


Figure 3.3: Dirty images are generated by applying the inverse Fourier transformation to the sampled amplitude and phase distribution for noiseless (left), noisy (middle), and white noise (right) maps. In addition to an underestimated flux density compared to the simulated source distribution shown in Figure 3.1, artifacts are visible next to the jet structures in all three cases. Only the core component is visible in the dirty image generated with white noise data. Here, artifacts are more random and distributed uniformly around the source position.

3.4 Radio Interferometer Simulations Using the RIME Formalism

In order to compare the reconstructions by deep learning models with images created by standard software, the second part of the simulation chain has to be adjusted. Established imaging software needs ungridded visibilities as input data, which cannot be achieved by simulation using sampling masks. However, a simple RIME formalism can calculate visibilities for each baseline. For this purpose, the identical simulated brightness distributions described in Section 3.1 are used as input for the RIME simulations. So that the data are comparable and the reconstruction of the visibilities is not too complicated, only the direction-independent phase delay term K is applied, which is defined as

$$\mathbf{K}(l, m) = e^{-2\pi i(ul+vm)}. \tag{3.7}$$

The phase delay term corresponds to the Fourier kernel and brings no further noise corruptions. It represents the connection between (l, m) coordinates in local space and (u, v) coordinates in visibility space. The formula for the calculation of the uncorrupted visibilities of the baseline pair pq results in

$$\mathbf{V}_{\mathrm{pq}} = \sum_{l} \sum_{m} \mathbf{K}_{\mathrm{p}}(l, m) \mathbf{B}(l, m) \mathbf{K}_{\mathrm{q}}^{H}(l, m), \tag{3.8}$$

with the source brightness distribution $B(l, m)$. The integrals introduced in equation (1.12) transform into sums as each pixel of the brightness distribution is treated as an individual source. The w-term is neglected as in VLBI observations, only a small section around a source is considered, and therefore, $\sqrt{1 - l^2 - m^2} \approx 1$ is valid in this case. u and v describe the coordinates of the respective baseline for the antenna pair pq at time t in direction cosines. Here, H denotes the conjugate transpose operation. The weighting of the visibilities is set to zero. A more detailed description of the RIME calculus is found in [116]. After simulation, the generated visibilities are stored in Flexible Image Transport System (FITS) format. For this purpose, a FITS-writer

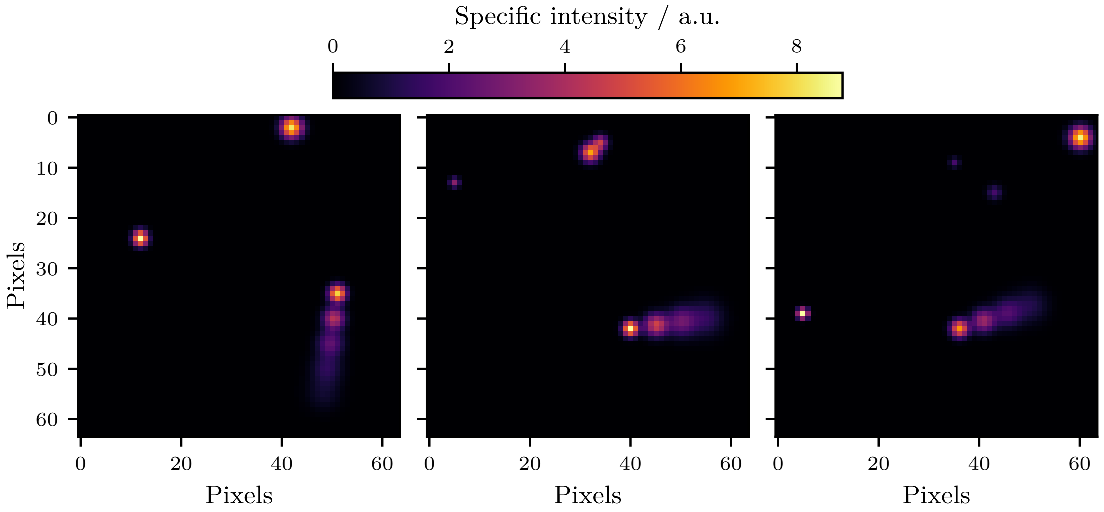


Figure 3.4: Exemplary images for the data set with mixed source types. Besides extended Gaussian radio galaxies, additional point-like Gaussian sources are simulated for the sky distributions.

was implemented according to the Astronomical Image Processing System (AIPS) memo 117 [36]. Both the implementation of the RIME simulation and the FITS writer are published in the open-source package `pyvisgen` [30].

3.5 Advanced Sky Simulations

On the path toward the analysis of observed data, tests with advanced radio sky simulations are necessary. For this reason, improvements were implemented to test the deep learning model's capability to reconstruct mixed source types, larger image sizes, and more complex source shapes.

Mixed Source Types

A new option was created to simulate sky images with additional point-like Gaussian sources besides the extended Gaussian radio galaxies. The point-like Gaussian sources differ in size and intensity. Their number per image is randomly chosen between one and six. Furthermore, the extended sources are not centered in the images anymore. The images are (64 × 64) pixels large. Therefore, they are comparable to sub-images of a more extensive sky survey. Figure 3.4 gives an overview of three different sub-image examples. Afterward, amplitude and phase maps are sampled following Section 3.2 to mimic radio interferometer observations.

Larger Image Sizes

The developed simulation chain allows for the straightforward creation of larger source images. Therefore, data sets consisting of image sizes closer to the sizes of actual radio interferometer images can be created. Figure 3.5 visualizes exemplary sources of a data set consisting of (128 × 128) pixel images. At this point, no linear scaling of the sources is performed so that the simulated radio galaxies appear more compact in these images. Amplitude and phase maps are generated following the procedure in Section 3.2.

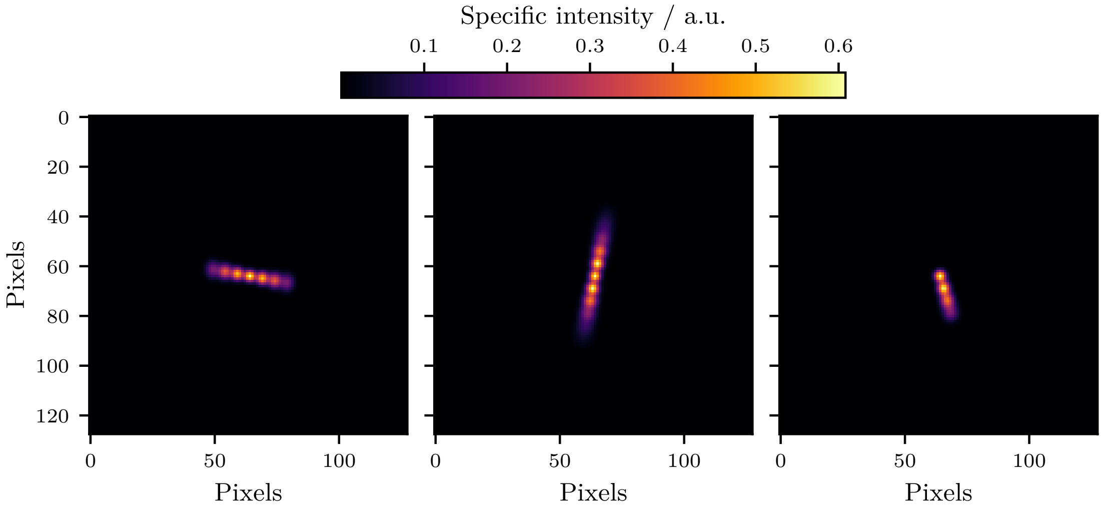


Figure 3.5: Exemplary source distributions for the data set with (128×128) pixel images. As the sources are not scaled with the increasing image size, they appear more compact.

New Source Shapes

In parallel, the creation of radio galaxies got a complete upgrade and was rebuilt inside the `radiosim` package [108], which I developed together with Arne Poggenpohl. Here, more complex source morphologies with bent jets and overlapping components are possible. Additionally, rotations in z-direction are taken into account. More detailed insights will be given in Section 8.2. Figure 3.6 gives an overview of the new source morphologies for two one-sided jets and one two-sided jet. Amplitude and phase maps are sampled following the descriptions in Section 3.2.

3.6 Data Sets

This chapter summarizes the different data sets created with the basic simulation routines. Table 3.1 gives an overview where some characteristics are quantified. The number of images, noise levels, and (u, v) coverages are summarized. All presented data sets consist of 50 k training images, 10 k validation images, and 10 k test images, except the larger image size data set. Here, the number of images is 5 k training images, 1 k validation images, and 1 k test images as this data set is dedicated to fine-tune an already existing pre-trained model. In the case of the simple RIME data set, only 1 k test images are generated. Furthermore, all data sets contain (64×64) pixel maps. The (u, v) coverages for the different data sets do not differ noticeably as the same sampling settings are used. The histogrammed sampling densities shown in Figure 3.7 explain these minimal deviations. An exception is the larger images data set, which consists of (128×128) pixel maps.

A re-scaling of the amplitude is performed utilizing the scaling rule

$$A_{\text{scaled}} = \left({}^{\log_{10}\left(A + 1 \cdot 10^{-10}\right)}/_{10} \right) + 1. \tag{3.9}$$

This scaling rule limits the amplitude to values between zero and two, leading to a smoother and faster model convergence. Zeros mark the information missing in the amplitude and phase distributions. Pixels with zero values have the advantage that they do not contribute to the result in the first convolution. Additionally, pixels with zeros have a large distance from the true

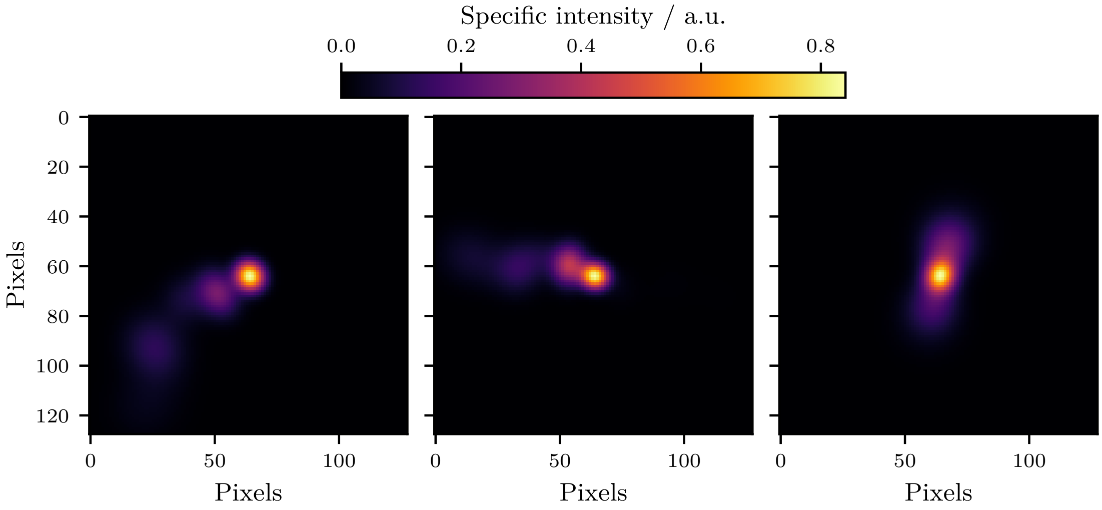


Figure 3.6: Exemplary images for the simulations with advanced source shapes. Bent jets and overlapping components allow the creation of more realistic radio galaxies.

value and contribute strongly to the loss. The network learns that the loss decreases quickly when suitable estimates are found for these pixels.

Table 3.1: Overview of the created deep learning data sets

Type	Number Images Train / Valid / Test	Noise Level / % Source	 Visibility	(u, v) coverage / %
Noiseless	50 k / 10 k / 10 k	–	–	25.65 ± 2.09
Noisy	50 k / 10 k / 10 k	5	–	25.65 ± 2.09
White noise	50 k / 10 k / 10 k	5	10	25.65 ± 2.08
Mixed	50 k / 10 k / 10 k	5	10	25.67 ± 2.11
New shapes	50 k / 10 k / 10 k	5	10	25.74 ± 2.16
Larger images	5 k / 1 k / 1 k	5	10	14.31 ± 1.45
`RIME` simple	– / – / 1 k	–	–	29.87 ± 13.38
MOJAVE	12 k / 1 k / 1 k	–	–	2.19 ± 0.39

Note: Each input map was generated with a different sampling mask but with a comparable sampling density. The noise level on sources corresponds to a Gaussian noise equivalent to 5 % of the sources' peak intensity, while noise on visibilities means Gaussian noise in the order of the lower 10 % of the noiseless visibilities.

The scaled amplitude and phase maps have different characteristics depending on the noise level. To give a better feeling and allow for a comparison, Figure 3.8 illustrates the amplitude and phase input maps for the noiseless (left), noisy (middle), and white noise (right) data set. The noiseless amplitude map (upper left) has its maximum in the central part except for the unsampled frequels. It continuously decreases towards the edges where the values are one-third of the central ones. The comparison with the noisy amplitude (upper center) reveals an increased overall amplitude. This effect is even more present for the white noise amplitude (upper right). Here, the decrease towards the edges is no longer present.

The noise corruption is visible in the phase input maps as well. While a uniformly striped pattern occurs for the noiseless map (lower left), this pattern is covered under spikes for the noisy

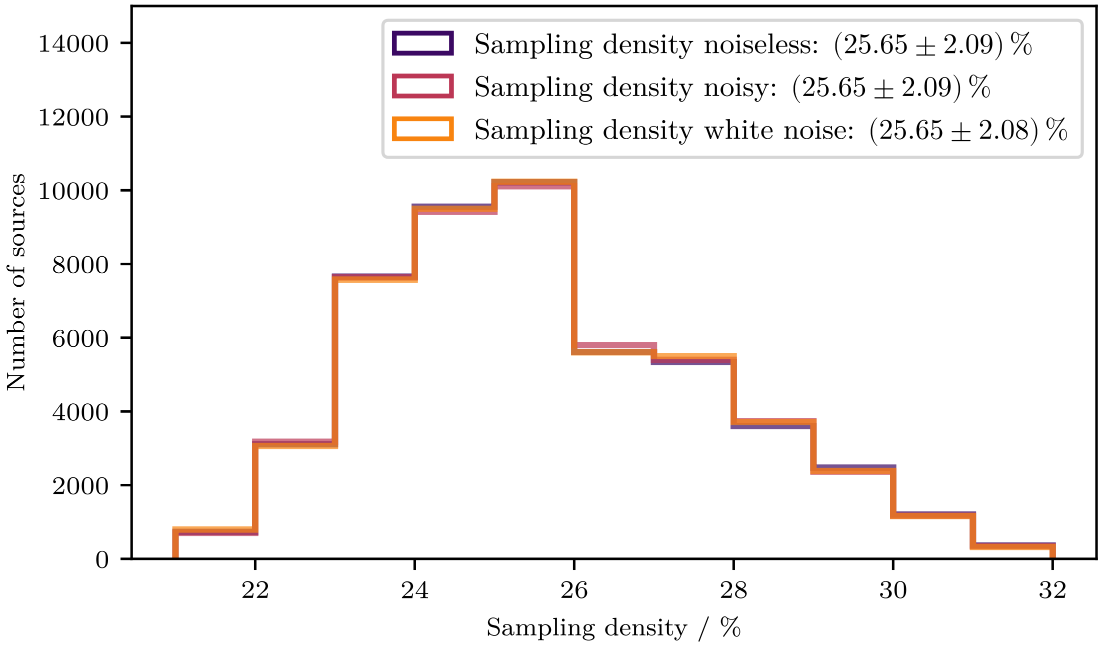


Figure 3.7: Histogrammed sampling densities for the three main deep-learning data sets. As the same sampling settings are applied, mean sampling density and standard deviation do not deviate noticeably.

phase (lower center). The white noise phase (lower right) is mainly shifted toward positive values except for some frequels in the central part. This shift is expected to increase the reconstruction complexity further.

Figure 3.9 visualizes visibility spaces for the advanced sky simulations introduced in Section 3.5. Examples are given for the mixed sources (left), new source shapes (middle), and larger image sizes (right) data sets split into amplitude (top) and phase (bottom) maps. Here, it becomes clear that amplitude and phase maps also become more intricate when the source distribution in the image space becomes more realistic. The point-like Gaussians introduced in the mixed sources data set break the point symmetry around the center, removing the flatness inside the phase maps. For the other two data sets, the pixel size of the input maps was doubled. Amplitude and phase maps for the larger sizes data set are comparable to the ones of the white noise data set. The new shapes data set combines the properties of the noisy and white noise amplitude and phase maps, showing flat amplitude distributions and spiky phase distributions. For this reason, the new shapes data set is expected to be the most difficult for the neural network to reconstruct.

Besides the data sets based on sampling masks, a visibility data set using a simple RIME formalism was simulated; see Section 3.4. This data set consists of 1000 FITS files, which include (u, v) information and associated complex values. For the reconstruction with the `radionets` framework and the analysis with WSCLEAN, further processing of the FITS files is necessary. Gridding of the single values is essential before working with the data. In the case of WSCLEAN, the gridder is included in the software. The gridder developed for the `pyvisgen` package is used before passing the RIME data to the deep learning model. This data set is not utilized for model training but only for testing existing models and comparing the deep learning approach with WSCLEAN. Figure 3.10 visualizes the gridded amplitude and phase map for the example source introduced in Figure 3.1. The field of view was set to 85 mas. More information about gridding and the influence of the chosen field of view can be found in Section 1.5.

The RIME simulation differs from the sampling mask simulations in the wide range of occurring

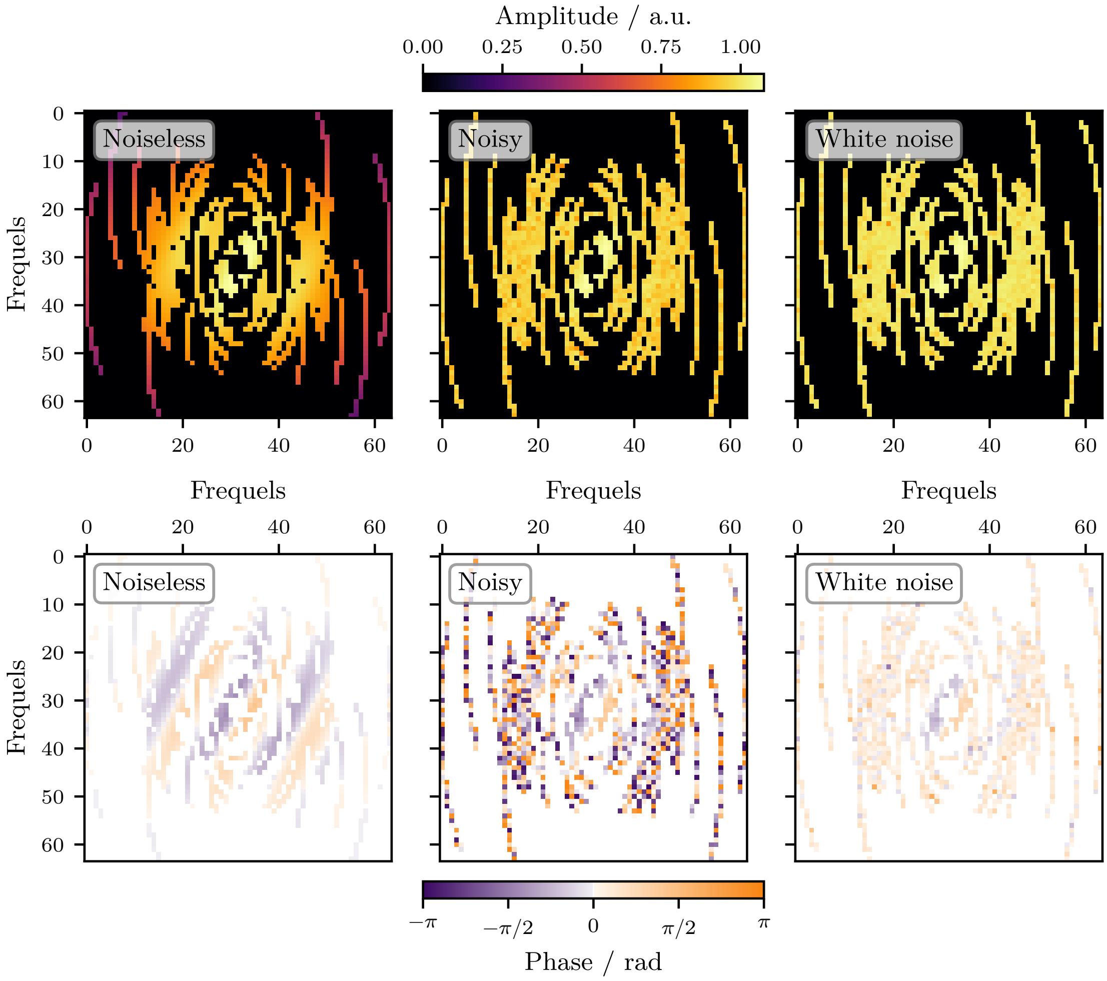


Figure 3.8: Comparison of the input maps for the noiseless (left), noisy (middle), and white noise (right) data set separated into amplitude (top) and phase (bottom). Visibilities are shown for the example source from Section 3.1. In all three cases, the same sampling map was applied.

(u, v) coverages. Individual antennas can now be excluded during a simulated observation, reducing this measurement's maximum resolution. These differences in resolution and considered antennas lead to a mean (u, v) coverage of $(29.87 \pm 13.38)\,\%$. In contrast to the RIME simulations, the sampling mask method simulates the best possible resolution for all observations.

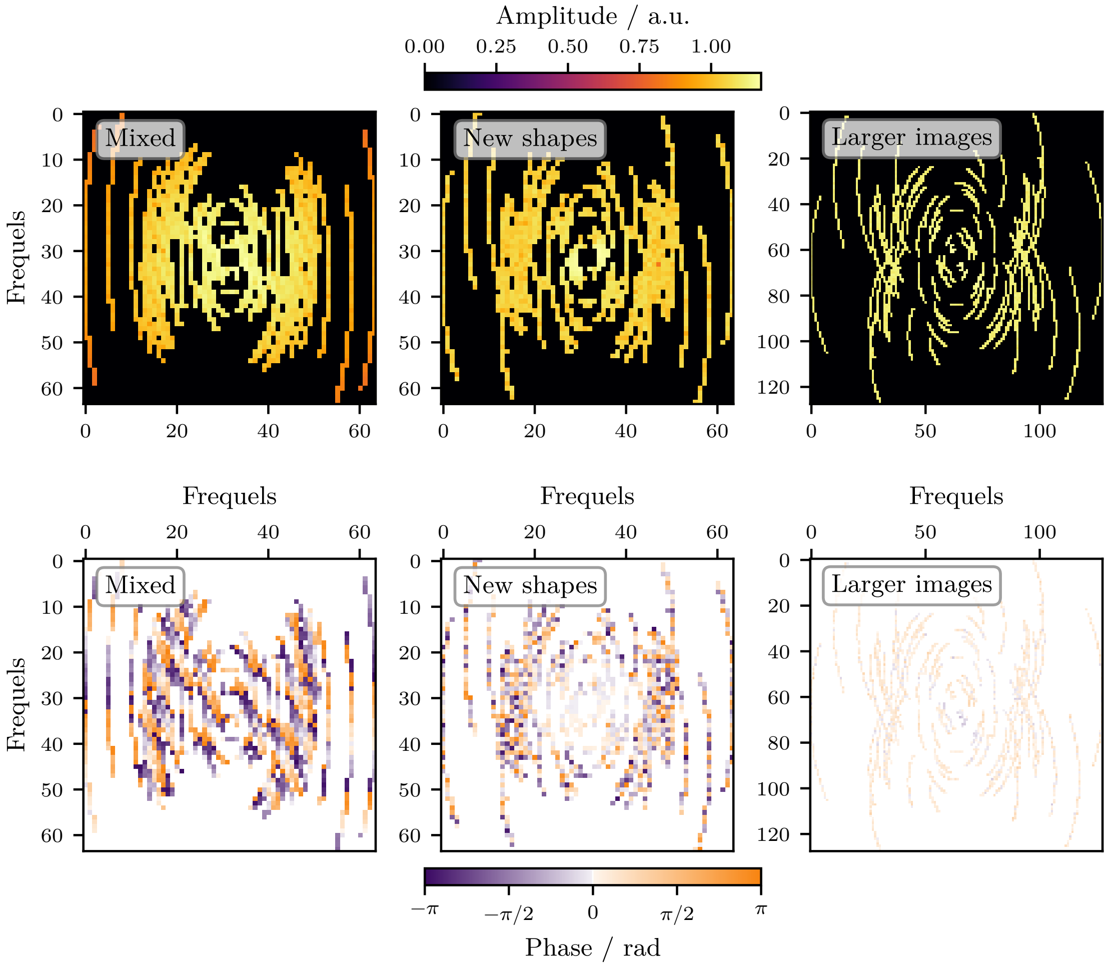


Figure 3.9: Comparison of the input maps for the mixed (left), new source shapes (middle), and larger images (right) data set separated into amplitude (top) and phase (bottom).

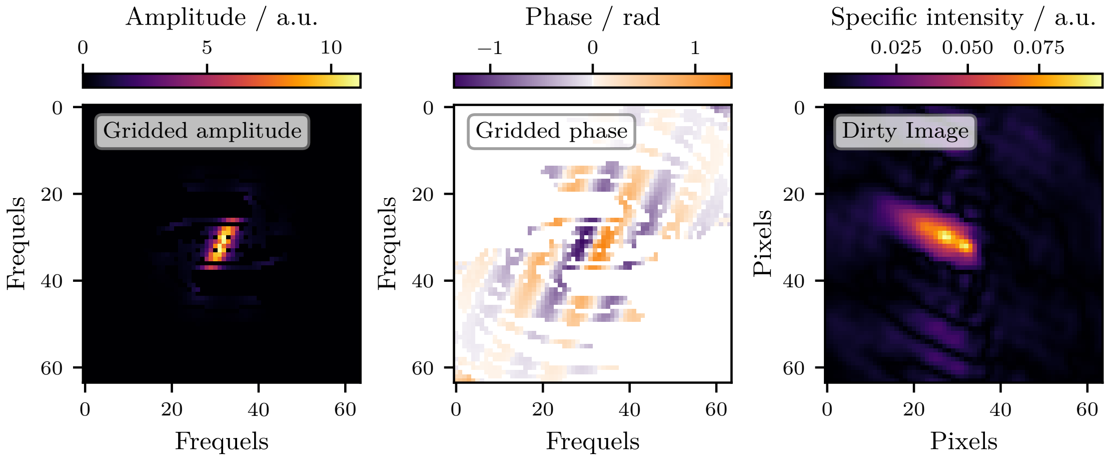


Figure 3.10: Amplitude (left) and phase (middle) map for the gridded simple RIME data set. The field of view is set to 85 mas. The dirty image (right) shows noise artifacts parallel to the jet and a blurry source structure.

4 Deep Learning Model

"I'm teaching you things all the time.
You might not be learning them, of course."
— Terry Pratchett, *Thief of Time*

When people hear about deep learning for the first time, they think of black boxes which magically solve machine-learning problems. Often the question arises, is the model learning to master the desired task, or does it deliver solutions that randomly fit this task? Of course, deep learning models are not the holy grail of computer science, but they are a feasible approach to simplify and speed up complex tasks in all science fields. With the training data at hand, the user can teach them every desired task. When the data set, the architecture, and the loss function are combined correctly, the deep learning model even starts to learn to solve this task.

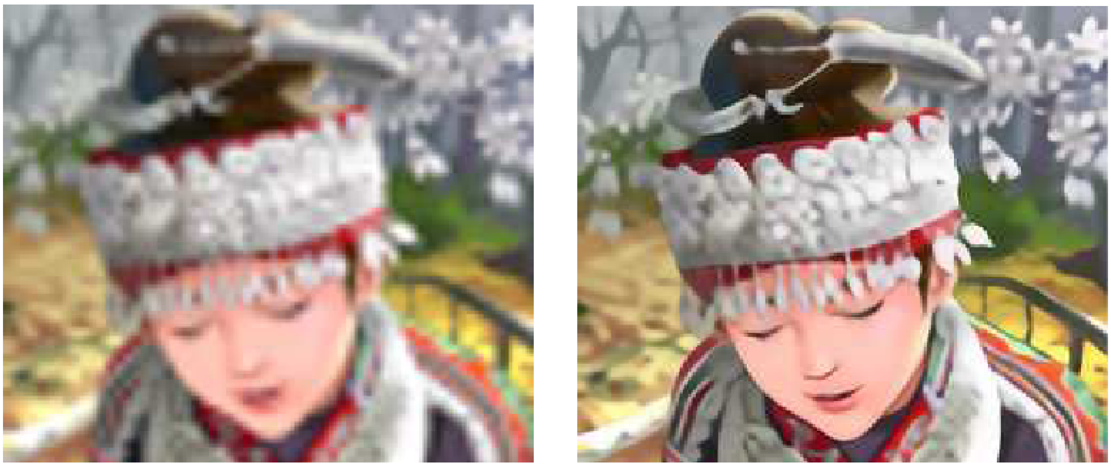
The first step to success is always a suitable problem definition. Deep learning courses [1] and literature about the current standards [63] help to find the appropriate architecture. The alternation between the evaluation of the results and hyperparameter adjustment ends up in a smoothly converging model which can reconstruct missing Fourier data. As stated before, this work contains knowledge gained from several theses I supervised [32, 29, 58, 11, 111, 62, 74, 17].

4.1 Problem Definition

The underlying problem is comparable to an image inpainting problem, starting with incomplete images because parts of them are noisy or damaged [39]. Another related problem is super-resolution imaging [28, 73]. Here, a low-resolution image forms the basis for creating an upsampled high-resolution version. The resulting image has a higher number of pixels, but the newly created pixels have no values at first. These pixels are called maximal corrupted pixels. One approach to fill them is to determine a suitable estimate by bicubic interpolation, as shown in Figure 4.1a [63]. The result is a larger version of the original image, but details within the image are now blurred. A deep learning-based approach to solve this problem is the application of Super Resolution Residual Networks (SRResNets) [63]. Here, convolutional layers and residual learning are utilized to find suitable estimates for the newly created pixels in the upsampling process. As seen in Figure 4.1b, details are still visible in the enlarged image version. The direct comparison to the nearest-neighbor approach reveals that the image is less blurred. At the same time, the style of the original image is maintained. In this work, a modified version of the SRResNet architecture was developed with Stefan Fröse, which can reconstruct missing information in radio interferometric measurements. The missing information in radio interferometer measurements is handled as an inpainting problem. Stefan has done many tests on this topic in his master thesis [29], and he had the initial idea for using this architecture.

In radio interferometry, measurements for specific parts of the Fourier space are known, but for many regions, no information is available. Still, existing data can be used to estimate frequels with unknown values. Notably, the measurements are also noisy and hence have no perfect

[1] fastai: https://course.fast.ai/



(a) Bicubic **(b)** SRResNet

Figure 4.1: Comparison of an upsampled image generated with bicubic interpolation and one upsampled using the SRResNet model. While the bicubic version looks blurry, the SRResNet version shows many details. Example taken from [63].

values, further complicating the problem. Frequels with measured values are slightly corrupted ones. The frequels without any information are maximal corrupted ones. In the case of radio interferometric data, upscaling of the images is not needed. Therefore, this part of the analysis is omitted entirely. The following chapters show how the SRResNet architecture was adopted to be applied to radio interferometric data.

4.2 Architecture

In this work, the SRResNet architecture from [63] is used, which consists of different stages: a `PreBlock`, several Residual Blocks (ResBlocks), a `PostBlock`, and a `FinalBlock`, compare Table 4.1. The ResBlocks, which are used after He et al. (2015) [45] and adapted by Gross and Wilber (2016) [37], form the core of the architecture. Before the ResBlocks, a `PreBlock` applies an initial convolution and activation to the input data. A `PostBlock` follows the ResBlocks. Afterward, the `FinalBlock` completes the architecture. The upsampling part of the original architecture is omitted as the image size does not need to be increased in the case of radio interferometry. A unique feature is the network training on data in Fourier space which means that all convolutions are applied to amplitude and phase distributions. In these distributions missing information is marked with zeros. The network's task is reconstructing these missing frequels. For a detailed overview of the training data sets, see Section 3.6. Figure 4.2 illustrates the architecture with its different stages schematically, as well as the input and the target data.

The settings of the convolutions are chosen so that the image size always remains the same. The model's predictions are the same size when input maps are (64×64) pixels. The sizes inside the model are controlled by the stride and padding settings. Table 4.1 summarizes all settings. As activation functions, PReLUs are used [46]. Their negative part is not constant at zero, but the network trains the negative coefficient. Only the first and the last convolution have a kernel size of (9×9) pixels. All others have kernel sizes of (3×3) pixels. The SRResNet architecture uses residual learning through different skip connections within the network. A skip connection transfers the input of one layer to another location in the network, where a pixel-wise summation is performed. The intermediate part only has to learn the difference between input and output. The mapping function F, which maps the input x to the output y, changes to $F(x) = y - x$. In

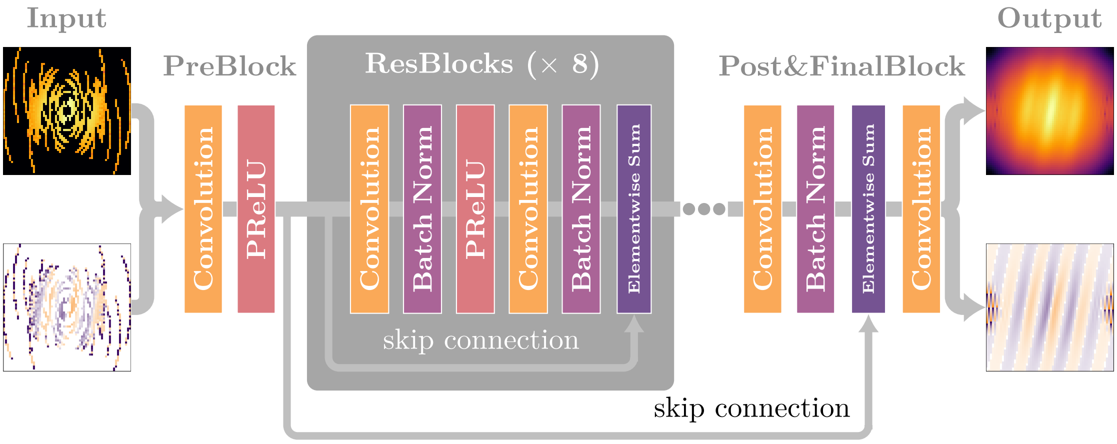


Figure 4.2: Overview of the developed architecture based on the SRResNet [45]. Sampled amplitude and phase maps serve as input (left). The different blocks process the input data and reconstruct missing information. The eight ResBlocks form the central part of the architecture. Skip connections enable residual learning and smoother convergence during training. After the `FinalBlock`, the reconstructed amplitude and phase map (right) are the output of the architecture.

Table 4.1: Detailed overview of the different blocks used in the developed architecture

Stage	Layer	C_{in}	C_{out}	Kernel	Stride	Padding
`PreBlock`	Convolution	2	64	(9×9)	1	4
	PReLU	64	64	-	-	-
`ResBlock` $(\times 8)$	Convolution	64	64	(3×3)	1	1
	Batch Norm	64	64	-	-	-
	PReLU	64	64	-	-	-
	Convolution	64	64	(3×3)	1	1
	Batch Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
`PostBlock`	Convolution	64	64	(3×3)	1	1
	Batch Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
`FinalBlock`	Convolution	64	2	(9×9)	1	4

Note: For each layer, the number of input C_{in} and output channels C_{out} is given. In the case of convolutional layers, the kernel size, stride, and padding are specified.

the used architecture, skip connections connect the output of the `PreBlock` with the output of the `PostBlock`. Additionally, for each ResBlock, a skip connection is used between input and output. The central gray block in Figure 4.2 sketches the effect of a skip connection schematically. Another notable feature is the group setting for the first and last convolution, which was set to two. This feature creates separate kernels for the first 32 and the last 32 channels, where the first ones are dedicated to the amplitude, and the last ones are dedicated to the phase.

