Università degli Studi di Napoli Federico II

Ph.D. Program in

Information Technology and Electrical Engineering

XXXV Cycle

Thesis for the Degree of Doctor of Philosophy

Resolution of Inverse Problems in Astrophysics through Deep Learning

by

Michele Delli Veneri

Advisor: Prof.Vincenzo Moscato

Co-advisor: Prof. Giuseppe Longo

Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

This Thesis is dedicated to all my friends,
my girlfriend Marta,
Peppe and my mum and dad Paola and Paolo.
Your support and the mental issues that you gave
me during this three years, made all this possible.
Ad Astra Per Aspera

Resolution of Inverse Problems in Astrophysics through Deep Learning

Ph.D. Thesis presented
for the fulfillment of the Degree of Doctor of Philosophy
in Information Technology and Electrical Engineering
by

Michele Delli Veneri

October 2022

Approved as to style and content by

Prof.Vincenzo Moscato, Advisor

Giuseppe Longo

Prof. Giuseppe Longo, Co-advisor

Università degli Studi di Napoli Federico II
Ph.D. Program in Information Technology and Electrical Engineering
XXXV cycle - Chairman: Prof. Stefano Russo

http://itee.dieti.unina.it

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Information Technology and Electrical Engineering is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
Parts of this dissertation have been published in international journals and/or conference articles (see list of the author's publications at the end of the thesis).

Napoli, November 23, 2022

Michele Delli Veneri

Abstract

Current and forthcoming Astronomical observatories are rapidly increasing the quantity, velocity and complexity of their data products pushing Astronomy in the Big Data regime. Extracting scientifically usable data from such instruments involves the resolution of ill-posed inverse problems traditionally solved with algorithms which cannot cope anymore with the rising complexity. In the last decade, Machine Learning has seen a deep rise in its use both within and outside Astronomy. In this Thesis, I have developed a set of Deep Learning (DL) based pipelines aimed at the resolution of two such problems: the Radio Interferometric Deconvolution, Source Detection and Characterisation problem for two different radio interferometers, the Atacama Large Millimeter/submillimeter Array (ALMA) and the Square Kilometer Array (SKA), and the TOLIMAN space telescope Astrometric signal detection problem. Given the novelty of the instruments and the need for controlled experiments for the development and comparison of solutions, all studies carried out in this Thesis use simulated data. SKA and TOLIMAN data were acquired through my participation in the SKA Data Challenge 2 and COIN TOLIMAN Focus meeting, while I developed a simulation framework able to generate the needed ALMA observations by levering parallel computing. The ALMA pipeline is composed of six DL models: a Convolutional Autoencoder (CAE) for source detection within the spatial domain of the integrated data cubes, a Recurrent Neural Network (RNN) for denoising and peak detection within the frequency domain, and four Residual Neural Networks (ResNets) for source characterisation. The detection performances of the pipeline were compared to those of other state-of-the-art methods within the field and significant improvements in performances and computational times are achieved. Source morphologies are detected with subpixel accuracies obtaining mean residual errors of 10^{-3} pixel (0.1 mas) and 10^{-1}

mJy/beam on positions and flux estimations, respectively. Projection angles and flux densities are also recovered within 10% of the true values for 80% and 73% of all sources in the test set, respectively. A direct comparison with TCLEAN, the current image deconvolution method employed by CASA, the ALMA data reduction pipeline, is made on simplified mock data achieving a substantial improvement in reconstruction quality and speed. The SKA pipeline, which I developed to address the shortcomings of the baseline pipeline developed during the Challenge in collaboration with COIN, is based on a combination of a classical Compressed Sensing algorithm, my 3D implementation of the Multi Vision Model, with six DL models: A 3D CAE for source detection, a 3D ResNet classifier to detect and remove false detections, and four 3D ResNet regressors to predict sources morphological parameters. The performances of the debugged, re-trained and optimised baseline pipeline and the revised pipeline are compared with those of the other solutions to the challenge. The revised pipeline reaches the highest score with slight improvements over the challenge winners. The TOLIMAN pipeline is the only unsupervised pipeline developed in this Thesis and it is based on a CAE tasked with compressing the TOLIMAN image time series into a monodimensional latent space which is then analysed through a Lomb-Scargle periodogram in search of periodic components. The pipeline performances in detecting increasingly small and realistic Astrometric signals embedded within a series of simulated TOLIMAN observations of the Alpha Cen star system are compared to those of other sparsity-based state-of-the-art solutions within the field. The signals are simulated as time-dependent shifts in the positions of two overlapping point spread functions in the TOLIMAN images. Our pipeline is the only one which can reliably detect the signal with an amplitude of 10^{-6} times the pixel size. The simulations contained only Poisson noise, in future works, all the more realistic sources of noise and systematic effects present in the real-world satellites will be injected into the simulations.

Keywords: methods: data analysis, methods: machine learning, techniques: image processing, techniques: interferometric, techniques: astrometric, software: simulations

Sintesi in lingua italiana

Gli attuali e futuri osservatori continuano ad incrementare la quantità, la velocità e la complessità dei dati prodotti spingendo l'Astronomia nel regime dei Big Data. L'estrazione di dati scientificamente utilizzabili da tali strumenti richiede la risoluzione di problemi inversi mal posti tradizionalmente risolti con algoritmi che non sono più in grado di tener testa all incrementale complessità dei dati. Nell'ultimo decennio l'utilizzo del Machine Learning ha visto un notevole incremento sia all'interno che al di fuori dell'Astronomia. In questo lavoro di Tesi, ho sviluppato una serie di "pipeline" basate su modelli di Deep Learning (DL) con lo scopo di risolvere due problemi inversi: il problema di deconvoluzione radio interferometrica e successiva rilevazione e classificazione delle sorgenti astronomiche per l'Atacama Large Millimeter/submillimeter Array (ALMA) e lo Square Kilometer Array (SKA), ed il problema di rilevazione del segnale astrometrico per il telescopio spaziale TOLIMAN. Data la natura degli strumenti e la necessità di eseguire esperimenti controllati necessari alo sviluppo ed il confronto di soluzioni ai problemi preposti, tutti gli studi portati avanti in questa Tesi usano dati simulati. I dati di SKA e TOLIMAN sono stati acquisiti tramite la mia partecipazione, rispettivamente, alla SKA Data Challenge 2 e al COIN TOLIMAN Focus meeting. I dati di ALMA sono stati creati tramite simulazioni in grado di generare i dati necessari sfruttando il paradigma di computazione parallela. La pipeline per ALMA è formato da sei modelli di DL: un Convolutional Autoencoder (CAE) per il relevamento delle sorgenti nel dominio spaziale dei data cubi integrati, una Recurrent Network (RNN) per ridurre il rumore e rilevare picchi di emissione in frequenza, e quattro Residual Neural Networks (ResNets) per caratterizzare le sorgenti. Le prestazioni della pipeline proposta sono state comparate a quelle di altri metodi che rappresentano lo stato dell'arte nel settore, dimostrando migliori prestazioni e

tempi computazionali. Le caratteristiche delle sorgenti sono registrate con accuratezze inferiori alle dimensioni del pixel ottenendo errori residui medi di 10^{-3} pixel (0.1 mas) e 10^{-1} mJy/beam rispettivamente sulle posizioni e sui flussi. Gli angoli di proiezione e i flussi sono entro il 10% del loro valore vero per rispettivamente l'80% ed il 73% di tutte le sorgenti presenti nel test set. Si è anche eseguita una comparazione diretta della qualità di ricostruzione con τCLEAN, l'algoritmo di deconvoluzione attualmente utilizzato dalla pipeline di riduzione dati di ALMA, sui nostri semplici dati simulati ottenendo sostanziali miglioramenti nella qualità della ricostruzione e nei tempi computazionali. La pipeline per SKA, è stata sviluppata per correggere le principali carenze dimostrate dalla pipeline, usata nel contesto della Tesi come "baseline", che ho sviluppato, durante la SKA Data Challenge 2, in collaborazione con altri membri COIN. La pipeline è basata sulla combinazione di un classico algoritmo di Compressed Sensing, un'implementazione 3D del Multi Vision Model, con sei algoritmi di DL: un 3D CAE per rilevare le sorgenti, una 3D ResNet usata come classificatore per rilevare e rimuovere falsi rilevamenti, e quattro 3D ResNet usate come regressori per predirre i parametri morfologici delle sorgenti. Le prestazioni della pipeline di baseline, dopo aver eseguito estensivo debugging, riaddestramento e ottimizzazione, e quelle della pipeline da me sviluppata sono confrontate con quelle degli altri partecipanti alla Challenge. La pipeline proposta ha notevolmente migliorato i risultati ottenuti nella Challenge con prestazioni leggemente superiori rispetto ai vincitori. La pipeline per TOLIMAN è l'unica non supervisionata tra quelle presentate nella Tesi ed è basata su un CAE addestrato per comprimere la serie di immagini di TOLIMAN in uno spazio latente monodimensionale, per poi essere analizzato in cerca di componenti periodiche con il periodogramma di Lomb-Scargle. Le prestazioni della pipeline proposta nel rilevare segnali incrementalmente più deboli e realistici ignettati all'interno di una serie di osservazioni da parte di TOLIMAN del sistema stellare Alpha Cen, sono

comparate con quelle di altre soluzioni basate su sparsità rappresentanti lo stato dell'arte nel settore. I segnali sono simulati come variazioni dipendenti dal tempo delle posizioni relative di due Point Spread Functions sovrapposte nelle imagini di TOLIMAN. La pipeline proposta è l'unica in grado di rilevare il segnale con ampiezza di 10^{-6} volte le dimensioni del pixel. I dati simulati contenenvano unicamente rumore Poissoniano, in lavori futuri prevediamo di considerare le ulteriori sorgenti di rumore e sistemetiche presenti nei satelliti reali.

Parole chiave: methods: data analysis, methods: machine learning, techniques: image processing, techniques: interferometric, techniques: astrometric, software: simulations

Contents

Acknowledgements

The author's work has been supported by a PhD scholarship funded by EUSTEMA S.p.A.

The author wants to thank The IBISCO HPC admin group for allowing us to extensively use their cluster to simulate the data and carry out all our experiments. Also I want to to thank the European Souther Observatory (ESO) and the Dept. of electric Engineering and Information Technologies at the University Federico II for partial financial support.

The author wants to thank the Bernoullibourg Faculty of Science and Engineering at the University of Groningen for partial financial support and for the use of computational resources.

The author's work has been partially supported by an ESO internal ALMA development study investigating interferometric image reconstruction methods and partially sponsored by EU through the SUNDIAL ITN.

The author's work was partially produced in collaboration with members of COIN for the duration of the SKA Data Challenge 2 and during the 2^{nd} COIN-Focus: TOLIMAN Event held in Rome, Italy, in November 2019.

The author wants to thank F. Guglielmetti, L. Tychoniec, E. Villard, A. Krone Martins, E. Ishida, C. Heneka, B. Fraga, M. Biehl, J. McKean, R. Abbruzzese, S. Cavuoti, M. Brescia, G. Sperli', V. Moscato, G. Longo and A. Picariello for their supervision in solving the various scientific problems addressed during my PhD and in this Thesis.

List of Acronyms

The following acronyms are used throughout the thesis.

ML	Machine Learning
DL	Deep Learning
PCA	Principal Component Analysis
PDF	Probability Density Function
PSF	Point Spread Function
ANN	Artificial Neural Network
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
AE	AutoEncoder
GRU	Gated Recurrent Unit
ResNet	Residual Neural Network
RNN	Recurrent Neural Network
CASA	Common Astronomy Software Applications

SNR	Signal to Noise Ratio
SGD	Stochastic Gradient Descent
LSTM	Long Short Term Memory Network
PCA	Principal Component Analysis
LSP	Lomb-Scargle Periodogram
ALMA	Atacama Large Millimeter/submillimeter Array
SKA	Square Kilometer Array
TF	Transfer Function
FoV	Field of View
FAST	Five hundred meter Aperture Spherical Telescope
MAP	Maximum A Posteriori
S + C	Smooth and Clip
WDL	Wasserstein Dictionary Learning
TOLIMAN	Telescope for Orbital Locus Interferometric Monitoring of our Astrometric Neighbourhood
ESO	European Southern Observatory
AI	Artificial Intelligence
FWHM	Full Width Half Maximum
RMS	Root Mean Square
IBISCO-HPC	Infrastructure for Big data and Scientific Computing

ReLU	Rectified Linear Unit
BCE	Binary Cross Entropy
DSSIM	Difference of Structural Similarity
MLP	Multi Layer Perceptron
TP	True Positive
FP	False Positive
BF	Blobs Finder
FN	False Negative
FPR	False Positives Rate
IoU	Intersection over Union
SKAO	SKA Observatory
SKADC2	SKA Data Challenge 2
COIN	Cosmostatistics Initiative
FFT	Fast Fourier Transform
FL	Focal Length
MRS	Mean Residual Score
FAP	False Alarm Probability
RA	Right Ascension
Dec	Declination
IFK	Fredholm Integral Equations of the first kind

List of Figures

List of Tables

List of Symbols

The following symbols are used within the thesis

ν	Frequency
θ	Projection angle
c	Speed of light in a vacuum
Hz	Hertz
i	Inclination angle
Jy	Jansky
km	Kilometers
s	Second

Chapter 1

Introduction

Hello there!

Master Obi-Wan Kenobi, Revenge of the Sith

In the last two decades, astronomical datasets underwent a rapid growth in size and complexity thus pushing Astronomy in the Big Data regime [97, 11, 119, 167, 22]. These early trends are even more relevant nowadays since new instruments, which are either ready to enter the operation phase (e.g. the Large Synoptic Survey Vera Rubin Telescope - LSST, or EUCLID) or will become operational in the next three years (e.g. the Square Kilometre Array - SKA and the Extremely Large Telescope - ELT), not to mention the already operational James Webb Telescope, will push the stream of newly processed data to unprecedented ranges of $\simeq 30 - 100$ TBs per day. Extracting scientific data from raw observations usually requires the resolution of ill-posed inverse problems and traditional approaches cannot cope anymore with the rising data complexity. Comes thus with no surprise, given that astronomers had pioneered and embraced data virtuous strategies and "data driven" approaches to optimise the data collection, distribution, and processing for most astronomical instruments (see, for example, early virtuous examples such as the creation of an accurate "astronomical ontology" or the Virtual Observatory Alliance - IVOA[1] or the creation of specific professional societies such as the the

[1] https://ivoa.net

Astrostatistics International Society and the International Astroinformatics Society[2]), that the ability of machine learning methodologies, both supervised and unsupervised, to cope with very complex data, has been extensively exploited by the community to solve a wide variety of problems spanning all aspects of the astronomical data life, from instrument monitoring to data acquisition and ingestion, to data analysis and interpretation [12, 82, 73, 102, 84, 109, 147, 169, 93, 43, 29, 41, 90, 123]. Chapter 2 summarises the basic concepts about inverse problems and briefly describes those which are encountered in this Thesis: the Radio Interferometric deconvolution and the subsequent source detection and characterisation problems and the Telescope for Orbital Locus Interferometric Monitoring of our Astrometric Neighbourhood Telescope for Orbital Locus Interferometric Monitoring of our Astrometric Neighbourhood (TOLIMAN) satellite signal detection problem. The Chapter shows how each of these problems can be formulated as an ill-conditioned inverse problem, what are the general strategies to solve it, and provides an outline of widely accepted strategies adopted by the community. Finding the solution to an ill-conditioned inverse-problem poses several difficulties regardless of whether analytical or numerical techniques are used, and, in order to make this problem well-conditioned (solvable), regularization techniques and iterative strategies can be employed. Regularization allows in fact to reduce the space of candidate solutions by incorporating some knowledge about the nature of the data. This is achieved by reformulating the problem in such a way that the solution to the new problem is less sensitive to perturbations. Prior information usually accounts for the "smoothness" class of solution and can go from the simple assumption of uniform smoothness to a much more complex knowledge about the geometrical structures of the solutions (e.g.: the shape of the Point Spread Function Point Spread Function (PSF), symmetry of the sources, isotropy, non-negativity etc.). An outline of Machine Learning based solutions to these problems is instead presented in Chapter 3. As a matter of fact, in recent years, Machine Learning has become more and more a powerful tool to solve ill-conditioned imaging inverse problems in many fields of science and industry. ML offers in fact the possibility to learn how to solve problems by automatically identifying patterns without human assistance. This is crucially important in astrophysics which

[2] http://astroinformatics.info/astroinfo

relies on observations and incomplete models of the underlying phenomena. Furthermore, the relatively large errors in the data may result in poorly constrained models which makes the solution to these problems even more problematic. Chapter 3 aims therefore at introducing those basic concepts behind Machine Learning Machine Learning (ML) and Deep Learning Deep Learning (DL) which are needed to understand the inner workings of all the DL models I built to solve the problems addressed in this Thesis. First, I present a broad overview of the AI field and the basic definitions and structures which are required by all ML models, and then I showcase a detailed analysis of all the DL and ML models I implemented and utilised in this Thesis. Finally, I also showcase some ML and DL applications to the resolution of other astrophysical inverse problems.

Chapters 4, 5 and 6 present my original work: i.e. the development of three DL pipelines for the resolution of three different inverse problems within the field of Astronomy, the first two in the setting of Radio Interferometry, the latter in the field of Narrow-Angle Astrometry. The three problems addressed in the respective Chapters are listed below:

- Chapter 4: **Detection and characterisation of the Atacama Large Millimeter Array Atacama Large Millimeter/submillimeter Array (ALMA) Sources through Deep Learning**: the fourth Chapter focuses on the performances of a DL-based pipeline I implemented in order to reconstruct, detect and characterise sources within simulated ALMA observations. These are 3D, noisy datacubes obtained by passing simulated sources through the mathematical model of the ALMA interferometric array. This investigation was needed to test if DL-based pipelines could be used to deliver improvements in both speed and performance over more traditional solutions. This involves the creation of realistic simulations of the ALMA data products, which are made available to the Astronomical community, and the comparison of the DL pipeline with competing solutions and other traditional counterparts in solving the image deconvolution and denoising problem, and then the source detection and characterisation problems;

- Chapter 5: **The Square Kilometre Array Square Kilometer Array (SKA) Data Challenge 2: Detection and Character-**

ization of SKA Sources through Deep Learning: the fifth Chapter of the Thesis investigates the performance of two novel DL-based pipeline in detecting and characterising sources within a simulated Square Kilometre Array (SKA) data cube. The investigation started with my participation, in collaboration with several members of the Cosmostatistic Initiative (COIN), in the SKA Data Challenge 2, which ended up developing a first pipeline that is used as a baseline in the Chapter. Two revised pipelines based on the baseline are presented and compared against other DL-based and classical pipelines which participated in the challenge. While the original pipeline did not show competing performances, nevertheless being in the top 10 scorer of the Challenge, the newly revised pipeline I implemented to address the shortcomings of our previous attempt, was able to compete with the winners of the Challenge.

- Chapter 6: **The TOLIMAN Signal Detection Problem**: the sixth Chapter of the Thesis investigates the performances of a DL-based pipeline I implemented to recover the astrometric signal embedded in TOLIMAN simulated data. Astrometric detection involves precise measurements of stellar positions, and it is regarded among the leading methods to find Earth-mass planets in temperate (i.e. within the habitability zone) orbits around nearby sun-like stars. The TOLIMAN space telescope [152] is a low-cost, agile space mission which will be devoted to performing narrow-angle astrometric monitoring of bright binary stars. In particular, the task will be optimised to search for habitable-zone planets around α Centauri AB. If the separation between these two stars can be monitored with sufficient precision, tiny perturbations due to the gravitational tug from an unseen planet can be witnessed and, given the configuration of the optical system, the scale of the shifts in the image plane is about one-millionth of a pixel. The Chapter showcases how a Convolutional Auto-Encoder is able to retrieve the astrometric signal from simplified simulations of the TOLIMAN data and we present the entire experimental pipeline to recreate our experiments from the simulations to the signal analysis. The performance of the unsupervised DL model is then compared with other state-of-the-art algorithms in the field of compress sensing.

While the three problems are presented in the above order, the research was carried out along a different path: the TOLIMAN signal detection was the first problem I tackled. Then came the SKA Data Challenge 2 and my stay in Groningen (NL) in which I worked on my revised solution based on the experience acquired within the COIN collaboration. Finally, I used the knowledge acquired on the SKA problem to revise a customised solution to the ALMA problem which became the main focus of this Thesis.

Chapter 7 summarises the work done and the main scientific results achieved, outlines whether DL algorithms can provide fast and reliable solutions for many challenges within Astronomy and outlines some future perspectives for the work done so far.

Chapter 2

Inverse Problems in Astrophysics

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.
There is another theory which states that this has already happened.

Douglas Adams

2.1 Inverse Problems

Many problems in science and engineering are about the derivation from a set of observations of the causal factors that produced them. In other words, we wish to find the relation connecting the physical parameters of a given model m, to a set of measurements b assuming that there is a causal relationship G connecting one to the other. Mathematically this kind of problem can be formulated as a set of equations:

$$\mathbf{G}(m) = \mathbf{b} \tag{2.1}$$

where $\mathbf{b} \in R^n$ are the observable quantities (or functions of observable quantities such as measurements), m is the mathematical model of the physical processes involved in the observing process (also a function of some analytical or numerical parameters), and $\mathbf{G}$ is the functional mapping relating the observations to the model. In this formalism, the forward problem consists in finding b given m, i.e. in predicting the observations given our knowledge about the phenomenon (simulations are forward problems). On the other hand, inverse problems consist in finding m given b. For that reason G is also known as the *forward operator* and, depending on the problem, it can be an ordinary differential equation (ODE), a partial differential equation (PDE) or a linear or nonlinear system of equations. In most imaging systems the noiseless observation can be approximately considered as a linear function of the model or, in other words, the G operator is linear [14]. Moreover, most imaging systems used in Astronomy are space invariant, and, as we will see later in this Chapter, this implies that the forward operator is a *convolution* with the Point Spread Function PSF of the imaging system. The considerations made so far do not take into account explicitly the fact that, in reality, measurements are always affected by errors and models are always an imperfect approximation of the true physics behind the phenomenon or the experimental setup. For this reason, it is better to formulate inverse problems as:

$$b = G(m) + \epsilon \tag{2.2}$$

where ϵ represents our uncertainty about the measurement process and the underlying physical process and it is commonly referred to as noise. This expression does not mean that we are assuming additive noise but only that there is a difference between the observed and the detected noiseless image. The aspect of inverse problems which makes them interesting and difficult to solve is that they tend to be ill-posed or in other words, they tend not to obey the Hadamard conditions. Namely: the existence of the solution; uniqueness of the solution; stability of the solution. These three conditions may be explained as follows:

1. Existence: for example, there may be no model that exactly fits the data and thus there is no analytical solution to the problem. This may happen because the model is not complex enough to fully cap-

ture the underlying physical processes or because the data contains noise;

2. Uniqueness: if an exact solution exists, it may not be unique, or in other words, there may be some other solutions G^* which fit the data.

3. Instability: an arbitrarily small perturbation of the data (or of the measurement process) may lead to an arbitrarily large perturbation of the solution.

To give a mathematical representation of the aforementioned conditions, let's start by recapitulating some basic concepts of integral equations. An operator G is a map between two functions domains X and Y that applies the function $\phi \in X$ to the function $\psi \in Y$ as it follows:

$$\begin{aligned} &G : X \to Y \\ &\phi \to \psi = G\phi \end{aligned} \tag{2.3}$$

we can also define the inverse operator G^{-1} where

$$\begin{aligned} &G^{-1} : G(X) \to X \\ &\psi \to \phi = G^{-1}\psi \end{aligned} \tag{2.4}$$

The existence and uniqueness of a solution for an equation using operators can be expressed as the existence of the inverse operator $G^{-1} : Y \to X$ (i.e the operator needs to be invertible). If this is the case then the solution can be expressed as $G^{-1}b = m$ and the operator is said to be bijective. The set of functions $G(X) = \{G\phi : \phi \ \in X\}$ is known as the rank of the operator. Given the problems addressed in this thesis, we can restrict our discussion to linear operators, i.e. one for which:

$$G(\alpha\phi_1 + \beta\phi_2) = \alpha G\phi_1 + \beta G\phi_2 \tag{2.5}$$

Two interesting properties of linear operators are boundness and compactness. An operator is said to be bounded if $\exists C > 0$ so that $\parallel G\phi \parallel \leq C \parallel \phi \parallel$. If a linear operator is closed and bounded it is said to be compact. It can be demonstrated that in order for a linear operator to be invertible it suffices that its inverse operator is bounded and that for a compact linear operator this is true only if its rank has finite dimensions. If the inverse operator is also continuous, then the third condition is satisfied, and the problem is considered well-conditioned. In practice, if any of these properties are not fulfilled, the problem is ill-conditioned and the search for the solution (if any) will encounter many numerical difficulties. Given the nature of the problems addressed in this Thesis, which are continuous, we can make a further step and confine our discussion to linear integral operators. Let us, therefore, assume that the inverse problem described by Eq. 2.1 is continuous, so we can express G as a linear integral operator

$$\int_a^b g(x,\xi)m(\xi)d\xi = b(x) \tag{2.6}$$

the function $g(x,\xi)$ is known as the kernel or PSF of the imaging system, and equations of this form (which arise in many inverse problems), where m(x) is the unknown, are known as *Fredholm Integral Equations of the first kind* Fredholm Integral Equations of the first kind (IFK). These equations are ill-conditioned problems unless constraints on the functional space exist or can be set by imposing that the spaces where the b and m functions belong have a finite dimension. This, of course, as we shall see later in this chapter, is not always possible. In many cases, the kernel can be written as explicitly dependent from $x-\xi$ thus leading to what is known as convolution equations

$$\int_{-\infty}^{+\infty} g(x-\xi)m(\xi)d\xi = b(x) \tag{2.7}$$

For instance, the problem of inverting a Fourier transform

$$\psi(f) = \int_{-\infty}^{+\infty} e^{-i2\pi f x}\phi(x)dx \tag{2.8}$$

to get $\phi(x)$ involves the resolution of an IFK.

