technische universität
dortmund

One for All

A comprehensive analysis pipeline for simulating and
reconstructing radio observations including uncertainty estimations

Felix Geyer

2024

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. Anna Franckowiak

This thesis is set in Latin Modern and Fira Code,
typeset using LaTeX with LuaTeX from TeX-Live 2023.

Abstract

With the help of radio interferometry, humanity can probe the universe at the highest resolutions possible, enabling in-depth studies of the physical processes driving the known cosmos. In recent years, the need for an increased automatization of the analysis pipelines in radio interferometry arose, mainly due to the increased data rates of the currently used and the planned generation of radio interferometers. In several areas of science, this increased automatization is handled by deep learning techniques, namely neural networks. In this thesis, I introduce a comprehensive analysis pipeline comprising physics-based radio galaxy simulations and the analytical simulation of the measurement process of radio interferometers. Furthermore, several deep learning models are trained to reconstruct the simulated incomplete observations to obtain a cleaned version of radio galaxies. Utilizing several evaluation metrics, the training process is adapted and improved with regard to the goal of reconstructing a protoplanetary disk from the Disk Substructures at High Angular Resolution Project (DSHARP) data set. Additionally, the neural network approach is extended to include the estimated uncertainty of the prediction. Finally, I present a neural network-based reconstruction of the protoplanetary disk "Elias 24".

Zusammenfassung

Mit Hilfe der Radiointerferometrie kann die Menschheit das Universum mit der höchstmöglichen Auflösung erforschen, was tiefgreifende Studien der physikalischen Prozesse ermöglicht, die den bekannten Kosmos antreiben. In den letzten Jahren ist der Bedarf an einer verstärkten Automatisierung der Analysepipelines in der Radiointerferometrie entstanden, vor allem aufgrund der erhöhten Datenraten der derzeit verwendeten und der geplanten Generation von Radiointerferometern. In verschiedenen Bereichen der Wissenschaft wird diese verstärkte Automatisierung durch Deep-Learning-Techniken, insbesondere neuronale Netze, bewältigt. In dieser Arbeit stelle ich eine umfassende Analysepipeline vor, die sowohl physikalisch basierte Simulationen von Radiogalaxien als auch die analytische Simulation des Messprozesses von Radiointerferometern umfasst. Darüber hinaus werden mehrere Deep-Learning-Modelle trainiert, um die simulierten unvollständigen Beobachtungen zu rekonstruieren und eine bereinigte Version der Radiogalaxien zu erhalten. Unter Verwendung verschiedener Bewertungsmetriken wird der Trainingsprozess angepasst und im Hinblick auf das Ziel der Rekonstruktion einer protoplanetaren Scheibe aus dem DSHARP-Datensatz verbessert. Zusätzlich wird das neuronale Netz erweitert, um die Unsicherheit der Vorhersage zu schätzen. Schließlich stelle ich eine auf einem neuronalen Netz basierende Rekonstruktion der protoplanetaren Scheibe „Elias 24“ vor.

Contents

1 Introduction

The universe is an enormous, vast, and largely unexplored space. In the search for the origins of life and the cosmos itself, astronomy has played a pivotal role for quite some time. Utilizing different wavelengths, various experiments both on Earth and in the atmosphere, and multiple decades of measurement time, humankind is trying to study the universe at even greater distances. In recent years, advances in the field of computer-based analyses have enabled the combination of observations at different wavelengths for the same cosmic event or source. This technique opened up entirely new perspectives of previously studied objects and led to new discoveries and deeper insights into critical physical processes. Especially radio interferometry acts as a cornerstone for such analyses due to its high resolutions and the permeability of the atmosphere in the range of ≈ 1 mm to 10 m wavelengths. The requirements for a successful contribution to a multiwavelength analysis include a reproducible analysis, which can be carried out on a short timescale. In several areas of science, these requirements are met by applying methods from the field of deep learning. Another advantage of using deep learning algorithms gains relevance with regard to planned interferometers like Square Kilometre Array (SKA) [27] or next-generation Very Large Array (ngVLA) [75]: The proposed high data rates will require some form of automated analysis, which neural networks can perform. Thus, this thesis aims to provide a comprehensive, automated analysis pipeline that utilizes neural networks to reconstruct radio interferometer observations.

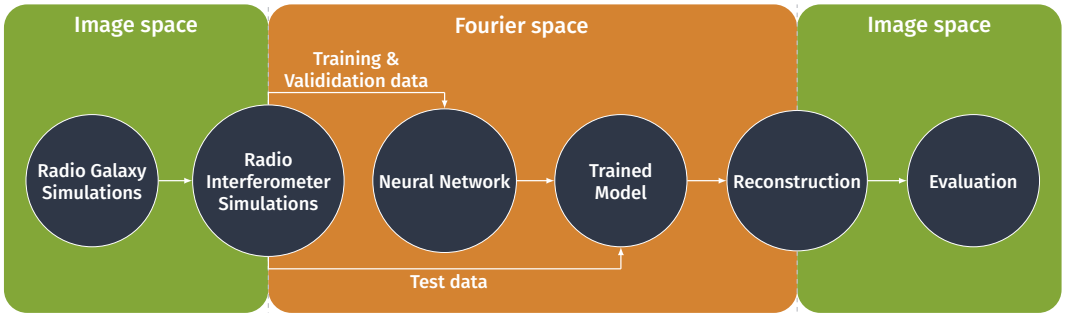

Figure 1.1: Overview of the analysis pipeline utilized in this thesis. The green parts take place in image space, while the orange parts are carried out in Fourier space.

The analysis pipeline created for and utilized in this thesis is visualized in Figure 1.1. It propagates from left to right and is divided into parts that take place in image space (green) and Fourier space (orange). The pipeline starts with the radio galaxy simulations, which can be any cosmic source visible in the radio regime. Then, the measurement process of a radio interferometer is simulated using an analytical description called the Radio Interferometer Measurement Equation (RIME), which is based on Jones matrices [58, 46]. The data simulated by that algorithm is located in Fourier space. After some processing steps, said data is split into training, validation, and test data sets in order to train a neural network model for the task of reconstructing the simulated data in Fourier space. As the last step, the reconstruction capability

of the trained model is evaluated both in Fourier and image space using several techniques; for example, it is examined whether the reconstructed flux densities and the source structures match the simulated ones. In the following chapters, each part of the analysis chain will be elaborated in detail. Parts of the ideas and results presented in this thesis have been published in Astronomy & Astrophysics [39].

In earlier work [100, 97], the foundations for using neural networks to reconstruct radio observations were laid, and important initial findings were made. Two-dimensional Gaussian distributions were positioned on a two-dimensional grid in order to create naive first simulations of radio galaxies. Furthermore, the measurement process of a radio interferometer and the resulting gaps of missing information were mimicked by generating Boolean masks, which were applied to the Fourier-transformed Gaussian distributions. Additionally, different architecture types were tested, and the first approaches to an uncertainty estimation were investigated. In this thesis, I want to improve upon and expand these ideas as well as present new approaches.

For this purpose, the analytical description of the measurement process of a radio interferometer, the RIME [102], is combined with approaches from the field of deep learning, Super Resolution Residual Networks (SRResNets) [66], among others, in order to form an analysis pipeline and to reconstruct interferometric measurements. Different types of cosmic sources and radio observatories are used for the simulations, as well as various techniques from the area of machine learning. In Chapter 2, the basics of radio interferometry and the mathematical foundation of the physics-based interferometric measurement simulations are described. Additionally, radio observatories that are significant for this work are introduced. Chapter 3 focuses on the theory of machine learning and deep learning techniques utilized in this thesis. In detail, Convolutional Neural Networks (CNNs) are explained. The radio galaxy simulations and radio interferometer simulations are established in Chapter 4. Next, the architecture and other specifics of the neural network model are specified in Chapter 5. Furthermore, example reconstructions and loss curves are shown for selected models. The more in-depth evaluation of said models takes place in Chapter 6. For this, various methods are introduced and utilized, for example, examining the source area or the Multi-Scale Structural Similarity (MS-SSIM) index. As an additional feature, the trained neural network models are also enabled to estimate the uncertainty of the reconstruction, which is described in Chapter 7. The previous chapters' results and knowledge are then combined to reconstruct interferometric measurements observed for the DSHARP, presented in Chapter 8. Finally, the summary of this thesis and future plans are listed in Chapter 9.

2 Modern Radio Astronomy

In modern astronomy, the combination of measurements with different wavelengths and observatories both in space and on Earth is becoming increasingly popular, among others, due to the advancing digitalization of measurement and analysis techniques. Radio astronomy plays a vital role in such campaigns due to its high-resolution measurements, e.g., in the multi-messenger follow-up observations of a binary neutron star merger event in 2017, which was triggered by the detection of gravitational waves [1]. This chapter explains the theory of modern radio astronomy and the implementation of the software used to simulate radio observations. In Section 2.1, the basics of radio interferometry are described using the RIME. The implementation of the RIME used for this work is elaborated in Section 2.2. In addition, two pairs of radio observatories and cosmic sources that are important for the context of this work are presented in Section 2.3 and Section 2.4, respectively.

2.1 Basics of Radio Interferometry

The probably most fundamental equation of radio interferometry is also the reason for the high resolutions achievable when utilizing interferometry techniques: the Rayleigh criterion [69]

$$\Theta \approx 1.22 \cdot \frac{\lambda}{D} \,. \tag{2.1}$$

It states that the angular resolution Θ, i.e., the minimal distance for which two separate objects are still distinguishable, is minimized if either the observing wavelength λ gets smaller or the diameter of the collecting area D is increased. Since the range of possible wavelengths is limited to a range of mm to m, the easiest way of minimizing Θ in (2.1) is to increase the diameter D. Experiences like the collapse of the 300-foot telescope in Green Bank, West Virginia, in 1988 [79] showed that the collecting area of a single-dish telescope is also limited due to technical reasons. Thus, the idea was to combine two or more radio antennas to form an array and combine the signals using methods from interferometry. In 1946, Ryle and Vonberg designed the first radio interferometer consisting of two antennas [93]. Especially with the invention of Very Long Baseline Interferometry (VLBI) in the 1970s [20], the limits of angular resolution were shifted again. By combining multiple antennas with distances of several thousand kilometers, angular resolutions down to milliarcseconds are possible, e.g., with the Very Long Baseline Array (VLBA) [76]. These high resolutions come at the cost of sparse data coverages due to the limited number of antennas [74].

The theoretical concepts of radio interferometry can be best described by combining the explanations from Smirnov [102] and Noordam [78] while utilizing the RIME. Furthermore, the following paragraphs in this section are also based on the descriptions of Thompson et al. [106]. For a better understanding of the dimensionalities, vectors are symbolized using a vector arrow $\vec{x}$ and matrices using bold symbols $\mathbf{X}$.

Initially, it is assumed that the signals are quasi-monochromatic and that the source is distant enough to consider plane waves. These assumptions are commonly made in radio interferometry. Also, for the sake of simplicity, the number of antennas is restricted to two. However, the

concept is easily expandable to more than two antennas because a radio interferometer array acts as multiple two-interferometer pairs. Then, the signal of a single point source fulfilling the assumptions from above emits a complex electromagnetic signal corresponding to

$$\vec{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} \tag{2.2}$$

in a (x, y, z) coordinate system, with z pointing in the direction of the source. In order to describe the effects influencing the signal on its way to the receivers of the antennas, the Jones calculus [58] with its Jones matrices $\mathbf{J}$ [46] is used. Arriving at the antennas, the receiver converts the signal $\vec{e}$ to a complex voltage $\vec{v}$ defined as

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \mathbf{J}\vec{e}\,, \tag{2.3}$$

resulting in $\vec{v}_p$ and $\vec{v}_q$ for the two antennas p and q. Since the signal is received at two spatially separated locations, it needs to be correlated. For this scenario, this leads to four pairwise correlations, which define the so-called visibility matrix $\mathbf{V}_{pq}$

$$\mathbf{V}_{pq} = 2 \begin{pmatrix} \langle v_{p1} v_{q1}^* \rangle & \langle v_{p1} v_{q2}^* \rangle \\ \langle v_{p2} v_{q1}^* \rangle & \langle v_{p2} v_{q2}^* \rangle \end{pmatrix} = 2 \left\langle \vec{v}_p \vec{v}_q^{\,H} \right\rangle \,. \tag{2.4}$$

Here, $\langle \cdot \rangle$ corresponds to an averaging over time and frequency, $\vec{v}^*$ denotes the complex conjugate of $\vec{v}$, and $\vec{v}^H$ marks the conjugate transpose. Utilizing the definition of the visibility matrix (2.4) and the introduction of the brightness matrix $\mathbf{B}$

$$\mathbf{B} = 2 \begin{pmatrix} \langle e_x e_x^* \rangle & \langle e_x e_y^* \rangle \\ \langle e_y e_x^* \rangle & \langle e_y e_y^* \rangle \end{pmatrix} = \begin{pmatrix} I + Q & U + iV \\ U - iV & I - Q \end{pmatrix} \,, \tag{2.5}$$

the basic RIME can be stated as

$$\mathbf{V}_{pq} = 2 \left\langle \mathbf{J}_p \left(\vec{e}\, \vec{e}^{\,H} \right) \mathbf{J}_q^H \right\rangle = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^H \,. \tag{2.6}$$

The brightness matrix $\mathbf{B}$ in (2.6) corresponds to the brightness distribution of the sky and are directly correlated to the Stokes parameters I, Q, V and U [71, 45].

The coordinate systems in which the equations above operate are illustrated in Figure 2.1. The two antennas p and q have a spatial distance called the baseline $\vec{b}$. Each baseline corresponds to one visibility in the Fourier space. The section of the sky to be observed operates in a (m, n, l) Cartesian coordinate system with n pointed at the observer. Because of the long distance between the observer and the source, the signal is assumed to arrive as a plane wave. Due to the baseline $\vec{b}$, the signal has to propagate different geometric path lengths in order to reach both antennas. This creates a phase delay τ_g, which has to be corrected for. Thus, the (u, v, w) coordinate system is defined, where w is pointed to the source and the (u, v) plane features the corrected, projected baseline $\vec{b}_{\text{proj}}$. By definition, the (u, v) plane is parallel to the plane wave coming from the source. Both coordinate systems (m, n, l) and (u, v, w) are direction cosines [107].

Next, it is shown how the fundamental equation of radio interferometry, the van Cittert-Zernike theorem [108], which describes the relation between the measured signal and the observed sky, is associated with the RIME. For this, the basic RIME in (2.6) has to be extended in order to consider more occuring effects. First, the phase delay τ_g needs to be represented, as it arises for every measurement and every single antenna pair, as illustrated in Figure 2.1. The phase delay is expressed in relation to a so-called phase center, introduced by the correlator during

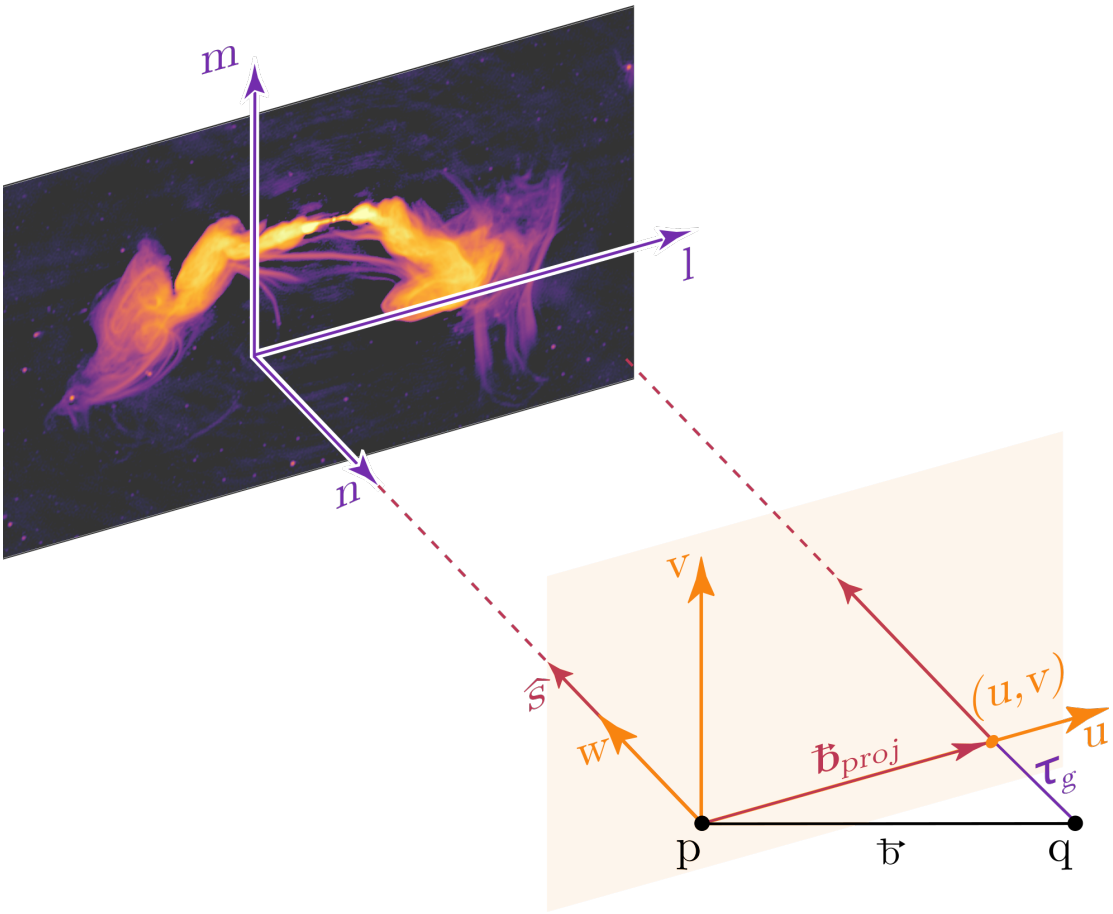

Figure 2.1: Visualization of the different coordinate systems used in radio interferometry. A cosmic source in a (m, n, l) coordinate system is observed by a two-antenna interferometer with the antennas p and q, which have the baseline $\vec{b}$. Here, the n axis is pointed to the observer. Due to the spatial distance between the antennas, the assumed quasi-monochromatic signal arrives with a phase delay τ_g at antenna q. Thus, the baseline $\vec{b}$ is adapted, resulting in the projected baseline $\vec{b}_{\text{proj}}$ in the antenna's coordinate system (u, v, w) with the (u, v) plane parallel to the plane wave coming from the source. The w axis is perpendicular to said plane wave. Both coordinate systems (m, n, l) and (u, v, w) are direction cosines [107]. Source image taken from [88], schematic from [97].

an observation, and is set to $(u, v) = (0, 0)$ for this example. If the antenna p is located at $\vec{u}_p = (u_p, v_p, w_p)$, the phase difference k_p at point $\vec{u}_p$ relative to the phase center is given by

$$k_p = 2\pi\lambda^{-1}\left(u_p l + v_p m + w_p(n-1)\right) . \tag{2.7}$$

Here, the coordinates $(l, m, n = \sqrt{1 - l^2 - m^2})$ are the direction cosines of the signal direction $\vec{\sigma}$ and λ is the signal's wavelength. Because it is conventional to express $\vec{u}$ in units of wavelength, the wavelength term in (2.7) can be dropped. Since the phase difference in (2.7) is a linear transformation of the signal, a scalar Jones matrix constituting the phase delay

$$\mathbf{K}_p = \mathrm{e}^{-ik_p} = \mathrm{e}^{-2\pi i(u_p l + v_p m + w_p(n-1))} \tag{2.8}$$

can be introduced. Thus, the RIME for a single point source without any corruption effects is defined as

$$\mathbf{V}_{pq} = \mathbf{K}_p \mathbf{B} \mathbf{K}_q^H . \tag{2.9}$$

Of course, corruption effects occur during a radio observation. Each considered effect is represented by its own Jones matrix, creating a so-called Jones chain. These effects can be split into two basic categories:

- The source- and direction-independent effects $\mathbf{G}_p$, which are the same for all sources and all directions, for example, effects at the antenna, such as the electronic gain
- The source- and direction-dependent effects $\mathbf{E}_{sp}$, which vary for each observation and include, for example, atmospheric corrections

Also, the observed sky features not only one but many point sources. Thus, the basic RIME is adapted to the contribution of multiple point sources and corruption effects. Each source s contributes with its own phase delay $\mathbf{J}_{sp} = \mathbf{K}_{sp}$ to (2.9), and since these contributions can be linearly added, (2.6) becomes

$$\mathbf{V}_{pq} = \sum_s \mathbf{J}_{sp} \mathbf{B}_s \mathbf{J}_{sq}^H = \sum_s \mathbf{K}_{sp} \mathbf{B}_s \mathbf{K}_{sq}^H . \tag{2.10}$$

Furthermore, if the source-dependent and -independent effects are also taken into account, $\mathbf{J}_{sp}$ becomes a chain of Jones matrices

$$\mathbf{J}_{sp} = \mathbf{G}_p \mathbf{E}_{sp} \mathbf{K}_{sp} , \tag{2.11}$$

resulting in (2.10) extending to

$$\mathbf{V}_{pq} = \sum_s \mathbf{G}_p \mathbf{E}_{sp} \mathbf{K}_{sp} \mathbf{B}_s \mathbf{K}_{sq}^H \mathbf{E}_{sq}^H \mathbf{G}_q^H = \mathbf{G}_p \left(\sum_s \mathbf{E}_{sp} \mathbf{K}_{sp} \mathbf{B}_s \mathbf{K}_{sq}^H \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H \tag{2.12}$$

Equation (2.12) now can be related to the van Cittert-Zernike theorem by applying the following steps:

- Generalizing the sky consisting of point sources to a sky of continuous brightness distribution $\mathbf{B}(\vec{\sigma})$, where $\vec{\sigma}$ is the unit direction vector. This integration over all directions results in a spherical, three-dimensional integral
- Transforming said spherical integral onto the (l, m) plane tangential at the field center using a sine projection, thus changing the dimensionality of the integral from three to two
- Substituting the non-coplanarity term of the integral into the direction-dependent matrices of the Jones chain. This helps to identify a part of the equation as a two-dimensional Fourier transformation, which is the central part of the van Cittert-Zernike theorem

After conducting these steps, the resulting full-sky RIME looks like this:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\int_l \int_m \mathbf{E}_p(l,m) \mathbf{B}(l,m) \mathbf{E}_q(l,m) \exp\left(-2\pi i \left[u_{pq} l + v_{pq} m\right]\right) \mathrm{d}l \, \mathrm{d}m \right) \mathbf{G}_q^H \tag{2.13}$$

The main conclusion of (2.13) is that the brightness distribution $\mathbf{B}(l,m)$ is connected to the measured visibilities $\mathbf{V}_{pq}$ of the antenna pair pq via a two-dimensional Fourier transformation, which is the primary statement of the van Cittert-Zernike theorem [108].

Since each baseline corresponds to one complex visibility in the (u,v) space, the number of baselines is crucial for coverage of the (u,v) space. As mentioned before, the number of radio antennas per observatory is limited; thus, the coverage is sparse. Due to Earth's rotation, which alters the projected baseline shown in Figure 2.1 over time, the corresponding visibilities slightly change during a designated observation period. This way, different, neighboring parts of the (u,v) space are sampled during an observation with the same baseline, leading to improved coverage. However, parts of the (u,v) space cannot be sampled, e.g., due to the layout, which results in missing information. This missing information creates artifacts and noise after applying the Fourier transformation, producing a so-called dirty image, which is not suitable for physical analysis. Thus, the need for cleaning or reconstruction of the missing information in the (u,v) space arises, which will be performed by neural networks in this thesis.

2.2 Implementation of the RIME

As explained in Section 2.1, the RIME produces complex visibilities. However, before they can be analyzed by established imaging software like the w-Stacking Clean (WSCLEAN) algorithm or the neural network-based approach presented in this thesis, several data processing steps need to be applied, which will be summarized in the following paragraphs.

In general, data measured by radio interferometers needs to be calibrated to account for, e.g., different antenna gains [86, 89]. For all data produced by the RIME in this thesis, the calibration is omitted. The reason is that the sky simulations used as input for the RIME in Section 4.1 and Section 4.2 are assumed to be calibrated. Also, the RIMEs do not feature any Jones matrices connected to calibration. Thus, when applying the trained models to measured radio interferometer data, it needs to be kept in mind that this data needs to be fully calibrated for the network to be able to reconstruct the missing information.

Furthermore, the data generated by the RIME is stored in a particular format, the Flexible Image Transport System (FITS). This data format was designed specifically to serve the needs of interferometer sky surveys. In order to be able to analyze RIME-generated data not only with neural networks but also with established software, the complex visibilities are saved using a FITS-writer implemented according to the Astronomical Image Processing System (AIPS) Memo 117 [43].

Eventually, one last step is necessary before the data is prepared to be fed to the analysis pipelines: the gridding [19]. This method arranges the irregularly spaced data into an evenly spaced grid. In particular, this offers the advantage that a Fast Fourier Transformation (FFT) can be used instead of a Discrete Fourier Transform (DFT) to perform the Fourier transformation required for the established software. Especially for grid sizes of 2^n, the FFT outperforms the DFT in both computation time and cost [83]. There are multiple approaches to gridding, for example, the convolutional resampling used by WSCLEAN [84]. However, for this thesis, a binning approach using a two-dimensional histogram was chosen for simplicity. More details about the gridding performed for this work are described in Section 4.1.

All RIME simulations presented in this thesis are produced using the `pyvisgen` [98] package. It features a GPU-accelerated implementation of the RIME with multiple modifiable parameters,

e.g. the antenna layout, in order to simulate a radio interferometer observation. Furthermore, it is possible to consider the effects of different types of corruption. Because it is assumed that the input data for the RIME is calibrated, only direction-independent effects are considered. Additionally, the integral in (2.13) is discretized because the sky distribution $\mathbf{B}(l, m)$ is an image consisting of discrete pixels. Thus, the RIME implemented in `pyvisgen` is defined as:

$$\mathbf{V}_{pq} = \sum_{l,m} \mathbf{E}_p(l, m)\mathbf{K}_p(l, m)\mathbf{B}(l, m)\mathbf{K}_q^H(l, m)\mathbf{E}_q^H(l, m) \tag{2.14}$$

$\mathbf{K}_{p,q}(l, m)$, defined in (2.8), accounts for the phase delay, while $\mathbf{E}_{p,q}(l, m)$ encodes the direction-independent antenna beam, which can be expressed as

$$\mathbf{E}_{p,q}(l, m) = \text{jinc}\left(\frac{2\pi}{\lambda} d \cdot \theta_{lm}\right), \text{ with jinc} = \frac{J_1(x)}{x}. \tag{2.15}$$

Here, d corresponds to the antenna diameter, θ_{lm} marks the angular distance between the pointing position and the source structure, and $J_1(x)$ is the Bessel function of the first kind [14, 87]. λ stands for the observing wavelength. A visual representation of $\mathbf{K}_{p,q}(l, m)$ and $\mathbf{E}_{p,q}(l, m)$ is shown in Figure A.4 and Figure A.3, respectively. For this work, the data is assumed to be unpolarized. Thus, the generated visibilities are consistent with Stokes I.

2.3 Radio Interferometry Arrays

In this thesis, data from different radio observatories is used. In the following, a brief introduction to the Karl G. Jansky Very Large Array (VLA) and the Atacama Large Millimeter/submillimeter Array (ALMA) is given.

2.3.1 VLA

The VLA consists of 27 25 m-antennas arranged in a Y shape. It is located on the plains of San Augustin in New Mexico and designed to map the sky at wavelengths of 0.01 m to 0.1 m. Furthermore, it reaches a sensitivity of approximately 100 µJy and maximal resolution of 0.01″ [109]. A particular feature of the VLA is the movable antennas, which are mounted on rails. This way, it is possible to vary the distances between the telescopes, thus impacting the number of longer and shorter baselines. There are four configurations, with A being the one with the longest and D the one with the shortest baselines. The configurations are periodically cycled trough with a period length of 16 months [80]. An overview of the arrangement of the antennas is shown in Figure 2.2. Currently, a successor experiment called ngVLA is planned, which is proposed to be built at the site of the VLA and is expected to improve both sensitivity and spatial resolution by a factor of 10 [75].

2.3.2 ALMA

In this thesis, also data taken by the ALMA telescopes is analyzed. The array is located in the Atacama desert in Chile at an elevation of 5000 m. Due to the very high and dry location, disturbance effects like water vapor are minimal. There are 192 antenna foundations for 66 antennas, as visible in Figure 2.3. Thus, the layout and the baseline length are variable. The length of the baselines ranges from a few meters to a maximum of 18 km. [118]

Due to the discovery of multiple molecules in the interstellar medium, concepts for interferometers capable of operating in the millimeter regime were developed. ALMA was one of the proposed projects to be funded and is capable of measuring in the range of 31 GHz to 950 GHz [118]. The reason why ALMA was chosen for this thesis is its DSHARP measurements. For

Figure 2.2: Aerial overview of the VLA. The antennas are mounted on rails and arranged in a Y formation. [82]

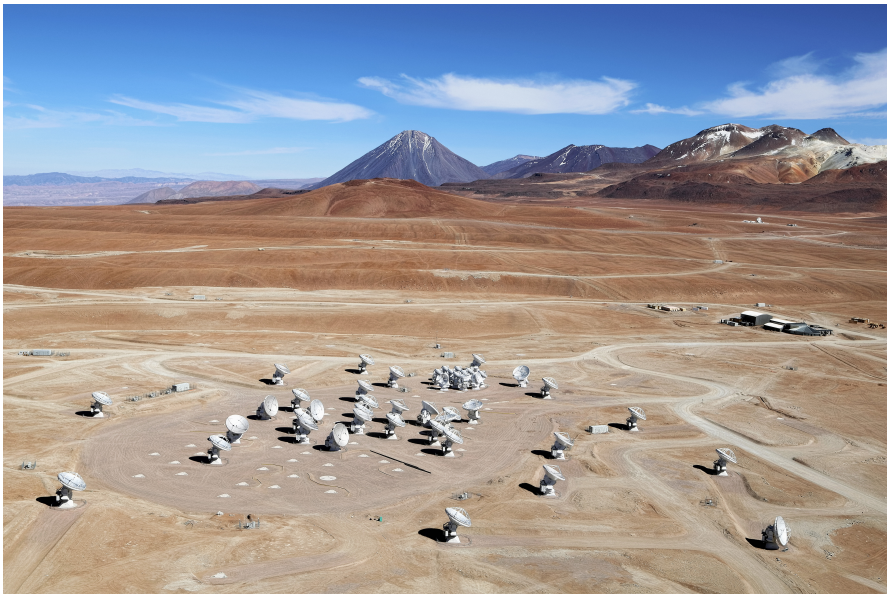

Figure 2.3: An overview of the Chajnantor Plateau in the Atacama desert in Chile. Several of the 66 antennas and 192 antenna foundations can be seen. The antennas can be repositioned to vary the interferometer layout and baseline lengths. [24]

this project, 20 protoplanetary disks were observed and analyzed. Due to the relatively simple yet variational appearance of these disks and the public availability of the data, the DSHARP observations were chosen as a simulation foundation. The final goal is to train a neural network with simulated protoplanetary disks and reconstruct the measured interferometer data with said network. The simulation process is explained in Section 4.2, while the training and reconstruction are examined in Chapter 8.

Figure 2.4: The ALMA telescopes by night. The small and the large Magellanic clouds are visible in the center of the image [34].

2.4 Observation Objectives

There are many cosmological sources or events that produce radio emissions, which are subject to radio interferometry investigations, such as neutron stars [110], gamma-ray bursts [38] or black holes [105]. Due to reasons explained below and in Section 4.1 and Section 4.2, Active Galactic Nuclei (AGN) and protoplanetary disks were selected to be examined in this thesis. The following subsections give an insight into the theory of the two cosmological objects.

2.4.1 AGN

AGN are among the most interesting and most studied objects in the Galaxy. They are visible in the entire spectrum of electromagnetic radiation, oftentimes the brightest objects visible in certain wavelengths. Furthermore, they act as large particle accelerators because the core component of an AGN is a supermassive black hole in the center of a host Galaxy. This black hole is accreting matter from the host Galaxy, forming a hot accretion disk and an enclosing dust torus. Additionally, this results in particle outflows called jets [8]. A visualization of the structure is shown in Figure 2.5.

Since the core and the jets are the most noticeable emission regions of AGN in the radio regime, brightness differences are indicated by the classification scheme of Fanaroff and Riley. If the core of the AGN is brighter than the jets, the Galaxy is classified as an Fanaroff-Riley Class I (FRI). If the core appears faint and the jets or lobes are intensively visible, the Galaxy

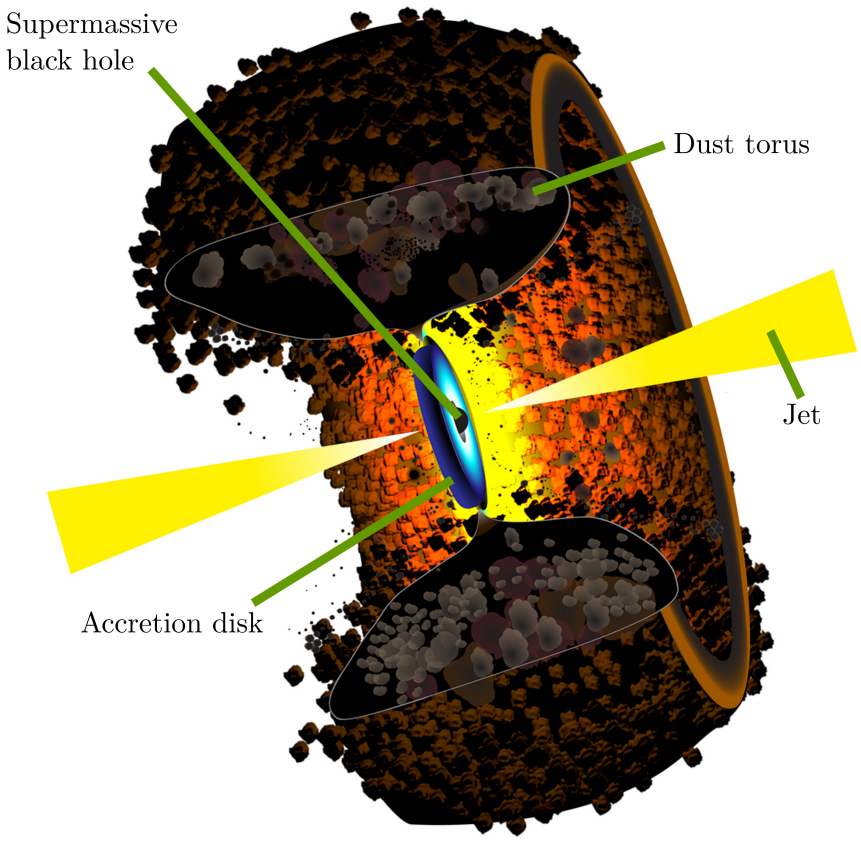

Figure 2.5: Artistic design of a AGN. The supermassive black hole in the center is surrounded by an accretion disk, which in turn is enclosed by a dust torus. Perpendicular to the axis of the accretion disk, jets emerge from the black hole. Image taken and adapted from [95].

qualifies as an Fanaroff-Riley Class II (FRII) [36]. There are other classification types of AGN based on the orientation of the jet relative to the observer's line of sight [112, 77]. However, they are omitted since they are not relevant to this thesis. Furthermore, some AGN appear to feature only one jet, which is impossible due to symmetry reasons. This effect is based on the relativistic boosting of the jet and the observation angle. If one jet points in the vicinity of the observer's line of sight, the particles in the jet experience a high relativistic boosting and thus produce high emissions. However, the counter-jet is boosted away from the observer, and thus, the emission from this jet is hard to measure, resulting in a AGN with seemingly just one jet [18].

The main reason for emission in radio wavelengths stemming from AGN is the synchrotron radiation of relativistic electrons. These electrons are boosted to relativistic speeds inside the jet and deflected by magnetic fields, resulting in energy losses by synchrotron emission perpendicular to their velocity [32, 11]. The most commonly used models of relativistic jets and the physical processes inside them were devised by Blandford and Königl [12, 63] and state that the stable extension of those jets can reach distances of kiloparsecs. Furthermore, their shape resembles that of a narrow cone with continuous plasma inflows.

Since AGN rank among the most observed sources of radio emission and both have already been [68] and will be [56] the subject of extensive surveys, the ability of reliably reconstructing measurements of these sources on a high-level is a crucial trait to be desired for the neural network-based reconstruction. Thus, the FRI and FRII subclasses of AGN are used as a basis for simulating training data for the neural network, explained in Section 4.1.

2.4.2 Protoplanetary Disks

Protoplanetary disks are part of the planet-forming process, consisting of dust and gas. They receive their form from the conservation of angular momentum during star formation through gravitational collapse. The star in the center of the disk is called the protostar, and its accreting matter from the surrounding molecular cloud forms the protoplanetary disk. Furthermore, it has not yet reached hydrostatic equilibrium [48]. The disk temperature varies with the distance to the protostar: hot near the star and cooler at the edges. Thus, the emission from the disks has wavelengths ranging from µm to mm and is therefore visible to radio interferometers. Additionally, the disks persist longer than their natal cores and thus can be studied extensively. [119]

The following is a briefly introduces the main processes of planet formation, resulting in smaller terrestrial planets and larger gas giants within protoplanetary disks. The development process of the terrestrial planets in the inner Solar System is described by the planetesimal hypothesis. The information this theory is based on was gathered from different areas: the structure of the Solar System, the chemical and isotopic compositions of Earth, Mars, and meteorites as well as the astronomical observations of young stars. The starting point is a young sun surrounded by a protoplanetary disk, which consists of gas and dust, as mentioned above. The dust grains merge and form mountain-sized bodies called planetesimals. These planetesimals themselves combine via collisions and gravitational interactions, resulting in larger planetary embryos. Typically, a low double-digit number of objects with a size somewhere between the Moon and Mars are developed during a period of 10^5 to 10^6 years. Eventually, also the planetary embryos collide due to their mutual gravitational force and form terrestrial planets in a period of 10×10^6 to 100×10^6 years. Earth's moon is probably a result of these late-stage collisions. [22]

For the process of forming large gas giants, there are two main options, which depend on the radius of the protoplanetary disk: For radii $r > 100\,\text{au}$, the development process is driven by the gravitational instability model, and for $r < 100\,\text{au}$, the core accretion model is the dominating process [13]. For the latter, the beginning is similar to the planetary embryos explained above. If the mass of the solid core exceeds a certain critical mass (roughly ten times the mass of the Earth),

the gravitational force of the core becomes strong enough for rapid gas accretion. Eventually, this leads to the formation of a gaseous planet. However, this model is ineligible to explain the existence of giant gas planets like Saturn or Jupiter. The reason for that is that the minimal time necessary to obtain the critical mass for the core lies in the middle of the period at which young, solar-type stars lose their gaseous disk [103]. If the gas of the protoplanetary disk has disbanded by the time the critical mass is achieved, then planets like Uranus form instead of Jupiter- or Saturn-like objects [15]. For planets of the latter type, the gravitational instability model is utilized. The process only occurs in regions of the protoplanetary disk that are sufficiently cool or develop a high enough surface density [29]. It is similar to the star formation process and is triggered if the initial gas density is so high that a dynamical gravitational instability occurs. This, in turn, leads to the collapse of a local patch of the disk. Afterwards, gas is again accreted onto the core, and a large gas giant forms. [6, 15]

Nevertheless, some physical details of all the above models are subject to debate. Thus, high-resolution measurements of protoplanetary disks can help improve our understanding of the protoplanetary disks and the planet formation processes occurring inside. This is one of the reasons why one basis for the simulations created in this thesis is the DSHARP measurements conducted by ALMA. Through improvements to the data processing pipelines by the neural network-based approach, the goal is to contribute to an enhanced analysis of protoplanetary disks.

3 Machine Learning

In most parts of science, machine learning methods are well-established methods, such as biology [42], chemistry [73], and various areas of physics [96, 60, 100]. In the following paragraphs, the intent is to explain the basics of machine learning and then switch to the specifics of neural networks, the backbone of this thesis.

3.1 Basics

The basic idea of machine learning is that a computer can be taught to learn to solve a specific problem. The learning process is called training, and the corresponding algorithm is called the learner. To aid and alter the training process, one can tune so-called hyperparameters to map better and fit the task at hand. The trained machine learning algorithm is also called a model.

For a deeper look into different types of machine learning algorithms, one needs to distinguish between unsupervised and supervised training processes. The difference lies in the additional attribute the data needs to have: The so-called label. These labels function as ground truth and often descend from simulation processes. In supervised training, like the ones used in this thesis, labels are used to evaluate the success of each step during training by statistically comparing the altered input data to the pre-existing label. Examples of this are random forests [17] or neural networks. Contrary to this, unsupervised algorithms do not require labels and do not compare the input data with a ground truth during training. Instead, for example, clustering algorithms like k-Means [67] use the given information to find similarities and group the input data. [2] Furthermore, there are more specific types of learning processes, such as reinforcement learning [28], which will not be discussed in detail here.