4.3 Model Training

Different models were trained for the different data sets summarized in Section 3.6. All training sessions were performed under the same conditions. The models were trained for 300 epochs with a batch size of 64 images and a learning rate of 2×10^{-4}. The learning rate was selected using the learning rate finder implemented by Howard and Gugger [49]. An adapted Mean Absolute Difference (L1) loss

$$\text{Loss} = 10 \cdot \text{L1}\left(x_{\text{amp}}, y_{\text{amp}}\right) + \text{L1}\left(\text{HardTanh}(x_{\text{phase}}), y_{\text{phase}}\right), \tag{4.1}$$

$$\text{with}\quad \text{L1}(x, y) = |\ x - y\ |, \tag{4.2}$$

$$\text{and}\quad \text{HardTanh}(x) = \begin{cases} \pi, & \text{if x} > \pi \\ -\pi, & \text{if x} < -\pi \\ x, & \text{otherwise} \end{cases}, \tag{4.3}$$

was used as the loss function, where x is the predicted output of the network, and y is the true image. The hard hyperbolic tangent (HardTanh) function limits estimates for the phase to a range between $-\pi$ and π. Theoretically, higher values are possible due to the periodic properties of the phase. As multiples of π bring no new information and complicate the learning process, the HardTanh function ensures smooth training.

When training different models, the L1 loss outperformed the Mean Squared Error (MSE) loss because small-scale structures in (u, v) space could be reconstructed in more detail. In contrast, models trained with the MSE produced blurrier reconstructions.

The Adaptive Moment Estimation (ADAM) optimizer [59] was used for parameter optimization. Training processes with this optimizer converge faster than ones performed using momentumless SGD [4] at the cost of more training parameters. The training time for one epoch took around 220 s on a computer setup with the specifications summarized in Section 2.4. For all 300 epochs, this results in a training time of 18 h. Training times are compared with application times in Section 6.4. The default training configuration is shown in Appendix 3.

The primary data sets are noiseless, noisy, and white noise input data. Figure 4.3 shows the loss curves for these three cases. When comparing the loss curves for the different data sets, it can be seen that the lowest loss is reached with the noiseless data. Here, some spikes of the validation loss occur during the first half of the training. For the last 100 epochs, the training has been very smooth. While the loss for noisy and white noise data do not differ significantly, more spikes appear in the validation loss of the white noise training. However, the model recovers after the outliers, so the trained model is still suitable for reconstruction.

During the training processes, data augmentation helps to circumvent overtraining. One data augmentation routine is rotating input and output images to generate new examples in each batch creation step. Furthermore, different sampling masks for every image lead to different data coverages for each image. Tests show that models trained on different data coverages converge more smoothly and are more robust in application than models trained only on data sets with one sampling mask. Another advantage is the model's applicability to observations with similar data coverage but different source positions.

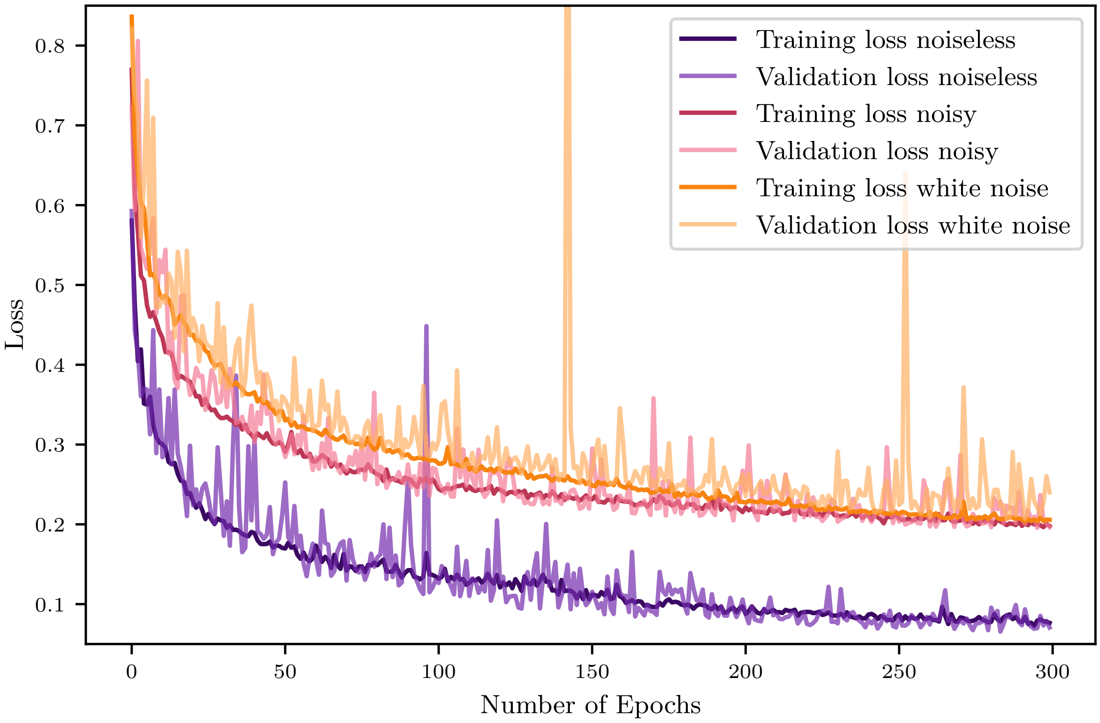


Figure 4.3: Loss curves for the training sessions with noiseless (purple), noisy (red), and white noise (orange) input data. Loss values against the number of epochs are shown for training and validation separately.

4.4 Layer Insights

The following chapter gives an insight into the functions of different model layers trained on noiseless data. For this purpose, a test image is sent through one block after the other, and the respective output images are displayed graphically. Here, only the convolutions are taken into account, and operations like batch norms and activations are dropped. The expectation is that with increasing network depth, the images will be filled up to the edges. Step by step, the available information from the convolutions is used to generate estimates for the frequels with missing information. Figure 4.4 shows a schematic illustration of the reconstruction process.

The insights into the different output stages reveal that the model is predicting reasonable values for frequels up to the edges with increasing depth. Already the output of the `PreBlock` has filled up values to the edge of the image. These values occur due to the learned bias, which is added to the output after the convolutional layer. Still, the shown illustration represents only one channel. Especially the convolution inside the `FinalBlock`, where 64 input channels contribute to the output channels for amplitude and phase map, the most remarkable improvement takes place. To give an overview of the outputs for all channels would go beyond the scope of this work.

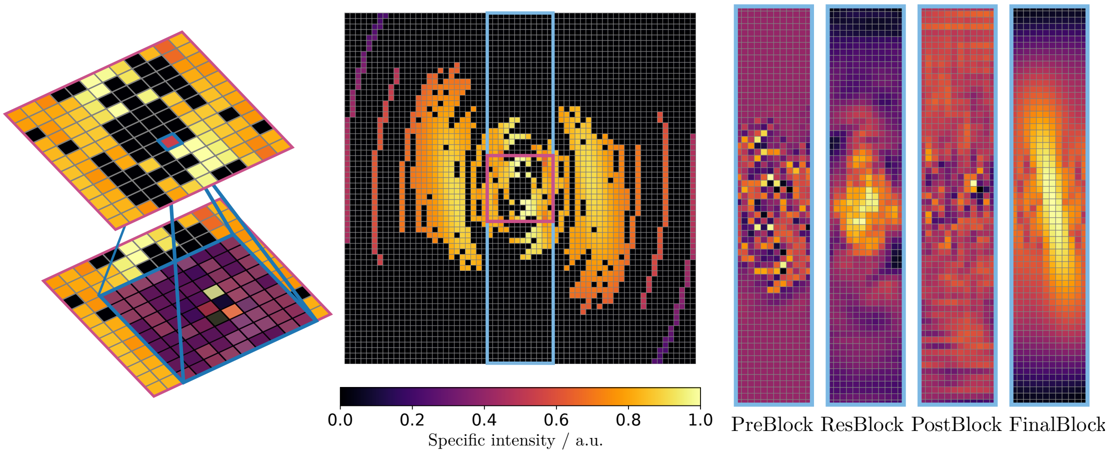


Figure 4.4: Schematic illustration of the reconstruction of missing information inside the deep learning model for one channel. The (9×9) convolution on the left demonstrates how frequels with sampled information serve to find values for frequels without information. The displayed extract (dark blue) originates from the central region of the incomplete amplitude map shown in the middle. Unknown frequels do not contribute to the output of the first convolution as they have a value of zero. The light-blue highlighted area indicates the reconstruction process after the different model stages. The model predicts reasonable values for frequels up to the edges with increasing depth. For better visualization, all displayed images are rescaled to values between zero and one.

5 Deep Learning-based Cleaning

> *"What he was in FACT doing was moving the dirt around with a broom, to give it a change of scenery and a chance to make new friends."*
> — Terry Pratchett, *Maskerade*

An essential task when analyzing radio interferometer observations is cleaning the data before it is science ready. The raw inverse Fourier transform is called a dirty image for a reason. Sometimes the dirt is difficult to distinguish from the sources' emissions. For this reason, the inexperienced analyst is not protected from cleaning artificial sources into the final sky distributions. Since the goal is reproducible science whose results are independent of the analyst's experience, the idea of a machine-learning-based analysis emerged. The ability of deep learning algorithms to analyze image data quickly makes them an ideal tool for cleaning observations of radio interferometers. This chapter's trained deep learning models are applied to dedicated test sets to assess their reconstruction ability.

5.1 Reconstruction of Visibility Data and Source Distributions

After training, the deep learning models are suitable for reconstructing missing information in the Fourier space. The example source from Section 3.1 is analyzed visually for a first evaluation of the reconstruction quality. For this analysis, the reconstructed distributions are visualized side by side. Additionally, the difference between reconstruction and simulated truth highlights areas with reconstruction problems. This way, issues can be identified, and a general impression of the reconstruction quality arises. The following shows the example source for the noiseless, noisy, and white noise input data. A summary of the exact characteristics of the three data sets was given in chapter Section 3.6.

In the next step, the reconstructed amplitude and phase distributions can be used to create a clean image of the radio galaxy. First, the scaling from equation (3.9) is undone, and the values are transformed into complex numbers with real and imaginary parts using Euler's formula. The reconstructed clean image in local space is obtained by applying the inverse Fourier transform. Again, reconstruction, simulated truth, and the difference between both are displayed next to each other for a visual comparison. Furthermore, the viewing angle for the reconstructed and the simulated source distribution is determined to compare them. The viewing angle is determined by applying a Principal Component Analysis (PCA) [93]. A PCA searches for a new basis in the two-dimensional images. The corresponding one-dimensional projection maximizes the information contained inside the image. In the case of radio galaxies, this projection runs directly through the sources' jet axis, revealing the source's jet orientation. After determining the covariance matrix, eigenvalues and eigenvectors of this matrix are calculated. Finally, the tangent relation is utilized to compute the orientation angle of the radio galaxy. Comparing the reconstructed source distribution's value with the one for the simulated source distribution illustrates the reconstruction offset. The results for the example source are again shown for all three data sets.

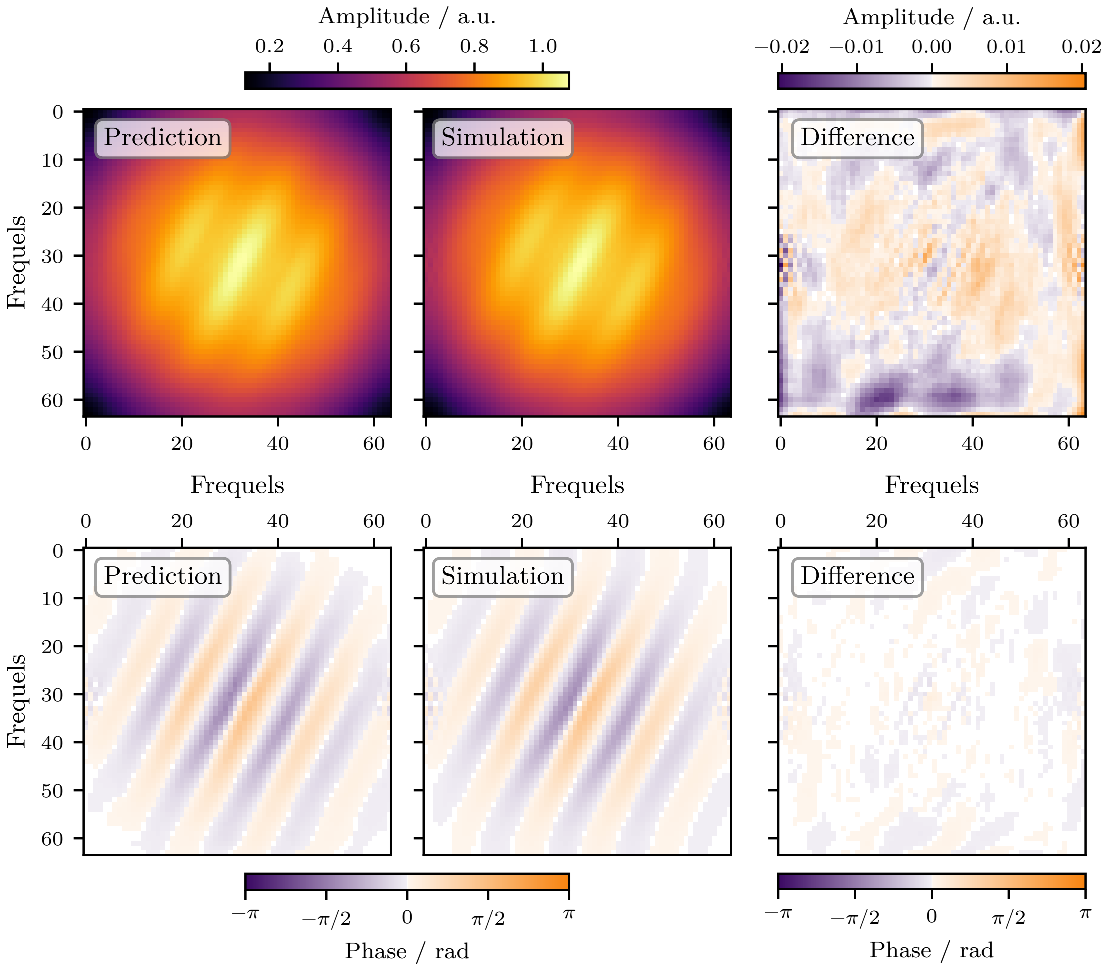


Figure 5.1: Reconstruction of amplitude and phase maps using a model trained on noiseless input data. Visualization of prediction (left), simulated distribution (middle), and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

Noiseless Input Data

Figure 5.1 shows the reconstruction generated by the neural network trained on noiseless input data. The figure is split into amplitude (top) and phase distributions (bottom). The first, second, and third columns visualize prediction, simulated truth, and the difference between prediction and truth, respectively. Analyzing the difference between predicted and simulated amplitude reveals a good agreement with maximum deviations amounting to around 2 %. These maximum deviations occur at the edges of the amplitude distributions. The same is valid for the phase distribution reconstructions.

This conformity between prediction and truth directly translates to the clean image shown in Figure 5.2. Again, the figure is separated into three columns: prediction (left), simulation (middle), and the difference between prediction and simulation (right). The difference between predicted and simulated source brightness distribution indicates a reasonable reconstruction of the radio galaxy's small and large-scale features. All simulated source components are visible in the reconstructed source distribution. Still, their brightness is slightly underestimated, which is valid for the entire source region, leading to deviations around 3%. Additionally, the jet

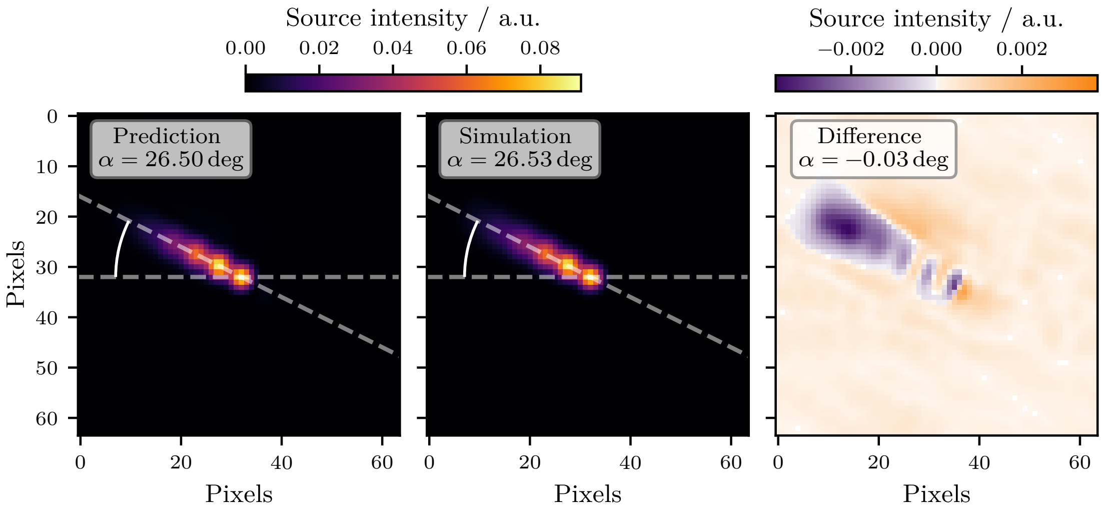


Figure 5.2: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for noiseless input data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right). The jet orientation α, calculated using a PCA, is given for both reconstructed ($\alpha = 26.50°$) and simulated source distributions ($\alpha = 26.53°$) leading to an offset of $-0.03°$.

orientation is restored correctly. The slight offset of −0.03° between simulated and reconstructed source jet orientation underlines this result.

Noisy Input Data

A visual analysis of the reconstructed amplitude and phase distributions generated with noisy input data, illustrated in Figure 5.3, shows that the results hardly differ from those generated with noiseless input data. Here, too, the amplitude is slightly underestimated. The maximum deviation is about 5 % for the amplitude and 15 % for the phase.

When transforming the reconstructed distribution into local space, a minor influence of the additional noise is recognizable in the cleaned source image shown in Figure 5.4. Here, the brightness is underestimated, with slightly higher deviations than in the case of the noiseless input data, while the core component is overestimated. The source's jet orientation reconstruction worked successfully with a difference of 0.77°, which is 25 times the amount compared to the case with noiseless input data.

White Noise Input Data

For the third case, besides the noise in the image space, additional white noise was added to the visibility space. This noise corruption is expected to complicate the reconstruction of the data further. That this assumption is invalid for the shown example is directly recognizable when looking at the reconstructed source distribution shown in Figure 5.5. The reconstructed source orientation deviates in the same order of magnitude as for noisy input data. The offset is 0.63°. Furthermore, no increase in artifacts in the counter jet direction is visible compared to the ones in the images generated with noiseless and noisy input data. Again, an underestimation of the source brightness is visible.

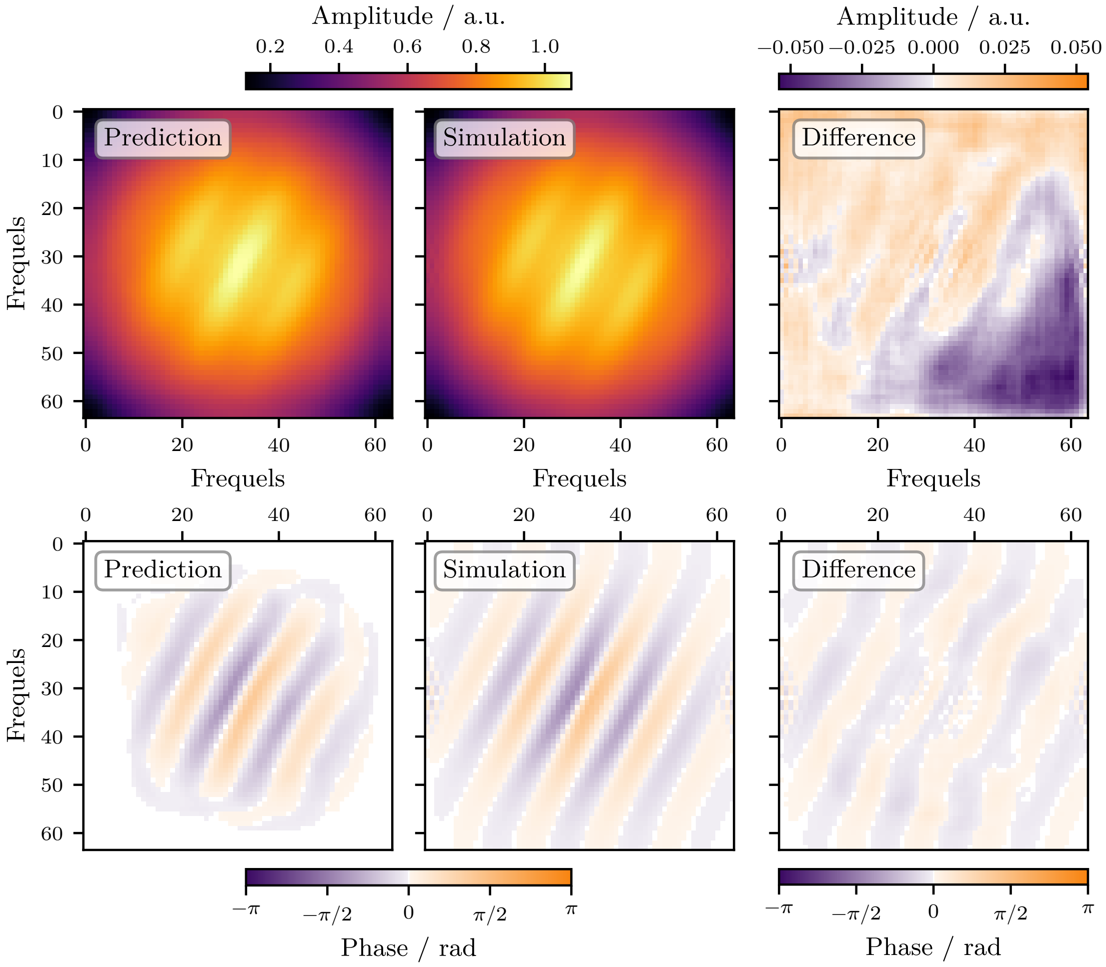


Figure 5.3: Reconstruction of amplitude and phase maps using a model trained on noisy input data. Visualization of prediction (left), simulated distribution (middle), and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

Table 5.1: Comparison of the reconstruction results for the example source with different input data types.

Input type	Amplitude		Phase		Source Brightness	
	diff_{max}	$\text{diff}_{\text{mean}}$	diff_{max}	$\text{diff}_{\text{mean}}$	diff_{max}	$\text{diff}_{\text{mean}}$
noiseless	0.02	0.01 ± 0.01	0.31	0.02 ± 0.02	0.004	0.0003 ± 0.0001
noisy	0.05	0.01 ± 0.01	0.38	0.08 ± 0.07	0.010	0.0005 ± 0.0011
white noise	0.05	0.01 ± 0.01	0.38	0.07 ± 0.06	0.008	0.0005 ± 0.0010

Note: diff_{max} and $\text{diff}_{\text{mean}}$ denotes the maximal and mean absolute difference between prediction and simulation for the amplitude, phase, and source brightness distributions. Shown values are in arbitrary units.

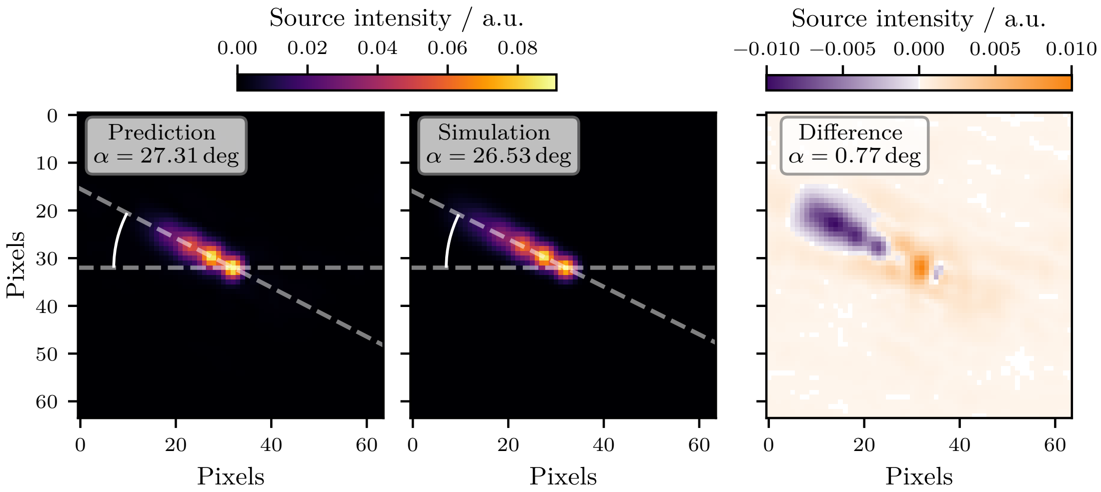


Figure 5.4: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for noisy input data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right). The jet orientation α, calculated using a PCA, is given for both reconstructed ($\alpha = 27.31°$) and simulated source distributions ($\alpha = 26.53°$) leading to an offset of 0.77°.

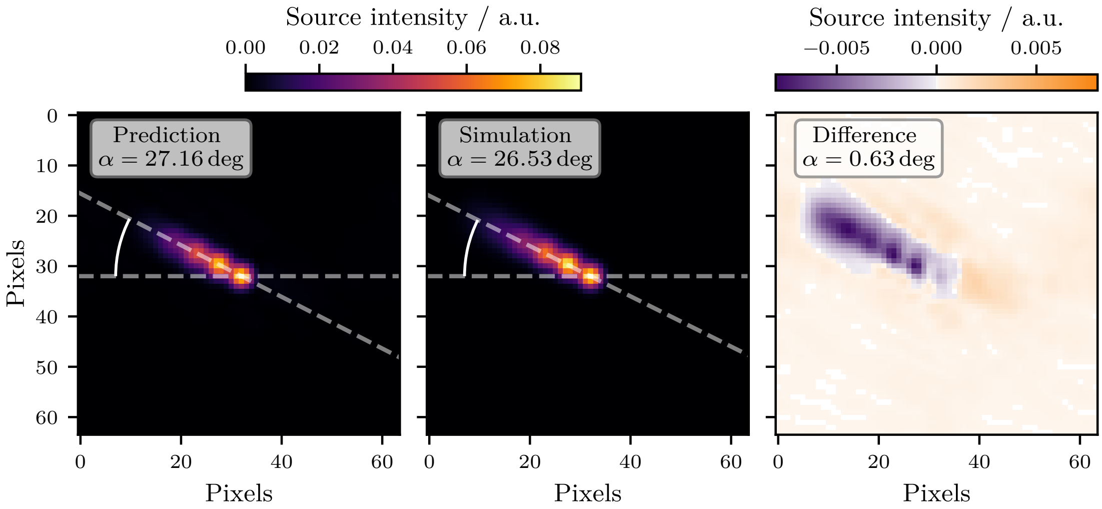


Figure 5.5: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for white noise input data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right). The jet orientation α, calculated using a PCA, is given for both reconstructed ($\alpha = 27.16°$) and simulated source distributions ($\alpha = 26.53°$) leading to an offset of 0.63°.

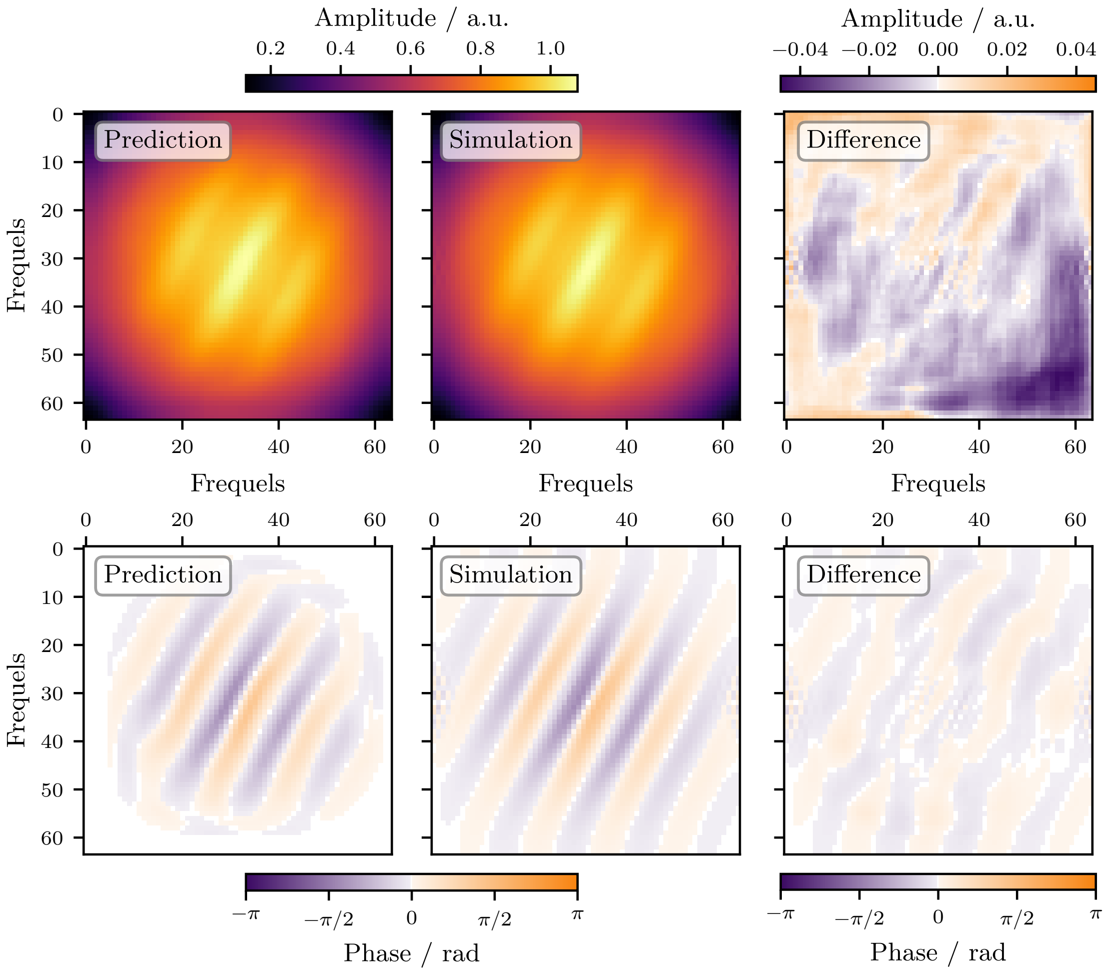


Figure 5.6: Reconstruction of amplitude and phase maps using a model trained on white noise input data. Visualization of prediction (left), simulated distribution (middle), and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

Table 5.2: Overview of the parameter settings utilized to create the clean images using WSCLEAN

Parameter	Setting
size	64 pixels
scale	1.25 masec
mgain	0.1
gain	0.01
niter	50 000

More significant deviations between predictions and simulations are also noticeable in the amplitude and phase distributions illustrated in Figure 5.6. Here, the reconstruction quality for the amplitude distribution decreases significantly, as, from the previous examples, known underestimations at the edges extend further to the inner regions. Contrary, this decrease is not visible in the phase distribution.

Comparison

After the visual investigation, the reconstruction quality of the presented example predictions becomes more quantitative. Therefore, the absolute difference between prediction and simulation is analyzed. The maximum and mean values for absolute amplitude, phase, and source brightness difference are calculated for the example with noiseless, noisy, and white noise input data. Table 5.1 summarizes the results. The values for noisy and white noise input data do not differ much. Values are slightly higher than the ones for the noiseless data example. The maximum amplitude difference doubles when not using noiseless data. In the case of the phase, the maximum difference stays in the same range for all three cases. The decrease in reconstruction ability for noisy and white noise input data is visible when looking at the phase's mean difference, which is four times higher than for the noiseless input data. All in all, deviations are small, strengthening the visual inspection results. Although the results for amplitude and phase distributions are slightly getting worse, this has no significant effect on the reconstructed source distribution. Advanced evaluation techniques have to be applied to get a more profound overview. Therefore, more detailed evaluations using different metrics on a complete test data set will be presented in Chapter 6.

5.2 Comparison to Established Cleaning Software

Comparing the reconstruction results with those of established cleaning software allows for highlighting the pros and cons of the developed neural network approach. To generate a data set suitable to be analyzed with WSCLEAN, the one from Section 3.1 is used as a basis. Afterward, it is preprocessed as described in Section 3.4. As WSCLEAN cannot read FITS files, a conversion into measurement set format using Common Astronomy Software Applications (CASA) [19] is necessary. Therefore, the `casa importuvfits` task is utilized. In the next step, the resulting visibility data can be cleaned utilizing WSCLEAN with the cleaning parameters summarized in Table 5.2. The obtained clean images of the radio galaxies are convolved with the clean beam of the observation, compare Section 1.5. This convolution is crucial because WSCLEAN iteratively subtracts point sources from the data and thus creates a source model. The point source model is then convolved with the beam of the best possible resolution to generate the reconstructed source distribution. The clean beam is irrelevant for the neural network models since reconstruction occurs in Fourier space. Also, in the case of the deep learning model, the FITS files must be converted before they can be reconstructed. The gridder implemented in the `pyvisgen` package is applied for this task. Afterward, the incomplete amplitude and phase maps are reconstructed utilizing the deep learning model trained on noiseless data. Figure 5.7 visualizes the `radionets` reconstruction, the simulated source distribution, and the clean image generated using WSCLEAN. Both reconstructions match the simulated source distribution well. Additionally, the reconstructed jet orientation slightly differs from the simulation. The WSCLEAN reconstruction shows increased background noise, which is not present in the `radionets` reconstruction, although the model was not trained on gridded data. Section 6.3 provides a more quantitative comparison between the `radionets` and WSCLEAN reconstructions.

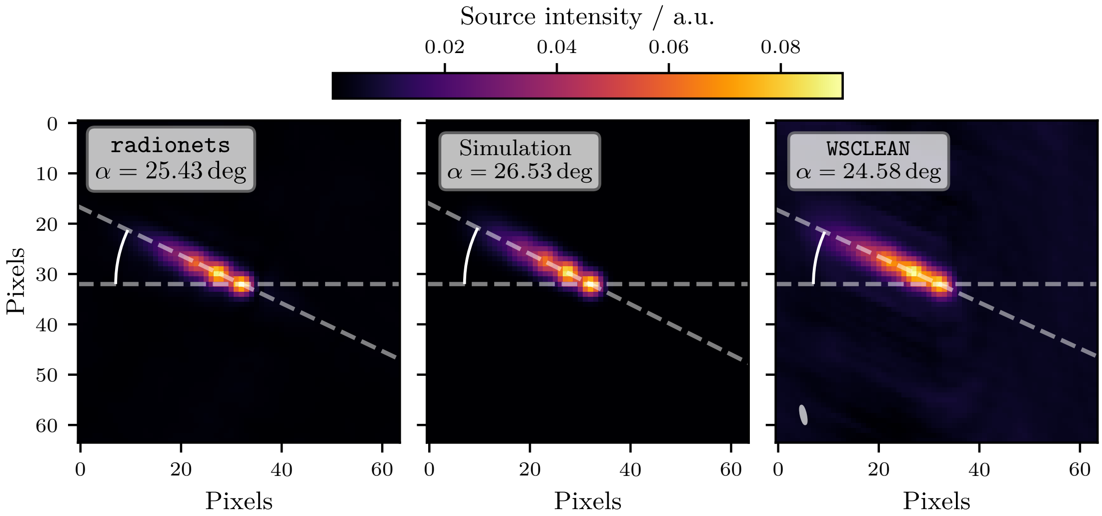


Figure 5.7: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions of the deep learning model trained on noiseless data (left), simulated brightness distribution (middle), and the clean image generated using WSCLEAN. In the case of WSCLEAN, the clean beam is shown in the lower left corner. The jet orientation α, calculated using a PCA, is given for the `radionets` reconstruction ($\alpha = 25.43°$), the simulated source distributions ($\alpha = 26.53°$), and the WSCLEAN reconstruction ($\alpha = 24.58°$).