A remarkable property of linear time-invariant (or space-variant problems in 2D and in 3D) is that the forward problem can be described as a convolution

$$b(t) = \int_{-\infty}^{+\infty} g(t-\tau)m(\tau)d\tau \tag{2.9}$$

in this case, the independent variable is the time, and the observations, the model and the kernel are all functions of time. Convolution generally can be used to describe all the mappings between models and observations of any linear spatially or time-invariant physical process, including the output of instruments such as radio interferometers. We can show that any linear time-invariant operator G which transforms an unknown model $m(t)$ into a set of observables $b(t)$ and that follows the principles of superimposition

$$G(m_1(t) + m_2(t)) = G(m_1(t)) + G(m_2(t)) \tag{2.10}$$

and scaling

$$G(\alpha m(t)) = \alpha G(m(t)) \tag{2.11}$$

can be reformulated as a convolution by utilising the sifting property of delta functions $\delta(t)$. Where the delta (or Dirac) function is described as

$$\delta(t) = \lim_{\tau \to \infty} \tau^{-1} \Pi(t/\tau) \tag{2.12}$$

where $\tau^{-1}\Pi(t/\tau)$ is an unit area rectangle function with a height of τ^{-1} and a width of τ. To see that we can reformulate this problem as a convolution, we can employ the sifting property to compute the value of a function within the integral

$$\int_a^b f(t)\delta(t-\tau)dt = \begin{cases} f(t_0), & \text{if } a \leq t_0 \leq b. \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

for any $f(t)$ continuous at finite $t = t_0$. Thus any input signal, $m(t)$, can be rewritten as a summation of delta functions

$$m(t) = \int_{-\infty}^{+\infty} m(\tau)\delta(t-\tau)d\tau \tag{2.14}$$

and the general linear system response $b(t)$ to the input $m(t)$ can be written as

$$b(t) = G\left[\int_{-\infty}^{+\infty} m(\tau)\delta(t-\tau)d\tau\right] \tag{2.15}$$

and by rewriting the integral as a limit of quadrature sum of infinitesimal areas of area Δt as Δt goes to zero

$$b(t) = G\left[\lim_{\Delta\tau\to 0}\sum_{n=-\infty}^{\infty} m(\tau_n)\delta(t-\tau_n)\Delta\tau\right] \tag{2.16}$$

Because G is linear, we can move it inside the summation (superimposition property) and factor out $m(\tau_n)$ obtaining

$$b(t) = \lim_{\Delta\tau\to 0}\sum_{n=-\infty}^{\infty} m(\tau_n)G[\delta(t-\tau_n)]\Delta\tau \tag{2.17}$$

which, taking the limit, becomes Eq. 2.9. Hence, the inversion of a Fourier transform is an IFK and the known relationship between convolution and the Fourier transform exists

$$G(f) = F[g(t)] = \int_{-\infty}^{+\infty} g(t)e^{-i2\pi ft}dt \tag{2.18}$$

and its inverse operation

$$g(t) = F^{-1}[G(t)] = \int_{-\infty}^{+\infty} G(f)e^{i2\pi ft}df \tag{2.19}$$

where F is the Fourier operator and F^{-1} is its inverse. An important property of Fourier transforms which we will use later is the *Convolution Theorem* which states the convolution of two functions in the time domain (or image domain in case of 2D and 3D data) equals the multiplication of their Fourier transforms in the frequency domain (also known as uv space for radio interferometers).

$$F[m(t)*g(t)] = \int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty} m(t)g(t-\tau)d\tau\right)e^{-i2\pi ft}dt \tag{2.20}$$

which, introducing the change of variables $\xi = t-\tau$, can be demonstrated as follows

$$
\begin{aligned}
&F[m(t) * g(t)] = \textstyle\int_{-\infty}^{+\infty} m(t)\left(g(t-\tau)e^{-i2\pi f t}dt\right)d\tau \\
&= \textstyle\int_{-\infty}^{+\infty} m(t)\left(g(\xi)e^{-i2\pi(\xi+\tau)}d\xi\right)d\tau \\
&= \left(\textstyle\int_{-\infty}^{+\infty} m(\tau)e^{-i2\pi f\tau}d\tau\right)\left(\textstyle\int_{-\infty}^{+\infty} g(\xi)e^{-i2\pi f\xi}d\xi\right) \\
&= M(f)G(f)
\end{aligned}
\tag{2.21}
$$

By utilising the *convolution theorem* and by denoting with the capital letter the Fourier Transform of a function, from Eq. 2.1 and Eq. 2.7, we get:

$$B(f) = G(f)M(f) \tag{2.22}$$

where $G(f)$ is known as the transfer function Transfer Function (TF) and describes the behaviour of the system in Fourier space. If the TF is zero outside a bounded domain (i.e. the operator is bounded), then the PSF is said to be band-limited (and thus well posed), but the addition of noise (which is not band-limited) and the limitation of the observing instruments (which make the uncertainty outside the physical bound of the detector impossible to tackle) make these problems ill-posed.

2.1.1 Classes of Inverse Problems in Astrophysics

As was already mentioned, given the nature of most telescopes, most data processing problems in Astrophysics can be reformulated as linear inverse problems

$$Y = HX \bullet N \tag{2.23}$$

where Y is a set of measurements, N is the unknown noise term and the operator $\bullet$ encodes the way noise contaminates the data. The noise term can be either a stochastic measurement noise induced by the sensor or a deterministic effect due, for example, to an imperfect forward model. X is the solution of the problem, and H is the linear operator. There are many such inverse problems in Astrophysics. I shall shortly describe the most relevant ones for the purposes of the present work:

1. Deconvolution: if the observing system is linear and shift-invariant, the relation between the data Y and the unknown signal X is a convolution with a blurring kernel also known as the Point Spread Function of the system (PSF);

2. Radio-Interferometric Image Reconstruction: this is a special case of a deconvolution problem in which the observed data correspond to a subset of the Fourier components of the unknown X. In this case, the PSF, called *dirty beam*, is not compact in direct space and contains many zeroes in the Fourier domain;

3. "Blind" Source Separation (also known as Signal Detection, Source Detection or Image Segmentation): in this case, each measurement is the linear mixture of n source processes. As each measurement is a different mixture, source separation techniques aim at recovering the original sources by utilising information on how the signals are mixed in the observed data.

As we have seen previously if proper conditions are not ensured, these problems are ill-posed and regularization is necessary in order to find a solution. Regularization involves reformulating the problem by taking into consideration all available information both on the observation process and the signal of interest (a priori knowledge such as non-negativity in images). Several regularization techniques have been developed but all are based on the principle that the best transformation should be the simplest which can describe the forward model. In the following Sections, first, we give a description of the Radio-Interferometric Image Reconstruction problem, which is the main focus of this Thesis, and then traditional strategies for the resolution of the problem are presented.

2.2 An introduction to Radio Astronomy and the Radio Interferometric Image Reconstruction problem

Radio Astronomy is the study of astronomical objects within the radio regime of the electromagnetic spectrum of light. While a unique and exact definition of what constitutes the radio regime is not formally defined, radio astronomical surveys have been observing the sky between 10 MHz and 1 THz. This range is not arbitrarily defined, but it is bound by the physics of the interaction between the radio waves and the Earth's atmosphere. All waves with frequencies between 30 MHz and 50 GHz can penetrate the atmosphere without undergoing any particular interaction, while the atmosphere reflects all radio waves below its characteristic plasma frequency of 10 MHz, and water vapour shows a significant absorption signature in the radio above 50 GHz in the troposphere. While the reflection at low frequencies could only be avoided by removing the atmosphere, i.e. by observing in space, the absorption at high frequencies is drastically reduced by building telescopes in particularly dry places on Earth, such as the Atacama desert in Chile where the Atacama Large Millimeter/submillimeter Array (ALMA) is operated. This broad range of observational frequencies allows for the study of a large variety of astronomical sources thanks to the wide range of physical mechanisms which emit in that range: synchrotron radiation, atom and molecular transitions, black body radiation, free-free radiation and inverse Compton scattering. This allows the investigation of properties of a large variety of objects: radio galaxies, active galactic nuclei, quasars, inter-galactic and interstellar medium, the centre of the Milky Way, supernova remnants, supernovas, black holes, stars, planets and even extra-terrestrial intelligence. This simple list clearly shows how relevant is to optimise data reduction and analysis in the radio regime. Radio waves are received by both dipole antennas and dishes, but the majority of modern radio telescopes are built in the form of parabolic antennas (*dishes*) which can directly focus the radiation on the receiver and can be used more efficiently on the wide frequency ranges in which radio telescopes operate. A dish is an electromagnetic transductor which transforms the incoming electromagnetic waves into electric signals that can be recovered and processed. The area of the sky to which the antenna is sensitive is called the field of view Field of View (FoV) and, given the geometry of the antenna and the observing wavelength, an antenna is characterised by a directional gain pattern (or beam) which describes the variation of the antenna response to the electromagnetic waves across the FoV. The beam pattern of a dish is constituted by the main lobe, which defines the angular resolution or resolution power of the antenna, and by several side lobes created by various types of interferences between the electromagnetic waves incoming to the dish [17]. At a given wavelength λ, the resolution power of a dish, i.e. the smallest angle θ under which two point sources (sources which are so distant from the observer that can be approximated as points without loss of generability) are detected as separate entities, is connected to the diameter D of the dish through the Rayleigh criterion:

$$\theta = 1.22\frac{\lambda}{D} \tag{2.24}$$

Where θ is in radians. This equation tells us that the wider is the dish, the higher it is the resolution power but, as can be seen, to achieve resolving powers similar to optical telescopes (0.01 arcsec), impossibly large dishes are required. For example, to resolve a source emitting at a wavelength of 1 meter with the resolution of 1 arcsecond, a dish with a diameter of 251 km would be required. At the time of writing this thesis, the largest single-dish radio telescope in the world is the Five-hundred-meter Aperture Spherical Telescope Five hundred meter Aperture Spherical Telescope (FAST) in China with an astonishing diameter of 500 m.

Since the beginning of the sixties, Astronomers have started to combine several radio dishes arranged in such a way as to replicate a single instrument through the correlation of the electromagnetic signal in a pairwise fashion. The instrument is called an interferometer and the imaging technique is called aperture synthesis. Each antenna observes simultaneously the same source at the same frequency ν. Since each wave takes a slightly different path to reach each antenna, the difference in distance travelled creates a phase difference between them which creates an interferometric pattern on the receiver or, in other words, a measurable geometric delay in the arrival time of the wave at each antenna. This delay is connected to the distance between each pair of antennas, which is called the *baseline length* B, and the resolving power of an interferometer, in a first approximation, is obtained by substituting D with B in Eq. 2.24. After performing the delay compensation on each baseline signal, the signals are averaged together in time and frequency and then stored. Each baseline produces a fringe interference pattern multiplied by the beam pattern to form the instrument Point Spread Function (PSF). Interferometers, with respect to single-dish radio telescopes, can vary the baseline lengths between antennas by moving them over rail tracks. Larger baselines allow the

Figure 2.1. A spectacular bird-view of the FAST Radio Telescope in Pingtang county, China. The telescope observes the radio sky between 70 MHz and 4 GHz with an angular resolution of 4 arcseconds.

observation of compact structures, while smaller baselines, the observation of extended sources. To understand how an interferometric measurement is made, let us consider a set of two antennas and an emitting source so far away from the two antennas that the emitted waves can be approximated as planar. The electric signal received by the antennas at the frequency ν are, respectively:

$$E_1(t) = E_0(t, \hat{x}) exp \left[2\pi i (t\nu + \vec{x} \cdot \vec{k}) \right] \tag{2.25}$$

$$E_2(t) = E_0(t, \hat{x}) exp \left[2\pi i (t\nu + (\vec{x} - \Delta\vec{x}) \cdot \vec{k}) \right] \tag{2.26}$$

where $\vec{x}$ is the source position, $\Delta\vec{x}$ is the difference in path length between the two antennas, and $\vec{k}$ is the wave vector which is equal for both the antennas given the planar approximation of the wavefront. Since the whole field of view of the antennas contributes to the collected signal, we can integrate over all directions $\hat{x} = \vec{x}/|\vec{x}|$

$$E_1(t) = \int d^2\hat{x} E_0(tm\hat{x}) exp \left[2\pi i (t\nu + \vec{x} \cdot \frac{\hat{x}}{\lambda}) \right] \tag{2.27}$$

$$E_2(t) = \int d^2\hat{x} E_0(tm\hat{x}) exp \left[2\pi i (t\nu + (\vec{x} - \Delta\vec{x}) \cdot \frac{\hat{x}}{\lambda}) \right] \tag{2.28}$$

The two signals are then correlated by averaging their product. The resulting quantity is usually called the visibility pair W_{12}

$$W_{12} = \frac{1}{T} \int dt E_1^{-}(t) E_2(t) \tag{2.29}$$

where the bar indicates that we have taken the complex conjugate of the vector. If we assume that the incoming radiation is uncorrelated (which is true for all astronomical sources), then we can simplify the integral as follows:

$$W_{12} = \frac{1}{T} \int dt E_1(t) \bar{E}_2(t) \simeq \delta(\hat{x} - \hat{y}) |E_0(\hat{x})|^2 \tag{2.30}$$

The correlated signal can, then, be written in the form

$$W(\vec{u}) = \int I(\hat{x}) exp[2\pi i \vec{u} \cdot \hat{x}] d^2\hat{x} \tag{2.31}$$

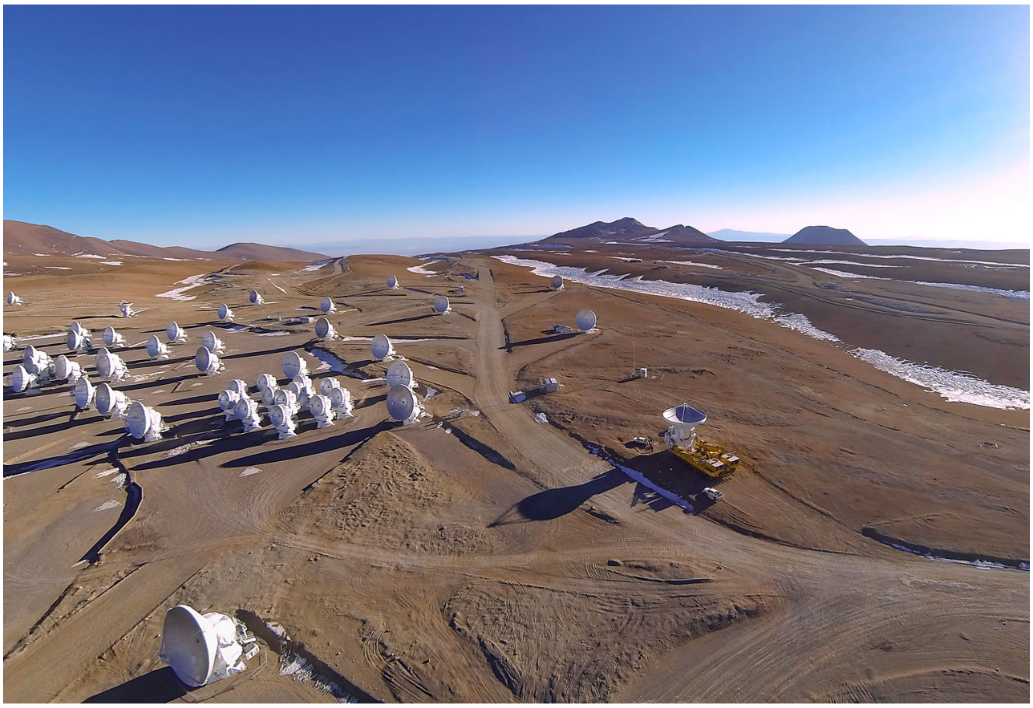

Figure 2.2. The Atacama Large Millimeter / submillimeter Array (ALMA) (Chile) (image courtesy of the European Southern Observatory (ESO))

where $\vec{u} = \frac{\Delta x}{\lambda}$ is the displacement vector between the two antennas divided by the observing wavelength, and $I(\hat{x}) = |E_0(\hat{x})|^2$ is the intensity of the sky. Equation 2.31 is known as the radio interferometry measurement equation and the space on which the visibilities are defined, it is called the *visibility space*.

The equation resembles a spherical Fourier transforms, and we already know from Sec. 2.1 that inverting it implies the resolution of an IFK. Given that the intensities are only measured at the Fourier-like vectors $\vec{u}$ which corresponds to the antenna's displacements, the visibility space is only partially sampled. Moreover, the dirty beam is not compact in the image space and thus solving this equation constitutes an ill-posed deconvolution problem. In order to solve this problem, in practice, a way of filling the visibility space must be found. The main difference between an interferometer and a single-dish detector is that in the latter case the visibility space is continuous and not discrete. Since the dishes cannot take arbitrarily close positions, there are usually "gaps" in various positions of the visibility space which are partially filled by changing the configuration of the array and by utilising the rotation of the Earth to sweep a higher fraction of the visibility space. One could be content with not trying to invert Eq. 2.31, but the problem is that interferometers do not directly produce images of the sky but instead they measure the brightness distribution or intensity of the sky at a given frequency I_ν for a given angular resolution in Fourier space (visibilities), and an image can be obtained after performing a Fourier transform on the sampled visibilities.

In the case that all baselines can be reduced to a plane, the u, v plane, then the theory of aperture synthesis (see the convolution theorem in Sec. 2.1) states that the map obtained by taking the Fourier transform of the measured visibilities $W(u, v)$ is proportional to the convolution of the true brightness distribution T_b (convolved with the primary beam of the P antennas configuration) and the synthetized beam G:

$$P(l, m) * T(l, m) * G(l, m) \propto \int\int W(u, v)g(u, v)e^{2\pi i(ul+vm)}dudv \quad (2.32)$$

where l and m are the sky coordinates in the image plane (directional cosines with respect to the u and v axes) and $g(u, v)$ is the function used to weight the complex measured visibilities $W(u, v)$. u and v are usually expressed in wavelenghts. We call *synthesized beam* or *dirty beam* the normalized Fourier transform of this weighting function

$$G(l, m) = C_n \int\int g(u, v)e^{2\pi i(ul+vm)}dudv \quad (2.33)$$

where C_n is a suitable normalization function. Even if we assume, that all visibilities lie on a plane, in order to calculate the Fourier integral in Eq. 2.2 the product $W \times g$ must be known for all baselines. However, due to the gaps in the visibility space, the correlation function $W(u, v)$ is only known for the positions at which it has been measured. A simple way to fill the gaps is to set, for all positions in which W is not measured, $g = 0$. This weighting scheme, however, leads to the creation of spurious patterns in the produced image. The image resulting from the Fourier inversion is called the *principal solution* or *dirty map*, i.e. a member of an infinite family of possible solutions which all are in agreement with the measurements. The sidelobes cause oscillations in the principal solution around strong sources and sharp edges which make recognising weak features in the image a very difficult task. The *dirty image* or *dirty map* is thus the convolution of the true sky brightness distribution with the instrument PSF or *dirty beam* and for the aforementioned reasons it is not usable for scientific purposes (i.e. to study the properties of the astrophysical sources within it) and a series of steps are applied to the image in order to remove as much as possible the instrumental response. Several algorithms have been developed over the years to solve this ill-posed interferometric deconvolution problem, i.e. to recover the true sky distribution from the dirty map. To reduce the effect of sidelobes and high-frequency noise, one could think of convolving this map with a suitable smoothing function such as a Gaussian. Unfortunately, filtering cuts high-frequency components making faint features (or faint sources) detection in the image more difficult if not plainly impossible. Furthermore, given that convolution is a linear operation, it cannot generate nonzero values at unmeasured spatial frequencies, nor can extrapolate into noise-contaminated regions of the visibility space. For these reasons, the community has developed several nonlinear methods which use prior information about the data to fill the gaps in the visibility plane with non-zero values. Solving the linear system of equations $\overrightarrow{V} = [F][S_{dd}]\overrightarrow{T}$ where $\overrightarrow{V}$ are the calibrated visibilities, F represent a 2D Fourier transform, S_{dd} represents a 2D spatial frequency sampling function that can include direction dependent instrumental effects such as the primary beam, and $\overrightarrow{(}T)$ are a list of parameters that model the sky brightness (for example the image pixels), given the ill-posedness of the problem, forces the algorithms to be iterative, which necessarily involves repeated passages between the image and Fourier space. In fact, in order to constraint solutions, whatever reconstruction is obtained in the image domain, can be transformed back into the Fourier domain to measure the agreement with the observations through a single constraint statistical metric (usually chi-squared). Through the minimisation of this metric, the optimal solution may be found. Iterations begin with an initial guess for the image model, each major cycle consists of the prediction of model visibilities, the calculation of residual visibilities and the construction of a residual image. This residual image contains the effects of the incomplete sampling of the spatial-frequency plane but is otherwise normalized to the correct sky flux units. The minor cycle is to iteratively build up a model of the true sky by separating it from the point spread function. This step is also called deconvolution and it is equivalent to the process of solving the normal equations as part of a least squares solution [42]. Different reconstruction algorithms can operate as minor cycle iterations, allowing for flexibility in (for example) how the sky brightness is parameterised.

2.3 Source Detection and Deconvolution Algorithms in Radio Astronomy

The detection of sources in Astronomical images, i.e. the task of separating pixels belonging to the sources from those belonging to the background (a task also known as image segmentation), is a crucial step for the extraction of scientific results from astronomical observations and it is one of the obligatory steps for the measurement of their properties. While the task of separating luminous spots from a much darker and noisy background may seem trivial with respect to other problems in computer vision, reality shows that it is not simple at all. Astronomical objects have not clearly defined boundaries since: their surface brightness steeply decreases outwards and quickly become fainter than the noisy background (hence the signal is heavily contaminated), their size and intensity may vary greatly resulting in an image with a large dynamic range (the ratio between the brightest and dimmest pixels within the image) and a large spatial dynamic range (the ratio between the largest and the smallest detectable structure in the image). Given the ever-growing size and rate of data coming from astronomical instruments, the astronomical community has devoted much time to the development of automatic source detection algorithms. These usually employ knowledge about the data and a likelihood or loss function to measure the probability that the predicted model of the underlying emission has produced the detected image. Once this is set, and the non-negativity of the emission is imposed, the problem is reformulated as a classical Least Squares problem and the *general solution* is found through gradient descent. Classical source detection algorithms in Astronomy can be broadly categorised into three main categories:

2.3.1 The Bayesian approach

Algorithms that fall under the Bayesian approach are based on the framework of Information Field Theory [45] and employ Bayesian inference where a set of observations is used to update the probability that the models representing them are true. In other words, it tries to estimate the values of a set of parameters θ in some reasonable model of the data b, which in our case are images. To make this estimation, the likelihood i.e. an expression of the probability that any given model may provide the observed data under a certain parametric configuration is considered. Also, it needs to impose a prior probability on the parameters based on some (prior) knowledge about their values. The approach itself then consists in constructing the conditional probability density function:

$$p(\theta|I) = \frac{p(I|\theta)p(\theta)}{P(I)} \tag{2.34}$$

which gives the posterior distribution $p(\theta|I)$ in terms of the likelihood $p(I|\theta)$, the prior $p(\theta)$ and the evidence $p(I)$. The evidence is usually set to a constant parameter which results in an un-normalised posterior distribution. The maximum a posteriori Maximum A Posteriori (MAP) method, searches for the best solution by maximising the likelihood so that, under the assumed statistical model, the observed data is most probable. In practice, the likelihood is often based on exponential models which account for the signal and the noise model (Gaussian, Poisson, etc.). Known examples of this kind of algorithm are [53, 80];

2.3.2 The Matched Filtering and Thresholding approach

Algorithms that fall under this approach use convolving kernels to highlight sources and suppress background fluctuation and noise. Modern implementations of such algorithms usually smooth the data with several kernels with variegated shapes and resolutions and after employing a threshold, based on the dynamic range of the convolved image, select source pixels. A known and widely used algorithm in Radio Interferometry which utilises this approach is SOFIA [135] which is a state-of-the-art flexible line finding algorithm capable of detecting and parameterising HI sources within 3D radio data cubes. It uses the *Smooth and Clip* algorithm

Smooth and Clip (S + C), to detect meaningful emission in the cube by convolving it with 3D kernels specified by the user at multiple angular and velocity resolutions. At each resolution, voxels are detected if their absolute value is above a threshold given by the user (in noise units). The final mask is the union of the masks constructed at various resolutions. The algorithm was updated and renamed Sofia-2 [159], rewritten in the C programming language while making use of OpenMP for multi-threading of the most time-critical algorithms. Sofia-2 is substantially faster and comes with a much-reduced memory footprint compared to its predecessor. Another widely used algorithm is BlobCat[57] which employs the flood fill algorithm to detect and catalogue islands of pixels representing sources in 2D astronomical images. The algorithm determines the Root Mean Squared Root Mean Square (RMS) noise at each spatial pixel in an image by extracting the distribution of pixel values within a local mesh, iteratively clipping the most deviant values until convergence is reached at 3σ about the median. The choice of mesh size (in pixels) is critical for successful source retrieval and thus it is a parameter that the user needs to fine-tune. If it is too small, the local RMS estimate may be biased due to the lack of statistically independent measurements or overestimated due to the presence of real sources. If it is too large, any true small-scale variations in local RMS noise may be washed out. The RMS of the noise is utilised to create a signal-to-noise rate (SNR) map of the image and two threshold values: detection (T_d) and cut (T_f), are used to decide which peaks in the image are good candidates for blobs and where to cut the blobs boundaries around them (in other words: pixels with an SNR higher than T_f are selected to form islands and island boundaries are defined by T_d);

2.3.3 Compressed Sensing and the Multi Scale Approach

A signal, considered as a vector in $\mathbb{R}^{\mathbb{N}}$, is *sparse* if most of its entries are equal to zero. If the number of zero entries is k with $k << N$, then the signal is said to be *k-sparse*. In the case where only a selected few entries to have large values and the rest is close to zero, the signal is said to be weakly sparse or compressible. Generally, signals are not compressible but can be made so by suitable transformations. For example, the Fourier transform of a sinusoidal signal is certainly 1-sparse in the Fourier domain, but it is not sparse at all in the time domain. If a signal is sparse, then it can be represented as a linear expansion

$$X = \Psi\alpha = \sum_{i=0}^{T} \psi_i \alpha[i] \tag{2.35}$$

where $\alpha[i]$ are the coefficients of X, Ψ is the dictionary, and ψ_i are the atoms of the dictionary, i.e. the elementary waveforms in which the signal is decomposed. When no constrain on the dictionary or the coefficients are set, the problem amounts to computing the singular value decomposition of X or, equivalently, the diagonalisation of the variance matrix of X. As we have seen (see Sec. 2.1), this route is most of the time unfeasible, and the problem may be simplified by adding priors or constraints on the dictionary Ψ [100], on the coefficients α or on both. For example, an l_0 or l_1 norm penalty on the coefficients yields a sparse representation of the data. Techniques that employ sparsity are based on the concept that the optimal dictionary is the one in which the function has the sparsest representation, and Compressed Sensing is a framework of techniques that search for that solution. In practice, the dictionary is a $N \times T$ matrix whose columns are the normalised coefficient vectors. For example, these could be normalised to a unit l_2-norm, i.e.

$$\forall i \in [1, T], \parallel \phi_i \parallel_2^2 = \sum_{n=1}^{N} |\phi_i[n]|^2 = 1 \tag{2.36}$$

One could try and use a huge over-complete dictionary ($T >> N$) but this would lead to prohibitive computational times for calculating the coefficients. A balance thus must be struck between the complexity of the dictionary (size of the dictionary) and computational times. In a more general treatment, we can define for a vector X his l_p-norm as $|X|_p^p = \sum_i |x_i|^p$ for $p > 0$, and by setting $p = 0$ we get the l_0 pseudo norm which counts the number of non-zero entries in the vector X. A l_0 regularised problem amounts to the minimisation of

$$\tilde{\alpha} \in \operatorname{argmin}_\alpha \parallel Y - H\Psi\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_0 \tag{2.37}$$

where λ is the regularization parameter. Regardless of the imposed constraints, Eq. 2.37 is a known NP-complete problem [139], but can be relaxed by substituting the l_0-norm with an l_1-norm and thus reformulating the minimisation problem as

$$\tilde{\alpha} \in \text{argmin}_{\alpha} \parallel Y - H\Psi\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_1 \tag{2.38}$$

This latter problem can be solved under appropriate transformations and usually, for many Ψ it is shift-invariant (a set of convolutions). Unfortunately, the resolution of the problem with this formulation does not admit a closed-form solution and, therefore as already explained, it must be solved iteratively. Closely related to this problem is that of reducing dimensionality [116] of data with minimal information loss which is important for feature extraction, compact coding and computational efficiency, to eliminate redundancies and enforce constraints. Given the interest in finding methods capable of solving the inverse problem, other constraints have been imposed and the resulting techniques usually take their name from them: sparse Principal Component Analysis Principal Component Analysis (PCA) [39] when the sparsity constraint is also imposed on the dictionary and not only the coefficients; independent component analysis (ICA [74]) when the statistical Independence between the coefficients is imposed; non-negative matrix factorization (NMF [88]) when a positivity constraint is imposed on both the coefficients and the dictionary. The third approach to dictionary learning more similar to that of ML, is that of Wasserstein dictionary learning [134] Wasserstein Dictionary Learning (WDL) in the sense that it generates a database of dictionaries Ψ and a set of associated weights to those dictionaries Λ. A set of Wasserstein barycenters are generated and an operator P is set to approximate X as $P(\Psi, \Lambda)$. The operator maps the atoms of the dictionaries and their weights to the barycenters and the optimal solution is found by searching for minima in a nonconvex energy function. This is achieved by performing automatic differentiation on the iterative scheme used to compute the barycenters so that gradients with respect to both the dictionary atoms and the weights can be obtained. Weights and dictionaries can be then updated through a quasi-Newtonian solver.