Next, some basic definitions are introduced, which are going to be helpful when the theoretical definitions of neural networks are explained in Section 3.2. The first definitions are training, validation, and test data set. These are used when conducting a supervised training process. The training data set is actively utilized during training. It is the basis for the adjustments to the hyperparameters of the algorithm. In contrast, the validation data set is used to compute the optimization metric and is not directly involved in the training process. After the training is completed, the test data set is used to evaluate the completed training as a whole, as opposed to the validation data set used in each step during training. Evaluating the training process leads to the following essential definitions: Over- and underfitting. Overfitting occurs when the machine learning algorithm memorizes the training data set instead of extracting features from it and generalizing it. Various effects can trigger this, for example, poorly chosen hyperparameters, insufficient data set splitting, or too many adjustable parameters in the algorithm. Contrary to that, underfitting occurs when the machine learning algorithm is too simple or lacks adjustable parameters, so it is impossible to handle the complexity of the given problem. A graphic visualization of over- and underfitting is shown in Figure 3.1. The graphic shows data points (black), which are approximated by three different approaches:

- A simple linear equation (blue), which is too simple for this kind of distribution and therefore underfits the problem

- A high dimensional power function (red), which exactly follows each data point and is not suited to describe a similar distribution of data points, which is called overfitting
- A lower dimensional power function (green), which follows the general trend of the data distribution and can be considered a reasonable fit

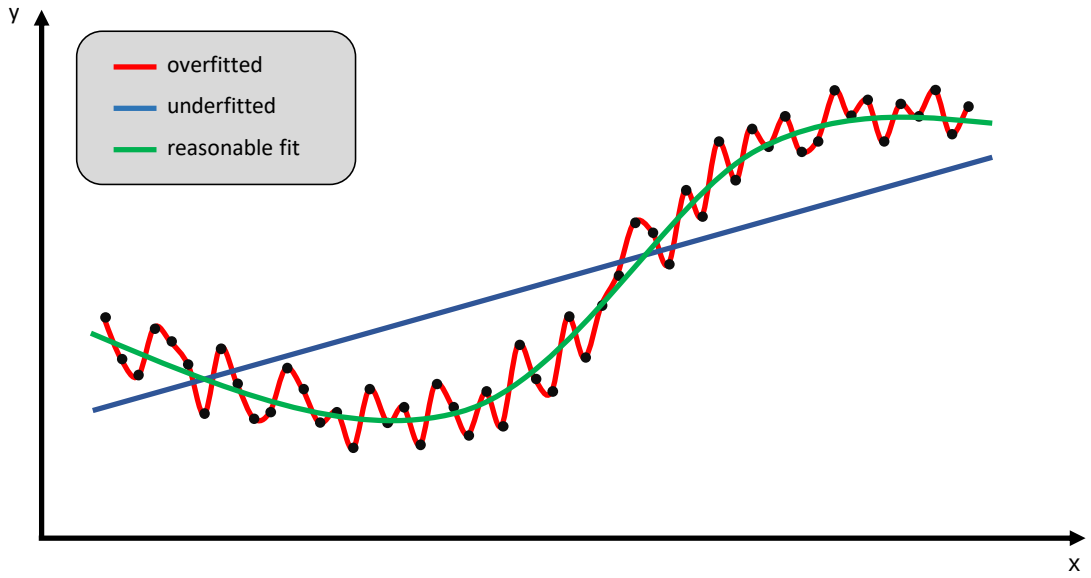

Figure 3.1: Visualization of over- and underfitting. The data points (black) are approximated by a linear equation (blue), a high dimensional power function (red) and a lower dimensional power function (green), which underfit, overfit and reasonably fit the distribution of data points, respectively.

In particular, overfitting often occurs when using neural networks. Two approaches to tackle this are regularization and data augmentation. When using regularization, one limits the range of some hyperparameters and effectively reduces the parameter space available to the model. Data augmentation, on the other hand, is utilized to alter the training data set by, e.g., rotating or masking it. This increases the parameter space of the data and helps the model generalize better by providing a variety of inputs. The regularization and data augmentation used in this thesis are specified in Section 5.1.

3.2 Neural Networks

Neural networks are part of machine learning algorithms and have been on the rise in almost every area of science since the mid-2000s when computational power increased dramatically. The algorithm is modeled after the human brain, so the innermost components are so-called "neurons" [28]. There are many different types of neural networks, so this section is centered around the basics, which are similar to most kinds of networks. In Section 3.3, the focus lies on CNNs, which is the kind used in this thesis.

One of the simplest forms of a neural network is a fully connected network consisting of three sets of neurons called layers. Such a simple fully-connected network is visualized in Figure 3.2. Each neuron owns a set of weights $\mathbf{W}$, so the modulation of an input value $\vec{x}$ by the neuron can

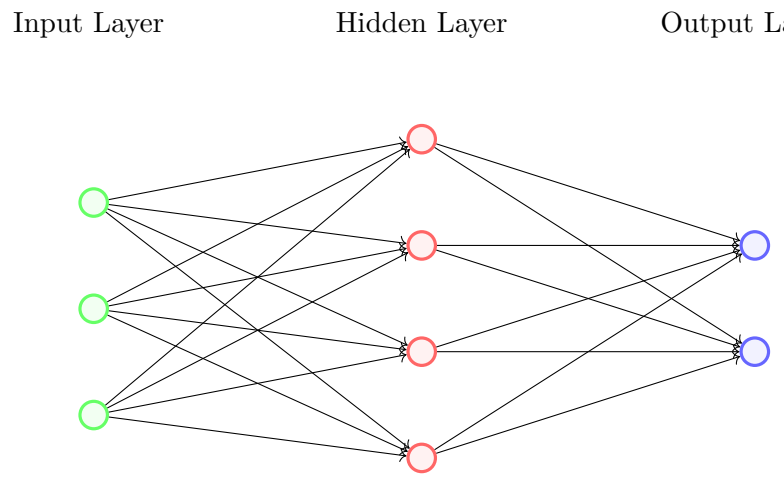

Figure 3.2: Schematic representation of a simple fully connected neural network. The green circles represent the neurons of the input layer, the red ones neurons from the hidden layer and the blue ones neurons from from the output layer.

be expressed like this

$$\vec{y} = \mathbf{W} \cdot \vec{x}\,, \tag{3.1}$$

where $\vec{y}$ is the output of the neuron. So a set of two-neurons, with $\vec{x}_{1,2}, \vec{y}_{1,2}$ and $\mathbf{W}_{1,2}$ respectively and the second neuron being fed the output of the first ($\vec{y}_1 = \vec{x}_2$) would look like this

$$\vec{y}_2 = \mathbf{W}_2 \cdot \vec{x}_2 = \mathbf{W}_2\mathbf{W}_1 \cdot \vec{x}_1\,. \tag{3.2}$$

Since the laws of linear algebra permit to redefine $\mathbf{W}_2\mathbf{W}_1$ to a new matrix $\mathbf{W}_3$, (3.2) looks the same as (3.1) after a simple matrix-vector computation. So, one could redefine a two-neuron calculation to a one-neuron calculation. This leads directly to the need for non-linearity, introduced as the first important definition for neural networks: activation functions. These functions provide the necessary non-linearity that build a deep neural network and allow learning of non-linear problems. An activation function $f(x)$ comes with a bias vector $\vec{b}$, which adds a constant offset to the output of the matrix-vector computation

$$\vec{y}_1 = f\left(\mathbf{W}_1 \cdot \vec{x}_1 + \vec{b}\right)\,. \tag{3.3}$$

This bias vector is trainable and useful if the training data features patterns or a constant offset.

One of the most commonly used activation functions is the Rectified Linear Unit (ReLU) [64], which is defined by

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else } . \end{cases} \tag{3.4}$$

For negative input values, the resulting values and the corresponding derivatives are zero, which slows training and prevents the weights from being efficiently updated. This and an improvement in accuracy led to the invention of the Parameter Rectified Linear Unit (PReLU) activation function [50], a particular version of ReLU,

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \tag{3.5}$$

with a learnable parameter a for negative input values. This activation function is utilized for the architecture used in this thesis, as specified in Section 5.1. Figure 3.3 illustrates the function graphically in comparison to a ReLU function.

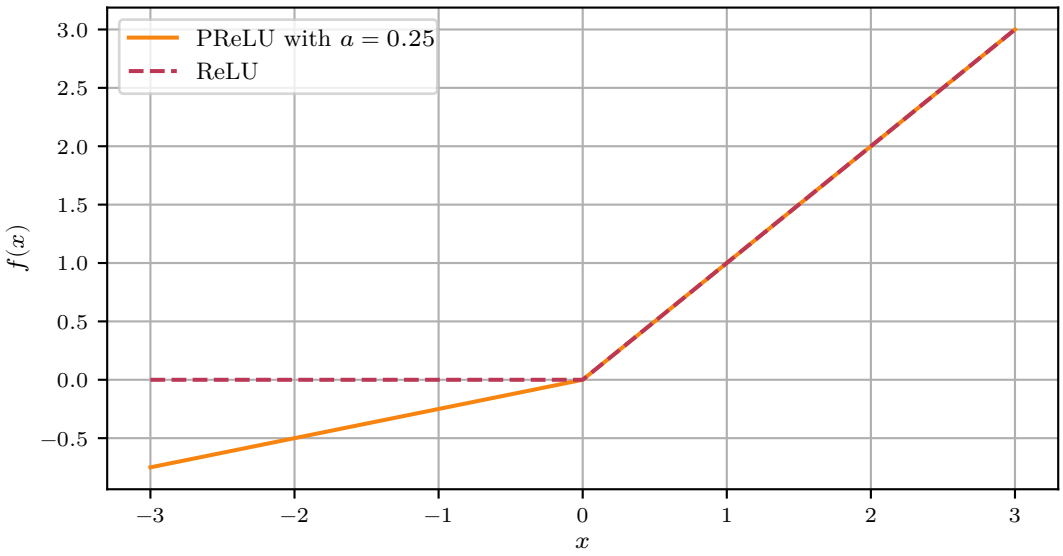

Figure 3.3: Visualization of a PReLU activation function with $a = 0.25$ and a ReLU activation function.

Returning to the weights of each neuron introduced in (3.1), the learning process of neural networks essentially means adjusting these weights repeatedly during training. This optimization problem is solved by minimizing the so-called loss function, which is a central part of each neural network training and the mathematical expression of the present problem. In order to be applicable to the task at hand, the function needs a global minimum at the point of optimal agreement between the predicted and the desired output. The loss functions used in this thesis are explained in Section 5.2.

After feeding the training data set to the network, the output of the model and the labels are compared and penalized according to the chosen loss function. The higher the resulting value, the worse the training performance. To minimize the loss function and to improve the results for the next training iteration, backpropagation [90] is applied. This is done by calculating the derivatives of the loss function starting from the last and going to the first layer while exploiting the chain rule. Next, the weights are updated accordingly, using mostly methods from the field of stochastic gradient descent are used [16]. Over the years, the methods of optimization have evolved. In this thesis, the Adaptive Moment Estimation (ADAM) optimizer [62] is utilized, which has an additional momentum parameter for improved stability.

Another important hyperparameter in conjunction with weight optimization is the learning rate λ. It controls the influence of the computed gradients $\mathbf{g}$ on the update of the new weights $\bar{\mathbf{W}}$

$$\bar{\mathbf{W}} = \mathbf{W} + \lambda \mathbf{g} \tag{3.6}$$

with $\mathbf{W}$ being the previous weights. Choosing the learning rate too high may result in leaving the global minimum and worsen the result, while choosing it too low slows or prevents convergence at all in a reasonable time. Specifics for the learning rate are elaborated in Section 5.2.

Since computing gradients can be computationally expensive, it is common not to feed the whole training data set all at once to the network but to split the data in so-called minibatches. Thus, the computation on these subsets of the data is faster and cheaper. Consequently, the parameters are updated after each minibatch. The size of each minibatch is a tunable hyperparameter called batch size. When all minibatches have been fed to the neural network, one so-called epoch of the

training process has been completed. [40]

3.3 Convolutional Neural Networks

For image data, which is the focal point of this thesis, it is expedient to utilize its grid-like topology. This can be leveraged using convolutions and, therefore, CNNs. A convolution is mathematically defined as

$$s(t) = \int x(a)w(t-a)\,\mathrm{d}a \tag{3.7}$$

with x as the input, w as the kernel and s as the feature map. This terminology is often used in the context of CNNs. (3.7) changes to a summation over two individual axes when using the image I and the kernel K

$$S(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)\,. \tag{3.8}$$

(3.8) still differs from the actual implementation of most neural network libraries, including PyTorch [85]. Instead, these algorithms implement the cross-correlation

$$S(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)\,, \tag{3.9}$$

because there is no need to preserve the commutative property. The kernel K is often chosen to be smaller than the image size in order to group features in small groups of pixels, also called sparse connectivity. This reduces the amount of parameters to be saved and reduces computation time. Also, the same set of weights in the kernel is used for the whole image, further reducing the amount of parameters to be stored [40]. The number of pixels traversed by the kernel for one step is referred to as stride. A stride higher than one means that one set of pixels is skipped for the convolution. In some cases, depending on the kernel and the image size, it is necessary to add pixels to the edges of the image to achieve the desired image size after the convolution has been applied. This is called padding and can be performed in multiple ways, for example, by mirroring the edge pixels or adding zeros. In this thesis, the pixel values on the edges are reflected if padding is utilized. A visualization of the calculations during a convolution is shown in Figure 3.4. In Figure 3.5, a more task-related visualization of the reconstruction procedure performed by a convolutional layer is shown.

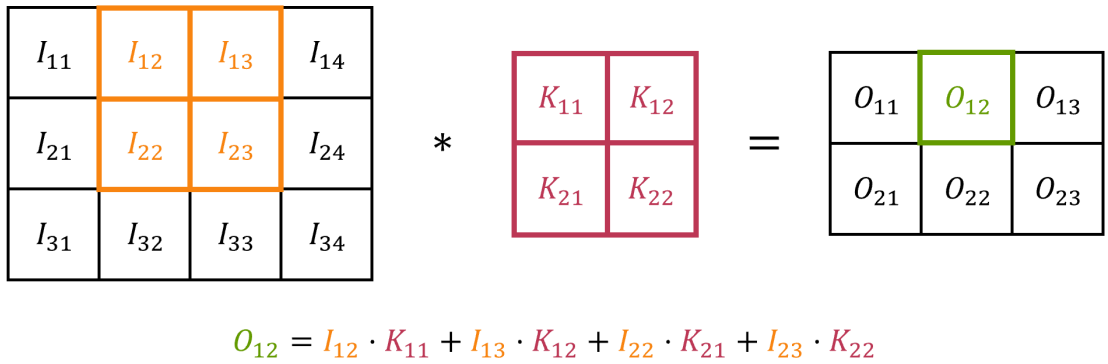

Figure 3.4: Schematic illustration of the functionality of a convolution. With an image size of 3 px × 4 px and a kernel size of 2 px × 2 px without padding and a stride of 1, the resulting output size is 2 px × 3 px. The weights K_{11}, K_{12}, K_{21}, and K_{22} are the same for the whole image, which is called parameter sharing.

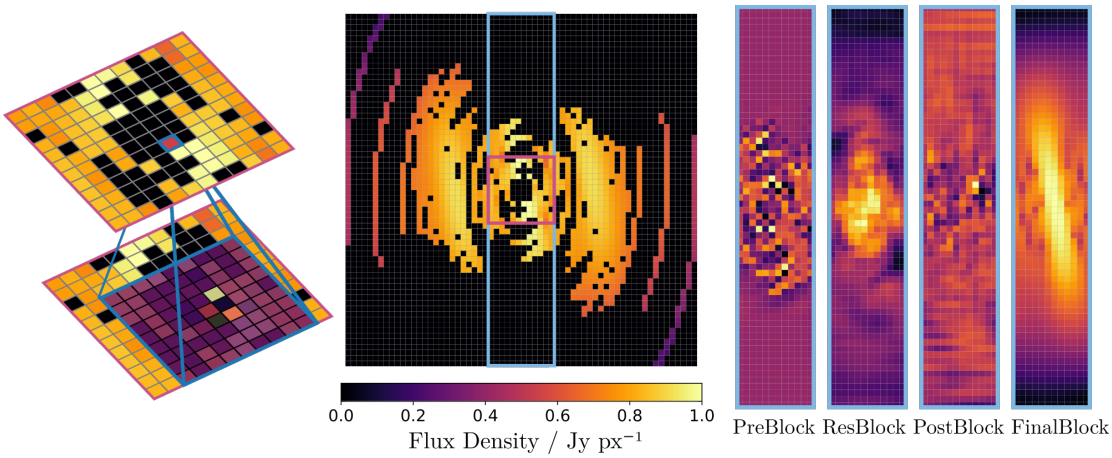

Figure 3.5: Visualization of the operation of a convolution layer in the context of Fourier data. Exemplary (u, v) data with missing information (black pixels) is shown in the middle. The purple highlighted area is shown on the left, visualizing the number of pixels which contribute to the calculation of one pixel value with missing information. The light blue area is visualized again on the right side after several key points in the architecture. The PreBlock is the first block, the ResBlock forms the core of the model, and the PostBlock and the FinalBlock are located at the end of the architecture. It can be seen that the missing information, especially at the edges of the image, is reconstructed, and structures form. Taken and adapted from [97].

4 Simulations

An often overlooked factor for the successful training of a deep learning model is the selection of the appropriate data set. Suppose essential features of the problem at hand, in terms of image reconstruction, for example, brightness or position and orientation of objects, are absent in the training data set. In that case, the model will hardly be able to recognize and reconstruct those features. This chapter deals with the creation and step-by-step adaptation of the simulated data sets so that the deep learning model is able to, first, reliably reconstruct the simulated data and, second, generalize enough to be able to reconstruct interferometric measurements. In Section 4.1, the simulations based on artificial radio galaxies [65, 91] are explained, while Section 4.2 presents the simulations based on the DSHARP [4] measured by ALMA [118].

4.1 GAN Simulations

In previous publications, the simulated data consisted of simple sources built from Gaussian components [100]. For this thesis, a more sophisticated and realistic approach was chosen: Generative Adversarial Network (GAN)-generated radio galaxies [65, 91]. Such a generative adversarial model consists of two parts: a generator and a discriminator, as displayed in Figure 4.1. The generator's task is to create structured data from statistically random noise that follows a predefined distribution given by the data. The discriminator counteracts this by receiving both the generator's output and the target data from which the generator has learned the distribution and attempting to distinguish between them. At the beginning of the training, the generator has not yet fully learned the underlying distribution of the target data. Thus, the adversarial discriminator can easily separate the generator output from the target data. As the training progresses and the generator learns, its output images are more complex to classify as simulated. In an ideal setup, the underlying distribution gets fully learned, and the discriminator cannot reliably distinguish between generated and true data. Hence, the generator is then able to create an arbitrary amount of data following the distribution learned in the training process just from random noise. [41, 94]

In this particular framework, an advanced version of a GAN was used, the wGAN [5]. For this version, the loss function contains the Wasserstein distance, and the discriminator is called critic because there is no longer a classification in the context of statistics, and thus, there is no discrimination between generated data and target data. This technique improves the stability of the learning process and solves numerous problems of the traditional GAN approach. Furthermore, both parts of the wGAN are trained on an additional feature, a morphological class label. There are four possible labels: FRI, FRII, the "bent", and the "compact" class. For this thesis, only the FRI and FRII sources [36] are used. The labels themselves originate from the fact that the wGAN was trained on 5″ images from the Faint Images of the Radio Sky at Twenty-cm (FIRST) survey [7] conducted by the VLA [109]. Summarizing, the trained generator from [65, 91] was used to generate FRI and FRII images looking as measured during the FIRST survey. Some examples of GAN-simulated source distributions are shown in Figure 4.2.

For the purposes of my recent publication [39] and this thesis, 30 000 FRI and 30 000 FRII sources were simulated and split in a 5:1 ratio, resulting in 50 000 training and 10 000 validation

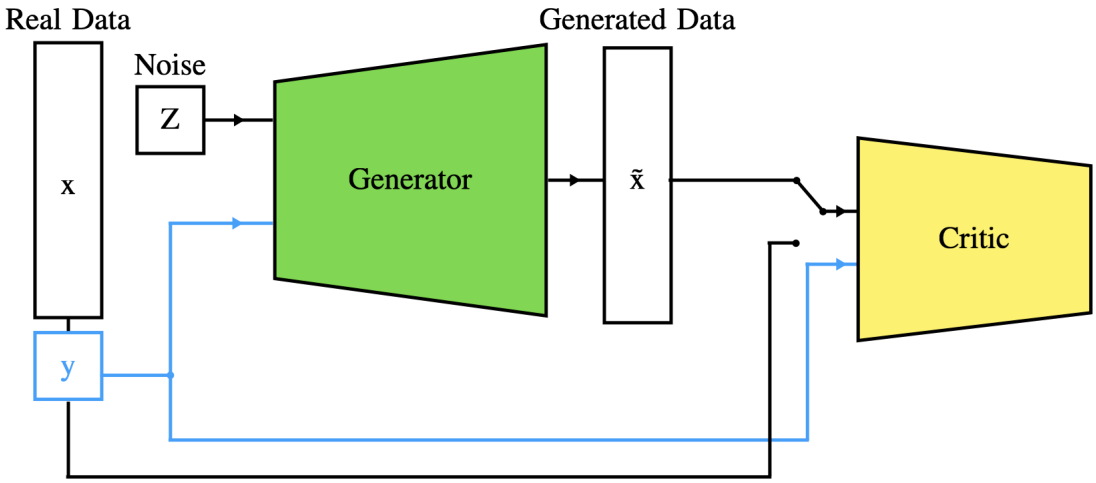

Figure 4.1: Schematic representation of the Wasserstein generative adversarial network (wGAN) architecture used in [91]. y corresponds to the class labels of the real x or generated data $\tilde{x}$. The generator (green) generates data $\tilde{x}$ utilizing the noise Z and the class labels y. After that, $\tilde{x}$ is compared to y by the critic (yellow), which distinguishes between real and generated data.

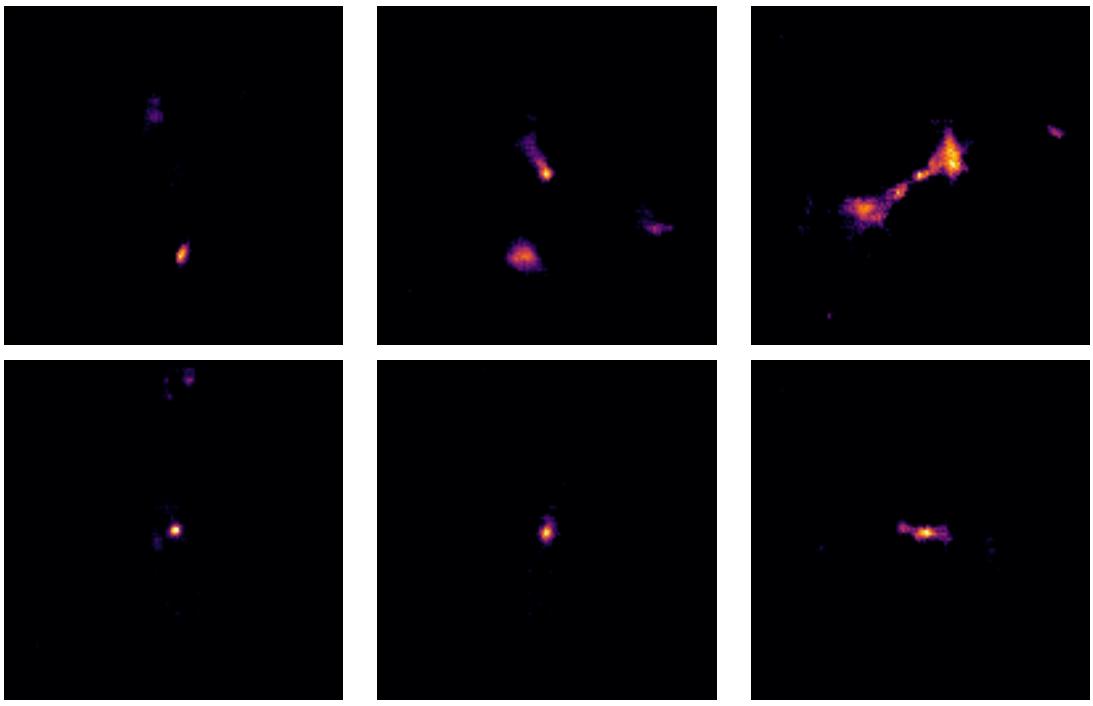

Figure 4.2: Example images generated by the GAN. Both the core-dominated FRI class and the jet-dominated FRII class are visible in this selection, for example, on the bottom left and the top right, respectively. Furthermore, the sources vary in terms of location and number.

images. Additionally, 10 000 testing images, consisting equally of FRI and FRII sources, were created.

For the training of the wGAN, some alterations have been made to the training data set. First, the noise of the measured FIRST data was reduced by setting all pixels below three times the local Root Mean Square (RMS) noise to that threshold. Furthermore, the pixel values were scaled to the range of -1 to 1. Due to the nature of GANs, the generated images inherit these properties [91]. For the purposes in this paper [39] and this thesis, the size of the output images was set to 128 px × 128 px. Additionally, the following postprocessing steps were taken:

- Gaussian smoothing was necessary to discard steep falloffs in intensity between neighboring pixels. These falloffs were relics of the wGAN training process and not observed in comparable, by traditional methods cleaned images. Thus, a Gaussian smoothing was performed. The width of the Gaussian kernel was an important hyperparameter to tune since it was necessary to prevent the creation of false structures and the vanishing of substructures. The chosen value was $\sigma = 0.75\,\text{px}$ because it best fulfilled the abovementioned criteria. A comparison of the effects of different widths on the same source is shown in Figure 4.3.

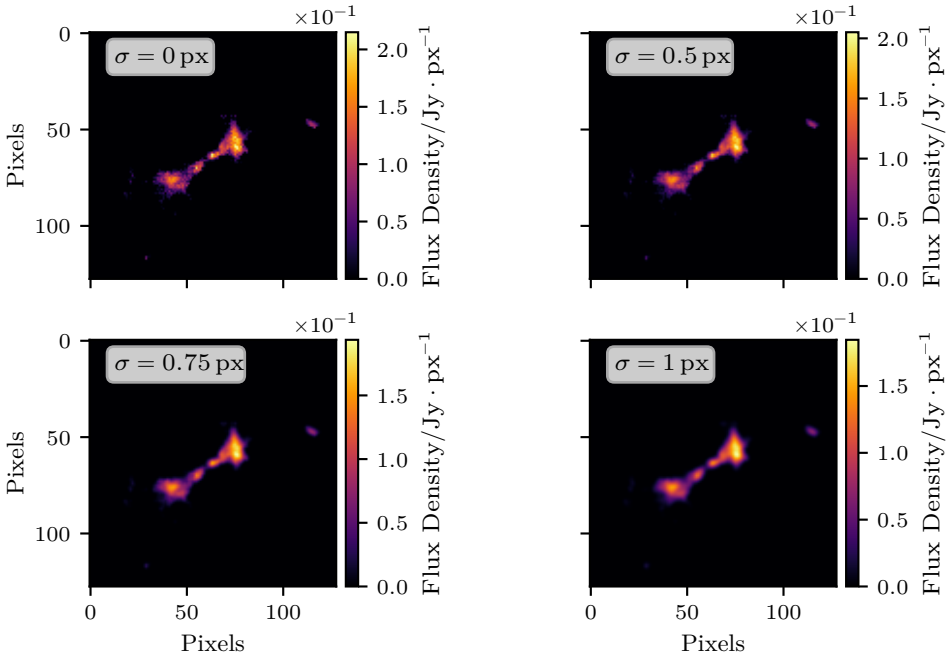

Figure 4.3: Overview of the effect of Gaussian smoothing on the same source from the GAN-generated data set. A Gaussian kernel width of 0 px (top left), 0.5 px (top right), 0.75 px (bottom left), and 1 px (bottom right) are applied. The kernel width of 0.75 px was used going forward since it best combined the two premises of preventing both the creation of fake structures and the vanishing of small structures.

- The intensity of the sources was scaled between 1 mJy and 300 mJy to ensure a signal-to-noise ratio (SNR) ranging between 1 and 100. The aim is to make the network more robust regarding data quality and signal visibility. With the help of the WSCLEAN algorithm [84], the SNRs were computed by dividing the maximum of the cleaned image by the

standard deviation of the residual

$$\mathrm{SNR} = \frac{\max\left(I_{\mathrm{clean}}\right)}{\mathrm{std}\left(I_{\mathrm{residual}}\right)} \,. \tag{4.1}$$

The residual image is the remaining flux after the cleaned flux has been subtracted from the dirty image. In Table 4.1, the chosen parameters for the cleaning are listed. The resulting distribution of SNRs in the test data set is visualized in Figure 4.4. Additionally, the effect of different SNRs on the reconstruction quality of WSCLEAN using the wGAN-generated data set can be seen in Figure 4.5. Further explanation and the application to the evaluation methods can be found in Section 6.2.

Table 4.1: Overview of the parameter settings used to create the clean images and to compute the SNRs using WSCLEAN

Parameter	Setting
size	128 px
scale	1.56 asec
mgain	0.8
gain	0.01
niter	1 000 000
auto-mask	5
autothresh	1
weight	uniform

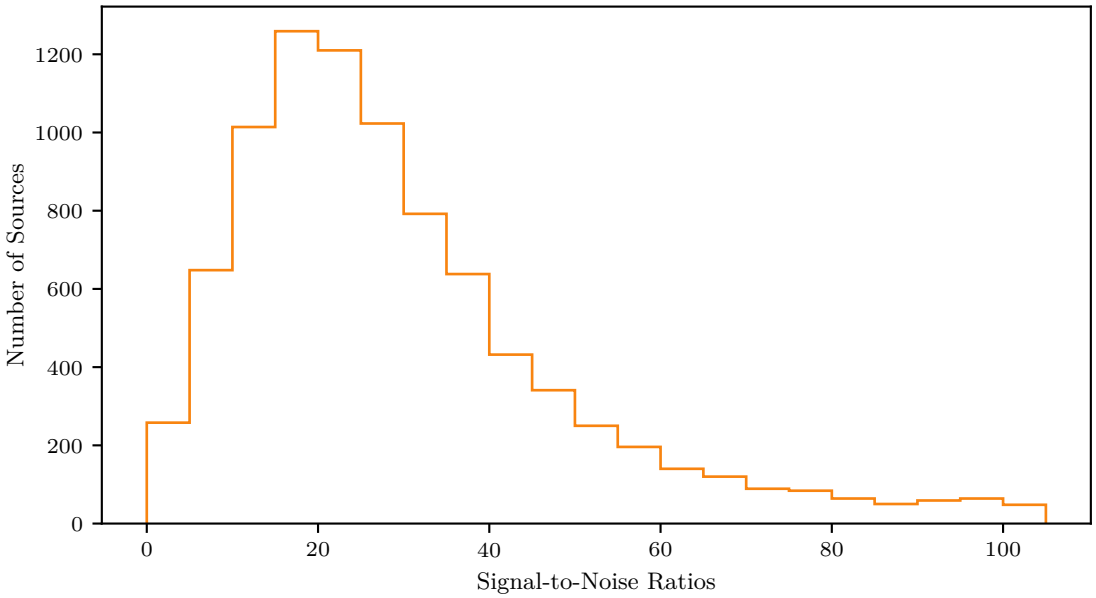

Figure 4.4: Histogram of SNRs in the test data set. The majority of the sources has a SNR around 20.

Looking back at Figure 1.1, the radio galaxy simulations are finished, and so the simulation of a radio interferometer measurement using RIME is next. As explained in Section 2.2, the

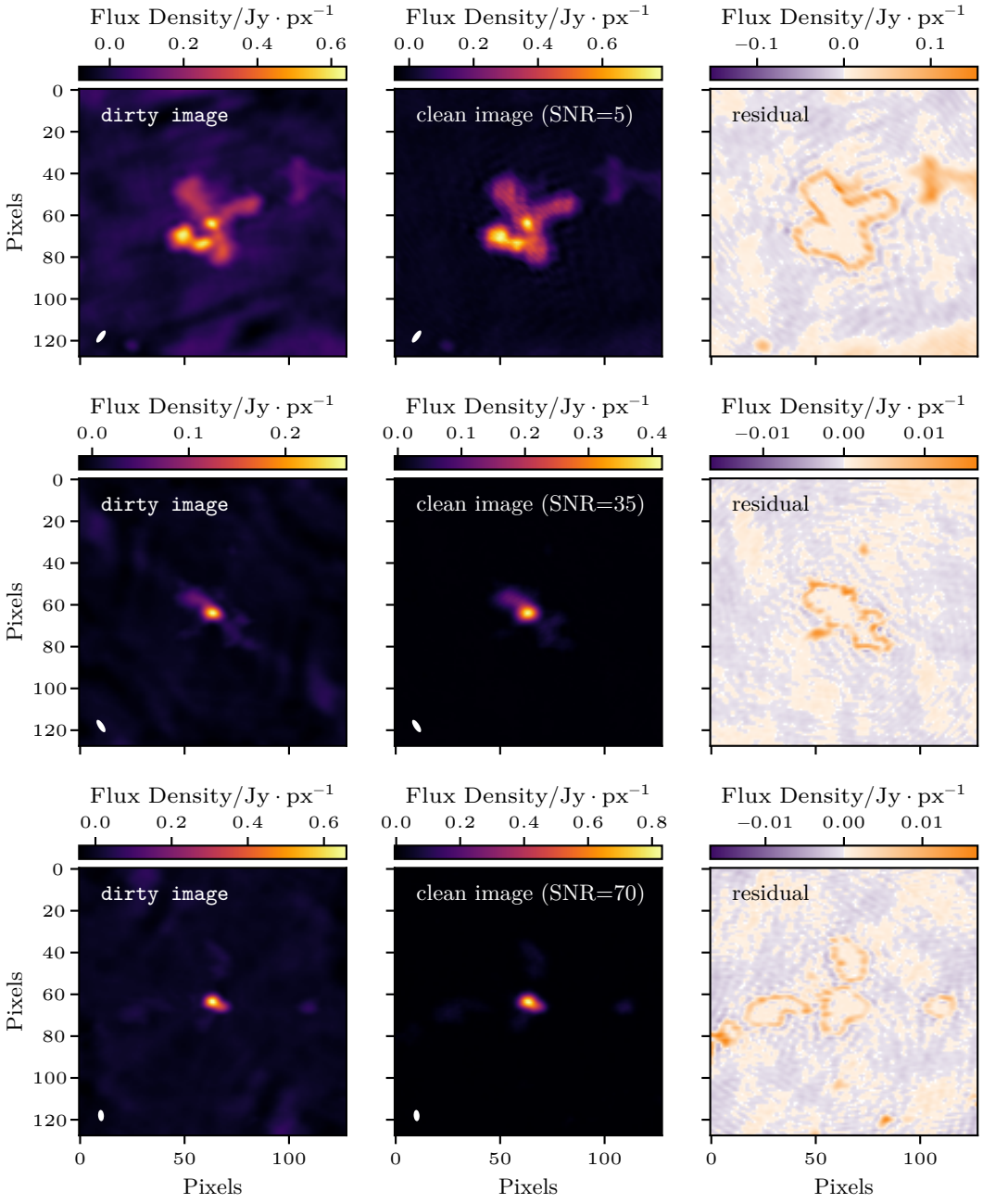

Figure 4.5: Effect of different SNRs on the cleaning performance of WSCLEAN. From left to right: Dirty image, cleaned image and residual image. From top to bottom: Sources with SNR of 5, 35 and 70. It can be seen that the reconstruction quality increases with increasing SNR.

antenna layout is one of the choosable parameters of the `pyvisgen` package [98]. In line with [7], the VLAs B-configuration was used. The corresponding antenna positions are listed in Table 4.2. An exemplary (u, v) coverage using this layout and RIME is shown in Figure 4.6. It can be seen that while the center is well covered, the outer parts are not due to the relatively short baselines of the VLA compared to experiments performing VLBI. For the model training in Section 5.2, this attribute has to be kept in mind since it will be difficult to reconstruct values at the edges of the images.

Table 4.2: Overview of the antenna positions of the VLA within the Earth-centered coordinate system, used as a parameter for the RIME. The given positions correspond to the B-configuration of the VLA used for the FIRST observations.

Station Name	X / m	Y / m	Z / m
W32	1640.03	−4329.93	−2416.71
N24	−1660.49	−259.40	2454.42
W20	733.35	−1932.98	−1078.11
E24	765.21	2889.45	−1108.88
N32	−2629.09	−410.65	3885.60
E36	1534.56	5793.91	−2223.48
N16	−801.40	−124.97	1182.12
W8	152.76	−401.27	−223.40
W28	1316.45	−3443.31	−1913.53
W12	306.17	−804.58	−448.08
W36	2000.07	−5299.80	−2962.89
N28	−2091.44	−326.59	3089.41
E32	1253.25	4733.62	−1816.91
N36	−3217.58	−502.67	4756.12
E28	998.67	3764.31	−1443.47
N4	−74.82	−11.76	111.63
E4	35.60	133.65	−51.10
W4	46.92	−122.02	−67.61
W16	499.84	−1318.00	−735.21
N8	−243.60	−38.05	360.04
E16	376.99	1440.99	−556.13
E8	114.43	438.69	−169.49
E20	560.10	2113.27	−810.70
E12	229.47	879.60	−339.87
N20	−1174.34	−183.30	1734.24
N12	−489.31	−76.31	721.52
W24	1005.43	−2642.99	−1472.20

Because real measurements depend on a large number of variable parameters, these same parameters can also be freely selected in `pyvisgen`. Additionally, it is also possible to provide a range to certain values, which then serve as boundaries for a random number generator, providing the necessary scope in, e.g., pointing position that a real radio interferometer sky survey requires. Furthermore, it enables variations of the simulated measurements, which is beneficial for the generalization capacity of the neural network. The corresponding parameters chosen to simulate the observation of the wGAN-generated radio galaxies are summarized in Table 4.3. The values have been chosen in accordance with the specifications from [7].

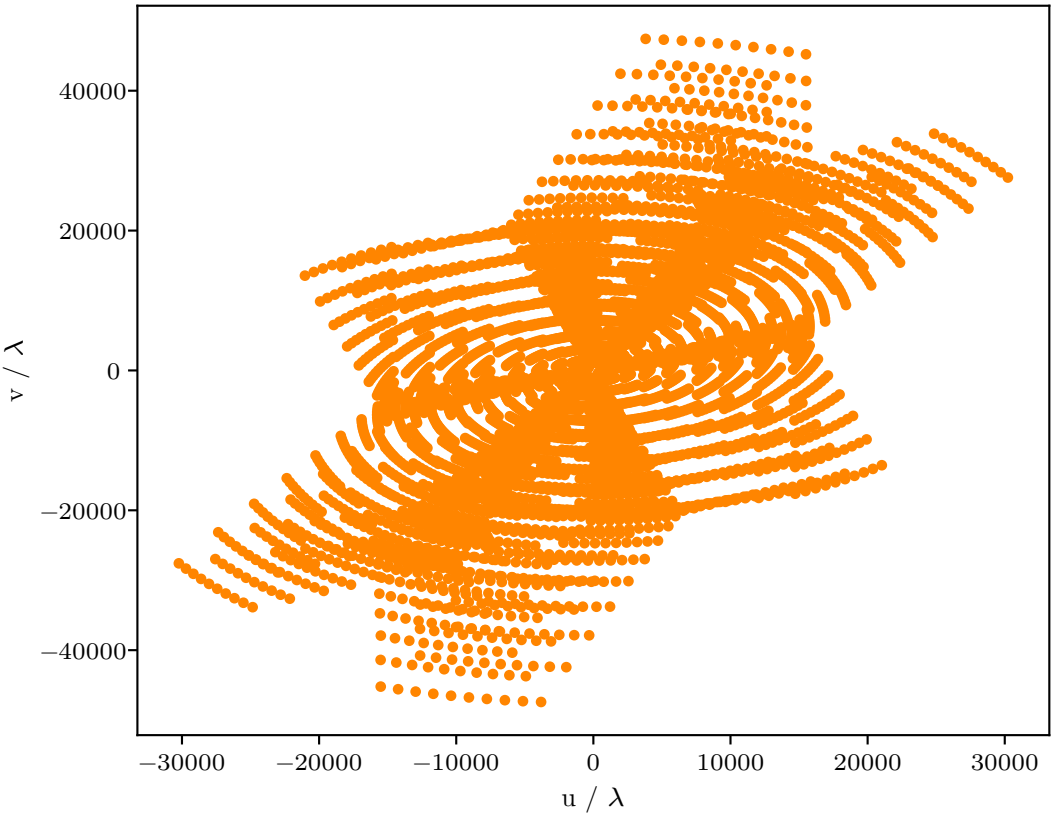

Figure 4.6: Exemplary (u, v) coverage for a simulated FIRST measurement using RIME. For the sake of visibility, just one out of four possible spectral windows is shown. While the (u, v) coverage in the central parts is very dense, the outer parts are only covered sparsely.

Table 4.3: Parameters chosen for simulating FIRST measurements using RIME in `pyvisgen`. In cases where multiple values are specified, these values serve as bounds of random values drawn within these bounds.

Sampling option	Value
Image size	128 px
FOV center ra	[100°, 110°]
FOV center dec	[30°, 40°]
FOV size	350
Corr. int. time	30 s
Number of scans	2
Scan duration	[30 s, 40 s]
Scan separation	360 s
Reference frequency	1.365 GHz
Bandwidths	3 MHz
# Bandwidths	4
Frequency offset	64 MHz

Furthermore, system noise also has to be considered. This corruption effect depends on the observation bandwidth $\Delta\nu$ and the correlator accumulation time τ_{acc} and is Gaussian distributed with a standard deviation that follows [104]

$$\Delta S = \frac{1}{\eta_s} \frac{\mathrm{SEFD}}{\sqrt{2\,\Delta\nu\,\tau_{\mathrm{acc}}}}\,. \tag{4.2}$$

Here, η_s corresponds to the system efficiency factor, which is different for each experiment and takes losses from electronics into account [26]. The most important part is the System Equivalent Flux Density (SEFD), which can be considered as an overall measure of system performance [104]

$$\mathrm{SEFD} = \frac{T_{\mathrm{sys}}}{K} \tag{4.3}$$

with the system temperature T_{sys} and K as a composition of the antenna efficiency and the area of the antenna [26]. The specific values for the variables appearing in (4.2) are taken from Table 4.3 or from [81], with $\eta_s = 0.93$ and $\mathrm{SEFD} = 420\,\mathrm{Jy}$. The resulting noise is in the order of 10 mJy and is added separately to the real and the imaginary part of the simulation. By varying the peak flux, `pyvisgen` is able to simulate measurements with different SNRs utilizing (4.2). This will be used in detail in Section 6.2, where an evaluation of the neural network results based on the source's SNR is performed.