Reconstruction Evaluation 6

"You don't have to test everything to destruction just to see if you make it right."
— Terry Pratchett and Neil Gaiman, *Good Omens*

Generally, the reconstruction quality of deep learning models can not be judged only by looking at single source distributions. A more sophisticated method is needed to see if the deep learning-based reconstruction works fine. Of course, we can only test some aspects in the smallest detail. Therefore, the approach is to predefine clear benchmarks and test the developed models on a dedicated test data set to get an estimate of the goodness of the reconstructions.

In the context of this thesis, three evaluation methods were developed for the `radionets` framework in close collaboration with Felix Geyer. These methods test the reconstruction performance in the context of the source areas, jet orientations, and the core component's mean intensity.

6.1 Evaluation Techniques

Advanced evaluation methods were developed to assess the reconstructions in more detail using the deep learning models. These are also used to compare the neural network-based methods with the established cleaning method WSCLEAN. Three methods were implemented to quantify the reconstruction quality and applied to a test data set of 10 000 images. The default evaluation configuration is shown in Appendix 4.

Source Area Ratio

Correct reconstruction of the radio galaxy's morphology is essential when cleaning radio interferometric data. An underestimation or overestimation of the source area is unwanted. The source area ratio between the reconstructed and true source area is calculated to get a statement on an extensive test data set. The first step defines a source boundary of 10 % of the peak flux. Afterward, the source area is calculated using the Leibnitz sector formula [130]. The Leibniz sector formula describes the connection between the path integral and the enclosed area. The contour line determines the enclosure of the source through `matplotlib`. Then the path integral is approximated numerically. Finally, the Leibnitz sector rule yields the searched area. A statement about the reconstruction quality can be made by comparing the true and the reconstructed area. For an ideal reconstruction, the ratio is one. Lower values indicate an underestimation of the area, while higher values indicate an overestimated area. This evaluation method shows if the reconstructed source distributions have many background artifacts. Images with reconstruction errors show lesser flux in the source area. At the same time, there appear more background artifacts in these cases. Figure 6.1 illustrates the distributions of the source area ratios for the deep learning models with different noise levels. The distributions for the deep learning model trained on noiseless (purple), noisy (red), and white noise (orange) input data match in height and location. All three distributions peak at a mean value of ≈ 1, representing the optimal value

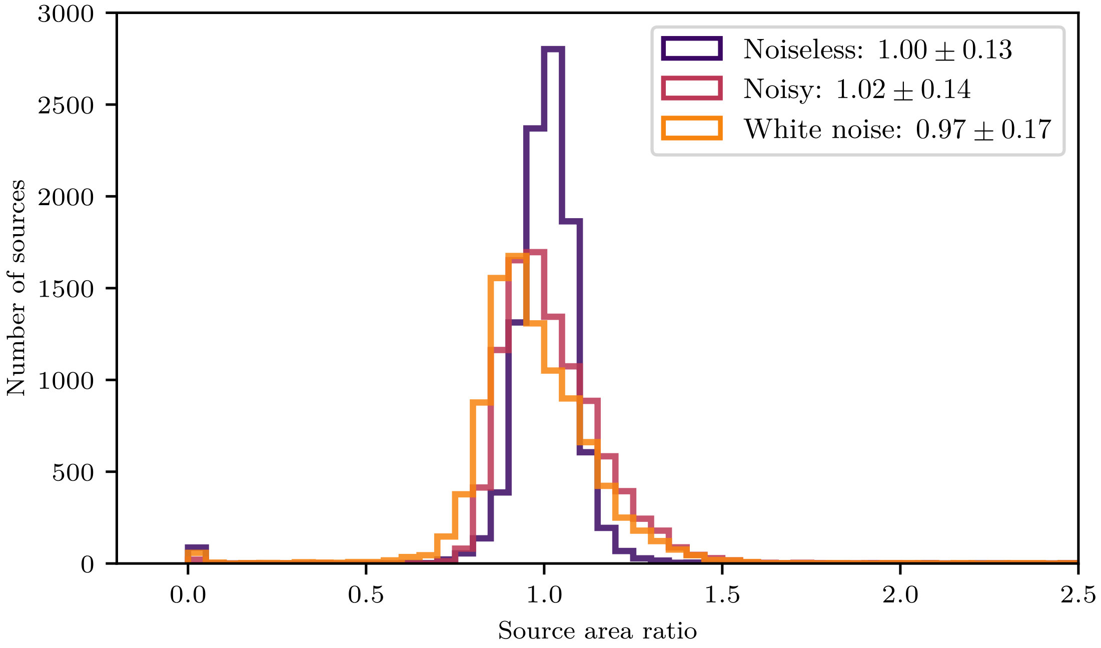


Figure 6.1: Histogrammed source area ratios. Results for the deep learning model trained on noiseless data (purple), noisy data (red), and white noise data (orange) are displayed. Mean and standard deviation help to compare the three distributions.

for the source area ratio. With increasing complexity, the standard deviation of the distributions increases as expected. The bin at zero indicates that some source reconstructions fail. This amount is low for all three data types. The deviations between predicted and simulated source areas result in mean and standard deviations of 1.00 ± 0.13 (noiseless), 1.02 ± 0.14 (noisy), and 0.97 ± 0.17 (white noise).

The excellent agreement for the reconstruction results obtained for the model trained on noiseless and noisy input data indicates that the architecture reaches the same reconstruction level for both input types.

Source Orientation Offset

In Chapter 5, it was shown that the evaluation of source orientation helps to investigate the reconstruction quality of the trained deep learning model. The difference between reconstructed and true source orientation, called orientation offset, is shown for the test data set in Figure 6.2. Histogrammed distributions are illustrated for the different deep learning models trained on noiseless (purple), noisy (red), and white noise (orange) input data. Again, all three distributions generated from the deep learning-based results match well, supported by the mean and standard deviations of $(0.07 \pm 0.68)^\circ$ (noiseless), $(0.14 \pm 1.00)^\circ$ (noisy), and $(0.04 \pm 1.08)^\circ$ (white noise). While the mean values do not differ significantly, an increase in the standard deviation is visible with the increasing complexity of the input data. The slight deviation from the optimal offset of zero indicates an excellent source orientation reconstruction of the deep learning models.

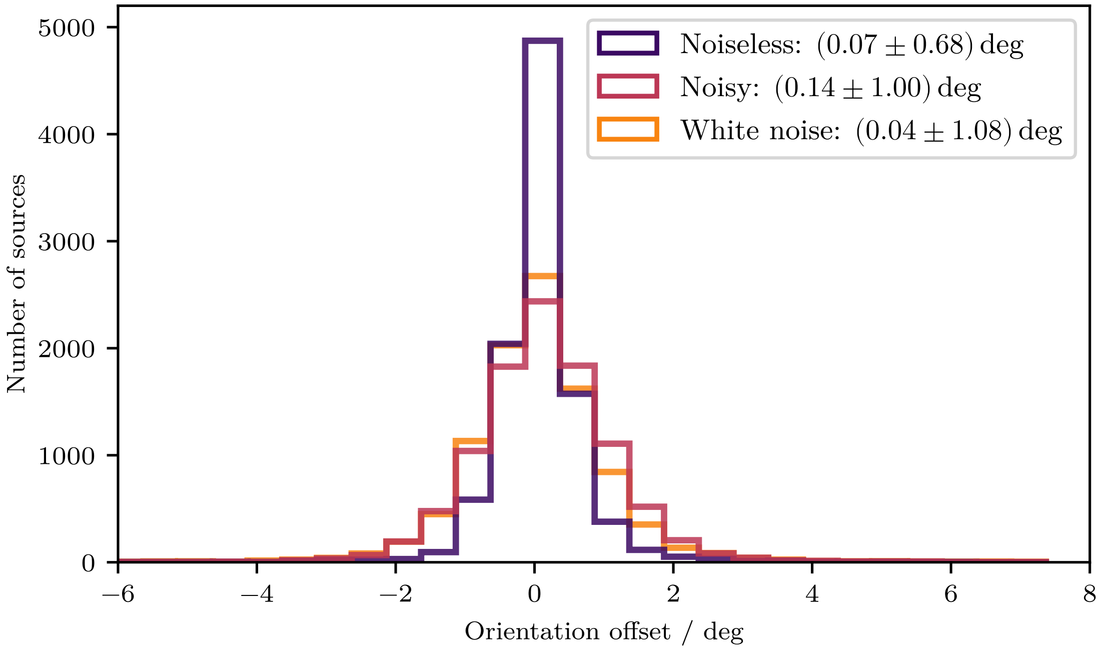


Figure 6.2: Histogrammed source orientation offsets. Results for the deep learning model trained on noiseless data (purple), noisy data (red), and white noise data (orange) are displayed. Mean and standard deviation help to compare the three distributions.

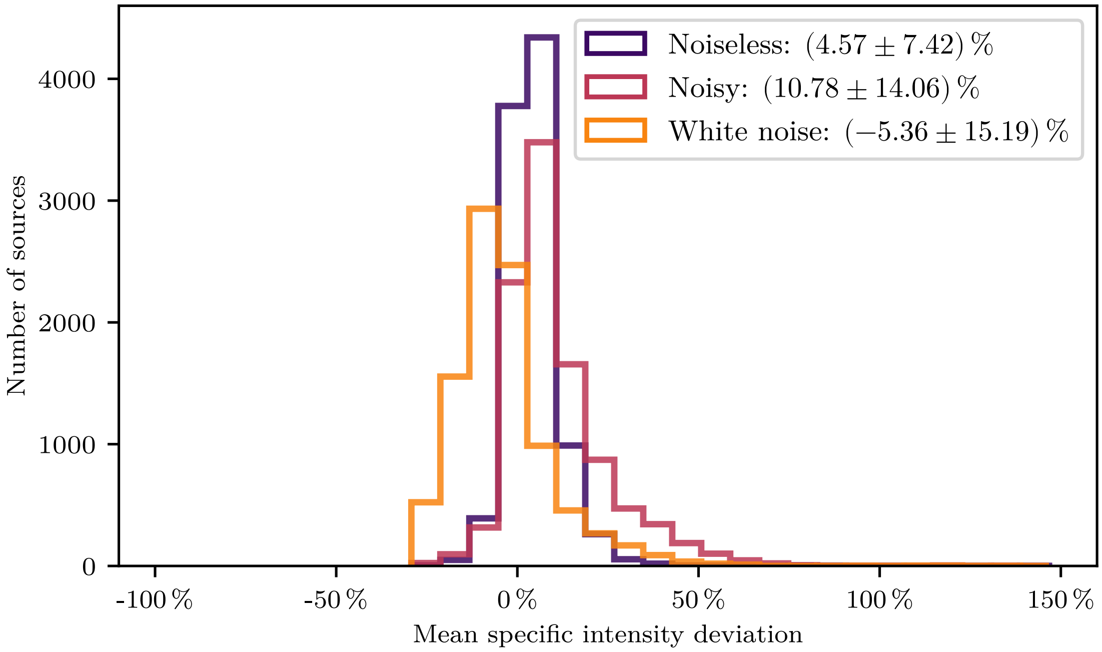


Figure 6.3: Histogrammed mean source intensity deviation for the sources' core components. Results for the deep learning model trained on noiseless data (purple), noisy data (red), and white noise data (orange) are displayed. Mean and standard deviation help to compare the three distributions.

Table 6.1: Overview of the mean results of the three evaluation methods with clean, noisy, and white noise input maps. The part in the middle gives a summary of input maps with different mask filling. Furthermore, results generated with WSCLEAN and on a gridded data set are presented in the lower part.

Model type	Orientation offset / °	Brightness deviation / %	Area ratio
Noiseless	0.07 ± 0.68	4.57 ± 7.42	1.00 ± 0.13
Noisy	0.14 ± 1.00	10.78 ± 14.06	1.02 ± 0.14
White noise	0.04 ± 1.08	−5.36 ± 15.19	0.97 ± 0.17
Sampling Density			
10 %	0.26 ± 1.52	−13.35 ± 17.41	0.74 ± 0.18
30 %	0.07 ± 0.71	10.95 ± 9.86	1.01 ± 0.11
50 %	0.15 ± 0.72	12.54 ± 9.27	1.02 ± 0.11
70 %	−0.03 ± 0.90	25.44 ± 14.09	1.06 ± 0.13
RIME simple	−4.19 ± 6.01	89.48 ± 92.06	1.33 ± 0.69
WSCLEAN	−4.43 ± 6.74	13.66 ± 76.35	1.82 ± 1.24

Note: A new training set is used for different noise models. The data sets consist of 50 000 training maps, 10 000 validation maps, and 10 000 test maps. We do not train a new model for different sampling densities but create dedicated test data sets consisting of 10 000 test maps. The evaluation is done using the deep learning model trained on noiseless input data. The results created with WSCLEAN and the gridded data set were calculated using 1000 test maps. The gridded data set was evaluated with the deep learning model trained on noiseless data.

Mean Specific Intensity Deviation

The core component's mean specific intensity is examined by the third method. First, the source's components are determined using the `blob detection` from `scikit-image` [129]. Then the intensity of the brightest component is averaged. In the case of the simulated radio galaxies, the brightest component is the core component of the source. Finally, the flux values are compared, and the deviation between reconstruction and simulation is calculated. Figure 6.3 illustrates the histogrammed results for the different models. The deep learning-based models' results differ a bit for this evaluation method. While the models trained on noiseless and noisy input data tend to overestimate the reconstructed mean brightness, it is underestimated by the model trained on white noise data for most cases. These characteristics can also be seen from the mean and standard deviations of 4.57 ± 7.42, 10.78 ± 14.06, and -5.36 ± 15.19 for the models trained on noiseless (purple), noisy (red), and white noise (orange) input data, respectively. As before, the standard deviation increases with the advanced complexity of the input data.

6.2 Influence of Sampling Densities

The upper part of Table 6.1 summarizes the results of the evaluation techniques introduced in Section 6.1 for the deep learning models trained on noiseless, noisy, and white noise input data. These data have a (u, v) coverage of around $(25 \pm 2)\,\%$. The exact values are summarized in Section 3.6. Besides the observation length, the radio interferometer's orientation to the source position influences the sampling density.

To evaluate the influence of the input data's sampling density on the reconstruction ability of the deep learning model, data sets with different sampling densities were generated and evaluated

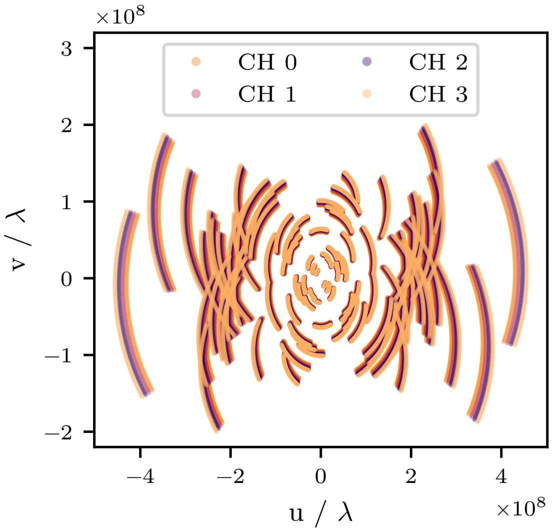


Figure 6.4: (u, v) coverage for a simulated observation with four frequency channels. The additional channels lead to a slightly improved coverage of amplitude and phase frequels.

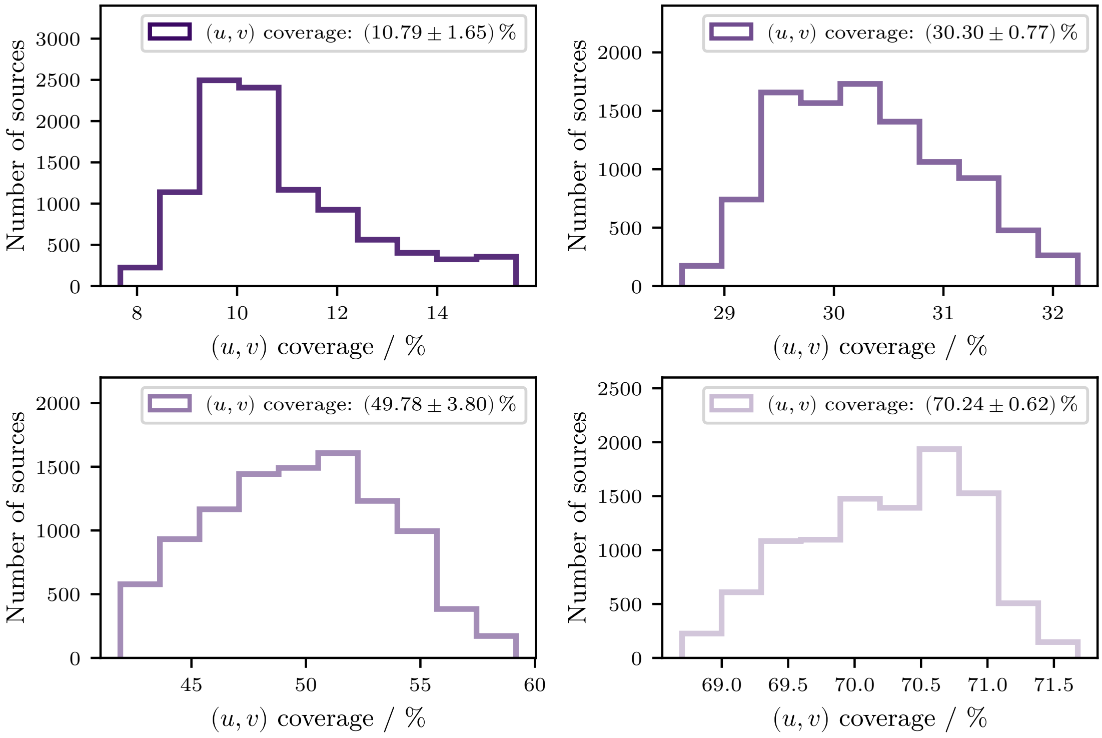


Figure 6.5: Histogrammed (u, v) coverages for the four different data sets. The aim was to reach a 10 %, 30 %, 50 %, and 70 % (u, v) coverage for the input maps. Mean and standard deviations help to validate the results.

by applying the model trained on noiseless input data. The tested sampling densities are 10 %, 30 %, 50 %, and 70 %. Such high densities were only achievable by introducing the multi-channel option for the sampling mask generation. This option enables the simulation of several frequency channels, which increases the (u, v) coverage of the simulated observation. Figure 6.4 illustrates the improved (u, v) coverage on the example of a simulation with four frequency bands. Here, the number of samples was increased by a factor of four. However, most of the samples result in more information collected per frequel, which increases the sensitivity in actual observations as the signal-to-noise ratio is improved.

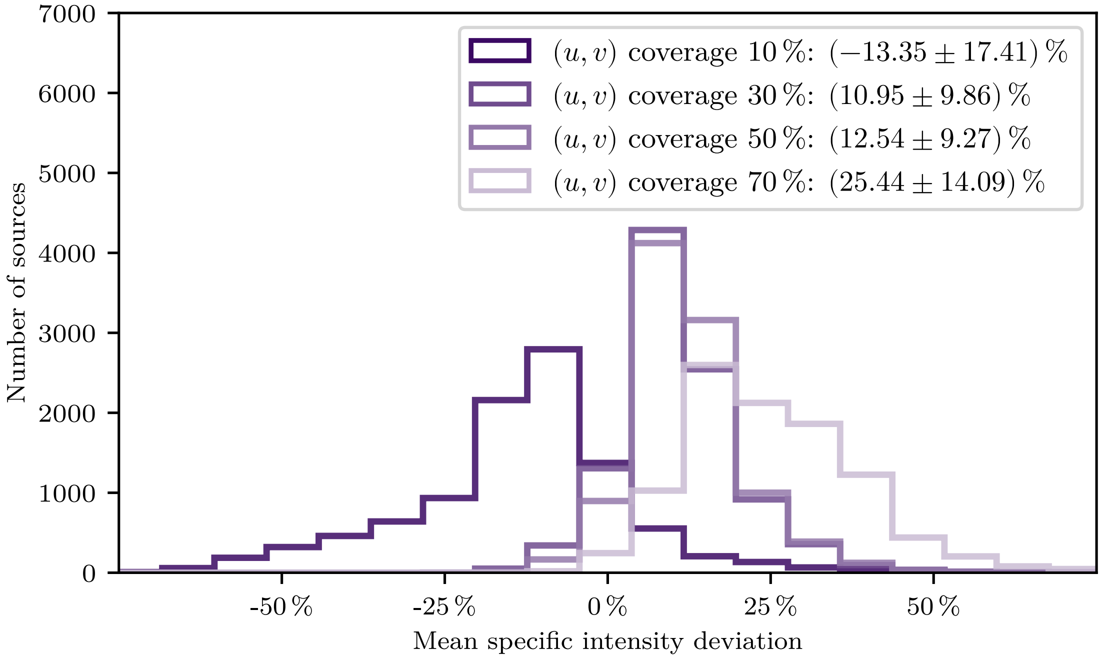


Figure 6.6: Histogrammed mean source brightness deviation for the sources' core components. Results for the deep learning model trained on noiseless data for input data with 10 %, 30 %, 50 %, and 70 % (u, v) coverages of the input maps. Mean and standard deviation help to compare the four distributions.

This effect on the signal-to-noise ratio is neglectable in the case of sampling mask-based simulations. A highly increased number of channels is essential to generate data sets with (u, v) coverages of 50% and higher. Additionally, the simulated observation length has to be increased. As source positions are chosen randomly, resulting sampling masks and, thus, resulting (u, v) coverages differ. Here, the aimed sampling densities of 10 %, 30 %, 50 %, and 70 % are mean values for the different data sets. An overview of the exact sampling densities is given in Figure 6.5. When comparing these (u, v) coverages with the ones of the noiseless data set, see Figure 3.7, it becomes clear that their standard deviations are minor for all but the 50 % data set. Therefore, the spread of (u, v) coverages is not expected to cause a performance decrease when the noiseless model gets applied to the data. The lower part of Table 6.1 summarizes the reconstruction results evaluated with the methods introduced in this chapter. The distributions' mean and standard deviations show that the reconstruction quality is correlated with the mask fillings. This correlation becomes especially clear when looking at the mean brightness deviation.

When only a little information is available in the input maps, the brightness of the sources' core component is mainly underestimated. Generally, the brightness reconstruction ability of the model is decreased for this input data leading to the distribution shown in Figure 6.6. The mean and standard deviation calculated on the evaluation of 10 000 test sources is (-13.35 ± 17.41) %. In the case of a higher (u, v) coverage than the one trained on, an overestimation of the core component's brightness occurs.

This overestimation increases with increasing (u, v) coverage, which is also shown in Figure 6.6. The mean and standard deviation for the distributions generated with 30 %, 50 %, and 70 % (u, v) coverage data sets are (10.95 ± 9.86) %, (12.54 ± 9.27) %, and (25.44 ± 14.09) %, respectively.

The histogrammed source area ratios visualized in Figure 6.7 indicate a performance decrease for the data set with a 10 % (u, v) coverage. Here, the mean area ratio is 0.74 ± 0.18. Input data

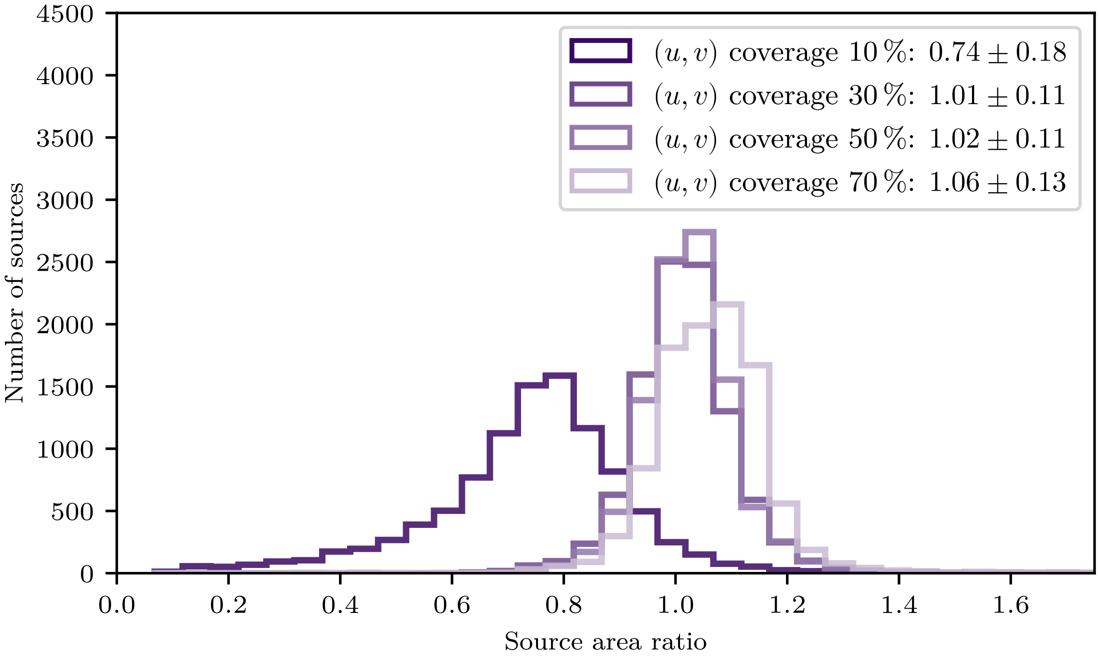


Figure 6.7: Histogrammed source area ratios. Results for the deep learning model trained on noiseless data for input data with 10 %, 30 %, 50 %, and 70 % (u, v) coverages of the input maps. Mean and standard deviation help to compare the four distributions.

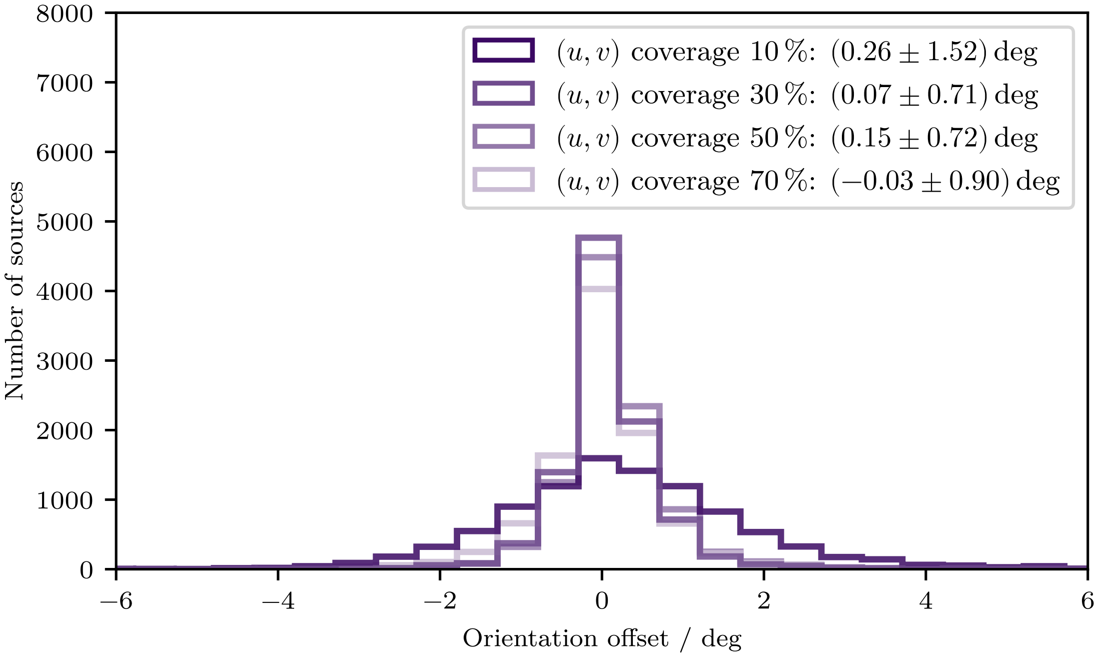


Figure 6.8: Histogrammed source orientation offsets. Results for the deep learning model trained on noiseless data for input data with 10 %, 30 %, 50 %, and 70 % (u, v) coverages of the input maps. Mean and standard deviation help to compare the four distributions.

with increased (u, v) coverages seem not to influence this evaluation technique as the mean and standard deviation of 1.01 ± 0.11, 1.02 ± 0.11, and 1.06 ± 0.13 for a 30 %, 50 %, and 70 % (u, v) coverage, respectively, do not differ significantly. Additionally, they are in the same order as the results calculated with noiseless input data; see Table 6.1.

Finally, the source orientation offset is examined in Figure 6.8. Input data with 10 % (u, v) coverage slightly underperforms with a mean and standard deviation of $(0.26 \pm 1.52)°$ compared with $(0.07 \pm 0.71)°$, $(0.15 \pm 0.72)°$, and $(-0.03 \pm 0.90)°$ for input data with 30 %, 50 %, and 70 % (u, v) coverage, respectively. Still, the deviation is minimal for all four cases. Again, the results are in the same range as the ones for the model evaluation with noiseless input data.

6.3 Comparison with WSCLEAN

In order to compare the deep learning approach with established cleaning software, a simulated data set consisting of visibilities was generated; see Section 3.4. Here, a simple RIME formulation simulates the uncorrupted visibilities of the antenna pairs in the interferometer array, which are then saved in FITS format. The exact formula can be found in equation (3.8). Additionally, a post-processing of the FITS files is implemented as WSCLEAN uses measurement sets (ms) as input format. The conversion is performed using the `casa importuvfits` task. Afterward, WSCLEAN runs with the cleaning parameters summarized in Table 5.2 to evaluate a test set consisting of 1000 images.

In parallel, the simulated visibilities are gridded with the implementation inside the `pyvisgen` package. This way, 1000 amplitude and phase maps were generated. They are reconstructed using the deep learning model trained on noiseless data. By applying the inverse Fourier transform, reconstructed source distributions are obtained.

As the next step, the evaluation methods are used to inspect and compare the results for both cleaning approaches. For all three methods, the overall performance is worse than for the previously described results in this chapter. Figure 6.9 visualizes the histogrammed source area ratios. While the distribution for the `radionets` approach peaks at 1.25, no specific peak can be found in the distribution for WSCLEAN. This result indicates that WSCLEAN has problems handling the different (u, v) coverages in the test data set. Furthermore, the cleaning settings are not optimized for the different visibility sets, which is usually done when analyzing actual observations. Such parameter optimization by hand is not feasible for a data set consisting of 1000 sources. Therefore, a compromise with fixed parameter settings must be found for cleaning the complete data set, leading to a mean and standard deviation of 1.33 ± 0.69 and 1.82 ± 1.24 for `radionets` and WSCLEAN, respectively.

The main problem with the RIME data is the wide variety of maximum resolutions for the different simulations. When gridding radio interferometer data, the maximum resolution has to be considered to achieve the best possible results. Here, a field of view of $85\,\mathrm{m}''$ is chosen for all clean images, and no individual adjustment is performed. Thus, in the cases where long baselines are excluded in the simulations, the pixels are chosen smaller than the maximum resolution of this measurement. This incorrect choice leads to edge effects and reconstruction errors. Furthermore, the (u, v) coverage differs more than in the other data sets; see Table 3.1. For the RIME simulations, the mean (u, v) coverage is $(29.78 \pm 13.38)\,\%$. For the source area ratio, these effects are more noticeable in the case of wsclean. The evaluation with `radionets` is more robust against the resolution differences in the data.

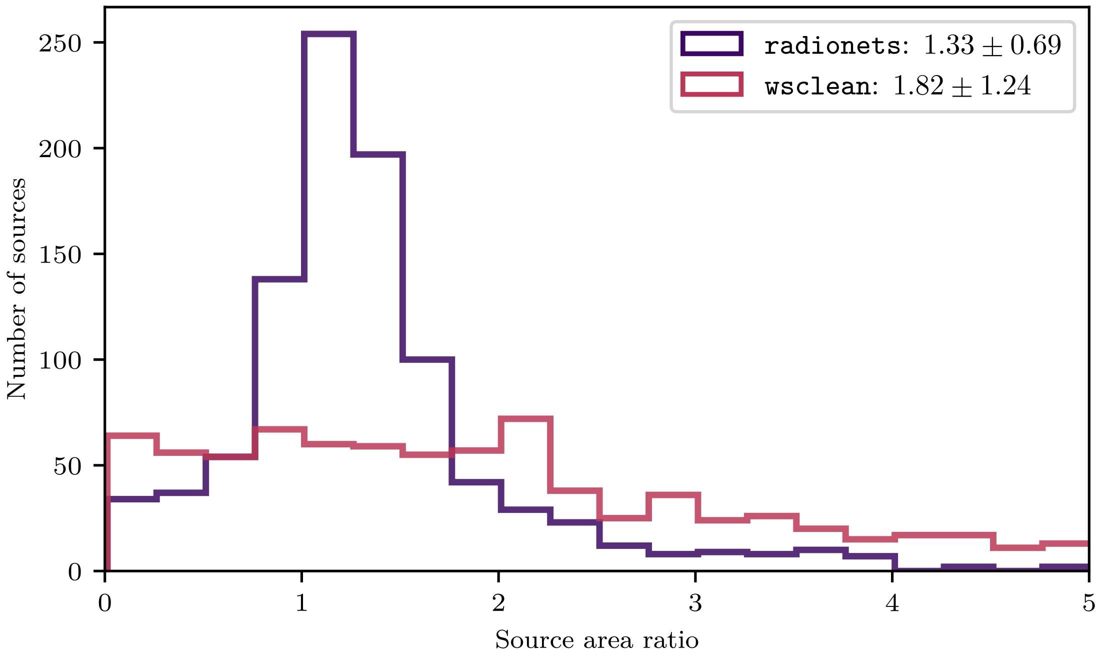


Figure 6.9: Histogrammed source area ratios. Results for the deep learning model trained on noiseless data evaluated on the gridded RIME data set (purple) and results generated with WSCLEAN (red). Mean and standard deviation help to compare the distributions.