The Starlet Transform and the Multi-Vision Model

Astronomical data generally have a complex hierarchical structure, and for this reason, a more suitable way to represent it is in the multi-scale space. Thus, images are decomposed into components at different scales (different spatial frequencies), and objects become highlighted in some scales. Depending on the nature of the sources, they may appear in several scales, and closer to low or high-frequency scales. Once the decomposition is complete, a basic detection algorithm can be applied to the different scales, as if they were single-scale images. The Starlet wavelet transform [143] decomposes an image into a set of coefficients as a superimposition:

$$c_0[k,l] = c_j[k,l] + \sum_{j=1}^{J} w_j[k,l] \tag{2.39}$$

where c_0 is the original image, c_j is a smoothed version of the original image and w_j capture details of the image at spatial scales of 2^{-j}. Thus the algorithm outputs $J+1$ coefficient maps for each input image. The decomposition is achieved using a filter bank (h_{2D}, $g_{2D} = \delta - h_{2D}$, $\tilde{h_{2D}} = \delta$, $\tilde{g_{2D}} = \delta$) where h_{2D} is the tensor product of two 1D filters. The passage from one resolution to the next is obtained using the "a trous" (with holes) algorithm [143]

$$c_{j+1}[k,l] = \sum_m \sum_n h_{1D}[m] h_{1D}[n] c_j[k + 2^j m, l + 2^j n] \tag{2.40}$$

$$w_{j+1}[k,l] = c_j[k,l] - c_{j+1}[k,l] \tag{2.41}$$

B_3-spline is chosen for the scaling function. The algorithm goes as follows:

1. initialize j to 0 and be k the number of pixels belonging to the signal;
2. carry out a discrete convolution of the data $c_{j,k}$ using a filter h. The distance between the central pixel and the adjacent ones in 2^j;
3. after this smoothing, the discrete wavelet transform is obtained from the difference $c_{j,k} - c_{j+1,k}$;
4. if j is less then the maximum number of resolutions J, the algorithm increments j and goes back to step 2;

5. the set $W = w_1, ...w_J, c_J$ represents the wavelet transform of the data.

Using sparse modelling, we can consider the observed signal X as a linear combination of a few atoms of the wavelet dictionary $\Psi = [\psi_1, ..., \psi_T]$ from which different filter banks or kernels can be created. By modelling the deconvolution (or source detection) problem 2.23 as follows

$$Y = HX + N = H\Psi\alpha + N \tag{2.42}$$

where $\alpha = w_1, ..., w_J, c_J$, we can define the multi-resolution support of the input image by the following condition

$$M_j[k,l] = \begin{cases} 1, & \text{if } w_j[k,l] > K\sigma_j. \\ 0, & \text{otherwise.} \end{cases} \tag{2.43}$$

where $k\sigma_j$ is a given detection limit usually estimated through noise modelling. The Multi Vision Model describes an object as a hierarchical set of structures. In fact, at each scale of the multi-resolution support, a set of significant connected wavelet coefficients form structures which can be isolated through a clump find algorithm. At each scale, we can define a *structure* $S_{j,k}$ as the set of significant connected wavelet coefficients at scale j, and an *object* as a set of connected structures. In fact, by checking if the pixels belonging to different structures at different scales have significant overlap (they share common pixels), a link between two structures can be formed if their overlap is higher than a given threshold. The co-addition of all the reconstructed objects is a filtered version of the input data. If we denote with α_i the set of wavelet coefficients belonging to a given object in the image, it is clear that they are a subset of the full set of coefficients $\tilde{\alpha_i}$ of the wavelet transform of X_i, $\tilde{\alpha_i} = WX_i$. Given that it is possible that not all coefficients belonging to X_i have been detected, the image reconstruction problem consists in searching for the image X_i such that its wavelet transform reproduces the observed coefficients α_i. If we set W as the wavelet transform operator, and P_w as the projection operator in the subspace of the detected coefficients (i.e. the thresholding operator that let us select and preserve relevant connected coefficients and set to zero all the rest), the solution of this problem can be found in the least square

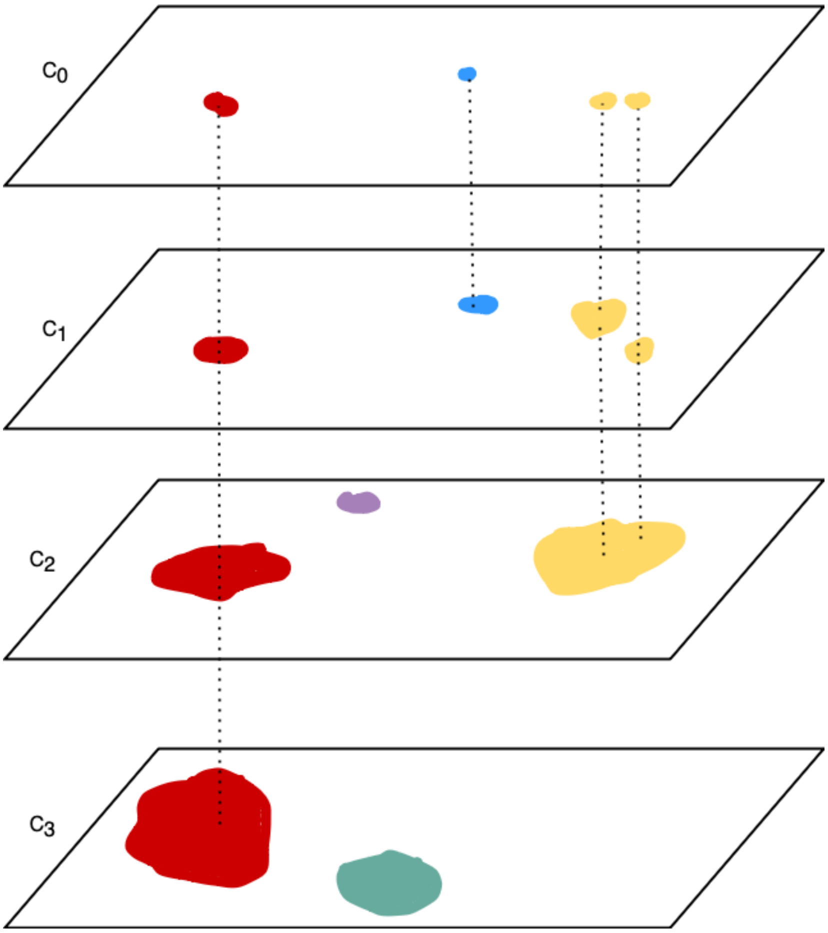

Figure 2.3. Illustration of the Multi Vision Model linking process between detected objects at several coefficient scales. Links (dotted lines) are created if the spatial overlap is registered between objects.

fashion by setting a positive constrain on the solution

$$\min_{X_i} ||\alpha_i - P_w(WHX_i)||^2 \tag{2.44}$$

In particular, this can be solved with the Van Citter method [142] which leads to the following iterative scheme:

$$X_i^{(n+1)} = X_i^n + R\Big(\alpha_i - P_w(WHX_i^{(n)})) \tag{2.45}$$

where R is the inverse wavelet transform.

1. set $n = 0$;

2. find the initial estimation $X_i^{(n)}$ by applying an inverse wavelet transform to the set α_i corresponding to the detected wavelet coefficients in the data;

3. convolve $X_i^{(n)}$ with the PSF $H : Y_i^{(n)} = HX_i^{(n)}$;

4. determine the wavelet transform $\alpha^{(Y_i^{(n)})}$ or $Y_i^{(n)}$;

5. threshold all wavelet coefficients in $\alpha^{(Y_i^{(n)})}$ at positions and scales where nothing has been detected (P_w). We get $\alpha_t^{(Y_i^{(n)})}$;

6. determine the residuals $\alpha_r = \alpha_i - \alpha_t$;

7. reconstruct the residual image $R^{(n)}$ by applying an inverse wavelet transform.;

8. add the residual to the solution $X_i^{(n+1)} = X_i^{(n)} + R^{(n)}$;

9. threshold negative values in $X_i^{(n+1)}$;

10. if $\sigma(R^{(n)})/\sigma(X_i^{(0)}) < \epsilon$, then $n = n + 1$ and go to step 3;

11. $X_i^{(n+1)}$ contains the deconvolved reconstructed objects.

2.3.4 CLEAN

In 1973, Hogbom [70] developed the CLEAN algorithm, a procedure to solve the deconvolution problem, and since then, for more than 40 years, through constant updates, it has been the standard data cleaning procedure for most radio interferometers. The algorithm is an iterative procedure that operates in the image plane using the known information about the dirty beam, to separate real structures (the underlying astrophysical signal) from spurious structures created by the interferometer and noise (sidelobes and artefacts). There are several variants of the CLEAN algorithm, some operate with delta functions, others with multi-scale sky models, but all parametrise the sky brightness in a sparse basis. The *dirty beam* described before contains both positive and negative sidelobes. These areas of negative values cannot be real brightness distributions given that real brightness distributions must be positive definite and thus the difference between the observed map and the true sky brightness can be reduced by substituting the negative pixels with zeros. The Fourier Transform of the resulting map will no longer agree with the known values of W measured at the baseline but the agreement can be restored by adding Fourier components that make up for the difference. This operation will produce new areas of negative intensities which can be processed in the same way until no change is perceived between one iteration and the next. In order to understand the main working principle of CLEAN, let's assume the very simple example of an ideal noiseless observation of a point source situated in the centre of the beam. An empty sky image could be obtained by subtracting from the image the dirty beam scaled for the measured amplitude at the centre of the beam. The convolution of the obtained empty sky map with a clean beam, i.e. an ideal dirty beam without the presence of sidelobes, should provide an image which should be equal to that obtained by applying the inverse Fourier transform to a completely covered visibility space. In order to restrict the number of solutions, the algorithm makes the assumption that any significant correlation (higher than the set noise threshold) between the dirty map (DM) and the dirty beam (DB) will be caused by a source or source component at the position of the maximum correlation. And the convolution function is given by the convolution of the dirty beam with the dirty image. It follows from the convolution theorem that

$$DM \xleftarrow{FT} WgDB \xleftarrow{FT} gDM * DB \xleftarrow{FT} Wg^2 \tag{2.46}$$

If we have set $g = 1$ at all measured baselines and zero elsewhere then, given the complex nature of the visibilities, Wg^2 will be equal to Wg and thus $DM * DB$ must be equal to DM and thus the dirty map is by itself the correlation functions. By searching for maxima within the dirty map, sources can be detected. CLEAN starts by making a full analysis of the map and assumes that all meaningful peaks in the image must be positive. Negative peaks are used to gauge the noise level in the image. At each iteration, the algorithm searches for the maximum positive peak in the image and subtracts from it a fraction of the dirty beam at the location of the peak. The resulting image is called the *residual map*, which in the most optimal outcome should contain only background emission. The algorithm jointly fills another image (called the CLEAN component map) with the *clean components*, i.e. delta functions located at the position of the detected peaks. The algorithm stops when no more peaks pass the noise threshold criterion or when the maximum number of iterations is reached. The CLEAN component map is then convolved with the CLEAN beam and added to the *residual map* to form the restored image. Over the years many improvements have been made to the CLEAN algorithm to improve its speed, reliability and accuracy of the source characteristic measured from the restored image. Clark [32] published an updated version of Hogborn's algorithm, with two main differences from the predecessor: it performs residual updates only on a small patch of the PSF (an operation that speeds up the minor cycle but may introduce aliasing errors; it controls aliasing errors by transitioning to the major cycle when the peak residual reaches the level of the highest sidelobe for the strongest feature.

One of the latest versions of CLEAN is the Multi-Scale Multi-Frequency Synthesis (MS-MFS) [34] algorithm which is an algorithm that models the sky distribution as a collection of inverted, tapered paraboloids of different scales and sizes and whose amplitude follows a polynomial in frequency. A linear-least squares approach is used along with the standard clean-type iterations to best fit spectral and spatial parameters. It accounts for extended source characteristics at multiple spatial frequencies within the image. To conclude, the CLEAN algorithm is an L1-norm basis-pursuit method and this means that in a crowded field of point sources, especially with a PSF with high sidelobes, the algorithm is more error-prone in the low SNR regime.

2.3.5 CLEAN Limitations

So far in our discussion, we have only talked about images, but as previously stated, radio interferometers can be used to detect signals over a wide range of frequencies. When an interferometer is pointed towards a sky sources source, it usually makes observations at several contiguous frequencies (frequency channels). The step in frequency that separates one image from the next is dictated by the frequency resolution (or sensitivity) of the interferometer. The resulting data product is thus a cube characterized by two spatial axes and a frequency axis. Because radio sources usually emit at several wavelengths in the radio regime, their signals in the cube show correlation in both the frequency and spatial domain. CLEAN was developed to work on images and thus does not take into consideration correlations along the frequency axis. Without leveraging the correlation in frequency, the algorithm has no mean of separating a source (peak in the image) from a particularly high noise spike (which should not show a correlation in frequency) and, even more problematic, the cleaning process makes the separation in the restored image even more difficult, given that a noise peak, if mistaken for a source, would be deconvolved several times with the instrumental PSF and the recovered delta function would be convolved with the *clean beam* (i.e. a Gaussian approximation of the PSF) thus producing a structure morphologically similar to the actual sources that underwent the same iterative deconvolution process. Another issue with the usage of CLEAN on data cubes is its speed which is affected by the fact that each image (slice of the cube) is processed independently and no information about previous slides is used.

2.4 An introduction to Astrometry and the TOLIMAN Deconvolution problem

Among the many questions that Astronomy seeks to answer, like many hard sciences, it also tries to answer our deepest questions. Where did it all begin and how is it going to end? Are we alone in the Universe? Is there life beyond our biosphere – or conversely is Earth and our planetary system in some way unique? Such inquiries have given rise to the fields of astrobiology and exoplanetary research. Despite our long-term commitment to exploring these questions, the development of instruments capable of detecting planets around distant stars has proven to be one of the most challenging astronomical quests [87]. The first exoplanet orbiting a Sun-like star was detected through small deviations caused in radial velocity measurements of its host [104], a pioneering work that was subsequently awarded the 2019 Nobel Prize in Physics. A little more than twenty years later, there are more than 4000 confirmed exoplanets[1]. The celestial garden is therefore a fertile ground for discovery, and the synergy between new astronomical missions and modern statistical and machine-learning techniques promises an exceptionally bright future for this rapidly expanding field. Discovery and characterisation of exoplanets are particularly suited to combinations of approaches that can push the boundaries in both the acquisition of exceptionally clean, low-noise data, as well as the ability to sift large volumes of observations in order to extract subtle signals that are often submerged under orders of magnitude of statistical and systematic noise. Every technology in this area has to face these problems because, on a cosmic scale, exoplanets are almost completely negligible. They contribute only infinitesimally to the mass or energy budget of galaxies. Even in our own solar system, major gas-giant planets such as Neptune and Uranus evaded detection until the advent of the modern telescope; the challenge of discovery at light-year distance scales can seem forbidding. The most successful techniques to reveal exoplanets are *indirect* in that they do not witness signals from the planet itself, but rather the planet's influence on its host star. One is the transit method which witnesses a dip in starlight as the planet traverses the observer's line of sight to the star. An alternative method is through measurements of radial velocity, which records to-and-from perturbations in the velocity of the star, as it is perturbed by the gravitational field of the planet. The TOLIMAN (Telescope for Orbital Locus Interferometric Monitoring of our Astrometric Neighbourhood) [152] program was motivated by the realisation that neither of these methods are suited to answer a fundamental question: are there any

[1] http://exoplanet.eu/catalog/

potentially habitable exoplanets around the Sun's nearest neighbour twin system – α Centauri AB? Unfortunately, the transits require an alignment, a very rare event, while radial velocity can find massive gas-giant planets, but not small rocky exo-Earths in the habitable zone of the system. A very promising alternative method is the most traditional branch of *Astrometry*: the study of deviations in the position of the star in the plane of the celestial sphere that, in this case, are imposed by the motion of the star and the exoplanet around a common centre of mass. Like all signals in this domain of science, the deviations in position are very small, of the order of one micro-arcsecond. To give a sense of scale, for an observer on Earth, this is the angle subtended by a coin held edge-on ($\sim 2\,\mathrm{mm}$) while standing on the moon. For the specific case we are interested in, the situation is even more interesting. α Centauri is a binary star system, with two stars (A and B) orbiting around each other. If their motion could be monitored, for example by taking a series of images at different epochs (or time stamps), one would see the distance between the centres of the stars changing as their orbit evolves. After one orbital period, this pattern would repeat - thus, by observing the separation between the stars during some time one would detect a periodic signal. Given the interactions between orbital bodies in a gravitational bound system such as a solar system, the presence of any significant orbital bodies apart from the two stars would have an effect on their relative motion. For example, the expected signal would be slightly different if one considers the presence or absence of an Earth-like planet as adjoint elements in this system – and that is the type of perturbation which the TOLIMAN mission aims at measuring. One can imagine that at such scales even the smallest deviations in the position of the satellite or thermal effects in its structure and instruments are enough to build up noise in each image, which is orders of magnitude higher than the signal. This makes the problem a classical ill-posed inverse problem where small perturbations in the system can lead to widely different solutions. TOLIMAN has been designed to implement innovative optical principles to deliver a robust estimate of this signal, despite the inevitable presence of many competing random processes and systematic noise. Details can be found in [152] and in Ssec. 2.4.3 later in this Chapter. A critical component for the success of the mission is the clear ability to extract periodic signals at the milliarcsecond level from a data stream consisting of over a million of images downlinked from the satellite.

2.4.1 Astrometry

Astrometric detection involves precise measurements of stellar positions and it is widely regarded as the leading concept presently ready to find Earth-mass planets in temperate orbits around nearby sun-like stars [136, 56, 149, 101]. The principle for detecting a planet using astrometry is the same as that adopted by the hunters of unseen companions of stars [21] about two hundred years ago. As a planet orbits the star, the latter is tugged in a small ellipse around the centre of mass of the system, thus, by careful measurements of the position of the star over time (either in a relative or global frame of reference, that must be more stable than the signal produced by the invisible companion), these tiny displacements yield a solution for the planet mass and orbit. Unlike other methods, like those based on transit timing, photometry or radial velocity measurements, the signal generated by companions increases with planet-star separation, converse to both radial velocity and transit methods, following the equation [51]:

$$S = \frac{m}{M}\frac{r}{d} \tag{2.47}$$

where m and M are respectively the masses of the planet and the companion star, r is the semi-major axis of the planet's orbit and d is the distance between the observer and the star. From this equation it can be seen that the amplitude of the intrinsic signal (the amplitude of the periodic angular wobble on the sky) is inversely proportional to distance, favouring stars in the immediate neighbourhood of the Sun, and scales with the planet-mass ratio, covering thus the search for massive planets around relatively small stars. However, if enough care is taken in the observation stability, several valuable targets of potential close stars are worthy of being observed through astrometric techniques. Despite the potential promise, astrometric detection for exoplanetary discovery has not yet entered the mainstream. The angular excursions induced by habitable-zone Earth-analog planets are small, of the order of one micro-arcsecond even for best-case targets, such as Alpha-Centauri (which is the second closest star to the Sun).

Ground-based high-precision astrometry campaigns must fight considerable sources of noise, such as the starlight path through the Earth's turbulent atmosphere. Long-baseline optical interferometers have historically delivered precisions better than 100 micro-arcseconds, with a recent resurgence of interest prompted by ESO's GRAVITY instrument [7] with accuracy an order of magnitude better, but still not sufficient to detect Earth-mass planets around nearby stars. Furthermore, the nearest stars to Earth present a large apparent angular diameter and are correspondingly difficult to observe on long baselines, since they are over-resolved objects, and thus present challenges to the interferometric technique, due to low fringe contrast. These intrinsic challenges for ground-based astrometric observation have increased the interest in space-borne missions which exchange the aforementioned problems with the challenges connected to satellite observations (inherent instabilities and data transmission constraints). Global, large space astrometric surveys over wide angles have proved to be extremely productive delivering fundamental stellar positions, distances and kinematics with the ESA/HIPPARCOS mission [1], and its ambitious successor ESA/Gaia [8, 6], which is now measuring a billion stars with precisions of the order of $\sim 10\mu$as. Although the Gaia mission is expected to deliver a rich harvest of gas giant planets [26, 122], in order to detect and study Earth-like planets in temperate orbits, we need to push detection thresholds down to levels better than 1μas, something that will require dedicated new concepts.

Conventional astrometry approaches measure the position of a star, using a grid of reference nearby objects. This requires relatively large fields of view since the distance between science targets and sufficiently bright reference objects is of the order of several arcminutes. However, maintaining long-term instrumental stability over such large angles is notoriously challenging. Several interesting missions have been proposed by groups in Europe [149], the US [153] and China [27], addressing different concepts, which, at least, theoretically, may be capable to solve this problem with highly stable and continuously monitored spacecraft and instruments. This poses, however, an additional non-negligible problem: the instrumental cost scales significantly with the field of view. Thus it is natural to ask the question if it is possible to obtain micro-arcsecond level measurements for certain targets, like Alpha-Centauri, using much narrower fields-of-view, thus avoiding the high costs associated with the stability of large field-of-view concepts.

2.4.2 Narrow-Angle Astrometry

Our ability to perform narrow-field astrometric science ultimately rests on the ability to precisely register the position of the stellar image in each exposure. This meets a fundamental photon noise limit, even with a perfectly stable optical apparatus. Typically, any bright nearby star will provide enough photons so that this theoretical limit is not a major problem, requiring only minutes or hours of integration with a telescope of reasonable aperture. However, the critical limit is not set by photons from the target star but by the absolute stability of the image plane sensor required to perform the measurement; something that can only be accomplished with continual monitoring and ongoing calibration. For the practical narrow-field astrometry, registration of the images is performed by simultaneous monitoring of a constellation of background stars, which provides instantaneous information about the exact plate scale and further order deformations. The astrometric detection error budget is therefore dominated by the accumulation of sufficient counts on these much fainter reference stars that, for a field of view of several arcminutes, are likely to be thousands of times fainter than the target star.

The concept underlying the TOLIMAN mission was developed on the principle that it is possible to entirely sidestep this dilemma for the special case of observations of bright binary stars. In fact, if two bright stars lie close together in the sky, the precise monitoring of their separation will deliver the key science with negligible photon noise. In particular, α Centauri is almost ideally tailored for a mission exploiting narrow-angle self-referenced astrometric detection. As our second nearest celestial neighbour system (the closest one being Proxima Cen), Alpha Cen's pair of stars, both similar to our Sun, is at the sweet spot for detectability within an attainable mission duration yielding signals factors of 2–10 times stronger than the next-best systems. The two habitable zones have wide enough orbits to have a good probability of yielding good signals, yet not so wide as to require an extended mission lifetime for detection.

2.4.3 TOLIMAN

The TOLIMAN space telescope [152] is a low-cost mission which aims to push the boundaries of astrometric measurements in binary star systems and to enable the detection of Earth-like planets around α Centauri. The mission is optimised to search for habitable-zone planets that, for Alpha Centauri, implies deflections with amplitudes of the order of $\sim 1\mu$as over roughly 1-year orbital periods. The detection of such a small astrometric signal (i.e. angle) has never been achieved before in any field of applied science.

To accomplish this task with an affordable spacecraft and mission profile is a daunting engineering problem requiring, among other things an innovative optical and signal encoding architecture. The project, in its present status, explores and reformulates the idea of a Diffractive Pupil based optical system. As originally envisaged, a diffractive pupil telescope would have a set of diffractive features, most simply a regular array of small opaque dots, embedded in the entrance pupil of the instrument [55]. These must be anchored to some element with extreme mechanical stability. The features cause starlight to form a pattern whose features are exactly known and stable so long as the diffractive pupil remains stable. For bright sources, this simple concept offers a cunning solution to the key problem that overwhelmingly dominates astrometric error budgets: the stability of the instrument. When trying to reference stellar positions at micro-arcsecond scales, a host of small imperfections and mechanical drifts, warps and creeps of optical surfaces, generate systematic instabilities that can be orders of magnitude larger than the true signal. Rather than trying to directly contain all these errors, the Diffractive Pupil approach sidesteps them. It creates a new ruler (or system of reference) of patterned starlight against which to register positions in the image plane. The cleverness of this approach is that the diffractive grid of starlight suffers identical distortions and aberrations to the signal that is measured. Therefore, drifts in the optical system cause identical displacements of both the object and the ruler used to measure it, thus making the data immune to a large class of errors that encompasses other precision relative astrometry approaches. The opaque dots pupil proposed by [55] results in a diffraction pattern where the image plane is populated by a regular grid of sidelobe images diffracted from the bright target star. However, when considering broadband (i.e. non monochromatic) illumination, bandwidth smearing of the starlight will draw each sidelobe into a narrow radial streak or ray. The signal recovery proceeds by registering the location of these rays against the background field stars. Because the diffractive ruler takes the form of long narrow radial rays, positional information recovered must be in the orthogonal ordinate. Therefore, the primary observable consists of the recovery of azimuthal positions of (a rich field of) background stars registered against the nearest diffraction rays. For the TOLIMAN mission, the diffractive pupil formulation described above has two fatal flaws: (1) it relies on background field stars and (2) with its radially smeared ruler it is unable to yield precision measurement of the separation of any binary star. To summarise, without using diffractive pupils, the observation of only a single pair of stars, as is the case of Alpha-Centauri, would require radial information measurements or other stars in the field to detect an orbiting planet. Instead, TOLIMAN proposes a novel form of diffractive ruler which generates fine-featured patterns capable of spanning the required separations between the components of a binary star system. The TOLIMAN pupil is capable of creating patterns, uniformly filled in the image plane, with a sharp structure extending in the radial direction. The pattern is characterised by features that have the highest gradient energy and that occupy the minimum span in dynamic range thus optimising our ability to accurately register the resulting pattern and fitting algorithms rely on regions where the image has the strongest slopes or sharp edges. The latter condition is required to spread the starlight preventing saturation of the detector, and spanning the separation of the binary with diffractive features so as to enable the diffractive pupil methodology. Figure 2.4 shows the TOLIMAN diffractive pattern that fills the entire diffractive region, including the core, with sharp structure thus creating clear gradients in the image plane and optimise the ability to precisely register the astrometric signal.

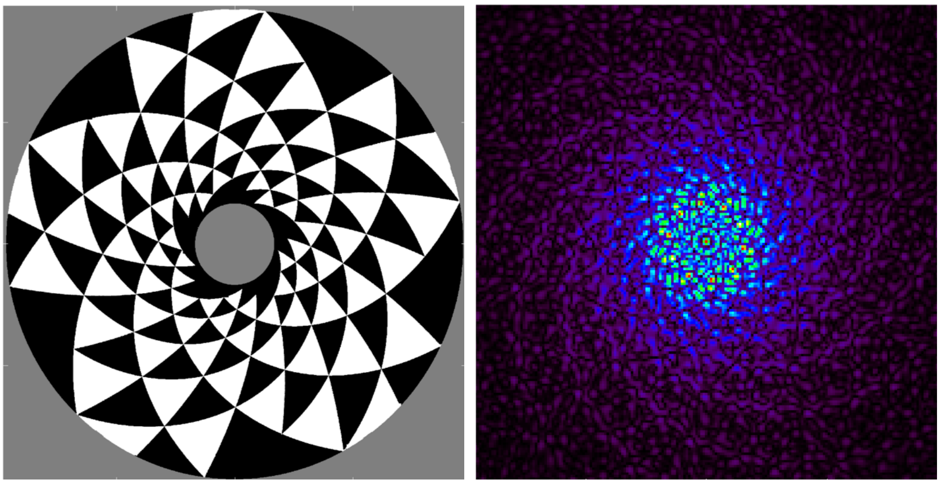

Figure 2.4. Left Panel: a conceptual design pupil for the TOLIMAN mission, with white/black regions indicating discrete phase steps of $0/\pi$. Right Panel: the monochromatic PSF generated yields a complex and strongly featured pattern extending from the core, uniformly filling the region with sharp fringes.

Chapter 3

Deep Learning and its application in Astrophysics

I'm sorry, Dave, I'm afraid I can't do that

HAL9000, 2001: a Space Odyssey

3.1 Machine Learning

Research on Artificial intelligence Artificial Intelligence (AI) began in the 1950s when the pioneers of computer science started wondering whether computers could be able to "think", a topic which is still of the utmost importance today. Today AI research covers a wide range of disciplines and it has been subjected to an enormous amount of scrutiny, confusion and faithful excitement [50]. The quest to truly reproduce and automate higher intellectual functions characteristic of the human brain within computer programs in order to let them solve problems by themselves has been the central position in the field. This resulted in the birth of machine learning and symbolic AI and, more recently, Deep learning. Symbolic AI was the dominant paradigm between the 1950s and the 1980s when scientists thought that human-level AI could be achieved by constraining programs with large sets of rules. While this approach provided solutions to well-defined logical problems, it failed in solving more complex ill-posed problems such as speech recognition, language translation, image classification and so on. Thus a new paradigm emerged from a very simple question: what if instead of giving rules to a program in order to receive an answer to a problem given the data, we give the program the question and its answer, and let it choose its own rules to find the correct answer? This question led to the inversion of the Symbolic AI paradigm: humans create the rules, and the program applies them, humans provide the answers and the program finds the rules. This latter approach has a practical and philosophical advantage over its predecessor which led to its widespread success: it relies way less on the human understanding of the decision process involved in the resolution of the problem or in other words, it requires way less knowledge of the true mathematical model which governs the processes involved in the problem. A characteristic of ML models, thus, is that they are trained or, in other words, presented with a large number of instances related to the task, in order for them to develop rules based on the statistical structures within the examples. The main categories that machine learning techniques fall into are Supervised and Unsupervised learning. In supervised learning, the answers to the problem at hand (labels in case of classifications, parameter values in case of regression) are provided with the training data. Unsupervised learning does not involve the use of labels, so the algorithm has to infer structures using the data only. In this Thesis, I have studied and implemented both types.