For the last step in the simulation chain, the resulting visibilities coming out of the RIME are gridded since both traditional radio interferometry imaging software like WSCLEAN and the neural network approach require two-dimensional images. This is also performed by `pyvisgen`, with the grid being defined by the number of pixels and the Field of View (FoV). In contrast to the convolutional gridding in WSCLEAN, the gridder in `pyvisgen` utilizes a two-dimensional histogram. The data set containing Figure 4.6 was gridded with a image size of 128 px × 128 px and a FoV of 200″, which corresponds to a pixel size of 1.56″. The complex visibilities are then divided into the real and the imaginary parts, which are saved separately. In Figure 4.7, an exemplary gridded real and imaginary part is shown.

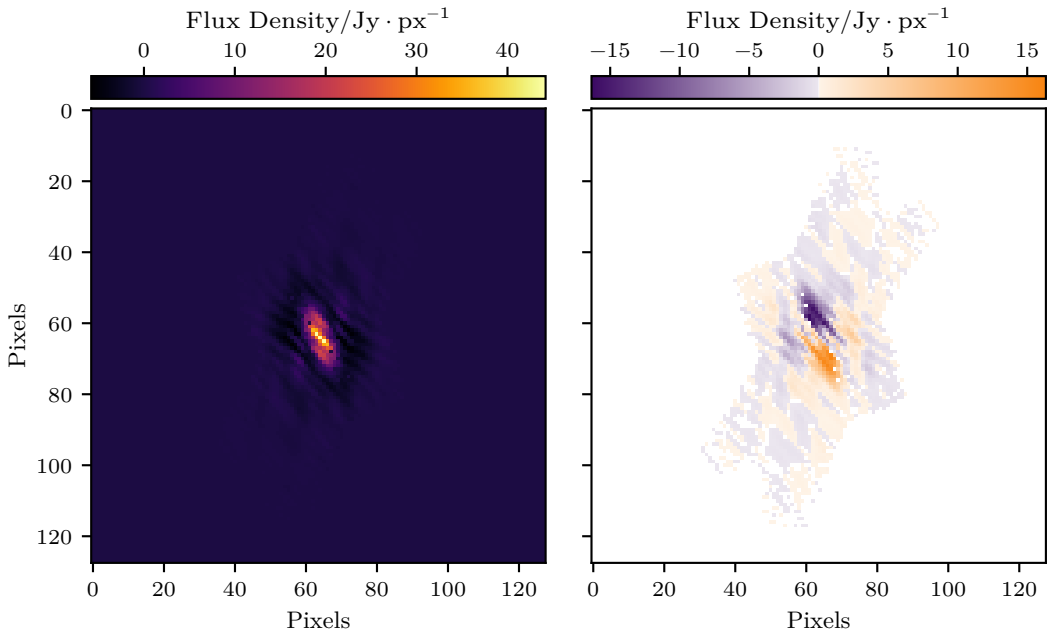

Figure 4.7: Gridded real (left) and imaginary (right) part of the visibilities shown in Figure 4.6. It is visible that information is missing, especially in the case of the imaginary part.

4.2 DSHARP Simulations

With regard to the planned future reconstruction of measured interferometric data, see Chapter 8, another type of astronomical object was simulated: protoplanetary disks. These simulations are based on the work of Andreas Maisinger [70]. The goal is to simulate a data set of protoplanetary disks and use it to train a neural network model to be able to reconstruct measurements taken in the context of DSHARP.

The DSHARP project was conducted by ALMA. Its goal was to perform a survey of 20 nearby protoplanetary disks in order to examine the prevalence, forms, locations, sizes, and amplitudes of small-scale substructures in the disk material and their relation to planet formation processes. This was done by investigating the 240 GHz continuum and the ^{12}CO J=2-1 line emission of these protoplanetary disks. This project is particularly suitable for testing the reconstruction approach presented in this thesis because, among other things, the source morphology is relatively simple compared to other objects. A cleaned image of all twenty sources examined is shown in Figure 4.8. [4]

The foundation of the simulations of protoplanetary disks in this thesis is the geometric crescent model [59], which can be used to simulate black holes. The idea behind the model is to create one large and one small disc and subtract the smaller one from the larger one. This way, a crescent-shaped structure remains, which is blurred and adjusted to match the event horizon of a black hole. In Figure 4.9, an example image of the crescent model is shown.

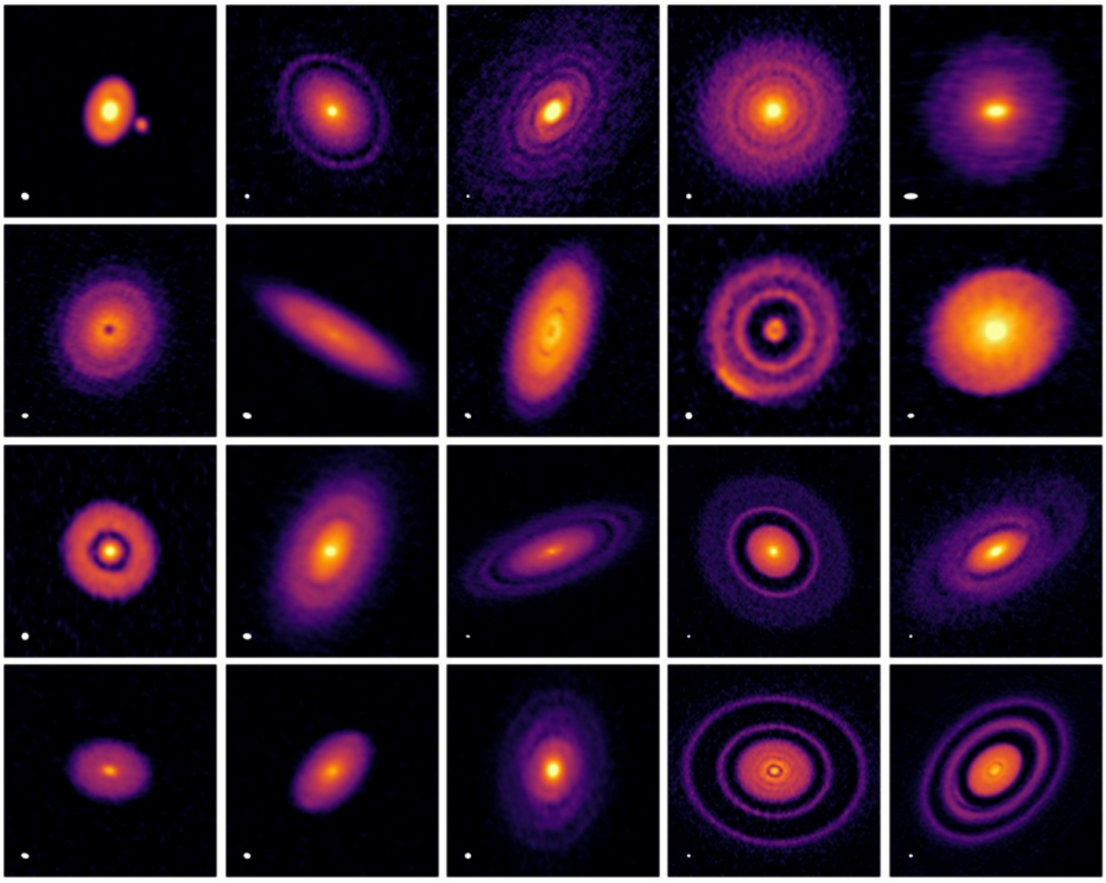

Figure 4.8: Overview of the sources measured for DSHARP. The beam is shown in the bottom left corner of each image. Especially the orientation, the number of rings around the core, and the eccentricity vary between the different sources. [4]

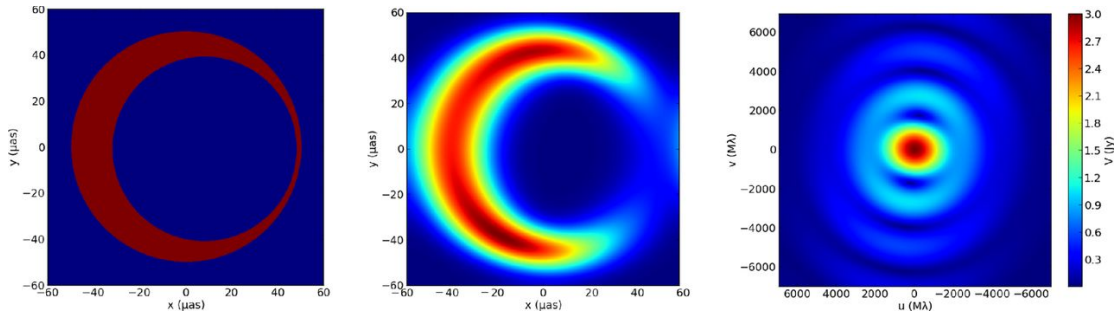

Figure 4.9: Example of a sampled crescent image. A sampled crescent image on the left, the corresponding blurred image in the middle and the blurred visibility amplitde on the right side are shown [59].

Due to mismatches between the simulations and the DSHARP data, the discs were replaced by ellipses simulations, which more accurately represent the source morphologies present in the measurements. In the following the simulation steps are listed:

1. Create an elliptic core
2. Convolve the core with a two-dimensional Gaussian kernel to create blurring effects
3. Create the rings around the core
   a) Generate two ellipses of different sizes
   b) Subtract the smaller ellipse from the larger one
   c) Convolve the resulting crescent-shaped structure with a two-dimensional Gaussian kernel
4. Repeat these steps for the chosen amount of rings
5. Combine the core and the rings

Most of the parameters are chosen randomly (in certain bounds), for example, the number of rings, the distance between the core and the rings, or the ratio between the two semi-axes of the ellipse. In theory, this creates a large enough parameter space for the neural network model to be able to extract the features necessary for learning to reconstruct measurements of protoplanetary disks.

For the next step, the intensities of the crescent-shaped structures had to be adjusted to those of the DSHARP data. For this purpose, the intensity profile of each source was plotted. It turned out that the decrease in intensity was exponential. In order to be able to base simulations on these intensity profiles, it was necessary to fit each profile with

$$y = \mathrm{e}^{mx+b}\,. \tag{4.4}$$

Due to the exponential nature of the data, the logarithm of (4.4) was used, which resulted in

$$\ln y = mx + b\,. \tag{4.5}$$

After obtaining 20 different gradients and y-axis intercepts, they were grouped. It was possible to summarize the values into four different categories with a mean gradient and mean y-axis intercept. One of these four mean gradients and y-axis intercepts is now chosen randomly for every newly simulated source. This ensures that the intensity profile of the simulations matches one of the four subgroups of the DSHARP data set.

In contrast to the work of Andreas Maisinger [70], a few adjustments had to be made to reduce data-simulation mismatches further. First, the image size was increased from $400\,\mathrm{px} \times 400\,\mathrm{px}$ to $512\,\mathrm{px} \times 512\,\mathrm{px}$. This was done because the original image size of the DSHARP data set is $3000\,\mathrm{px} \times 3000\,\mathrm{px}$ and it is necessary to steadily increase the size of the simulated images in order to eventually match the image sizes of measurement data like DSHARP. Due to restrictions in the domain of computational resources, this needs to happen in small steps; thus, $512\,\mathrm{px} \times 512\,\mathrm{px}$ was chosen. Furthermore, the coordinates of the center of the core were adjusted. Previously, they were fixed to the center of the image. This proved to be too restricting when applying a neural network, which was trained on fixed-location simulations, to the measurement data of DSHARP. The protoplanetary disks in the measurement data set are not always centered in the middle of the image but deviate from it. Thus, the location of the core was also randomized in the range of 200 px to 300 px in both dimensions. In Figure 4.10, several simulated protoplanetary disks are shown, which were calculated according to the approach explained above. It is apparent

that the randomized parameters greatly impact the appearance of the simulated protoplanetary disks, as intended. The number of rings, the eccentricity of the core, and the location of the core vary.

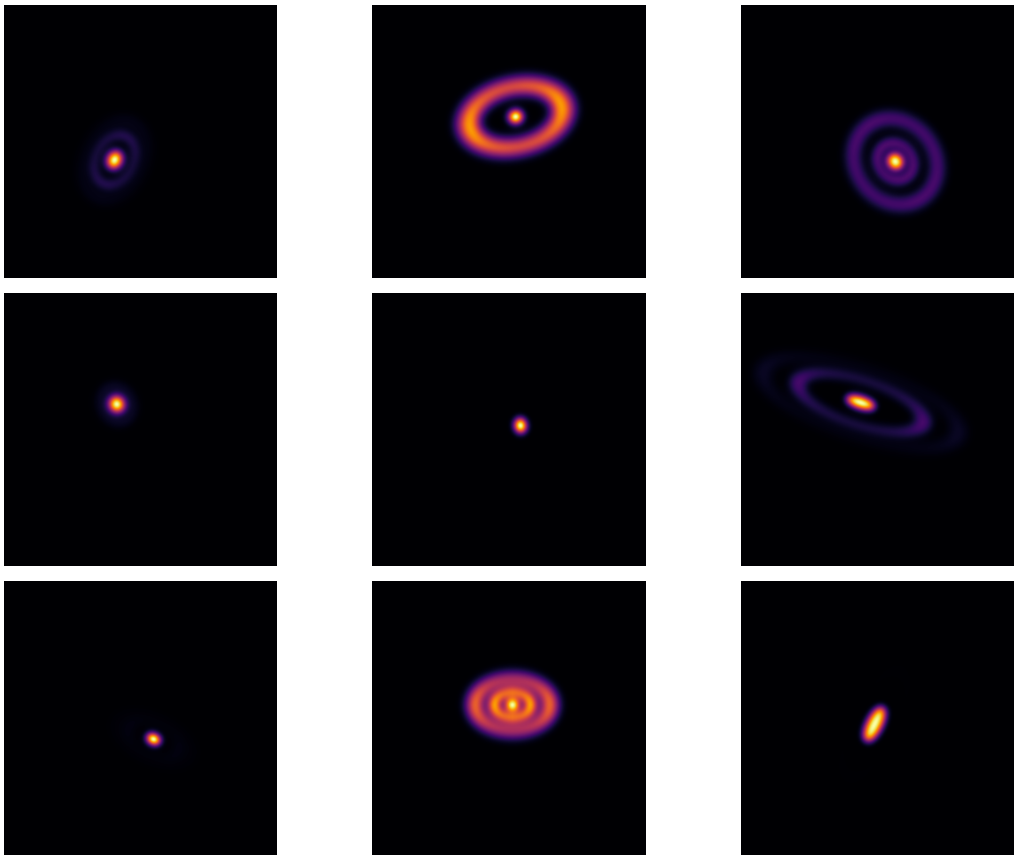

Figure 4.10: Example simulations of protoplanetary disks. There are multiple variations with no rings, weakly developed rings or bright rings that shine brighter than the core. The number of rings is variable too.

Similar to Section 4.1, after simulating the protoplanetary disks, they were fed to the RIME formalism via the `pyvisgen` package. For this, the hyperparameters of the algorithm are adjusted in accordance with [4] while utilizing the antenna locations of ALMA listed in Table A.1. Also, the SEFD, used in the system noise calculation in (4.2), is changed to 97 Jy. This value was measured for ALMA during the DSHARP observations [3]. The simulation parameters for `pyvisgen` are listed in Table 4.4. In Figure 4.11, one exemplary result of the RIME simulation using the DSHARP parameter values is shown. The differences compared to the layout of the VLA are directly visible. Also, more information is missing between the visibilities than in Figure 4.6. The corresponding gridded real and imaginary parts are visualized in Figure 4.12. The gridding was performed with an image size of 512 px × 512 px and a FoV of 9″, which correponds to a pixel size of 0.017 58″.

Table 4.4: Parameters chosen for simulating DSHARP measurements using RIME in `pyvisgen`. In cases where multiple values are specified, these values serve as bounds of random values that are drawn within these bounds.

Sampling option	Value
Image size	512 px
FOV center ra	$[-124°, -123°]$
FOV center dec	$[-23°, -22°]$
FOV size	12
Corr. int. time	30 s
Number of scans	[10, 11]
Scan duration	[30 s, 40 s]
Scan separation	300 s
Reference frequency	233 GHz
Bandwidths	125 MHz
# Bandwidths	16
Frequency offset	125 MHz

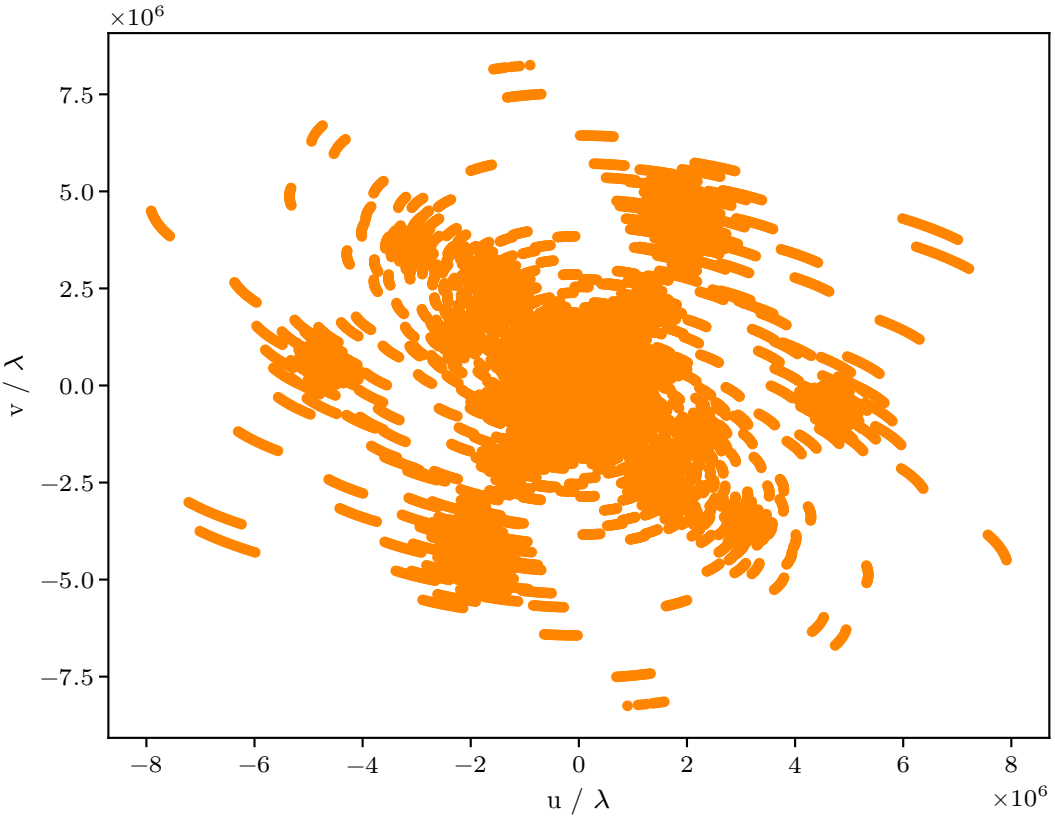

Figure 4.11: Exemplary (u, v) coverage of a simulated measurement of a protoplanetary disk. For the sake of visibility, just one out of the 16 simulated spectral windows is shown. In contrast to Figure 4.6, more information is missing between the visibilities.

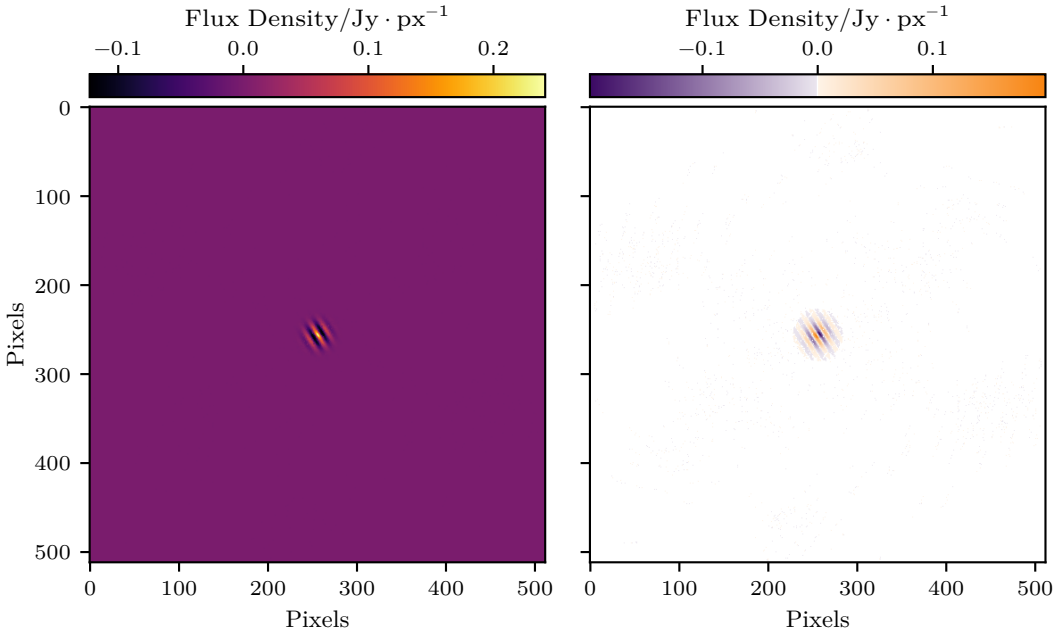

Figure 4.12: Exemplary gridded visibilities of a simulated measurement of a protoplanetary disk.

5 Neural Network Model

After simulating multiple kinds of astronomical objects and simulating measurements conducted by different experiments like ALMA and VLA, the next stage of Figure 1.1, the neural network itself, is reached. The centerpieces of the training process of a neural network are its architecture and the model training. There are a lot of different kinds of networks, but for image data with a grid-topology, CNNs are among the best possible choices. However, one has to specialize even more depending on the task at hand since multiple possible tasks like classification, segmentation, or regression exist. In this chapter, the architecture, as well as the modifications and parameters set for the training process, are explained in Section 5.1 and in Section 5.2, respectively.

5.1 Architecture

In order to understand the choice of building blocks for the architecture, it is necessary to grasp the task that comes with reconstructing unsampled visibility data shown in Figure 4.6. In the fields of, e.g., computer vision and medical imaging, researchers deal with similar problems as presented in this thesis, namely image impainting [9, 33] and super-resolution [37, 23]. Image impainting was initially used for removing noise or physical issues like kinks or blurry parts of an image and has nowadays found its application in image processing procedures like discarding artifacts in images or text removal [92, 61]. On the other hand, Super-resolution describes methods to obtain high-resolution shapes and textures while upsampling the image dimensions [66]. As an often used first approach, interpolation methods like bicubic interpolation have been used in the past to upsample images. However, this frequently leads to noisy upsampled images, as shown in the left image of Figure 5.1. By using a SRResNet, it is possible to reconstruct fine-scale structures and generally a less noisy image [66], as seen in the right image of Figure 5.1. The approach mainly depends on estimating the correct value for the newly created pixels during the upsampling process by applying residual learning.

For the context of this thesis, it is essential to reconstruct missing pixel information, especially at the edges of the image, by utilizing the information in the near vicinity. This is a well-known problem in radio interferometry, as due to the concept, it is impossible to fill all gaps in the (u, v) plane, as outlined in Section 2.1, and therefore, the missing information has to be estimated. Furthermore, the pixel information obtained by the telescopes is corrupted by noise, both extrinsic and intrinsic, which further complicates the task.

In order to fill the missing pixel information, a CNN-based SRResNet following the ideas presented in [100] is constructed. The general network follows [66] with Residual Blocks (ResBlocks) as core components, as published by [49] and adapted by [44]. A schematic overview of the network is visualized in Figure 5.2. The training takes place solely in Fourier space, as indicated in Figure 1.1. Therefore, the incomplete and gridded real and imaginary visibilities are fed to the network as input. In the beginning, the so-called PreBlock is implemented. It consists of a single convolutional layer and a PReLU activation function. The purpose is to increase the number of channels from 2 to 64 and to spread the information in the center to the edges of the image by utilizing a high kernel size of 9, compared to the rest of the architecture.

Figure 5.1: Comparison between bicubic interpolation (left) and a deep residual network (right). It can be seen that the bicubic interpolation results in a blurry, upsampled image, while the neural network approach restores finer structures and details of the image. Adapted from [66].

Next is the central building part of the architecture, the ResBlocks. 16 of these blocks are used for this architecture, and each of them is structured in the same way: A convolutional layer with a smaller kernel size (3) than in the PreBlock, a normalization layer, a PReLU non-linear activation function and another convolutional layer coupled with a normalization. The normalization layer performs either a batch normalization [55] or an instance normalization [111], whereby the latter has proven to perform better on two-channeled Fourier data and is used in every training except for Subsection 5.2.1. In the end, the input to the ResBlock is bypassed via an identity skip connection [51] and added elementwise to the output, which enables residual learning [49]. Residual learning changes the underlying mapping the network attempts to learn. Suppose x is the input to the ResBlock and $H(x)$, the mapping to be learned. In that case, utilizing an identity skip connection and the elementwise summation leads to the following new mapping

$$F(x) := H(x) - x\,. \tag{5.1}$$

Investigations, e.g., [49], have shown that these kinds of layers are easier to optimize in comparison to the traditional standard stacking of layers. Furthermore, it even enables the implementation of deeper networks without significantly increasing the training error and the convergence time, which is the case for the layer-stacking method. Additionally, these layers are more robust to outliers and thus increase the stability of the trained model.

After passing through 16 ResBlocks, the data is eventually fed to the PostBlock and the FinalBlock. The PostBlock consists of a convolutional layer, a normalization layer, and, most importantly, another skip connection and elementwise summation, bypassing and adding the PreBlock output to the PostBlock output. In the end, the FinalBlock concludes the architecture with a single convolutional layer with a kernel size of 9, which reduces the number of channels from 64 to the original 2, the real and the imaginary parts. A detailed overview of the parameters for each layer can be found in Table 5.1. The target images are the complete real and imaginary part of the visibilities, as seen in Figure 5.2 on the right side. More detailed, these images are created by taking the Fourier transformation of the wGAN-simulated sources as shown in

Figure 4.2. Therefore, the (u, v) plane is filled without gaps, and taking the inverse Fourier transformation of the real and the imaginary part results in the perfect recovery of the radio galaxy simulations the analysis chain started with.

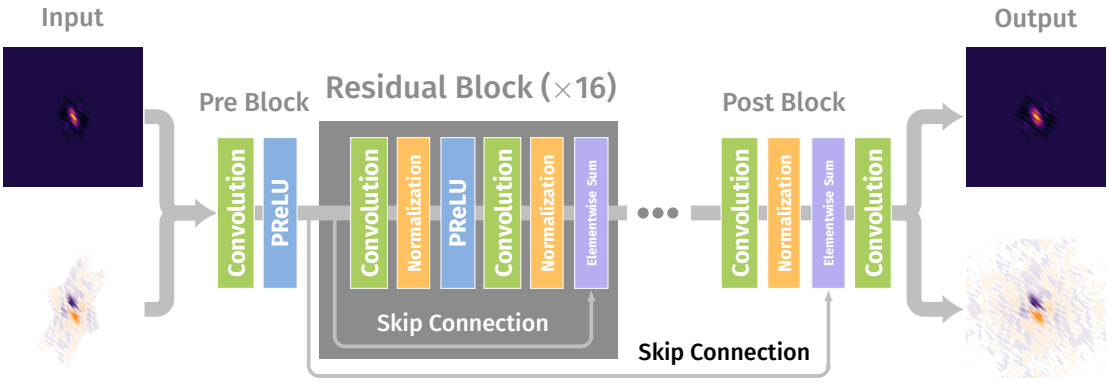

Figure 5.2: Schematic overview of the architecture used, among others, for the training of wGAN-simulated sources from Section 4.1.

Table 5.1: List of parameters for the architecture depicted in Figure 5.2

Module	Layer	C_{in}	C_{out}	Kernel	Stride	Padding
PreBlock	Convolution	2	64	(9×9)	1	4
	PReLU	64	64	-	-	-
ResBlock (×16)	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	PReLU	64	64	-	-	-
	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
PostBlock	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
FinalBlock	Convolution	64	2	(9×9)	1	4

Unless stated otherwise, all of the following neural network models were trained with the architecture depicted in Figure 5.2. Later in this thesis, this architecture is altered to prepare it for the additional prediction of uncertainties, which will be covered in generally in Chapter 7 and specifically in Section 7.1.

5.2 Model Training

In the following paragraphs, the pre-processing steps and essential parameters of the training process are explained. Unless stated otherwise, these steps are done before every training described in this thesis.

Loss Function For the loss function, the Mean Absolute Difference (L1) (5.2) was chosen because it is more efficient for image reconstruction tasks than the Mean Squared Error (MSE) and is able to reconstruct finer details [120]. Specifically for the purpose of this work, tests have shown a clear advantage for L1 over MSE.

$$\mathrm{L1} = |x - y| \tag{5.2}$$

Here, x corresponds to the input image and y to the target image. However, the loss is computed channel-wise and added afterward so that

$$\mathrm{L1}_{\mathrm{total}} = \mathrm{L1}\left(x_{\mathrm{real}}, y_{\mathrm{real}}\right) + \mathrm{L1}\left(x_{\mathrm{imag}}, y_{\mathrm{imag}}\right) \tag{5.3}$$

Image Size Reduction Due to the periodic nature of the Fourier space [10], it is sufficient to train only on one half of the image and to complete the rest by utilizing the symmetry. This saves computational resources and time and increases the efficiency. For the evaluation, the images are restored to their original size by rotating the prediction and concatenating it to the original output of the network. Due to the fact that the FFT of `numpy` [47] creates a central pixel also for even image sizes, the symmetry is slightly offset for the simulated visibilities in Fourier space. To account for that, the image is still rotated by 180°, but the row with the central pixel is not changed. This means that one column is missing, which is approximated with the corresponding column on the other side of the image. This approximation holds for the images in this thesis since the pixel values at the edges are close to zero. In Figure 5.3, the process is visualized.

Normalization Methods In order to enable a smoother training process and prevent a bias towards images with higher values, which get a higher weighting when using (5.2), normalization methods were implemented. For the wGAN-generated sources, the mean μ and the standard deviation σ of the whole training data set were computed and divided into real and imaginary parts. Before each batch, all non-zero pixels of each image in the batch were shifted using

$$f(x) = \frac{x - \mu}{\sigma}, \text{ for } x \neq 0 \tag{5.4}$$

with x being an image in the batch. The transformation was applied to both input and target images. This results in $\mu = 0$ and $\sigma = 1$ for the whole training data set. Sometimes, this is also called Z-Score and a common normalization technique for machine learning algorithms [21].

Later in the thesis, another normalization method is used. It is based on (5.4), but has the goal of normalizing each image, so that $\mu_{\mathrm{image}} = 0$ and $\sigma_{\mathrm{image}} = 1$, utilizing

$$f(x) = \frac{x - \mu_{\mathrm{image}}}{\sigma_{\mathrm{image}}}\,. \tag{5.5}$$

For this purpose, the mean and standard deviation of each input image in the batch is determined. Afterward, equation (5.5) is used on both input and target images. Note that this is performed on all pixels, not just the non-zero ones, as for equation (5.4). Before any evaluation methods can be applied, the output of the neural network needs to be rescaled in order to restore the original values. This is done by inverting the mathematical operations, which results in

$$x_{\mathrm{original}} = x_{\mathrm{normalized}} \cdot \sigma_{\mathrm{image}} + \mu_{\mathrm{image}}\,. \tag{5.6}$$

The different effects of these two normalization methods on the training process are evaluated in detail in Chapter 6.

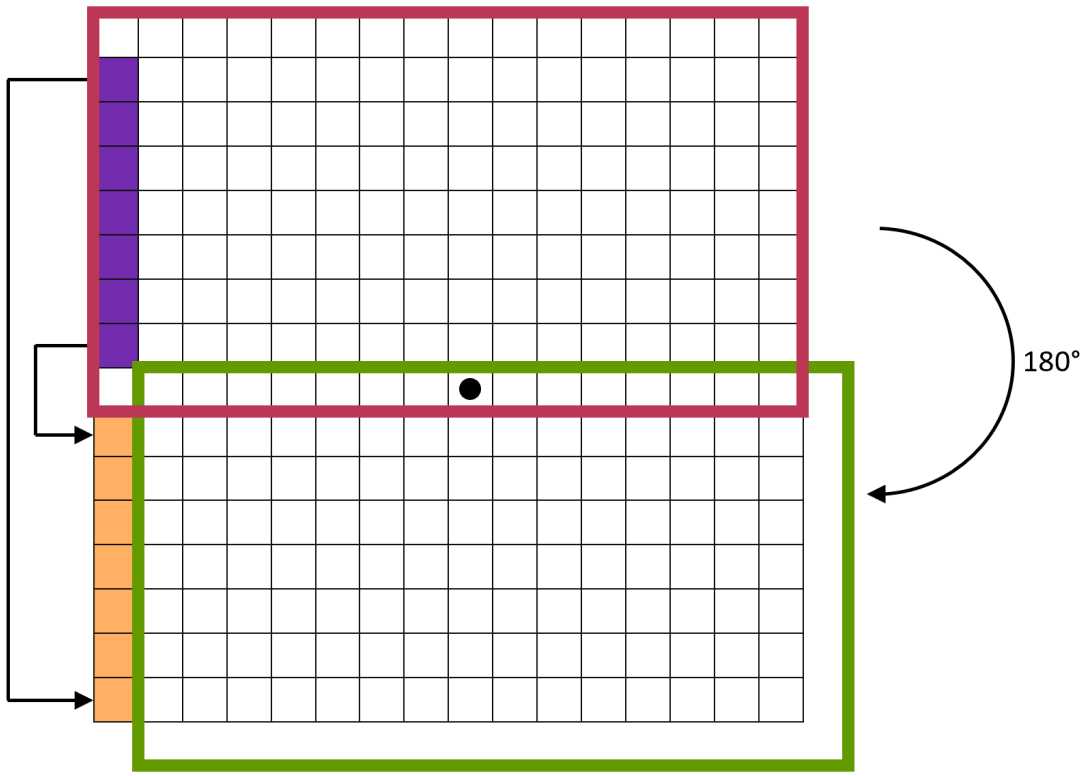

Figure 5.3: Schematic representation of the symmetry operation applied for the evaluation of the neural network. The red part has the size [`image_size`/2 + 1, `image_size`] and is used for the training of the neural network. If the image is now rotated around the central pixel, the row at the bottom is redundant, and one part of the left column is missing, as seen for the green rectangle. The redundant row is discarded, but the information of the column (yellow) needs to be approximated using the corresponding column (purple) of the red rectangle.

Adaptive Learning Rate As early as 1988, it was known that it is helpful for optimization problems to adjust the learning rate over time [57] to achieve faster convergence times. For this thesis, this is taken into account by using a learning rate finder [52] and an adaptive learning rate scheduler provided by `fastai` [53], called "cosine annealing learning rate scheduler". The learning rate starts at 8×10^{-4}, peaks after one-quarter of the epochs at 1×10^{-3}, and then falls until 1×10^{-4} at the end of the training. The detailed progression of the function is shown in Figure 5.4.

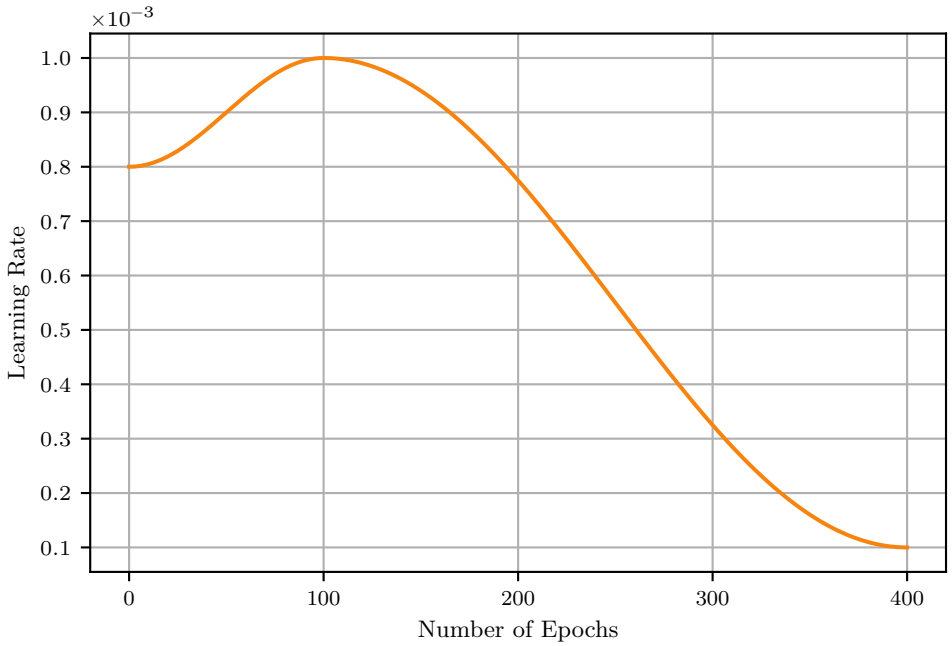

Figure 5.4: Visualization of changes in the adaptive learning rate over the course of the training.

5.2.1 wGAN-simulated Sources

First, a training instance using the wGAN-simulated sources presented in Section 4.1 was conducted with a batch size of 100 and a learning rate adaption as shown in Figure 5.4. The corresponding loss curve of the trained 400 epochs is visualized in Figure 5.5.

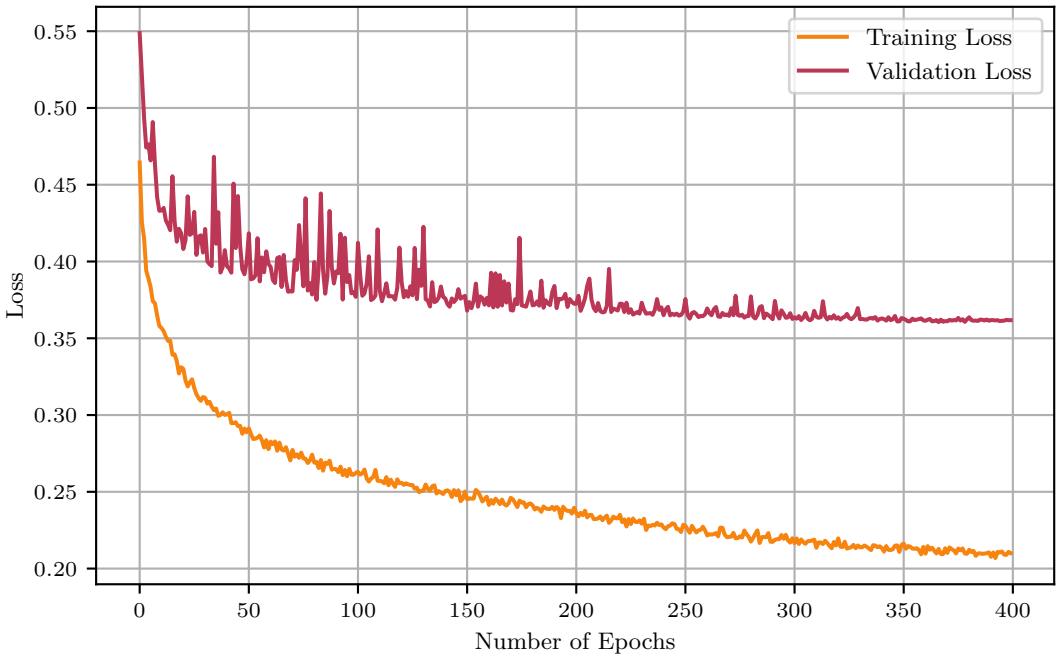

Figure 5.5: Training and validation loss for a model training with wGAN-simulated data.

After approximately ten epochs, a gap between the training and the validation loss forms. This gap increases, but no overtraining occurs since the validation loss declines more slowly than the training loss. For upcoming models, both losses will be closer to each other due to improvements in the normalization techniques.

An exemplary prediction in Fourier space is shown in Figure 5.6. The graphic is separated into real (top) and imaginary parts (bottom). Both parts show a good agreement between prediction and simulation, which is validated by the difference plots. The maximal deviation is an order of magnitude smaller than the maximum flux density, which is valid for both channels and underlines this statement.

The corresponding prediction in image space is shown in Figure 5.7. Again, both prediction and simulated images show good compliance, underlined by the difference visualized in the plot on the right. The main area of missing flux density is located outside of the main structure of the source. Altogether, both exemplary Fourier and image space reconstructions are in good agreement with their corresponding simulation. Advanced evaluation techniques based on the analysis of a multitude of images are presented in Chapter 6.