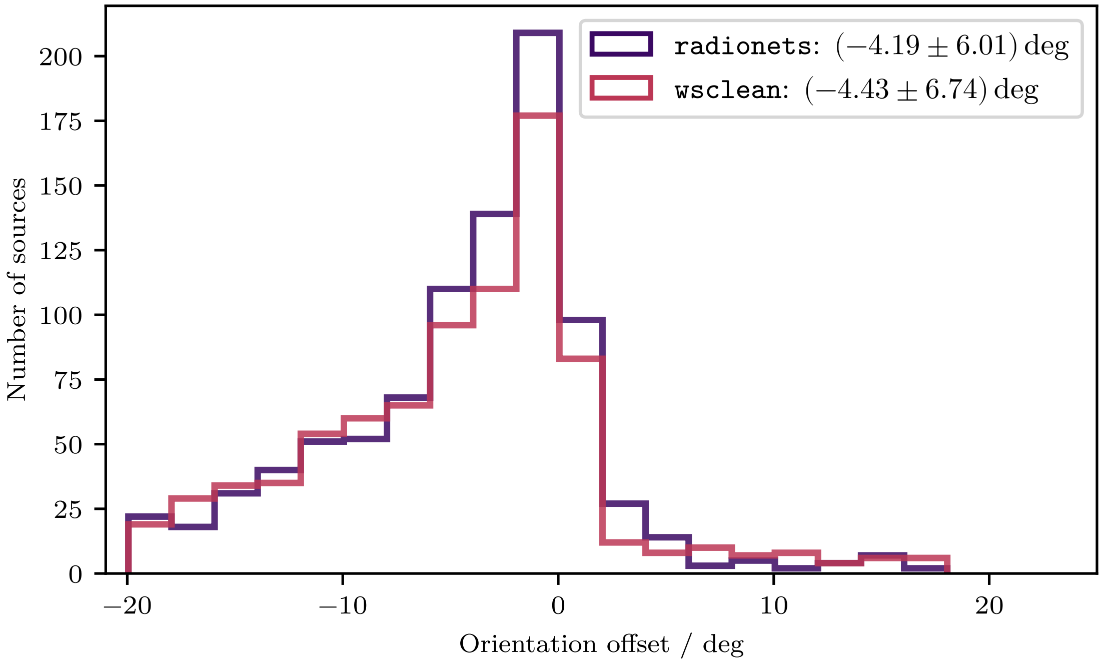


Figure 6.10: Histogrammed source orientation offsets. Results for the deep learning model trained on noiseless data evaluated on the gridded RIME data set (purple) and results generated with WSCLEAN (red). Mean and standard deviation help to compare the distributions.

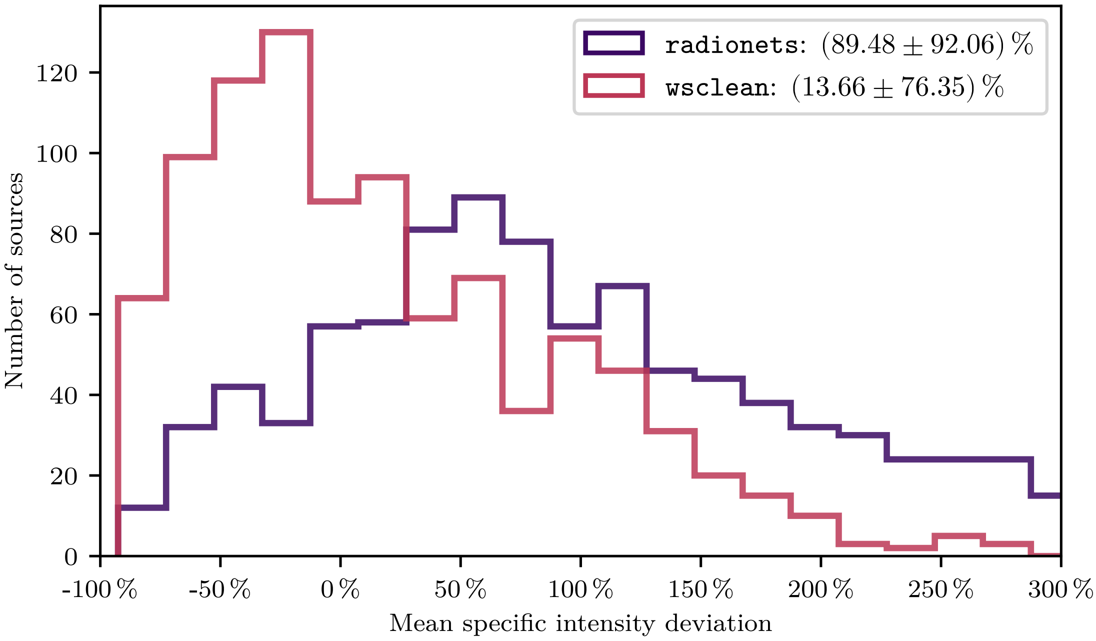


Figure 6.11: Histogrammed mean source brightness deviation for the sources' core components. Results for the deep learning model trained on noiseless data evaluated on the gridded RIME data set (purple) and results generated with WSCLEAN (red). Mean and standard deviation help to compare the distributions.

When analyzing the sources' jet orientation offsets, a systematic negative offset is apparent. Figure 6.10 gives an overview of the histogrammed values, visualizing the larger tail of the distribution towards negative values. As this tail occurs for both reconstruction methods, a connection to the gridding of the data is expected. Further analyses have to be performed to find the exact explanation. The mean and standard deviation is $(-4.19 \pm 6.01)°$ for the `radionets` results and $(-4.43 \pm 6.74)°$ for the WSCLEAN results.

The histogrammed values of the mean specific intensity deviation of the sources' core components are shown in Figure 6.11. Here, different results are apparent for `radionets` and WSCLEAN reconstructions. For WSCLEAN, the mean intensity of a considerable amount of core components is underestimated. The distribution peaks at around $-30\,\%$, and the mean and standard deviation are $(13.66 \pm 76.35)\,\%$. The wide range of values causes this extensive standard deviation. In the case of `radionets`, the mean intensity of the core components is overestimated for most of the test sources. The distribution peaks at around $50\,\%$, and the mean and standard deviation are $(89.48 \pm 92.06)\,\%$. Therefore, the reconstructed intensities are worse for the deep learning approach.

These evaluations do not take into account the occurrence of background structures which are more common in reconstructions with WSCLEAN. Observing random samples has shown that the reconstruction of one-sided jets results in better reconstructions than those of two-sided jets. When considering the presented comparison, one must remember that WSCLEAN is not developed to analyze small images with (64×64) pixels. Furthermore, the deep learning model was not trained on gridded visibility data but on data generated using the sampling mask method. Finally, a deep dive into the world of gridding radio interferometer data is essential in the future.

Table 6.2: Comparison of run-times to reconstruct the data with our trained `radionets` neural network model for (64 × 64) pixels, (128 × 128) pixels, and (256 × 256) pixels images

Image size / pixels	`radionets` run-time / s	Evaluation run-time / ms
64	2.00 ± 0.04	2.0 ± 0.1
128	2.01 ± 0.03	6.4 ± 0.1
256	2.24 ± 0.07	41.7 ± 0.2

Note: The programs are run 100 times on one image. The presented values are the mean run times with standard deviations. `radionets` run-times are pure deep learning model reconstruction times. Evaluation run-times include data loading, model loading, and data saving.

6.4 Execution Times

In the context of deep learning applications, fast execution times are an advantage over classical analysis strategies. A run-time analysis is done for (64 × 64) pixels and (128 × 128) pixels images to classify the execution times of the developed models in this work. Although the training time of 14 h is long compared to conventional analysis methods, the application to data sets can be performed very quickly without adjusting additional parameters in the case of the developed deep learning models. Table 6.2 summarizes the run times for the model trained on noiseless data. The reconstruction was performed 100 times on the same test image. Afterward, the mean run times and standard deviations were computed. The run times are split for using the whole `radionets` framework, where packets and data are loaded, and for the pure application of the deep learning model. It becomes clear that the pure application of the model is swift. The mean evaluation for (64 × 64) pixels images takes (2.0 ± 0.1) ms, while the mean evaluation for (128 × 128) pixels images increase by a factor of three, resulting in (6.4 ± 0.1) ms. Applying the complete framework, without preloaded packages, preloaded data, and saving the results takes around (2.00 ± 0.04) s for both image sizes. For (256 × 256) pixels images, the evaluation time is still short, although the number of pixels increases by a factor of four. Here, the pure evaluation takes (41.7 ± 0.2) ms, and the application of the whole framework requires (2.24 ± 0.07) s. Even for larger image sizes, the short application times of the deep learning model do not significantly affect the execution times. These fast application times make the developed neural network approach ideal for cleaning many similarly obtained data sets like the ones in sky surveys. In order to validate this, the `radionets` framework will be used to reconstruct source images from the Faint Images of the Radio Sky at Twenty-cm (FIRST) survey [6] in a follow-up project.

Advanced Analysis 7

> *"...it is well known that a vital ingredient of success is not knowing that what you're attempting can't be done."*
>
> — Terry Pratchett, *Equal Rites*

When the idea of deep learning-based imaging was new, the feedback after talks and discussions was often pessimistic. People did not believe neural networks were suitable for reconstructing radio interferometer data. Forgetting voices that do not believe in the approach and cooperation with supportive people made it possible to create `radionets`. The previously presented results reveal that a deep learning-based approach can generate clean images of simulated source distributions. This success is the first step towards a straightforward cleaning software that produces reproducible results. With advanced analysis strategies, the possibilities for applying the developed models to actual observation data are tested and further improved. Moreover, the capability to provide uncertainty estimates is examined.

7.1 Mixed Source Types

To further investigate the reconstruction capabilities of the deep learning model, a model is trained on a data set with mixed source types; see Section 3.5. In addition to the extended sources, which are no longer in the center of the images, additional point sources are added. The training is performed on a data set consisting of 50 000 train, 10 000 valid, and 10 000 test maps. More detailed information about the data set is given in Section 3.6. The model is trained for 300 epochs; the corresponding loss curve is displayed in Figure 7.1. During the first 140 epochs, the validation loss spikes. Then a significant drop occurs, indicating that the algorithm left a local minimum. Afterward, the training is smoother, and validation loss spikes are smaller than before.

Figure 7.2 shows an exemplary amplitude and phase reconstruction. While the amplitude reconstruction worked for the complete map, the upper and lower edges are not reconstructed for the phase. Still, frequels that are successfully reconstructed for the phase match the simulated values well. Overall, the differences between reconstruction and simulation are higher than in the previous chapter, which can be connected to the more complex distributions. Figure 7.3 visualizes the corresponding source reconstruction. Both extended and existing Gaussian point sources are visible in the cleaned image.

For an evaluation of more test images, a data set with 10000 test sources are used, and the mean source intensity is evaluated, similar to Section 6.1. Figure 7.4 shows the histogrammed values split into the point-like Gaussians (purple) and extended sources' core components (red). For both source types, the intensity is mainly underestimated. Both distributions peak around $-25\,\%$. No significant overestimation occurs for the extended sources, so the distribution's mean and standard deviation is calculated to $(-22.79 \pm 14.37)\,\%$. In the case of the point-like Gaussian sources, more intensity overestimations take place, leading to a mean and standard deviation of $(-17.75 \pm 22.23)\,\%$.

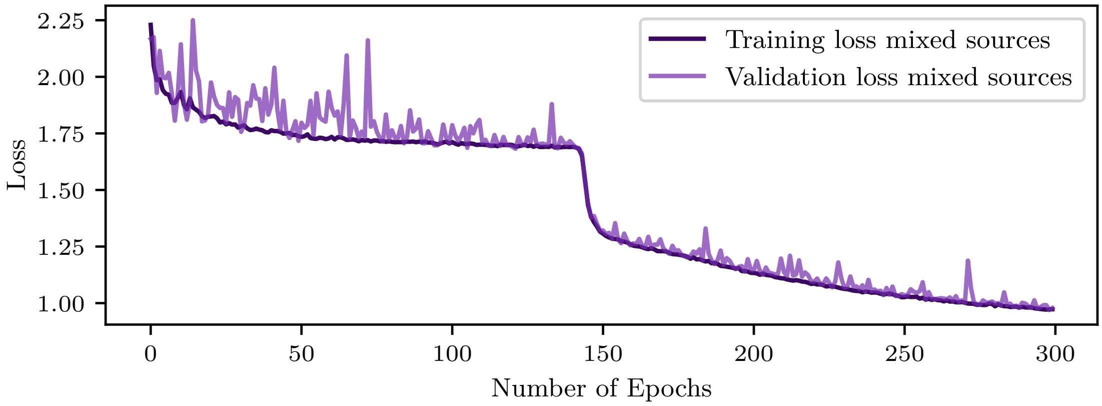


Figure 7.1: Loss curves for the training sessions with mixed source types input data. Loss values against the number of epochs are shown for training and validation separately.

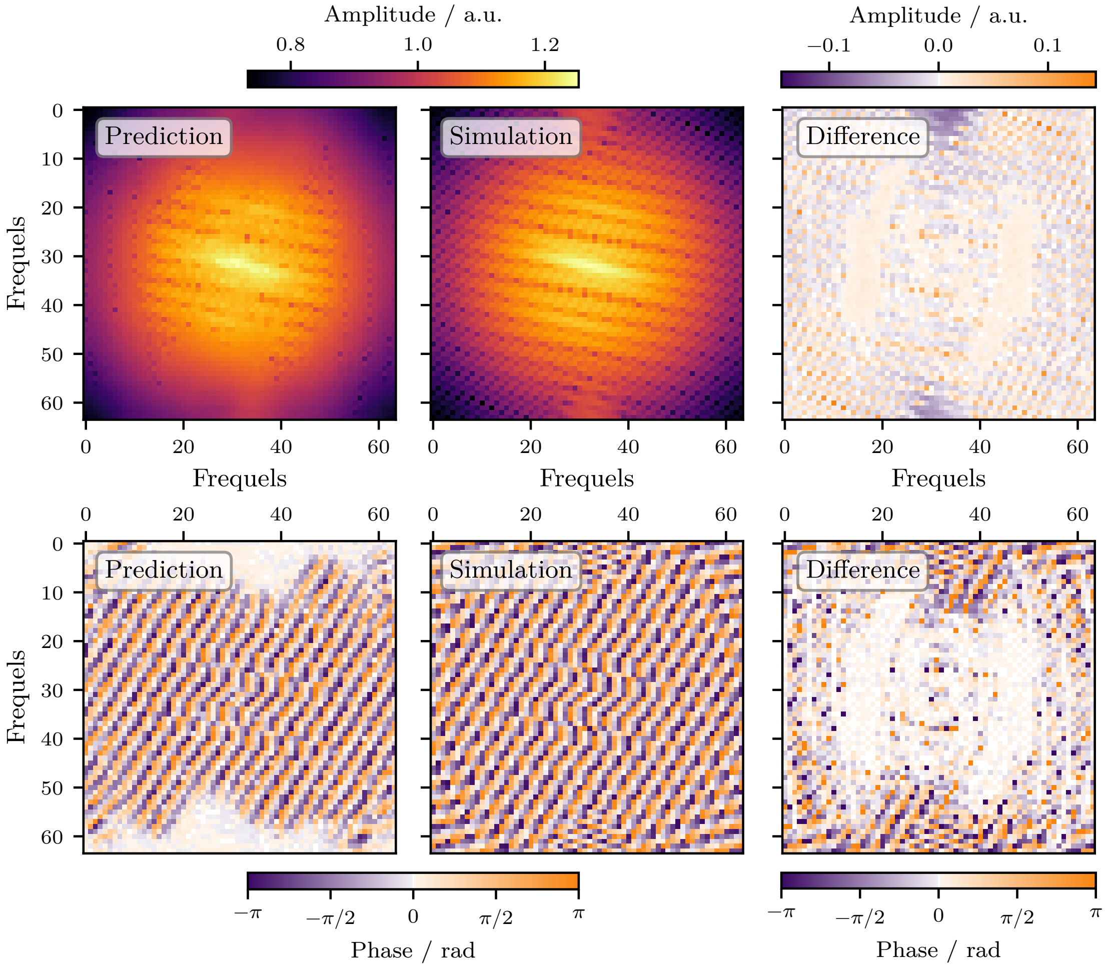


Figure 7.2: Reconstruction of a model trained on mixed source types. Visualization of prediction (left), simulated distribution (middle) and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

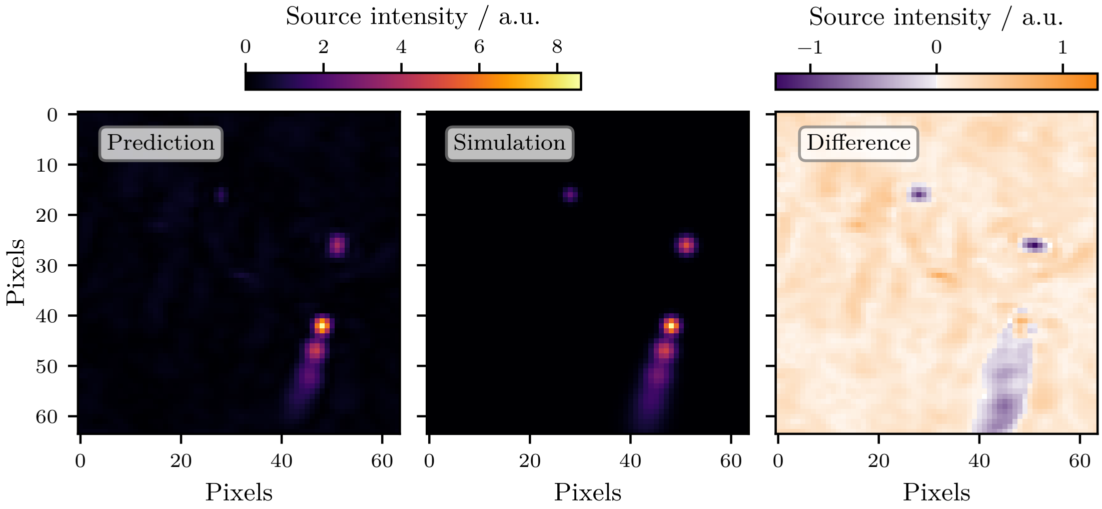


Figure 7.3: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for noisy input data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right).

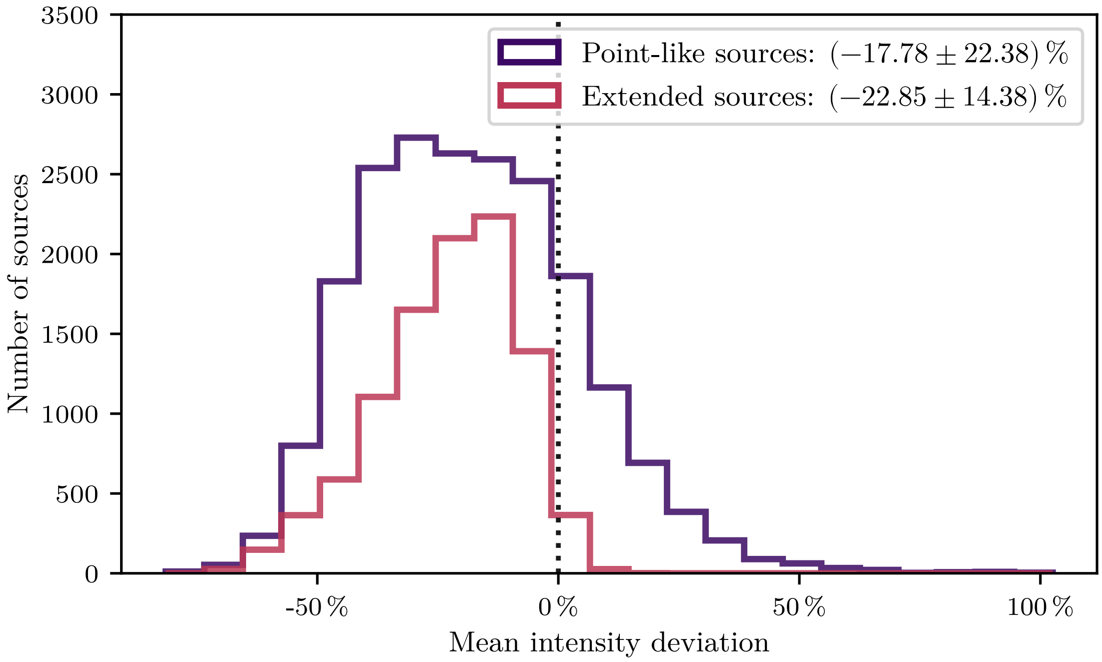


Figure 7.4: Histogrammed mean source intensity deviation for the extended sources' core components (red) and the point-like Gaussians (purple). Mean and standard deviation help to compare the two distributions.

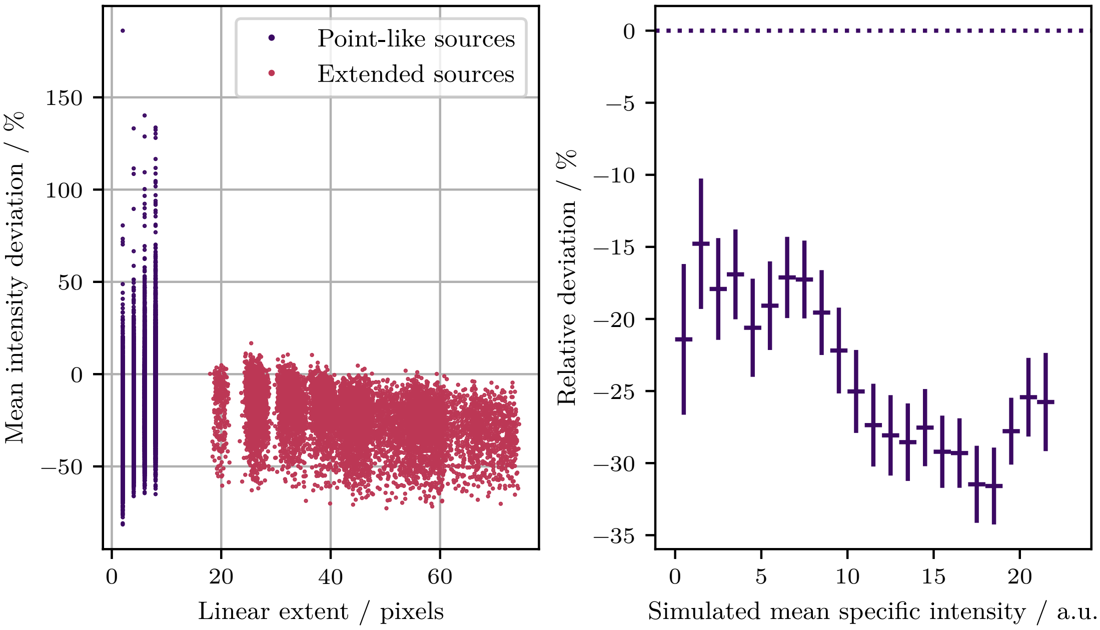


Figure 7.5: The relation between linear extent and mean source intensity deviation (left) split into point-like Gaussian sources (purple) and extended Gaussian sources' core components (red). By visual inspection, no correlation between mean intensity deviation and linear extent is visible. The relative deviation between predicted and simulated mean intensity for different intensity levels (right) visualizes a static increase towards higher intensities. The mean values are shown for the different intensity bins. The bin width is illustrated by the x error bars. The standard deviation of the relative deviation is represented by the y error bars.

Additionally, a connection between the linear expansion of the sources and the mean intensity deviation is investigated. Their standard deviation determines the point sources' linear extent. The distance of the most separated components and their standard deviation for the extended sources is used. Figure 7.5 shows the mean intensity deviation against the linear extend on the left for point-like Gaussian sources (purple) and extended Gaussian sources (red). While the extended Gaussian sources are generally underestimated, the point-like Gaussian sources have more outliers where an overestimation occurs. Still, the majority of point-like Gaussian sources' intensity is underestimated. The visual inspection suggests no correlation between linear extent and mean intensity deviation.

The relative deviations for different intensity bins are examined and presented in Figure 7.5 on the right to get a more detailed insight into the reconstructed intensities. For lower intensities, the difference between predicted and simulated mean intensity is around −18 %. With increasing mean intensity, the relative deviations also increase as expected from Figure 7.4. The standard deviation is in the same order of magnitude for all intensity bins except for the first and the last bin.

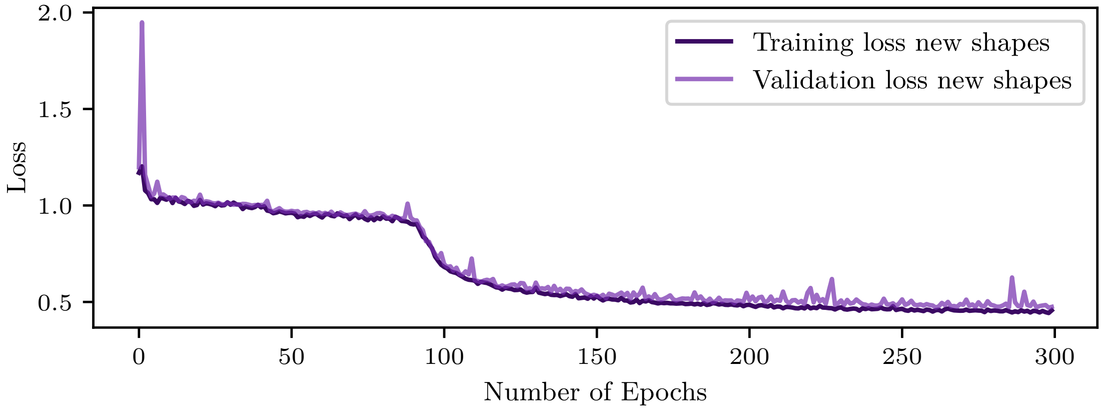


Figure 7.6: Loss curves for the training sessions with new source shapes input data. Loss values against the number of epochs are shown for training and validation separately.

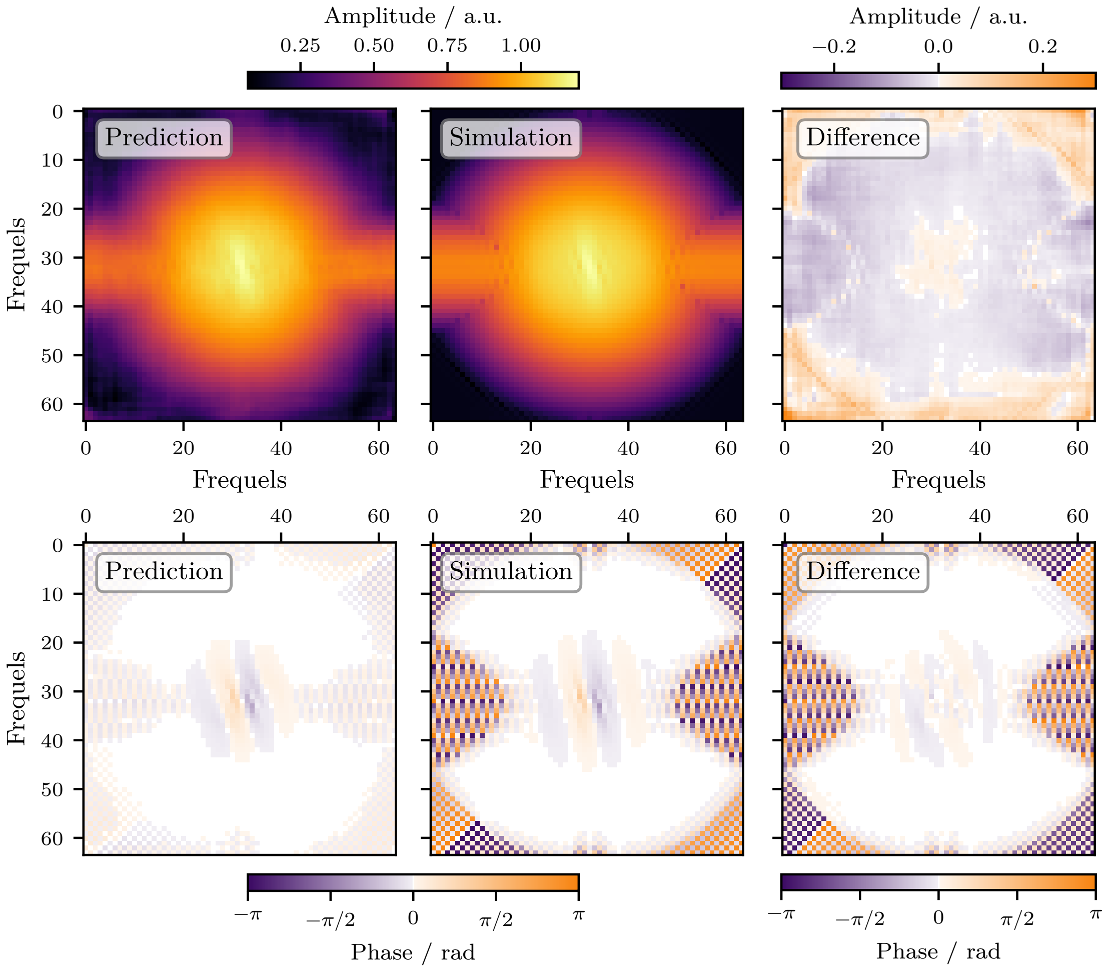


Figure 7.7: Reconstruction of a model trained on new source shapes. Visualization of prediction (left), simulated distribution (middle) and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

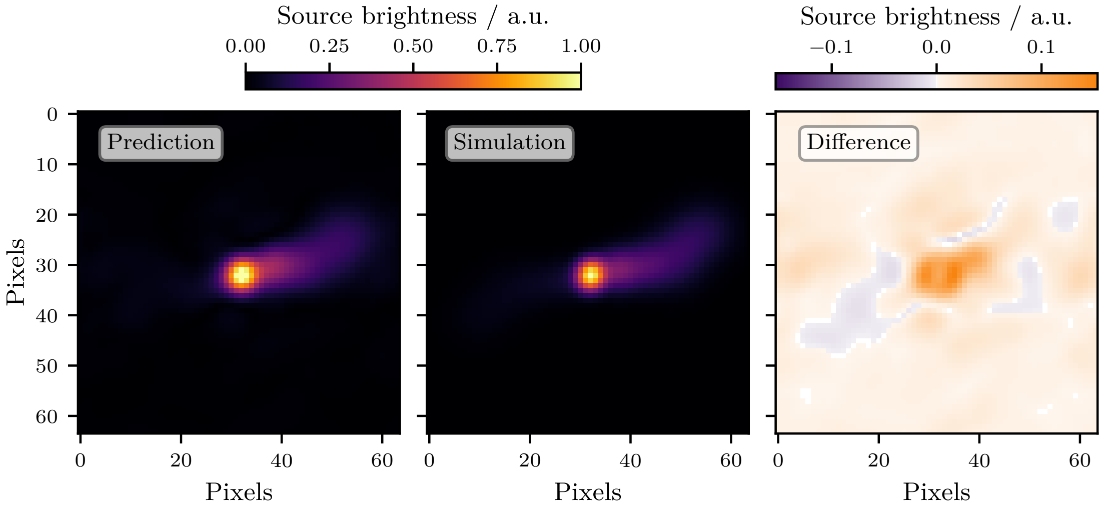


Figure 7.8: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for new source shapes data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right).

7.2 New Source Shapes

The source shapes presented in Section 3.1 allow for testing the deep learning-based imaging approach. Improvements are necessary to create simulated source distributions that better describe reality, which is essential to reconstruct actual radio interferometer observations. For this reason, a new model is trained for 300 epochs on a data set made from the advanced source shapes described in Section 3.5. Figure 7.6 visualizes the loss curve for this training. First, the validation loss shows large spikes following a flat decrease until epoch 90. Here, the model seems to leave a local minimum, and the loss is halved during the next 30 epochs. A smooth decrease with only minor spikes is visible for the rest of the training.

Figure 7.7 illustrates an exemplary amplitude and phase reconstruction for the new shapes data set. Reconstructed and simulated amplitude (top row) show slight deviations. Only at the edges does their difference increase. In the case of the phase (bottom row), the reconstruction quality appears much worse at first look. Only the central values fit the simulation. Values at the edges are underestimated. This underestimation occurs due to a separate weighting of the different phase map areas during training. In the loss calculation, the central circular area with a diameter of 30 frequels is weighted by a factor 10, while for the other frequels only unweighted values are considered. This weighting is necessary as phase distributions for the new source shapes show frequent jumps at the edges. These patterns are hard to learn for the deep learning model at the training start. After some epochs, the central part is predicted well, and the model reconstructs the edge patterns. Without the weighting, the models only learn to reconstruct the amplitude distributions. The phase predictions are just empty.

The reconstructed source distribution is presented in Figure 7.8. The core component and jet structure are reconstructed well. The intensity of the central source area is slightly overestimated. No background structures are visible, indicating that lower phase values at the edges do not restrain the reconstruction quality.

Validating the source area reconstruction on a test set consisting of 1000 samples helps to give a more profound insight into the reconstruction quality of the model. Reconstructed and simulated source areas are compared with the procedure described in Section 6.1. Figure 7.9

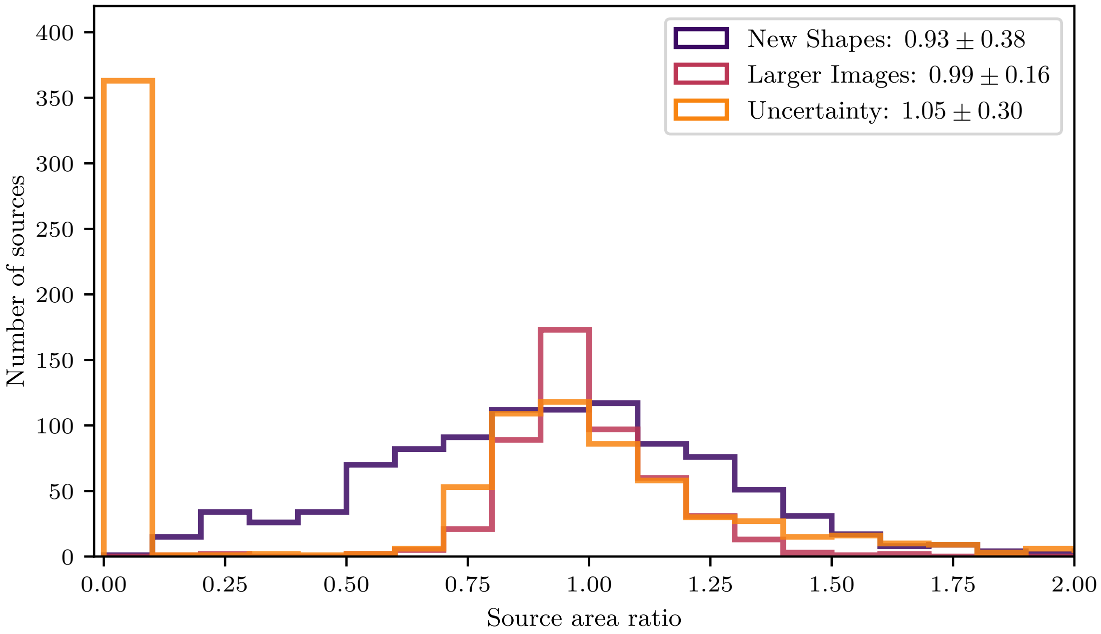


Figure 7.9: Histogrammed source area ratios for the improved data sets. Figure combines results from different sections. Results for the deep learning model trained on new shapes data (purple), larger image sizes data (red), and white noise data reconstructed with the uncertainty architecture (orange) are displayed. Mean and standard deviation help to compare the three distributions.

shows the histogrammed source areas for the new shapes data set (purple). The distribution of values is vast and has a broad peak around one. Still, many ratios are close to the optimal value of one. Area ratios between 0.1 and 1.7 occur, leading to a mean and standard deviation of 0.93 ± 0.38.

The comparison with the results from Chapter 6 shows a decrease in the reconstruction ability. As expected, the improved source shapes are more difficult to reconstruct for the deep learning model. Bent jets create more complex phase distributions. Furthermore, the intensity of the jet components decreases much more towards the outer parts. These aspects make the Fourier space reconstruction more problematic.

7.3 Fine-tuning Pretrained Deep Learning Models

A widespread and well-functioning approach when adjusting deep learning models to a specific problem is fine-tuning an already pre-trained model. In the case of radio interferometry, it is questionable whether a Residual Network (ResNet) trained on RGB-image data can bring a performance boost. Nevertheless, fine-tuning is also a promising approach in the context of the `radionets` models. In the following, the deep learning models trained on white noise data form the basis for creating fine-tuned models capable of reconstructing mixed source types, larger image sizes, and new source shapes data sets.