3.1.1 Artificial Neural Networks

Artificial Neural Networks Artificial Neural Network (ANN) are a class of ML models which can be used to approximate any nonlinear functional relationship between a set of inputs and outputs [38]. Each layer of a neural network transforms a vector of inputs $x \in R^N$ as follows:L

$$\mathbf{y} = f(\mathbf{Wx} + \mathbf{b}) \tag{3.1}$$

where $W \in R^{(N \times K)}$ is a matrix of weights, $b \in R^K$ is a bias term, and the nonlinear activation function $f : R \to R$ is applied component-wise. The bias term shifts the baseline activation function input away from zero, providing richer behaviour for modelling the functional relationship between the input and output variables. In networks with multiple layers, the output of each layer is connected to the input of the following one,

$$\hat{y_{l+1}} = f_{l+1}(W_l h_l + b_l) = f_{l+1}(W_l f_l(...(f_1(W_0 x + b_0) + b_l) \tag{3.2}$$

where h_l is the hidden layer or feature vector of layer l. The input is processed through all the layers until it reaches the output of the network $\hat{y}$. A network constituted by multiple layers stacked one after the other is commonly known as Multi Layer Perceptron. The parameters (weights and biases) of the network are selected to minimize a *loss function*, such as a mean square error, summarizing the difference between the network output and a desired or observed target value y. Stochastic gradient descent Stochastic Gradient Descent (SGD) is a common optimization process for neural networks: at each stage of training, the network parameters are updated by a small vector proportional to the gradient of the loss function with respect to those parameters. This is straightforward for the output layer; weights and biases in overlying layers can be efficiently calculated through successive applications of the chain rule for derivatives, in a process called *backpropagation*. The use of derivative information for efficient network training requires that the loss function be smooth.

3.1.2 Deep Learning

Convolutional Neural Networks Convolutional Neural Network (CNN) differ from fully connected neural nets only in that their architecture exploits the localized structure of images to reduce the number of network parameters needed. Instead of connecting each neuron in a layer to every other neuron in the next layer, the connection structure of CNN layers is sparse, and parameters are shared across a layer to enforce the translation invariance of features extracted on each scale across the image. Five types of layers are typically used: Convolutional Layers, Pooling Layers, Normalization Layers, Drop Out Layers and Fully-Connected Layers. In the following, we will analyse in detail their functioning and inner workings.

Convolutional Layer

The Convolutional Layer is the most computationally intensive part of a CNN architecture; its parameters consist of a set of learnable filters.

Every filter, also known as a *kernel*, is spatially small (usual sizes are 3×3, 5×5 and so on, where three and five are sizes in the number of pixels), but includes weights for each channel of its input. For the first layer, these channels are the data channels (for example R, G, and B in a three-channel image). In subsequent layers, each channel corresponds to the output of a single kernel from the previous layer. During the forward pass, each kernel slides across the spatial dimensions of the input, computing the dot product between itself and the part of the input volume that it encompasses (*convolution*). As the kernel slides, it produces a bi-dimensional activation map that encodes the responses of the kernel at every spatial position. The content of the activation map at each location is a direct response to some visual feature present in the image to which the kernel is sensitive, such as an edge or a colour. Each convolutional layer employs multiple different filters, producing a set of activation maps that are stacked along the depth to produce a multi-channel output. Due to the limited size of the filters, neurons are not connected to the full extent of the input volume but only to a small region (the *receptive field*). The connections are thus local in space (width and height of the input), but are always fully connected in-depth (i.e. across learned/extracted features).

The structure of the output volume of a convolutional layer is controlled by three hyper-parameters:

- *depth*: the number of filters learned in the layer;
- *stride*: the number of pixels the filter is shifted along the spatial dimensions of the input volume. It is usually set to one but it can be set to higher values, depending on the image geometry, to achieve less redundancy in the output volume. The stride thus can control the spatial dimensions of the output volume thus achieving a similar compression effect to what pooling can achieve;
- *padding* or *zero-padding*: the width in pixels of a spatial region on the borders of the output that is filled with zeros. It controls the spatial dimensions of the output volume and can be used to preserve the spatial dimensions through the layer.

Finally, to ensure that each kernel is learning a single feature that has a consistent interpretation across the spatial extent of the input, all neurons in the same depth slice share the same weights and biases, irrespective of where across the extent of the input they are applied. Thus the action of each filter in the forward pass becomes a discrete convolution of a single set of kernel weights with the input.

A convolutional layer acts to encode its inputs into a latent space spanned by the features it learns. However, some architectures we will consider in later sections also involve the transformation from the learned latent space back into the image domain. Thus, while convolutional layers typically decrease the spatial extent of their inputs and increase the number of features, we will also need *deconvolutional layers* which increases the spatial dimensions and decreases the number of channels, recombining a potentially large number of learned features into a flat image. Mathematically both convolutional and deconvolutional layers can be summarized as

$$l^h = f\left(\sum_{i \in L} x^i \otimes w^h + b^h\right), \tag{3.3}$$

where l^h is the latent representation of the h-th activation map of the current layer, f is the activation function, and x^i is the i-th activation map of the L-feature activation of the previous layer in the network (or the l-th channel of an L-channel image in the case of the first convolutional layer after the input image). w^h and b^h are, respectively, the weights and biases of the h-th activation map (shared by all neurons of the map) of the current layer. Given that x^i has size $m \times m$ and the filters have size $k \times k$, a convolutional layer produces an output feature map with shape $(m-k+1) \times (m-k+1)$, thus reducing the size of the input. A de-convolutional layer outputs a feature map with shape $(m+k-1) \times (m+k-1)$, thus increasing the size of the input.

Pooling Layer

The architectural function of a Pooling Layer is to reduce the spatial size of the representation, which reduces the number of parameters, lightens the computational load, and mitigates overfitting. The pooling operation is carried out independently on each input feature, leaving the number of input features unchanged. Different criteria in the literature exist to perform the pooling operation, including *max*, *average* and *L2-*

norm pooling; max-pooling is the most commonly used. There are also *un-pooling* layers to desegregate and expand activation maps in transformations back towards the image domain.

A max-pooling layer pools features by computing the maximum within the feature map and outputs a feature map with reduced size, according to the chosen size of the pooling kernel. To perform a successive un-pooling, the max-pooling layer also records a set of switch variables which describe the positional information relative to the pooled features. The un-pooling layer restores the max-pooled features to the correct position specified by the relative switch variable values. The combination of max-pooling and un-pooling layers is thus able to retain both the image magnitude (answering the "what" question) and the positional information (the "where" question). An alternative to pooling and un-pooling layers is the use of the stride parameter of the convolution and de-convolutions.

Normalisation Layer

Training Deep Neural networks can be a tricky endeavour for several reasons: a) the distribution of each layer of a DL model can show highly varying means and variances and thus careful choices of initialization and learning rate must be taken. While training, the layers distributions change, a phenomenon known as *covariate shift* [137], and thus the layers have to constantly need to adapt to new distributions; b) the deeper and more complex a network is, the more is prone to overfit the problem and that means that the introduction of regularization is critical to properly converge to a true minimum of the optimisation process (see Sec. 2.3.3); c) prepossessing the data before feeding it to a model has been shown to be beneficial to keep the estimation problem controlled [54]. Batch Normalisation [76], which is the normalisation technique employed in all DL models developed in this Thesis, conveys all three benefits into a single solution and it works by standardising its input and then it applies a scaling coefficient and an offset to recover the lost degrees of freedom

$$\begin{aligned}&BN(x) = \gamma \odot \tfrac{x-\mu_B}{\sigma_B} + \beta \\ &\mu_B = \tfrac{1}{|B|} \textstyle\sum_{x \in B} x \\ &\sigma_B = \tfrac{1}{|B|} \textstyle\sum_{x \in B} (1-\mu_B)^2 + \epsilon\end{aligned} \tag{3.4}$$

where B is the current input batch and ϵ is a small constant to avoid division by zero.

Dropout Layer

As previously stated ANNs and CNNs can theoretically map any functional relationship between their input and outputs but, realistically, due to the limited size of the training data they tend to overfit but this can also be prevented through regularization. Another issue of DL models is that they are usually characterised by fairly large number of hyperparameters and thus finding the optimal parameter's set can be computationally intensive. Dropout achieves regularization and reduces the number of searches that have to be performed to optimise the model hyperparameters by randomly dropping out units in the model (and thus their relative connections). Applying dropout thus consists in sampling a "thinned" version of the model formed by all units which survived the dropout. A model net with n units can be seen as a collection of 2^n possible thinned neural networks. These networks all share weights so that the total number of parameters is still $O(n^2)$, or less but, for each presentation of each training case, a new thinned network is sampled and trained. So training a neural network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing, where each thinned network gets trained very rarely, if at all. At test time, the prediction is made through a weighted geometric mean of the predictions from the "thinned models". If a unit is retained with probability p during training, the outgoing weights of that unit are multiplied by p at test time.

3.1.3 Architectures for inverse problems in imaging

The main advantage of DL over most classical dimensionality reduction techniques is that the layered structures of Deep Neural Networks (DNNs) can encode an input representation with increasing levels of abstraction in successive layers [86, 50]. For that reason, in the last decade, Deep Learning has been successfully applied across a wide range of applications including computer vision, speech recognition, bioinformatics and astroinformatics. "*CNNs networks are exquisitely well suited for image processing because they can quickly extract relevant statistics from the input and utilise* *them to solve the inverse problem*" [79]. Several DL models are built on CNNs, in the following I showcase the ones which are relevant to this Thesis:

Convolutional Autoencocoder

The idea behind an Autoencoder [66] is that relevant features in the data, such as the shape of the structures within the image, should be relatively robust with respect to the noise components, and hence can be learned by compressing the data to a lower dimensional latent space. The latent space representation can then be used to reconstruct a *noiseless* version of the input data. An Auto-Encoder model AutoEncoder (AE) is composed of an encoder and decoder part; the encoder $f : X \to H$ transforms the input into a lower-dimensional representation (the *latent space*), while the decoder $g : H \to X$ tries to reconstruct the original input from this representation. By constraining the latent space to be of a lower dimension than the original input data, we can force the Autoencoder to capture the most important features of the input data in order to reproduce it successfully. This type of restriction can be used for feature extraction and for dimensionality reduction. During the learning process, network parameters are adjusted to minimize a loss function

$$L(x, g(f(x))) \tag{3.5}$$

that encodes the difference between the input x and its reconstruction $g(f(x))$. As for the ANNs discussed in Sec. 3.1.1, L must be smooth in order to use gradient-based minimization algorithms such as SGD. If L is chosen to be linear, the auto-encoder performs a dimensionality reduction similar to Principal Component Analysis (PCA); in fact, the latent space h ends up being the principal subspace of the input data. If instead, L is non-linear the auto-encoder can learn much more complex representation. Generally, autoencoders are built by two shallow fully connected NNs joined through a lower-dimensional latent space. A Convolutional Autoencoder Convolutional Autoencoder (CAE), instead, contains, in the encoder part, a stack of convolutional and max-pooling layers (or strided convolutions) before the fully connected layer and, in the decoder part, a stack of up-sampling and de-convolutional layers after a fully connected layer. It

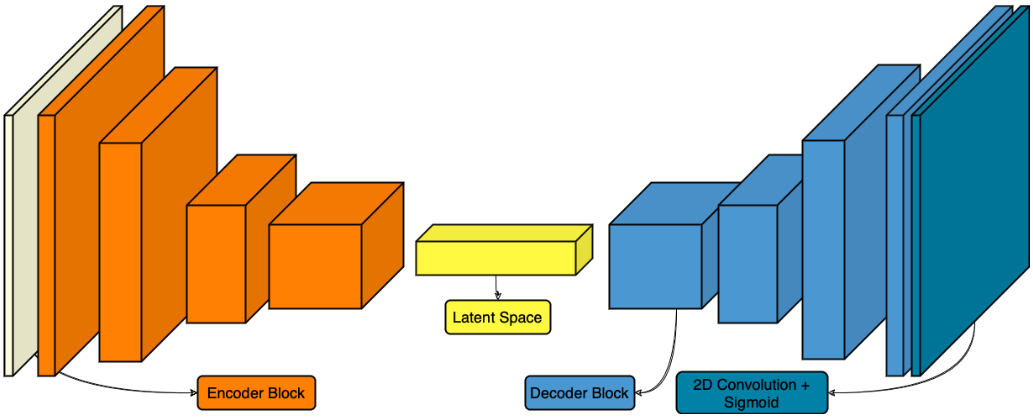

Figure 3.1. Schema of the CAE architecture.

has been shown [170] that CAEs are better suited, with respect to AEs, for image processing and reconstruction tasks, due to the full utilisation of the CNNs capacity to extract a hierarchical set of features from the images. These have been proven to show a better performance over shallow neural networks when working with noisy or complex images. Moreover, the combination of a convolutional and max-pooling layer allows the higher-layers representations to be invariant to small rotations and translations thus helping with the inevitable systematic characteristic of astronomical instruments (such as jitters, thermal noise, and atmospheric effects).
In recent years CAEs have been applied to solve a wide range of problems in the Astrophysical context; to model the Point Spread Function of Wide Filed Small Aperture Telescope [78], to uncover and separate the faint cosmological signal from the epoch of reionization [91], to classify galaxies Spectral Energy Distributions [47], to identify Strong Lenses candidates in the simulated data of the Euclid Space Telescope [30] and to solve the Star - Galaxy classification problem [59]. Moreover, in the fields of Computer Vision and Image Processing, CAEs have been successfully used to recover structured signals from natural images [110], for image compression [44], [150], [10], [151], [154], achieving compressing performances similar or better than the JPEG 2000.

Residual Networks and Residual Connections

As we have seen many computer vision tasks have been successfully solved with CAEs and DL models in general, but it was discovered that a model, while training, may encounter the *degradation problem*. Degradation happens when increasing its depth by adding layers, its accuracy platoes and then degrades rapidly. Unexpectedly it was discovered that this error it is not due to overfitting [63, 140] but due to increasing network complexity. Consider a shallower architecture and a deeper counterpart. The only explainable reason why the two architectures are both converging is that the added layers of the deeper network are identity mappings of layers from the shallower model. The existence of this constructed solution implies that the two models should show similar performances, but experimentally the shallower model shows a lower error. Residual Networks [65] solve the degradation problem through the addition of residual or shortcut connections between layers. Instead of assuming that stacked layers should fit the underlying mappings, layers are explicitly asked to fit a residual mapping. The underlying hypothesis is that it is easier to optimise the residual mapping than to optimise the original mapping. In practice, shortcut connections perform identity mappings and their outputs are added to the outputs of the stacked layers while not increasing the number of learnable parameters or the model complexity. The basic component of a Residual Neural Network Residual Neural Network (ResNet) is the so-called Residual Block outlined in Fig. 3.2.

3.1.4 Recurrent Neural Networks

Recurrent Neural Networks Recurrent Neural Network (RNN) are deep learning models that capture the dynamics of sequences via recurrent connections, which can be thought of as cycles in the network of nodes. This kind of network is unrolled across time steps (or sequence steps), with the same underlying parameters applied at each step. While the standard connections are applied synchronously to propagate each layer's activations to the subsequent layer at the same time step, the recurrent connections are dynamic, passing information across adjacent time steps. RNNs can be thought of as feedforward neural networks where each layer's parameters (both conventional and recurrent) are shared across time steps. A known

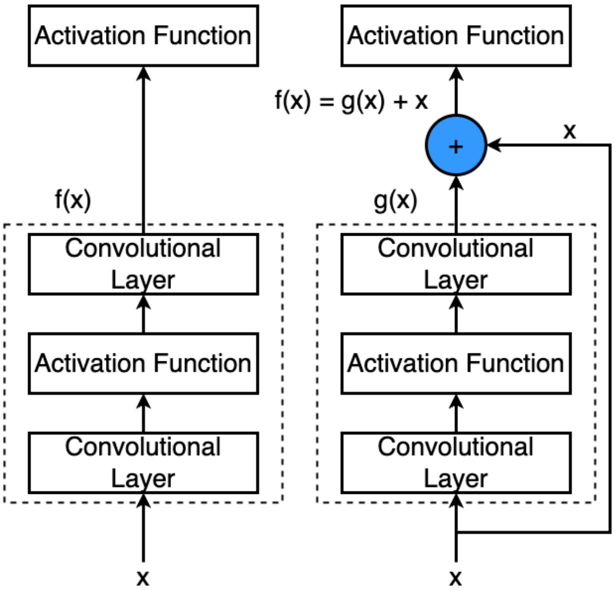

Figure 3.2. Left: standard convolutional block which has to learn the mapping $f(x)$; Right: the residual block which has to learn the residual mapping $f(x) = g(x) - x$

problem of RNNs is that of *vanishing and exploding gradients* [13], i.e. during model training the gradient approaches zero (vanishing) or infinite (exploding) which renders the model's weights unchangeable and usually results in the optimization process getting stuck in some local minimum. The problem was partially solved with the introduction of Long Short Memory Networks Long Short Term Memory Network (LSTM) [67] in which recurrent nodes are replaced by memory cells containing an *internal state*, i.e. a node with a self-connected recurrent connection of fixed weight set to unity and a number of multiplicative gates that determine whether i) a given input should have an impact on the internal state (*input gate*), ii) the internal state should be flushed to zero (*forget gate*), or iii) the internal state of a given neuron should be allowed to affect the cell's output (*output gate*). The Gated Recurrent Unit Gated Recurrent Unit (GRU) [31] is a streamlined version of the LSTM where the three gates introduced by the LSTM are replaced by two gates: the *reset gate* and the *update gate* which, respectively, control how much of the previous hidden state must be remembered and the degree to which the current hidden state is similar

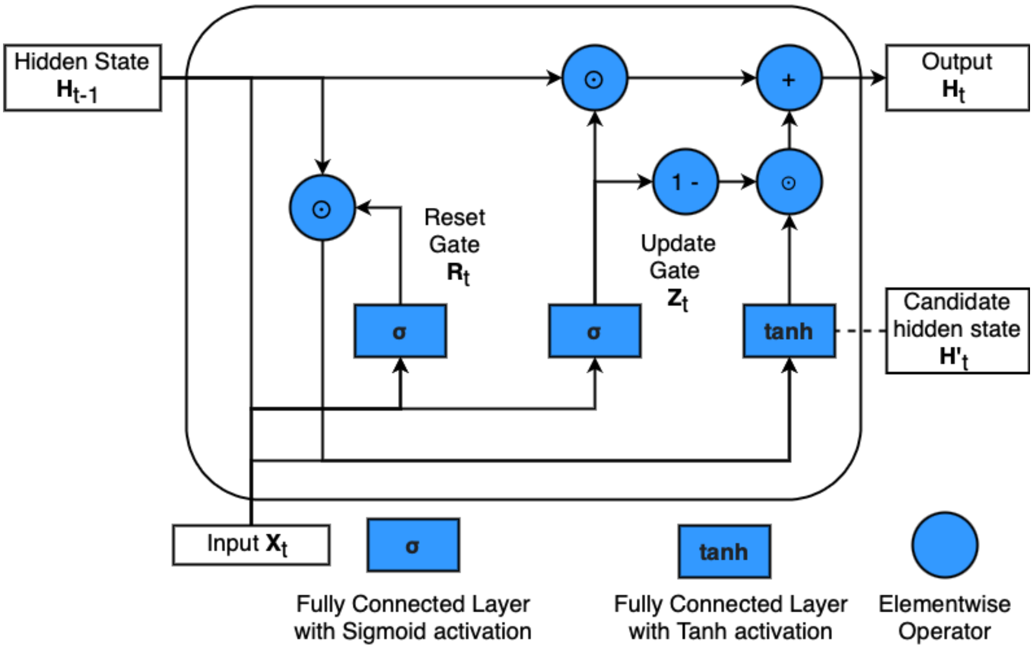

Figure 3.3. Gated Recurrent Unit architecture showing the flow of data within the network.

to the previous one at each frequency iteration. The first helps capture short-term correlations in the data, while the second one captures long-term correlations. As shown in Fig. 3.3, which outlines the data flow inside a GRU, these gates are implemented through fully connected layers with a sigmoid activation function:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{3.6}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{3.7}$$

where X_t is the input vector, H_t is the hidden state at the previous frequency step, and W and b are the weights and biases of the fully connected layers.

The reset gate is multiplied (element-wise) with the previous hidden state and the effect of this multiplication is passed along to the candidate hidden state

$$H'_t = tanh(X_t W_{ht} + (R_t \odot H_{t-1})W_{hh} + b_h) \tag{3.8}$$

where W_{ht} and W_{hh} are the model weights, b_h is the bias term and $\odot$ is the Hadamard element-wise product. If the entries of R_t are close to 1, the GRU behaves just as a simple RNN, while if it is close to 0, the previous hidden state is forgotten and the RNN behaves like a Multi Layer Perceptron Multi Layer Perceptron (MLP). The final hidden state is given by the element-wise convex combination of H_{t-1} and H'_t

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot H'_t \tag{3.9}$$

If the entries of Z_t are close to 1, the hidden state at the previous iteration is retained and the information brought by the current input X_t is ignored. Otherwise, if the entries of Z_t are close to 0, the old hidden state is updated with the new candidate hidden state H'_t. Up until now, we have focused on defining networks consisting of a sequence input, a single hidden RNN layer, and an output layer. Despite having just one hidden layer between the input at any time step and the corresponding output, there is a sense in these networks are deep. Inputs from the first time step can influence the outputs at the final time step T. However, we may also wish to retain the ability to express complex relationships between the inputs at a given time step and the outputs at that same time step (which is ensured by having deep chains of nonlinear activation functions [38]). To do so we could construct RNNs that are deep not only in the time direction but also in the input-to-output direction. The standard way to achieve this is to stack multiple RNNs on top of each other. Given a sequence of length T, the first RNN produces a sequence of outputs, also of length T. These, in turn, constitute the inputs to the next RNN layer, and so on. A Deep GRU is thus obtained by stacking multiple GRUs one after the other interlaced by layers of non-linear activation functions.

3.2 Deep Learning applications in Astrophysics

Given the showcased rapid growth in size and complexity of Astronomical datasets, Machine learning methodologies, both supervised and unsupervised, given their ability to cope with very complex data, have been extensively exploited by the community to solve a wide variety of problems spanning all aspects of the astronomical data life, from instrument monitoring to data acquisition and ingestion, to data analysis and interpretation [12, 82, 73, 102, 84, 109, 147, 169, 93, 43, 29, 46, 162, 108, 128, 120, 126, 23]. This has also led to the implementation of many deep learning-based pipelines in the field of Astrophysics. To quote only some of the most recent and interesting applications, [18] utilised a CNN and a UNet to measure the photometric properties of blended systems of monochromatic mock simulations of EUCLID observations of high redshift galaxies. To train and test the models, they utilized artificially blended observations taken from the CANDELS survey. Despite the simplistic constraints imposed on the data: two galaxies per stamp, one galaxy pinned at the image centre and no blends with completely overlapping galaxy centroids (also referred to as unrecognized blends), the networks have demonstrated that, on monochromatic images, they outperform the traditional SEXTRACTOR [15] approach with respect to photometric accuracy, precision, outliers fraction and stability towards different morphological types. [112] have compared the performances of several architectures (ResNet, LSTM, 2D and 3D CNN) for the measurements of cosmological parameters from SKA light cone simulations. They found that a simple 3D CNN was able to reliably infer the cosmological parameters. [77] in their paper they present γ-PhysNet, a deep multi-task architecture for gamma-ray reconstruction from images taken by an Imaging Atmospheric Cerenkov Telescope (IACT). The architecture is constituted by a backbone feature extractor (ResNet-56) interlaced by several Dual Attention modules [146] followed by a multi-task block composed of a global feature network and a local feature one, both based on fully connected layers. The global feature part performs a global average pooling followed by a fully connected layer, while the local takes directly the backbone features, flattens them and then feeds them to two fully connected layers. [106] proposed to use a Convolutional Variational Autoencoder (VAE) in an unsupervised fashion to diagnose system health for modern radio telescopes. The model project the high-dimensional data to a low-dimensional descriptive space which telescope operators can use to spot failures in the system. Moreover they tested the use of a Support Vector Machine (SVM) classifier [35] to automatically classify failures from the low-dimensional projection. A qualitative assessment of the model performance was performed on real Low-Frequency Array Telescope (LOFAR) data obtaining a classification accuracy of 0.9. [25] proposed a novel sequential classification model for astronomical time series based on a Recurrent Convolutional Neural Network (RCNN) and an image simulation framework capable of simulating seven classes of astronomical objects (two non-variating ones, and five variating classes). The RCNN is constituted by a CNN which extracts features from the input sequence of images and an LSTM which is tasked with classifying the extracted time sequence. Their architecture is tested against a Random Forest performing classification based on features extracted from the images through traditional methods (photometric features extracted through FATS [113]) obtaining a similar classification accuracy but lower computational times. Their results show that the proposed RCNN model is able to classify correctly the simulated test set, as well as the real data set after performing fine-tuning. [62] presented *Morpheus* a new model for generating pixel-level morphological classifications of astronomical sources. Morpheus leverages advances in deep learning to perform source detection, source segmentation, and morphological classification pixel-by-pixel via a semantic segmentation algorithm (U-Net). By utilising morphological information about the flux of real astronomical sources during object detection, Morpheus shows resiliency to false positives. Morpheus is evaluated by performing source detection, source segmentation, and morphological classification on the Hubble Space Telescope data in the five CANDELS fields with a focus on the GOODS South field, and demonstrates high completeness in recovering known GOODS South 3D-HST sources with $H < 26$ AB. [138] employed a variant of a Generative Adversarial Network (GAN) known as Spatial GAN (SGAN) to generate images resembling Hubble Space Telescope eXtreme Deep Fields (XDFs) showing that the properties of the galaxies in the generated images have a high level of fidelity with respect to those of galaxies in the real XDFs. Their technique can be generalized to any appropriate imaging training set, offering a new purely data-driven approach for producing realistic mock surveys. [111] employed a DeepGRU to provide an early classification of astronomical time series data (trained on 12 transient classes) in order to identify transients in the upcoming wide-field surveys such as the Zwicky Transient Facility (ZTF) ant the Large Synoptic Survey Telescope (LSST). The classification accuracy improves over the lifetime of the transient as more photometric data becomes available, and across the 12 transient classes, we obtain an average area under the receiver operating characteristic curve of 0.95 and 0.98 at early and late epochs, respectively. [85] presented a hybrid Bayesian framework for inverse problems that combines analytic forward modelling for the likelihood with deep generative models (PixelSNAIL [28] for complex data-driven signal priors. their approach makes explicit use of physically motivated problem structure and prior knowledge from high-quality observations. When applied to the blind-source separation problem of galaxy blending, the method can retrieve multi-component models of astronomical scenes that are by construction robust to changes in observational conditions. [40] proposed to combine a CNN with a Mixture Density Network (MDN) which is a modification of the MLP capable of generating distributions instead of point estimates. In particular, the MDN interprets the output of the network as the means, standard deviation and weights of a mixture model employing n Gaussians as basis functions. Substituting the last fully connected layer with the MDN allows thus to generate Probability Density Functions Probability Density Function (PDF) of photometric redshifts. In order to evaluate the performance of the proposed method, three experiments have been performed on galaxies, quasars and a mix of two and the method is compared with an MDM and a Random Forest showing that the proposed architecture outperforms the two competing models. [164] utilised a custom deep learning model to detect Low Brightness Galaxies within SDSS images. Their architecture is composed of three models: a ResNet50 to extract features from the images, a CNN to regress the class probabilities for the galaxies within the images, and another CNN to predict the coordinates of the galaxies within the images. As loss functions, they employed the Binary Cross Entropy loss for the classification problem and a probability-weighted mean squared loss for the regression problem. [2] proposed a novel deep learning-based pipeline for detecting quadruply lensed quasars. The pipeline was trained on simulated observations comprising both lensed and non-lensed quasars and is composed of several deep learning models combined together to solve a binary classification problem: the lower-dimensional representation extracted from four Variational Autoencoders, two ResNets, one with rectangular convolution, one with polar, and a U-Net-like model. [49] presented a novel pipeline called Removal of BOgus Transients (ROBOTs) in order to assess transient candidates within the Deeper, Wider, Faster (DWF) program. One of the main bottlenecks of DWF is the time required to assess candidates for rapid follow-up and manual inspection prior to triggering space-based or large ground-based telescopes, the authors create a pipeline to address this main bottleneck. The pipeline employs a combination of a CNN and a CART: the CNN is used to classify the input images, while the CART is used to further classify candidate light curves identified by the CNN. In fact, the CNN algorithm is used to determine the quality of individual data points rather than that of the light curves, and thus the CART is used to incorporate the temporal information within the decision framework. As it will become clear. As we will demonstrate, their strategy of utilising extra information in the temporal domain to further classify candidates is similar to how we use frequency information to deblend the potential candidates found by the Autoencoder in our ALMA source detection and characterisation pipeline (see Chapter 4 Sec. 4.2.4 step 8). [117] utilised a CNN trained with both real observations from the SDSS [16] and simulated observation from EAGLE [37] to identify galaxy mergers. By comparing the performances on both real and simulated data, they concluded that a CNN trained on simulated data could be used to detect mergers in real observations. Also in the field of Radio Astronomy, some interesting examples can be found. For instance, [19] introduced attention-gated networks for radio galaxy classification demonstrating that these networks can perform similarly to CNN-based classifications while utilising significantly fewer trainable parameters and thus reducing the complexity of the model and the risk of overfitting. Moreover, they employed the produced attention maps [61] to help the interpretation of the results. [168] proposed a novel deep-learning pipeline, Concat COnvolutional Neural Network (CCNN), for selecting pulsar candidates within the Commensal Radio Astronomy FasT Survey [89]. The main idea behind their pipeline is to use several specialized CNNs to extract the low-dimensional latent information from the pulse profile, the Dark Matter profile, the frequency versus phase plot, and the time versus phase plot. The latent spaces are then concatenated and fed to a Multi-Layer Perceptron that performs a binary classification task. [131] employed a CNN to detect point sources within images.