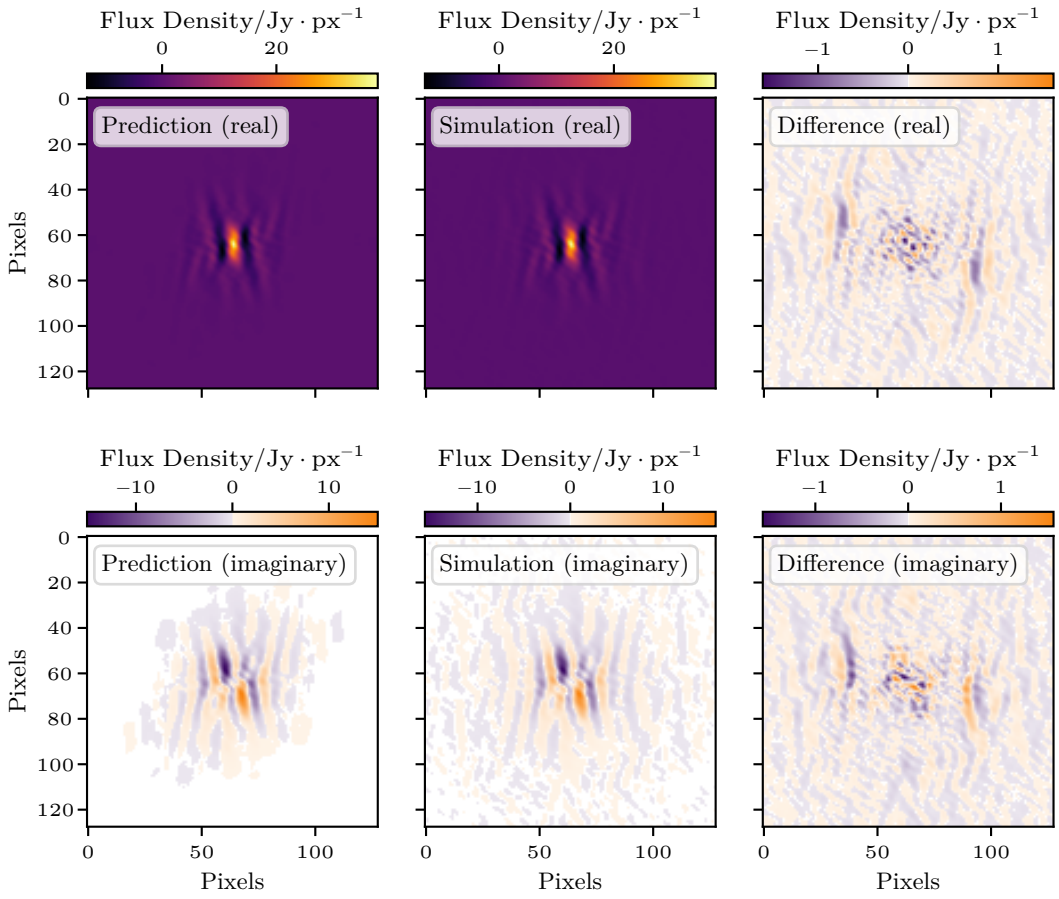

Figure 5.6: Exemplary Fourier space predictions generated by the neural network. From left to right: prediction, simulation, and difference between both images. In the top row, the real part is shown, while the imaginary part is visualized in the bottom row.

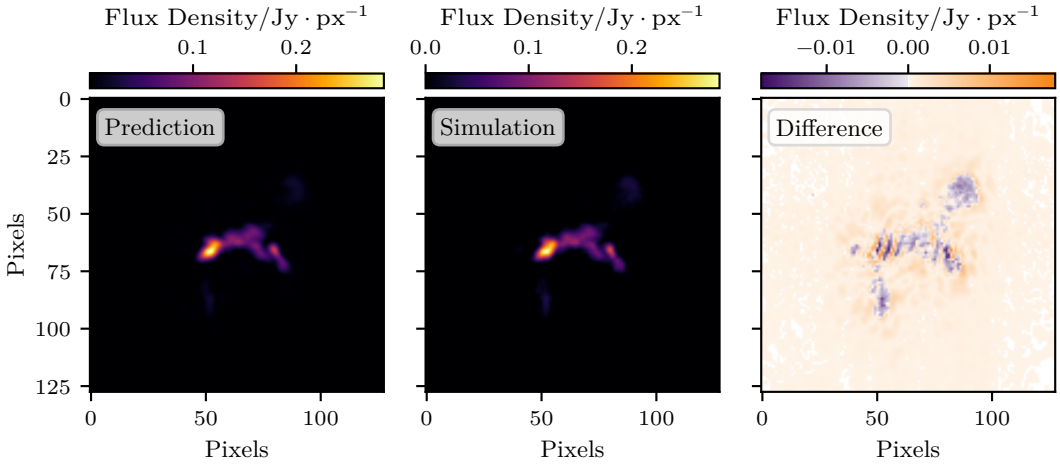

Figure 5.7: Exemplary source predictions generated from the predicted real and imaginary part displayed in Figure 5.6. From left to right: prediction, simulation, and difference between both images.

5.2.2 Image-based Normalization

As already mentioned in Section 5.2, two different normalization techniques were used. The impact of the second image-based method on the loss curve and example images is discussed in this subsection, while the comparison between both methods based on the evaluation of a whole test data set is performed in Chapter 6.

The corresponding loss curve is shown in Figure 5.8. The progression is very similar to Figure 5.5, but the validation loss is less spiky. Again, a gap forms between training and validation loss, but comparing it to the previous versions is not meaningful since the absolute loss values change due to the different normalization techniques.

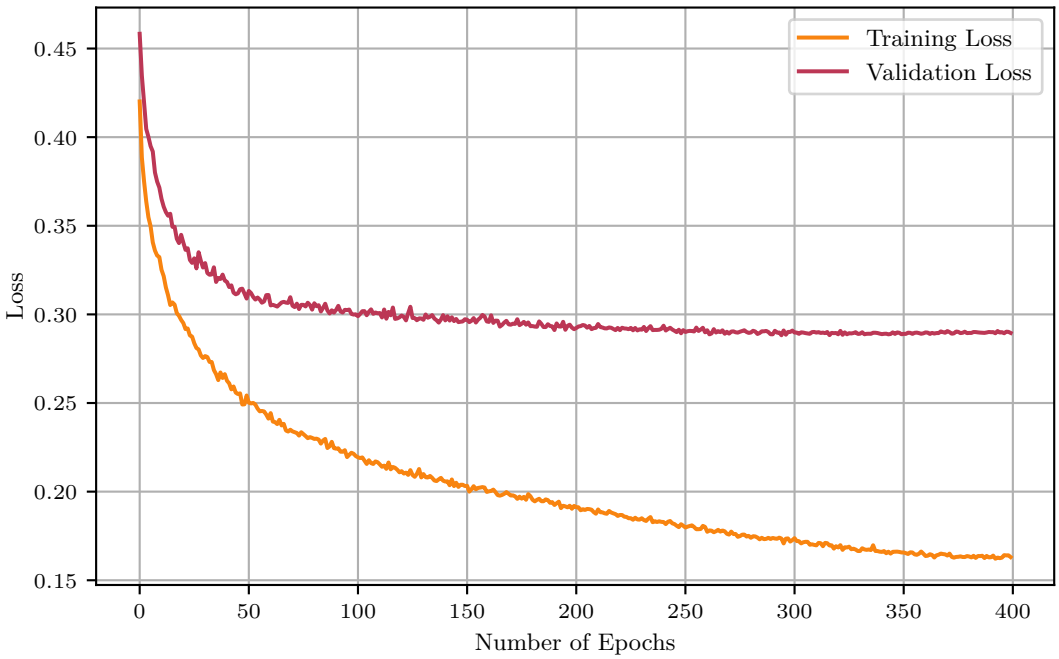

Figure 5.8: Training and validation loss for a model training with wGAN-simulated data utilizing the per-image-based normalization method.

In Figure 5.9 and Figure 5.10, predictions for the same input image used in the previous subsection are shown. Overall, both plots do not differ much from Figure 5.6 and Figure 5.7. At the edges of the imaginary prediction, information is missing. This is not surprising since the VLA is missing coverage at the edges due to its layout, as seen in Figure 4.6. However, these values do not contribute significantly to the reconstruction in image space, and the maximum difference is again one order of magnitude smaller than the maximum flux density. The image space reconstruction in Figure 5.10 validates that the main properties of the source are well reconstructed.

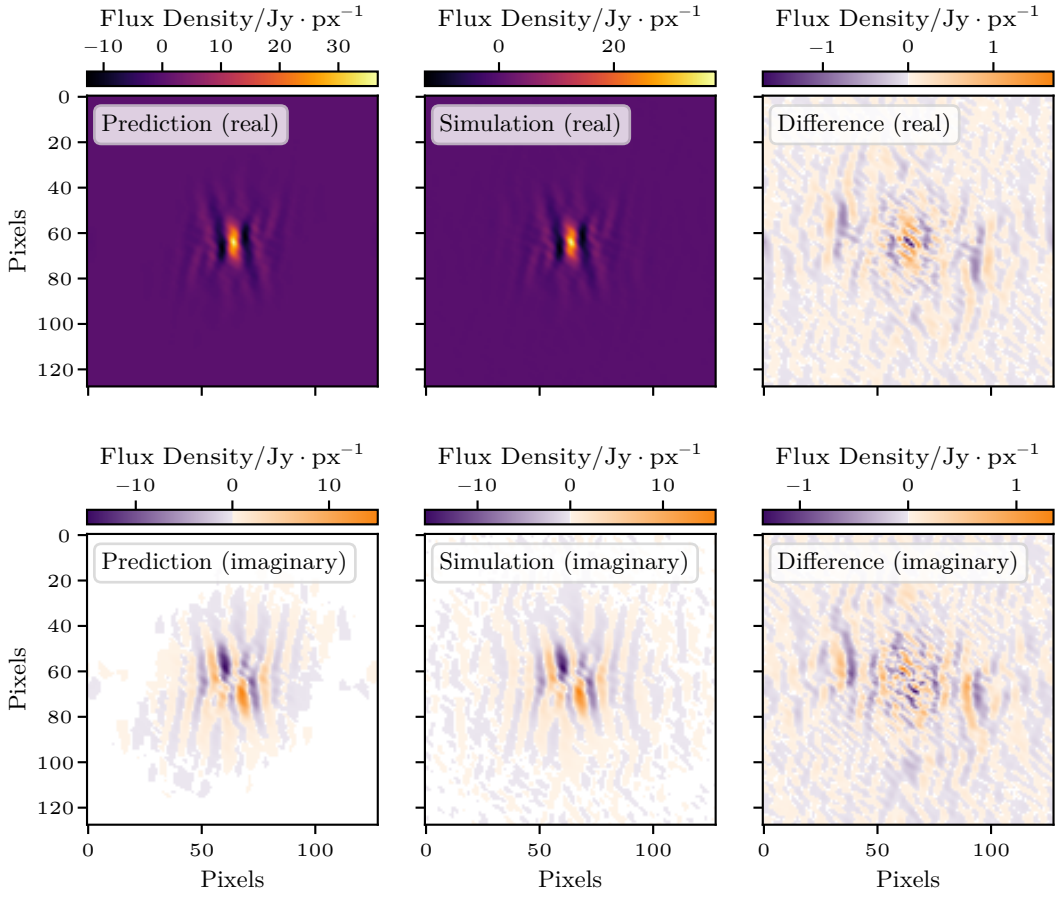

Figure 5.9: Exemplary Fourier space predictions for a wGAN data training normalized with equation (5.5). From left to right: prediction, simulation, and difference between both images. In the top row, the real part is shown, while the imaginary part is visualized in the bottom row.

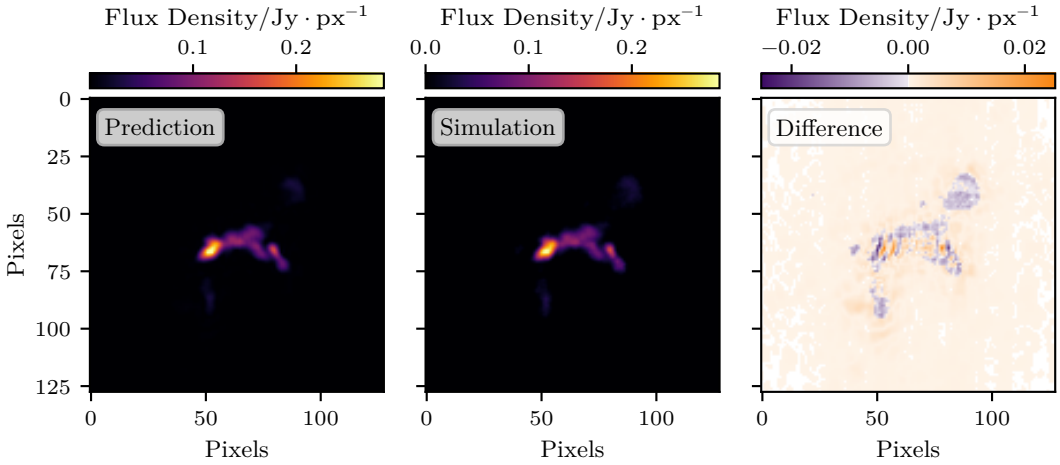

Figure 5.10: Exemplary source prediction for a wGAN data training normalized with equation (5.5). From left to right: prediction, simulation, and difference between both images.

5.2.3 Noise Training

In this subsection, some images of the wGAN data set were replaced with pure white noise images. This serves several causes: First, it helps the network with its task of generalization, and second, it accounts for the fact that some measurements can be too noisy to reconstruct anything but pure white noise. Another reason for the replacement is the frequently asked question of whether the network tries to reconstruct a source in every image, regardless of whether a source is present. Thus, the noise images would be expected to come out of the network unaltered. For both training and validation data, 10 % of the images were swapped out with pure Gaussian noise images, generated with a mean of $\mu = 0$ and a standard deviation of $\sigma = 0.4$, and then Fourier transformed. This ensures that the pure noise images feature a flux density in the range of the telescope's system noise of $\approx 10\,\mathrm{mJy}$, which is the minimum threshold the interferometer is able to observe. The exact numbers of images can be found in Table 5.2.

Table 5.2: Number of source and noise images in the data set

	Training	Validation	Test
# Noise images	5000	1000	1000
# Gan sources	45 000	9000	9000
total	50 000	10 000	10 000

The parameters for this training, such as batch size and learning rate, are the same as for the training in Subsection 5.2.1, except for the normalization layers shown in Figure 5.2. Instead of batch normalization, as used for the previous training process, instance normalization is now performed. Also, the improved normalization technique from Subsection 5.2.2 is utilized. When comparing the loss curves shown in Figure 5.5 and Figure 5.11, one can see that the previously mentioned gap between training and validation loss is smaller now. Again, no signs of overtraining appear as the validation loss decreases until the training ends. This can be attributed to both adding noise images and utilizing the instance normalization, which performs the normalization channel-based for each image in the batch instead of calculating the mean and the standard deviation for the whole batch and all channels.

In Figure 5.12, a neural network prediction in the Fourier space is shown. The results are very similar to those in Figure 5.6, which underlines the robustness of the model. Again, the maximal difference is an order of magnitude smaller than the maximum flux density, and the main structures are well reconstructed. This indicates that the performance of the model has not suffered from the addition of pure noise images. Instead, the gap between validation and training loss decreased.

Similar to the results shown in Subsection 5.2.1, Figure 5.13 displays the Fourier-transformed source prediction of the real and imaginary part shown in Figure 5.12. As already visible in the Fourier space prediction, the source is well reconstructed in image space. The reconstruction is nearly indistinguishable from those produced with the network trained without pure noise images.

To answer the question from the beginning of the subsection, the network's response to a noise image as input is depicted in Figure 5.14. As anticipated, the prediction does not differ from the simulated image, and no source structure is to be found in the image. This shows that the network is able to distinguish between source and noise images and is not forcing a source in every image it gets.

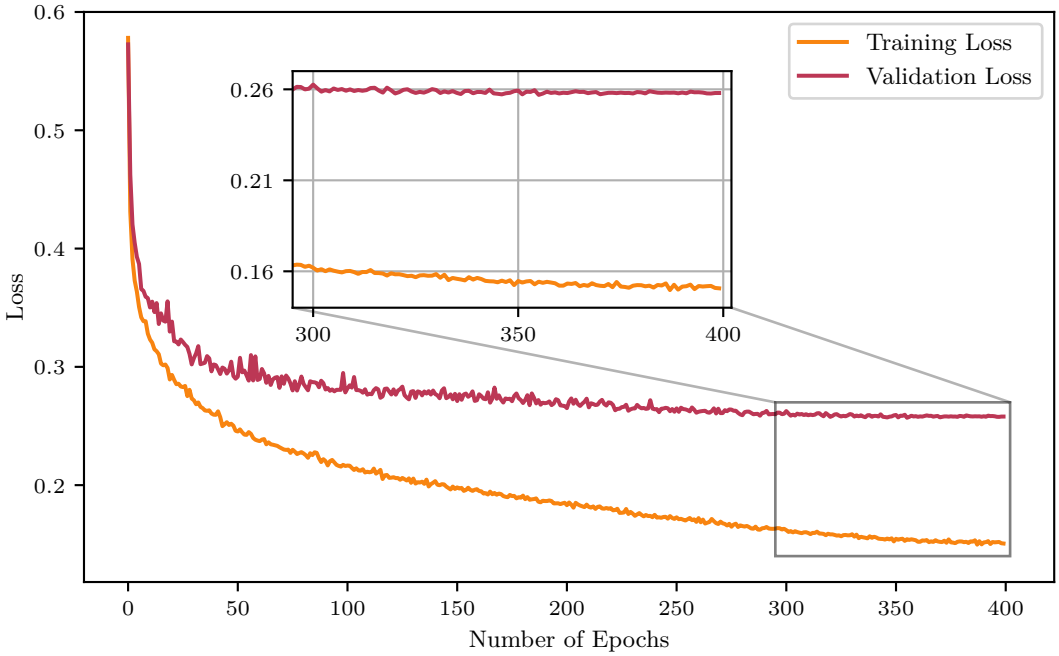

Figure 5.11: Training and validation loss for a model training with wGAN-simulated data and noise images.

All in all, the results show that the model is robust against pure noise images in the data and does not forcefully reconstruct a source in every image. Furthermore, the addition of noise images and improved normalization layers helped to close the gap between training loss and validation loss.

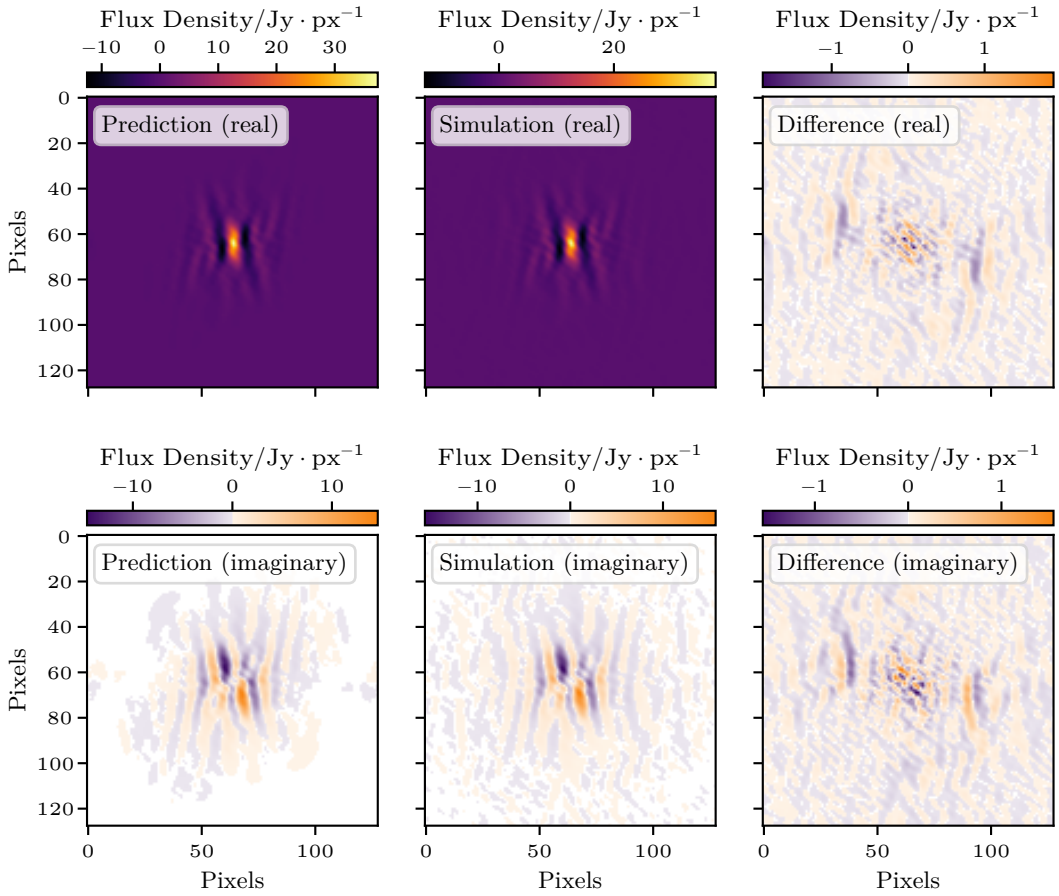

Figure 5.12: Exemplary Fourier space predictions for a training with noise in the data set. From left to right: prediction, simulation, and difference between both images. In the top row, the real part is shown, while the imaginary part is visualized in the bottom row.

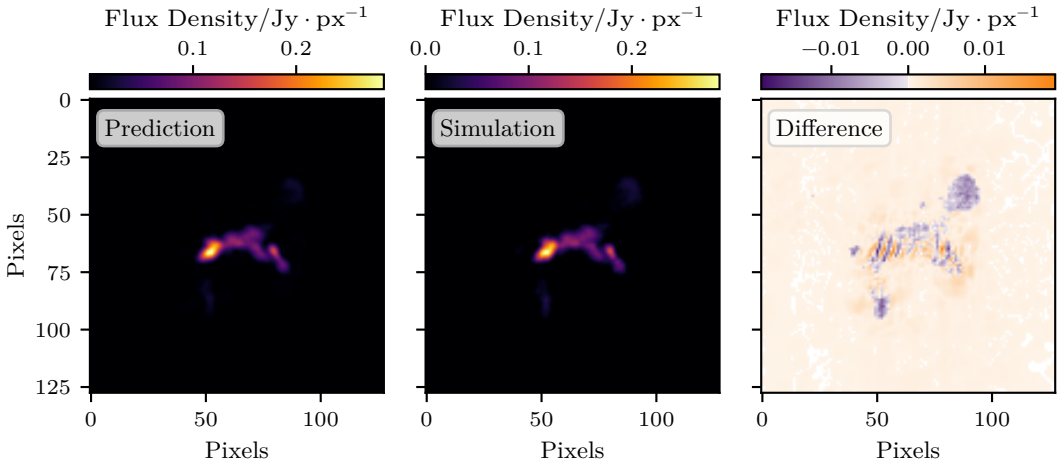

Figure 5.13: Exemplary source predictions for a training with noise in the data set. From left to right: prediction, simulation, and difference between both images.

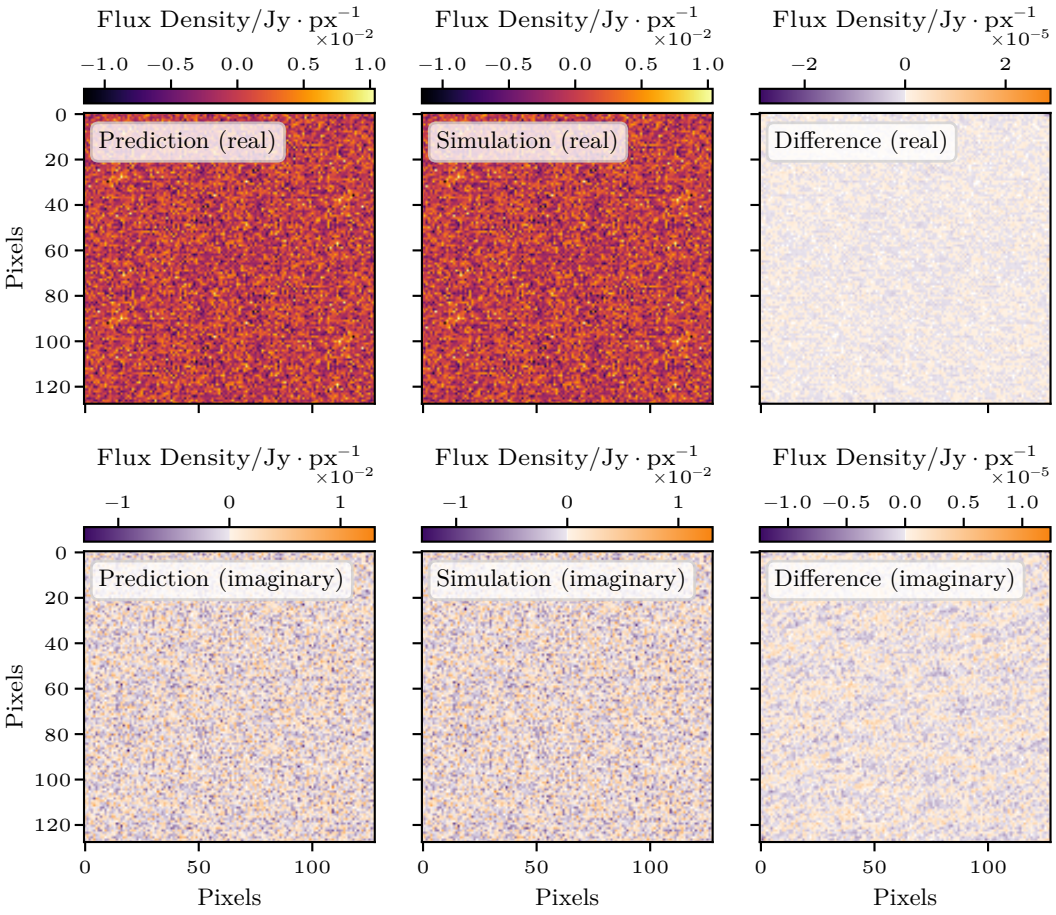

Figure 5.14: Exemplary Fourier space predictions for an image composed of noise. From left to right: prediction, simulation, and difference between both images. In the top row, the real part is shown, while the imaginary part is visualized in the bottom row.

6 Evaluation

In machine learning, quantifiable evaluation methods are necessary in order to validate the results of the models and also to be able to compare trained models with different parameters. Especially when it comes to CNNs, showing exemplary image reconstructions, as in Section 5.2, suffices for a first, quick evaluation, but the impact of varying specific parameters only shows when evaluating a large number of images at the same time. Thus, this chapter introduces different metrics and techniques to quantify the reconstruction performance in image space, which corresponds to the last point in Figure 1.1. In Section 6.1, four metrics are introduced and deployed on the three trained models from Subsection 5.2.1, Subsection 5.2.2 and Subsection 5.2.3, which will be referred to in the following as base, normalized and noise training for simplicity reasons. Section 6.2 provides a detailed look at the performance measured with those four metrics based on the SNR of the images from the test data set. Finally, Section 6.3 examines in detail the performance of the model during the training process instead of solely at the end of the training.

6.1 Metrics

All of the following evaluation methods were deployed on a dedicated test data set consisting of 10 000 wGAN-generated images created with the properties described in Section 4.1. Each of the 10 000 images was fed to network to obtain a prediction, which was then Fourier-transformed to image space. For all upcoming metrics, the goal is to compare said predictions in image space to their simulated counterparts and to quantify deviations.

Source Area

For the first metric, the area of the sources is investigated. This is an important property in terms of astrophysics, as the morphology of cosmic objects allows conclusions to be drawn about, e.g., their emission mechanisms. Comparing the predicted and simulated source area ensures that the morphologies match and no artifacts are created during the network's inference process.

First, in order to compute the source area, contour levels based on 10 % of the maximum simulated flux density are calculated using `matplotlib` [54]. These contour levels surround each part of the source and can be used to compute the area using the Leibniz sector formula [115]. In detail, the link between the path integral, approximated by numerically integrating the contour levels, and the enclosed area is exploited. This is done for both prediction and simulation, and the obtained values are divided, thus creating a ratio. The optimal value is one in the case where both areas are the same. When the predicted source area is smaller than the simulated one, the ratio is smaller than one, and vice versa.

In Figure 6.1, the results for the base, the noise, and the normalized training using the source area evaluation metric are shown. All three metrics peak around the optimal value of one. The mean values of each model are also very close to one and the standard deviations in the regime of 1×10^{-2}. Both the normalized and the noise training outperform the base training in terms of mean and standard deviation, thus indicating the advantage gained from the improved normalization technique. When comparing the results for the noise and the normalized model, it

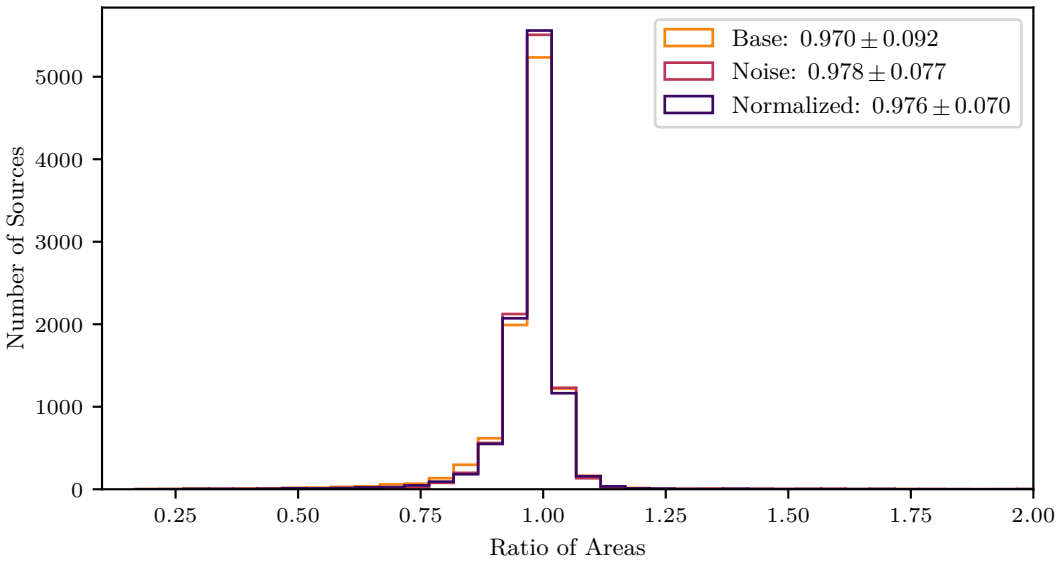

Figure 6.1: Histogrammed ratios of the areas for the wGAN-simulated sources. Results shown for the basis training, the training utilizing noisy images, and the model using the improved normalization method.

turns out that the performance is on a similar level. This proves that the noise model is able to generalize well and is not negatively affected, in terms of the source area, by the addition of the noise images. In Table 6.1, all mean values and standard deviations for the evaluation method investigating the source area are summarized.

Peak Flux Density

Next, the peak flux density is compared between simulation and prediction. This is important to evaluate the reconstruction of the source's core, which is usually the location of the peak flux density. Suppose the core of the source and the peak flux density are well reconstructed. In that case, it enables the comparison with previous measurements of the same source and possibly detects fluctuations, which may be caused by changes in the source itself.

First, the same 10 % flux threshold as before is applied to the images. Second, the peak flux density of both simulation and prediction is computed and compared by creating ratios. The resulting histograms are shown in Figure 6.2. Again, the optimal value is one, while values lower than one indicate an underestimation of the simulated peak flux density, and values higher than one indicate an overestimation. All three models produce results that clearly peak around the optimal value of one, with minor deviations in both directions. The noise model is the one with the lowest standard deviation, while the base model has the closest mean value to one. However, the differences are marginal, so no superior model can be identified using this metric. In summary, all three models are able to reconstruct the peak flux density with high precision. In Table 6.1, the mean values and standard deviations for this metric are listed.

Integrated Flux Density

For the third evaluation metric, the integrated flux density is computed. This property is important because, so far, only the area of the source has been evaluated. However, for a good reconstruction, the integrated flux density must also be regarded. With this metric, the pixel

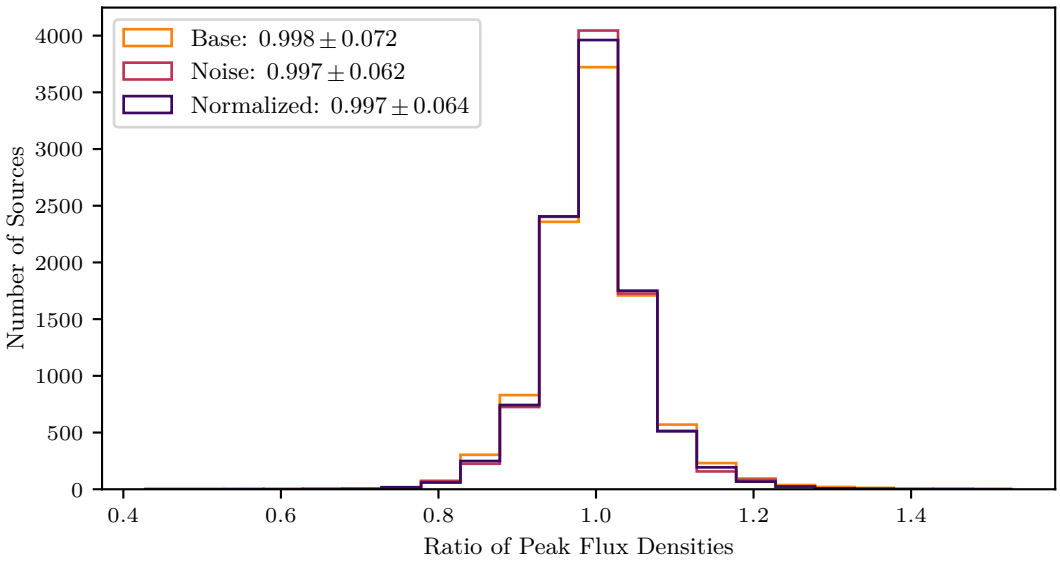

Figure 6.2: Histogrammed ratios of the peak flux densities for the wGAN-simulated sources. Results shown for the basis training, the training utilizing noisy images, and the model using the improved normalization method.

values inside the source are also checked by summing them up and creating a ratio with the corresponding values from the simulated source, thus comparing the integrated flux densities.

The boundaries of the source are defined by the 10 % simulated flux cut. After summing up the corresponding flux densities and creating a ratio, the results shown in Figure 6.3 are obtained. As before, the optimal value is one, and values under or over one correspond to an underestimation or an overestimation, respectively. Again, all the models produce results that peak around the optimal value with small standard deviations. However, for this metric, it is clear that both the noise and normalized model outperform the base model since both mean value and standard deviation are closer to their respective optimal value. The noise model especially stands out since it provides a mean value similar to the one from the noise model, but the standard deviation is closer to zero. As with the source area metric, the improved normalization technique offers an advantage over the base model. Furthermore, the noise model is robust enough not to be negatively affected by adding pure noise images. Table 6.1 provides an overview of all three models' mean values and standard deviations for the integrated flux density.

MS-SSIM Index

For the last evaluation metric, the MS-SSIM index [116] was used. For years, this has been a proven and well-known metric in the field of computer vision. It is based on the Single-Scale Structural Similarity (SSIM) [117], which divides the index into three comparison measures: the luminance

$$l(x,y) = \frac{2\,\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\,, \tag{6.1}$$

the contrast

$$c(x,y) = \frac{2\,\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\,, \tag{6.2}$$

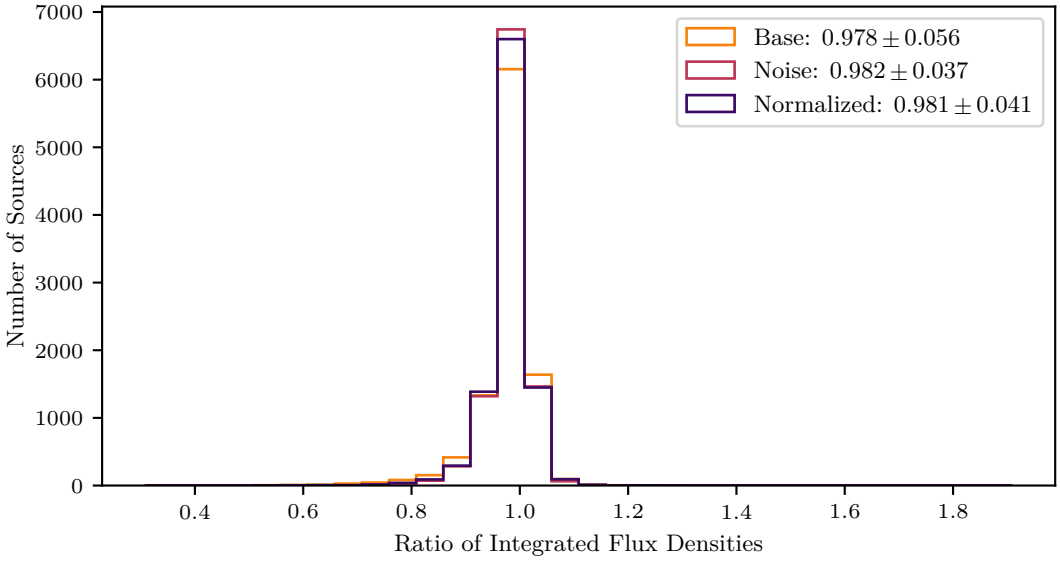

Figure 6.3: Histogrammed ratios of the integrated flux densities for the wGAN-simulated sources. Results shown for the basis training, the training utilizing noisy images, and the model using the improved normalization method.

and the structure

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}\,. \tag{6.3}$$

Here, x and y are the two images to be compared, $\mu_{\{x,y\}}$ the mean of the corresponding image, $\sigma_{\{x,y\}}$ the standard deviation of the corresponding images and σ_{xy} the covariance between the two images. C_1, C_2 and C_3 are given by

$$C_1 = (K_1\,L)^2 \tag{6.4}$$
$$C_2 = (K_2\,L)^2 \tag{6.5}$$
$$C_3 = \frac{C_2}{2}\,. \tag{6.6}$$

Furthermore, L corresponds to the dynamic range of the pixels and $K_1, K_2 \ll 1$ to two small, scalar constants. The reason for the introduction of $C_{\{1,2,3\}}$ can be seen when inserting Equation (6.1), Equation (6.2) and Equation (6.3) into the general definition of the SSIM

$$\mathrm{SSIM}(x,y) = [l(x,y)]^\alpha\,[c(x,y)]^\beta\,[s(x,y)]^\gamma\,, \tag{6.7}$$

which results in

$$\mathrm{SSIM}(x,y) = \frac{(2\,\mu_x\mu_y + C_1)(2\,\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}\,. \tag{6.8}$$

α, β, and γ are parameters used for weighting the different terms and have been set to 1 to obtain Equation (6.8). To prevent numerical instabilities stemming from the denominator of Equation (6.8) if it becomes too close to zero, C_1 and C_2 are added as small constants.

The important properties of both SSIM and MS-SSIM can be directly extracted from Equation (6.8):

- Symmetry: $\mathrm{SSIM}(x, y) = \mathrm{SSIM}(y, x)$
- Boundedness: $\mathrm{SSIM}(x, y) \leq 1$
- Unique maximum: $\mathrm{SSIM}(x, y) = 1$ for $x = y$

Thus, the optimal value utilizing this criterion in our case would be close to one and exactly one if the network was able to reconstruct the simulated target image perfectly.

The critical difference between SSIM and MS-SSIM is that the latter takes different image resolutions into account because not all details of the image are visible at all resolutions. Thus, the predicted and the simulated images are iteratively passed through a low-pass filter and downsampled by a factor of two. At each iteration j, Equation (6.2) and Equation (6.3) are applied to the filtered and downsampled images. After $M - 1$ iterations, Equation (6.1) is also computed. The combination of this leads to the definition of the MS-SSIM

$$\text{MS-SSIM}(x, y) = [l_M(x, y)]^{\alpha_M} \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \,. \tag{6.9}$$

α_m, β_j and γ_j are used for weighting the different terms in Equation 6.9. [116]

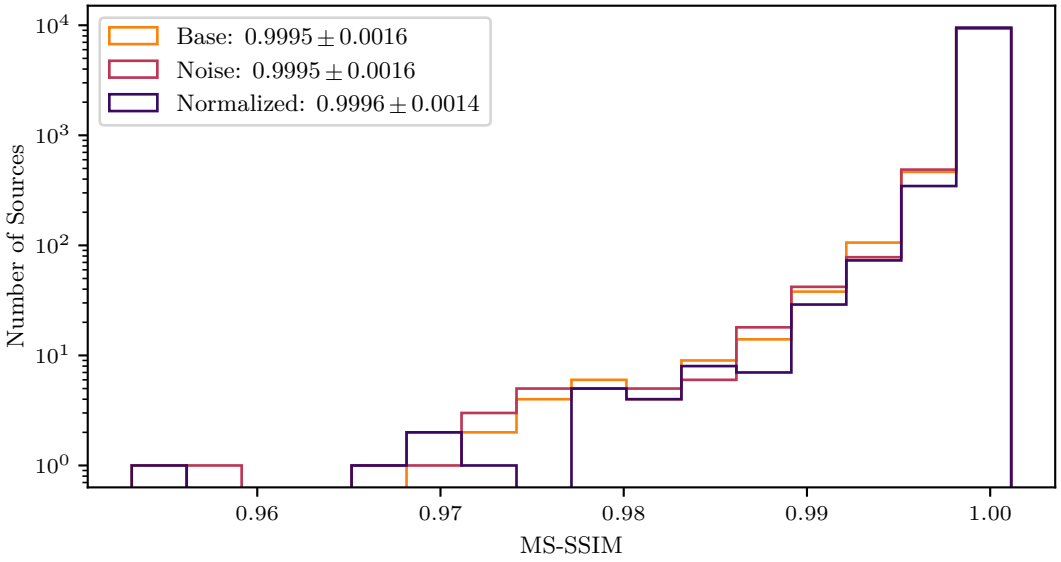

Figure 6.4: Histogrammed ratios of the MS-SSIM for the wGAN-simulated sources. Results shown for the basis training, the training utilizing noisy images, and the model using the improved normalization method.