Independence of Image Sizes

One feature of the chosen architecture is that the size of the input images remains the same throughout the network. The architecture is, therefore, independent of the image size. This independence results in the advantage that a model can be trained on small image sizes for a

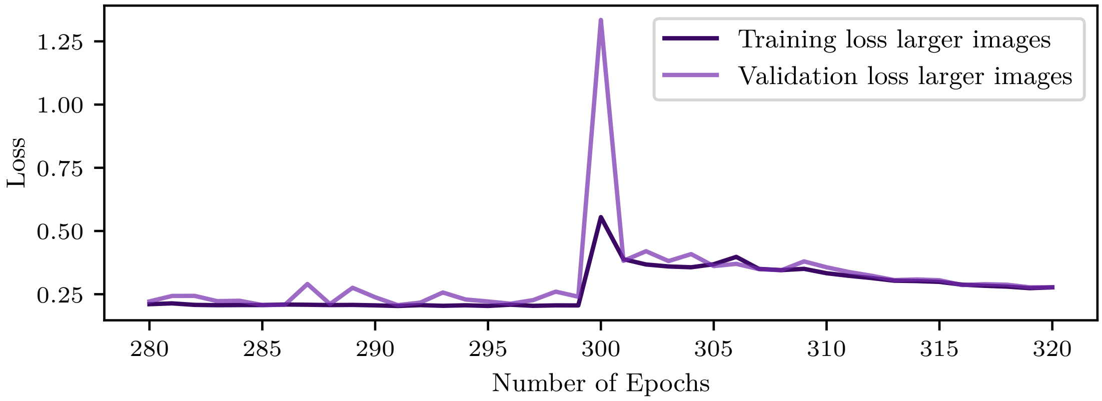


Figure 7.10: Loss curves for the fine-tuning of the model trained on white noise data with larger image size data. Loss values against the number of epochs are shown for training and validation separately.

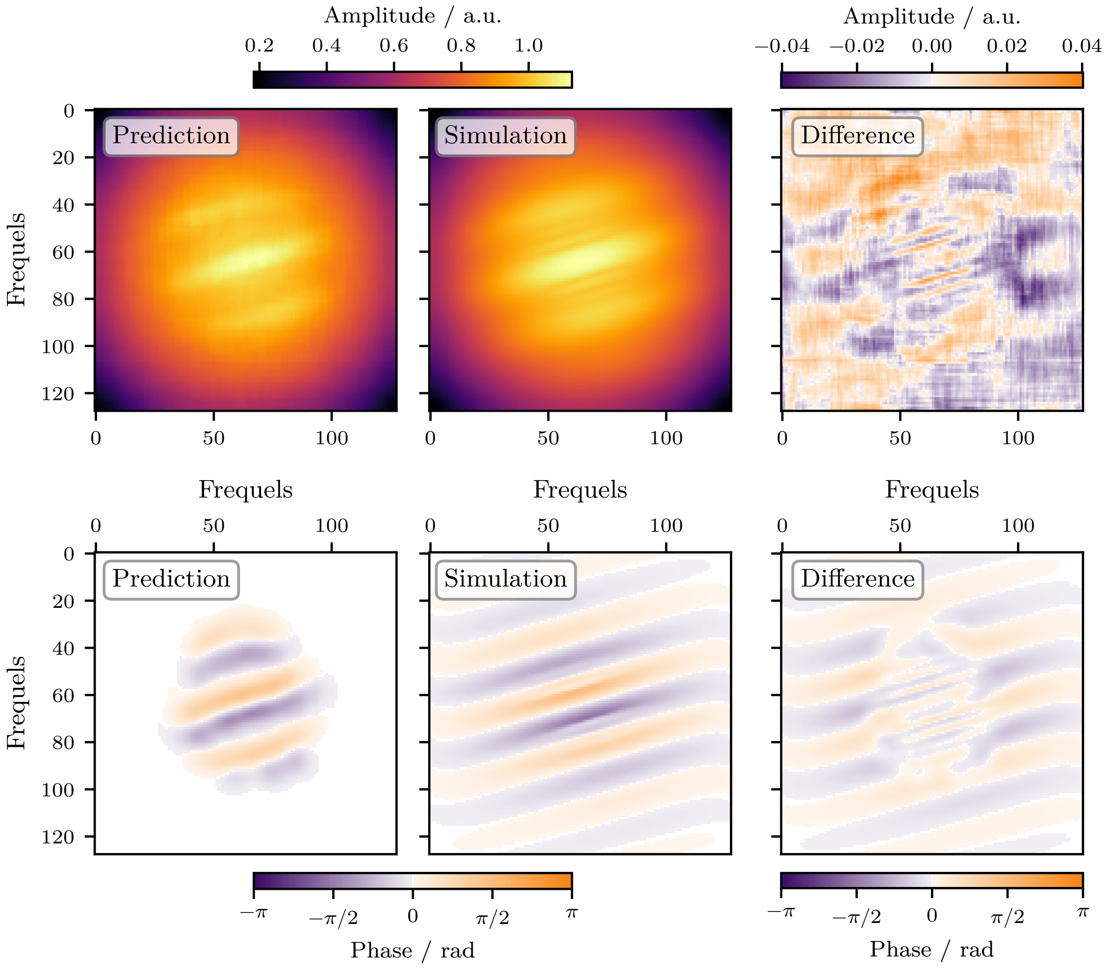


Figure 7.11: Reconstruction of a model trained on mixed source types. Visualization of prediction (left), simulated distribution (middle) and the difference between both (right). Results are shown separately for amplitude distributions (top) and phase distributions (bottom).

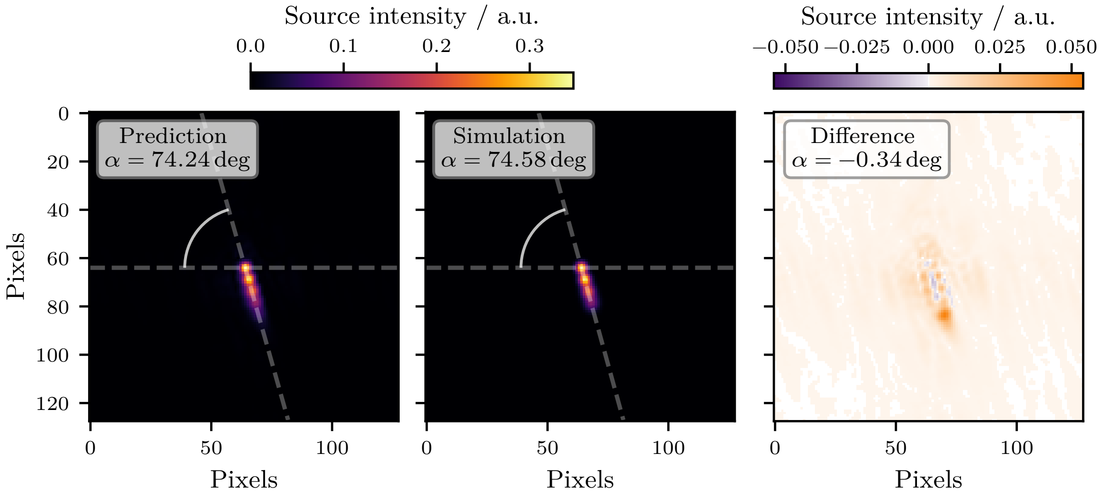


Figure 7.12: Reconstruction of the source distribution generated with the predicted amplitude and phase distributions for noisy input data. Resulting clean image (left), simulated brightness distribution (middle), and the difference between both (right).

long time in a resource-saving manner. A data set consisting of (128 × 128) pixels images is created to test if a model trained on (64 × 64) pixels images can be fine-tuned to reconstruct larger images; see Section 3.6. A larger field of view and a higher resolution is simulated for these sources. The model trained on white noise data is fine-tuned for 20 epochs on 5000 train and 1000 validation maps for 30 min. Figure 7.10 illustrates the loss curve for the fine-tuning. The shown loss for epochs 280 to 300 belong to the training on the small images data set. The following loss increase marks the beginning of the fine-tuning. A smooth decrease without spikes is visible in the next 20 epochs.

Figure 7.11 shows the exemplary reconstruction of larger amplitude and phase maps. The reconstruction quality of the amplitude (top row) is on the same level as the results presented in Section 5.1. The phase (bottom row) is only reconstructed in the central part, which has no significant effect on the reconstructed source distribution presented in Figure 7.12. Like in the reconstructions of the smaller images, no background artifacts are visible, and the individual jet components are reproduced well. Furthermore, the predicted source orientation fits the simulated one with a slight offset of −0.34°.

Without fine-tuning, reconstructed source components show an overestimated specific intensity, and more background artifacts occur. In conclusion, fine-tuning enables the adaption of an existing model for a slightly new task with little effort.

Additionally, the source area ratio introduced in Section 6.1 is calculated on the test data set consisting of 1000 test maps. The histogrammed values are visualized in Figure 7.9. The distribution for the larger images data set (red) has its peak close to the optimal value of one. Most of the area ratios lay between 0.7 and 1.4, leading to a mean and standard deviation of 0.99 ± 0.16. The comparison with the results from Chapter 6 shows that fine-tuning is an excellent way to adjust the model to a larger-image-sizes data set. The results can compete with all previously presented area ratios.

7.4 Uncertainty Estimate

Conventional imaging software cannot give a detailed uncertainty estimate for the cleaned images. As their reconstruction quality depends on the (u, v) coverage and quality of the input data, resulting source distributions are only valid to a certain degree. For a scientific interpretation, an additional uncertainty map is advantageous. The loss function must be adjusted to enable uncertainty estimates with the developed deep-learning model. Therefore, the technique proposed in Abbasi et al. (2021) [53] was an inspiration. When predicting the reconstructed amplitude and phase maps, it is assumed that the values originate from a Gaussian distribution with the parameters μ and σ for every pixel. The negative log-likelihood of a Gaussian distribution is

$$-\mathcal{L} = 2\log(\sigma) + \frac{(x-\mu)^2}{\sigma^2}, \tag{7.1}$$

with the mean μ and the standard deviation σ. This likelihood function forms the basis for the new minimization function of the neural network. The adjustments proposed in [112] were implemented for stability reasons. The final loss function, called β-NLL loss, has the form

$$\mathrm{L} = 0.5 \cdot \left(\log(\sigma^2) + \frac{(y-\mu)^2}{\sigma^2}\right), \tag{7.2}$$

$$\mathrm{Loss} = \mathrm{L} \cdot \mathrm{stop}(\sigma^2)^\beta \quad \text{with} \quad \beta = 0.5. \tag{7.3}$$

Here, no longer the standard deviation, but the variance is estimated. A β of 0.5 is chosen as recommended in [112]. Tests performed by Olivia Locke confirm that this value is also well-suited in the case of Fourier data reconstructions. In addition to the loss function, the architecture must also be adjusted to enable uncertainty estimations. Besides the reconstructed amplitude and phase map, an uncertainty map for amplitude and phase must be predicted. The base architecture introduced in Section 4.2 stays mostly the same. Just the number of ResBlocks is increased from 8 to 16.

An additional uncertainty architecture is added, which takes the predictions plus the initial amplitude and phase maps as input. Figure 7.13 gives an overview of this uncertainty architecture. The central part is the three `ConvBlocks`, which consist of a convolutional layer, a batch norm layer, and a PReLU activation. Settings for the convolutional layer are kept similar to the ones in the SRResNet. Stride and padding keep input and output sizes the same. A new feature here is the increasing number of channels in each convolutional layer. The exact values are summarized in Table 7.1. In the `FinalBlock`, a locally connected layer [119] processes the information of all channels and creates two channels out of it. The convolutions do not take place in one channel as previously described, but they are done on one frequel over all 64 channels. The result of this layer is one map of amplitude uncertainty and one map of phase uncertainty. Finally, an ELU function with an offset of $(1 + 1 \cdot 10^{-7})$ is applied to the values allowing only positive values above $1 \cdot 10^{-7}$ forming the variance maps σ^2. The chosen offset limits the minimum variance to $1 \cdot 10^{-7}$ while it is unlimited upwards. While the prediction of the reconstruction architecture serves as input for the uncertainty architecture, it is additionally bypassed to form the predicted mean values μ.

The improved architecture is trained on the white noise data set for 450 epochs. Training duration does not increase significantly, as convolutions are fast on (64×64) pixels images. Figure 7.14 visualizes the loss values for the training session. As a likelihood function is minimized, negative values are allowed. After a sharp decrease, the loss asymptotically approaches the value zero from the negative direction.

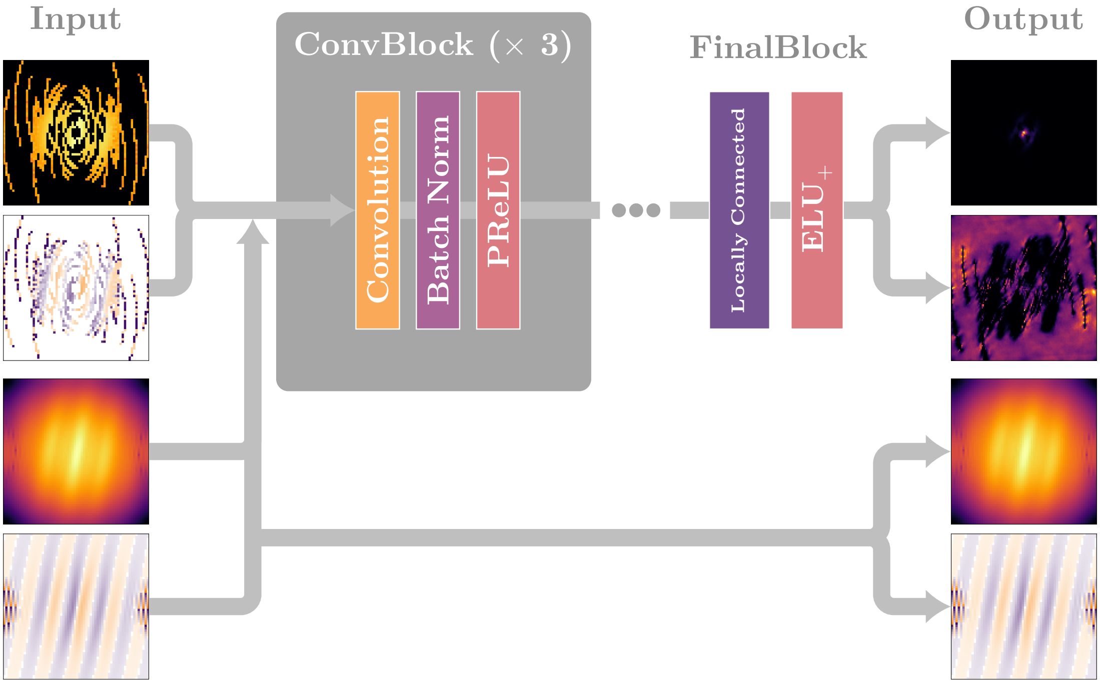


Figure 7.13: Overview of the developed uncertainty architecture. Sampled amplitude and phase maps and reconstructions of the SRResNet-based architecture serve as input (left). The three `ConvBlocks` form the central part of the architecture. After the `FinalBlock`, the predicted amplitude and phase uncertainty and the bypassed reconstructions (right) are the output of the architecture.

Table 7.1: Detailed overview of the different blocks used in the uncertainty architecture

Stage	Layer	C_{in}	C_{out}	Kernel	Stride	Padding
ConvBlock 1	Convolution	4	16	(3×3)	1	1
	Batch Norm	16	16	-	-	-
	PReLU	16	16	-	-	-
ConvBlock 2	Convolution	16	32	(3×3)	1	1
	Batch Norm	32	32	-	-	-
	PReLU	32	32	-	-	-
ConvBlock 3	Convolution	32	64	(3×3)	1	1
	Batch Norm	64	64	-	-	-
	PReLU	64	64	-	-	-
`FinalBlock`	Locally Connected	64	2	(1×1)	1	0
	ELU_+	2	2	-	-	-

Note: For each layer, the number of input C_{in} and output channels C_{out} is given. In the case of convolutional layers, the kernel size, stride, and padding are specified. An offset of $(1 + 1 \cdot 10^{-7})$ is added to the output of the ELU function in the `FinalBlock`.

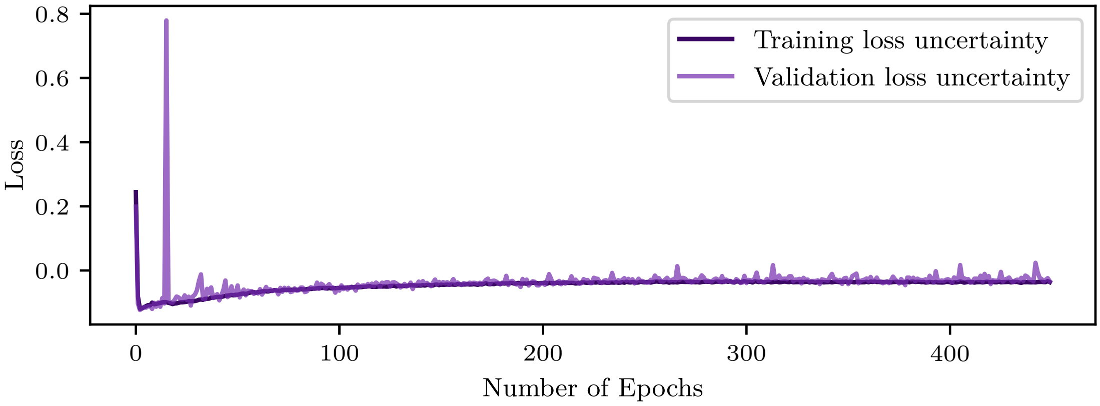


Figure 7.14: Loss curves for the uncertainty training session with white noise input data. Loss values against the number of epochs are shown for training and validation separately. The used β-NLL loss allows negative loss values.

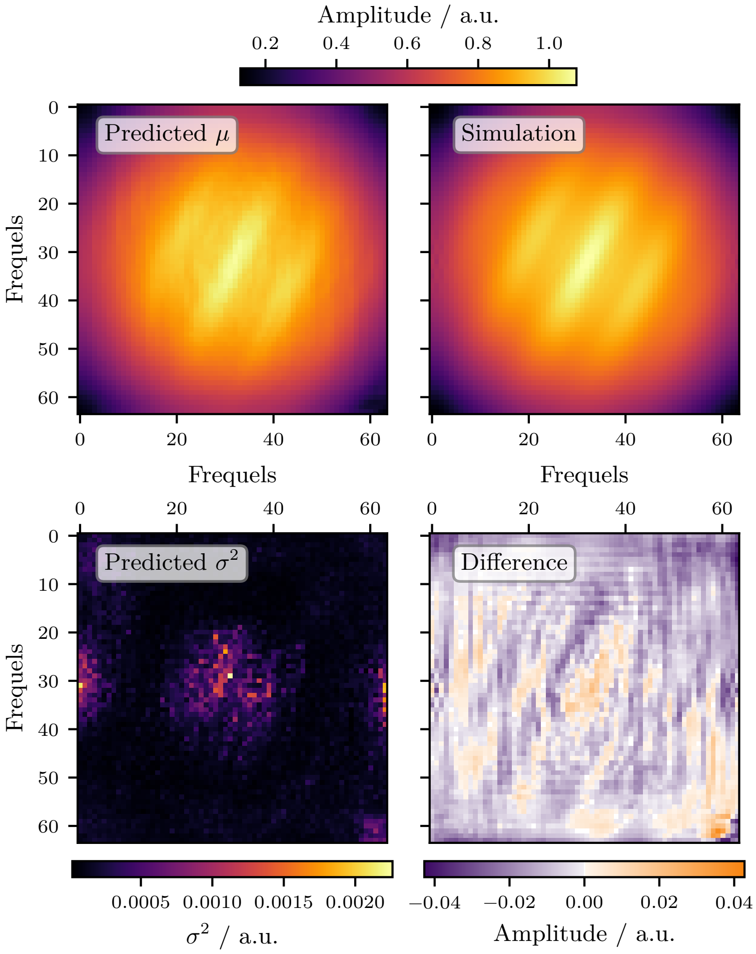


Figure 7.15: Amplitude reconstruction of the uncertainty model trained on white noise data. Visualization of prediction μ (upper left), simulated distribution (upper right), predicted variance σ^2 (lower left), and the difference between prediction and simulation (lower right).

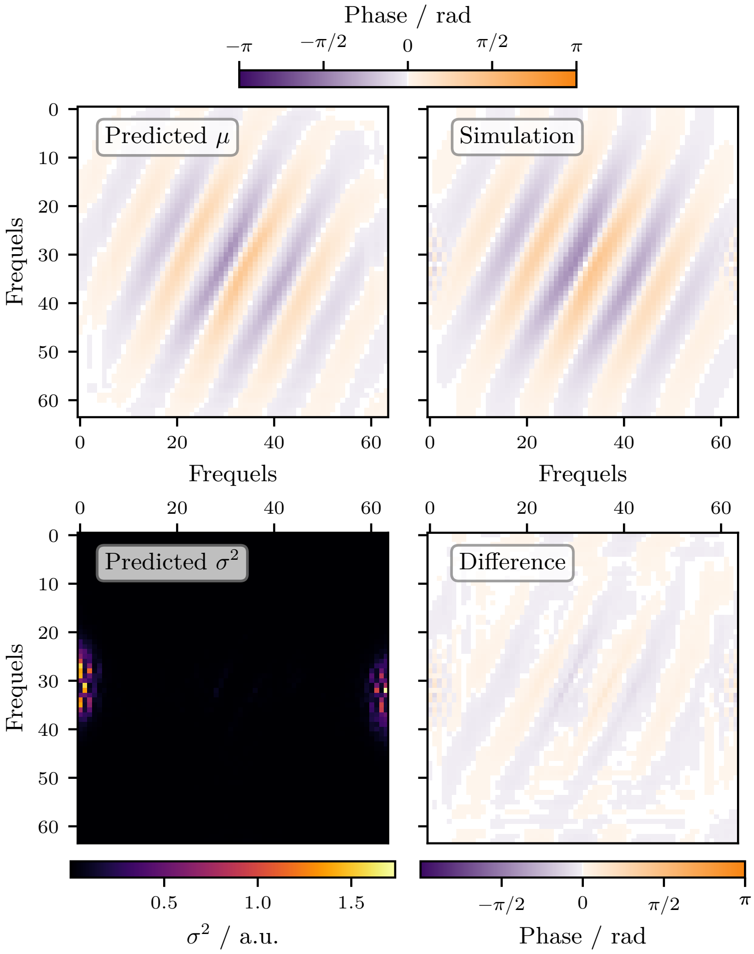


Figure 7.16: Phase reconstruction of the uncertainty model trained on white noise data. Visualization of prediction μ (upper left), simulated distribution (upper right), predicted variance σ^2 (lower left), and the difference between prediction and simulation (lower right).

This section shows amplitude and phase prediction overviews in two separate figures. Figure 7.15 visualizes an example of the predicted mean μ_{amp} (upper left) and the predicted variance σ^2_{amp} (lower left) generated with the uncertainty model for the amplitude. Additionally, the simulated truth (upper right) and the difference between predicted μ_{amp} and simulation (lower right) are shown. The difference reveals a good agreement between prediction and simulation. Predicted variances are small, which matches the minor differences. Variance increases highlight larger deviations in the central or edge region. An exemplary prediction for the phase is illustrated in Figure 7.16. Again, the predicted mean μ_{phase} (upper left), the predicted variance σ^2_{phase} (lower left), the simulated truth (upper right), and the difference between predicted μ_{phase} and simulation (lower right) are shown. Slight offsets between prediction and simulation occur. Still, the overall phase distribution is reconstructed well. The values of the predicted uncertainties are generally higher. These high values occur at the edges of the prediction and only fit the fine-scale structures. At this stage, no correlation of the frequels is taken into account.

An ensemble of reconstructions is received through the mean and variance prediction. No longer a single value is obtained per frequel, but a complete Gaussian distribution. Furthermore, the model generates these distributions for amplitude and phase separately. Figure 7.17 illustrates the

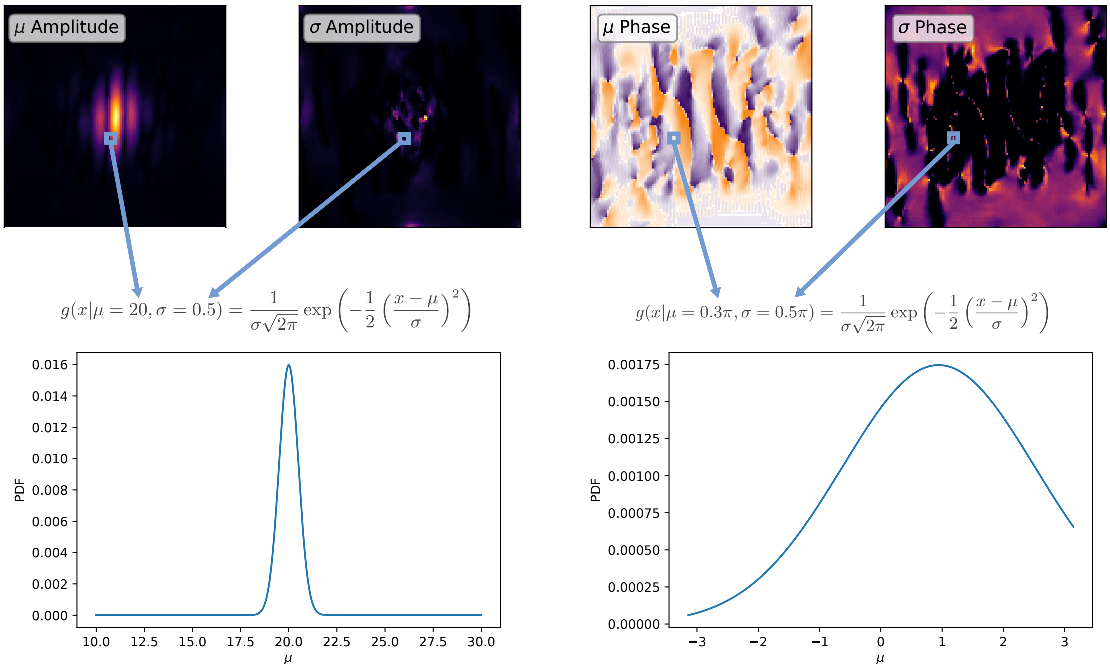


Figure 7.17: Predicted μ_{Amp} (left), σ^2_{Amp} (middle left), μ_{Phase} (middle right), and σ^2_{Phase} (right). Each $\mu - \sigma^2$ pair describes a Gaussian distribution resulting in pixels^2 distributions for both amplitude and phase. Two exemplary Gaussian distributions with ($\mu_{\text{Amp}} = 20$, $\sigma^2_{\text{Amp}} = 0.5$) and ($\mu_{\text{Phase}} = 0.3\pi$, $\sigma^2_{\text{Phase}} = 0.5\pi$) are shown (bottom row).

obtained Gaussian distributions for an amplitude (left) and a phase (right) example. Predicted μ and σ are plugged into the formula. Afterward, values can be sampled from the distribution. The corresponding Probability Density Functions (PDFs) are shown in the lower part of Figure 7.17. Standard normal distributions are truncated to meaningful ranges; $[0, \inf]$ for the amplitude and $[-\pi, \pi]$ for the phase. The PDFs allow the sampling of different amplitude and phase distributions. This work generates 10 000 versions for each. Then, the amplitude and phase samples are used to create 10 000 versions of the cleaned image by applying the inverse Fourier transform. The pixel-wise mean is calculated to get the most probable source distribution. The pixel-wise standard deviation allows for given an uncertainty map. The source distribution sampling is illustrated in Figure 7.18.

Figure 7.19 visualizes the reconstruction in image space. Here, the prediction (upper left) corresponds to the mean source distribution generated from the sampled versions. Again, the simulated source distribution (upper right) and the difference between prediction and simulation (lower right) are shown. The uncertainty map (lower left) is the standard deviation derived from the sampled source versions. Prediction and simulation are in good agreement. While the source's core component is slightly overestimated, an underestimation occurs for the jet intensity. Comparing the uncertainty map and difference reveals that the structures match well. Background structures in the difference are also visible inside the uncertainty map. Furthermore, the uncertainty is highest at the source region, as expected. The uncertainty values cover the difference between prediction and simulation. Still, the uncertainties are conservative, as they exceed the difference by a factor of five. For this example, the uncertainty approach works very well and enables a better interpretation of the deep learning model's results.

The source area ratio introduced in Section 6.1 is calculated on the test data set consisting of

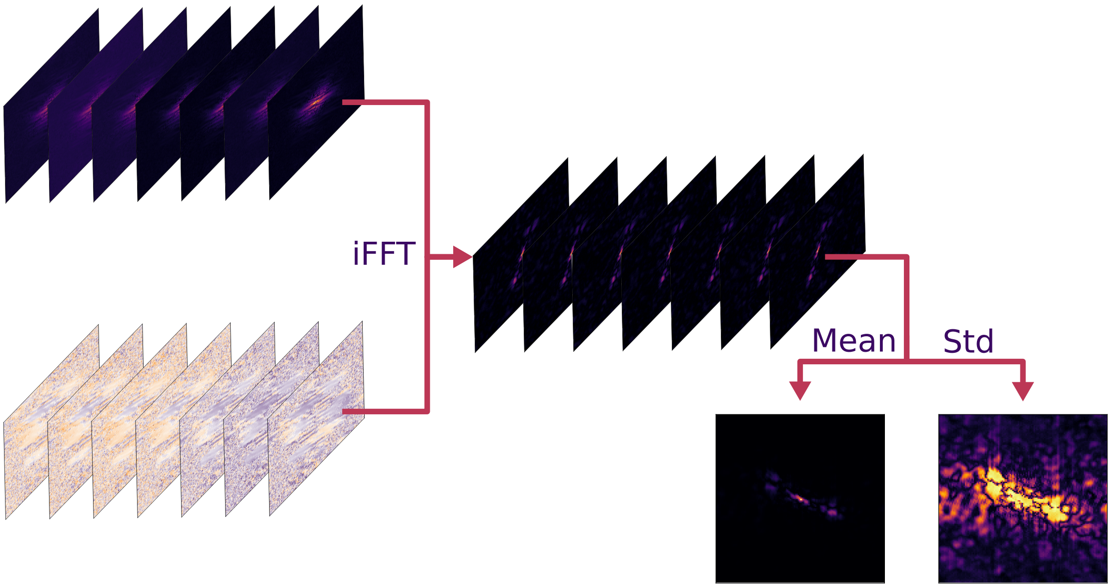


Figure 7.18: Visualization of the uncertainty sampling. Gaussian distributions displayed in Figure 7.17 are used to create n versions of the amplitude and phase distributions. Each pixel has its Gaussian distribution from which values are sampled. The inverse Fourier transform is applied to the n amplitude and phase pairs, creating n versions of the reconstructed source distribution. The most probable reconstruction and the corresponding uncertainty map are created by calculating the pixel-wise mean and standard deviation.

1000 test maps to evaluate the uncertainty approach in more detail. Figure 7.9 visualizes the histogrammed area ratios for the mean prediction from the sampled source versions (orange). While the distribution shows a peak at one, the reconstruction failed for around 40 % of the sources. These failed reconstructions generate the peak at zero. When only the successfully reconstructed source distributions are taken into account, the mean and standard deviation of the distribution is calculated to 1.03 ± 0.30. The comparison with the results from Chapter 6 shows that the area ratios generated with the uncertainty model are in good agreement with the ones generated for the white noise model when the failed reconstructions are excluded. When the source reconstruction in local space is calculated with the predicted mean μ, no reconstruction problems occur. These problems originate from large uncertainty predictions σ^2, which produce edge effects in the sampling process. Therefore, the results are highly correlated with the predicted σ^2.

For a further investigation of the predicted uncertainties, the assumption that they originate from a Gaussian distribution is tested. First, the difference between sampled prediction and simulated truth is calculated. Afterward, the difference is divided by the predicted uncertainty σ. Then, a Gaussian distribution is fit to the resulting values. This procedure is done separately for amplitude, phase, and source distribution. Figure 7.20 visualizes the histogrammed values and the fitted Gaussians. In the case of Gaussian uncertainties, a standard normal distribution is expected. For the amplitude (left), the standard deviation fits the optimal value of one. The mean of 0.37 unveils a shift to positive values. Therefore, the amplitude prediction is still improvable. In the case of the phase (middle), the result is in good agreement with a standard normal distribution. The resulting distribution has a mean and standard deviation of -0.01 ± 0.95. Predicted amplitude and phase uncertainties can be transformed into a source brightness uncertainty as the Fourier transform is linear. As for the source area evaluation, the

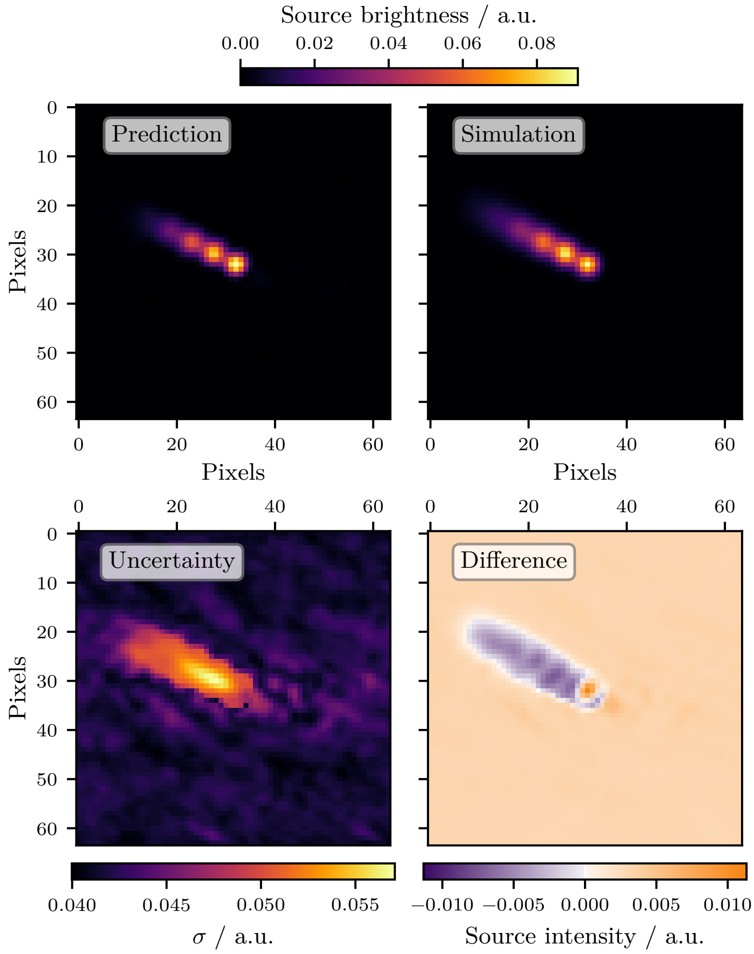


Figure 7.19: Reconstruction and uncertainty map of an exemplary source distribution from the white noise data set. While the shown reconstruction (upper left) is generated by calculating the mean of the sampled versions, their standard deviation gives the uncertainty map (lower left). The difference (lower right) between prediction and simulation (upper right) reveals a good agreement with slight deviations along the jet.

histogrammed values for the source brightness reveal that there is still a problem for a large fraction of sources. The shift towards positive values from the amplitude is again evident. The fitted Gaussian distribution has a mean of 0.28. The standard deviation of 0.72 is also a bit off from the optimal value of one. The result shown in Figure 7.19 confirms that successful uncertainty predictions work very well. Future works will focus on a lower reconstruction failure rate.