The CNN is trained on simulated maps with known point source positions and learns how to amplify the Signal to Noise Ratio (SNR) in the input map. The map is then converted into a catalogue using a dynamic blob detection algorithm. The authors compared their pipeline performances with PyBDSF (a classical, widely used peak detection algorithm, [107]), obtaining better results on all metrics. [98] presented a novel deep learning model called ConvoSource, which utilises 2D Convolutions to map the input observations to the target signal map using a binary cross-entropy loss function. The model performance was found to be similar to that of PyBDSF. Both [131] and [98] use as input images already correlated data and thus rely on the CLEAN algorithm [69]: an assumption which as we shall discuss below, may affect the final performances. [33] developed a novel deep learning pipeline called POLISH for image super-resolution and deconvolution of radio interferometric images. Their model aims to learn the mapping between the input "dirty" images and the target sky model images. Their architecture is a modified version of the Wide Activation for Efficient and Accurate Image Super-Resolution [166] capable to allow for high-dynamic range, multifrequency output and uncertainty in the PSF. To test the performances of their model they generated DSA-2000 [58] simulated data, and compared the performances of their model on the dirty images with *source extractor* applied on cleaned images. The cleaning was performed with the CLEAN algorithm [69]. POLISH achieved both better mean peak signal-to-noise ratio (PSNR) and better structural similarity index (SSIM) as well as a lower fraction of false detections when compared to the true sky models. [133] presented a deep-learning architecture for reconstructing incomplete Fourier data in radio interferometric images. By reconstructing missing data in the visibility space, a clean sky model image can be obtained by simply taking the inverse 2D Fourier transform of the reconstructed data. In order to train and test the performances of their architecture, the authors generated simulated VLBI $u-v$ maps both noiseless and noisy, corresponding for both point-like and Gaussian sources and compared the results of their model with those by *wsclean* [114].

The direct reconstruction of the sky model from the $u-v$ map has been tackled with classical statistical approaches by several authors [71, 83, 3] but, to the best of our knowledge, [133] were the first to tackle the problem using a Deep Learning approach.

Finally, [127] presented a novel deep learning pipeline (DECORAS) for detecting and characterising sources in VLBI simulated images. Their pipeline consists of two Deep Convolutional Autoencoders (similar in scope and structure to the first deep model of our ALMA pipeline, i.e. Blobs Finder) used to detect sources within the images and an XGboost model for the regression of source peak surface brightness. Other morphological parameters are derived by fitting 2D Gaussian models to the sources. The authors train and test their pipeline on simulated VLBI simulations and directly compare with BLOBCAT [57] (see Sec. 2.3 for details) showing that the two methods have similar completeness levels for $SNRs <= 5.5$. While for $SNRs > 5.5$, DECORAS outperforms BLOBCAT by a factor of two.

Chapter 4

Detection and Characterization of ALMA Sources through Deep Learning

Do. Or do not. There is no try.

Master Yoda, The Empire Strikes Back

ALMA has been a game changer in high-resolution aperture synthesis imaging [4], and a development road map [24] has been approved with the goal of meeting its high standards. One of its main objectives is to broaden the instantaneous bandwidth of the receivers, upgrading the correlator to process the entire bandwidth. As a consequence, ALMA sensitivity and observing efficiency will improve, producing imaging products at least two orders of magnitude larger than the current cube size (which already resides in the GB regime). As a result of these upgrades, ALMA cube imaging will become a very demanding task. In this thesis, we introduce a new approach to the image reconstruction problem capable to overcome some of the limitations introduced by the CLEAN algorithm outlined in Sec 2.3.4. We propose a deep-learning-based pipeline developed with the goal to detect emission lines in ALMA cubes while speeding up traditional methods and reducing spurious signal detection. For the aforementioned reasons, our pipeline uses as input information, 'dirty' calibrated ALMA cubes that have not undergone any prior deconvolution. The main novelty of our proposed pipeline with respect to the previously cited architectures [131, 98, 33, 133, 127] is that we combine spatial and spectral (frequency) information to detect and characterize sources. Utilising frequency information can both help in deblending spatially blended sources in the integrated images (obtained by integrating the cubes along the frequency axis), and help in the recovery of faint sources. Our deep learning pipeline can be decomposed into six logical steps:

1. 2D source detection is performed on the integrated cubes (along frequency) using a Deep Convolutional Autoencoder (hereafter Blobs Finder);

2. the detections (blobs) found by the Autoencoder are used to extract spectra which are then fed to a Recurrent Neural Network, capable to perform spectral denoising (Deep GRU). All the peaks in the spectra are fitted with 1D Gaussian distribution;

3. sources are spectrally focused by cropping a 64×64 pixels box around their centres (found by Blobs Finder) in the image plane and by integrating within their frequency emission ranges found in the previous step. False positives are detected in this step;

4. focused images are then fed to three dedicated ResNets that regress the sources morphological parameters: full-width half maxima in the x and y directions and the projection angle θ;

5. the found morphological parameters are used to construct a 3D models of the sources, which are used to mask the cubes and produce line emission images and mean continuum images. Once the continuum is subtracted from the line emission images, they are finally fed to a specialized ResNet that regresses the source flux densities.

This Chapter is arranged as follows: In Section 4.1 we describe the simulation data that we used to generate realistic ALMA observations. The simulation algorithm is used to train and test our pipeline. The simulation parameters and the characteristics of the output cubes and the sources within them are explained. In Section 4.2 the architectures of the employed deep learning models of our pipeline, together with a complete data flow are described. The data flow is used to explain the inner workings of the pipeline, as well as a description of the training strategies adopted for all the models. In Section 4.3.1 we discuss the pipeline performances in detecting and characterizing sources within the test set. We compare the detection performances with those of two traditional methods BLOBCAT [57], Sofia-2 [160], and another deep learning pipeline DECORAS [127]. In Section 4.4 we compare the Blobs Finder performances in solving the ALMA image reconstruction problem (recovery of the true sky model from dirty images) with those of tCLEAN. Finally, in Section 4.5 we discuss our results, draw our conclusions, and lay the prospect for future work. Both the simulations code (ALMASim) and the pipeline (DeepFocus) are written in Python [155], and all software has been made publicly available through GitHub in order to allow for further developments, testing and reproducibility.

4.1 Simulations

To train and test the capabilities of the proposed pipeline, we needed thousands of ALMA model and dirty cube pairs. We generate our own simple but realistic simulations of ALMA observations by combining python and bash scripting with the Common Astronomy Software Application Common Astronomy Software Applications (CASA) v. 6.5.0.15 [105] python libraries. The need to use simulated data instead of real ALMA observations arises from several practical necessities: a) in order to evaluate the pipeline's performance and reliability and its dependence on sources' observational and morphological parameters, we need full control over the data properties; b) in order to assess the reconstruction quality of our pipeline and its ability to solve the deconvolution problem without relying on CLEAN deconvolution, we need noiseless sky observations which are unattainable for real observations. The closest alternative to the use of simulated cubes would be to use CLEAN to produce deconvolved cubes of ALMA observations in which source detection had been performed. These deconvolved cubes, however, would be dependent on CLEAN solution (of the deconvolution problem), thus biasing our models. Given that the main foreseen scientific application of the pipeline presented in this work is the search of serendipitous sources within high redshift ($z > 3$) ALMA observations, we simulate, in each cube, a central primary source with a given Signal to Noise Ratio Signal to Noise Ratio (SNR) (simulating this way a realistic target observation, in the phase centre of the antennas array), surrounded by less bright serendipitous sources that can occupy any position in the cube. To generate 3D sources, we combine 2D Gaussian Components in the spatial plane with 1D Gaussian components (emission lines) in frequency space. For each source, first, the emission line is generated by sampling from a uniform distribution of line parameters, i.e., the line position or central frequency, the line amplitude or the value in dimensionless units of the peak, and the line Full-Width Half Maximum Full Width Half Maximum (FWHM) in a number of frequency channels. Once the parameters have been sampled, the line is generated through the following equation:

$$l(z) = ae^{-\frac{(z-z_{cen})^2}{2(FWHM_z/2.35482)^2}} \tag{4.1}$$

where a is the line amplitude, z_{cen} is the central frequency and $FWHM_z$ is the line full width half maximum.

Then a 2D Gaussian profile is generated by sampling from a uniform distribution of source parameters, i.e., the source position x, y, the source FWHMs in both the x and y planes $FWHM_x$ and $FWHM_y$, the source projection angle θ, the source amplitude a and the source spectral index σ, i.e. the dependence of the source radiative flux density on frequency. Once the parameters have been sampled, the 3D Gaussian profile is generated through the following set of equations:

$$\begin{aligned} g(x,y,z) = {} & 10^{(log_{10}(a)+spid\times log_{10}(v_1/v_2))}. \\ & \cdot e^{-\frac{((x_c-x)/FWHM_x)^2+((y_c-y)/FWHM_y)^2}{2}} \end{aligned} \tag{4.2}$$

where

$$\begin{aligned} & x_c = x \cdot cos(pa) - y \cdot sin(pa) \\ & y_c = y \cdot sin(pa) + y \cdot cos(pa) \\ & v_1 = 230 \cdot 10^9 - (64 \cdot 10^6) \\ & v_2(z) = v_1 \cdot z \cdot 10^6 \\ & G(x,y,z) = g(x,y,z) + l(z) * g(x,y,z) \end{aligned} \tag{4.3}$$

Table 4.1. Sampling intervals of the model source parameters. Sources are generated by randomly sampling from the outlined uniform distributions. The first column shows the parameter name, the second the range from which the parameter values are sampled, and the third the units.

Parameter Name	Range	Unit
Number of components	[2 – 5]	–
Amplitude of 2D Gaussian compoment	[1 – 5]	arbitrary
FWHM of the 2D Gaussian component	[2 – 8]	pixel
Spectral index	[-2 – 2]	–
Position in the xy plane	[100 – 250]	pixel
Position angle	[0 – 90]	deg
Line amplitude	[1 – 5]	arbitrary
Line center	[10 – 110]	chan
Line FWHM	[3 – 10]	chan

Each model cube is thus created by first generating a central source and then a random number between 2 and 5 additional sources such that their emission peaks are lower or equal to that of the central source. The full list of parameters, their units, and the ranges from which they are sampled, are shown in Tab 4.1.

All source parameters are stored in a .csv file in order to use them as targets for the ResNets in the last stages of the Deep Learning source detection pipeline (see Sec. 4.2.4).

We feed the sky models to the *simobserve* task to simulate interferometric measurements sets through a series of observing parameters. The full list of parameters is outlined in Tab 4.2.

We use the ALMA Cycle 9 C-3 for the antenna configuration, simulating 43 antennas in the 12-m Array with a maximum baseline of 0.50 km (ALMA Cycle 9 Technical Handbook). Once the measurement sets have been produced, to create the dirty cubes, we feed them to the TCLEAN task initialized with the parameter *niter* set to zero. This forces the task not to run the TCLEAN algorithm but only to perform the fast Fourier transform of the visibility data and the gridding using the pixel parameters outlined in Tab 4.2. The resulting data products are noisy cubes convolved with the ALMA synthesized beam with an angular size of 10 squared arc-

Table 4.2. Full list of *simobserve* parameters to generate the measurement sets, i.e. the interferometric visibility data cubes. The first column shows the parameter name, while the second shows our chosen value.

Parameter Name	Value
inbright	0.001 Jy / Pix
indirection	J2000 03h59m59.96s -34d59m59.50s
incell	0.1 arcsec
incenter	230 GHz
inwidth	10 MHz
integration	10 seconds
totaltime	2400 seconds
thermalnoise	tsys-atm
user_pwd	0.8

seconds and a total bandwidth of 1.28 GHz. The spatial dimensions of the cubes are 360 × 360 pixels and they are characterised by 128 channels in frequency. Nevertheless, the simulations are already using Pardo's ATM library to construct an atmospheric profile for the ALMA site comprising a water layer in the atmosphere and thermal noise. After the dirty cubes are produced, we inject 3D uncorrelated white Gaussian noise. The RMS of the noise is automatically adjusted to reach a target SNR measurement for the central source. This way, we can produce simulations containing sources with specific SNRs, which helps test the pipeline capabilities and its detection limits. After the dirty cubes are produced, the morphological parameters, stored in the .csv parameter file, are used to measure further source properties of interest, such as the continuum, the SNR, and the total surface brightness. The sky model and dirty cubes can be produced both sequentially or in parallel through the *Slurm* workload manager [165] if a multi-node architecture is available. Fig. 4.1 and 4.2 show several examples of frequency-stacked clean and dirty cube pairs.

While we plan to improve our simulation code beyond its current capabilities, at this stage it does not model galaxies' internal kinematics, it does not include complex galaxy morphologies and it does not account for galaxy interactions. That being said, given the main scientific target of the present work, these parameters should not play an important role in shap-

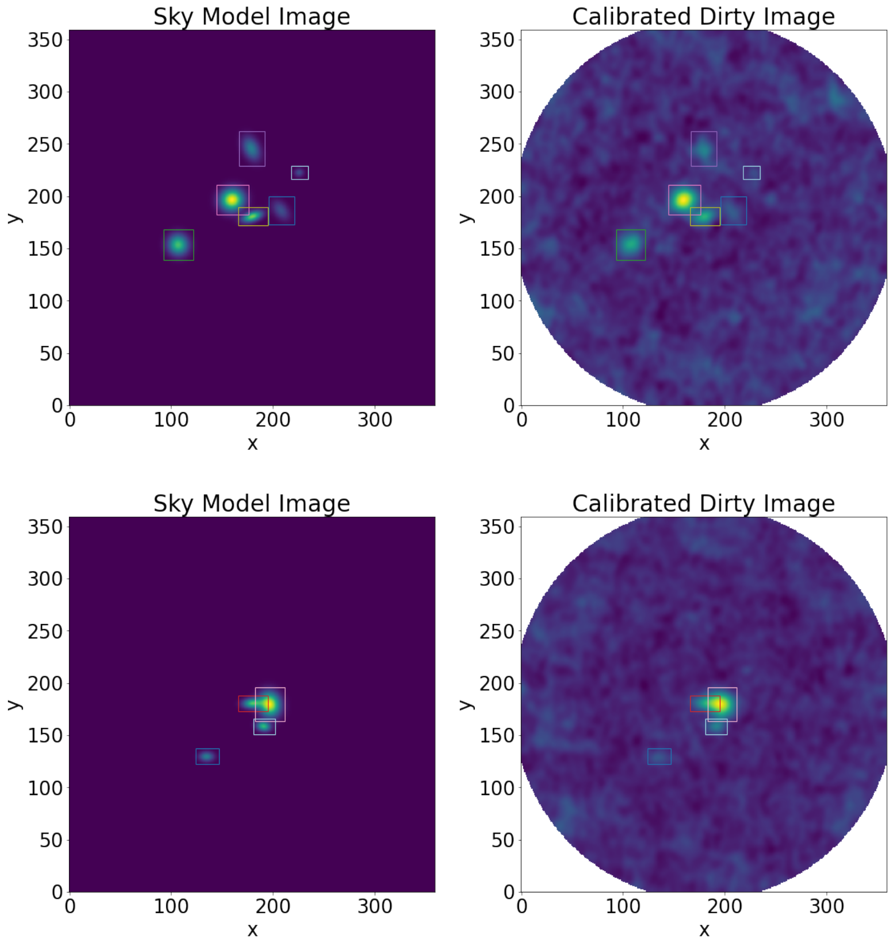

Figure 4.1. Several examples of frequency stacked dirty/clean cube pairs generated through our simulation code. Sources within the cubes are outlined with coloured bounding boxes.

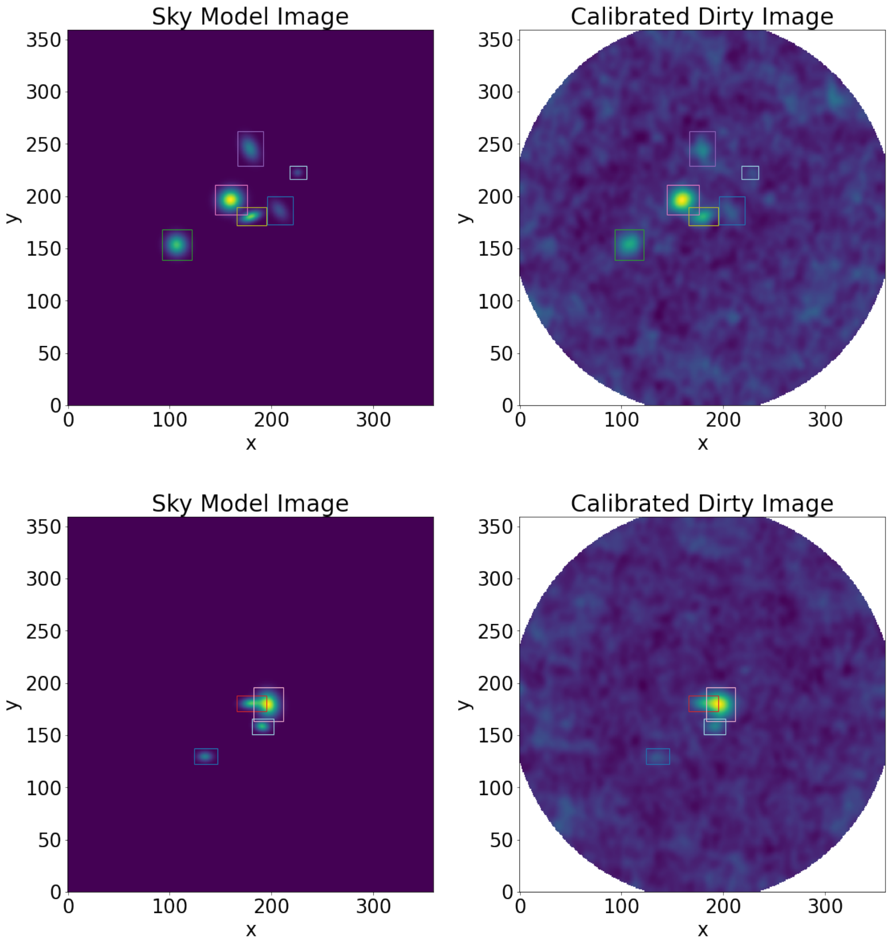

Figure 4.2. Several examples of frequency stacked dirty/clean cube pairs generated through our simulation code. Sources within the cubes are outlined with coloured bounding boxes.

ing the real target morphologies given that they are mostly unresolved. For this reason, we generate sources with a 2D Spatial Gaussian with an FWHM between 0.2 and 0.8 arcsec. Regarding the simulation of complex atmospherical and instrumental effects, the CLEAN *SimALMA* pipeline, that we employ to simulate ALMA observations of the sky models, should produce fairly realistic ALMA observations in case of point-like sources [148]. While it is out of the scope of this work to present the full simulation pipeline, it is our intention, given the relevance of comparing the performances of deep learning models on the same data, to make publicly available on GitHub the simulation code with several baseline datasets, such as the one we used to train and test our pipeline.

4.1.1 The Data

We generated $5,000$ simulated cube pairs containing $22,532$ simulated sources and randomly divided them into train, validation, and test sets using the rather usual 60%, 20%, 20% splitting criterium. We just remind the reader that the training set is used to train the DL models within the pipeline; the validation set is used to measure the training progress and assess generalization capabilities, and the test set is used to measure the pipeline performances in detecting sources and in regressing their parameters (see Sec. 4.2.4 and 4.2.5). The three sets contain respectively $13,512$, $4,465$, and $4,556$ simulated sources. The distributions of the source parameters are shown in Figs 4.3, 4.4, 4.5 ,4.8, 4.7, 4.6. Projection angles, positions, and extensions of sources are uniformly generated in their respective parameter ranges (see Tab. 4.2), while the SNR, Surface Brightness, and Continuum Brightness distributions confirm that we are generating a bright central source with a $SNR > 10$ surrounded by less bright serendipitous sources. The minimum and maximum flux densities generated are respectively 0.97 and 407.4 mJy/beam. The data were generated on the Infrastructure for Big data and Scientific Computing (IBISCO-HPC) (Infrastructure for Big data and Scientific Computing) at the University of Naples Federico II (IBISCO-HPC).

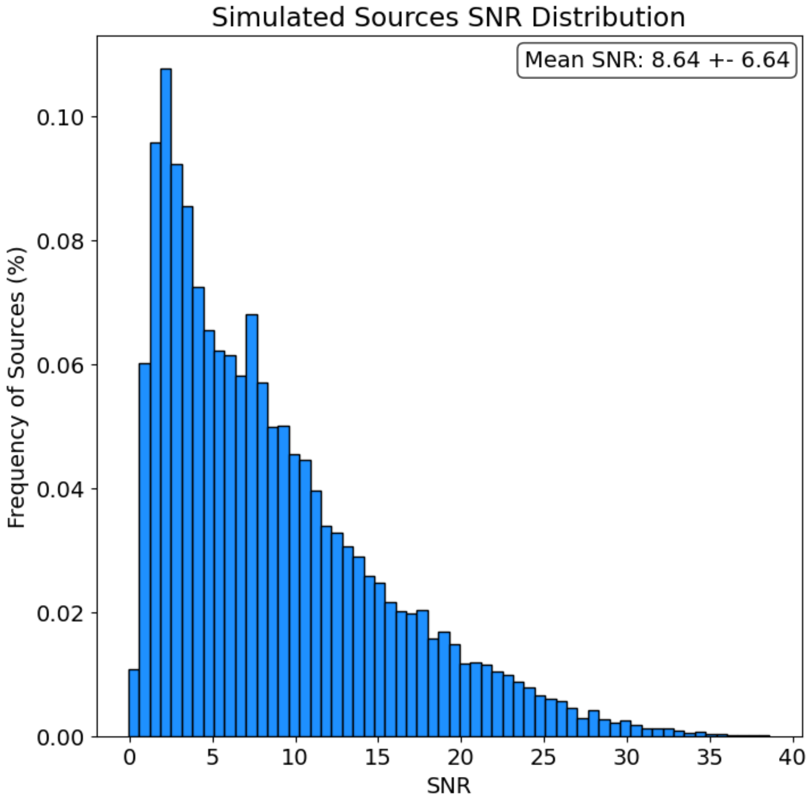

Figure 4.3. Distribution of the Signal to Noise Ratio of the simulated sources: fraction of simulated sources versus measured SNR (see Eq. 4.12). The box in the top right corner shows the mean SNR ± its standard deviation.

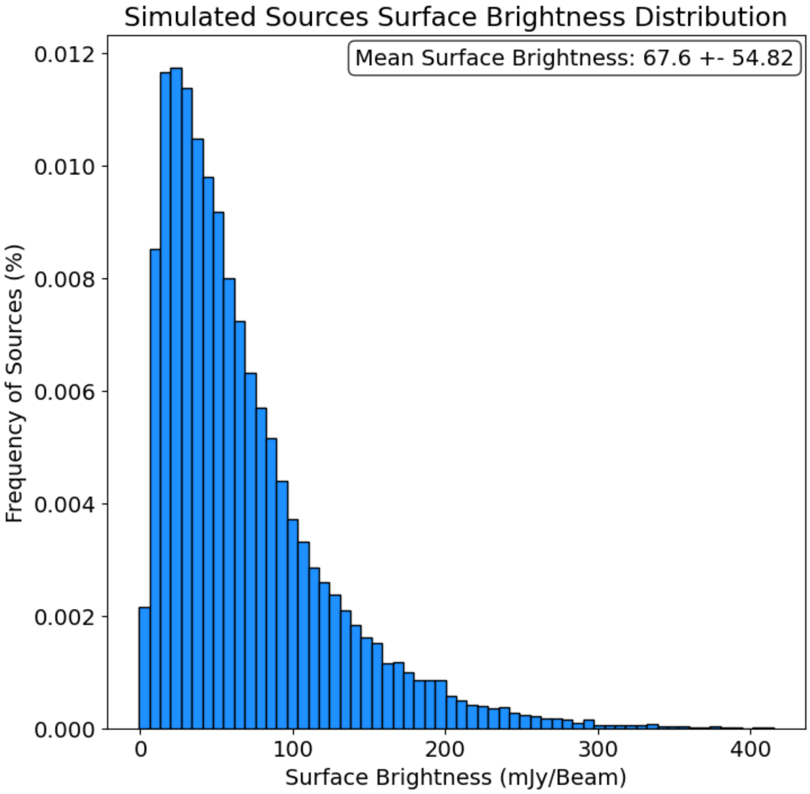

Figure 4.4. Distribution of the total brightness of the simulated sources. On the x-axis, the measured total brightness [mJy / beam] is obtained by summing the voxel values in the dirty cubes within sources bounding boxes. On the y-axis, the fraction of simulated sources is provided. The box in the top right corner shows the mean brightness ± and its standard deviation.