The results of applying Equation (6.9) to the test data set are shown as a histogram in Figure 6.4. All three models score excellent results for this metric, which is proven by the fact that an extra order of magnitude must be specified for the mean values and the standard deviations to clarify differences. All histograms peak in the last bin, which is the closest to the desired value of one. The number of sources in the second most filled bin is an order of magnitude smaller compared to the last bin. Although the MS-SSIM is not suitable for comparing the three models with each other, it shows that in terms of computer vision problems, the network is able to reconstruct images that are almost indistinguishable from the original, simulated images. The mean values and standard deviations for this metric are summarized in Table 6.1.

Table 6.1: Mean values and standard deviations of the source area, the peak flux density, the integrated flux density, and the MS-SSIM metric. Results are shown for all three investigated models.

Method	Base	Noise	Normalized
Source Area	0.970 ± 0.092	0.978 ± 0.077	0.976 ± 0.070
Peak Flux Density	0.998 ± 0.072	0.997 ± 0.062	0.997 ± 0.064
Integrated Flux Density	0.978 ± 0.056	0.982 ± 0.037	0.981 ± 0.041
MS-SSIM	0.9995 ± 0.0016	0.9995 ± 0.0016	0.9996 ± 0.0014

6.2 SNR-based Evaluation

SNRs play a vital role in radio interferometry and VLBI. Frequently, the SNR acts as a detection threshold for sources. Modern interferometers like Canadian Hydrogen Intensity Mapping Experiment (CHIME) [25], for example, have a SNR-based detection threshold of 15 [72]. Thus, it is interesting to investigate the reconstruction performance of the neural network approach based on the SNR of the evaluated source.

As already mentioned in Section 4.1, the sources have been scaled in a way that their SNRs range between 1 and 100. On the one hand, this was done to provide the network with a variety of possible SNRs, which helps the network to generalize and to distinguish source and noise in the image. On the other hand, this enables the previously mentioned evaluation based on the SNR of the sources. In this section, the metrics introduced in Section 6.1 are analyzed in more detail regarding the SNRs of the sources in the test data set. The distribution of SNRs in the test data set is visualized in Figure 4.4. The general procedure is the same for all metrics:

1. Definition of categories, starting from 0 to 10 SNR, followed by 10-20 and so on until the last category, which contains all SNRs above 60
2. Creation of boolean masks based on the categories and masking the test data set
3. Evaluating the test data set with the chosen metric, thus obtaining 10 000 ratios or MS-SSIM indices, respectively
4. Applying the masks to the results and sorting them into the categories
5. Computing mean and standard deviation in each category, resulting in 14 values for each metric, two for each category
6. Visualizing the results in a scatter plot

With this approach, it is not only possible to compare the SNR categories with each other and verify the starting hypothesis that the results improve with increasing SNR, but it enables another detailed comparison of the three models introduced in Section 5.2 and previously analyzed in Section 6.1. Thus, the above-described procedure is applied to the base, the noise, and the normalized model. The hypothesis of improving results for increasing SNR is based on the fact that for low SNRs, it is more difficult to distinguish the source from the background noise, and thus image reconstruction is more complex. This is expected to be visible in the evaluation metrics.

Source Area

In Figure 6.5, the mean values and standard deviations of the seven SNR categories for the area metric are shown. The values are also listed in Table 6.2. It is directly visible that the expected

outcome is achieved since the values for all models converge to the optimal values of one for the mean and zero for the standard deviation. So, this confirms the theory of improving results for increasing SNR for the area metric. When comparing the three models with each other, it becomes apparent that the base model produces outliers with high standard deviations both for low and high SNR, for example, in categories 10 and 60, where the standard deviation is much higher than in the neighboring groups. Noise and normalized models both perform better than the base model, which underlines that the alterations to the training routine improved the trained models. Already for the first category, arguably the hardest because of the low SNR, the value for the normalized model (0.945 ± 0.098) is closer to one and zero, respectively, than the value for the base model (0.937 ± 0.152). Similar to the normalized model, the noise model features a value of 0.945 ± 0.074 for the first SNR category. Thus, the adapted normalization helped the training routine, especially in terms of generalization, because the number of outliers and the distance to the optimal value were reduced, resulting in lower standard deviations. Also, it helped to improve the results for low SNRs. Furthermore, the noise model is again seemingly unaffected by the pure white nose images in the training data set, as it scores similar values in comparison to the normalized model. This underlines the robustness of the approach.

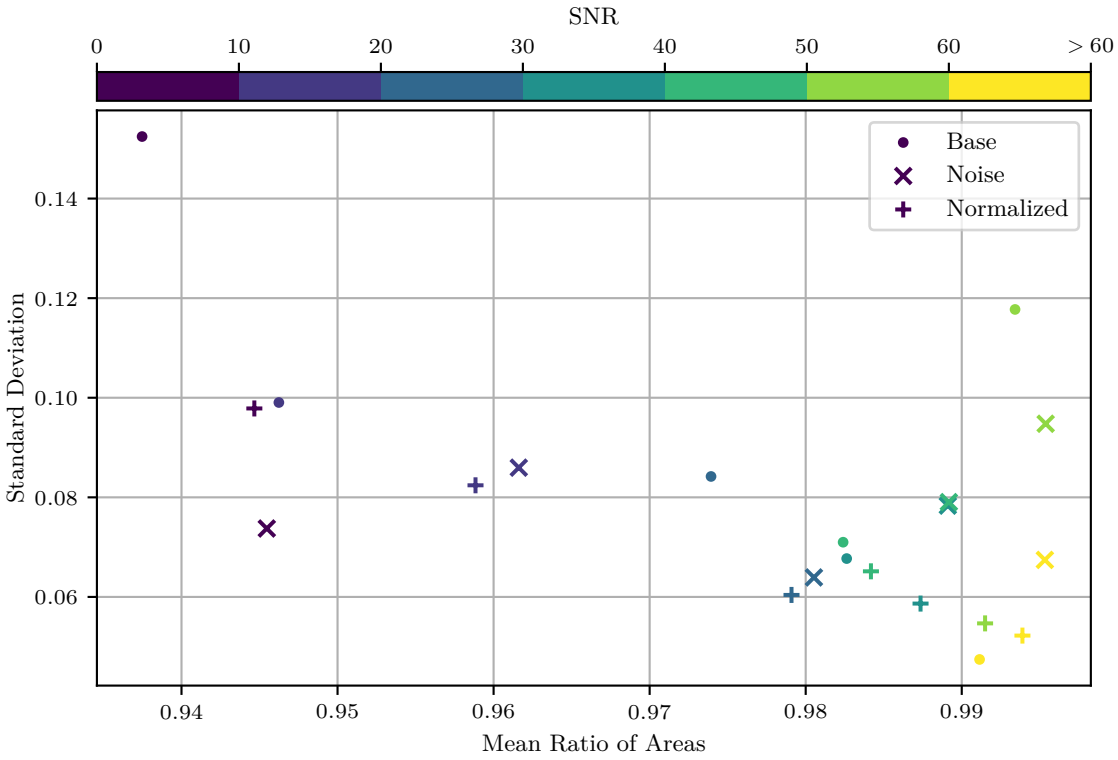

Figure 6.5: SNR-based evaluation for the source area metric. Results shown for the base, the noise, and the normalized model. The SNR categories are color-coded with different shades of blue, green and yellow which get brighter with increasing SNR.

Peak Flux Density

The SNR-based evaluation of the peak flux density metric is visualized in Figure 6.6. The corresponding values are shown in Table 6.3. Overall, all three models perform well, especially

Table 6.2: Mean and standard deviations of the SNR-based area evaluation for the base, the noise, and the normalized model

	Base			Noise			Normalized		
SNR	mean		std	mean		std	mean		std
10	0.937	±	0.152	0.945	±	0.074	0.945	±	0.098
20	0.946	±	0.099	0.962	±	0.086	0.959	±	0.082
30	0.974	±	0.084	0.981	±	0.064	0.979	±	0.060
40	0.983	±	0.068	0.989	±	0.078	0.987	±	0.059
50	0.982	±	0.071	0.989	±	0.079	0.984	±	0.065
60	0.993	±	0.118	0.995	±	0.095	0.991	±	0.055
> 60	0.991	±	0.047	0.995	±	0.067	0.994	±	0.052

regarding the mean values. Deviations from the optimal value of one appear in the third decimal place. The normalized model provides the lowest standard deviations of all models, reinforcing the impression gained from the area metric. In terms of performance per SNR category, outside of category 10, all SNRs score comparable results. For category 10, the values are considerably worse than for every other category. Both effects are expected as this metric compares the peak flux density, which is not as affected by high noise as the rest of the source. All in all, all models score excellent results in most SNR categories, with the normalized model being the best by a slim margin due to the smallest standard deviations.

Table 6.3: Mean and standard deviations of the SNR-based peak flux density evaluation for the base, the noise, and the normalized model

	Base			Noise			Normalized		
SNR	mean		std	mean		std	mean		std
10	1.004	±	0.103	0.994	±	0.068	0.993	±	0.071
20	0.994	±	0.068	0.996	±	0.062	0.996	±	0.062
30	0.997	±	0.066	0.997	±	0.059	0.997	±	0.062
40	1.000	±	0.067	1.000	±	0.060	0.998	±	0.061
50	1.001	±	0.069	1.000	±	0.063	0.998	±	0.065
60	0.997	±	0.072	1.001	±	0.065	0.999	±	0.068
> 60	0.999	±	0.068	0.997	±	0.065	0.996	±	0.064

Integrated Flux Density

Figure 6.7 shows the distribution of the SNR-based evaluation of the integrated flux density, Table 6.4 summarizes the values. Similar to the peak flux density, all three models score good results even for the lowest SNRs, so deviations are visible only on small scales. For the first three categories, the models diverge the most, with the noise model outperforming the rest by a slim margin. For higher SNRs, the differences get smaller, and the base, the noise and, the normalized model each produce mean values close to one and standard deviations close to zero. For all categories, the noise model provides the slightest standard deviations for each category. Furthermore, the general trend of improving results for increasing SNR is also visible for this evaluation metric with no significant outliers. To summarize, the models perform equally

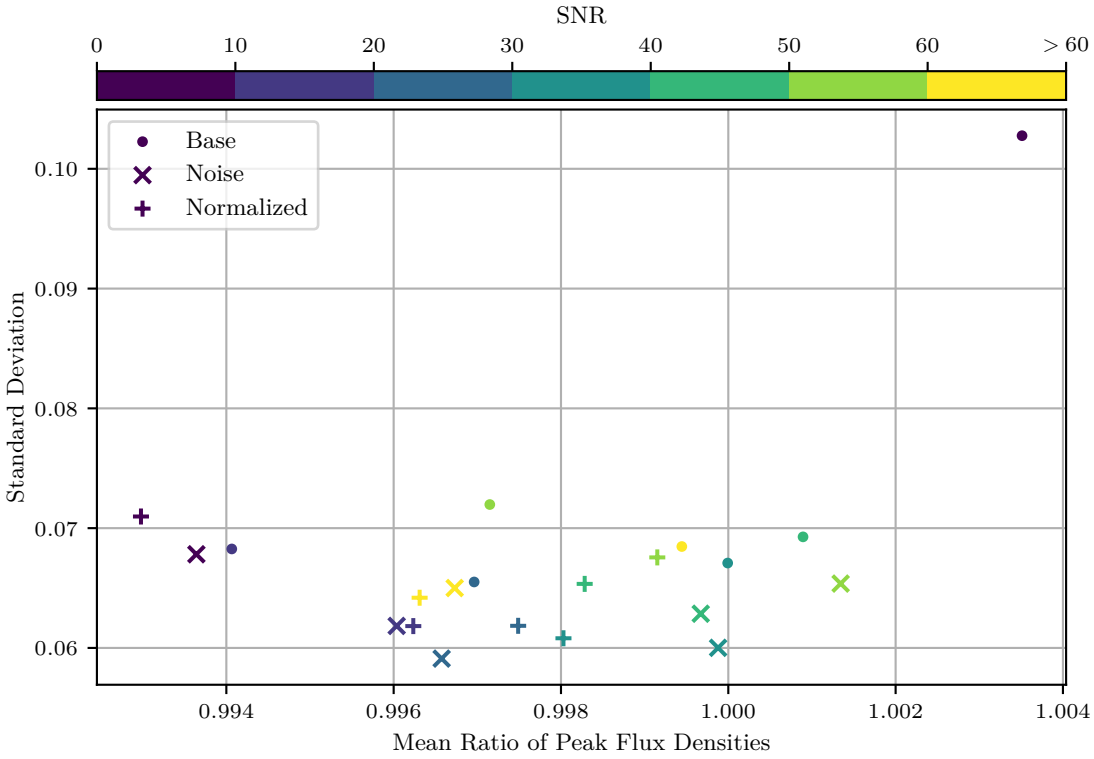

Figure 6.6: SNR-based evaluation for the peak flux density metric. Results shown for the base, the noise, and the normalized model. The SNR categories are color-coded with different shades of blue, green and yellow which get brighter with increasing SNR.

well for the high SNR categories, but for the low SNR categories, the noise model and the normalized model outperform the base model, with the noise model scoring the best results. The improved normalization technique applied to the normalized and the noise model helped the network generalize better, which can be seen from the small standard deviations and significantly improved the performance for low SNR sources. Again, the general approach is robust enough not to be influenced by the noise images in the training data set, as indicated by the good performance of the noise model.

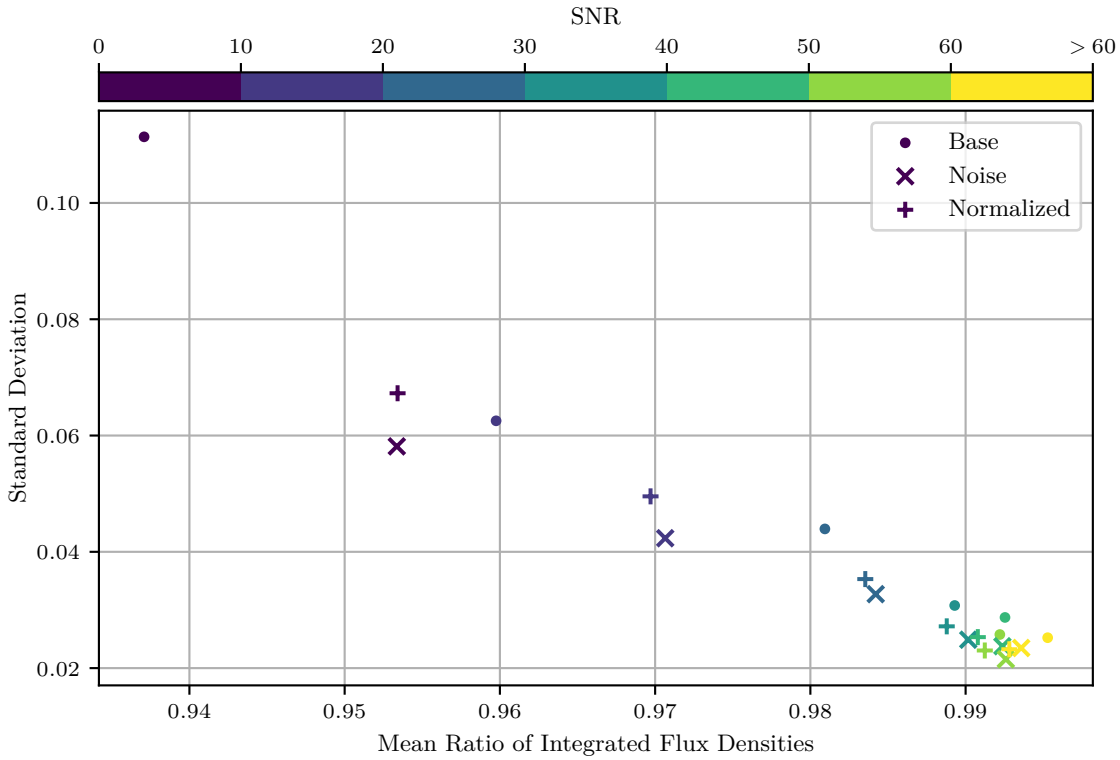

Figure 6.7: SNR-based evaluation for the integrated flux density metric. Results shown for the base, the noise, and the normalized model. The SNR categories are color-coded with different shades of blue, green and yellow which get brighter with increasing SNR.

MS-SSIM Index

Finally, the results for the SNR-based evaluation of the MS-SSIM are visualized in Figure 6.8. The associated values are listed in Table 6.5. For this metric, the deviations between the different models and the different SNR categories are even smaller than for the previous metrics. This is especially visible for the three categories with the highest SNRs, where four decimal places are necessary to see the differences between the models and between the categories. Again, the results improve for increasing SNRs, and in the categories with low SNR, the normalized model has the edge over the base model and the noise model. All in all, performance differences between the three models are small and only significant for the first two categories. However, the general hypothesis of improving results for increasing SNR also applies to the MS-SSIM evaluation metric.

Table 6.4: Mean and standard deviations of the SNR-based integrated flux density evaluation for the base, the noise, and the normalized model

	Base			Noise			Normalized		
SNR	mean		std	mean		std	mean		std
10	0.937	±	0.111	0.953	±	0.058	0.953	±	0.067
20	0.960	±	0.063	0.971	±	0.042	0.970	±	0.050
30	0.981	±	0.044	0.984	±	0.033	0.984	±	0.035
40	0.989	±	0.031	0.990	±	0.025	0.989	±	0.027
50	0.993	±	0.029	0.992	±	0.024	0.991	±	0.025
60	0.992	±	0.026	0.993	±	0.022	0.991	±	0.023
> 60	0.995	±	0.025	0.994	±	0.023	0.993	±	0.023

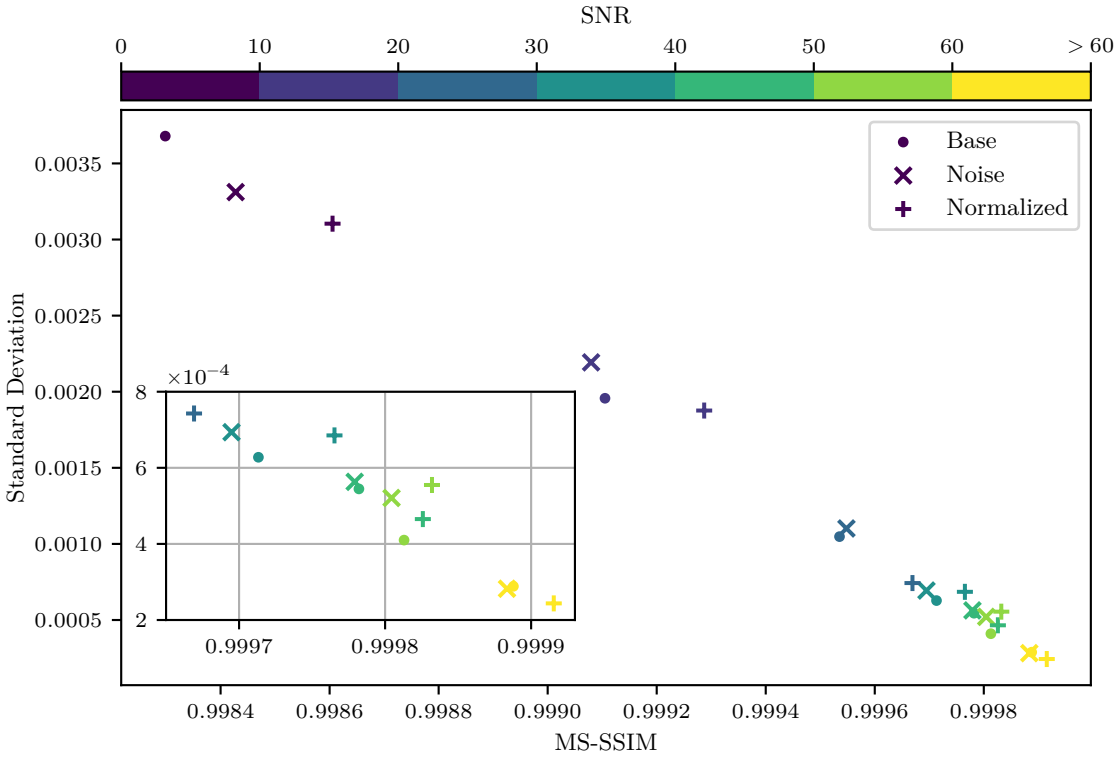

Figure 6.8: SNR-based evaluation for the MS-SSIM. Results shown for the base, the noise, and the normalized model. The SNR categories are color-coded with different shades of blue, green and yellow which get brighter with increasing SNR.

Table 6.5: Mean and standard deviations of the SNR-based MS-SSIM evaluation for the base, the noise, and the normalized model

	Base			Noise			Normalized		
SNR	mean		std	mean		std	mean		std
10	0.9983	±	0.0037	0.9984	±	0.0033	0.9986	±	0.0031
20	0.9991	±	0.0020	0.9991	±	0.0022	0.9993	±	0.0019
30	0.9995	±	0.0010	0.9995	±	0.0011	0.9997	±	0.0007
40	0.9997	±	0.0006	0.9997	±	0.0007	0.9998	±	0.0007
50	0.9998	±	0.0005	0.9998	±	0.0006	0.9998	±	0.0005
60	0.9998	±	0.0004	0.9998	±	0.0005	0.9998	±	0.0006
> 60	0.9999	±	0.0003	0.9999	±	0.0003	0.9999	±	0.0002

6.3 Evaluation over Epochs

So far, only the performance of the model at the end of the training after 400 epochs has been analyzed. In order to examine the effects of noise images in the training data set or an improved normalization technique, one can also inspect the reconstruction capability during the course of the training and investigate if these techniques speed up the convergence time. For this purpose, during the training, a temporary model was saved after every tenth epoch for all three models (base, noise, normalized). This enables the utilization of the previously introduced evaluation metrics on said temporary models and, thus, the analysis of the reconstruction capability and how it changes during the model training. For all four evaluation metrics, the strategy is the same:

1. Applying the metric to each of the 40 temporary models for each of the base, the noise, and the normalized model
2. Calculating the mean and the standard deviation of the results
3. Creating three subplots for every possible combination of the three trained models
4. Plotting the mean values against the epoch with the standard deviation as an error band

Utilizing that, it is possible to compare the development of the mean values and the standard deviations of the base, the noise, and the normalized model and compare the convergence speed. In contrast to Section 6.1, the cut value on the simulated flux density is 1 % instead of 10 % for the source area, the peak flux density, and the integrated flux density metric. This way, one is able to analyze the results on an even finer scale.

Source Area

The development of the area reconstruction during the training process is shown in Figure 6.9. Since values smaller than 0 are not physically meaningful, the error band is capped at 0. When comparing the three models, it becomes visible that both noise and normalized models outperform the base model during the whole training process. In particular, the oscillations in the standard deviation, which indicate a temporary departure from the minimum of the loss function, are barely visible or not visible at all for the other two models. This again underlines the robustness of the noise and the normalized model gained by adding noise images and improving the normalization technique. When comparing the noise and the normalized model, both models perform similarly, especially regarding the mean value. For the standard deviation, the noise model scores lower

values for the later epochs. This again indicates that the noise model is not negatively affected by the pure white noise images.

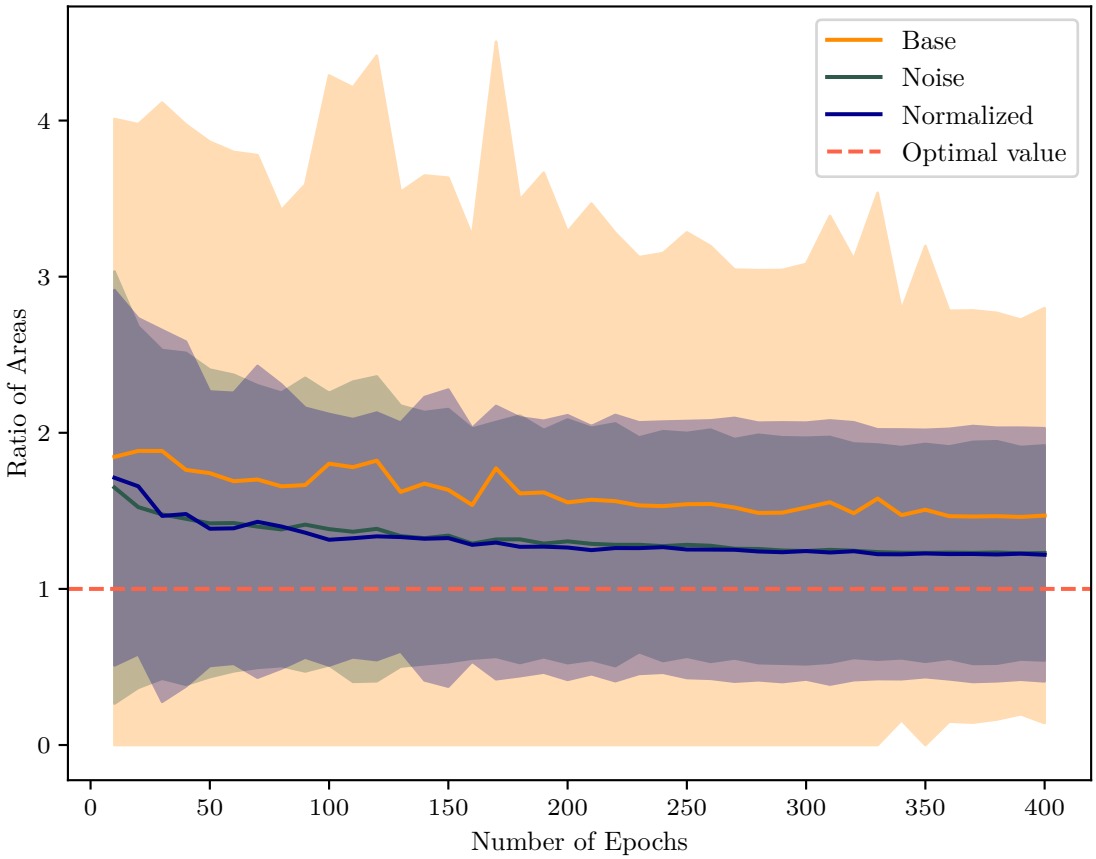

Figure 6.9: Evaluation over epochs for the area metric. Ratio of areas shown against number of epochs. The results for the base (orange), the noise (green), and the normalized (blue) model are shown. The lines correspond to the mean values; the error bands are created by adding and subtracting the standard deviation from the mean.

Peak Flux Density

The results of the temporary model analysis for the peak flux density metric are shown in Figure 6.10. First of all, it becomes apparent that the base model produces massive spikes in the standard deviation with almost a factor of two to the optimal value. These oscillations get smaller during the course of the training, but this shows that the base model is much less stable compared to the other two models in terms of predicting the peak flux density. The mean value also spikes during the first 300 epochs of the training, which is not visible for the noise and the normalized model. Furthermore, the noise and the normalized model score similar results for both the mean values and the standard deviations, with the normalized model providing fewer fluctuations in the first 150 epochs. or the peak flux density, the training of the normalized model could have been terminated after about 50 epochs, and it would have achieved comparable results to the base model, which was trained for 400 epochs. This means that by using the improved normalization technique it is possible to reduce computing time by $\approx 87.5\,\%$ and still get the same good results for the peak flux density as without the improved normalization.

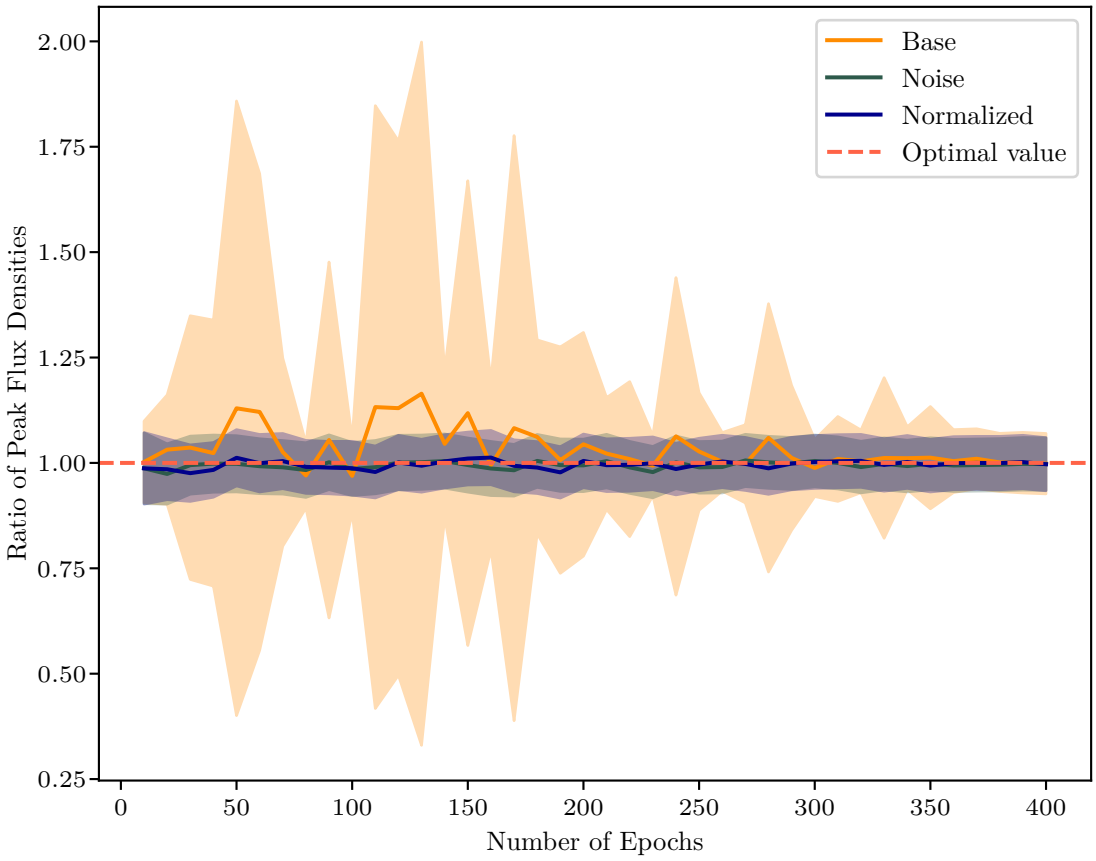

Figure 6.10: Evaluation over epochs for peak flux density metric. Ratio of peak flux densities shown against number of epochs. The results for the base (orange), the noise (green), and the normalized (blue) model are shown. The lines correspond to the mean values; the error bands are created by adding and subtracting the standard deviation from the mean.

Integrated Flux Density

In Figure 6.11, the results of the evaluation over epochs analysis for the integrated flux density are visualized. Similar to the results for the source area metric, the normalized model and the noise model perform on a similar level while clearly outperforming the base model. Furthermore, the standard deviations computed for the base model spike multiple times, indicating instabilities in the learning process that are much smaller or do not occur at all for the other two models due to their specifications. In contrast to the result of the base model in Figure 6.10, the mean values and the standard deviations of the base model do not improve significantly during training but remain mainly on the same level. The noise and the normalized model, however, improve their reconstruction capability and thus show a learning effect regarding the integrated flux density metric. In summary, adding noise images to the training data set and improving the normalization technique significantly raise the model's capability to enhance its reconstruction for the integrated flux density and provide an overall more stable training process.

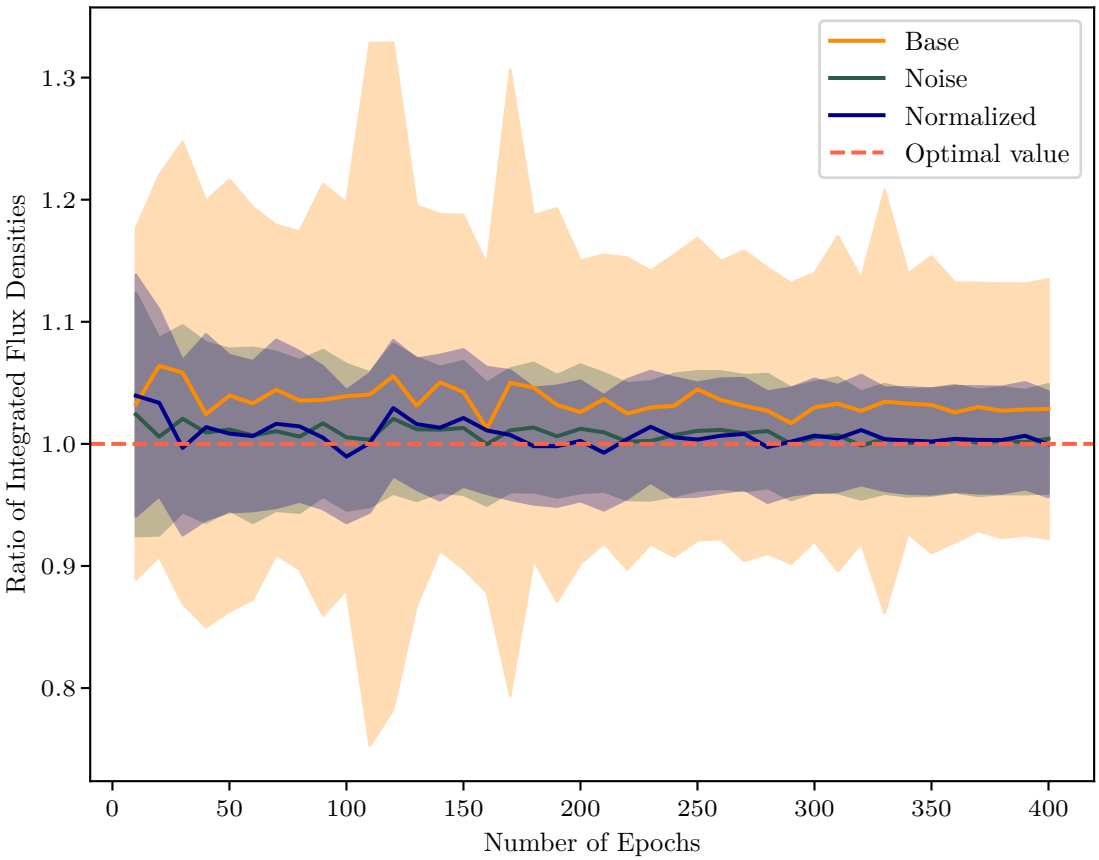

Figure 6.11: Evaluation over epochs for integrated flux density metric. Ratio of integrated flux densities shown against number of epochs. The results for the base (orange), the noise (green), and the normalized (blue) model are shown. The lines correspond to the mean values; the error bands are created by adding and subtracting the standard deviation from the mean.

MS-SSIM Index

The results of the evaluation over epochs analysis for the MS-SSIM index are shown in Figure 6.12. Since values over one are not possible for this metric, the upper limit of the error band is capped at one. Similar to the results of Section 6.1 and Section 6.2, the differences between the three models are only visible on a small scale, in this case in the third decimal place. Similar to the previous results, the noise and the normalized model both score results on a similar level and outperform the base model for every epoch in terms of mean values and standard deviations. Also, there are no significant spikes in the standard deviation for any of the models. Thus, the normalization technique's effectiveness and the noise model's robustness are again verified.

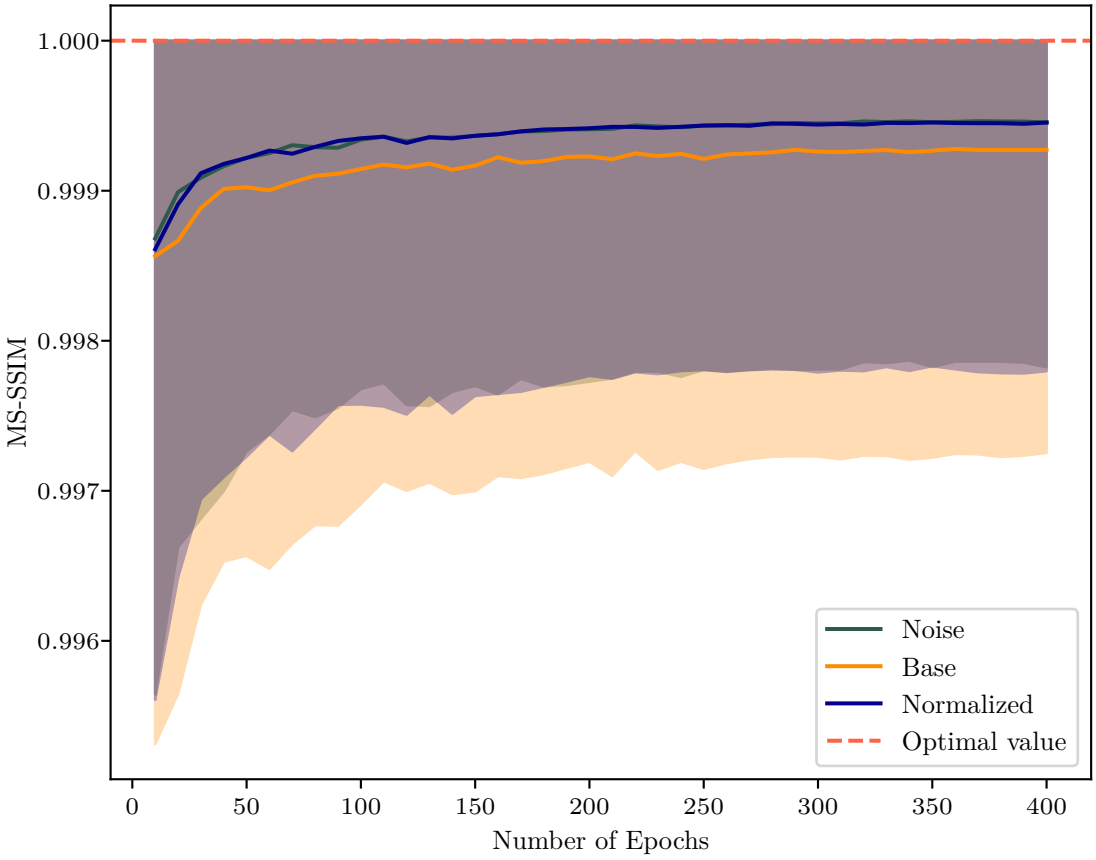

Figure 6.12: Evaluation over epochs for the MS-SSIM. MS-SSIM shown against number of epochs. The results for the base (orange), the noise (green), and the normalized (blue) model are shown. The lines correspond to the mean values; the error bands are created by adding and subtracting the standard deviation from the mean.

In summary, the results of this chapter show that the normalized model is an upgrade over the base model. It outperforms the base model in all four evaluation techniques, provides a better reconstruction capability for low SNR sources, and scores better mean and standard deviations for the evaluation metrics after ten epochs than the base model produces after 400. Furthermore, it allows a more stable training process. The noise model also outperforms the base model for all instances, and the overall scores are similar to those of the normalized model, thus proving the approach's resilience to noise and artifacts in the data. Consequently, the improved normalization technique is used for all subsequent training routines presented in this thesis.

7 Uncertainty Estimation

The ability to provide uncertainty estimations in addition to the reconstruction of a neural network plays a vital role in the integration of deep learning algorithms in the everyday world and gets increasingly popular. Especially in medicine, there are promising approaches for uncertainty estimations computed by neural networks in the fields of radiology [35] and Magnetic resonance imaging (MRI) [31], which share some similarities with the topic of this thesis. By analyzing an uncertainty map, one can get insights into features that are hard to reconstruct and thus need more representation in the training data set. This chapter presents the idea which enables the network to provide uncertainty estimates alongside the predictions and shows results for some of the models introduced in previous chapters. In Section 7.1, the necessary adaptions to architecture and other parameters of the training process are shown. The sampling process transforming the uncertainties from Fourier space to image space is explained in Section 7.2. Finally, the results and evaluation methods of the uncertainty models are presented in Section 7.3.

7.1 Adaptions to the Training Routine

In order to prepare the model introduced in Chapter 5 for the task of providing an uncertainty estimation in addition to the already existing assignment of reconstructing the input images, several critical components of the training routine have to be altered. The first one is the architecture itself. In contrast to previous approaches [97], which were based on two separate architectures, one for the predictions and one for the uncertainty estimation, the existing architecture from Section 5.1 is expanded. This is visualized in Figure 7.1. The architecture largely remains unchanged except for the last convolutional layer, which has an increased number of output channels, from 2 to 4. Thus, the model cannot only predict the real and the imaginary part but also estimate the corresponding uncertainty for both the real and imaginary part, which is shown in Figure 7.1 on the right side. A detailed overview of the layers and their parameters is provided in Table 7.1.