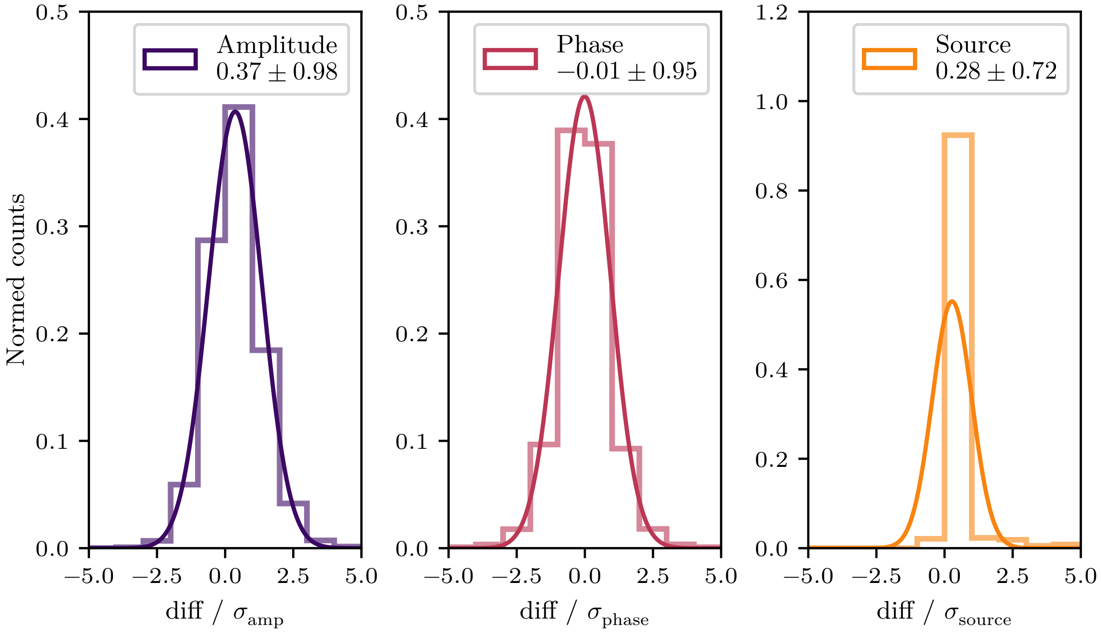


Figure 7.20: Normed counts of the difference between mean prediction and simulation divided by the predicted variance, testing the assumption that predicted mean and uncertainty originate from a Gaussian distribution. Fitting a Gaussian distribution to the normed counts results in parameters of a standard normal distribution in the best case. Fitted mean for amplitude and source distributions are shifted to positive values. The standard deviation for the source distribution is underestimated.

Reconstructing VLBA Observations 8

“Coming back to where you started is not the same as never leaving.”
— Terry Pratchett, *A Hat Full of Sky*

The last parts of this work have shown that cleaning radio interferometric data is possible using neural network models. The subsequent step is to apply these models to actual observational data. Since the previously described simulations use basic source models and only partial effects of telescope responses are considered, more sophisticated simulations are needed to analyze data measured by radio interferometers. Therefore, coming back to the start is essential, but it is not a beginning from scratch. The knowledge from the previous chapters will help further improve the complete analysis chain. Noise effects from telescope responses can be accounted for when calculating complex visibilities with the help of the RIME formalism [116]. Combining all improvements enables the creation of a data set suitable for training neural network models applicable to MOJAVE data.

8.1 MOJAVE Data Archive Statistics

In Section 3.1, simulations created extended radio sources by appending several Gaussian components. Source shapes observed inside the MOJAVE program serve as a basis for the source morphologies, but they lack a detailed physical justification concerning their exact simulation parameters. Loosely chosen values for the sources' linear extend, the field of view, and the peak flux of the sources served to create a primary toy data set to perform a feasibility study of the analysis approach. This chapter's detailed analysis of the MOJAVE data archive will help create a more sophisticated and physically motivated data set.

At the moment of this analysis, the MOJAVE archive contains 9196 observations of jetted radio sources. An adaption of the simulations to real-world characteristics is mandatory so that the neural network can reconstruct actual observations of the VLBA. For the simulations in this work, the parameters visualized in Figure 8.1 are of particular importance.

One aim is to adjust the simulated sources' flux densities to the MOJAVE program's observations. Therefore, examining the peak flux densities of the cleaned images from the MOJAVE archive helps to find the underlying distribution. First, the flux densities must be converted from Jy per beam to Jy by multiplying all pixels by the factor $^{\text{pixel size}}/_{\text{beam area}}$. Figure 8.1a shows the resulting peak flux densities.

In addition to the peak flux densities, describing the flux density decrease along the jet axis is also necessary to quantify the sources' morphologies. By fitting an exponential function to the flux density distribution within the source structure, defined as

$$f(x) = a_0 + a \cdot x^k, \tag{8.1}$$

the determination of the power-law index k helps quantify this decrease. Figure 8.1b shows the distribution of the resulting power-law indices.

Additionally, the simulated data set's source sizes must be close to those from the MOJAVE program. For this reason, the sources' linear extent assesses their sizes by placing an axis through their center and jet. Afterward, the distance the sources extend along this axis is determined. Figure 8.1c shows that many more compact sources with extensions smaller than 20 mas are present in the data set, which is reasonable since the MOJAVE program mainly analyzes the innermost structures of AGN.

Another essential aspect of the simulated images is the selected Field of View (FoV). The used FoV does not correspond to the actual telescope's FoV but restricts the sky region of interest. In the case of the MOJAVE program, the values 52 mas, 102 mas, or 202 mas are used for the field. Choosing too small FoVs excludes data and will affect the source distribution's reconstruction quality.

Besides the source characteristics, the observation parameters also play an essential role. When examing the distribution of the source positions, it becomes clear that all possible values between 0° and 360° occur for the right ascension. On the other hand, a restriction of the declination is visible caused by the VLBA's localization in North America. The occurring parameters for the declination range from −30° to 90°.

Evaluating the number of visibilities per observation ensures that the simulated measurement times match those in the MOJAVE program. The distribution shown in Figure 8.1d visualizes the wide range of visibilities per observation. The first peak comes from short snapshot observations, and the second smeared-out peak comes from regular observations. Here the number of visibilities peaks around 7000 visibilities per observation. Considering the different simultaneously used frequency channels, 4 and 8 are the prioritized number of channels. Recently, the favored observation mode uses 4 frequency channels.

In Table 8.1, the characteristics are summarized again quantitatively. The displayed values will help draw source parameters from probability distributions to represent parameter ranges comparable to actual observations.

Table 8.1: Quantification of the source and observation characteristics derived from the observations in the MOJAVE data archive.

	Mean	Std	Median	Minimum	Maximum
Peak Flux / mJy	33.6	47.4	18.0	0.1	594.6
Power law index	−0.2	0.2	−0.1	−0.8	0.0
Linear Extend / mas	43.2	62.6	10.2	1.3	382.2
Number Visibilities	8487.6	9162.8	6718.5	401.0	158 220.0

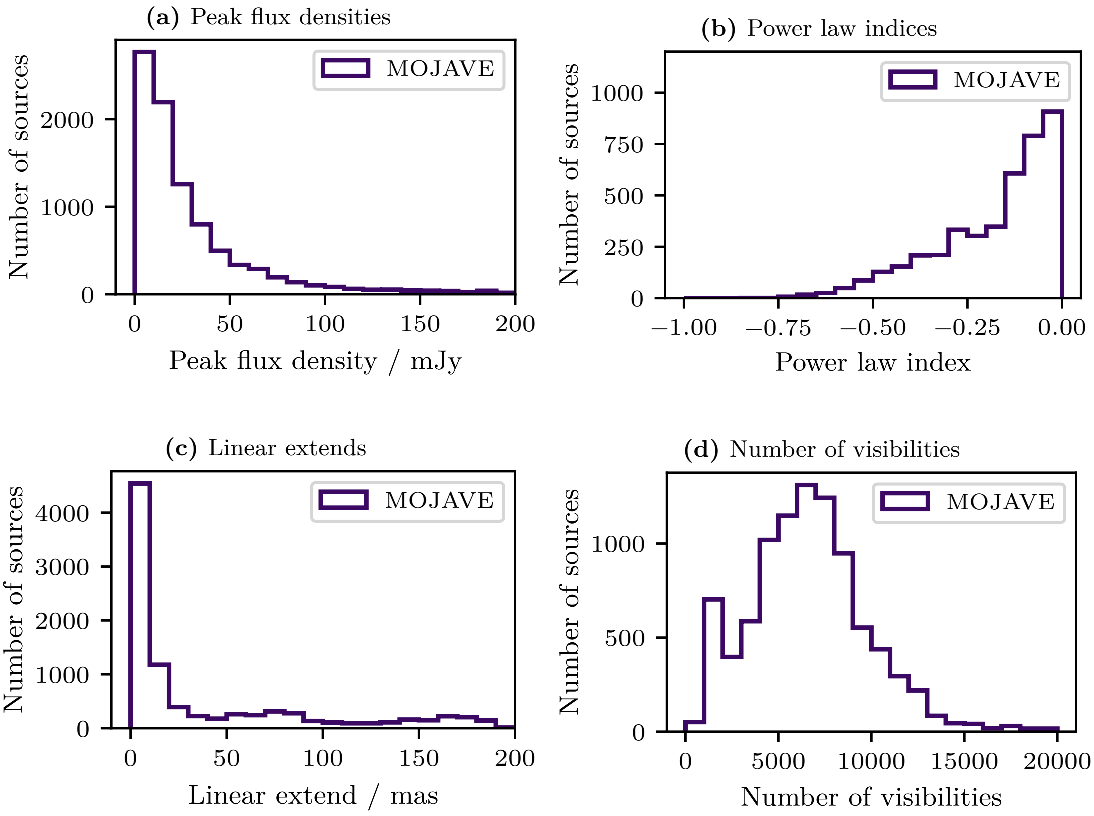


Figure 8.1: Source and observation characteristics of the MOJAVE data archive. Distribution of peak flux densities, power-law indices, linear extends, and the number of visibilities are shown. The visualized source parameters are of particular importance for the simulations.

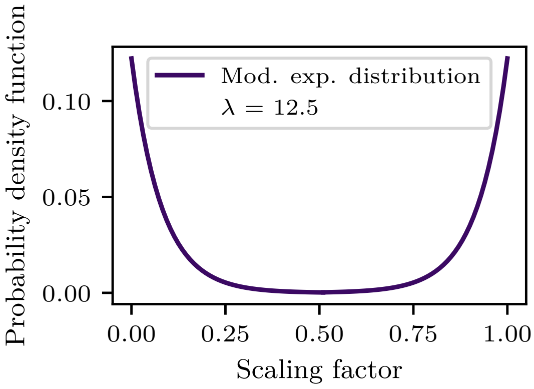


Figure 8.2: Probability function to draw scaling factors for the counter-jet flux densities. With the substantial increases around 0 and 1, one-sided and two-sided jets are favored.

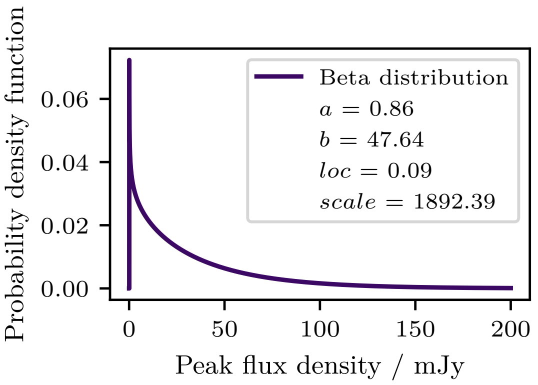


Figure 8.3: Beta distribution to draw peak flux densities for the radio galaxy simulations. The MOJAVE data archive provided the data to adjust the parameters a, b, loc, and $scale$.

8.2 Simulation of VLBA Data

The information obtained from the MOJAVE data archive now enables the simulation of realistic source distributions utilizing the `radiosim` package. The `radiosim` project was created together with Arne Poggenpohl, in the context of his master's thesis (in prep.), for generating realistic images of radio galaxies with `Python`. This package improves the simulation chain presented in Section 3.1 by enhancing it with different features. The simulated radio galaxies still consist of two-dimensional Gaussian components. The improvements are that now the size of the respective component is varied by different σ_x and σ_y. Additionally, the distance between the components is changed so that the expansion of the sources changes noticeably. Another point is that jet structures are now bent. These bent morphologies simulate the redirection of jets that can occur in reality, for example, by interacting with interstellar matter [10]. Simulation routines in Section 3.1 already create single- and double-sided sources, so the orientation in space was considered to some extent. Nevertheless, the finished source distribution corresponded only to lateral two-dimensional views. Here, the improvements allow for simulating the orientation in three-dimensional space. Therefore, the sources' jets can point towards or away from the observer. Using the two-dimensional projection for the source images results in overlaying Gaussian components.

Since all sources have two jets and the counterpart sometimes cannot be seen only because of the orientation and the resulting relativistic boosting, all source images are created double-sided. In some cases, the flux density of the counter-jet is strongly scaled-down, simulating this relativistic boosting. The scaling factors are drawn from the distribution shown in Figure 8.2, which is defined as:

$$f(x, \lambda = 12.5) = \begin{cases} \lambda \cdot \exp(-\lambda \cdot x) & 0 \leq x \leq 0.5 \\ \lambda \cdot \exp(-\lambda \cdot [1 - x]) & 0.5 < x \leq 1 \end{cases} \tag{8.2}$$

This distribution has the advantage of generating mainly one-sided and two-sided jets. Only in exceptional cases are sources simulated where the counter-jet has between 20 % to 80 % of the flux density of the jet.

The height of the flux densities is generally simulated based on the MOJAVE sample. For this purpose, a beta function,

$$\beta(x, a, b) = \frac{\Gamma(a + b) x^{a-1} (1 - x)^{b-1}}{\Gamma(a)\Gamma(b)}, \tag{8.3}$$

is fitted to the distribution shown in Figure 8.1a resulting in the probability density function displayed in Figure 8.3 with the parameters:

$$\begin{aligned} a &= 0.86 \\ b &= 47.64 \\ loc &= 0.09 \\ scale &= 1892.39. \end{aligned}$$

Here, Γ stands for the gamma function. With *loc* and *scale* resulting values are shifted to $y = {}^{(\beta(x,a,b) - loc)}/_{scale}$. The drawn values allow the flux density scaling of the simulated radio galaxies.

Furthermore, performing a logarithmic decrease with different factors mimics a realistic flux density decrease along the jet axis. The MOJAVE archive factors shown in Figure 8.1b form a basis for the used distribution. The source's linear extent is the third crucial parameter to reproduce the source morphologies as realistically as possible. Using between 3 and 6 Gaussian components

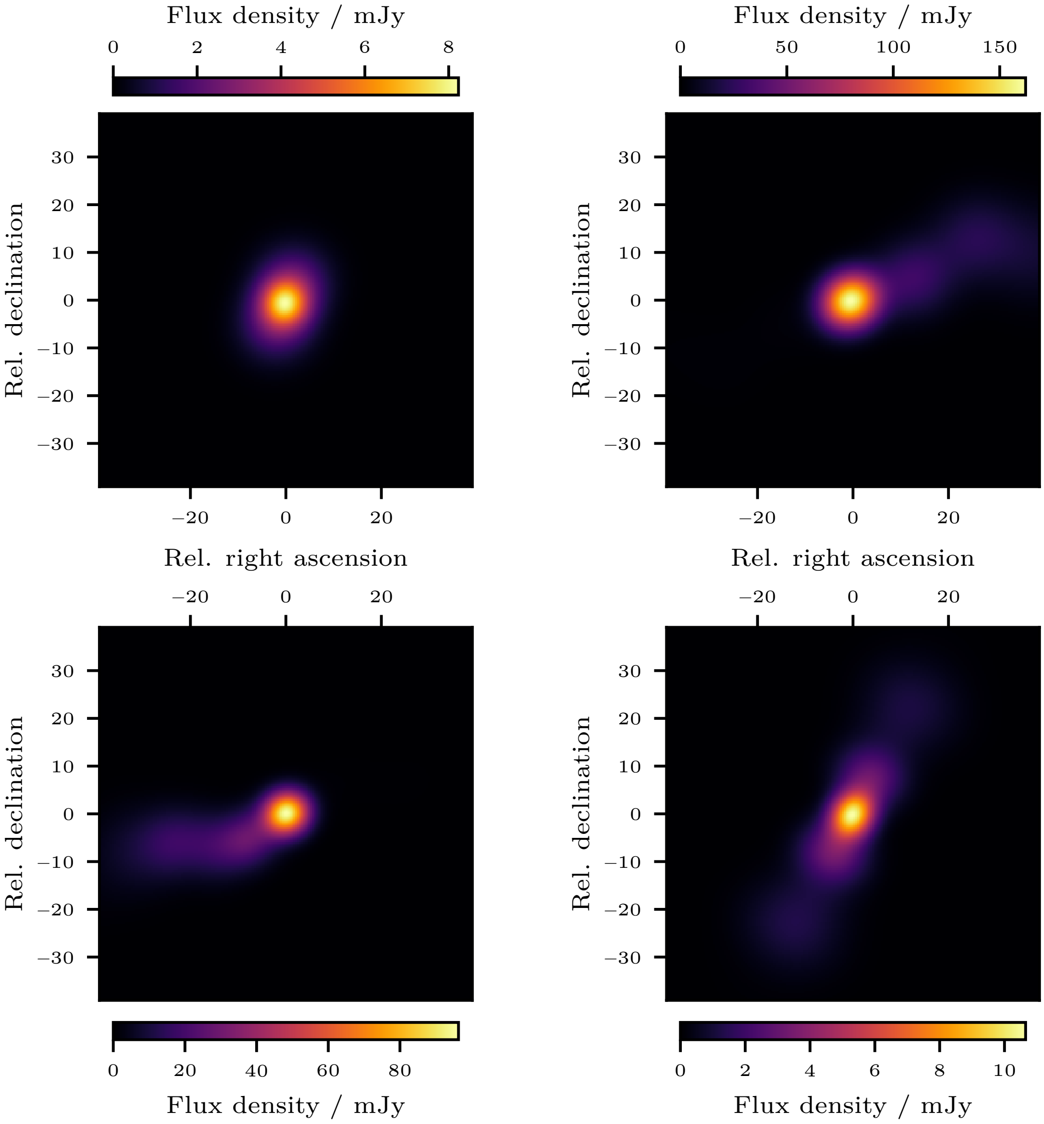


Figure 8.4: Overview of radio galaxies created with `radiosim` [108] using simulation parameters derived from the source characteristics of the MOJAVE data archive.

for jet and counter jet and adjusting their size ensures a good match of the distribution in the simulated data set and the one shown in Figure 8.1c. Since all source orientations occur inside the MOJAVE archive, drawing directions from an equal distribution ranging from 0° to 360° is required. With these characteristics, a data set of extended radio galaxies is created, including 12 000 training images, 1000 validation images, and 1000 test images. Figure 8.4 gives an overview of example source distributions. Here, it becomes clear that the source morphologies are similar to the MOJAVE measurement shown in Figure 8.5. The default simulation configuration is illustrated in Appendix 5.

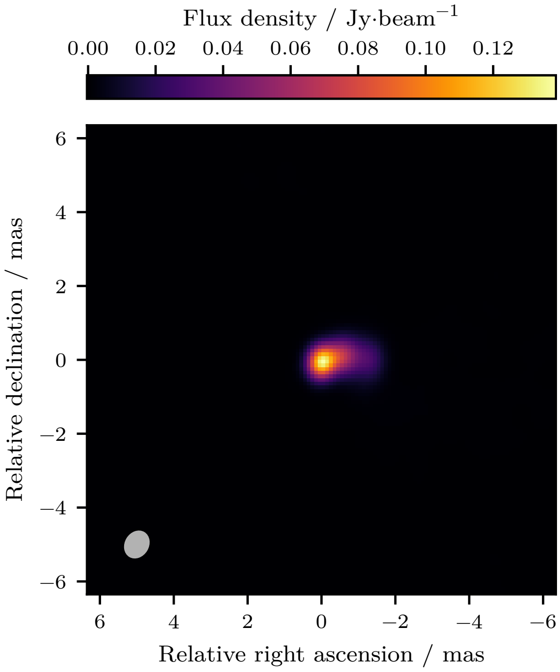


Figure 8.5: VLBA observation of 7C 1055+5644 in the context of the MOJAVE 15 GHz program taken on August 4th, 2019. The white ellipse in the lower left represents the beam of the observation. [75]

8.3 Simulation of Visibilities

A good simulation of telescope characteristics and observational properties is essential to represent a radio interferometer's influence on data as described in Section 1.4. Since the simulation routines introduced in Section 3.2 only mimic observations by applying sampling masks, I developed the `Python` package `pyvisgen` [30] together with Stefan Fröse to calculate complex visibilities directly. Using the RIME formalism as the basis for the calculations makes `pyvisgen` superior to the previously used simulation routine. Here, Jones matrices express the different effects on the data as described in Section 3.4. The advanced possibilities make the RIME formalism a powerful tool for describing radiation propagation starting at the source until the data point arrives from the correlator. The RIME formalism is equivalent to matrix formalism in optics used to describe light paths propagating through optical components, e.g., lenses.

Only the direction-independent effects are simulated in this work since it is standard to perform direction-dependent calibrations before passing the data into imaging software. Therefore, the jones matrices used are K-Jones, the Fourier or phase-delay kernel, defined as

$$\mathbf{K}(l,m) = \exp\left(-2\pi \cdot i \cdot (ul + vm)\right), \tag{8.4}$$

and E-Jones, which considers the effects of the telescope responses, designated as

$$\mathbf{E}(l,m) = \text{jinc}\left(\frac{2\pi}{\lambda_\text{obs}} d \cdot \theta_{lm}\right). \tag{8.5}$$

Here, jinc is the jinc function, which is represented by

$$\text{jinc}(x) = \frac{J_1(x)}{x}, \tag{8.6}$$

where J_1 corresponds to the Bessel function of the first kind. Furthermore, d is the telescope's diameter, and θ_{lm} describes the angular distance to the source position. Since polarization effects are currently omitted for simplification, only the Jones parameter I is calculated. The Jones chain to calculate the visibility of the baseline pair pq results in

$$\mathbf{V}_\text{pq} = \sum_l \sum_m \mathbf{E}_\text{p}(l,m)\mathbf{K}_\text{p}(l,m)\mathbf{B}(l,m)\mathbf{K}_\text{q}^H(l,m)\mathbf{E}_\text{q}^H(l,m), \tag{8.7}$$

Table 8.2: Overview of the antenna locations and characteristics of the VLBA.

Station	X / m	Y / m	Z / m	d / m	El_{low} / deg	El_{high} / deg
MKO	−5 464 075	−2 495 248	2 148 297	25	15	85
OVRO	−2 409 150	−4 478 573	3 838 617	25	15	85
BR	−2 112 065	−3 705 357	4 726 814	25	15	85
NL	−130 873	−4 762 317	4 226 851	25	15	85
HC	1 446 375	−4 447 940	4 322 306	25	15	85
KPN	−1 995 679	−5 037 318	3 357 328	25	15	85
PT	−1 640 954	−5 014 816	3 575 412	25	15	85
FD	−1 324 009	−5 332 182	3 231 962	25	15	85
LA	−1 449 753	−4 975 299	3 709 124	25	15	85
SC	2 607 849	−5 488 070	1 932 740	25	15	85

Note: X, Y, and Z define the station positions in Earth-centered coordinates. d denotes the diameter of the antenna dishes. El_{low} and El_{high} quantify the minimal and maximal elevation for an observation.

with the source brightness distribution $B(l, m)$. The integrals introduced in equation (1.12) transform into sums as each pixel of the brightness distribution is treated as an individual source. The w-term is neglected as $\sqrt{1 - l^2 - m^2} \approx 1$ is valid in this case.

The antenna positions and characteristics summarized in Table 8.2 are used to simulate observations with the VLBA. Additionally, the iterative pointing, typical for observations in the MOJAVE program, is simulated by taking 3 min measurement blocks followed by pauses of 1 h. Choosing different amounts of measurement blocks enables variation in the number of simulated visibilities.

The simulation algorithm in `pyvisgen` is structured as follows:

1. At measurement time t, the baselines of the interferometer are calculated relative to the source position and then converted to (u, v) coordinates. The resulting projected baselines already consider the propagation time differences of the signal to the individual telescopes, as described in Section 1.3.

2. For each projected baseline, the validity is checked, i.e., baselines with a blocked line of sight to the source at the current timestep are discarded.

3. For each projected and accepted baseline, the algorithm computes one complex visibility using the RIME formalism described in equation (8.7). Here, the simulation of multiple frequency channels is possible.

4. The simulated visibilities are stored in Flexible Image Transport System for (u, v) data (UVFITS) format enabling their import into all programs that support UVFITS input, including DIFMAP and CASA.

In Figure 8.6, the (u, v) coverage for a measurement simulated with `pyvisgen` is shown in red. For comparison, the purple dots represent the (u, v) coverage for the measurement of the source 7C 1055+5644 visualized in Figure 8.5. Simulation and actual measurement have similar data coverages, making the simulations suitable to train a model capable of applying to actual observation data. Appendix 6 summarizes the default simulation configuration. Calculating visibilities takes the longest computing time in the complete analysis, although simulations run parallel on a computing cluster. The calculation of the visibilities of one observation takes around 2 min, which leads to simulation times of around three days for the whole data set, which consists of 12 000 training, 1000 validation, and 1000 test images.

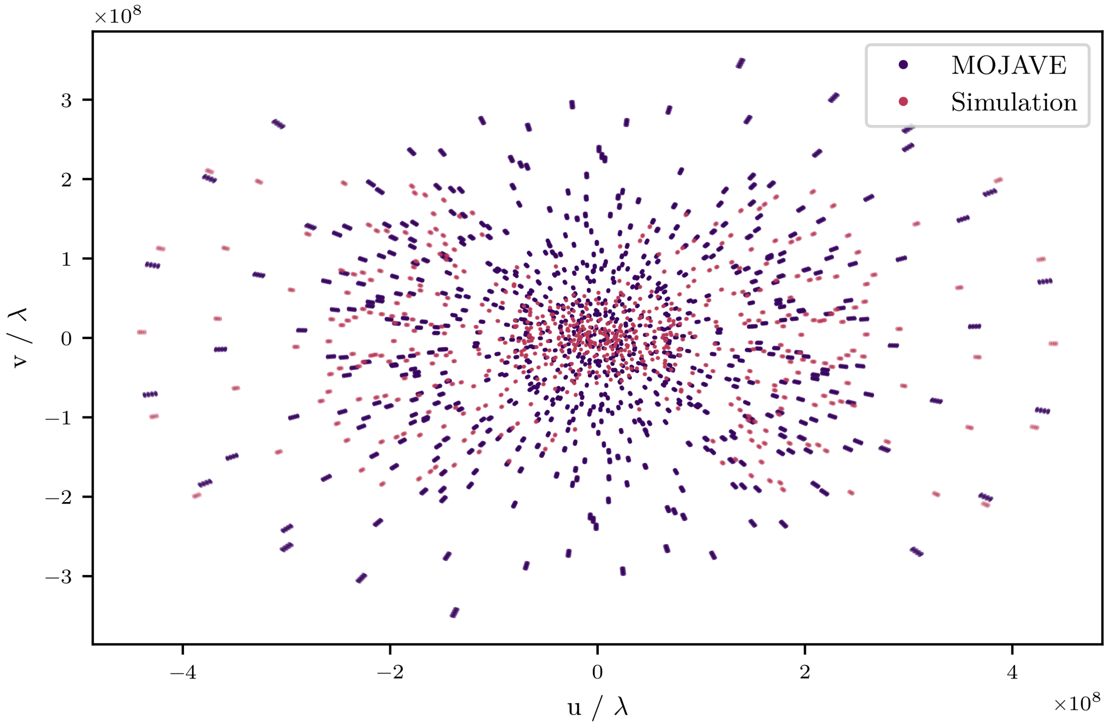


Figure 8.6: (u, v) coverage of the observation of 7C 1055+5644 (purple) and exemplary (u, v) coverage of a simulated observation from the data set created with `pyvisgen` (red).

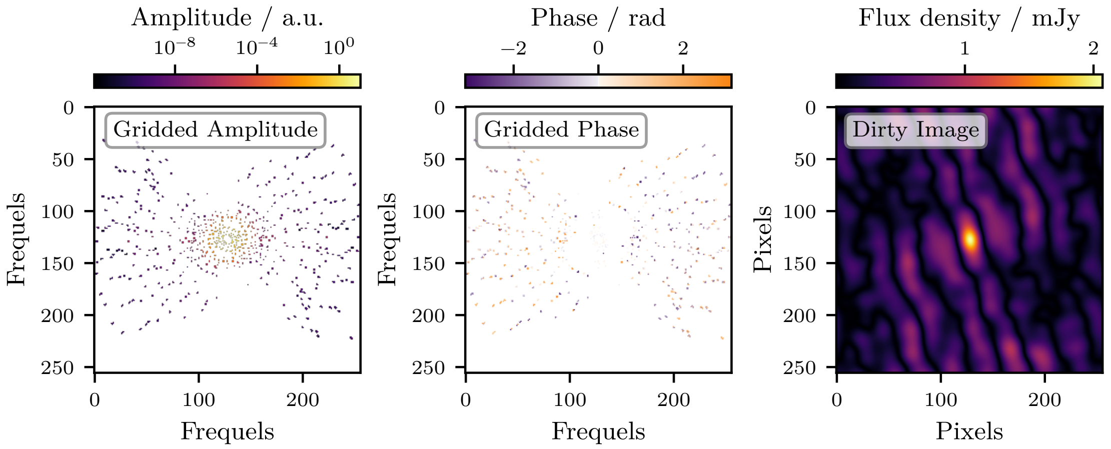


Figure 8.7: Amplitude (left) and phase (middle) map for the gridded advanced RIME data set. The field of view is set to 100 mas. The dirty image (right) shows noise artifacts parallel to the jet and a blurry source structure.

After generating the data set containing complex visibilities, the individual observations still have to be converted into a format that is suitable as input for neural networks. In `pyvisgen`, to each visibility, a (u, v) coordinate is assigned. As the developed deep learning models need tensors with regular grids as input data, combining the visibilities on a regular grid with the help of a so-called gridder is necessary. For this purpose, the gridder implemented in `pyvisgen` is used. Figure 8.7 visualizes exemplary input maps and the corresponding dirty image. After gridding the data, real and imaginary parts of the complex values are converted to amplitudes

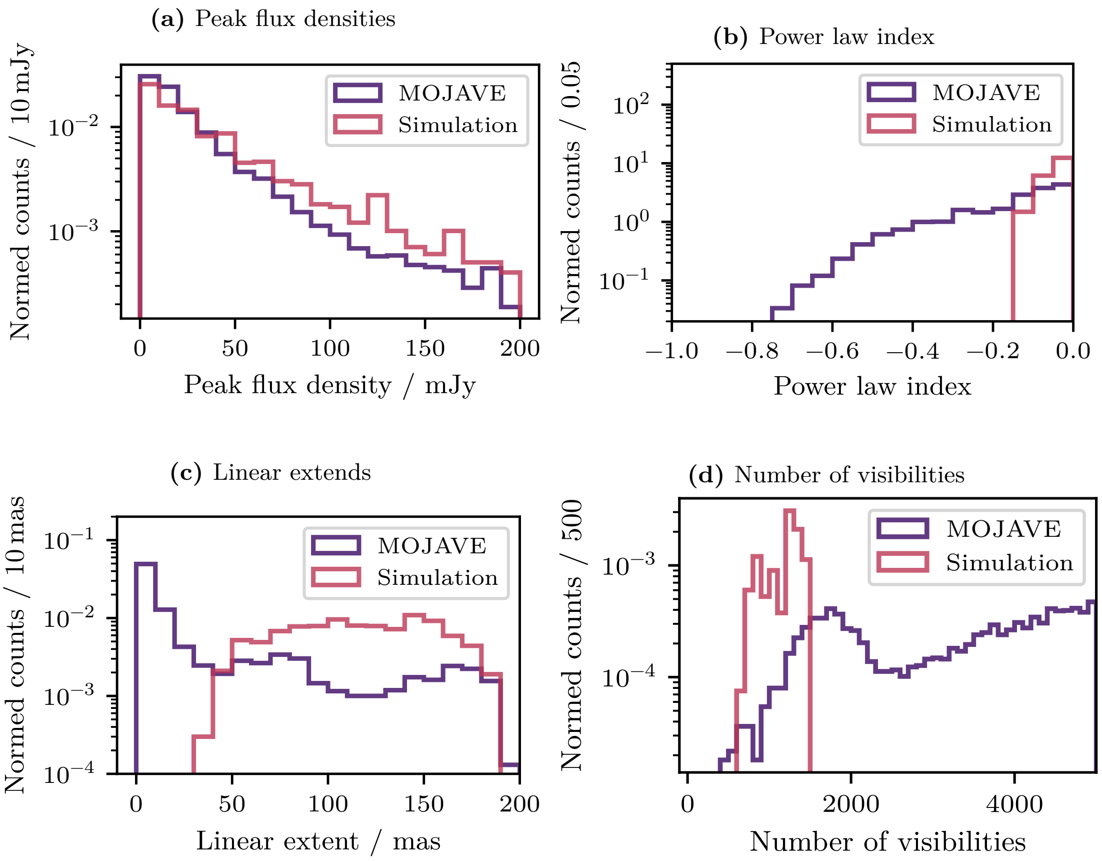


Figure 8.8: The simulated data set's source and observation characteristics (red) and, as a comparison, the same characteristics for the MOJAVE archive (purple). Peak flux densities, power-law indices, linear extends, and the number of visibilities per observation are shown.

and phases and stored in Hierarchical Data Format 5 (HDF5) format together with the target distributions. Again, like the data set described in Section 3.6, the incomplete maps serve as input, and the complete maps as targets when training the neural network model. Frequels are marked by a value of zero when their information is missing.

8.4 Data Set Overview

Finally, the properties of the simulated MOJAVE data set are evaluated. On the one hand, this serves to check whether the properties are comparable with the MOJAVE data archive. On the other hand, it gives a good overview of which source types the neural network learns when training with this data set. Of course, the model should learn to reconstruct missing information in general in the Fourier space. Nevertheless, it has to be mentioned that reconstructions for source types on which training has been done can always be predicted better.

For a direct comparison with the data from the MOJAVE archive [76], the parameters from Figure 8.1 are visualized again in Figure 8.8. The normalized distributions are shown in purple and red for the archive and simulated data. The main features of the distributions match very well. No deviations are visible for the distributions of the peak flux densities in Figure 8.8a. The power law indices shown in Figure 8.8b reveal a lack of indices below −0.18, meaning intensity

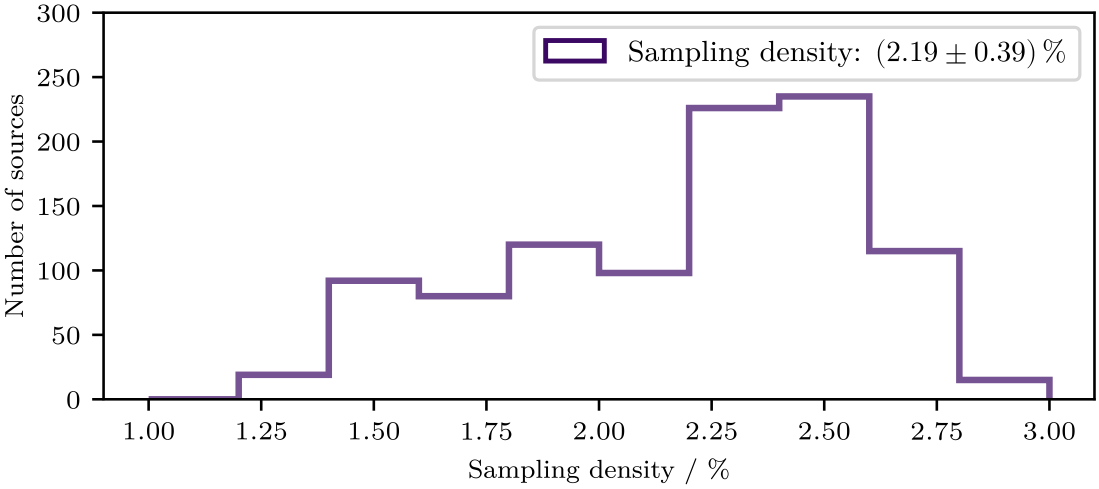


Figure 8.9: Histogrammed sampling densities for the simulated MOJAVE data sets. The distribution is sharp with a mean and standard deviation of 2.19 ± 0.39.