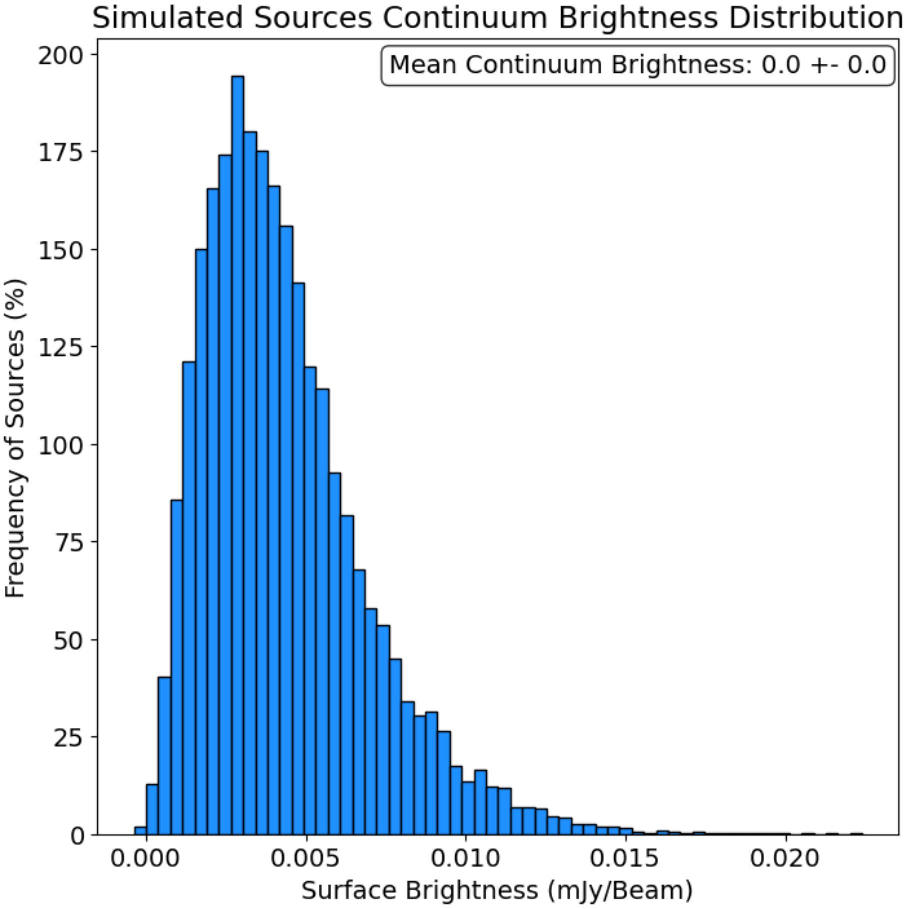

Figure 4.5. Distribution of the continuum mean brightness of the simulated sources. On the x-axis, the measured mean continuum brightness [mJy / beam] is obtained by selecting all voxels within the x and y limits of the sources bounding boxes but outside their boundaries in frequency $[z - fwhm_z, z + fwhm_z]$. On the y-axis, the fraction of simulated sources is shown. The box in the top right corner provides the mean continuum brightness ± its standard deviation.

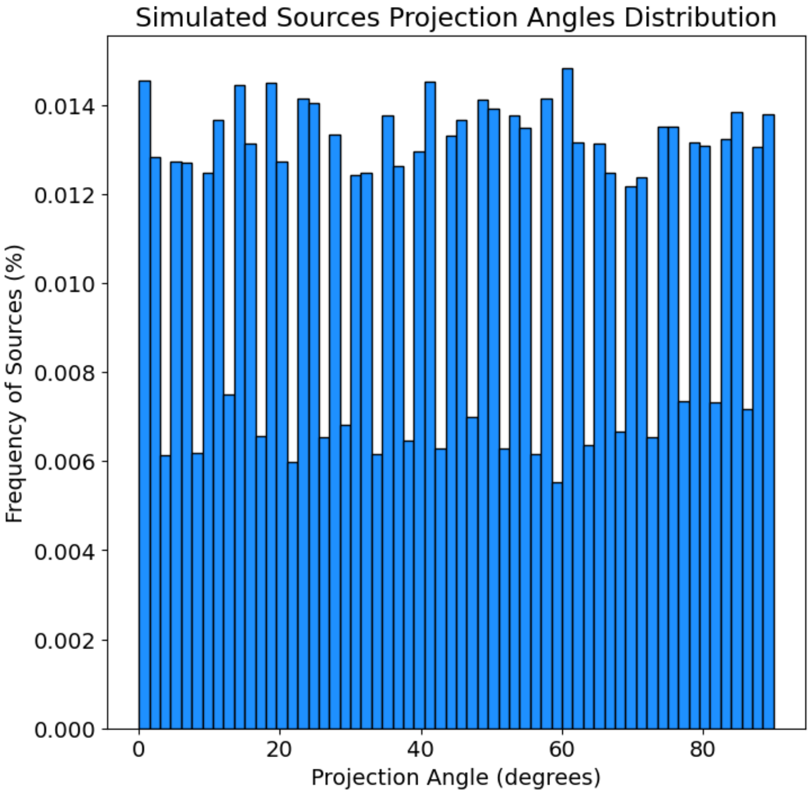

Figure 4.6. Distribution of the projection angles of the simulated sources: fraction of simulated sources versus projection angles in degrees.

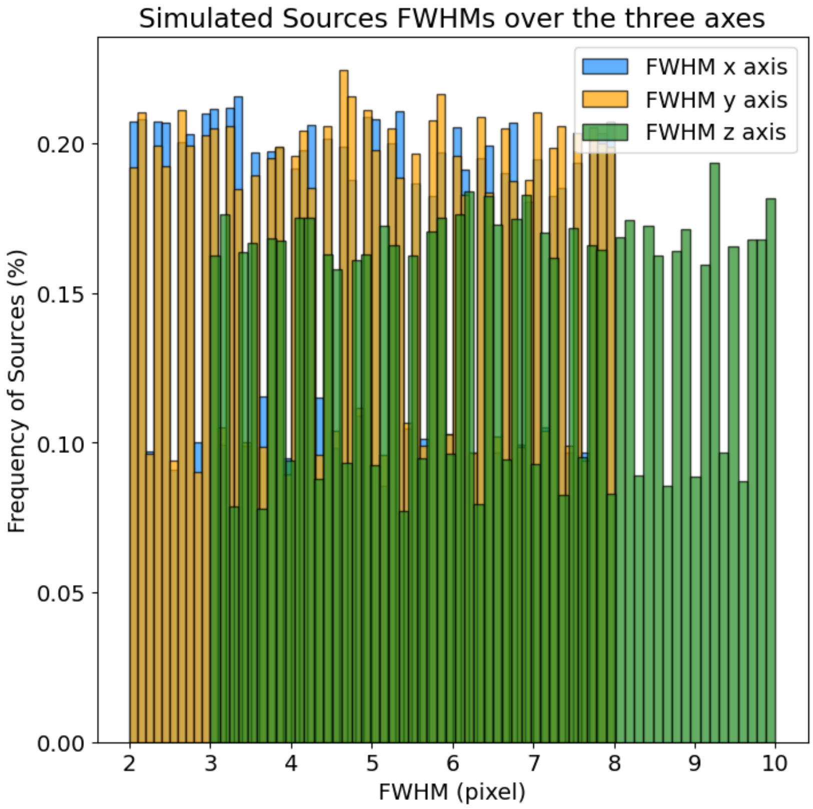

Figure 4.7. Distributions of the $FWHMs$ of the simulated sources. In blue, orange and green, the $FWHMs$ over the x-axis, the y-axis and the z-axis are provided. On the histogram, the x-axis and y-axis provide the $FWHMs$ values in pixels and the fraction of simulated sources, respectively.

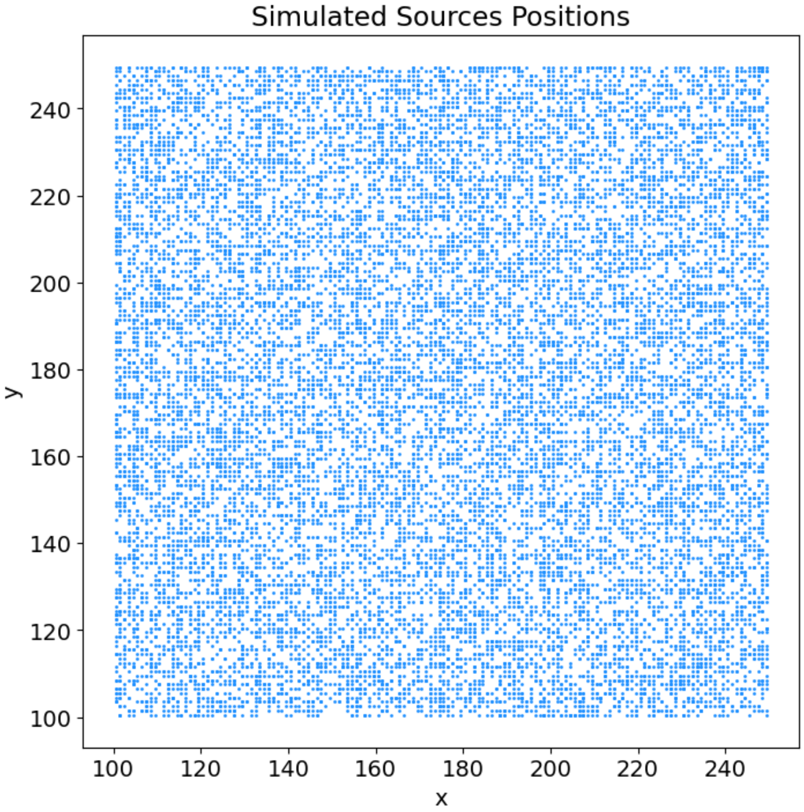

Figure 4.8. Scatter plot showing uniformity in the positions on the xy plane of the simulated sources.

4.2 The Deep Learning Pipeline

The Deep Learning pipeline we present in this work, as it will be more apparent in Sec. 4.2.4, can be described as a decision graph interconnecting six deep learning models, each one taking a specialised role in order to detect and characterise sources within the input dirty cubes. The types of architectures were chosen on the basis of their strengths: Convolutional architectures (Blobs Finder and ResNets) to process spatial information, and the Recurrent Neural Network (Deep GRU) to process sequential information. Before describing the flow of data within the pipeline, we shortly describe the individual DL models. Note that all our models were implemented through the PyTorch library [115].

4.2.1 Blobs Finder

Blobs Finder is a 2D Deep Convolutional Autoencoder trained to solve the image deconvolution problem

$$D[x,y] = P[x,y] \times M[x,y] + N[x,y] \tag{4.4}$$

where $D[x,y]$ is the integrated dirty cube produced integrating along the frequency the dirty cube, $P[x,y]$ is the dirty PSF, and $N[x,y]$ is the combination of all noise patterns in the data. $M[x,y]$ is the reconstructed and denoised integrated sky image. To guide the autoencoder towards the best possible solution, two factors are critical [50]: the preprocessing and augmentation of input and target variables, and the choice of the loss function used to measure the error between the target variable and the Autoencoder's prediction. In our case, Blobs Finder is trained with the integrated dirty cubes as inputs, and the sky model images as targets. Both input and target image are normalized to the $[0,1]$ range, which helps with the training process and allows us to make a probabilistic interpretation of the Autoencoder's output. Blobs Finder architecture, as shown in Fig. 3.1, is that of a Convolutional Autoencoder [103]. The Encoder is structured by four convolutional blocks that progressively reduce the spatial dimension of the input while increasing the number of channels or feature maps. More in detail, each block contains a 2D Convolution layer with stride 2 and a kernel size of 3, a Leaky Rectified Linear Unit Rectified Linear

Unit (ReLU) activation function and a 2D Batch Normalization layer followed by a 2D Convolution layer with a stride of 1 and a kernel size of 3, another Leaky ReLU activation and a 2D Batch Normalization Layer. In this way, each block halves the spatial extent of the input and doubles the number of channels. After the convolutional blocks, there is the final fully connected layer.

The Decoder has a symmetric architecture, i.e. a fully connected layer, followed by four deconvolutional blocks and a final identity layer which is constituted by a 2D Convolution followed by a Sigmoid activation function. Each deconvolutional block is constituted by a 2D Bilinear interpolation with stride of 2, followed by a Leaky ReLU activation function and a 2D Batch Normalization layer (upsampling block), a 2D Transposed Convolution layer with a stride of 2 and a kernel size of 3, followed by a Leaky ReLU activation function and a 2D Batch Normalization layer (learnable upsampling block). The output of the upsampling block and learnable upsampling block are then concatenated and passed along to a convolutional block constituted by a 2D Convolution layer with stride 2 and a kernel size of 3, a Leaky ReLU activation function and a 2D Batch Normalization layer followed by a 2D Convolution layer with a stride of 1 and a kernel size of 3. The bilinear upsampling operation transforms the input layer in the desired spatial resolution without using any parameters and the resulting features should contain most of the information of the original features. These upsampled features are concatenated to the output of the parametric upsampling performed through the Transposed Convolution in order to create a residual-like connection. This additive upsampling [161] should improve prediction capabilities and remove gridding artefacts produced by the subsequent Transposed Convolutions. The final block is a 2D Convolution layer with a stride of 1 and a kernel size of 1, followed by a Sigmoid activation function. The scope of this final layer is to normalize its input to the $[0, 1]$ range. A detailed description of the spatial transformation performed by Blobs Finder can be seen in Tab. 4.3. To train Blobs Finder, we selected as a loss function the weighted combination of two well-known losses in the DL image reconstruction and denoising framework: the l_1 loss and the Structural dissimilarity loss Difference of Structural Similarity (DSSIM) which is based on Structural Similarity Index measurement.

Block Name	Input Size	Output Size
Conv Block 1	$[b, 1, 256, 256]$	$[b, 8, 128, 128]$
Conv Block 2	$[b, 8, 128, 128]$	$[b, 16, 64, 64]$
Conv Block 3	$[b, 16, 64, 64]$	$[b, 32, 32, 32]$
Conv Block 4	$[b, 32, 32, 32]$	$[b, 64, 16, 16]$
FC 1	$[b, 64 \times 16 \times 16]$	$[b, 1024]$
FC 2	$[b, 1024]$	$[b, 64 \times 16 \times 16]$
DeConv Block 1	$[b, 64, 16, 16]$	$[b, 32, 32, 32]$
DeConv Block 2	$[b, 32, 32, 32]$	$[b, 16, 64, 64]$
DeConv Block 3	$[b, 16, 64, 64]$	$[b, 8, 128, 128]$
DeConv Block 4	$[b, 8, 128, 128]$	$[b, 1, 256, 256]$
Final Block	$[b, 1, 256, 256]$	$[b, 1, 256, 256]$

Table 4.3. Input and Output shapes for each layer of Blobs Finder, where b indicates the batch size, and the horizontal line separates the Encoder from the Decoder network.

The losses are mathematically defined as follows:

$$l_1(x, y) = MAE(x, y) = mean([l_1, ..., l_N]^T) \tag{4.5}$$

where N is the number of pixels in the images and

$$DSSIM(x, y) = \frac{1 - SSIM(x, y)}{2} \tag{4.6}$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.7}$$

$$c_1 = (k_1 L)^2 \tag{4.8}$$

$$c_2 = (k_2 L)^2 \tag{4.9}$$

$$L = 2^{precision} - 1 \tag{4.10}$$

where μ_x and μ_y are the averages of x and y, σ_x^2 and σ_y^2 are their variances, σ_{xy} the covariance, c_1 and c_2 two variables needed to stabilize the division with small denominator values, and, finally, L is the dynamic range of the pixel values. In our case, the images are stored in single-precision floating-point format and thus $L = 2^{32} - 1$ and the two constants are respectively

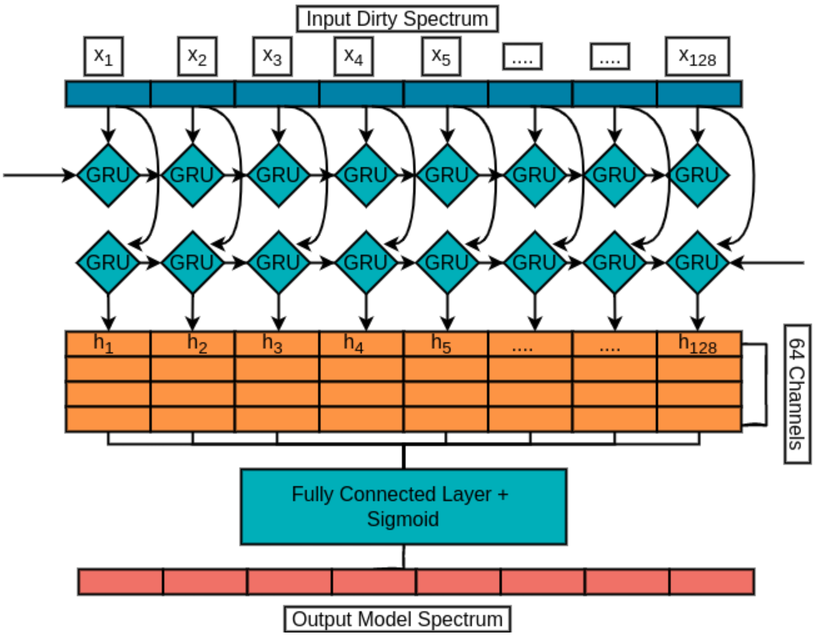

Figure 4.9. Deep GRU's architecture is constituted by two layers of GRUs followed by a Fully connected layer and a Sigmoid activation function.

$k_1 = 0.01$ and $k_2 = 0.03$.

The choice of these two losses was empirically determined by trial and error in our experiments, while the remaining hyperparameters such as the hidden layer size, the number of channels in each Block, and the choice of activation function were optimized through a grid-search strategy. More information about how these losses are used and their weighting during training is outlined in Sec. 4.2.5.

4.2.2 Deep Gated Recurrent Unit (GRU)

Deep Gated Recurrent Unit (GRU) is a Recurrent Neural Network (RNN) [130] constructed by combining together two layers of GRUs and a fully connected layer, as shown in Fig. 4.9. For a detailed description of GRUs see Chapt. 3 Sec. 3.1.4. In our case, we want to use the Deep GRU to solve the denoising problem

$$Y[z] = X[z] + N[z] \quad \text{with } z \in [1, 128] \tag{4.11}$$

where z is the frequency index in the cube, $Y[z]$ are noisy galaxy spectra, $X[z]$ are the underlying emissions, and $N[z]$ are the various noise components. When a spectrum is fed to the network, each hidden state at frequency step t is passed to both the next frequency step of the current layer and the current frequency step of the next layer. Because there is no preferable frequency direction in the data, the two layers pass along information in opposite directions: one from low frequencies to high frequencies, and the other in the opposite direction. In our implementation, each layer of GRUs outputs 32 hidden states (or channels) which are then concatenated to form a latent vector of size $[b, 64 \times 128]$ before being fed to a fully connected layer. The layer transforms its input into a vector with the same size as the input signal and then a Sigmoid activation function is applied to normalize it to the $[0, 1]$ range. As a loss function, we use again the l_1 loss (see Eq. 4.5).

4.2.3 ResNet

Residual Neural Networks [64] are a class of Deep Convolutional Neural Networks (CNNs) which implements skip connections between adjacent layers in order to avoid the problem of vanishing gradients [141] by easing the process of information flow in the layers and to mitigate the degradation problem, a phenomenon for which the more layers are added to a CNN the more training error and instability increase. The basic component of a ResNet is the so-called Residual Block outlined in Fig. 4.10.

Our implementation of the ResBlock depends on whether spatial downsampling is applied to the input or not. If spatial downsampling is needed, the input is processed via two convolutional blocks constituted by a 2D Convolutional Layer with kernel size of 3 and a stride of 2 (which downsamples the input), a ReLU activation function and a 2D Batch Normalization layer. While the first convolutional block increases the number of channels by a factor of two, the second maintains constant the number of channels. In parallel, the input is also processed by a single convolutional block, and the two outputs are concatenated together (skip-connection) before being fed to a final ReLU layer. In the other case, if downsampling is not necessary, the input is processed via two convolutional blocks with a stride of 1 (which thus does not downsample the input) and is concatenated with the unaltered input (skip-connection) before being fed to the final ReLU.

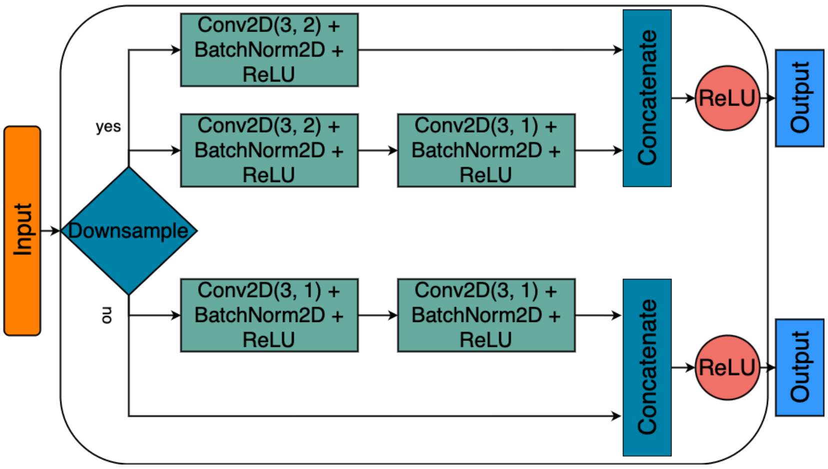

Figure 4.10. Architecture of a Residual Block, the basic component of a ResNet. The architecture is divided into two main pathways depending if downsampling must be applied in the layer. In the affirmative case, the 2D Convolutions are applied with a kernel size of 3 and with a stride of 2 or 1. As it can be seen the output of the previous layer is brought forward through a skip-connection and concatenated with the output of the current layer before applying the final activation function.

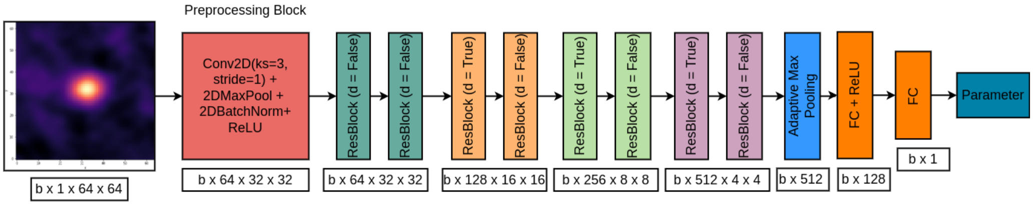

Figure 4.11. Our implementation of the ResNet architecture. The input image is first preprocessed by a 2D Convolution layer, followed by 2D Max Pooling, 2D Batch Normalization, and a ReLU activation function, and then is forwarded through four blocks of two Residual Blocks (see Fig. 4.10). The output is then processed via an Adaptive Max Pooling layer and fed to two fully connected layers which map the latent vector of 512 elements to a single scalar (the value of the parameter of interest for the ResNet).

The ResNet architecture, which is shown in Fig. 4.11, is constructed by a first convolutional block that performs 2D Max Pooling followed by four ResBlocks which gradually spatially downsample the input image of size $[b, 1, 64, 64]$ to $[b, 512, 4, 4]$ where 64×64 are the spatial dimensions of the input image, b is the batch size, 1 is the initial number of channels and 512 is the number of feature maps which are fed to the final MLP. Average Max Pooling is applied to collapse the spatial dimensions to a single vector which is then fed to two fully connected layers interconnected by a ReLU activation function (MLP). This final layer outputs a single scalar value, i.e. the predicted parameter of interest. In other words, the first part of the network extrapolates a vector of 512 features from the input image, which is then fed to a Multi-Layer Perceptron (MLP) that makes the functional mapping between the features space and the target parameter space. In our pipeline, we use several ResNets; each one specialized in solving the regression of specific morphological parameters of the input source. Technically, these parameters could be regressed at the same time by a single ResNets with an output vector of size m instead of a scalar (where m is the number of parameters one wants to regress). Given that the dynamical range of the parameters varies a lot from one parameter to the other (see Sec. 4.1.1), we expect that such a general network would show lower performances with respect to the specialized counterparts. This expectation was empirically confirmed by us on a trial and error base. As a loss function to train the ResNets, we used again the l_1 loss (see Eq. 4.5). The architectural choices such as the number of convolutional layers within a Residual Block, the number of Blocks, the choice of activation function, the kernel sizes, the number of channels and the number of extrapolated features were optimized to improve performance. First, we optimize for the number of Residual layers and depth of those layers by fixing all other parameters, and then, once the best ResNet architecture has been found, we performed the optimization of the remaining parameters through a grid-search strategy. In fact, by changing the number of residual blocks and the depths (i.e. number of convolutions within each Block) and by introducing bottlenecks (i.e. a chain of a 1×1 convolution, followed by a 3×3 convolution and a 1×1 convolution) several ResNet architectures can be reproduced and tested (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152). All these architectures share the same first convolutional block and the same final MLP and are only distinguished by the number and depth of the central layers. We tested the performance of each network on the flux parameter estimation by training each one of them for 50 epochs with the Adam optimizer and a fixed kernel size of 3. To register the models' performances, we measure the average l_1 loss on the entire test set. No meaningful difference in performance was registered between the five tested architectures and thus our choice fell upon the lighter and faster of them, i.e. the one showcased in Fig. 4.11. Once the architecture was chosen, the remaining hyperparameters were fine-tuned through a grid-search strategy.

4.2.4 The Pipeline

The overall objective of the pipeline is to detect and fit the sources within calibrated ALMA image cubes. A full overview of the pipeline can be seen in Fig. 4.12, where the arrows outline the flow of data in the pipeline. The order in which the operations are performed can be understood by the numbering (in orange). As already explained, the pipeline can be divided into six logical blocks: 2D source detection, frequency denoising, emission detection, source focusing, morphological parameters estimation, 3D model construction and flux density estimation. To relate the logical blocks to the pipeline schema shown in Fig. 4.12, in the following explanation, we mark the logical blocks with the corresponding numbers as shown in Fig. 4.12. To ease the logical flow of the pipeline, we assume that all the DL models have been trained to act as simple, functional maps between their inputs and outputs. In Sec 4.2.5 we go over the details of how each Deep Learning model is trained within the pipeline. All the images shown in this section are relative to the Test set; more examples and a detailed description of the pipeline performances are described in Sec. 4.3.1.

The flow of an input dirty cube in the pipeline goes as it follows:

1. **2D Sources Detection (1 - 4)**: the image cube is normalized to the $[0, 1]$ range and integrated along frequency to create a 2D image. We refer to this image as the **integrated dirty cube** (1). The integrated dirty cube is first cropped from the centre to a shape of $[256, 256]$ pixels which removes the edge of the images characterized by a low SNR (See Sec. 4.1), then it is normalized to the $[0, 1]$ range (2) and,

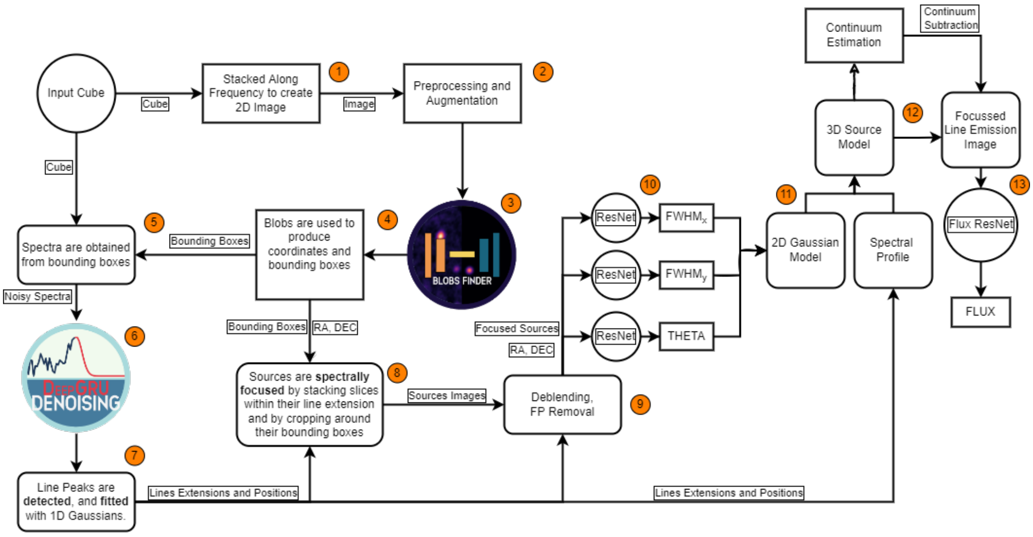
Input Cube
Cube
Stacked Along Frequency to create 2D Image
1
Image
Preprocessing and Augmentation
2
Cube
BLOBS FINDER
3
Blobs are used to produce coordinates and bounding boxes
4
Bounding Boxes
Spectra are obtained from bounding boxes
5
Noisy Spectra
DeepGRU DENOISING
6
Bounding Boxes
RA, DEC
Sources are spectrally focused by stacking slices within their line extension and by cropping around their bounding boxes
8
Sources Images
Deblending, FP Removal
9
Focused Sources
RA, DEC
Line Peaks are detected, and fitted with 1D Gaussians.
7
Lines Extensions and Positions
Lines Extensions and Positions
ResNet
FWHM$_x$
ResNet
FWHM$_y$
ResNet
THETA
10
2D Gaussian Model
11
Spectral Profile
3D Source Model
12
Continuum Estimation
Continuum Subtraction
Focussed Line Emission Image
Flux ResNet
13
FLUX

Figure 4.12. The full pipeline schema for source detection and characterisation within the input ALMA dirty data cube (top left of the schema). Numbers show the logical flow of the data within the pipeline. The pipeline takes as input an ALMA dirty data cube, performs source detection and false detection removal, and then proceeds to characterize the detected sources. A detailed explanation of all the steps outlined in the schema is given in 4.2.4.

finally, it is fed to the first DL model **Blobs Finder** (Sec. 4.2.1). The autoencoder processes the image and predicts a **2D probabilistic map** of source detection (3). A hard thresholding value of 0.15 is used to binarize the probabilistic map and then the *scikit-learn* [118] *label* and *regionprops* functions are used to extract bounding boxes around all blobs of connected pixels (or source candidates) (4). The thresholding value is chosen to be 0.15 in order to peak all the signals detected by Blobs Finder, while excluding small fluctuations in the background. The exact thresholding value was empirically determined in order to strike a compromise between the number of false detections and the percentage of the detected signal. Figures 4.13, 4.14, and 4.15 show, respectively, an example of an input integrated dirty cube containing 6 simulated sources (outlined by green bounding boxes and two of which are spatially blended), the target sky model image (with in green the target bounding boxes and in red the predicted bounding boxes extracted through the thresholding of the predicted 2D probabilistic map), and the 2D prediction map with the predicted bounding boxes highlighted in red.