However, more than just increasing the number of output channels is required to enable uncertainty estimation by the model. The general idea behind the expansion of the number of output channels of the last convolutional layer is that the network should not only predict one value per pixel but two: The mean μ and the variance σ^2 of a Gaussian distribution. Thus, the loss function also needs to be adjusted. For this, the so-called β-Negative Log-Likelihood (NLL) loss [101] was chosen. It is based on the NLL of a Gaussian distribution, which is defined by the μ and σ^2 predicted by the architecture:

$$L_{\mathrm{NLL}}(x, y) = \frac{\log \sigma^2(x)}{2} + \frac{(\mu(x) - y)^2}{2\sigma^2(x)}\,. \tag{7.1}$$

x and y represent the input and the target image, respectively. This loss is also referred to as variance attenuation and has the problem of underestimating the gradients of poorly predicted values [113]. Thus, an improved version of Equation (7.1) was proposed in [101] defined by:

$$L_{\beta-\mathrm{NLL}}(x, y) = \mathrm{stop}\left(\sigma^{2\beta}\right) L_{\mathrm{NLL}}(x, y)\,. \tag{7.2}$$

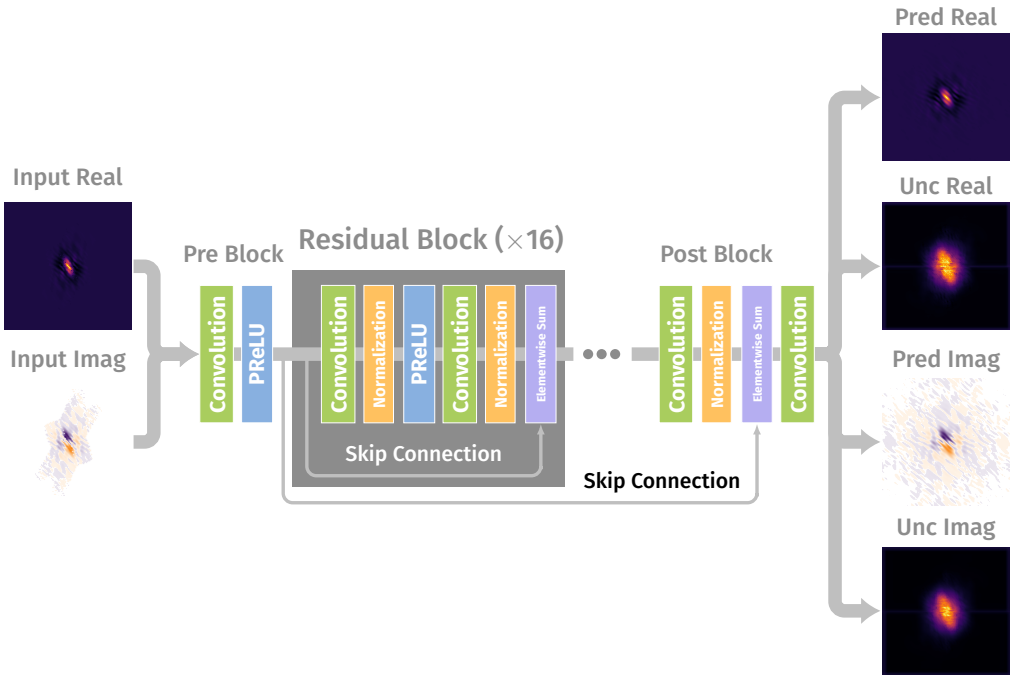

Figure 7.1: Schematic representation of the architecture used for the uncertainty estimation. The central part consists again of 16 residual blocks utilizing skip connections. The architecture is finished with a post block whose final convolution features four instead of two output channels. These additional two output channels are utilized to estimate uncertainty.

Table 7.1: List of parameters for the architecture depicted in Figure 7.1. C_{in} and C_{out} stand for the number of input and output channels, respectively. The main difference compared to Table 5.1 is the number of output channels C_{out}, which was increased from 2 to 4.

Module	Layer	C_{in}	C_{out}	Kernel	Stride	Padding
PreBlock	Convolution	2	64	(9×9)	1	4
	PReLU	64	64	-	-	-
ResBlock (×16)	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	PReLU	64	64	-	-	-
	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
PostBlock	Convolution	64	64	(3×3)	1	1
	Instance Norm	64	64	-	-	-
	Elementwise Sum	64	64	-	-	-
FinalBlock	Convolution	64	4	(9×9)	1	4

Here, "stop" corresponds to the gradient-stop operation of pytorch, which prevents the computation of gradients on this part of the loss function. The inner part of the "stop" in Equation (7.2) consists of the predicted variance with an extra parameter β in the exponent and is utilized for weighting the loss function. This extra term reduces the adverse effects of Equation (7.1) on poorly-predicted values. The role of $0 < \beta \leq 1$ is to control this weighting with the boundary cases of $\beta = 0$ (effectively using Equation (7.1)) and $\beta = 1$ (converting Equation (7.2) to a specialized MSE loss that is capable of estimating uncertainties). As suggested in [101] and derived by tests, for this thesis, β is chosen to be 0.5.

Since the loss function requires that the network predicts the mean and the variance, limiting the possible value range of the two uncertainty channels to only positive values ≥ 0 is necessary. However, as the variance is featured in the denominator of Equation (7.1), it needs to be ensured that the values are not too close to zero to prevent numerical instabilities. Thus, a traditional ReLU function, e.g., cannot be used because it cuts all negative input values to zero, as seen in Equation (3.4). Hence, an adapted version of the ReLU function was created, which is called GeneralReLU and which adds a small value of 1×10^{-10} in order to prevent numerical instabilities. This function is applied after the last convolutional layer of the architecture and only on the channels dedicated to the uncertainty estimation. A visualization of this function is shown in Figure 7.2.

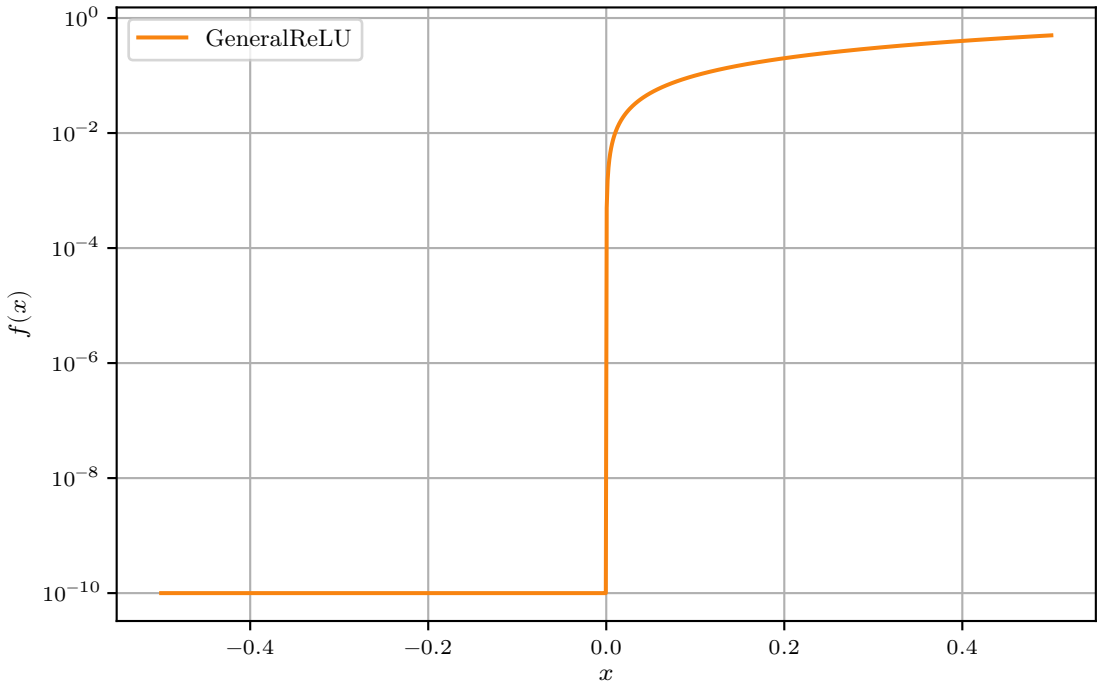

Figure 7.2: Altered ReLU function for the uncertainty estimation, called GeneralReLU. Negative input values x lead to a constant value of 1×10^{-10} for $f(x)$.

7.2 Creating Uncertainty Maps in Image Space

Since the original idea of reconstructing the real and the imaginary part in Fourier space has stayed the same with the implementation of uncertainty estimations, it is necessary to transfer these uncertainty estimations to image space. One possible way to achieve this is outlined in

Figure 7.3. As described in Section 7.1, the output of the neural network consists of a mean value μ and a variance σ^2 on a per-pixel basis. This can be used to define a Gaussian distribution for each pixel in the image. From this distribution, a fixed amount of values n_{samples} is drawn, e.g., 100, resulting in 100 versions of the same pixel and, thus, 100 versions of the same image. In an ideal scenario, the following statements are valid:

1. Each value of each pixel is close to the value of the simulated pixel at the corresponding position, because $\mu_{\text{pred}} \approx \mu_{\text{simulated}}$.
2. The 100 versions of the sampled pixel value are similar to each other, because $\sigma^2 \approx 0$.

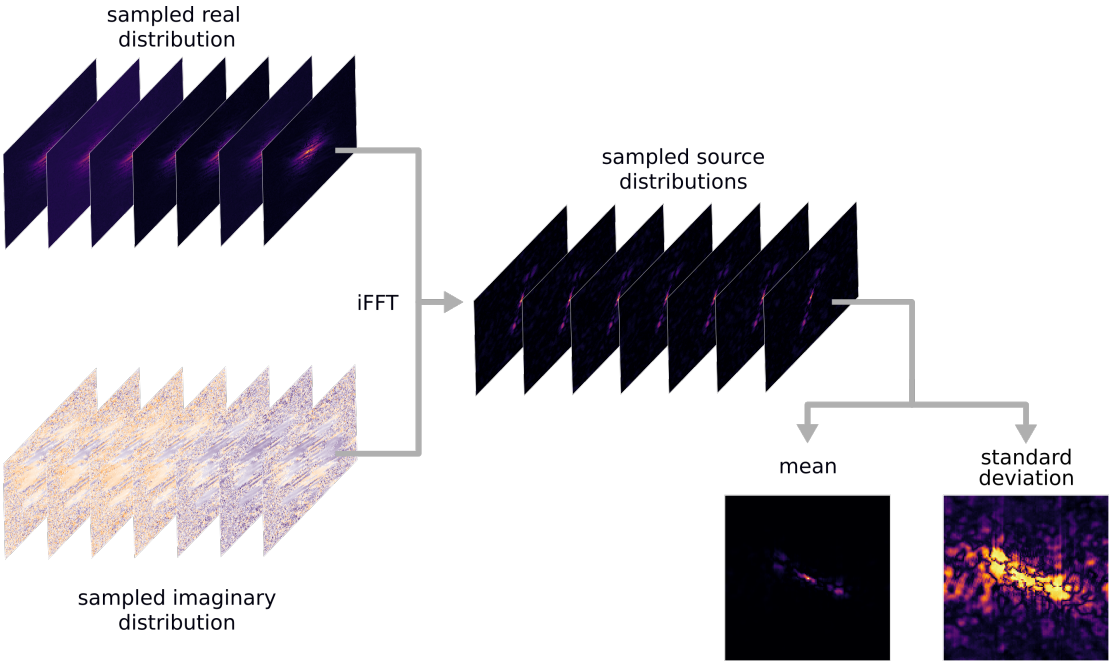

Figure 7.3: Visualization of the sampling process. For both real and imaginary part, n_{samples} pixel values are sampled from the Gaussian distributions defined by μ_{pred} and σ^2. Next, the sampled real and imaginary distributions are transformed using the inverse Fourier transformation, resulting in n_{samples} source distributions in image space. From these, pixel-wise means and a standard deviations are computed, resulting in a prediction in image space and a corresponding uncertainty map. Taken and adapted from [97].

If both of these statements are true, the sampled versions of the real and the imaginary part differ only marginally. In the case of a non-optimal training, however, the n_{samples} versions of the pixels may differ significantly or feature pixels that do not match their simulated counterparts. After sampling, the inverse Fourier transformation is applied to the sampled real and imaginary distributions, thus creating n_{samples} source distributions in image space. From this, the pixel-wise mean and standard deviation are calculated, producing a prediction and a corresponding uncertainty map in image space. Examples of the results of this sampling process are shown in Section 7.3.

However, one particular aspect must be taken into account: If the normalization function (5.5) is used, the output of the neural network needs to be rescaled using Equation (5.6) before inference can occur, as explained in Section 5.2. Thus, the uncertainty estimation of the network

also needs to be adjusted accordingly. This is done by error propagation. For the sake of simplicity, correlation effects are disregarded. Under the assumption of small standard deviations predicted by the network, the following applies:

$$\Delta y = \frac{\mathrm{d}y}{\mathrm{d}x}\Delta x \quad \text{with} \quad y = x_{\text{normalized}} \cdot \sigma_{\text{image}} + \mu_{\text{image}} \tag{7.3}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \sigma_{\text{image}} \quad \Rightarrow \quad \Delta y = \sigma_{\text{image}} \cdot \Delta x\,. \tag{7.4}$$

Δx and Δy are the standard deviations before and after the propagation, respectively. The basis for this is the Taylor series

$$y(x + \Delta x) = y(x) + \frac{1}{1!}\frac{\mathrm{d}y(x)}{\mathrm{d}x} \cdot \Delta x + \frac{1}{2!}\frac{\mathrm{d}^2y(x)}{\mathrm{d}x^2} \cdot (\Delta x)^2 + \frac{1}{3!}\frac{\mathrm{d}^3y(x)}{\mathrm{d}x^3} \cdot (\Delta x)^3 + \ldots \tag{7.5}$$

which can be neglected after the second term because the contribution of the remaining terms is vanishingly small due to small standard deviations Δx. After rearranging some terms in Equation (7.5), $\Delta y = \frac{\mathrm{d}y}{\mathrm{d}x}\Delta x$ is obtained, which is the first term of Equation (7.3). Thus, the predicted standard deviations are multiplied with σ_{image}, which is acquired when applying the normalization using Equation (5.5), in order to propagate the uncertainties.

7.3 Results and Evaluation Methods

In the following subsections, uncertainty models trained on different training data are evaluated using the four evaluation methods introduced in Section 6.1. In accordance with the sampling procedure explained in Section 7.2, the evaluation methods are deployed on a test data set consisting of sampled images. Additionally, the number of simulated source pixel values within the $1\,\sigma$ interval of the predicted pixel values of the source is examined. For this purpose, the estimated mean and standard deviation per pixel are used to mark the $1\,\sigma$ interval for all pixels, clearing the threshold of 1 % of the maximum simulated flux density. This threshold is introduced to prevent potential noise artifacts in the background from skewing the distribution. The number of simulated pixel values lying within the interval is then determined using a boolean mask. This number is then divided by the total number of pixels in the image to obtain a percentage. This procedure is performed for every image in the test data set, thus obtaining one value per image. Eventually, the percentages are illustrated in a histogram. This enables a quick and straightforward evaluation of the agreement between the predicted mean and standard deviation on the one side and the simulated pixel values on the other. High percentages generally indicate a good reconstruction. However, this method has a drawback: Overestimated standard deviations can also cause high percentages, which can thus falsely indicate a good agreement between prediction and simulation. Therefore, the orders of magnitude of both mean and standard deviation are also considered while analyzing the results.

7.3.1 FRI and FRII Sources

First, the GAN-simulated sources introduced in Section 4.1 are used as the training data set for an uncertainty training. The parameters and pre-processing steps described in Section 5.2 are also applied here with the exceptions explained in Section 7.1. Due to the different loss function, which features two parameters to be estimated, the training process is less stable. This is also reflected in the loss curve shown in Figure A.1.

Figure 7.4 illustrates the real part of the neural network prediction in Fourier space. The simulated real part is shown on the top left, the predicted μ and the predicted σ for the real

part on the top right and the bottom left, respectively, and the difference between the μ and the simulated image is visualized on the bottom right. The simulation and the prediction are in good agreement both in shape and in flux density, which is confirmed by the difference plot. There are no major background artifacts visible, and the maximal difference is an order of magnitude smaller than the maximum flux density. Furthermore, the highest values of the estimated uncertainty are located in the region where also the difference has its maximum, which is the expected behavior. Additionally, the estimated uncertainty seems not to be overestimated as the values reside in approximately the same range as the difference values, which is important for the $1\,\sigma$ interval evaluation method introduced above.

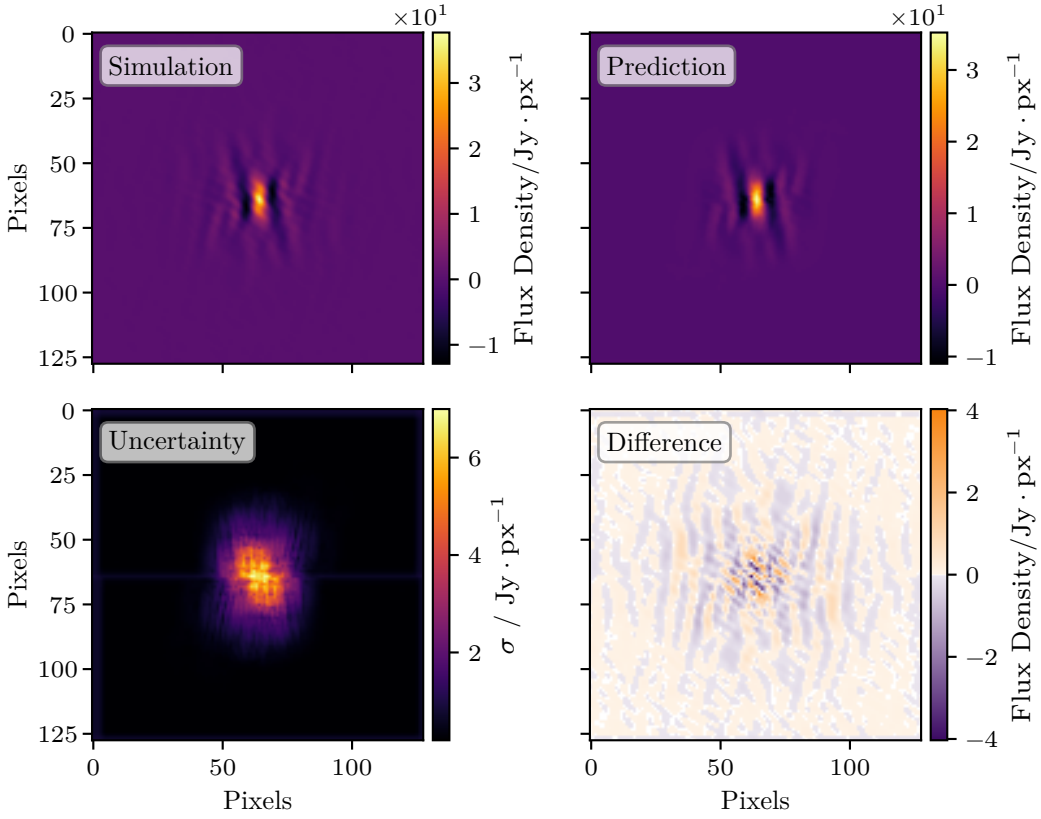

Figure 7.4: Prediction and estimated uncertainty for the real part of a reconstructed GAN-generated source. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference.

Similar conclusions can be drawn from the imaginary part shown in Figure 7.5. Again, the simulated imaginary part is visualized on the top left, the predicted μ and σ on the top right and the bottom left, respectively, and the difference between μ and simulation is illustrated on the bottom right. The central part of the prediction matches the corresponding part of the simulation very well, which can also be seen from the small differences. However, two vertical lines of pixels at the top (and due to the symmetry also at the bottom) of the image feature the same value and thus are not correctly reconstructed. Nevertheless, this is not overly significant since this region only marginally contributes to the source in image space. Also, the regions are not represented in the estimated uncertainty, indicating a possible systematic offset. For the other parts of the images, the uncertainty roughly matches the shape of the highest deviations visible in the difference.

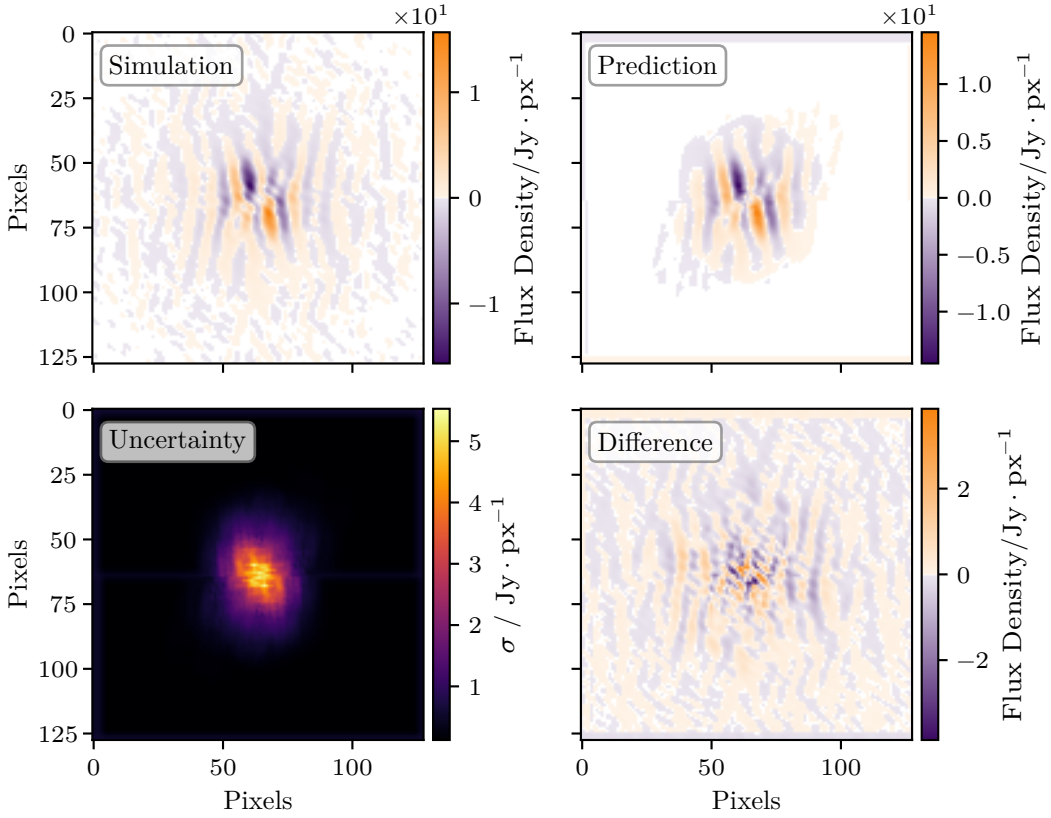

Figure 7.5: Prediction and estimated uncertainty for the imaginary part of a reconstructed GAN-generated source. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference.

After applying the sampling process described in Section 7.2 on the real and the imaginary predictions shown in Figure 7.4 and Figure 7.5, the reconstructed source in image space is obtained, which is visualized in Figure 7.6. The structure of the plot is the same as before: Simulation and prediction on the top half of the image, sampled uncertainty, and difference on the bottom half. Simulation and prediction are in good agreement both in shape and flux density. The central part of the source is well reconstructed, which is confirmed by the small differences. Two small structures above and below the primary source are missing in the reconstruction, probably due to their low flux density. Furthermore, the estimated uncertainty clearly matches the shape of the difference except for one of the missing structures. In terms of flux density values, both difference and estimated uncertainty feature values an order of magnitude smaller than the maximum simulated flux density, and thus, no indication of an overestimation of the uncertainty is present.

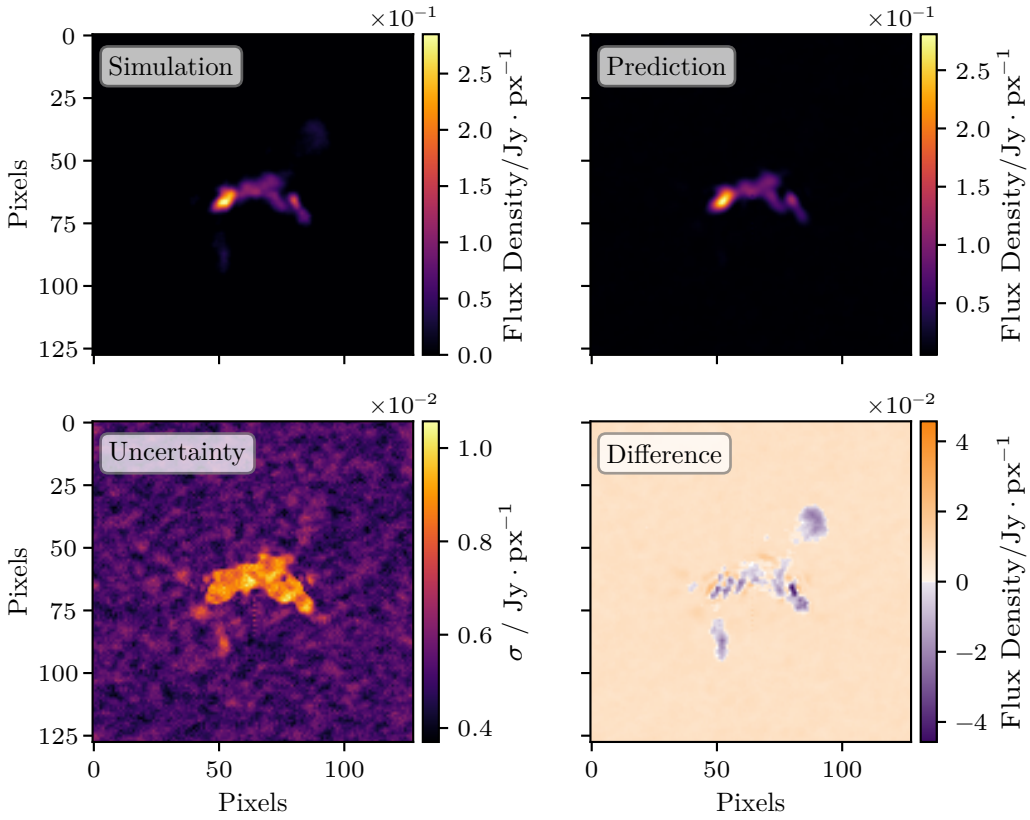

Figure 7.6: Sampled prediction and uncertainty estimation of the GAN-generated source in image space. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference.

In Figure 7.7, the histogram of the percentages of simulated flux densities within the predicted $1\,\sigma$ interval is presented. The distribution is spread from approximately 20 % to 100 %, with a mean of 62.492 % and a standard deviation of 11.890 %. Not all pixels are expected to be within the $1\,\sigma$ interval since, for some sources, parts with low flux density are not well reconstructed, and not always are these parts covered by the uncertainty. Overall, more than half of the pixels are within the interval, which is a good result but leaves room for improvement.

For the last evaluation step, the four metrics introduced in Section 6.1 are applied to a sampled data set of 10 000 sources. The results are plotted in Figure 7.8. On the top half, the source

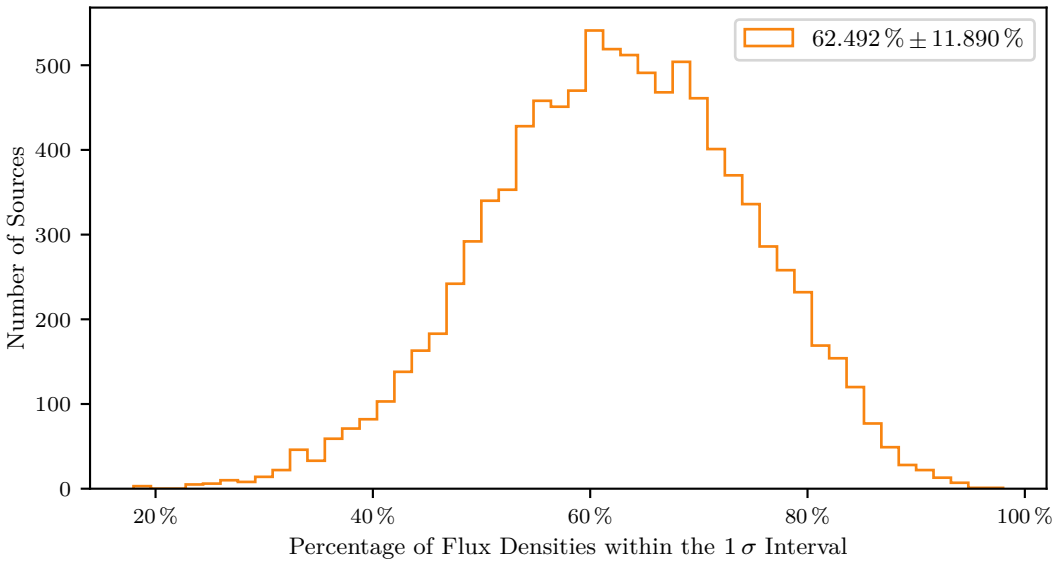

Figure 7.7: Histogram of the percentage of simulated flux densities within the predicted 1 σ interval. The distribution peaks around 60 %, with images reaching nearly 100 % and images going as low as 20 %.

areas (left) and the integrated flux densities (right) are shown, and on the bottom half, the peak flux densities (left) and the MS-SSIM (right) are visualized. All four metrics produce equally good results, with their distributions peaking clearly around their respective optimal values. Furthermore, the mean values are all above 0.9 with minor standard deviations. Thus, the conclusion can be drawn that the model is still able to perform the reconstruction at a high level despite the additional effort required to estimate the uncertainty.

7.3.2 Protoplanetary Disks

Furthermore, an uncertainty model was trained on the simulated protoplanetary disks introduced in Section 4.2. The architecture was adapted in comparison to Section 7.1: the application of the modified ReLU function shown in Figure 7.2 was excluded from the gradient calculation during the backpropagation. Since this operation is based on a statistical constraint, it should not be part of the training process and, therefore, not be modified in any way or contribute to the loss calculation. Also, the number of images in the training, the validation, and the test data set were adjusted. This was necessary due to rising computational costs triggered by the increased image sizes of 512 px × 512 px compared to 128 px × 128 px for the GAN-generated sources from Section 4.1. The training data set now comprises 1500 images, while the validation data set contains 300 images. Still, the 20 % split is in effect. The trained model was then evaluated on a test data set of 1000 additional images. The training and the validation loss are visualized in Figure A.2.

In Figure 7.9, an exemplary real part of the neural network output is visualized. The simulated image is shown on the top right, the predicted μ and σ on the top left and bottom left, respectively, and the difference between the simulation and μ is illustrated on the bottom right. As large parts of the image contain small values, a 90 px × 90 px segment of the central part of the image is shown. Overall, the predicted μ is in good agreement with the simulated values. The structures are also well-reconstructed. This is confirmed by the difference, which shows only small deviations

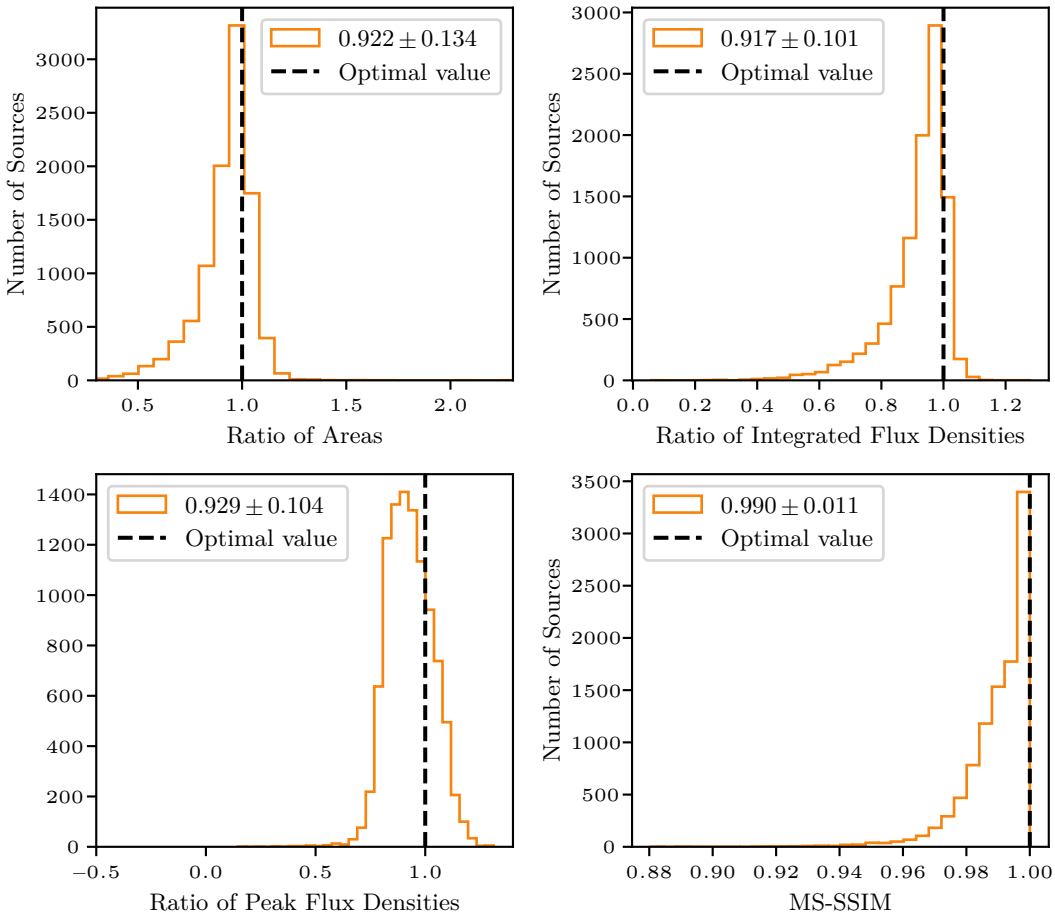

Figure 7.8: Histograms visualizing the results of the four evaluation methods introduced in Section 6.1. From top to bottom, left to right: Source area, integrated flux density, peak flux density and MS-SSIM. For every histogram, the results' respective optimal value, mean, and standard deviation are shown. All four evaluation methods are in good agreement with their respective optimal values.

an order of magnitude smaller than the maximum flux. The uncertainty estimation features higher values in the segments where also the main structures of the prediction are located, which is expected since the reconstruction is not perfect. Additionally, the uncertainty seems not to be overestimated, which is supported by comparing the maximum uncertainty of $\approx 6 \times 10^{-3}$ Jy/px to the maximum flux density of $\approx 6 \times 10^{-1}$ Jy/px.

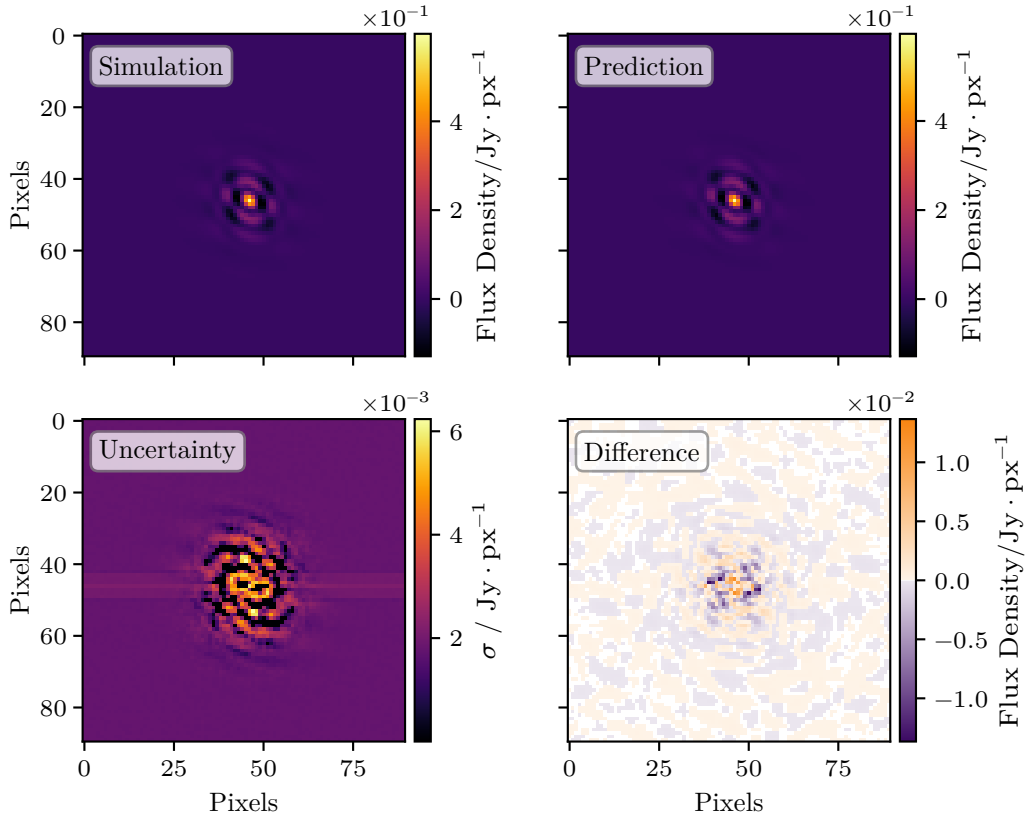

Figure 7.9: Prediction and estimated uncertainty of the real part. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference. For a better visualization a section of 90 px × 90 px in the central part of the image is shown.

The corresponding imaginary part is illustrated in Figure 7.10. Again, the simulated image is shown on the top left, the predicted μ and σ are shown on the top right and the bottom left, respectively, and the difference between simulation and μ is illustrated on the bottom right. Similar to the images displayed in Figure 7.9, just a section of 90 px × 90 px in the central part of the images is shown. The shapes and pixel values of both μ and the simulation match well. Furthermore, the difference shows no major structures, which confirms that the shapes are in good agreement. Also, the maximum difference is an order of magnitude smaller than the maximum flux density of the prediction, which underlines this statement. Similar to Figure 7.9, the estimated uncertainty is two orders of magnitude smaller than the prediction's maximum flux density, indicating that the uncertainty is not overestimated. Additionally, the uncertainty features no artifacts in regions outside of the main structure.

When using the real part from Figure 7.9 and the imaginary part from Figure 7.10 as the input for the sampling process described in Section 7.2, the protoplanetary disk visualized in Figure 7.11 is obtained. Similar to the figures shown before, the top half illustrates the

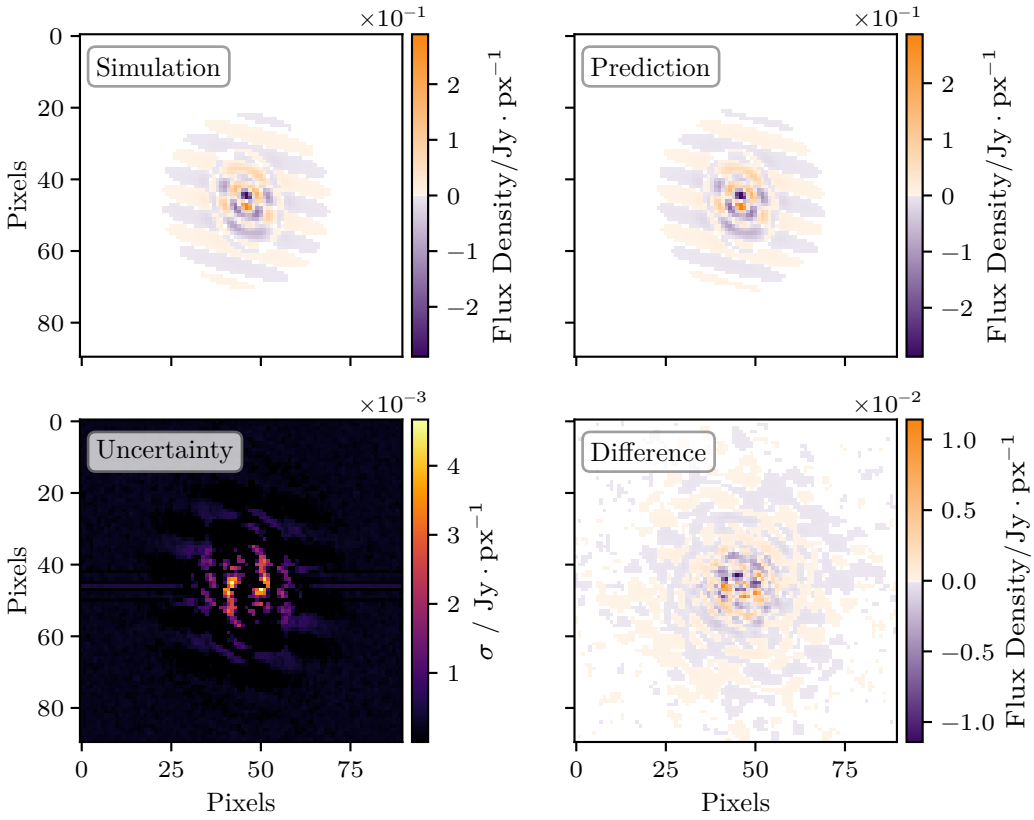

Figure 7.10: Prediction and estimated uncertainty of the imaginary part. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference. For a better visualization a section of 90 px × 90 px in the central part of the image is shown.

simulated protoplanetary disk on the left and the sampled prediction on the right, and the bottom half features the estimated uncertainty on the left and the difference between prediction and simulation on the right. The prediction matches the simulation very well in both shape and pixel values. Only a small underestimation of flux density in the outer ring and the core is visible, as confirmed by the difference. Furthermore, the prediction does not feature artifacts or blurred structures, indicating a good reconstruction. The shape of the estimated uncertainty matches almost completely the structure of the difference and shows no other hotspots, which is in line with the expectations of a reasonable uncertainty estimate.