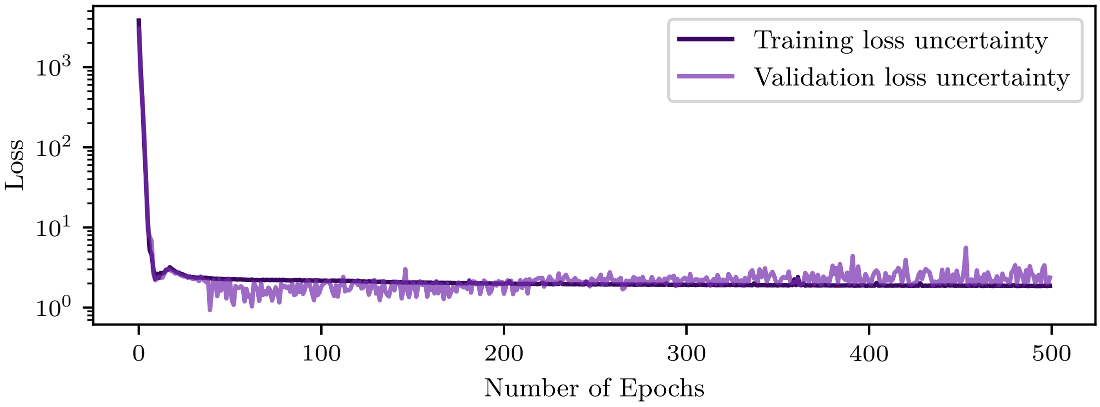


Figure 8.10: Loss curves for the uncertainty training session with the simulated MOJAVE data. Loss values against the number of epochs are shown for training and validation separately. The β-NLL loss is the used loss function.

decreases in the simulated data set are too flat. Figure 8.8c compares the linear extents of simulated sources and the ones observed in the MOJAVE program. It becomes clear that the simulated sources are generally more extensive than the observed ones. For the number of visibilities shown in Figure 8.8d, the distribution for the simulations peaks at a lower number than the one for the MOJAVE observations. This gap indicates a possibly worse (u, v) coverage for the simulations.

The histogrammed data coverage for the gridded (u, v) simulations in the simulated data set is visualized in Figure 8.9. The mean and standard deviation of the distribution are 2.19 ± 0.39. Therefore, nearly $98\,\%$ of the frequels of the input maps for the neural network contain no information and have to be reconstructed completely. Compared with MOJAVE observations, which have (u, v) coverages around $3\,\%$, this (u, v) coverage is realistic.

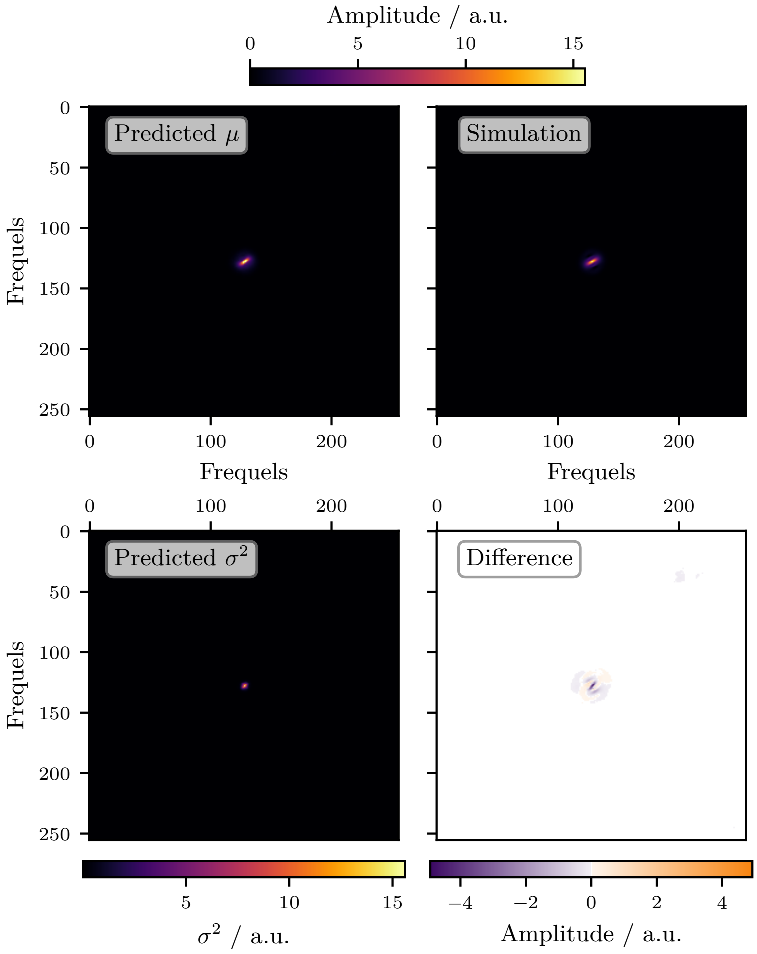


Figure 8.11: Amplitude reconstruction of uncertainty model trained on the MOJAVE data set. Visualization of prediction μ (upper left), simulated distribution (upper right), predicted variance σ^2 (lower left), and the difference between prediction and simulation (lower right).

8.5 Model Training and Reconstruction

The improved architecture, introduced in Section 7.4, was trained on the simulated MOJAVE data for 500 epochs. The β-NLL loss serves as the loss function. Figure 8.10 visualizes the loss values for the training session. After a sharp drop, the training loss smoothly decreases. For the first 220 epochs, the validations loss is slightly smaller than the training loss. Afterward, this trend shifts, and the validation loss stays higher for the rest of the training.

Again, amplitude and phase prediction are presented in two separate figures. Figure 8.11 visualizes an example of the predicted mean μ_{amp} (upper left) and the predicted variance σ^2_{amp} (lower left) generated with the uncertainty model for the amplitude. Additionally, the simulated truth (upper right) and the difference between predicted μ_{amp} and simulation (lower right) are shown. For the MOJAVE simulations, only the central amplitudes are significant, as the outer ones are close to zero. The difference reveals a good agreement between prediction and simulation, but the values in the center show deviations around 25 %. These deviations indicate the complexity of reconstructing the maximum amplitudes. Additionally, the predicted variances are high for these frequels. An exemplary prediction for the phase is illustrated in Figure 8.12. Again, the predicted mean μ_{phase} (upper left), the predicted variance σ^2_{phase} (lower left), the

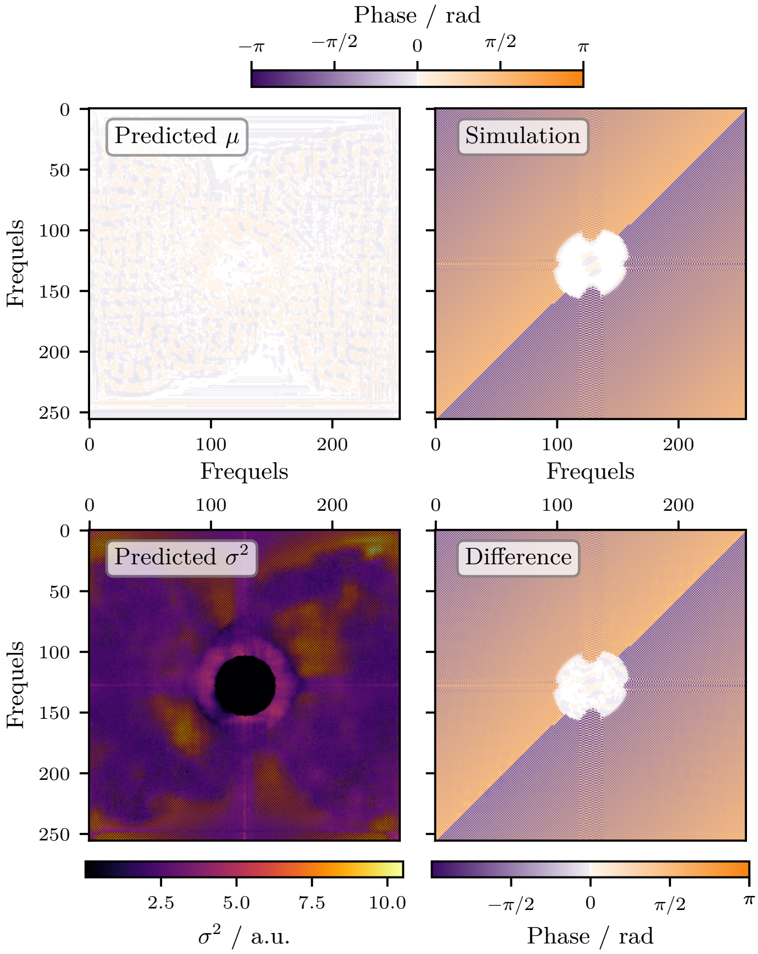


Figure 8.12: Phase reconstruction of uncertainty model trained on the MOJAVE data set. Visualization of prediction μ (upper left), simulated distribution (upper right), predicted variance σ^2 (lower left), and the difference between prediction and simulation (lower right).

simulated truth (upper right), and the difference between predicted μ_{phase} and simulation (lower right) are shown. Here, offsets between prediction and simulation are even more prominent. Still, the overall phase distribution is reconstructed to some extent. The values of the predicted uncertainties are generally high except for the central part. This part is extraordinary because of its lower phases. As in Section 7.4, no correlation of the frequels is considered.

An ensemble of reconstructions is received through the mean and variance prediction. No longer a single value is obtained per frequel, but a complete Gaussian distribution. Furthermore, the model generates these distributions for amplitude and phase separately. 1000 amplitude and phase samples are used to create 1000 versions of the cleaned image by applying the inverse Fourier transform. The pixel-wise mean is calculated to get the most probable source distribution. The pixel-wise standard deviation allows for giving an uncertainty map.

Figure 8.13 visualizes the reconstruction in image space. Here, the prediction (upper left) corresponds to the mean source distribution generated from the sampled versions. Again, the simulated source distribution (upper right) and the difference between prediction and simulation (lower right) are shown. The uncertainty map (lower left) is the standard deviation derived from the sampled source versions. Prediction and simulation are in good agreement. While the

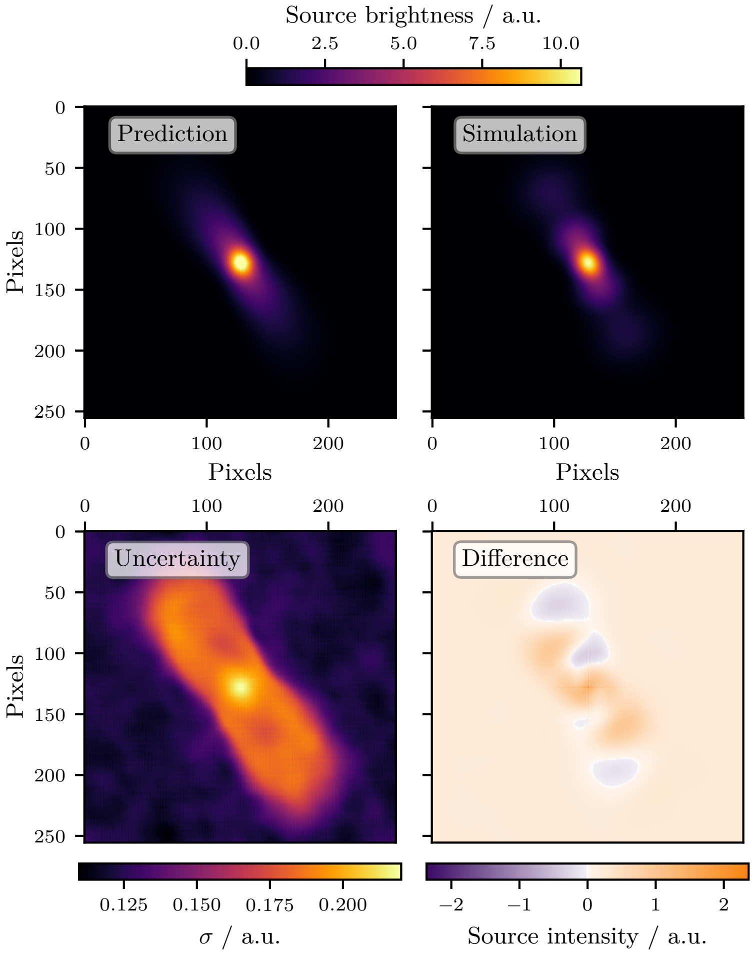


Figure 8.13: Reconstruction and uncertainty map of an exemplary source distribution from the MOJAVE simulations. While the shown reconstruction (upper left) is generated by calculating the mean of 1000 sampled source versions, their standard deviation gives the uncertainty map (lower left). The difference (lower right) between prediction and simulation (upper right) reveals a good agreement of the core component and slight deviations along the jet.

faintest source parts are missing in the reconstruction, the source's core component is slightly overestimated. Comparing the uncertainty map and the difference reveals that the uncertainty is highest at the source region. Background structures in the difference are also visible inside the uncertainty map. The uncertainty values do not cover the difference between prediction and simulation for this example.

8.6 Concluding Remarks and Future Prospects

The improvements described in this chapter have taken the complete analysis to an advanced level. Source characteristics have been adapted to the observations from the MOJAVE program, and by utilizing the advanced RIME formalism, the simulated visibilities match actual measurements. The RIME formalism is superior to the sampling mask method introduced in Section 3.2. There is still much potential as other effects, like the noise influence of neighboring sources and phase shifts due to the vast telescope distances in VLBI, can be added in the future.

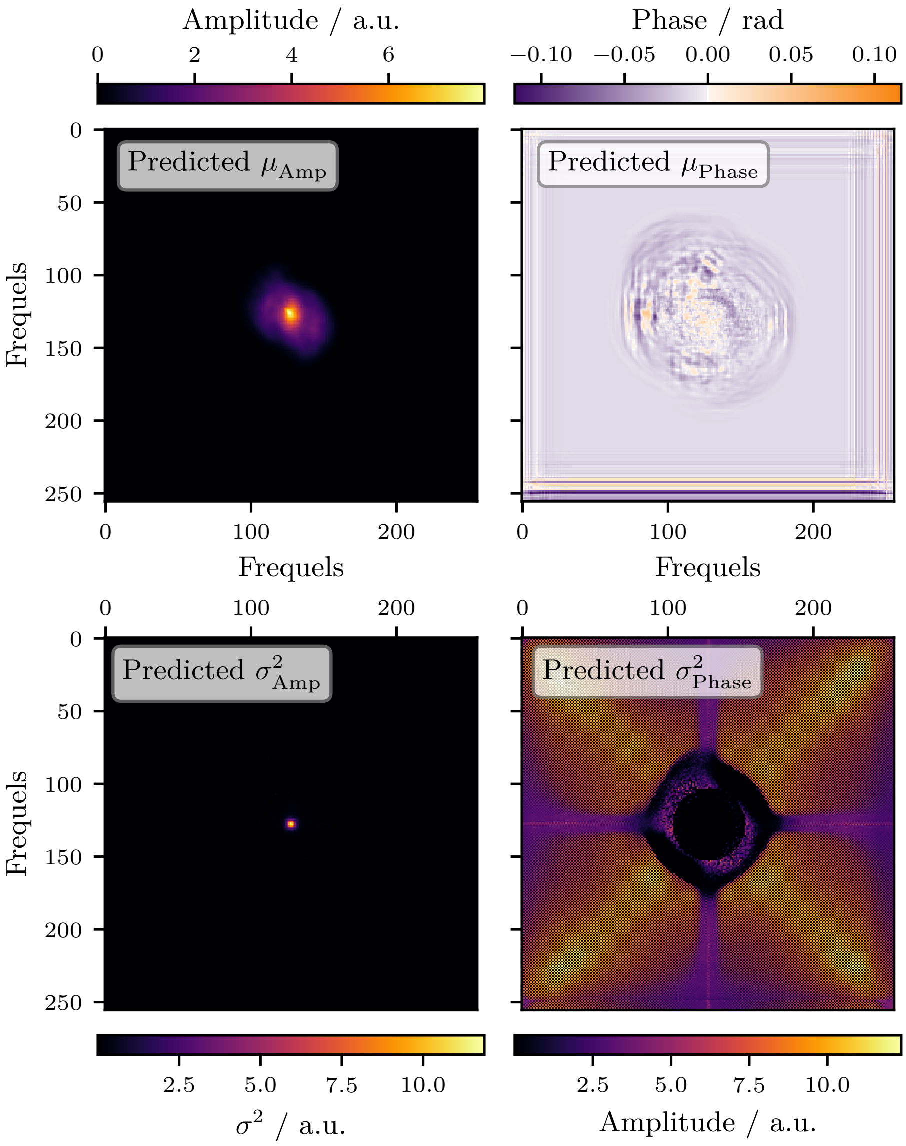


Figure 8.14: Amplitude and phase reconstructions of the 7C 1055+5644 observation, carried out on August 4th, 2019, using the VLBA. The uncertainty model trained on the MOJAVE simulations was used for the predictions. Predicted means (top row) and corresponding variances (bottom row) are visualized.

The results in Section 8.5 reveal a decreased reconstruction ability when compared to the reconstructions in Chapter 5, Chapter 6, and Chapter 7. This decrease is not surprising since the flux densities of the sources cover two orders of magnitude in the MOJAVE data set, complicating the amplitude reconstruction. Furthermore, the sampling densities for (256 × 256) pixels images of 2.5 % are much smaller than in previous data sets, compare Table 3.1. Additional noise effects introduced by the advanced RIME simulations also explain the model's decreased reconstruction ability. Nevertheless, the trained model shall finally be applied to actual observation data from the MOJAVE program. Figure 8.14 displays the amplitude and phase reconstruction for the 7C 1055+5644 observation of the VLBA carried out on August 4th, 2019. The predicted μ_{Amp} (top left) and σ^2_{Amp} (bottom left) show reasonable reconstructions. The amplitude distribution peaks in the center and has no artifacts in the outer regions. The predicted variance has its

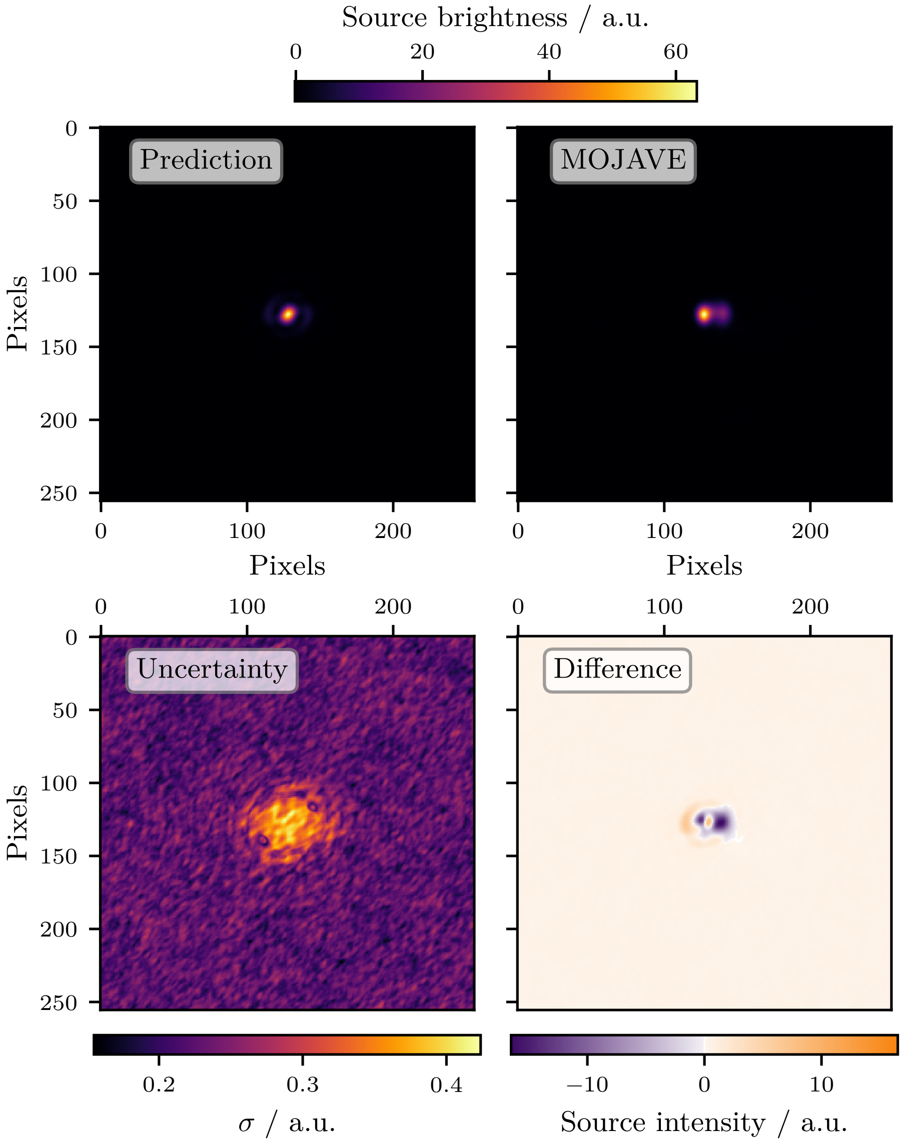


Figure 8.15: Reconstruction and uncertainty map of the 7C 1055+5644 observation, carried out on August 4th, 2019, using the VLBA. While the shown reconstruction (upper left) is generated by calculating the mean of 1000 sampled source versions, their standard deviation gives the uncertainty map (lower left). The difference (lower right) between the prediction and clean image from the MOJAVE archive (upper right) reveals a good agreement of the source's core component. The deep learning model did not reconstruct the jet structure. The clean image from the MOJAVE archive serves as a comparison and does not display the accurate source distribution.

highest values in the central part, indicating that the central frequel is the most complicated to reconstruct for the neural network model. The predicted μ_{Phase} (top right) and σ^2_{Phase} (bottom right) unveil more problems. While the phase distribution is blurry in the central part, it shows many artifacts at the edges. Still, the predicted variance prediction is high for all parts but the central one, as expected for this kind of prediction. The difficulties of the phase reconstruction are also transferred to the reconstructed source distribution, displayed in Figure 8.15. Here, the prediction (upper left) reveals the core component of the source. The comparison with the clean image from the MOJAVE archive (upper right) by creating the difference between both (lower right) shows that the predicted flux density fits the one from the MOJAVE archive. The jet structure is completely missing in the deep learning reconstruction. The predicted uncertainty map has the highest values in the source area, but they are low compared to the source's peak flux. As this is a reconstruction of actual observation data, no ground truth is available. The clean image from the MOJAVE archive suits as a comparison and does not display the accurate source distribution.

With such a big update, it is clear that deep learning-based analysis can not work immediately without minor problems. The results in this chapter directly point out the areas that still need a more detailed investigation. The two main points are to minimize further the mismatch between simulations and the actual observations from the MOJAVE archive and to examine the phase jumps in the simulated truths, as seen in Figure 8.12. The results again emphasize the importance of detailed simulations and give the perfect motivation to improve them further.

9 Conclusions and Outlook

"But the reward for lots of work seemed to be lots more. If you dug the biggest hole, they just gave you a bigger shovel."
— Terry Pratchett, *The Shepherd's Crown*

Modern radio interferometers with steadily increasing data rates require enhanced analysis strategies that quickly generate reproducible results utilizing affordable computing resources. In this thesis, I proposed a novel deep learning-based approach for imaging radio interferometer data, which is adequate to achieve this goal. Starting from scratch, as only a few people have explored the capability of neural networks for the imaging task before, I set up a complete framework called `radionets` [107]. The developed framework includes three main points: simulations, model training, and evaluation.

A new simulation chain has been established, ranging from radio galaxy simulations to mimicking radio interferometer observations and examining various noise simulation techniques to get as close to reality as possible. The advantages of utilizing the RIME formalism were also explored in this context. For the deep learning model training, a high-level environment was built based on `fastai` and `pytorch`. A suitable data class, architecture, and loss function was investigated here. Furthermore, the possibility of giving uncertainty estimates for the resulting cleaned images was tested. Several evaluation methods were implemented to benchmark model results, access the reconstruction quality, and compare the deep learning approach with established imaging software. Last, the first attempts to evaluate actual observation data with a model trained on a MOJAVE-inspired data set were performed.

Beginning with such a vast project is only possible as a team. In several theses I supervised, side projects were explored, and additional simulation packages like `radiosim` [108] and `pyvisgen` [30] were developed. Together we have built the foundation to explore further the advantages of deep learning-based analyses in radio interferometry and make it accessible to the whole community.

The developed simulations are fast and reliable. Everything is modular, allowing different parts to be adjusted without great effort. The sampling mask method enables the simulation of (u, v) spaces comparable to VLBA observations. The resulting incomplete (u, v) data is realistic, like those generated in radio interferometric observations. The deep learning model takes these incomplete (u, v) spaces as input and reconstructs the missing information.

Inspecting the results reveals that the deep learning-based reconstruction of Fourier data using convolutional neural networks works. The developed model quickly generates reproducible and reliable results when applied to radio galaxies composed of Gaussian components. The developed architecture enables a smooth convergence during training. Furthermore, adjustments to the loss function to fit the underlying problem improved amplitude and phase reconstructions.

Testing different noise levels on source distributions and visibilities confirmed the model's capability to deal with noisy input data. The developed model can correct noise present in the input data. This noise handling keeps background artifacts minimal in the reconstructed source distributions. The results are good when inspected visually, and evaluation methods on test data sets prove the excellent reconstruction quality. For example, the source area, the jet orientation offset, and the mean specific intensity deviation of the sources were examined to demonstrate the quality. Additional testing has shown that the models perform best on input data with (u, v) coverages comparable to that on which the model was trained. Therefore, the developed analysis is appropriate for evaluating sky survey data. Here, the observation characteristics are similar for all observations. Thus, a robust deep-learning model is ideally suited to reconstruct the different sub-images of the survey.

Comparison with the established imaging software WSCLEAN shows that the deep learning analysis is more robust when applied to an extensive test data set. WSCLEAN is not designed to analyze a wide range of similar data without optimizing the cleaning parameters. The neural network model can handle this range as it has learned to generalize during the training time of 18 h.

Another difference between established methods and the deep learning-based approach is that no convolution with a clean beam is necessary. The neural network model does not form a point source model, as is usually true when applying `CLEAN` algorithms. Instead, the (u, v) data is reconstructed directly in Fourier space. The developed model can reconstruct more complex data sets consisting of images with mixed source types and radio galaxies with advanced source shapes reasonably well. The investigation of uncertainty estimates generated by the model reveals that creating uncertainty maps with this analysis strategy is possible. As expected, the uncertainty maps highlight the noisy regions in the reconstructed source distributions. At the moment, the uncertainties estimates in Fourier space are more reliable than the ones transformed into image space.

Finally, it has been shown that a model can also be trained on a data set simulated utilizing an advanced RIME. Physical properties like the telescopes' beam size and the sources' extent were considered for this data set, resulting in a data set similar to observations from the MOJAVE archive. The first reconstruction of actual measurement data shows that the simulations need to be further adjusted to the conditions of observations performed by actual radio interferometers in the future. In conclusion, a proof-of-concept analysis was done in the scope of this thesis, proofing that deep learning models are applicable to reconstruct incomplete Fourier data in radio interferometric imaging. The framework `radionets` is available as an open-source package on GitHub accessible for the radio interferometry community [107].

Lots of work is already done, and the results have revealed that there is still lots more.

Uncertainty estimates enhance the interpretability of the cleaned radio images. Therefore, the presented approach will be further investigated in a future publication currently in preparation.

Simulations based on the RIME are essential for realistic descriptions of the effects influencing the radio signal approaching the receiver. RIME enables the simulation of correlated noise, which is not yet considered in the data sets used in this work. Additionally, the effects of bright neighboring sources on amplitudes and phases can be considered. Enhancing the RIME formalism implemented in `pyvisgen` will gradually decrease the simulation data mismatch.

Investigating the opportunity of reconstructing sky survey data using `radionets` in detail is another future project. The idea is to perform tests with data from the FIRST survey [6]. Generative Adversarial Network (GAN) simulated galaxies, provided by colleagues from Hamburg University, will serve as data to train the deep learning model [61].

Of course, the ultimate goal is a reliable reconstruction of actual observation data using `radionets`. In addition to the continued development of the deep learning-based cleaning of VLBA observations from the MOJAVE archive, more application scenarios are explored by participating in the SKA foreground challenge [118, 115].

Bibliography

1. V. A. Acciari et al. “Radio Imaging of the Very-High-Energy γ-Ray Emission Region in the Central Engine of a Radio Galaxy”. *Science* 325:5939, 2009, pages 444–448.
   DOI: 10.1126/science.1175406
2. A. Achterberg et al. “First year performance of the IceCube neutrino telescope”. *Astroparticle Physics* 26:3, 2006, pages 155–173.
   DOI: https://doi.org/10.1016/j.astropartphys.2006.06.007
3. J. Aleksić et al. “The major upgrade of the MAGIC telescopes, Part II: A performance study using observations of the Crab Nebula”. *Astroparticle Physics* 72, 2016, pages 76–94.
   DOI: https://doi.org/10.1016/j.astropartphys.2015.02.005
4. S.-I. Amari. “Backpropagation and stochastic gradient descent method”. *Neurocomputing* 5:4, 1993, pages 185–196.
   DOI: https://doi.org/10.1016/0925-2312(93)90006-O
5. D. Arnett and W. D. Arnett. *Supernovae and nucleosynthesis: an investigation of the history of matter, from the big bang to the present.* Princeton University Press, 1996
6. R. H. Becker, R. L. White, and D. J. Helfand. “The VLA’s FIRST Survey”. *Astronomical Data Analysis Software and Systems III.* Vol. 61. Astronomical Society of the Pacific Conference Series. 1994, page 165.
   URL: https://ui.adsabs.harvard.edu/abs/1994ASPC...61..165B
7. V. Beckmann and C. R. Shrader. *Active Galactic Nuclei.* 2012.
   URL: https://ui.adsabs.harvard.edu/abs/2012agn..book.....B
8. P. L. Biermann and P. A. Strittmatter. “Synchrotron Emission from Shock Waves in Active Galactic Nuclei”. *Astrophysical Journal* 322, 1987, page 643.
   DOI: 10.1086/165759
9. R. D. Blandford and A. Königl. “Relativistic jets as compact radio sources.” *Astrophysical Journal* 232, 1979, pages 34–48.
   DOI: 10.1086/157262
10. R. Blandford, D. Meier, and A. Readhead. “Relativistic Jets from Active Galactic Nuclei”. *Annual Review of Astronomy and Astrophysics* 57:1, 2019, pages 467–509.
    DOI: 10.1146/annurev-astro-081817-051948
11. P. Blomenkamp. “A Neural Network Architecture for Radio Interferometric Imaging. Advanced Simulations and Testing for Event Horizon Telescope Data”. Masters Thesis. TU Dortmund University, 2021
12. M. Born and E. Wolf. *Principals of Optics.* Pergamon Press, 1964
13. L. Bottou, F. E. Curtis, and J. Nocedal. *Optimization Methods for Large-Scale Machine Learning.* 2016.
    DOI: 10.48550/ARXIV.1606.04838

14. A. H. Bridle. “Sidedness, field configuration, and collimation of extragalactic radio jets.” *Astronomical Journal* 89, 1984, pages 979–986.
DOI: 10.1086/113593

15. D. S. Briggs, F. R. Schwab, and Sramek. “7.3 Gridding the Visibility Data”. *Synthesis Imaging in Radio Astronomy II*. Vol. 180. Astronomical Society of the Pacific Conference Series. 1999, pages 134–141.
URL: https://ui.adsabs.harvard.edu/abs/1999ASPC..180..127B

16. N. W. Broten et al. “Long Base Line Interferometry: A New Technique”. *Science* 156:3782, 1967, pages 1592–1593.
DOI: 10.1126/science.156.3782.1592

17. M. N. Büchel. “Rekonstruktion von VLBI Simulationsdatensätzen mit Hilfe von Neuronalen Netzen. Ansätze zur Dimensionssenkung durch Schätzung von Gaußparametern”. Bachelors Thesis. TU Dortmund University, 2021

18. G. Butcher. *Tour of the Electromagnetic Spectrum*. 3rd. NASA, 2016.
URL: http://science.nasa.gov/ems

19. CASA Team et al. “CASA, the Common Astronomy Software Applications for Radio Astronomy”. *Publications of the Astronomical Society of the Pacific* 134:1041, 2022, page 114501.
DOI: 10.1088/1538-3873/ac9642

20. B. G. Clark. “An efficient implementation of the algorithm ’CLEAN’”. *Astronomy&Astrophysics* 89, 1980, page 377.
URL: http://adsabs.harvard.edu/abs/1980A%26A....89..377C

21. C. Collaboration. *Science with the Cherenkov Telescope Array*. World Scientific, 2019.
DOI: 10.1142/10986

22. T. J. Cornwell. “Multiscale CLEAN Deconvolution of Radio Synthesis Images”. *IEEE Journal of Selected Topics in Signal Processing* 2, 2008, pages 793–801.
DOI: 10.1109/JSTSP.2008.2006388

23. DeepL SE. *DeepL Translator*. 2022.
URL: https://www.deepl.com/translator visited on 2022-11-23

24. P. E. Dewdney et al. “The Square Kilometre Array”. *Proceedings of the IEEE* 97:8, 2009, pages 1482–1496.
DOI: 10.1109/JPROC.2009.2021005

25. F. R. Elder et al. “Radiation from Electrons in a Synchrotron”. *Physical Review Journal* 71, 11 1947, pages 829–830.
DOI: 10.1103/PhysRev.71.829.5

26. ErUM-Data-Hub. *Event-Programme*. 2022.
URL: https://erumdatahub.de/en/veranstaltungen/ visited on 2022-11-23

27. B. L. Fanaroff and J. M. Riley. “The Morphology of Extragalactic Radio Sources of High and Low Luminosity”. *Monthly Notices of the Royal Astronomical Society* 167:1, 1974, pages 31–36.
DOI: 10.1093/mnras/167.1.31P

28. S. Farsiu et al. “Advances and challenges in super-resolution”. *International Journal of Imaging Systems and Technology* 14:2, 2004, pages 47–57.
DOI: https://doi.org/10.1002/ima.20007

29. S. Fröse. “A Neural Network Architecture for Radio Interferometric Imaging. Advanced Simulations and Testing for Event Horizon Telescope Data”. Masters Thesis. TU Dortmund University, 2021

30. S. Fröse et al. *pyvisgen*. Version 0.0.15. 2022.
    URL: https://github.com/radionets-project/pyvisgen

31. T. Fuchs. *Development of a Machine Learning Based Analysis Chain for the Measurement of Atmospheric Muon Spectra with IceCube*. 2017.
    DOI: 10.48550/ARXIV.1701.04067

32. F. Geyer. “Reconstructing Radio Interferometric Data Using Neural Networks. Creation of toy datasets and feasibility studies based on convolutions”. Masters Thesis. TU Dortmund University, 2020