2. **Frequency Denoising and Line Detection** (**5 - 7**): bounding boxes around source candidates are used to extract **dirty spectra** from the input cube. The spectrum for each source candidate is extracted by adding the pixels inside its bounding box over all the 128 frequency slices of the cube. The spectra of the detected source candidates are extracted from the cube, standardized, i.e. rescaled to have null mean and standard deviation with unity value, and then fed to **Deep GRU** (Sec. 4.2.2). The Deep Gated Recurrent Unit denoises the standardized spectra and outputs 1D probabilistic maps of source emission lines or **cleaned spectra** (6). The cleaned spectra are then analysed with the *scipy* [157] *find_peaks* functions with a threshold value of 0.1 in order to detect emission peaks. Each peak is then fitted with a 1D Gaussian model through the *astropy* [9] *models.Gaussian1D* function. As fitting algorithm, we employ the *LevMarLSQFitter* algorithm, and as the initial values for the mean and amplitude of the 1D Gaussian model, we use, respectively, the previously detected peak position and its relative amplitude. All detected Gaussian peaks' positions over the frequency axis (z) are

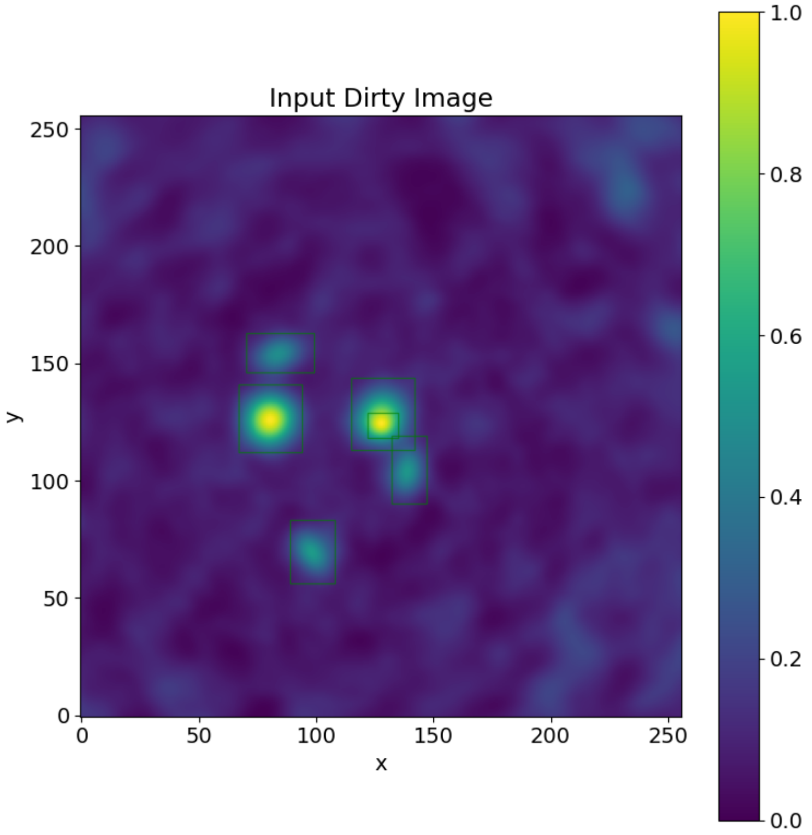

Figure 4.13. An example of Blobs Finder's input 2D integrated dirty cube produced by integrating an input dirty cube over the entire frequency range. Superimposed in green, are the target bounding boxes outlining the emissions of the 6 sources present in the cube. The image contains an example of two spatially blended sources located around the centre of the image, one is a bright point-like source, the other a fainter and diffuse source laying behind.

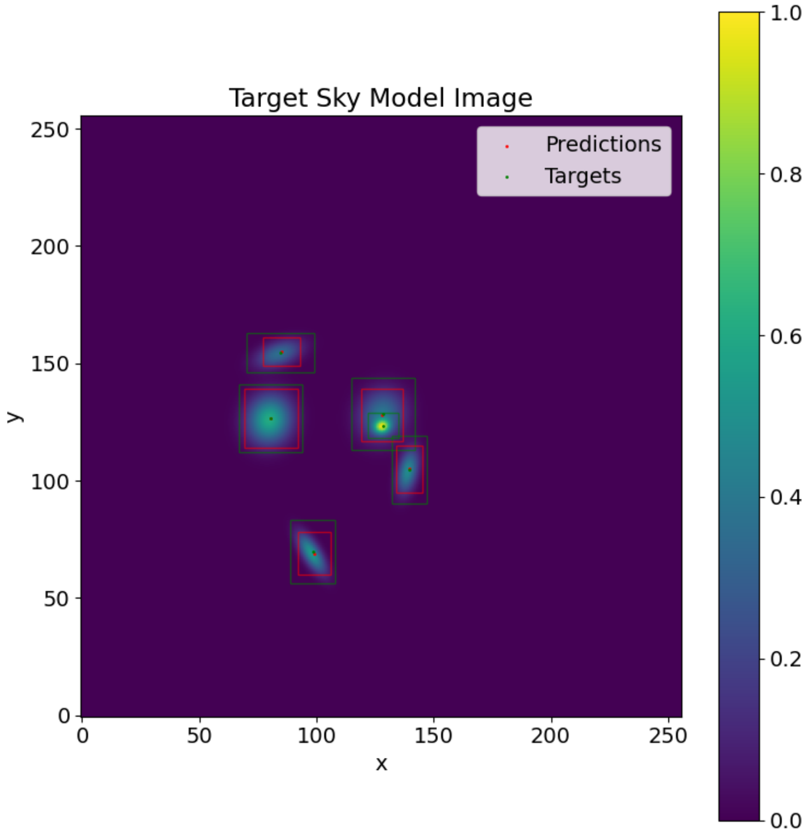

Figure 4.14. An example of Blobs Finder's target 2D Sky Model image with the target bounding boxes highlighted in green and the predicted bounding boxes extracted through the thresholding operation on Blobs Finder's probabilistic output, highlighted in red. Predicted and true bounding box centres are also plotted as, respectively, red and green dots.

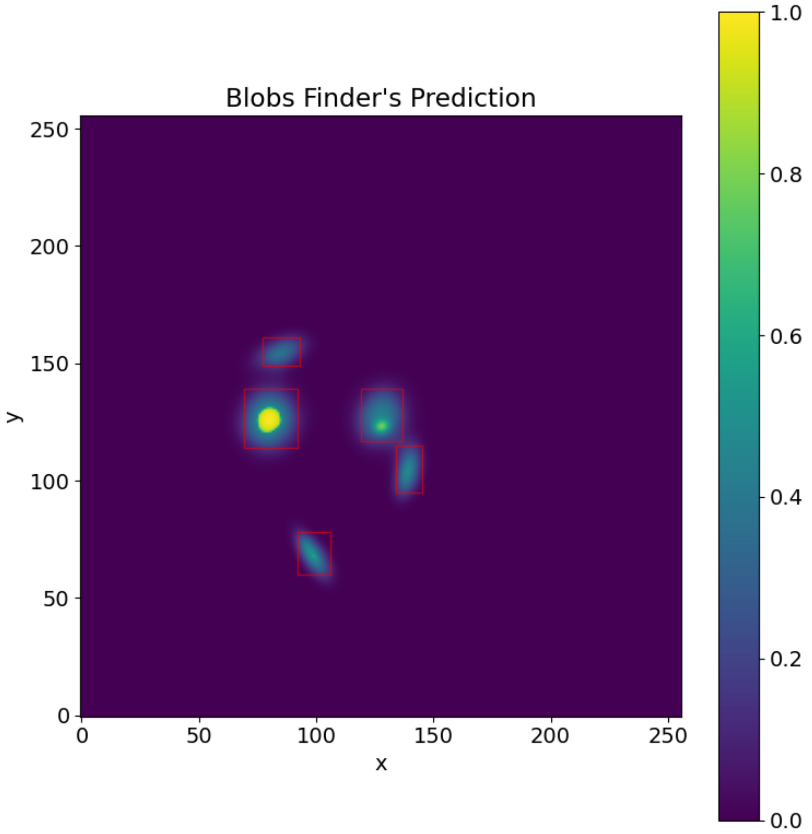

Figure 4.15. An example of Blobs Finder's output 2D probabilistic source detection map with the predicted bounding boxes extracted through thresholding, highlighted in red.

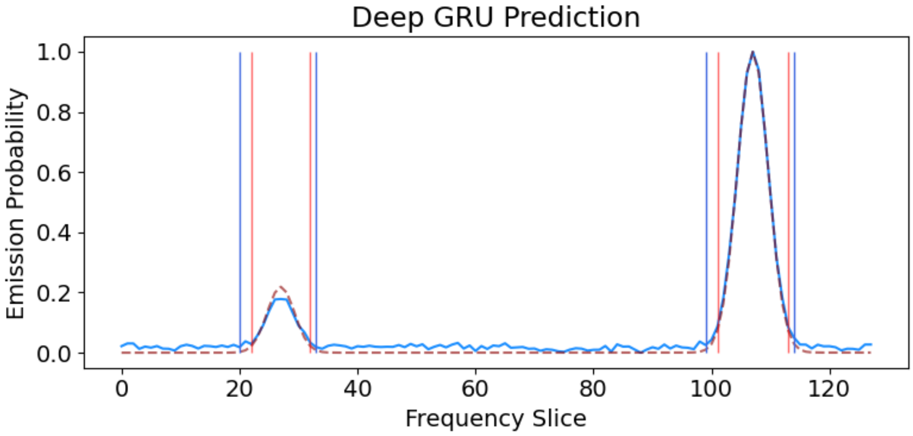

Figure 4.16. In blue the dirty spectrum extracted from the central source bounding box predicted by Blobs Finder (Fig. 4.14), in dotted-red the Deep GRU's prediction. Vertical blue bars delimit the true emission ranges, while red bars are the predicted emission ranges. A secondary fainter source emission peak is detected by Deep GRU and thus the source is flagged for deblending.

recorded alongside their extensions Δ_z defined as $\Delta_z = 2 * FWHM_z$ where $FWHM_z$ is the FWHM of the Gaussian peak (7). In order to detect possible false positives produced by Blobs Finder, all potential candidates showing no meaningful peak in their spectra are removed. If more than one peak is found alongside the spectrum, the candidate is likely the superimposition of blended sources and thus it is flagged for deblending. All the remaining detections are then passed to the next stage. The peaks Δ_z are used to cut out the emission regions from the dirty spectra, which are fitted with a simple linear model implemented through the *astropy models.Linear1D* function in order to take into account any possible trend of the continuum with frequency. Fig. 4.16 shows the dirty spectrum extracted from the two blended sources showcased in Fig. 4.15, and the Deep GRU's predicted emission probability map. Blue vertical bars limit the true emission ranges of the two sources within the spectrum, while red ones limit the predicted emission ranges.

3. **Source Spectral Focusing** (**8 - 9**): this phase has three main objectives: (i) to feed to the ResNets the best possible input image

of the potential candidates in order to ease as much as possible the morphological parameters detection; (ii) to deblend the sources, and (iii) to remove false detections. Regarding the first objective, the idea is to increase as much as possible the Signal to Noise Ratio (SNR) of the source in the image by cropping a $[64, 64]$ pixel box around its bounding box and integrating it only within its peak FWHM. This operation is what we call *spectral focusing* and the resulting image is the **spectrally focused image**. In order to estimate the SNR of a source, we introduce two SNR measurements:

- **Global SNR**: we define the Global SNR as:

$$SNR = \frac{median(x_s(r))}{var(x_n(R-r))} \tag{4.12}$$

  where $x_s(r)$ are the values of the source pixels contained within the circumference of radius r that inscribes the source bounding box, and $x_n(R-r)$ are the pixel values within an annulus of internal radius r and external radius R which has the same area of the inscribed circumference;

- **Pixel SNR**: we define the Pixel SNR as:

$$snr = \frac{x_i}{var(X)} \tag{4.13}$$

  where x_i is the value of the given pixel, and $var(X)$ is the variance computed on the full image.

The two SNR measurements are used respectively to distinguish false detections from true sources and to deblend overlapping sources within a blob. Fig. 4.17 summarises the false positive detection pipeline which works in the following way: if a potential source has a Global SNR higher than 6 (empirical bright source SNR threshold) in the integrated dirty cube and was not flagged for deblending (due to the presence of multiple detected peaks in the extracted spectrum), it is first focused and then passed along to the next stage of the pipeline; if it has a SNR lower than 6 in the integrated dirty cube, but there is an increase in the Global SNR when focused, it is also passed along to the next stage, otherwise, it is discarded as a false detection (a condition marked with 1 in Fig. 4.17). The latter case, in fact, could only happen if the source is integrated outside its true emission peak, for example over a noise spike. If more than one peak is found in the potential source spectrum, then there is a chance that multiple blended sources make the blob (detected by BlobsFinder). First, the source is focused on the highest peak (primary peak) by integrating within its extension and the Global SNR calculation is made to understand if the potential source must be discarded (see the previous step). Also, the Pixel SNR measurement is used to find the highest SNR pixel in the image $p(x, y)$, which will act as a reference for the next phase of the deblending process. The potential source is then focused around the secondary peaks. For each of these, if the peak is not superimposed in frequency with the brightest peak, then by integrating within the extension of the secondary peak, the brightest source should disappear from the focused image (because it is integrated outside its emission range), and the pixel SNR measurement is used to find the highest SNR pixel in the image $s(x, y)$. If this pixel is different from the previously found reference pixel (their distance is higher than a user defined number of pixels, in our case 3), then the neighbouring pixels around this pixel are linked with a friend of friends algorithm in an iterative manner. For each iteration, the Global SNR is measured. Pixels are added in this fashion until a plateau or saturation level in the Global SNR is reached. A bounding box is then created in order to encompass all the selected pixels, and a $[64, 64]$ pixel image is cropped around the bounding box. If instead, the secondary peak overlaps the primary peak in frequency, and the two pixels coincide spatially ($p(x, y) = s(x, y)$), then the secondary peak is discarded as a false detection (i.e. not considered an independent source with respect to the primary, a condition marked as 2 in Fig. 4.17). If an increase in SNR is recorded by integrating over the multiple peaks, then the secondary peak is deemed as part of the primary source and the source emission range is extended accordingly, otherwise, it is discarded as a noise spike (false detection). The first condition may happen if DeepGRU mistakenly predicts a single true emission peak as two separate peaks or if Blobs Finder predicts a single true source as two very close blobs, while the latter if DeepGRU overpredicts the true emission range. Finally, all spectrally focused sources with a global SNR lower than 1 are flagged and removed (9).

The deblending method assumes that all sources can be fairly approximated as 2D Gaussians (in the spatial plane) with a single emission peak in frequency (1D Gaussian), and thus it will need to be modified whenever our galaxy models will also simulate the velocity dispersion and inclination along the line of sight. The introduction of these parameters in our simulations would, in fact, create more complex spectral profiles, for example, a source with a high dispersion or inclination along the line of sight could show multiple peaks in the spectrum. In this case, the simple logic that we have described above will need to be revised.

Fig 4.18 showcases the results of the Spectral Focusing of the potential sources detected by Blobs Finder and Deep GRU in the test cube already displayed in Fig 4.13. By focusing on the two peaks detected by the Deep GRU (Fig 4.16, the two blended sources produce two different images (Focused Source 0 and 1) which now can be analysed independently. It is also worth noticing that by focusing the source within its effective emission range, the signal from other sources is suppressed and noise variation is minimised resulting in a higher SNR than that registered in the reference dirty integrated image.

4. **Morphological Parameters Estimation** (**10-11**): the spectrally focused images are normalized to the $[0, 1]$ range and then fed to three specialized ResNets (Sec. 4.2.3). Each ResNet is specialized in predicting a given morphological parameter of the source in the image. Namely: the FWHM in the x and y directions and the projection angle (pa). The first two parameters are predicted in pixel values, while the angle is directly predicted in degrees (10) measured in a clockwise fashion. The x and y positions of the source are computed as the photometric barycenters (pixel-weighted centres) of the Blobs Finder predicted bounding boxes. The predicted parameters are then used to generate a 2D model of the source, which is a 2D Gaussian generated at the source position with the predicted mor-

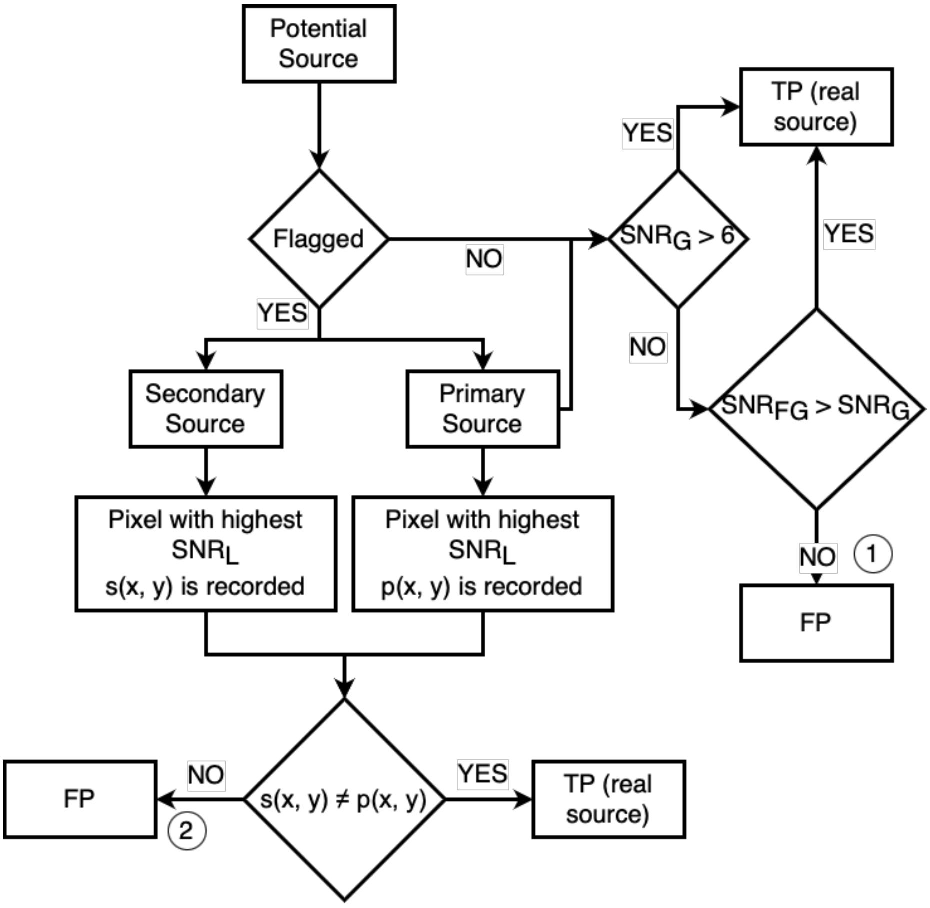

Figure 4.17. Schema of the False Positives detection and source deblending pipeline which constitutes step 9 of the source detection and characterisation pipeline (Fig. 4.12). The numbers 1 and 2 mark the two possible conditions which may lead to a potential source being defined as a false positive detection and thus discarded from further analysis. The under-script FG (focused global) indicates that the global SNR is measured on the focused source, while L implies a (local) Pixel SNR measurement. With *flagged* we indicate that multiple peaks are detected within the potential source's spectrum and thus is flagged for deblending. For an in-detail description of this schema see Sec. 4.2.4 (iii) Source Spectral Focusing.

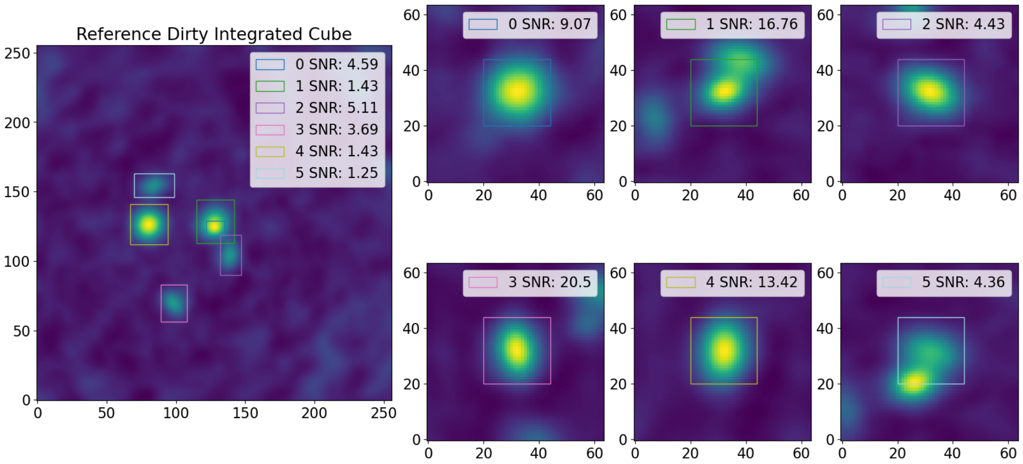

Figure 4.18. An example of source spectral focusing of sources within a test set image. On the Left, as a reference, we plot the dirty integrated cube with the predicted 2D bounding boxes obtained by Blobs Finder highlighted in different colours. The legend matches the source number to the bounding box colour in the image and the measured Global SNR (see Eq. 4.12). On the right, there are the 6 Spectrally Focused images obtained by integrating over the predicted line extensions found by the Deep GRU and cropping a [64, 64] pixel image around Blobs Finder's predicted bounding boxes centres. In each focused image it is also showcased the measured Global SNR. As can be seen there is a substantial increase in SNR when sources are focused around their actual emission ranges.

phological parameters;

5. **3D Model construction** (**12**): having fitted the source both in frequency and in the image plane, we combine the two Gaussians to create a 3D profile of the source, which then is converted into a 3D segmentation map, i.e. a binary cube with the same shape as the input dirty cube with all pixels that belong to the source set to 1 and the rest set to 0. To account for the fact that the convolution with the dirty beam spreads both the continuum and the line emission in the image, we dilate the segmentation map by a factor of 1.5. This is performed to make sure that all the source signal is contained within the 3D segmentation map.

6. **Flux Estimation** (**13**): the dilated 3D segmentation map is then used to create the model-masked cube by multiplying it with the dirty cube. The model-masked cube is, therefore, a version of the dirty cube in which all the pixels outside the source 3D model are set to zero. The inverse mask is instead used to capture the continuum cube of shape $[128 - \Delta_z, 256, 256]$ where Δ_z is the size of the segmentation mask in frequency channels. The continuum image is then created by averaging the continuum cube in frequency and the line emission cube is created through the following formula:

$$L_z[x, y] = M_z[x, y] - f(z) * C[x, y] \quad \text{with } z \in \Delta_z \tag{4.14}$$

   where $L_z[x, y]$ it the 2D line emission image at slice z, $M_z[x, y]$ is the model masked 2D image at slice z, $C[x, y]$ is the continuum image and $f(z)$ is the 1D continuum model. The line emission cube is integrated along the frequency to create the line emission image which is fed to a specialized ResNet predicting the source flux density in mJy/beam.

4.2.5 Training Strategies

In the pipeline, the data flows from one DL model to the next. In particular, the dirty spectra extracted from Blobs Finder predicted model images are the inputs of Deep GRU; the outputs of Deep GRU - the denoised spectra - are combined to produce the *focused* images used as input for the parametric ResNets, and the 3D models constructed with the ResNets parameters prediction are used to produce the line images, which are the input of the flux ResNet. Since we work with simulated data, both the true model images and the parameters of sources within them are known to us and thus all models within the pipeline can be trained at the same time on different GPUs by preemptively using the true source parameters to extract from the cubes the needed spectra and spectrally focused images, and the line emission images to train respectively Deep GRU and the ResNets. The problem with this training strategy is that it would not take into account the fact that Deep GRU and the ResNets, in production, will not receive perfectly extracted spectra or focused images but the product of the imperfect prediction of the previous models in the pipeline schema. Deep GRU could receive (as input) spectra which contain only partially the source emission, and the ResNets not perfectly focused images. To account for this, we first trained all models in parallel in order for them to learn how to solve their respective problems while optimising the pipeline total training time (which benefits from the fact that the models' trainings are carried on at the same time), and then we also trained each model (with the same training strategies that we will outline for each model, and with the exclusion of Blobs Finder which is the first model in the pipeline) on the un-augmented training set predictions of the previous model. In this way, each model should be able to correct the mistakes (biases in the data) of the previous one.

Blobs Finder is trained on pairs of dirty input images (input dirty data cubes integrated along frequency) and target sky model images (target sky model cubes integrated along frequency). To achieve translational and rotational invariance in the network, at each iteration, both input and target images are randomly rotated by an angle θ with $\theta \in [0°, 360°]$, randomly flipped in the x and y planes with a probability of 1 and then cropped from $[360, 360]$ pixels to $[256, 356]$. The cropping operation brings no loss of data, given that all the values in images outside the primary beam of the simulated ALMA observations ($r \sim 138$ pix) are set to nan values, and those at the edges of the beam have unreliable SNR. After cropping, both integrated dirty cubes and target sky model images are normalized to the $[0, 1]$ range before being fed to Blobs Finder.

The model is trained with an Adam Optimiser [81], which is the optimization algorithm used to update the weights of the model on the basis of the loss function. The amount of change imparted to the model's weights to minimize the loss is moderated through the *learning rate*. Adam utilises a different learning rate for each parameter and updates them on the basis of the first and second moments of the gradients. Another problem in DL model's training, is the possibility of overfitting the data batches in the first training iterations, given the model's inherent initial instability due to the random initialisation of its weights. To prevent that, we adopt a *warm up* strategy for the learning rate [52] in which we start with a learning rate of 0, and we uniformly increase it to 1×10^{-4} in the first 10 iterations. The model is then trained for a maximum of 300 epochs, but we also employ an early stopping criterion based on the validation loss. If no improvement of validation loss with respect to the moving average of the last 10 validation losses is registered for 10 consecutive steps, then training is halted. As outlined in Sec. 4.2.1, Blobs Finder is trained with the weighted combination of the l_1 loss and the $DSSIM$ loss.

$$l(x, y, t) = a(t) \times l_1(x, y) + b(t) \times DSSIM(x, y) \tag{4.15}$$

where x and y are the prediction and target variables and t is the epoch counter. The weighting is performed with two variables a and b, which depend on the epoch counter t. The training begins with $a(0) = 1$ and $b(0) = 0$ in order to first allow the model to learn a *median* representation of the data (which should contain information about the PSF and noise patterns, assumed roughly constant in the data) through the minimization of the l_1 loss. At each epoch, a is decreased by δ and b is increased by the amount in order to slowly transition to the $DSSIM$ loss. In the last epochs, only the $DSSIM$ loss is effectively used in order to learn the nuances in the data, such as source positions or morphological properties (the shape and sizes of the galaxies).