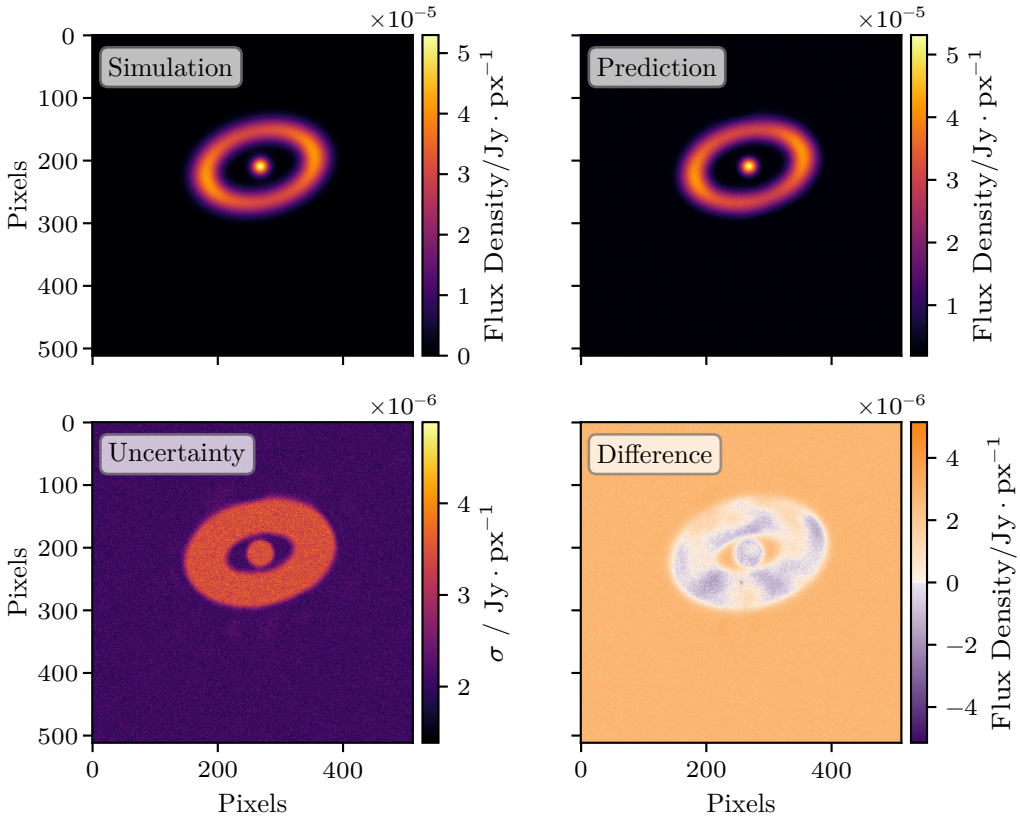

Figure 7.11: Sampled prediction and uncertainty estimation of the simulated protoplanetary disk in image space. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference.

The results of the method described above for evaluating the number of simulated flux densities within the predicted 1 σ interval are displayed in Figure 7.12. Most of the source pixels lie within the interval and thus indicate an excellent reconstruction and a good estimated uncertainty. Furthermore, as confirmed by Figure 7.9, Figure 7.10, and Figure 7.11, there are no signs of a potential overestimation of the uncertainty, validating the statement above.

Finally, Figure 7.13 displays the results of the four evaluation methods described in Section 6.1 applied to a data set of 1000 sampled protoplanetary disks. The source area is shown on the top left, the integrated flux densities on the top right, the peak flux densities on the bottom left, and the MS-SSIM is visualized at the bottom right. All plots show the respective optimal value, the mean, and the standard deviation. Overall, the performance of all four methods is excellent. Every distribution peaks clearly at the optimal value with a mean close to one and a

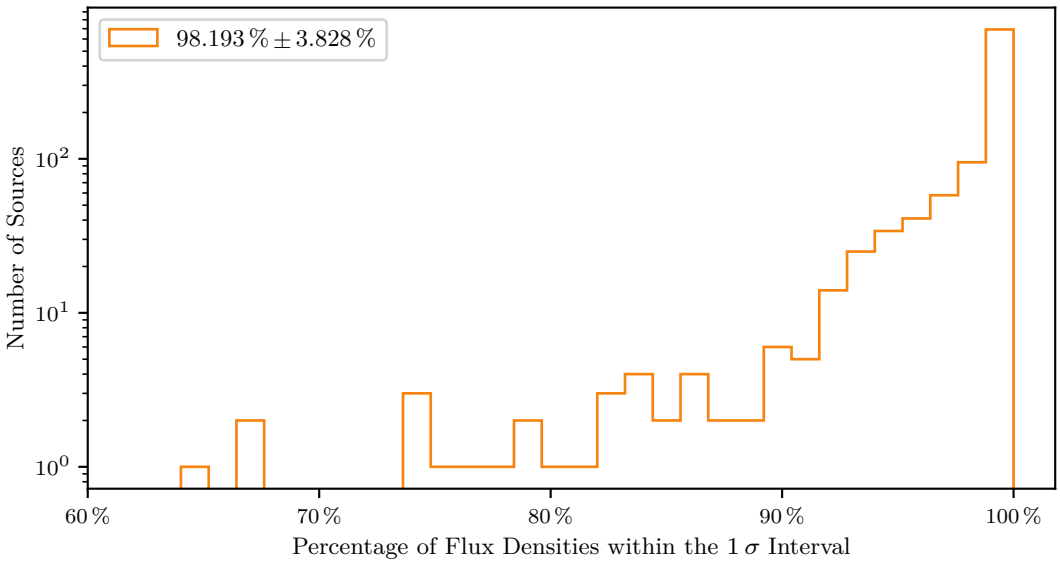

Figure 7.12: Histogram of the percentage of simulated flux densities within the predicted 1 σ interval. The distribution peaks near 100 %, which shows that almost all simulated flux densities of the source are within the 1 σ interval.

small standard deviation. Especially the results for the integrated flux densities and for the peak flux densities show no significant outliers and produce standard deviations an order of magnitude smaller than the mean. The results for the MS-SSIM are nearly indistinguishable good, as they do not differ until the seventh decimal place.

To summarize, the uncertainty model trained on the protoplanetary disks produced excellent results for both the reconstruction of the protoplanetary disk as well as the estimated uncertainty. All four evaluation methods provide distributions that clearly peak around the optimal value, and the histogram in Figure 7.12 illustrates that the overwhelming majority of the simulated source pixels are within the 1 σ interval of the prediction. Furthermore, there are no indicators that the uncertainty is overestimated, as can be seen in Figure 7.9, Figure 7.10, and Figure 7.11.

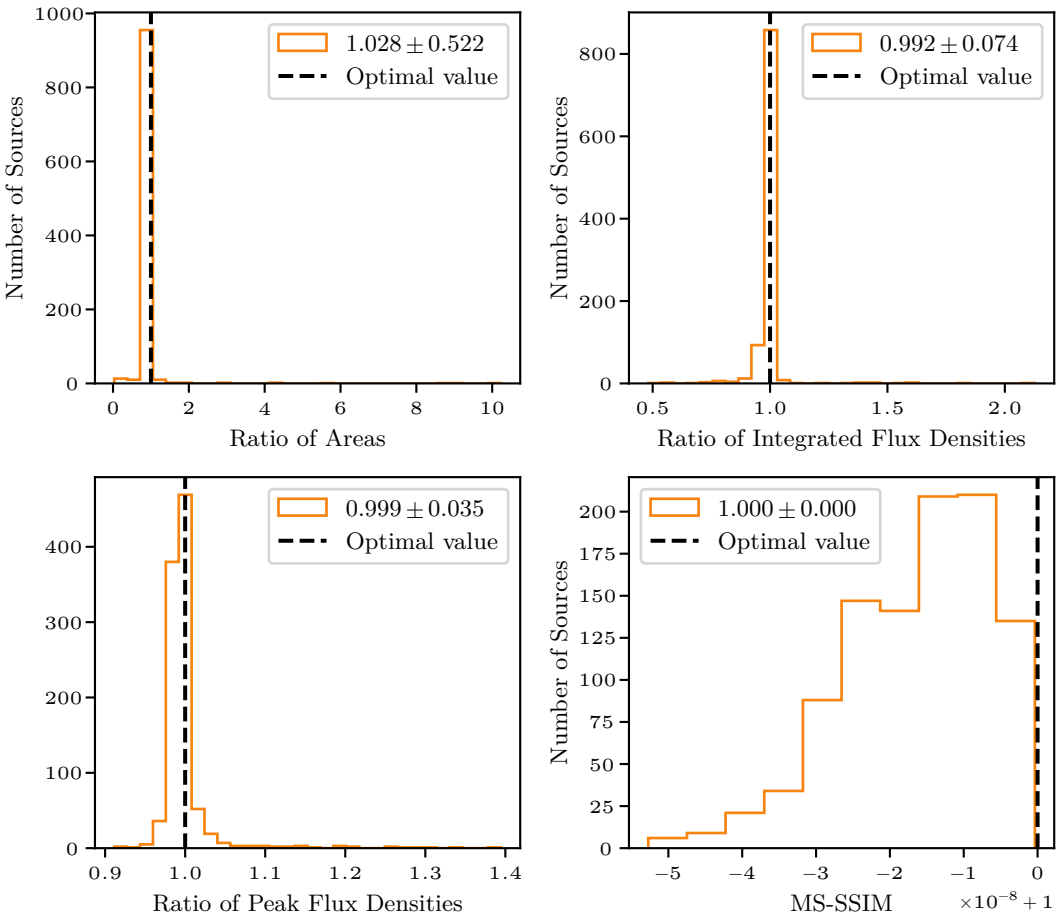

Figure 7.13: Histograms visualizing the results of the four evaluation methods introduced in Section 6.1. From top to bottom, left to right: Source area, integrated flux density, peak flux density, and MS-SSIM. For every histogram, the results' respective optimal value, mean, and standard deviation are shown. All four evaluation methods are in good agreement with their respective optimal values. Especially, the integrated flux densities clearly peak at one without significant outliers. The results for the MS-SSIM are very close to the optimal value of one and do not differ until the seventh decimal place.

7.3.3 Unknown Shapes

A frequently discussed question when dealing with neural networks is how the trained model handles unknown data, whether in flux density or shape. Since the network can only derive information about the parameter space from the data with which it was trained, the goal for the training data is always to provide a balanced and comprehensive set of examples to support the generalization of the network. In physics and astronomy, in particular, it is essential to prevent an over-restriction of algorithms used during the analysis to be able to identify new physics or cosmic source types. Providing an uncertainty estimate in addition to the reconstruction of a neural network is a helpful tool for such a task. Thus, in this subsection, the trained models introduced above are fed with unknown data, and the output is analyzed.

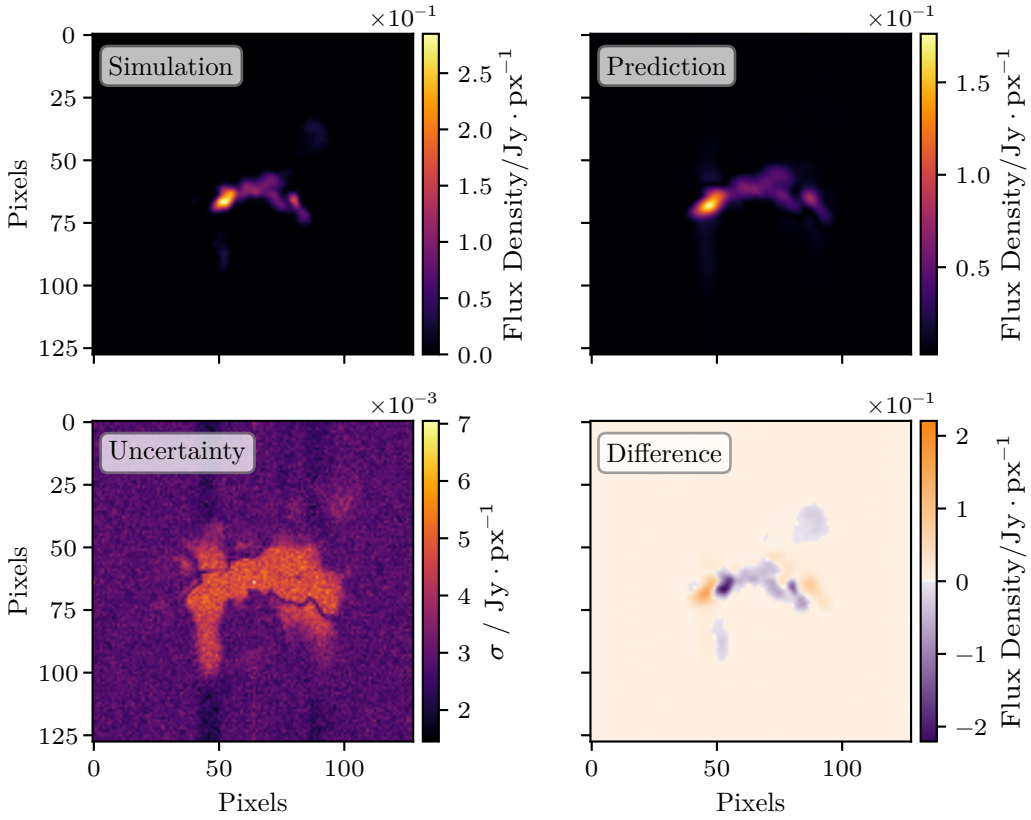

Figure 7.14: Output of the model trained on the protoplanetary disks for a GAN-generated source. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference.

First, the GAN-generated source illustrated in Figure 7.6 was fed to the model trained on protoplanetary disks, resulting in the output shown in Figure 7.14. The reconstruction is worse than the one produced by the model trained on those sources, which is expected since it is a different source type. However, the main source shape can still be recognized in the prediction, and the flux density is in the correct order of magnitude. This cannot be taken for granted, as even the image size is different. As described above, the protoplanetary disk simulations feature image sizes of 512 px × 512 px, while the GAN-generated sources have dimensions of 128 px × 128 px. Furthermore, as presented in Section 4.1 and Section 4.2, the FoV is also different for both simulations, explaining why there seems to be a zoom effect for the prediction in comparison to the simulation. Thus, the model is capable of adapting to new image sizes and different scales

on which information is missing in Fourier space. It should also be noted that both simulations were carried out with different array layouts, the VLA for the GAN-generated sources and ALMA for the protoplanetary disks. Therefore, only certain parts of the (u, v) plane are accessible for the respective simulation, as can be seen when comparing Figure 4.6 and Figure 4.11. Nevertheless, the protoplanetary-disk-model is able to reasonably fill in the gaps between the visibilities of the VLA-based simulation. Additionally, the uncertainty estimation matches the shape of the difference, except for the unreconstructed part at the top right of the primary source, which has a low flux density. The values for the uncertainty are a bit underestimated in comparison to the difference in flux density between the simulation and the prediction. To summarize, the model trained on the protoplanetary disks is able to handle unknown source shapes measured by a different observatory, even though the reconstruction performance suffers a bit and the uncertainty is underestimated. If this was new physics in the form of a previously unknown source type, physicists would be able to get a reasonable estimate of the source's shape paired with an indication from the uncertainty to perform a detailed analysis carried out manually.

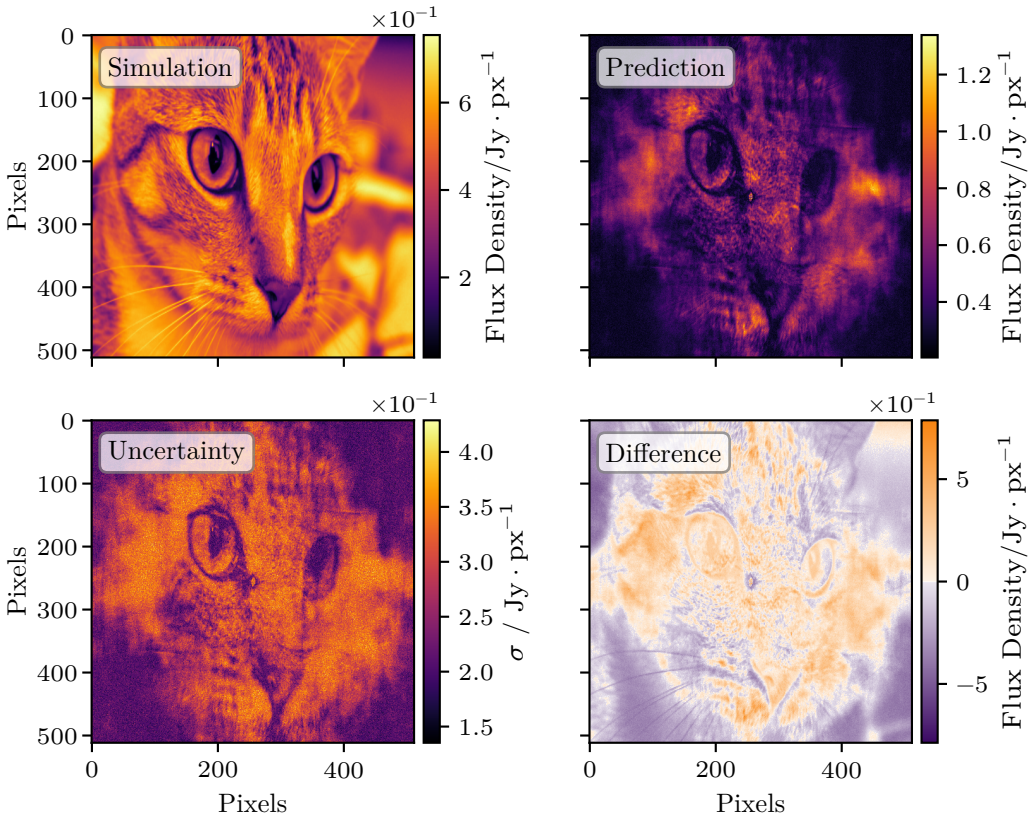

Figure 7.15: Reconstruction and uncertainty estimation of an image of a cat. The model was trained on the protoplanetary disks. From top to bottom, from left to right: Simulated image, prediction, uncertainty, and difference between simulation and difference. Image data taken from `scikit-image` [114], picture taken by Stefan van der Walt.

This statement can be confirmed by analyzing the reconstruction shown in Figure 7.15. In order to test the network's robustness, an image of a cat was fed to the model trained on protoplanetary disks. The image was resized beforehand to a size of 512 px × 512 px to match the image sizes the model was trained on. After Fourier-transforming the image, the real and

imaginary parts were fed to the network. Afterward, the output was sampled as described in Section 7.2. As shown in Figure 7.15, the central parts of the cat can be reconstructed, e.g., the eyes or the snout. Furthermore, a tiny protoplanetary disk is reconstructed at the center of the image, which is expected since the model only knows protoplanetary disks. When looking at the estimated uncertainty, it becomes evident that the sections of the image featuring the highest uncertainty are the disk-like structure in the center and the noise-like structures enclosing the center. Furthermore, the parts correctly adopted from the input, e.g., the eyes and parts of the snout, also show the lowest uncertainty. This indicates that the model is more confident with the cat's eyes and snout than with the protoplanetary disk, precisely the behavior one would expect from a model capable of estimating the prediction's uncertainty. Treating the cat image as a new source type, it becomes clear that the model is still trying to reconstruct the shape it was trained on but is able to indicate through the estimated uncertainty that the analyzer should take a closer look at the reconstruction and perform a manual analysis if necessary.

In summary, the adaptions to the architecture and the training routine introduced in Section 7.1 enable the model to estimate the uncertainty corresponding to the prediction. Thanks to the sampling approach described in Section 7.2, the uncertainty can also be translated to the image space. The results presented in Section 7.3 confirm that the approach is working while only slightly decreasing the reconstruction performance. When feeding unknown source shapes to pre-trained models, the reconstruction capability differs depending on the similarities between the new source shape and the one the model was trained on. In any case, the uncertainty estimation serves as a reliable indicator for potential unknown source types calling for an in-depth, manual analysis.

8 Reconstruction of Interferometric Measurements

As indicated in previous chapters, the ultimate goal is to apply the neural network approach to measured interferometric data in order to clean and reconstruct it. To achieve this, improvements to the training routine and architectures were first deployed and evaluated on the smaller GAN-generated images, see Chapter 5 and Chapter 6. The knowledge gained from these tests is incorporated here, such as parameters and alterations for the training routine (Chapter 5), modifications to the training data in order to minimize data-simulation mismatches (Chapter 4), and the ability to provide uncertainty estimations (Chapter 7). For the measured data, the DSHARP data set was chosen, as described in Section 4.2. The reasoning behind this is explained in Section 8.1, as well as the pre-processing steps performed before feeding the data to the network. In Section 8.2, the predictions in Fourier space are shown and analyzed, while in Section 8.3, the reconstructions in image space are visualized.

8.1 Pre-processing of the DSHARP Data

An overview of the cleaned protoplanetary disks observed for DSHARP was shown in Figure 4.8. There are multiple reasons why the DSHARP observations were chosen to be reconstructed by the neural network approach presented in this thesis:

- The number of sources measured by ALMA (20) is relatively small. This speeds up computation time and enables a more in-depth analysis for single sources, which would be more complicated for a larger data set.
- The source shape varies but in a limited range. Great deviations in shape complicate the simulation process because a large number of features have to be accounted for.
- There already is a simulation foundation, as mentioned in Section 4.2. The results of the bachelor thesis by Andreas Maisinger [70] proved to be a good starting point to refine and improve the existing simulations of protoplanetary disks inspired by DSHARP.
- The DSHARP data is publicly available[1] in multiple data formats such as FITS and measurement sets (ms), which is the input data format for WSCLEAN. Thus, the DSHARP data can also be cleaned by an established approach like WSCLEAN and then be compared to the results of the neural network.

Before the DSHARP data can be fed to the neural network model, some pre-processing steps have to be taken. First, the relevant data is extracted from the ms files and split into real and imaginary parts. At this point, the data can be gridded by the gridder implemented in `pyvisgen`. This ensures that the gridding process is comparable to the one used for the DSHARP simulations introduced in Section 4.2. Furthermore, the relevant parameters for the gridding were chosen to be the same for both simulations and observed data: An image size of 512 px × 512 px and a FoV of 9″. This is consistent with the original FoV chosen by researchers [4]. Additionally, some data points may have been flagged by the observers, meaning that problems occurred during

[1] https://almascience.eso.org/almadata/lp/DSHARP/

the observation and that these particular data points should not be considered for the analysis. This is also taken into consideration before gridding the visibilities. After the gridding process is completed, the DSHARP data is ready for reconstruction by the neural network model.

8.2 Reconstruction in Fourier Space

The uncertainty model trained on the DSHARP simulations was chosen to reconstruct the DSHARP observations. Exemplary reconstructions of said model on simulated data are shown in Section 7.3, e.g., Figure 7.9 and Figure 7.10. The model was selected not only because it was trained on the DSHARP simulation but also because it is able to provide an additional uncertainty estimation alongside its reconstruction. The concept behind the uncertainty approach was presented in Section 7.1.

In Figure 8.1, the real part of the neural network reconstruction of the protoplanetary disk "Elias 24" is visualized. On the left side, the input image is shown, the prediction by the neural network is displayed in the center, and the estimated uncertainty is plotted on the right side. Because large parts of the image are filled with near-zero values, a section of 90 px × 90 px in the central part of the image is shown. When comparing the input image and the prediction, it is noticeable that the missing pixel values in the central part of the section are filled in by the model, which is in line with the expected behavior. However, the overall flux density is underestimated by a factor of ≈ 100. For the estimated uncertainty, the parts that were filled in by the neural network feature a higher uncertainty than the parts that remain essentially unchanged, which matches the behavior seen in Section 7.3. The uncertainty values are also small in comparison to the prediction.

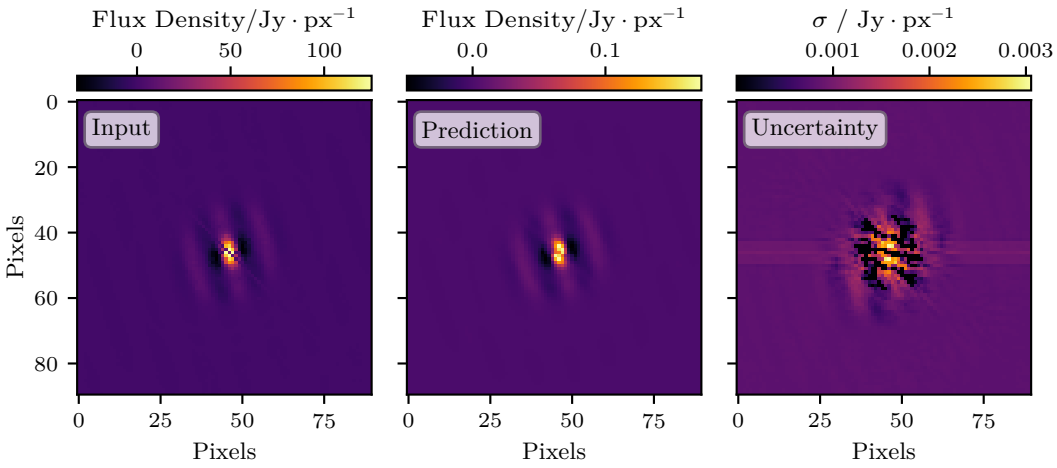

Figure 8.1: Output of the neural network model for the real gridded visibilities of the protoplanetary disk "Elias 24". From left to right: Input to the neural network, prediction and estimated uncertainty. For a better visualization, a section of 90 px × 90 px in the central part of the image is shown.

The corresponding imaginary part of the prediction is shown in Figure 8.2. Again, the input image is shown on the left, the prediction in the middle, and the estimated uncertainty on the right side. For the same reason as before, a section of 90 px × 90 px in the central part of the image is displayed. Similar to the real part, the pixels with a flux density of zero are reconstructed by the neural network model. Furthermore, the shape of the estimated uncertainty

matches the shape of the reconstructed pixels. However, similar to the real part, the predicted flux density is ≈ 100 times smaller than the input image, and the values for the uncertainty are three orders of magnitude smaller than the values for the prediction.

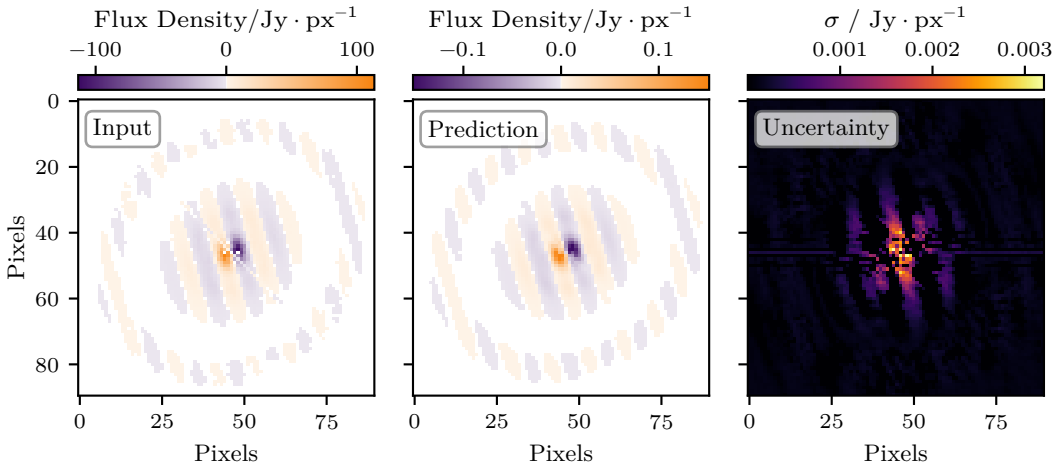

Figure 8.2: Output of the neural network model for the imaginary gridded visibilities of the protoplanetary disk "Elias 24". From left to right: Input to the neural network, prediction and estimated uncertainty. For a better visualization, a section of 90 px × 90 px in the central part of the image is shown.

8.3 Reconstruction in Image Space

Utilizing the sampling approach presented in Section 7.2, which translates the uncertainty from Fourier to image space, it is possible to obtain a reconstruction of the protoplanetary disk "Elias 24" in image space alongside an estimation of the uncertainty. For this, the real and the imaginary predictions shown in Figure 8.1 and Figure 8.2 are used as input for the sampling process. One hundred versions of the predicted protoplanetary disk in Fourier space are drawn from the Gaussian distribution defined by the prediction and the uncertainty. Next, the images are translated to the image space via a Fourier transformation, and the mean and the standard deviation are calculated for all pixels in the image, resulting in the reconstruction in image space displayed in Figure 8.3.

In addition to the output of the neural network, Figure 8.3 also features a reconstruction performed by WSCLEAN for comparison. The cleaning was obtained using the parameter settings listed in Table 8.1. Because WSCLEAN returns cleaned images with the unit Jansky per beam, the beam has to be corrected for in order to conduct a meaningful comparison between the results of the neural network and WSCLEAN. Thus, the beam area is calculated using

$$A_{\text{beam}} = \frac{2\pi\, b_{\text{min}} \cdot b_{\text{maj}}}{8\ln(2)}\,. \tag{8.1}$$

Here, b_{min} is the Full Width at Half Maximum (FWHM) of the minor axis of the beam and b_{maj} is the FWHM of the major axis [106, 39]. Note that this alters the flux densities, but the beam-smearing effect, e.g., visible through the expansion of source structures, remains unchanged. After that, the cleaned image is divided by the beam area value and multiplied by the squared pixel size to obtain the unit Jansky per pixel.

Table 8.1: Overview of the parameter settings of WSCLEAN used to clean the protoplanetary disk "Elias 24"

Parameter	Setting
size	512 px
scale	17.58 asec
mgain	0.8
gain	0.1
niter	500 000
auto-mask	1
auto-threshold	0.3
padding	1.3
weight	briggs

The resulting beam-corrected image is displayed on the top left of Figure 8.3. The sampled network-based prediction and the sampled standard deviation of the protoplanetary disk are shown on the top right and the bottom left, respectively. On the bottom right, the difference between the cleaned image from WSCLEAN and the neural network-based prediction is visualized. It is noticeable that the location of the core and the extent of the rings of the protoplanetary disk are very similar for both cleaning approaches, which is confirmed by investigating the difference plot. In contrast to the WSCLEAN cleaning, the prediction of the neural network features a decreased decline in flux density when moving from the core to the outer rings, resulting in increased visibility of said rings. The overall flux density is approximately one order of magnitude higher for the WSCLEAN cleaning. The estimated uncertainty also shows the general shape of the protoplanetary disk. However, sidelobes and artifacts are displayed that seem not directly related to the source. The overall uncertainty values are an order of magnitude smaller than in the prediction, indicating that the network is rather certain about the predicted flux densities.

In summary, the neural network-based approach is able to reconstruct a protoplanetary disk measured for DSHARP. The structure of the disk is clearly visible, and no significant artifacts or background noise is visible in the reconstruction. The location of the core and the rings is similar in comparison to an image cleaned by WSCLEAN. The estimated standard deviation is small compared to the prediction and features the general shape of the protoplanetary disk. Improvements can be made in terms of sidelobes and artifacts in the uncertainty estimation and the overall flux density, which could be underestimated.

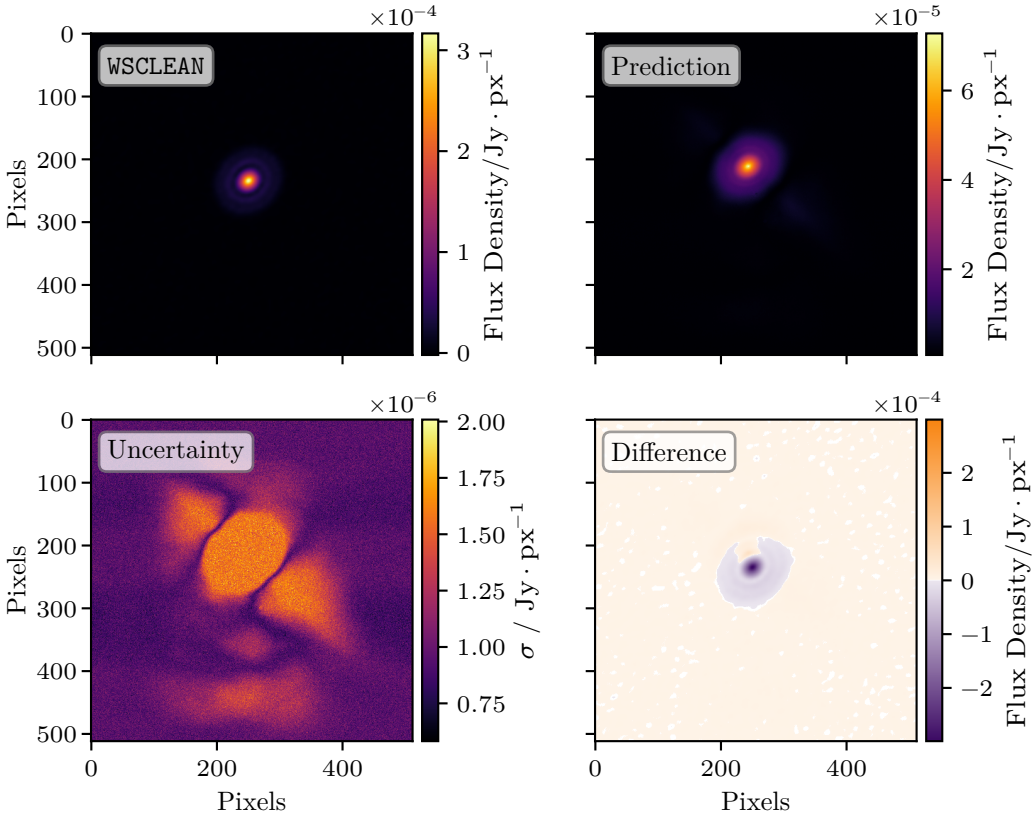

Figure 8.3: Reconstructed image of the protoplanetary disk "Elias 24". From top to bottom, from left to right: Output of WSCLEAN, sampled prediction, estimated uncertainty, and the difference between the first two images. Due to differences in the reconstruction processes for WSCLEAN and the neural network, the sampled image and the uncertainty have to be rotated by 90°. The cleaned image from WSCLEAN is shown for comparison and does not represent the "true" source in image space.

9 Summary and Outlook

Automatization and reproducibility are two main keywords associated with the analysis of complex and extensive data. In this thesis, I presented a holistic analysis pipeline ranging from radio galaxy simulations via the simulation of radio interferometer measurements to the training and evaluation of deep learning-based models trained on this simulated data. This analysis pipeline is based on two frameworks, `radionets` [99] for the training and evaluation of the neural networks and `pyvisgen` [98] for the radio interferometer simulations. Both of these frameworks were improved and expanded by me, e.g., for the improved estimation of uncertainty or a new normalization technique, which ultimately enabled the reconstruction of the protoplanetary disk "Elias 24" coupled with an uncertainty estimation. Utilizing the capabilities of these two frameworks enables astronomers to automatize their analyses in a reproducible manner, which will be necessary given the planned interferometers that aim for high data rates, like the SKA [27] and ngVLA [75].

A decisive prerequisite for the successful training of neural networks is the training data. It is essential that all data characteristics are included so that the network is able to reconstruct this type of data if it no longer contains specific information. Thus, a training data set containing different kinds of protoplanetary disks is necessary for this thesis since the ultimate goal is reconstructing a protoplanetary disk from DSHARP. However, the step from simple two-dimensional Gaussian blobs (see [100]) on a grid to reconstructing an observed protoplanetary disk is quite significant. So, it was clear that a series of tests and subsequent adjustments to the simulations and the training process are necessary in order to make this possible. Furthermore, a more physics-based simulation foundation was desired. Thus, a collaboration with several colleagues from Hamburg was formed, and they provided wGAN-based simulations of FRI and FRII galaxies, which were modeled after the FIRST survey conducted by the VLA [65, 91]. In addition to a physically motivated simulation foundation, the simulated images also had an increased image size of 128 px × 128 px, thus quadrupling the previous image size. This was important due to the fact that interferometric observations commonly have image sizes up to thousands of pixels per dimension, so a steady increase in image size was also a crucial requirement for the goal of reconstructing such an observation. After obtaining physics-based radio galaxy simulations, the RIME was utilized to simulate an observation by a radio interferometer. For this purpose, the `pyvisgen` framework with its Graphics Processing Unit (GPU)-accelerated implementation of the RIME was improved and expanded and then applied to the wGAN-generated data. In order to comprehensively mimic an observation by the VLA for the FIRST survey, the adjustable parameters of `pyvisgen` were set to values similar to the conditions of the actual FIRST measurements.

After obtaining RIME-simulated gridded visibilities of FIRST observations to create a training, a validation, and a test data set, different experiments and adjustments to the training process were conducted. In particular, the influence of images containing only white noise and the influence of an improved normalization technique were investigated. For this purpose, three models were trained: the base model, the normalized model, which was trained using an image-based normalization technique, and the noise model, which was additionally trained on training data partly containing white noise images. It was found that the neural network model was

able to identify and adequately reconstruct pure white noise images as well as images containing sources. That means that the model is not reconstructing a FRI or FRII source by any means necessary but instead is able to handle data sets containing noise images, which can occur during sky surveys. In order to also get quantitative measures to compare the models, four different evaluation methods were introduced, which examined the source area, the peak flux density, the integrated flux density, and the MS-SSIM. All three models produced results close to the respective optimal values for each evaluation technique, with the noise and the normalized model performing better than the base model for all categories. Especially for sources with low SNR, which are generally harder to reconstruct, the noise and the normalized model score lower standard deviations and mean values closer to one than the base model. Additionally, the reconstruction capability over the course of the training was examined. Again, the normalized model proved to be superior, providing a more stable training process with few outliers as well as being time and disk-usage efficient by converging after fewer epochs than the base model. The noise model scored similar results as the normalized model, underlining the robustness of the approach to pure white noise images. Therefore, it was proven that not only the network-based approach was able to handle RIME-simulated interferometer observations of wGAN-generated FRI and FRII galaxies, but that it also remains undisturbed by pure white noise images in the training data. Furthermore, these results could be improved upon by utilizing an image-based normalization technique, which was applied going forward. Especially the improved convergence time and the robustness against noise images enable the approach to be potentially utilized to accelerate analysis pipelines of radio observatories such as ALMA.

After analyzing the results of these experiments and subsequently adjusting the training routine, it was necessary to change the simulated radio galaxies from FRI and FRII to protoplanetary disks. For this purpose, the simulated protoplanetary disks from Andreas Maisinger [70] were taken as a foundation and expanded to serve the goal of reconstructing an observed disk from DSHARP. The simulations were based on characteristics of the DSHARP sources in terms of, e.g., the intensity decrease and the number of rings. Additionally, the image size was again increased, this time to $512\,\mathrm{px} \times 512\,\mathrm{px}$ and therefore reaching the range of image sizes produced by interferometer observations. Similar to the wGAN-based simulations before, the protoplanetary disks were used as input to the interferometer simulations by `pyvisgen`. The parameters were adapted to the characteristics of the DSHARP measurements to ensure compatibility.

The gridded visibilities of both the FRI and FRII galaxies and the simulated protoplanetary disks were then used to prepare the model for providing an uncertainty estimation in addition to the reconstruction. The architecture was enhanced, and the loss function was adapted so that the model was able to predict the mean μ and the variance σ^2 of a Gaussian distribution. Using a customized ReLU function ensured that the predicted variance had positive values. The mean values and standard deviations generated this way were then translated to image space using a sampling approach. Due to the image-based normalization, error propagation was required, which was also implemented. After applying the evaluation methods introduced before to the uncertainty models, the results showed that the models were able to give uncertainty estimations matching their reconstructions in shape and flux density both in Fourier space and in image space while only slightly decreasing the reconstruction quality. Furthermore, it was investigated how an uncertainty model handles unknown data. For this purpose, the uncertainty model trained on protoplanetary disks was fed a wGAN-generated source and an image of a cat. The model was able to generally reconstruct the GAN-generated source and give an appropriate uncertainty estimation, proving the model's capability to generalize. Additionally, the cat image could be partially reconstructed, and the protoplanetary disk, which was falsely reconstructed, was endowed with high uncertainty. This aligns perfectly with the expected behavior of a well-trained

uncertainty model subjected to unknown data, thus enabling researchers to provide statistically motivated uncertainty estimations alongside their reconstructions.

Finally, the protoplanetary disk "Elias 24" observed for DSHARP was fed to the uncertainty model trained on protoplanetary sources. The model was able to reconstruct the source, and comparisons to the cleaning result of WSCLEAN showed that the location of the core and the extent of the surrounding rings match well. To summarize, this thesis presented a comprehensive, deep learning-based analysis pipeline, which starts with simulating different kinds of radio galaxies. It ranges from analytically describing the measurement process of radio interferometers to evaluating neural networks trained on said simulated data and finishes with the reconstruction of the protoplanetary disk "Elias 24" using data measured for DSHARP. Using this pipeline, any researcher can create simulations for arbitrary radio galaxies and arbitrary radio observatories and use them as a training dataset for neural network models. Then, the trained models reconstruct and clean the input images and produce an appropriate uncertainty estimate. This way, radio observations can be processed and prepared for physical analysis, potentially even by astronomers from fields other than radio interferometry.

More improvements to the pipeline are necessary to prepare the approach for large-scale applications in the future. While the reconstruction of the protoplanetary disk "Elias 24" worked on a level comparable to WSCLEAN, other sources from DSHARP result in worse reconstructions. Here, potential data-simulation mismatches need to be further investigated. Also, the image size needs to be increased even more. With image sizes of $512\,\mathrm{px} \times 512\,\mathrm{px}$, the pixel size is significantly increased when maintaining the original FoV, leading to a coarse grid and a decrease in resolution. Preferably, image size up to $3000\,\mathrm{px} \times 3000\,\mathrm{px}$ should be able to be processed by the pipeline. Currently, this is not feasible due to limitations in the area of computational resources, especially when considering high data rates. Thus, software optimizations should be considered to reduce the need for disk space and computation time. Another possible approach is to reduce the dimensionality during training by utilizing the sparseness of the input data. Furthermore, the implementation of the RIME and the network need to be prepared for the reconstruction of large and sparse images featuring multiple point sources, which are commonly observed during sky surveys. For the RIME, additional features have to be considered, e.g., the influence of cosmic sources outside of the FoV and the impact of polarisation effects, which is currently examined. On the network side, several approaches are currently being investigated, like successively extracting single point sources or the application of transformer networks [30].