33. G. Ghirlanda et al. “Compact radio emission indicates a structured jet was produced by a binary neutron star merger”. *Science* 363:6430, 2019, pages 968–971.
    DOI: 10.1126/science.aau8815

34. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
    URL: http://www.deeplearningbook.org

35. K. Grainge et al. “Square Kilometre Array: The radio telescope of the XXI century”. *Astronomy Reports* 61:4, 2017, pages 288–296.
    DOI: 10.1134/S1063772917040059

36. E. W. Greisen. *AIPS Memos*. 2022.
    URL: http://www.aips.nrao.edu/aipsmemo.html visited on 2022-09-10

37. S. Gross and M. Wilber. *Training and investigating Residual Nets*. 2016.
    URL: http://torch.ch/blog/2016/02/04/resnets.html visited on 2022-02-05

38. R. Guidotti et al. “A Survey of Methods for Explaining Black Box Models”. *ACM Computing Surveys* 51:5, 2018.
    DOI: 10.1145/3236009

39. C. Guillemot and O. Le Meur. “Image Inpainting : Overview and Recent Advances”. *IEEE Signal Processing Magazine* 31:1, 2014, pages 127–144.
    DOI: 10.1109/MSP.2013.2273004

40. M. P. van Haarlem et al. “LOFAR: The LOw-Frequency ARray”. *Astronomy&Astrophysics* 556, 2013, A2.
    DOI: 10.1051/0004-6361/201220873

41. J. P. Hamaker, J. D. Bregman, and R. J. Sault. “Understanding radio polarimetry. I. Mathematical foundations”. *Astronomy and Astrophysics Supplement Series* 117:1, 1996, pages 137–147.
    DOI: 10.1051/aas:1996146

42. C. R. Harris et al. “Array programming with NumPy”. *Nature* 585:7825, 2020, pages 357–362.
    DOI: 10.1038/s41586-020-2649-2

43. T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009.
    DOI: https://doi.org/10.1016/j.astropartphys.2017.05.001

44. T. Hastie, J. Friedman, and R. Tibshirani. “Neural Networks”. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer New York, 2001, pages 347–369.
    DOI: 10.1007/978-0-387-21606-5_11

45. K. He et al. “Deep Residual Learning for Image Recognition”. *arXiv*, 2015.
    DOI: 10.48550/ARXIV.1512.03385

46. K. He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* 2015.
    DOI: 10.1109/ICCV.2015.123

47. D. N. Hoang et al. “Deep LOFAR observations of the merging galaxy cluster CIZA J2242.8+5301”. *Monthly Notices of the Royal Astronomical Society* 471:1, 2017, pages 1107–1125.
    DOI: 10.1093/mnras/stx1645

48. J. A. Högbom. “Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines”. *Astronomy and Astrophysics Supplement* 15, 1974, page 417.
    URL: http://adsabs.harvard.edu/abs/1974A%26AS...15..417H

49. J. Howard and S. Gugger. “Deep Learning for Coders with fastai and PyTorch”. O’Reilly Media, Inc., 2020. Chap. 5, pages 205–207.
    URL: https://course.fast.ai/Resources/book.html

50. J. Howard and S. Gugger. “Fastai: A Layered API for Deep Learning”. *Information* 11:2, 2020. ISSN: 2078-2489.
    DOI: 10.3390/info11020108

51. J. D. Hunter. “Matplotlib: A 2D Graphics Environment”. Version 1.4.3. *Computing in Science & Engineering* 9:3, 2007, pages 90–95.
    DOI: 10.1109/MCSE.2007.55

52. IceCube Collaboration et al. “Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A”. *Science* 361:6398, 2018, eaat1378.
    DOI: 10.1126/science.aat1378

53. IceCube collaboration et al. “A convolutional neural network based cascade reconstruction for the IceCube Neutrino Observatory”. *Journal of Instrumentation* 16:07, 2021, P07041.
    DOI: 10.1088/1748-0221/16/07/P07041

54. K. Jarrett et al. “What is the best multi-stage architecture for object recognition?” *2009 IEEE 12th International Conference on Computer Vision.* 2009, pages 2146–2153.
    DOI: 10.1109/ICCV.2009.5459469

55. J. L. Jonas. “MeerKAT—The South African Array With Composite Dishes and Wide-Band Single Pixel Feeds”. *Proceedings of the IEEE* 97:8, 2009, pages 1522–1530.
    DOI: 10.1109/JPROC.2009.2020713

56. S. de Jong et al. “High-energy emission processes in M87”. *Monthly Notices of the Royal Astronomical Society* 450:4, 2015, pages 4333–4341.
    DOI: 10.1093/mnras/stv927

57. Kaggle. *Learn.* 2022.
URL: https://www.kaggle.com/learn visited on 2022-11-23

58. Y. Kasper. “Applying a Grid Search Approach for Automated Imaging of VLBI Data”. Masters Thesis. TU Dortmund University, 2020

59. D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization.* 2017.
DOI: 10.48550/ARXIV.1412.6980

60. A. Königl. “Relativistic gasdynamics in two dimensions”. *The Physics of Fluids* 23:6, 1980, pages 1083–1090.
DOI: 10.1063/1.863110

61. J. Kummer et al. “Radio Galaxy Classification with wGAN-Supported Augmentation”. *INFORMATIK 2022.* Gesellschaft für Informatik, Bonn, 2022, pages 469–478.
DOI: 10.18420/inf2022_38

62. K. Laudamus. “Imaging Radio Interferometric Data with Neural Networks”. Bachelors Thesis. Universität Hamburg, 2020

63. C. Ledig et al. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2017

64. L. Linhoff et al. “Excluding possible sites of high-energy emission in 3C 84”. *Monthly Notices of the Royal Astronomical Society* 500:4, 2020, pages 4671–4677.
DOI: 10.1093/mnras/staa3521

65. M. L. Lister et al. “MOJAVE. X. Parsec-scale Jet Orientation Variations and Superluminal Motion in Active Galactic Nuclei”. *The Astronomical Journal* 146:5, 2013, page 120.
DOI: 10.1088/0004-6256/146/5/120

66. M. L. Lister et al. “MOJAVE. XV. VLBA 15 GHz Total Intensity and Polarization Maps of 437 Parsec-scale AGN Jets from 1996 to 2017”. *Astrophysical Journal Supplement Series* 234:1, 2018, page 12.
DOI: 10.3847/1538-4365/aa9c44

67. M. L. Lister et al. “MOJAVE. XVII. Jet Kinematics and Parent Population Properties of Relativistically Beamed Radio-loud Blazars”. *The Astrophysical Journal* 874:1, 2019, page 43.
DOI: 10.3847/1538-4357/ab08ee

68. F. Lord Rayleigh. “XXXI. Investigations in optics, with special reference to the spectroscope”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8:49, 1879, pages 261–274.
DOI: 10.1080/14786447908639684

69. J. C. Mather et al. “A preliminary measurement of the cosmic microwave background spectrum by the Cosmic Background Explorer (COBE) satellite”. *The Astrophysical Journal* 354, 1990, pages L37–L40

70. W. McKinney. “Data Structures for Statistical Computing in Python”. *Proceedings of the 9th Python in Science Conference.* 2010, pages 56–61.
DOI: 10.25080/Majora-92bf1922-00a

71. S. K. Meher and G. Panda. “Deep learning in astronomy: a tutorial perspective”. *European Physical Journal Special Topics* 230:10, 2021, pages 2285–2317.
DOI: 10.1140/epjs/s11734-021-00207-9

72. E. Middelberg and U. Bach. “High resolution radio astronomy using very long baseline interferometry”. *Reports on Progress in Physics* 71:6, 2008, page 066901.
DOI: 10.1088/0034-4885/71/6/066901

73. P. Milanfar, ed. *Super-Resolution Imaging*. CRC Press, 2017.
DOI: https://doi.org/10.1201/9781439819319

74. E. Z. Miranda. “Neural Networks Applied to Interferometric Imaging”. Bachelors Thesis. Universität Hamburg, 2021

75. MOJAVE team. *7C 1055+5644 Source Page*. 2022.
URL: https://www.cv.nrao.edu/MOJAVE/sourcepages/1055+567.shtml visited on 2022-11-22

76. MOJAVE team. *MOJAVE Archive*. 2022.
URL: https://www.cv.nrao.edu/MOJAVE/allsources.html visited on 2022-12-01

77. K. Morik and W. Rhode. *Technical report for Collaborative Research Center SFB 876 - Graduate School*. Technical report 2. TU Dortmund University, 2020

78. K. Morik, W. Rhode, and more. “Key Concepts in Machine Learning and Data Analysis”. *Machine Learning under Resource Constraints - Discovery in Physics*. Walter de Gruyter GmbH, 2022, pages 51–76.
DOI: 10.1515/9783110785968-001

79. W. R. Morningstar et al. “Analyzing interferometric observations of strong gravitational lenses with recurrent and convolutional neural networks”. *arXiv e-prints*, 2018.
arXiv: 1808.00011 [astro-ph.IM]

80. W. R. Morningstar et al. “Data-Driven Reconstruction of Gravitationally Lensed Galaxies using Recurrent Inference Machines”. *arXiv e-prints*, 2019.
DOI: 10.3847/1538-4357/ab35d7

81. S. Mozaffari et al. “Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review”. *IEEE Transactions on Intelligent Transportation Systems* 23:1, 2022, pages 33–47.
DOI: 10.1109/TITS.2020.3012034

82. C. A. Muller and J. H. Oort. “Observation of a Line in the Galactic Radio Spectrum: The Interstellar Hydrogen Line at 1,420 Mc./sec., and an Estimate of Galactic Rotation”. *Nature* 168:4270, 1951, pages 357–358.
DOI: 10.1038/168357a0

83. P. J. Napier et al. “The Very Long Baseline Array”. *Proceedings of the IEEE* 82:5, 1994, pages 658–672.
DOI: 10.1109/5.284733

84. T. Needham. “Visual Complex Analysis”. *Visual Complex Analysis*. Oxford University Press, 1999. Chap. 1, pages 10–14

85. H. Netzer. “Revisiting the Unified Model of Active Galactic Nuclei”. *Annual Review of Astronomy and Astrophysics* 53:1, 2015, pages 365–408.
DOI: 10.1146/annurev-astro-082214-122302

86. J. Noordam. “The measurement equation of a generic radio telescope, AIPS++ implementation note nr 185”. *Tech. Rep.*, 1996

87. H. J. Nussbaumer. “The Fast Fourier Transform”. *Fast Fourier Transform and Convolution Algorithms*. Springer Berlin Heidelberg, 1981, pages 85–90.
DOI: 10.1007/978-3-662-00551-4_4

88. M. Office. *Cartopy*. Version 0.17.0. 2011-2018.
URL: https://scitools.org.uk/cartopy/docs/latest/

89. A. R. Offringa et al. “wsclean: an implementation of a fast, generic wide-field imager for radio astronomy”. *Monthly Notices of the Royal Astronomical Society* 444:1, 2014, pages 606–619.
DOI: 10.1093/mnras/stu1368

90. E. Osinga et al. “Diffuse radio emission from galaxy clusters in the LOFAR Two-metre Sky Survey Deep Fields”. *Astronomy&Astrophysics* 648, 2021, A11.
DOI: 10.1051/0004-6361/202039076

91. Pandas development team. *pandas-dev/pandas: Pandas*. Version latest. 2020.
DOI: 10.5281/zenodo.3509134

92. A. Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019.
DOI: 10.48550/ARXIV.1912.01703

93. K. Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2:11, 1901, pages 559–572.
DOI: 10.1080/14786440109462720

94. T. Pearson and A. Readhead. “Image formation by self-calibration in radio astronomy”. *Annual review of astronomy and astrophysics* 22, 1984, pages 97–130

95. Planck Collaboration et al. “Planck early results. I. The Planck mission”. *Astronomy&Astrophysics* 536, 2011, A1.
DOI: 10.1051/0004-6361/201116464

96. A. M. Price-Whelan et al. “The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package”. *The Astronomical Journal* 156:3, 2018, page 123.
DOI: 10.3847/1538-3881/aabc4f

97. M. A. Prieto et al. “The spectral energy distribution of the central parsecs of the nearest AGN”. *Monthly Notices of the Royal Astronomical Society* 402:2, 2010, pages 724–744. ISSN: 0035-8711.
DOI: 10.1111/j.1365-2966.2009.15897.x

98. Python Software Foundation. *Python*. Version 3.9.7. 2022.
URL: https://www.python.org/

99. Ramatsoku, M. et al. “Collimated synchrotron threads linking the radio lobes of ESO 137-006”. *Astronomy&Astrophysics* 636, 2020, page L1.
DOI: 10.1051/0004-6361/202037800

100. U. Rau et al. “Advances in Calibration and Imaging Techniques in Radio Interferometry”. *Proceedings of the IEEE* 97:8, 2009, pages 1472–1481.
DOI: 10.1109/JPROC.2009.2014853

101. A. Readhead and P. Wilkinson. "The mapping of compact radio sources from VLBI data". *The Astrophysical Journal* 223, 1978, pages 25–36

102. Rhodes University Centre for Radio Astronomy Techniques & Technologies. *Fundamentals of Interferometry: 4 Visibility Space, Spatial Filtering*. 2022.
URL: https://github.com/ratt-ru/fundamentals_of_interferometry/blob/master/4_Visibility_Space/figures/spatialfiltering.png visited on 2022-11-17

103. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". *Nature* 323:6088, 1986, pages 533–536.
DOI: 10.1038/323533a0

104. M. Ryle, B. Elsmore, and A. C. Neville. "High-resolution observations of the radio sources in Cygnus and Cassiopeia". *Nature* 205:4978, 1965, pages 1259–1262

105. S. Santurkar et al. "How Does Batch Normalization Help Optimization?" *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.
URL: https://proceedings.neurips.cc/paper/2018/file/905056c1ac1dad141560467e0a99e1cf-Paper.pdf

106. K. Schmidt et al. "Deep learning-based imaging in radio interferometry". *Astronomy&Astrophysics* 664, 2022, A134.
DOI: 10.1051/0004-6361/202142113

107. K. Schmidt et al. *radionets*. Version 0.1.15. 2022.
URL: https://github.com/radionets-project/radionets

108. K. Schmidt et al. *radiosim*. Version 0.0.4. 2022.
URL: https://github.com/radionets-project/radiosim

109. F. R. Schwab. "Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry". *Astronomical Journal* 89, 1984, pages 1076–1081.
DOI: 10.1086/113605

110. U. J. Schwarz. "Mathematical-statistical Description of the Iterative Beam Removing Technique (Method CLEAN)". *Astronomy&Astrophysics* 65, 1978, page 345.
URL: https://ui.adsabs.harvard.edu/abs/1978A&A....65..345S

111. T. Sedlaczek. "Untersuchung der Jetstruktur des TeV-Radiogalaxie-Kandidaten IC 310 mit Hilfe von VLBA-Daten durch zwei Analyseprogramme". Bachelors Thesis. TU Dortmund University, 2019

112. M. Seitzer et al. *On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks*. 2022.
DOI: 10.48550/ARXIV.2203.09168

113. M. C. Shepherd, T. J. Pearson, and G. B. Taylor. "DIFMAP: an interactive program for synthesis imaging." *Bulletin of the Astronomical Society* 26, 1994, pages 987–989.
URL: https://ui.adsabs.harvard.edu/abs/1994BAAS...26..987S

114. T. W. Shimwell et al. "The LOFAR Two-metre Sky Survey - II. First data release". *Astronomy&Astrophysics* 622, 2019, A1.
DOI: 10.1051/0004-6361/201833559

115. SKAO. *SKA Foreground Challenge*. 2022.
URL: https://sdc3.skao.int/challenges/foregrounds visited on 2022-11-25

116. O. M. Smirnov. “Revisiting the radio interferometer measurement equation - I. A full-sky Jones formalism”. *Astronomy&Astrophysics* 527, 2011, A106.
DOI: 10.1051/0004-6361/201016082

117. M. Spada et al. “Internal shocks in the jets of radio-loud quasars”. *Monthly Notices of the Royal Astronomical Society* 325:4, 2001, pages 1559–1570.
DOI: 10.1046/j.1365-8711.2001.04557.x

118. M. Spinelli et al. “SKAO HI intensity mapping: blind foreground subtraction challenge”. *Monthly Notices of the Royal Astronomical Society* 509:2, 2021, pages 2048–2074.
DOI: 10.1093/mnras/stab3064

119. Y. Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2014

120. M. Telgarsky. “Benefits of depth in neural networks”. *29th Annual Conference on Learning Theory.* Vol. 49. Proceedings of Machine Learning Research. Proceedings of Machine Learning Research, 2016, pages 1517–1539.
URL: https://proceedings.mlr.press/v49/telgarsky16.html

121. The Astropy Collaboration et al. “Astropy: A community Python package for astronomy”. *Astronomy&Astrophysics* 558, 2013, A33.
DOI: 10.1051/0004-6361/201322068

122. A. R. Thompson, J. M. Moran, and G. W. Swenson. *Interferometry and synthesis in radio astronomy.* Springer Nature, 2017

123. A. R. Thompson, J. M. Moran, and G. W. Swenson. “Interferometry and synthesis in radio astronomy”. Springer Nature, 2017. Chap. 3: Analysis of the Interferometer Response, pages 92–93

124. A. R. Thompson, J. M. Moran, and G. W. Swenson. “Interferometry and synthesis in radio astronomy”. Springer Nature, 2017. Chap. 15.1: Van Cittert-Zernike Theorem, pages 767–775

125. A. R. Thompson, J. M. Moran, and G. W. Swenson. “Interferometry and synthesis in radio astronomy”. Springer Nature, 2017. Chap. 11: Further Imaging Techniques, pages 551–598

126. S. van der Tol, Veenboer, Bram, and Offringa, André R. “Image Domain Gridding: a fast method for convolutional resampling of visibilities”. *Astronomy&Astrophysics* 616, 2018, A27.
DOI: 10.1051/0004-6361/201832858

127. C. M. Urry and P. Padovani. “Unified Schemes for Radio-loud Active Galactic Nuclei”. *Publications of the Astronomical Society of the Pacific* 107:715, 1995, page 803.
DOI: 10.1086/133630

128. P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* 17, 2020, pages 261–272.
DOI: 10.1038/s41592-019-0686-2

129. S. van der Walt et al. “scikit-image: image processing in Python”. *PeerJ* 2, 2014, e453.
DOI: 10.7717/peerj.453

130. W. Walter. *Analysis I*. 2. Auflage. Springer-Verlag Berling Heidelberg GmbH, 1990, pages 285–286

131. D. H. Wolpert. "The Lack of A Priori Distinctions Between Learning Algorithms". *Neural Computation* 8:7, 1996, pages 1341–1390.
DOI: 10.1162/neco.1996.8.7.1341

132. R. Zanin et al. "MARS, The MAGIC Analysis and Reconstruction Software". *International Cosmic Ray Conference*. Vol. 33. International Cosmic Ray Conference. 2013, page 2937.
URL: https://ui.adsabs.harvard.edu/abs/2013ICRC...33.2937Z

Glossary

ADAM Adaptive Moment Estimation. 30

AGN Active Galactic Nucleus. 3, 4, 15

AIPS Astronomical Image Processing System. 20

CASA Common Astronomy Software Applications. 40, 78

CNN Convolutional Neural Network. 12

CTA Cherenkov Telescope Array. 2

DFT Discrete Fourier Transform. 9

DIFMAP Difference Mapping. 9, 78

EHT Event Horizon Telescope. 115

FFT Fast Fourier Transform. 9, 107

FIRST Faint Images of the Radio Sky at Twenty-cm. 51, 90

FITS Flexible Image Transport System. 19, 20, 23, 40, 48, 115

FoV Field of View. 72, 115

FR I Fanaroff-Riley Class I. 3

FR II Fanaroff-Riley Class II. 3

GAN Generative Adversarial Network. 90, 111

GPU Graphics Processing Unit. 109

HardTanh hard hyperbolic tangent. 30

HDF5 Hierarchical Data Format 5. 78, 107, 115

HI line neutral hydrogen line. 3

IceCube IceCube Neutrino Observatory. 2

L1 Mean Absolute Difference. 30

LiDO3 Linux Cluster an der TU Dortmund. 14

LOFAR Low Frequency Array. 1, 9

MAGIC Major Atmospheric Gamma-Ray Imaging Cherenkov. 2

MeerKAT Karoo Array Telescope in the Meerkat National Park. 3

ML2R Competence Center Machine Learning Rhine-Ruhr. 14

MOJAVE Monitoring Of Jets in Active galactic nuclei with VLBA Experiments. 1, 2, 9, 15, 71–77, 80–90, 113

ms measurement sets. 48

MSE Mean Squared Error. 30

PCA Principal Component Analysis. 33, 35, 37, 39

PDF Probability Density Function. 65

PReLU Parameter Rectified Linear Unit. 12, 13, 28, 63

PSF Point Spread Function. 9

ReLU Rectified Linear Unit. 12

ResBlock Residual Block. 28, 29, 62

ResNet Residual Network. 60

RIME Radio Interferometer Measurement Equation. 2, 4, 6, 7, 14, 18–21, 23–25, 44, 48–50, 71, 77, 79, 86, 87, 89, 90, 115

SED Spectral Energy Distribution. 3

SFB 876 Sonderforschungsbereich 876. 11

SGD Stochastic Gradient Descent. 12, 30

SKA Square Kilometre Array. 1, 3, 90

SRResNet Super Resolution Residual Network. 27–29, 63

UVFITS Flexible Image Transport System for (u, v) data. 78

VLA Very Large Array. 115

VLBA Very Long Baseline Array. 1, 2, 7–9, 16, 17, 71, 72, 77, 78, 86, 87, 89, 90, 107, 115

VLBI Very Long Baseline Interferometry. 4, 15, 19, 86

WSCLEAN w-Stacking Clean. 2, 9, 23, 39–41, 44, 48–50, 90

Appendix

1 Funding Acknowledgement and Data Usage Statement

”Profilbildung 2020” Funding

The project ”Big Bang to Big Data (B3D)” is receiving funding from the program ”Profilbildung 2020”, an initiative of the Ministry of Culture and Science of the State of Northrhine Westphalia. The sole responsibility for the content of this publication lies with the authors.

DFG Funding

This thesis was supported by funding from the Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Center 876 “Providing Information by Resource-Constrained Data Analysis”, DFG project number 124020371, SFB project C3.

MOJAVE: VLBA 15 GHz

This research has made use of data from the MOJAVE database that is maintained by the MOJAVE team (Lister et al. 2018) [66].

2 Default Config: radionets_simulation

```
# This is a TOML document.

title = "Simulation configuration"

[mode]
quiet = true

[paths]
data_path = "./example_data/"
data_format = "h5"

[gaussians]
simulate = true
num_components = [4, 10]

[point_sources]
simulate = false
add_extended = false

[image_options]
bundles_train = 100
bundles_valid = 20
bundles_test = 20
bundle_size = 500
img_size = 64
noise = false
noise_level = 5
white_noise = false
mean_real = 0.85
std_real = 0.0425
mean_imag = 0.2
std_imag = 0.01

[sampling_options]
fourier = true
real_imag = false
amp_phase = true
antenna_config = "vlba"
specific_mask = false
lon = -80
lat = 50
steps = 50
keep_fft_files = true
source_list = false
compressed = false
interpolation = false
multi_channel = false
bandwidths = 1
```

[mode]

quiet, Boolean Enable automatic overwriting of existing models

[paths]

data_path, String Path to train and validation data

data_format, Sting Data format: HDF5

[gaussians]

simulate, Boolean Enable extended sources simulation

num_components, List Minimum and maximum number of jet components [min, max]

[point_sources]

simulate, Boolean Enable point source simulation

add_extended, Boolean Add additional extended sources

[image_options]

bundles_train, Integer Number of training bundles

bundles_valid, Integer Number of validation bundles

bundles_test, Integer Number of test bundles

bundle_size, Integer Number of images per bundle

img_size, Integer Image size in pixels

noise, Boolean Enable noise corruption in image space

noise_level, Integer Standard deviation of Gaussian distribution used to draw noise values

white_noise, Boolean Enable noise corruption in visibility space

mean_real, Float Mean of Gaussian distribution used to draw noise values for the real part

std_real, Float Standard deviation of Gaussian distribution used to draw noise values for the real part

mean_imag, Float Mean of Gaussian distribution used to draw noise values for the imaginary part

std_imag, Float Standard deviation of Gaussian distribution used to draw noise values for the imaginary part

[sampling_options]

fourier, Boolean Enable data in Fourier space

real_imag, Boolean Save real and imaginary parts

amp_phase, Boolean Save amplitude and phase values

antenna_config, String Interferometer layout: VLBA

specific_mask, Boolean Simulate one specific mask

lon, Integer Longitude of the source

lat, Integer Latitude of the source

steps, Integer Number of time steps

keep_fft_files, Boolean Keep simulated FFT files

source_list, Boolean Save source list

compressed, Boolean Enable data compression

interpolation, Boolean Enable interpolation of missing pixels

multi_channel, Boolean Enable simulation of multiple bandwidths

bandwidths, Integer Number of bandwidths

3 Default Config: `radionets_training`

```
# This is a TOML document.

title = "Train configuration"

[mode]
quiet = true
gpu = true

[logging]
comet_ml = true
project_name = "thesis"
plot_n_epochs = 2
scale = true

[paths]
data_path = "./example_data/"
model_path = "./build/example_model/example.model"
pre_model = "none"
norm_path = "none"

[general]
fourier = true
amp_phase = true
source_list = false
arch_name = "SRResNet"
loss_func = "splitted_L1"
num_epochs = 300
inspection = true
output_format = "png"
switch_loss = false
when_switch = 25

[hypers]
batch_size = 100
lr = 1e-3

[param_scheduling]
use = true
lr_start = 1e-3
lr_max = 3e-3
lr_stop = 5e-4
lr_ratio = 0.25
```

[mode]

quiet, Boolean Enable automatic overwriting of existing models

gpu, Boolean Enable training on Graphics Processing Unit (GPU)

[logging]

comet_ml, Boolean Enable tracking with CometML

project_name, String CometML project name

plot_n_epochs, Integer Interval for plot creation

scale, Boolean Enable amplitude scaling

[paths]

data_path, String Path to train and validation data

model_path, String Path to save model weights

pre_model, String Path to pre-trained model, disable with ”none”

norm_path, String Path to normalization values, disable with ”none”

[general]

fourier, Boolean Enable data in Fourier space

amp_phase, Boolean Data set consisting of amplitude and phase values instead of real and imaginary parts

source_list, Boolean Enable source list training

arch_name, String Architecture

loss_func, String Loss function

num_epochs, Integer Number of training epochs

inspection, Boolean Enable inspection plots after training

output_format, String Format of inspection plots

switch_loss, Boolean Enable loss function switching

when_switch, Integer Number of epochs before switching loss function

[hypers]

batch_size, Integer Batch size

lr, Float Learning rate

[param_scheduling]

use, Boolean Enable learning rate scheduler

lr_start, Float Start learning rate

lr_max, Float Maximum learning rate

lr_stop, Float Stopping learning rate

lr_ratio, Float Ratio when the maximum learning rate is applied

4 Default Config: `radionets_evaluation`

```
# This is a TOML document.

title = "Evaluation configuration"

[mode]
quiet = true

[paths]
data_path = "./example_data/"
model_path = "./build/example_model/example.model"
model_path_2 = "none"

[general]
fourier = true
amp_phase = true
source_list = false
arch_name = "SRResNet"
arch_name_2 = "none"
output_format = "png"
diff = true

[inspection]
visualize_prediction = true
visualize_source_reconstruction = true
visualize_contour = true
visualize_dynamic_range = false
visualize_blobs = false
visualize_ms_ssim = false
random = false
num_images = 20

[eval]
batch_size = 100
save_vals = false
save_path = "./build/"
evaluate_viewing_angle = true
evaluate_dynamic_range = true
evaluate_ms_ssim = true
evaluate_mean_diff = true
evaluate_area = true
evaluate_point = false
predict_grad = false
evaluate_gan = false
```

[mode]

quiet, Boolean Enable automatic overwriting of existing models

[paths]

data_path, String Path to test data

model_path, String Path to model weights

model_path_2, String Path to a second model for separate predictions

[general]

fourier, Boolean Enable data in Fourier space

amp_phase, Boolean Data set consisting of amplitude and phase values instead of real and imaginary parts

source_list, Boolean Enable source list options

arch_name, String Architecture

arch_name_2, String Second architecture for separate predictions

output_format, String Data format of evaluation figures

diff, Boolean Enable visualization of the difference between prediction and simulation

[inspection]

visualize_prediction, Boolean Visualize model predictions

visualize_source_reconstruction, Boolean Visualize reconstructed source distribution

visualize_contour, Boolean Visualize contour lines of the reconstructed source distribution

visualize_dynamic_range, Boolean Visualize dynamic ranges

visualize_blobs, Boolean Visualize reconstructed components identified by blob detection

visualize_ms_ssim, Boolean Add calculated MS SSIM to visualization

random, Boolean Enable random test image choice

num_images, Integer Number of visualizations

[eval]

batch_size, Integer Batch size

save_vals, Boolean Enable saving of evaluation values

save_path, String Path to save evaluation values

evaluate_viewing_angle, Boolean Enable viewing angle evaluation

evaluate_dynamic_range, Boolean Enable dynamic range evaluation

evaluate_ms_ssim, Boolean Enable MS SSIM evaluation

evaluate_mean_diff, Boolean Enable mean difference of the core component evaluation

evaluate_area, Boolean Enable source area evaluation

evaluate_point, Boolean Enable point source evaluation

predict_grad, Boolean Enable model gradient evaluation

evaluate_gan, Boolean Enable GAN evaluation

5 Default Config: radiosim

```
# This is a TOML document.

title = "Simulation configuration"

[mode]
quiet = true

[paths]
outpath = "./build/example_data/"

[source_types]
jets = true
num_jet_components = [3, 6]
pointlike_gaussians = false
num_pointlike_gaussians = [2, 5]

[flux_scaling]
use_scaling = true
type = "mojave"

[image_options]
bundles_train = 50
bundles_valid = 10
bundles_test = 10
bundle_size = 200
img_size = 256
noise = false
noise_level = 5
```

[mode]

quiet, Boolean Enable automatic overwriting of existing simulations

[paths]

outpath, String Path to save simulations

[source_types]

jets, Boolean Enable simulation of jet sources

num_jet_components, List Minimum and maximum number of jet components [min, max]

pointlike_gaussians, Boolean Enable simulation of pointlike Gaussians

num_pointlike_gaussians, List Number of pointlike Gaussians [min, max]

[flux_scaling]

use_scaling, Boolean Enable scaling of simulated source images

type, String Scaling type: MOJAVE or normalization

[image_options]

bundles_train, Integer Number of training bundles

bundles_valid, Integer Number of validation bundles

bundles_test, Integer Number of test bundles

bundle_size, Integer Number of images per bundle

img_size, Integer Image size in pixels

noise, Boolean Enable pixel-wise noise

noise_level, Integer Standard deviation of Gaussian distribution used to draw noise values

6 Default Config: pyvisgen

```
# This is a TOML document.

[sampling_options]
mode = "basic"
layout = "vlba"
img_size = 128
fov_center_ra = [110, 130]
fov_center_dec = [20, 30]
fov_size = 0.1
corr_int_time = 30.0
scan_start = ["01-06-2021 18:00:00", "01-06-2021 23:00:00"]
scan_duration = [120, 150]
scans = [8, 10]
interval_length = 3600
base_freq = 15.21e9
frequsel = [0e8, 0.8e8, 1.44e8, 2.08e8]
bandwidths = [6.4e7, 6.4e7, 6.4e7, 6.4e7]
corrupted = true

[bundle_options]
in_path = "build/skies/"
out_path_fits = "build/uvfits"
out_path_gridded = "build/gridded"
num_test_images = 1000
bundle_size = 100
train_valid_split = 0.2
grid_size = 256
amp_phase = true
```

`[sampling_options]`

`mode, String` Simulation scheme: basic

`layout, String` Interferometer layout: Very Large Array (VLA), VLBA or Event Horizon Telescope (EHT)

`img_size, Integer` Set image size of sky simulations

`fov_center_ra, List` Minimum and maximum right ascension [min, max]

`fov_center_ra, List` Minimum and maximum declination [min, max]

`fov_size, Integer` FoV in mas

`scan_start, String` Integration time for one data point

`scan_duration, List` Minimum and maximum scan duration in seconds [min, max]

`scans, List` Minimum and maximum number of scans [min, max]

`interval_length, Integer` Interval length of one scan in seconds

`base_freq, Float` Base frequency of observation

`frequsel, List` List of frequency channels, values are added to the base frequency

`bandwidths, List` List of the channels' bandwidths

`corrupted, Boolean` Enable simulation using RIME for corrupted visibilities

`[bundle_options]`

`in_path, String` Path to input sky distributions

`out_path_fits, String` Path to save FITS files

`out_path_gridded, String` Path to save gridded HDF5 files

`num_test_images, Integer` Number of test images

`bundle_size, Integer` Number of images per HDF5 file

`train_valid_split, Float` Ratio of validation to training images

`grid_size, Integer` Grid size in pixels

`amp_phase, Boolean` Grid amplitude and phase values instead of real and imaginary parts

Eidesstattliche Versicherung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Dissertation mit dem Titel „Another One Cleans the Dust“ selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

Belehrung

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50 000.00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden (§ 63 Abs. 5 Hochschulgesetz –HG–).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z. B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen.

Ort, Datum

Unterschrift

Acknowledgements

First of all, I want to thank my supervisor Wolfgang Rhode. You made it possible for me to start a new physics field in Dortmund and supported me in making all the necessary contacts over the last five years by allowing me to travel a tremendous amount of time.
I want to thank Dominik Bomans for his interest in my deep learning-based cleaning approach and for taking on the role of the second referee.
Many thanks also to Dominik Elsaesser, who helped me at any time with words and deeds.
A big thank you goes to Marcus Brüggen for all the support. Thanks for your radio interferometry input, the countless discussions, and the dedication you put into our paper. I am already looking forward to the following joint projects.
Lena, thank you for starting the journey with me. Together we survived pure insanity, exploring the world of radio astronomy as complete beginners.
What began five years ago as a weekly trip to Würzburg evolved into the crazy idea to revolutionize the established analysis strategies in radio interferometry. Thank you, Matthias, for introducing me to the field with your friendly and helpful team.

I want to thank the E5b group for the pleasant and supportive working atmosphere.
A special thank goes to Felix, who supported me from the first minute. You and the rest of the `radionets` team have contributed significantly to the success of this thesis.
Starting a new analysis strategy is not easy all the time. Thanks for your motivation to continue, Kai, Richard, and Thorben, even when things did not work directly.
Thanks also to Andrea for the pleasant conversations in-between all the work and for keeping the bureaucracy away.

Thank you, Max, for your invaluable technical support. I guess I am another Kevin that owes you a beer for stealing your thesis template.

I want to thank the whole PeP team for all the pleasant hours inside and outside the university.

Thank you to my study group, Simone, Kevin, Helena, Maik, Quentin, Robin, Vukan, Lars, and Arne. You have made studying much more pleasant and saved me many times.

I want to thank my family for their assistance and understanding over the last few years. I know you have not seen me much in that time. Thank you for laying the foundation that made everything possible. Especially, I want to thank my dad for his endless support.

Mia and Andrea, you have reminded me that there is more in life than just physics.
I also want to thank Benny, Marc, and Max-Leon for their many years of friendship and support. Although we are all going our ways, we are still one unity when we come together.

My deepest gratitude goes to Marie for making every day a bright one for me. You know how to bring a smile to my face, and it touches me deeply how much you care for me. With you by my side, even writing a thesis is fun. Together we are unstoppable!

PS: This thesis would not have been possible without the endless motivation generated from the content by Q-Dance; WOW WOW!