Bibliography

1. B. P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger*”. *The Astrophysical Journal Letters* 848:2, 2017, page L12. ISSN: 2041-8213.
   DOI: 10.3847/2041-8213/aa91c9
2. C. C. Aggarwal. “An Introduction to Data Mining”. *Data Mining: The Textbook*. Springer International Publishing, Cham, 2015, pages 1–26.
   DOI: 10.1007/978-3-319-14142-8_1
3. ALMA Partnership et al. *ALMA Cycle 4 Technical Handbook*. 2016. ISBN: 978-3-923524-66-2.
   URL: https://arc.iram.fr/documents/cycle4/ALMA_Cycle4_Technical_Handbook-Final.pdf
4. S. M. Andrews et al. “The Disk Substructures at High Angular Resolution Project (DSHARP). I. Motivation, Sample, Calibration, and Overview”. *The Astrophysical Journal Letters* 869:2, 2018, page L41.
   DOI: 10.3847/2041-8213/aaf741
5. M. Arjovsky, S. Chintala, and L. Bottou. *Wasserstein GAN*. 2017.
   DOI: 10.48550/arXiv.1701.07875. eprint: 1701.07875
6. P. J. Armitage and W. Kley. *From Protoplanetary Disks to Planet Formation*. Springer Berlin, Heidelberg, 2019.
   DOI: 10.1007/978-3-662-58687-7
7. R. H. Becker, R. L. White, and D. J. Helfand. “The FIRST Survey: Faint Images of the Radio Sky at Twenty Centimeters”. *Astrophysical Journal* 450, 1995, page 559.
   DOI: 10.1086/176166
8. V. Beckmann and C. R. Shrader. *Active Galactic Nuclei*. 2012.
   URL: https://ui.adsabs.harvard.edu/abs/2012agn..book.....B
9. M. Bertalmio et al. “Image inpainting”. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’00. ACM Press/Addison-Wesley Publishing Co., USA, 2000, pages 417–424.
   DOI: 10.1145/344779.344972
10. A. Bienenstock and P. P. Ewald. “Symmetry of Fourier space”. *Acta Crystallographica* 15:12, 1962, pages 1253–1261.
    DOI: 10.1107/S0365110X6200331X
11. 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
12. R. D. Blandford and A. Königl. “Relativistic jets as compact radio sources.” *Astrophysical Journal* 232, 1979, pages 34–48.
    DOI: 10.1086/157262

13. A. C. Boley. “THE TWO MODES OF GAS GIANT PLANET FORMATION”. *The Astrophysical Journal* 695:1, 2009, page L53.
DOI: 10.1088/0004-637X/695/1/L53

14. M. Born and E. Wolf. “Fraunhofer diffraction at apertures of various forms”. *Principles of Optics: 60th Anniversary Edition*. 7th ed. Cambridge University Press, 2019. Chap. 8.5, pages 436–446.
DOI: 10.1017/9781108769914

15. A. P. Boss. “Giant planet formation by gravitational instability.” *Science* 276, 1997, pages 1836–1839.
DOI: 10.1126/science.276.5320.1836

16. L. Bottou, F. E. Curtis, and J. Nocedal. *Optimization Methods for Large-Scale Machine Learning*. 2018.
DOI: 10.48550/arXiv.1606.04838

17. L. Breiman. “Random Forests”. *Machine Learning* 45:1, 2001, pages 5–32. ISSN: 1573-0565.
DOI: 10.1023/A:1010933404324

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

19. 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

20. 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

21. K. Cabello-Solorzano et al. “The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis”. *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Ed. by P. García Bringas et al. Springer Nature Switzerland, Cham, 2023, pages 344–353.
DOI: 10.1007/978-3-031-42536-3_33

22. J. E. Chambers. “Planetary accretion in the inner Solar System”. *Earth and Planetary Science Letters* 223:3, 2004, pages 241–252. ISSN: 0012-821X.
DOI: 10.1016/j.epsl.2004.04.031. URL: https://www.sciencedirect.com/science/article/pii/S0012821X04002791

23. K. Christensen-Jeffries et al. “Super-resolution Ultrasound Imaging”. *Ultrasound in Medicine & Biology* 46:4, 2020, pages 865–891. ISSN: 0301-5629.
DOI: 10.1016/j.ultrasmedbio.2019.11.013. URL: https://www.sciencedirect.com/science/article/pii/S0301562919315959

24. Clem & Adri Bacri-Normier (wingsforscience.com)/ESO. *ALMA from the air*.
URL: https://www.eso.org/public/images/eso1312a/ visited on 2024-07-01

25. T. C. Collaboration et al. “The CHIME Fast Radio Burst Project: System Overview”. *The Astrophysical Journal* 863:1, 2018, page 48.
DOI: 10.3847/1538-4357/aad188

26. P. C. Crane and P. J. Napier. "Sensitivity". *Synthesis Imaging in Radio Astronomy*. Ed. by R. A. Perley, F. R. Schwab, and A. H. Bridle. Vol. 6. Astronomical Society of the Pacific Conference Series. 1989, page 139

27. 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

28. K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer London, London, 2019.
DOI: 10.1007/978-1-4471-7452-3_1

29. R. Durisen et al. *Gravitational Instabilities in Gaseous Protoplanetary Disks and Implications for Giant Planet Formation*. 2006. eprint: astro-ph/0603179.
URL: https://arxiv.org/abs/astro-ph/0603179

30. V. P. Dwivedi and X. Bresson. *A Generalization of Transformer Networks to Graphs*. 2021.
DOI: 10.48550/arXiv.2012.09699. eprint: 2012.09699

31. V. Edupuganti et al. "Uncertainty Quantification in Deep MRI Reconstruction". *IEEE Transactions on Medical Imaging* 40:1, 2021, pages 239–250.
DOI: 10.1109/TMI.2020.3025065

32. 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

33. O. Elharrouss et al. "Image Inpainting: A Review". *Neural Processing Letters* 51:2, 2020, pages 2007–2028. ISSN: 1573-773X.
DOI: 10.1007/s11063-019-10163-0

34. ESO/C. Malin (christophmalin.com). *The Atacama Large Millimeter/submillimeter Array (ALMA) by night, under the Magellanic Clouds*.
URL: https://www.eso.org/public/images/ann13016a/ visited on 2024-07-01

35. S. Faghani et al. "Quantifying Uncertainty in Deep Learning of Radiologic Images". *Radiology* 308:2, 2023, e222217.
DOI: 10.1148/radiol.222217

36. B. L. Fanaroff and J. M. Riley. "The morphology of extragalactic radio sources of high and low luminosity". *MNRAS* 167, 1974, 31P–36P.
DOI: 10.1093/mnras/167.1.31P

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

38. W. Fong et al. "RADIO CONSTRAINTS ON LONG-LIVED MAGNETAR REMNANTS IN SHORT GAMMA-RAY BURSTS". *The Astrophysical Journal* 831:2, 2016, page 141.
DOI: 10.3847/0004-637X/831/2/141

39. Geyer, F. et al. "Deep-learning-based radiointerferometric imaging with GAN-aided training". *A&A* 677, 2023, A167.
DOI: 10.1051/0004-6361/202347073

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

41. I. J. Goodfellow et al. *Generative Adversarial Networks*. 2014.
DOI: 10.48550/arXiv.1406.2661. eprint: 1406.2661

42. J. G. Greener et al. "A guide to machine learning for biologists". *Nature Reviews Molecular Cell Biology* 23:1, 2022, pages 40–55. ISSN: 1471-0080.
DOI: 10.1038/s41580-021-00407-0

43. E. W. Greisen. *AIPS Memos*. 2022.
URL: http://www.aips.nrao.edu/aipsmemo.html visited on 2024-07-09

44. S. Gross and M. Wilber. *Training and investigating Residual Nets*. 2016.
URL: http://torch.ch/blog/2016/02/04/resnets.html visited on 2024-04-23

45. Hamaker, J. P. and Bregman, J. D. "Understanding radio polarimetry. III. Interpreting the IAU/IEEE definitions of the Stokes parameters". *Astron. Astrophys. Suppl. Ser.* 117:1, 1996, pages 161–165.
DOI: 10.1051/aas:1996147

46. Hamaker, J. P., Bregman, J. D., and Sault, R. J. "Understanding radio polarimetry. I. Mathematical foundations". *Astron. Astrophys. Suppl. Ser.* 117:1, 1996, pages 137–147.
DOI: 10.1051/aas:1996146

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

48. C. Hayashi. "Evolution of Protostars". *Annual Review of Astronomy and Astrophysics* Volume 4, 1966, pages 171–192.
DOI: 10.1146/annurev.aa.04.090166.001131

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

50. K. He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pages 1026–1034.
DOI: 10.1109/ICCV.2015.123

51. K. He et al. "Identity Mappings in Deep Residual Networks". *Computer Vision – ECCV 2016*. Ed. by B. Leibe et al. Springer International Publishing, Cham, 2016, pages 630–645.
DOI: 10.1007/978-3-319-46493-0_38

52. 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

53. J. Howard and S. Gugger. "Fastai: A Layered API for Deep Learning". *Information* 11:2, 2020. ISSN: 2078-2489.
DOI: 10.3390/info11020108. URL: https://www.mdpi.com/2078-2489/11/2/108

54. J. D. Hunter. "Matplotlib: A 2D graphics environment". *Computing in Science & Engineering* 9:3, 2007, pages 90–95.
DOI: 10.1109/MCSE.2007.55

55. S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015.
DOI: 10.48550/arXiv.1502.03167. arXiv: 1502.03167 [cs.LG]

56. A. Iqbal et al. “Active Galactic Nucleus Feedback with the Square Kilometre Array and Implications for Cluster Physics and Cosmology”. *Journal of Astrophysics and Astronomy* 38:4, 68, 2017, page 68.
DOI: 10.1007/s12036-017-9491-4. arXiv: 1705.04444 [astro-ph.CO]

57. R. A. Jacobs. “Increased rates of convergence through learning rate adaptation”. *Neural Networks* 1:4, 1988, pages 295–307. ISSN: 0893-6080.
DOI: 10.1016/0893-6080(88)90003-2

58. R. C. Jones. “A New Calculus for the Treatment of Optical Systems. I. Description and Discussion of the Calculus”. *J. Opt. Soc. Am.* 31:7, 1941, pages 488–493.
DOI: 10.1364/JOSA.31.000488. URL: https://opg.optica.org/abstract.cfm?URI=josa-31-7-488

59. A. B. Kamruddin and J. Dexter. “A geometric crescent model for black hole images”. *Monthly Notices of the Royal Astronomical Society* 434:1, 2013, pages 765–771. ISSN: 0035-8711.
DOI: 10.1093/mnras/stt1068

60. G. Karagiorgi et al. “Machine learning in the search for new fundamental physics”. *Nature Reviews Physics* 4:6, 2022, pages 399–412. ISSN: 2522-5820.
DOI: 10.1038/s42254-022-00455-1

61. N. Kawai, T. Sato, and N. Yokoya. “Diminished Reality Based on Image Inpainting Considering Background Geometry”. *IEEE Transactions on Visualization and Computer Graphics* 22:3, 2016, pages 1236–1247.
DOI: 10.1109/TVCG.2015.2462368

62. D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017.
DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs.LG]

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

64. A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.
URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

65. J. Kummer et al. “Radio Galaxy Classification with wGAN-Supported Augmentation”. *INFORMATIK 2022, Lecture Notes in Informatics (LNI) - Proceedings*. Vol. P-326. Gesellschaft für Informatik, Bonn, 2022, pages 469–478.
DOI: 10.18420/inf2022_38

66. 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.
DOI: 10.48550/arXiv.1609.04802

67. A. Likas, N. Vlassis, and J. J. Verbeek. “The global k-means clustering algorithm”. *Pattern Recognition* 36:2, 2003, pages 451–461. ISSN: 0031-3203.
DOI: 10.1016/S0031-3203(02)00060-2. URL: https://www.sciencedirect.com/science/article/pii/S0031320302000602

68. 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

69. 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

70. A. Maisinger. “Simulation protoplanetarischer Scheiben im Kontext maschineller Lernverfahren zur Analyse von ALMA Beobachtungen”. Bachelors Thesis. TU Dortmund University, 2023

71. W. H. McMaster. “Polarization and the Stokes Parameters”. *American Journal of Physics* 22:6, 1954, pages 351–362. ISSN: 0002-9505.
DOI: 10.1119/1.1933744

72. J. Mena-Parra et al. “A Clock Stabilization System for CHIME/FRB Outriggers”. *The Astronomical Journal* 163:2, 2022, page 48.
DOI: 10.3847/1538-3881/ac397a

73. M. Meuwly. “Machine Learning for Chemical Reactions”. *Chemical Reviews* 121:16, 2021, pages 10218–10239.
DOI: 10.1021/acs.chemrev.1c00033

74. 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

75. E. J. Murphy et al. *Science with an ngVLA: The ngVLA Science Case and Associated Science Requirements*. 2018.
DOI: 10.48550/arXiv.1810.07524. eprint: 1810.07524. URL: https://arxiv.org/abs/1810.07524

76. 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

77. 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

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

79. NRAO. *300-foot telescope*.
URL: https://public.nrao.edu/telescopes/300-foot-telescope/ visited on 2024-07-05

80. NRAO. *Configuration Plans and Proposal Deadlines*.
URL: https://science.nrao.edu/facilities/vla/proposing/configpropdeadlines visited on 2024-07-04

81. NRAO. *Sensitivity*.
URL: https://science.nrao.edu/facilities/vla/docs/manuals/oss/performance/sensitivity visited on 2024-07-04

82. NRAO/AUI/NSF. *The Wye of the VLA*.
URL: https://public.nrao.edu/gallery/this-is-the-vla-wye/ visited on 2024-07-04

83. 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

84. A. R. Offringa, B. McKinley, Hurley-Walker, et al. "WSClean: an implementation of a fast, generic wide-field imager for radio astronomy". *MNRAS* 444:1, 2014, pages 606–619.
DOI: 10.1093/mnras/stu1368

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

86. T. J. Pearson and A. C. S. Readhead. "Image Formation by Self-Calibration in Radio Astronomy". *Annual review of astronomy and astrophysics* 22, 1984, pages 97–130.
DOI: 10.1146/annurev.aa.22.090184.000525

87. A. Quirrenbach, R. van Boekel, and C. Dullemond. *Beobachtende Astronomie: Diffraction theory*. Zentrum für Astronomie, Universität Heidelberg.
URL: https://www.ita.uni-heidelberg.de/~%20dullemond/lectures/obsastro_2011/chap_diffract.pdf visited on 2024-07-10

88. 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

89. 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

90. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". *Nature* 323:6088, 1986, pages 533–536. ISSN: 1476-4687.
DOI: 10.1038/323533a0

91. L. Rustige et al. "Morphological classification of radio galaxies with Wasserstein generative adversarial network-supported augmentation". *RAS Techniques and Instruments* 2:1, 2023, pages 264–277.
DOI: 10.1093/rasti/rzad016

92. T. Ružić and A. Pižurica. "Context-Aware Patch-Based Image Inpainting Using Markov Random Field Modeling". *IEEE Transactions on Image Processing* 24:1, 2015, pages 444–456.
DOI: 10.1109/TIP.2014.2372479

93. M. Ryle and D. D. Vonberg. "Solar Radiation on 175 Mc./s". *Nature* 158:4010, 1946, pages 339–340. ISSN: 1476-4687.
DOI: 10.1038/158339b0

94. T. Salimans et al. *Improved Techniques for Training GANs*. 2016.
DOI: 10.48550/arXiv.1606.03498. eprint: 1606.03498

95. B. Saxton and NRAO/AUI/NSF. *NRAO's Baseline Episode 3: Viewing Active Galaxies*.
URL: https://public.nrao.edu/news/key-feature-powerful-radio-galaxies/ visited on 2024-07-04

96. J. Schmidt et al. “Recent advances and applications of machine learning in solid-state materials science”. *npj Computational Materials* 5:1, 2019, page 83. ISSN: 2057-3960.
DOI: 10.1038/s41524-019-0221-0

97. K. Schmidt. “Another One Cleans the Dust. A Deep Learning-based Interpretation of an All-time Classic”. Ph.D. Thesis. TU Dortmund University, 2022.
DOI: 10.17877/DE290R-23151

98. K. Schmidt et al. *radionets-project/pyvisgen: v0.2.0*. Version v0.2.0. 2024.
DOI: 10.5281/zenodo.11622495

99. K. Schmidt et al. *radionets-project/radionets: v0.3.0*. Version v0.3.0. 2023.
DOI: 10.5281/zenodo.8205263

100. Schmidt, K. et al. “Deep learning-based imaging in radio interferometry”. *A&A* 664, 2022, A134.
DOI: 10.1051/0004-6361/202142113

101. M. Seitzer et al. “On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks”. *International Conference on Learning Representations*. 2022.
DOI: 10.48550/arXiv.2203.09168. URL: https://openreview.net/forum?id=aPOpXlnV1T

102. 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

103. S. E. Strom, S. Edwards, and M. F. Skrutskie. “Evolutionary Time Scales for Circumstellar Disks Associated with Intermediate-Type and Solar-Type Stars”. *Protostars and Planets III*. Ed. by E. H. Levy and J. I. Lunine. 1993, page 837

104. G. B. Taylor, C. L. Carilli, and R. A. Perley. “Synthesis Imaging in Radio Astronomy II”. *Synthesis Imaging in Radio Astronomy II*. Vol. 180. Astronomical Society of the Pacific Conference Series. 1999

105. The Event Horizon Telescope Collaboration et al. “First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole”. *The Astrophysical Journal Letters* 875:1, 2019, page L1.
DOI: 10.3847/2041-8213/ab0ec7

106. A. R. Thompson, J. M. Moran, and G. W. Swenson. *Interferometry and synthesis in radio astronomy*. 3rd ed. Springer Nature, 2017.
DOI: 10.1007/978-3-319-44431-4

107. 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

108. 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

109. A. R. Thompson et al. “The Very Large Array.” *Astrophysical Journal Supplement Series* 44, 1980, pages 151–167.
DOI: 10.1086/190688

110. S. E. Thorsett and D. Chakrabarty. “Neutron Star Mass Measurements. I. Radio Pulsars”. *The Astrophysical Journal* 512:1, 1999, page 288.
DOI: 10.1086/306742

111. D. Ulyanov, A. Vedaldi, and V. Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization*. 2017.
DOI: 10.48550/arXiv.1607.08022. eprint: 1607.08022

112. 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

113. M. Valdenegro-Toro and D. S. Mori. “A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pages 1508–1516.
DOI: 10.1109/CVPRW56347.2022.00157

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

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

116. Z. Wang, E. Simoncelli, and A. Bovik. “Multiscale structural similarity for image quality assessment”. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2.
DOI: 10.1109/ACSSC.2003.1292216

117. Z. Wang et al. “Image quality assessment: from error visibility to structural similarity”. *IEEE Transactions on Image Processing* 13:4, 2004, pages 600–612.
DOI: 10.1109/TIP.2003.819861

118. J. C. Webber. “The ALMA Telescope”. *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*. 2013, pages 1–3.
DOI: 10.1109/MWSYM.2013.6697433

119. J. P. Williams and L. A. Cieza. “Protoplanetary Disks and Their Evolution”. *Annual Review of Astronomy and Astrophysics* 49:Volume 49, 2011, 2011, pages 67–117.
DOI: 10.1146/annurev-astro-081710-102548

120. H. Zhao et al. *Loss Functions for Neural Networks for Image Processing*. 2018.
DOI: 10.48550/arXiv.1511.08861. eprint: 1511.08861

Glossary

ADAM Adaptive Moment Estimation. 18

AGN Active Galactic Nucleus. 10–12

AIPS Astronomical Image Processing System. 7

ALMA Atacama Large Millimeter/submillimeter Array. 8, 10, 13, 21, 29, 32, 35, 81, 83, 90, 106, 107

CHIME Canadian Hydrogen Intensity Mapping Experiment. 54

CNN Convolutional Neural Network. 2, 16, 19, 35, 49

DFT Discrete Fourier Transform. 7

DSHARP Disk Substructures at High Angular Resolution Project. iii, iv, 2, 8, 10, 13, 21, 29–33, 83, 84, 86, 89–91

FFT Fast Fourier Transformation. 7, 38

FIRST Faint Images of the Radio Sky at Twenty-cm. 21, 23, 26, 27, 89

FITS Flexible Image Transport System. 7, 83

FoV Field of View. 28, 32, 80, 83, 91

FRI Fanaroff-Riley Class I. 10, 12, 21–23, 69, 89, 90, 108

FRII Fanaroff-Riley Class II. 12, 21–23, 69, 89, 90, 108

FWHM Full Width at Half Maximum. 85

GAN Generative Adversarial Network. iv, 21–23, 25, 27, 69–73, 80, 81, 83, 90, 108

GPU Graphics Processing Unit. 89

L1 Mean Absolute Difference. 38

MRI Magnetic resonance imaging. 65

ms measurement sets. 83

MSE Mean Squared Error. 38, 67

MS-SSIM Multi-Scale Structural Similarity. 2, 51, 53, 54, 58–60, 64, 73, 74, 77–79, 90

ngVLA next-generation Very Large Array. 1, 8, 89

NLL Negative Log-Likelihood. 65

PReLU Parameter Rectified Linear Unit. 17, 18, 35, 36

ReLU Rectified Linear Unit. 17, 18, 67, 73, 90

ResBlock Residual Block. 35, 36

RIME Radio Interferometer Measurement Equation. iv, 1–4, 6–8, 24, 26–28, 32, 33, 89–91, 106

RMS Root Mean Square. 23

SEFD System Equivalent Flux Density. 28, 32

SKA Square Kilometre Array. 1, 89

SNR signal-to-noise ratio. iv, 23–25, 28, 49, 54–60, 64, 90

SRResNet Super Resolution Residual Network. 2, 35

SSIM Single-Scale Structural Similarity. 51–53

VLA Karl G. Jansky Very Large Array. 8, 9, 21, 26, 32, 35, 43, 81, 89, 111

VLBA Very Long Baseline Array. 3

VLBI Very Long Baseline Interferometry. 3, 26, 54

wGAN Wasserstein generative adversarial network. 21–24, 26, 36–38, 41, 43–46, 49–53, 89, 90

WSCLEAN w-Stacking Clean. 7, 23–25, 28, 83, 85–87, 91

Appendix

1 Funding Acknowledgements

Part of this work has been supported by Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Centers (Sonderforschungsbereiche – SFBs) SFB 876 “Providing Information by Resource-Constrained Analysis” and SFB 1491 “Cosmic Interacting Matters – From Source to Signal”. Furthermore, this work was supported by the project “Big Bang to Big Data (B3D)”, part of the program “Profilbildung 2020”, an initiative of the Ministry of Culture and Science of the State of North Rhie-Westphalia. The presentation and discussion of results obtained in this work at conferences were partly supported by the Wilhelm and Else Heraeus Stiftung.

I gratefully acknowledge the computing time provided on the Linux HPC cluster at Technical University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

I want to express my gratitude to the abovementioned funding agencies for providing me with the resources that made this work possible.

2 Antenna Locations for ALMA

Table A.1: Overview of the antenna positions of the ALMA, used as a parameter for the RIME.

Station Name	X / m	Y / m	Z / m
DA42	2225109.99	-5440002.41	-2481782.63
DA44	2224896.3	-5439906.96	-2482190.3
DA45	2224774.74	-5440235.55	-2481577.82
DA46	2225029.59	-5440081.85	-2481682.27
DA47	2225176.66	-5439964.25	-2481800.73
DA48	2225193.45	-5439993.76	-2481722.54
DA49	2225085.76	-5440062.1	-2481674.24
DA51	2225053.23	-5440093.37	-2481635.63
DA52	2225011.32	-5440147.99	-2481558.05
DA53	2225199.43	-5440058.59	-2481572.0
DA54	2225122.7	-5439951.13	-2481886.48
DA55	2224943.17	-5440088.85	-2481748.58
DA57	2225505.36	-5440025.43	-2481355.57
DA58	2224585.55	-5440348.81	-2481491.68
DA59	2225070.07	-5440067.18	-2481677.13
DA60	2225045.05	-5439869.96	-2482140.63
DA61	2225119.13	-5440069.22	-2481628.01
DA62	2224962.99	-5440337.73	-2481190.21
DA63	2224619.06	-5440204.46	-2481770.14
DA64	2225113.9	-5440088.81	-2481588.45
DA65	2225450.18	-5439794.98	-2481913.74
DV01	2224798.84	-5440161.3	-2481726.02
DV03	2224825.69	-5439918.56	-2482224.39
DV05	2225375.65	-5440111.88	-2481291.8
DV06	2224943.67	-5439974.24	-2482014.29
DV07	2225078.25	-5440185.64	-2481414.95
DV08	2224906.77	-5440173.24	-2481599.34
DV09	2224781.3	-5440342.02	-2481336.1
DV10	2225488.56	-5439872.88	-2481699.24
DV11	2225072.24	-5440148.43	-2481498.97
DV12	2225180.9	-5440025.86	-2481662.78
DV13	2224980.92	-5440130.82	-2481620.87
DV14	2225196.57	-5439865.59	-2482003.17

Table A.2: Continued from above

Station Name	X / m	Y / m	Z / m
DV15	2225119.37	-5440042.85	-2481684.94
DV16	2225569.05	-5439979.43	-2481387.23
DV17	2224948.42	-5440039.64	-2481852.43
DV18	2225099.28	-5440075.03	-2481631.75
DV19	2224759.17	-5440069.47	-2481944.57
DV20	2225188.12	-5440189.91	-2481301.78
DV22	2224910.49	-5440129.39	-2481688.89
DV23	2225090.69	-5440083.26	-2481622.45
DV24	2225255.26	-5440008.99	-2481623.35
DV25	2225269.67	-5439908.28	-2481832.2
DA41	2225094.8	-5440052.42	-2481687.28
DA43	2224863.87	-5440088.01	-2481814.53
DA56	2225270.73	-5440073.09	-2481471.42
DV04	2225027.12	-5439962.76	-2481958.28
PM01	2225113.04	-5440122.82	-2481517.73
PM02	2225042.28	-5440125.51	-2481574.9
PM03	2225046.0	-5440149.14	-2481520.12
PM04	2225104.7	-5440102.47	-2481568.69
DA41	2225147.95	-5439675.04	-2482445.48
DA42	2225279.58	-5439469.36	-2482833.13
DA43	2224801.83	-5439786.46	-2482539.25
DA44	2224236.18	-5439969.73	-2482606.26
DA46	2226329.78	-5440032.87	-2480531.34
DA48	2224116.15	-5442159.48	-2477930.45
DA49	2224469.6	-5440842.53	-2480524.66
DA50	2225718.6	-5439808.04	-2481613.94
DA51	2225714.63	-5440369.47	-2480423.24
DA54	2219253.64	-5442894.2	-2479851.61
DA55	2226163.04	-5439682.72	-2481421.98
DA60	2226265.19	-5438432.83	-2484137.88
DA63	2227320.32	-5436658.14	-2486450.59
DA64	2229254.52	-5440457.76	-2476683.55
DA65	2227504.63	-5439905.32	-2479594.88
DV01	2225669.04	-5439445.53	-2482517.53
DV02	2225434.81	-5439701.86	-2482133.29
DV04	2229234.14	-5440047.54	-2477588.72
DV07	2225871.28	-5439985.17	-2481088.6
DV09	2225117.81	-5440052.28	-2481665.8
DV10	2226186.01	-5439426.1	-2482003.42
DV11	2224148.99	-5440731.57	-2481004.97
DV15	2223501.9	-5440635.38	-2481765.56
DV17	2222794.8	-5441831.65	-2479695.83
DV22	2224521.0	-5441793.95	-2478423.75
DV23	2224986.55	-5440968.64	-2479799.62
DV24	2225376.5	-5439991.42	-2481543.23
PM01	2224781.47	-5440342.45	-2481336.3
PM02	2225523.32	-5440197.67	-2480975.27
PM03	2226201.28	-5439135.84	-2482771.54
PM04	2223832.08	-5440054.89	-2482719.9

3 Loss Curves for the Uncertainty Training

3.1 FRI and FRII Sources

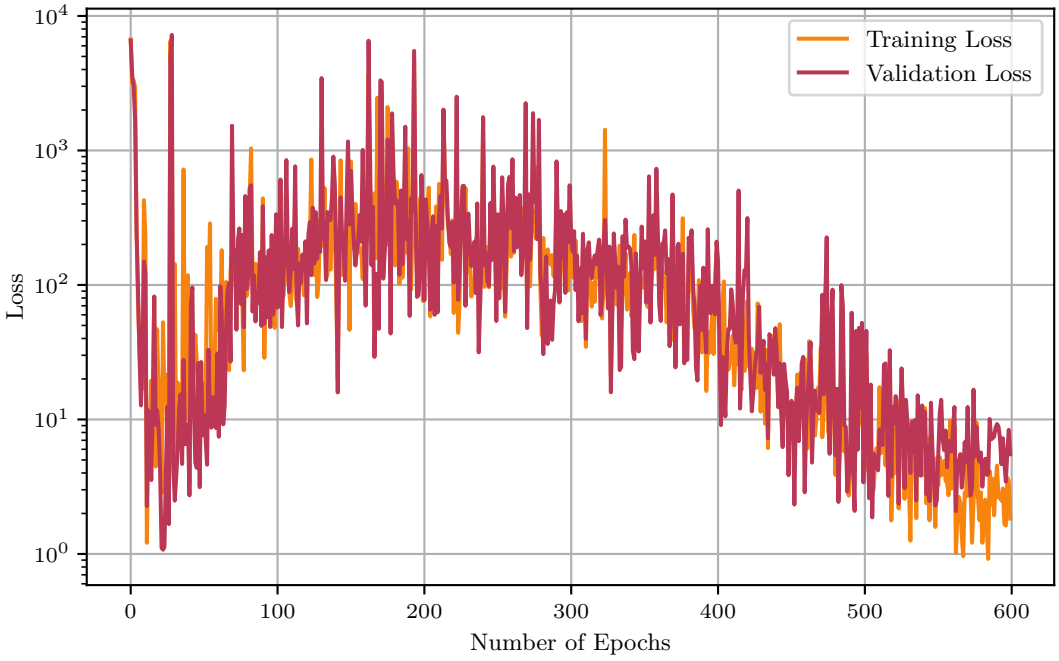

Figure A.1: Training and validation loss for an uncertainty training using the GAN-generated sources.

3.2 Protoplanetary Disks

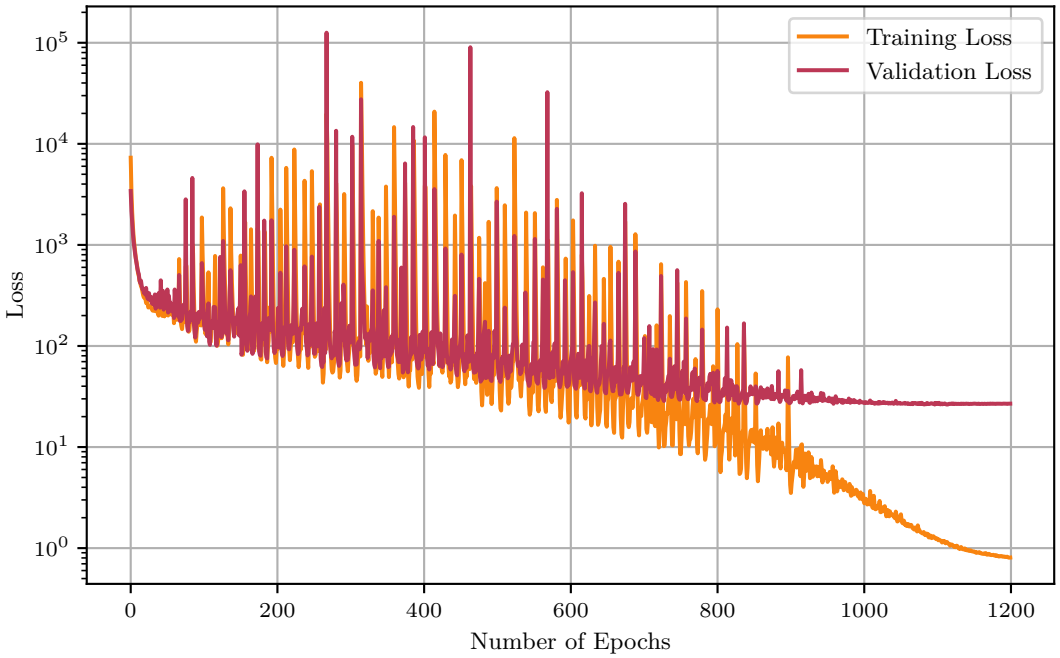

Figure A.2: Training and validation loss for an uncertainty training using the protoplanetary disks.

4 Visualization of Jones Matrices

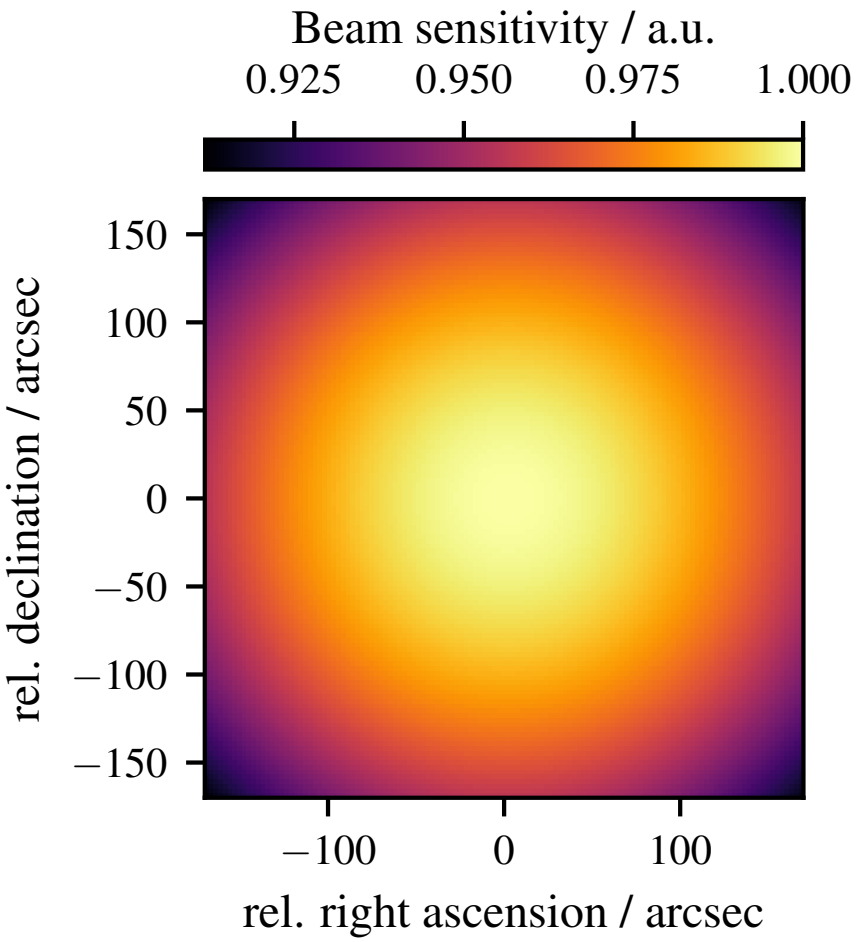

Figure A.3: Visualization of the $\mathbf{E}(l, m)$ matrix, which represents the direction-independent antenna beam [39].

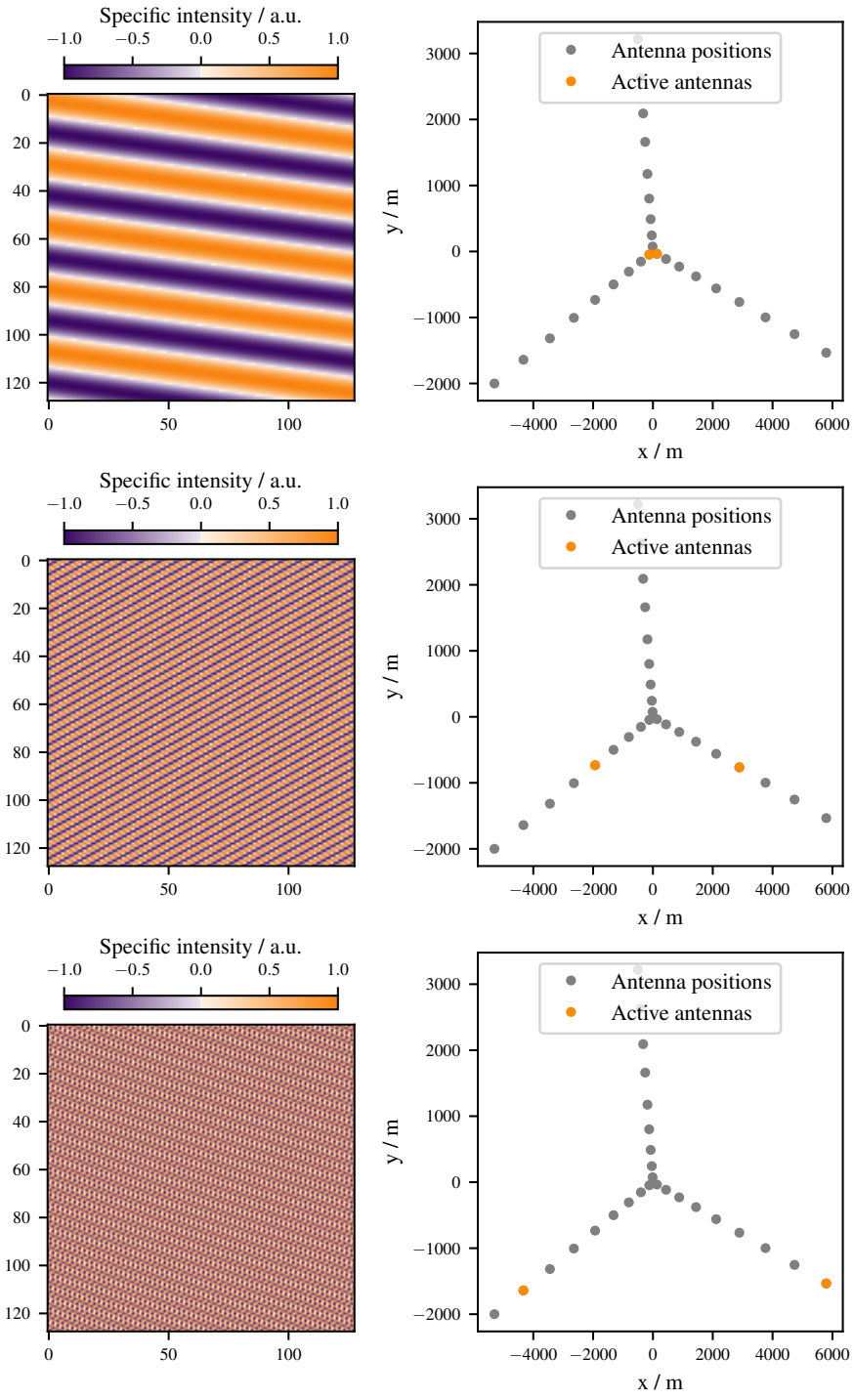

Figure A.4: Visualization of the $\mathbf{K}(l, m)$ matrix using the VLA antenna layout. On the right side, three different telescope pairs of the VLA are marked in orange. The corresponding Jones $\mathbf{K}(l, m)$ is shown on the left. One can see that for increasing distances, the power pattern gets finer. This is a visual representation of one of the core statements of radio interferometry: Small baseline are responsible for the coarse structures of the image, and large baselines measure the fine-sale structures. [39]

Acknowledgements

First, I would like to thank my supervisor, Wolfgang Rhode, for allowing me to pursue a PhD and providing me with all the resources and contacts necessary to achieve this goal. You encouraged me to explore new paths and made it possible for me to continue researching the topic in my dissertation after my Master's degree. Also, I want to thank Anna Franckowiak for acting as my second supervisor and for her interest in my work. Furthermore, thank you to Marcus Brüggen for the continuous support of the project and the valuable insights and discussions about radio interferometry, which answered so many open questions. Many thanks also to Dominik Elsässer for answering any of my astrophysical questions and giving the astroparticle physics lecture, which was the main reason I decided to focus my studies on this topic.

Thank you to the whole astroparticle physics working group for the pleasant working atmosphere and all the interesting discussions, whether work-related or not. A special thank you goes out to the past and present members of the radio group; it was always a pleasure to work and laugh with all of you. Kevin, thank you very much for all the advice, insights, and nonsense you shared with me since starting the radionets project in 2019. Working with you all these years was a pleasure, and I am grateful for your contribution to this thesis. Also, thank you to Andrea and Sarah for allowing me to focus on the fun parts of the PhD and helping me with every kind of bureaucracy I encountered during my time in this working group.

Another thank you is owed to the members of our study group "harter Kern", with which I spend almost all my time during my Bachelor's and my Master's degree. You helped me with both learning and forgetting all of the physics we encountered during our time in Dortmund. Furthermore, I want to thank Alex, Arthur, Froch, Julian, Konstantin, Laura, and Lukas for your friendship and all the evenings we spent together. Also, thank you to my past and present office colleagues Jonas, Kevin, Rune, and Yvonne, who made work much more enjoyable. To Rune: What a journey starting by being randomly matched together and ending up submitting our PhD theses on the same day.

A big thank you to my family, who always supported me with everything I needed. Your assistance and help mean everything to me.

My deepest gratitude belongs to Jannine, who was by my side all the way since the beginning. Without you, nothing of this would have been possible. Thank you for being you and being with me, and I cannot wait to spend the rest of my life with you.