Deep GRU is trained on pairs of dirty spectra (extracted from the dirty cubes) and clean spectra (extracted from the sky model cubes) through the bounding boxes obtained from Blobs Finder Prediction (or the known parameters in case of parallel training) The input dirty spectra are standardized while the target model spectra are normalized to the $[0, 1]$ range. To train the model, we again use the Adam optimization algorithm, a warm-up strategy for the learning rate with a working learning rate of 1×10^{-4}

and a weight decay of 1×10^{-5} to prevent the model from overfitting the training set. The model is trained for 300 epochs on spectra extracted through the known parameters and for 50 iterations on spectra extracted from Blobs Finder's predictions. We also employ an early-stopping criterium on the basis of the validation loss. The Deep GRU predictions are used in combination with Blobs Finder's predictions to extract the *spectrally focused* galaxy images, i.e., 64×64 pixels images with the source roughly in the centre and normalized to the $[0, 1]$ range (see Sec. 4.2.4 for more details). As targets for training, we use the true source parameters used to produce our simulations. The three ResNets for morphological parameters estimation are trained simultaneously for 300 iterations with the same stopping criterion set for Blobs Finder's training on perfectly spectrally focused images (obtained through the know source parameters) and for 50 iterations on the spectrally focused sources obtained through Blobs Finder and Deep GRU's predictions. Finally, the outputs of the ResNets (i.e., the sources' morphological parameters) are used to construct 3D models of the galaxies, which are used to create the segmentation masks from which we measure the continuum and create the line emission cubes and then the line emission images (for the details, see Sec. 4.2.4 points *v* and *vi*). The last ResNet is trained with the line emission images as inputs and the fluxes as targets. We use the same training strategy outlined for the other ResNets.

4.3 Source Detection and Characterisation

In this section, we present the performances of the different steps of the pipeline over 1,000 cubes which belong to the Test set. Whereas possible we also compare the performances of our pipeline with those of other pipelines widely used in the community: BLOBCAT [57], SOFIA-2 [160] and DECORAS [127].

4.3.1 Source Detection

We start with Blobs Finder's performance in detecting sources within the 2D Test set integrated dirty cubes. Following Sec. 4.2.4 step 1, the output 2D Probabilistic Maps are binarized through a hard threshold of

0.15, and bounding boxes around islands of connected pixels are extracted. To check if a source has been detected by Blobs Finder, we measure the 2D Intersection over Union 2D Intersection over Union (IoU) between the true 2D bounding box and the predicted one, while for Deep GRU the 1D IoU is measured between the true emission ranges and the detected ones. In both cases, a threshold of 0.6 is used. To ensure that the central part of the source emission of a True Positive True Positive (TP) is always detected, we require the distance between the centres of the true and predicted bounding boxes to be smaller than 3 pixels. The combination of these two thresholds should ensure that at least 60% of the 3D emission range of a source must be captured in order for its prediction to be deemed a TP. Also, given that the line emission image is created through the dilated segmentation map, the 0.6 IoU threshold guarantees that 90% of the true emission range is captured within it.

Getting a good estimate of the centres is important given that, except for blended sources that potentially will get a new centre later on in the pipeline based on the SNR reasoning outlined in Sec. 4.2.4 step 3, the ResNets will receive as input focused images cropped around the bounding boxes centres predicted by Blobs Finder. Hence, the closer the centres are to the true centres, the more sources will be centred in the focused images. As an example, by looking at the two blended sources in Fig. 4.14, one can see that the predicted bounding box (in red) encompasses most of the emission of the extended faint source and the totality of the emission of the compact source, so there is the potential of detecting both sources in the spectrum. When applied to the test set, Blobs Finder predicts 4056 (89%) sources (TP) against the true $4,556$ sources (using both the distance and IoU criteria). $4,205$ (92.3%) sources pass the 2D IoU criterion only, meaning that an additional 149 sources are detected by Blobs Finder but are spatially blended with another source. Blobs Finder also misses 354 sources False Negative (FN) and detects 4 False Positives False Positive (FP). The 4056 bounding boxes are used to extract a corresponding number of dirty spectra from the dirty cubes. The Deep GRU detects $4,202$ emission peaks out of the $4,205$ present in the extracted spectra but also produces 62 false positives. To detect and remove false detections and confirm true ones, sources are ”spectrally focused” within the predicted frequency emission ranges Δ_z, and SNR checks are made (see Sec. 4.2.4

(iii) Source Spectral Focusing). The full logic of the FP removal process is shown in Fig. 4.17. Regarding the 4 FPs detected by Blobs Finder, when the spectra extracted from the respective bounding boxes are fed to DeepGRU, it detects no emission peak within 3 of the 4, and thus the first 3 FPs are discarded as false detections. The last FP is eliminated through condition 2 of the schema shown in Fig. 4.17. Regarding the 62 FP peaks detected by DeepGRU, 29 were artificial peaks predicted near the spectral borders (the boundaries of the spectra may be recognised as peaks given the presence of a discontinuity in the signal) and, after being identified as primary sources (given the higher amplitude with respect to the real peaks), they failed the global SNR test when focused in their emission range (condition 1 in Fig. 4.17) and thus were discarded as false detections. Of the remaining 33 false detections made by DeepGRU, 31 are eliminated through condition 2 Fig. 4.17. These false peaks are all superimposed with their respective primary peaks in the spectra and the pixels with the highest SNR in the primary and secondary peak's spectrally focused images are less than 3 pixels apart and thus were considered either as noise spike or as part of the primary source. The remaining 2 false peaks produced spectrally focused images with a measured global SNR less than 1 and thus are flagged as false detections. Fig. 4.21 and 4.22 show some examples of Blobs Finder predictions on the test set. For each block, the first row shows the input integrated dirty cubes, the middle row the target sky models, and the bottom row, Blobs Finder predictions.

To compare with BLOBCAT ([57]) and SOFIA-2 ([160]), we run both algorithms on the 1000 dirty images in the test set. Blobcat requires two parameters: a detection (T_d) and cut (T_f) SNR threshold to decide which peaks in the image are good candidates for blobs and where to cut the blobs boundaries around them (in other words: pixels with an SNR higher than T_f are selected to form islands and island boundaries are defined by T_d). To make the fairest comparison possible, we measured BLOBCAT performances with different choices of T_d and T_f through a grid-search strategy ($T_d \in [2, 15]$, $T_f \in [1, 10]$) and, in this work, we report the best-obtained results ($T_d = 8\sigma$, $T_f = 4\sigma$).

The same criterion used for Blobs Finder was used to measure BLOBCAT performances. BLOBCAT successfully detects $2,779$ (61%) sources, produces $2,429$ false detection and misses 1777 sources. The majority of sources missed by BLOBCAT are spatially blended with brighter sources, or present a $SNR <= 5.0$, or are located at the edges of the images. The Sofia-2 smooth and clip algorithm (S + C) algorithm works by iteratively smoothing the data cube on multiple spatial and spectral scales to extract statistically significant emissions above a user-specified detection threshold on each scale.

Guided by the spatial and frequency sizes of the simulated sources, we employed spatial and frequency kernel sizes of $[3, 5, 7, 9, 11]$, and a source finding threshold of 0.5. The algorithm proceeds by linking together meaningful detections with a friend-of-friend algorithm; we employed a grid-searching strategy to find the best possible spatial linking radii in the image and frequency dimensions of the cube within the interval $[1, 5]$ pixels. Finally, the algorithm removes false positives on the basis of a reliability score based on the source SNR. We employ a SNR threshold of 2 and we let the algorithm automatically select the other thresholds. We limit the detection area of Sofia-2 by setting the masking to a 256 pixel size square centred in the image. This is performed in order to cut the low SNR outskirt of the dirty images. Sofia-2 detects 1010 (22%) sources, produces 4011 false detections and misses 3546 sources. Most false positives are located at the spatial edges of the image, while it misses most blended sources by merging their emissions with other sources which leads to all involved sources failing the 2D IoU or distance-based thresholds.

The DECORAS ([127]) pipeline is constituted by two Deep Convolutional Autoencoders, the first one works exactly like our Blobs Finder, the second one takes $[128, 128]$ pixel cropped images around the first Autoencoder's predictions and predicts the source structure which is then fitted with a 2D Gaussian Function to find the source morphology. Our simulated cubes contain multiple sources while DECORAS assumes that there is a single source in each cube and thus the characterisation part of the pipeline cannot be used to detect sources. For such reason, we compare our results with those by DECORAS only on the detection part of the problem. First of all, following the guidelines outlined in their article, we re-implemented their Convolutional Autoencoder. The main architectural differences between our Convolutional Autoencoder and theirs are: their latent space contains 256 atoms, while our 1024; they perform spatial upsampling by only using chained Transposed Convolutions, while we use a combination of Bilinear Interpolation, Transposed Convolutions and Convolutions; and their Encoder and Decoder contain half our convolutional layers. In their paper, they train their Autoencoder with two loss functions: the Binary Cross Entropy Binary Cross Entropy (BCE) loss and the Mean Squared Logarithm loss (MSLE), and deem as correct predictions only those sources which are predicted by both models and whose distance (measured from the source centres found through the *blob_ dog scikit-image* algorithm) is less than 3 pixels. We trained and tested their model with both loss functions using the same criteria outlined in Sec. 4.2.5 for Blobs Finder. After seeing that no meaningful result could be obtained with the BCE, we only use the model trained with the MSLE to perform the source detection on the 1000 dirty integrated images in the test set. Given that the *blob_ dog* algorithm only finds coordinates of the sources and does not output their emission boundaries, needed to create bounding boxes with which measure the IoUs, we make two performance measurements. The first one relaxes the source detection criterion and only uses the distance-based threshold, while the second one uses both the distance and IoU criteria. To get bounding boxes from *blob_ dog* outputs we use, for each blob detected by the algorithm, the best matching kernel standard deviations to create a radius of emission from which derive bounding boxes with the following equation:

$$r = \sqrt{(3 * \sigma_b)^2} \tag{4.16}$$

where r is the obtained radius, and σ_b is the standard deviation of the best fitting kernel. DECORAS detects $4,100$ (89.9%) sources, produces 759 false detections and misses 456 (10.1%) sources using only the distance based threshold. The number of detected sources drops to $3,560$ (78.2%) if also the IoU-based threshold is used, while the number of false positives stays the same, and the number of missed sources rises to 996 (21.9%). In Table 4.4 we summarise the source detection performances for all methods. The two DL-based pipelines achieve improved performances over their traditional counterparts. By taking a look at DECORAS performances, especially if only the distance-based criterium is used, one can see that they are similar to those of our Blobs Finder with respect to the number of TP while the number of FP is much higher. This behaviour could be connected to the use of subsequent Transposed Convolutions to perform spatial upsampling in the Decoder part of the network [161] which may

lead to artefacts being mistaken as blobs by *blob_ dog*. Regarding the FP detection and removal, we show both the performances without the use of FP detection and removal pipeline (BF + DeepGRU) and those with it (Pipeline).
Blobs Finder Blobs Finder (BF) and Deep GRU have a False Positive Rate False Positives Rate (FPR) respectively of $\sim 10^{-3}$ (Blobs Finder) and $\sim 10^{-2}$ (Deep GRU) and a combined number of 66 FPs which is already within acceptable limits. Nevertheless, the FP removal pipeline is able to correctly identify and remove all of them. While the false detections made by Blobs Finder were trivial to identify and remove and may continue to be so in the case of real data (no relevant peak was detected in the corresponding spectra by Deep GRU), the logic we used in removing the false detections made by Deep GRU that were due to a single true emission peak being detected as two superimposed peaks, will require further investigation. In fact, if the velocity dispersion and inclination with respect to the line of sight are taken into account (both factors are not yet present in our current simulations), the hypothesis that the spatial position of the brightest pixel of a source should remain constant within its spectral emission range no longer holds. Thus, the criterium which we employ to separate a true secondary peak from a false detection cannot be used anymore and needs to be modified. Blobs Finders' and DECORAS' low numbers of FPs with respect to SOFIA-2 and BLOBCAT are explainable due to the DL models' capabilities of approximating the dirty beam. While Blobs Finder improved performances over DECORAS is due to architectural choices and training strategies, we believe that if dirty beam variations are increased by simulating multiple antenna configurations and observing conditions (integration time or azimuth), the number of FPs detected by both models will probably increase due to the increased complexity required to approximate a more realistic variating PSF. Fig. 4.19 shows the histograms of flux densities (left) and the *blendness scores* (right) of the detections made by our pipeline, Blobs Finder, and by DECORAS. The blendness score is defined as the maximum IoU between a source true bounding box and all other source bounding boxes within an image and can be used as a proxy to quantify how much different sources are spatially

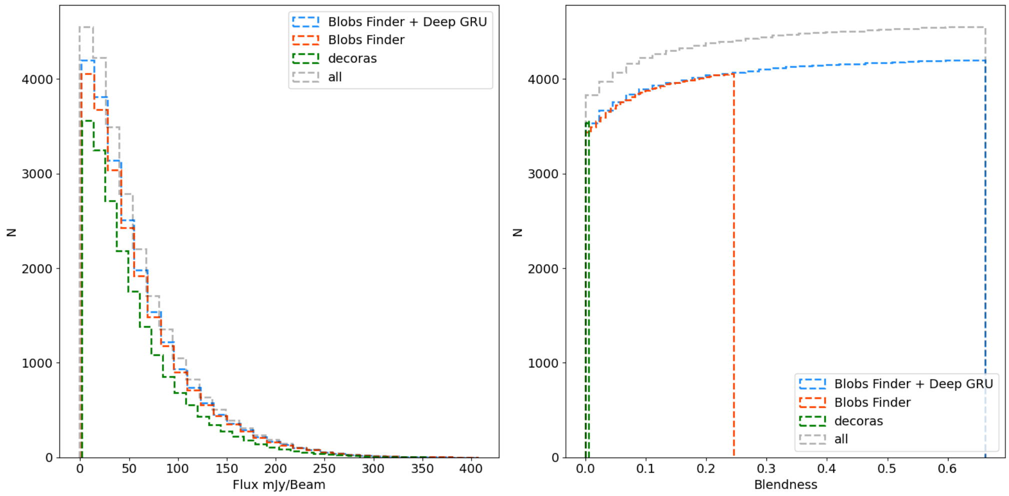

Figure 4.19. Left: histograms of the detected sources flux densities. Right: cumulative histogram of the detected sources *blendness score* (see the text). In both histograms, we compare our detection pipeline (Blobs Finder + Deep GRU), our implementation of Blobs Finder, DECORAS implementation of Blobs Finder and, we report the histograms for all the test set distribution.

blended. Mathematically we define it as:

$$b_i = max(IoU(box_i, box_j)) \; for \; j \in [1, N] \tag{4.17}$$

where N is the total number of sources within the image. If the blendness is 0, it means that the source is spatially isolated in the cube, while a blendness of 1 means that the source is spatially superimposed to at least another source in the cube. We also report all the test set values for comparison. Our implementation of Blobs Finder is equivalent to DECORAS's implementation regarding the minimum detected flux density of 1.31 mJy/beam but it seems to be more effective in deblending sources. Blobs Finder detects sources up to a blendness value of 0.232, while DECORAS reaches at most 0.021. It has to be said that also this difference in performance could be connected to the different ways the two pipelines extract bounding boxes from the model images produced by the Autoencoders and not to the quality of the images themselves. To eliminate the effect of the extraction algorithm from our comparison, we compare the

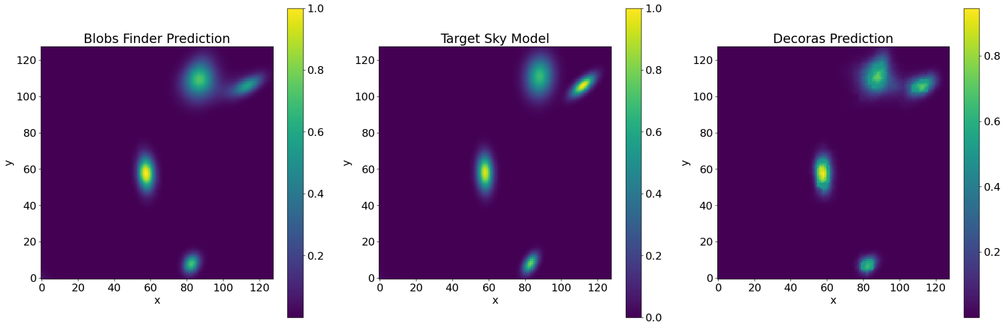

Figure 4.20. Left: Blobs Finder predicted 2D probabilistic map; Center: true sky model image; Right: DECORAS implementation of Blobs Finder predicted 2D probabilistic map. Predictions and target images have been cropped to 128 by 128 around sources, to better showcase the reconstruction quality.

measured mean structural similarity index (mSSIM) between the true sky model test set images, Blobs Finder predictions and DECORAS predictions. Blobs Finder achieved a mSSIM on the test set of 0.003, while our implementation of DECORAS achieved a mSSIM of 0.008. Fig. 4.20 shows the direct visual comparison between the true target sky model image (centre), Blobs Finder (left) and DECORAS (right).

Deep GRU (and Source Spectral Focusing) does not improve on the minimum detected flux, but deblends sources and in fact pushes the maximum detected source blendness to 0.66 which is the maximum blendness simulated in the data.

4.3.2 Source characterisation

The 4, 202 focused images are fed to the 3 ResNets (steps 10 and 11 of the pipeline) each one fine-tuned to predict one of the three morphological parameters: $FWHM_x$, $FWHM_y$, and pa. The source positions x and y are computed as the pixel-weighted centres of the sources bounding boxes, while the peaks frequency positions z and extensions Δ_z are computed by fitting 1D Gaussians to the clean peaks found by Deep GRU. The sources parameters are used to create the Line Emission Image (see 4.2.4 Flux Estimation)

Fig 4.23 shows the scatter plots of the true parameters versus the pre-

Algorithm	**TP /**	**FP**	**FN**
BF + DeepGRU	4202 (92.3%)	63	354 (7.7%)
Pipeline	4202 (92.3%)	0	354 (7.7%)
blobcat	2779 (61%)	2429	1777 (39%)
Sofia-2	1010 (22%)	4011	3546 (78%)
decoras	3560 (78.2%)	759	996 (21.9%)
Algorithm	**Precision**	**Recall**	**Mean IoU**
BF + DeepGRU	0.98	0.923	0.74
Pipeline	1.0	0.923	0.74
blobcat	0.53	0.609	0.61
Sofia-2	0.20	0.22	0.63
decoras	0.82	0.78	0.60

Table 4.4. Comparison between the sequential application of Blobs Finder and DeepGRU (BF + DeepGRU), the sequential pipeline completed with the Spectral Focusing for FPs removal and deblending (Pipeline), *blobscat*, Sofia-2 and decoras. Columns show true positives (TP), false positives (FP), false negatives (FN), precision, recall and mean intersection over union (Mean IoU) between true bounding boxes and predicted ones. TP and FN are also expressed as fractions over the total number of sources.

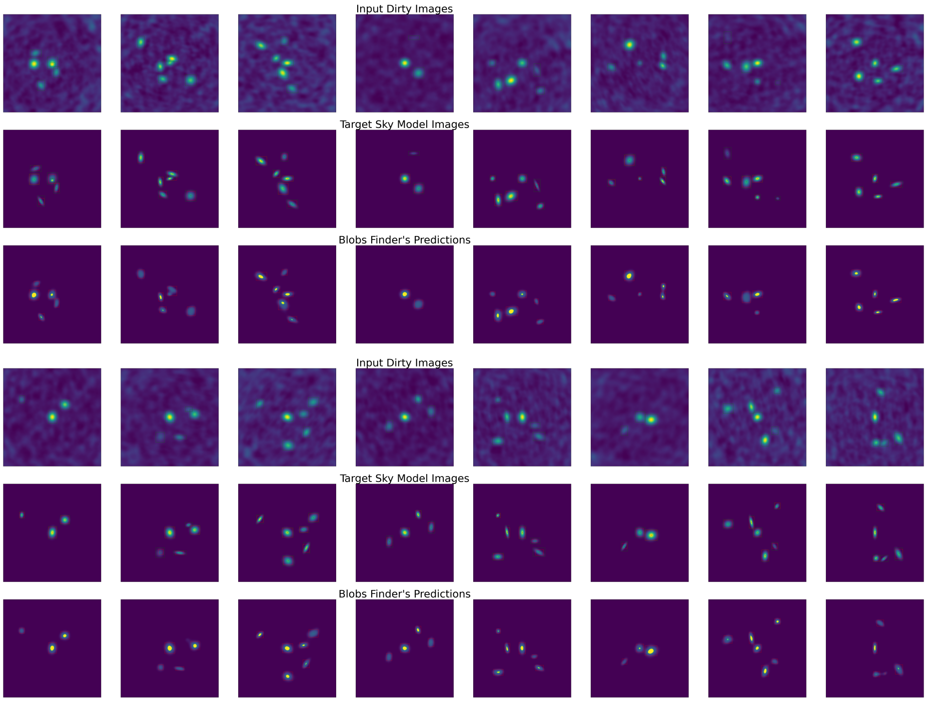

Figure 4.21. Examples of Blobs Finder predictions on the test Set. The first row shows input integrated dirty cubes, the middle row the target sky models, and the bottom row, Blobs Finder predicted 2D Source Probability maps. In green are outlined (in the dirty and sky models images) the true bounding boxes, while in red are the predicted bounding boxes extracted by thresholding the probability maps.

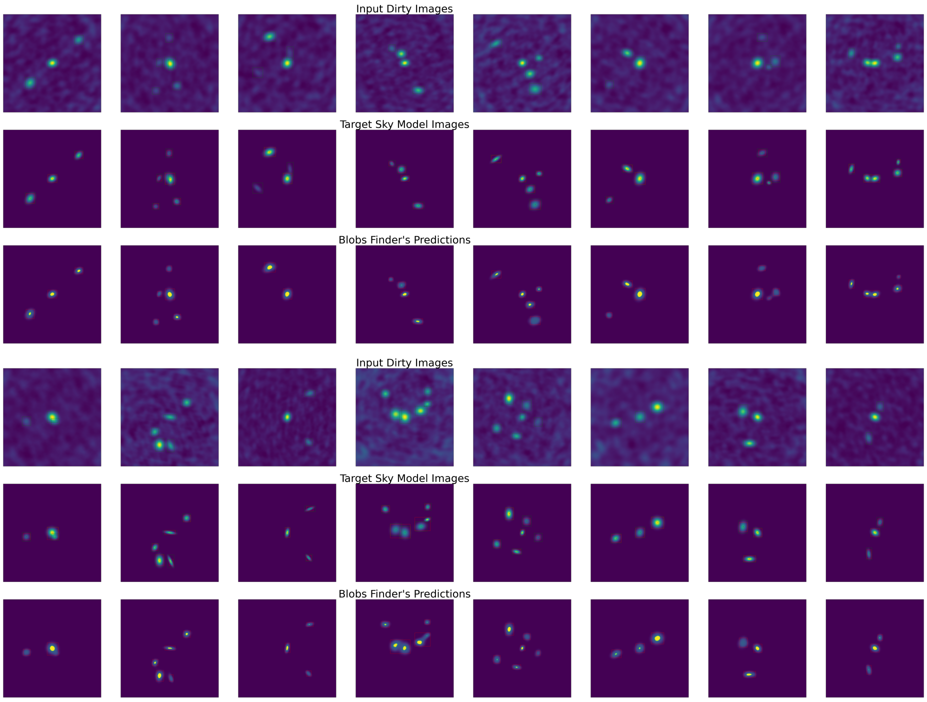

Figure 4.22. Examples of Blobs Finder predictions on the test Set. The first row shows input integrated dirty cubes, the middle row the target sky models, and the bottom row, Blobs Finder predicted 2D Source Probability maps. In green are outlined (in the dirty and sky models images) the true bounding boxes, while in red the predicted bounding boxes extracted by thresholding the probability maps.

Parameter Residual	**mean**	**std**
x (pixels)	-0.004	0.73
y (pixels)	-0.005	0.67
FWHMx (pixels)	-0.04	0.46
FWHMy(pixels)	-0.12	0.45
z(slices)	0.0	0.003
Δ_z (slices)	0.0	0.001
pa (degrees)	-0.65	20.28
flux (mJy/beam)	-9.56	20.08

Table 4.5. This table shows the mean and standard deviation of all the residual distributions between the true target parameters and the predictions made by our pipeline. The x and y positions are computed from Blobs Finder predicted blobs, z positions and extensions Δ_z are computed from Deep GRU predictions, and the remaining parameters are predicted from the four ResNets. Alongside each parameter, we also indicate their unit of measurement.

dicted ones and the corresponding residuals histograms. The residuals are produced, for each parameter through the following equation:

$$res_i = (t_i - p_i) \tag{4.18}$$

where res_i is the residual for the i-th parameter, t_i is the true parameter value, while p_i is the predicted one. Tab. 4.5 summarises the performances showing, for each parameter, the mean and standard deviation of the residual distribution.

The source positions (spatial, frequency) and extensions ($FWHM_x$, $FWHM_y$ and Δ_z) are detected with sub-pixel accuracies. Concerning the performances on the flux densities and projection angle regression, the relative error is defined as follows:

$$rel_i = (t_i - p_i)/t_i \tag{4.19}$$

where rel_i, t_i and p_i are, respectively, the relative error, the true parameter value and the predicted one for the i-th parameter. The achieved relative errors for the flux densities are 0.07 with a standard deviation of 0.36. In particular, 68% of sources have a relative error on the flux density which

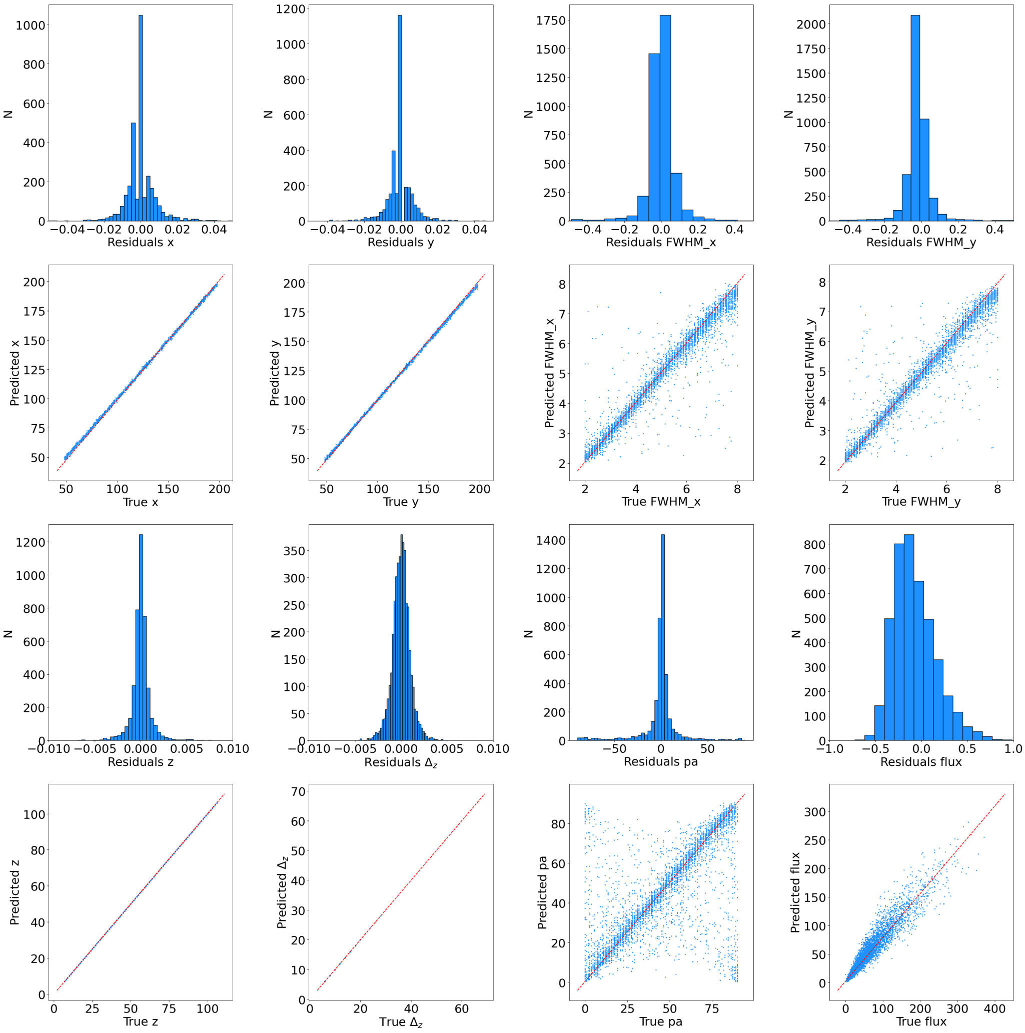

Figure 4.23. Scatter plots of the true parameter values against the models' predictions and the corresponding residuals histograms. The red dotted lines in each scatter plot represent the bisector of the quadrant, i.e. if all instances were perfectly predicted, they would all lie on the red line.

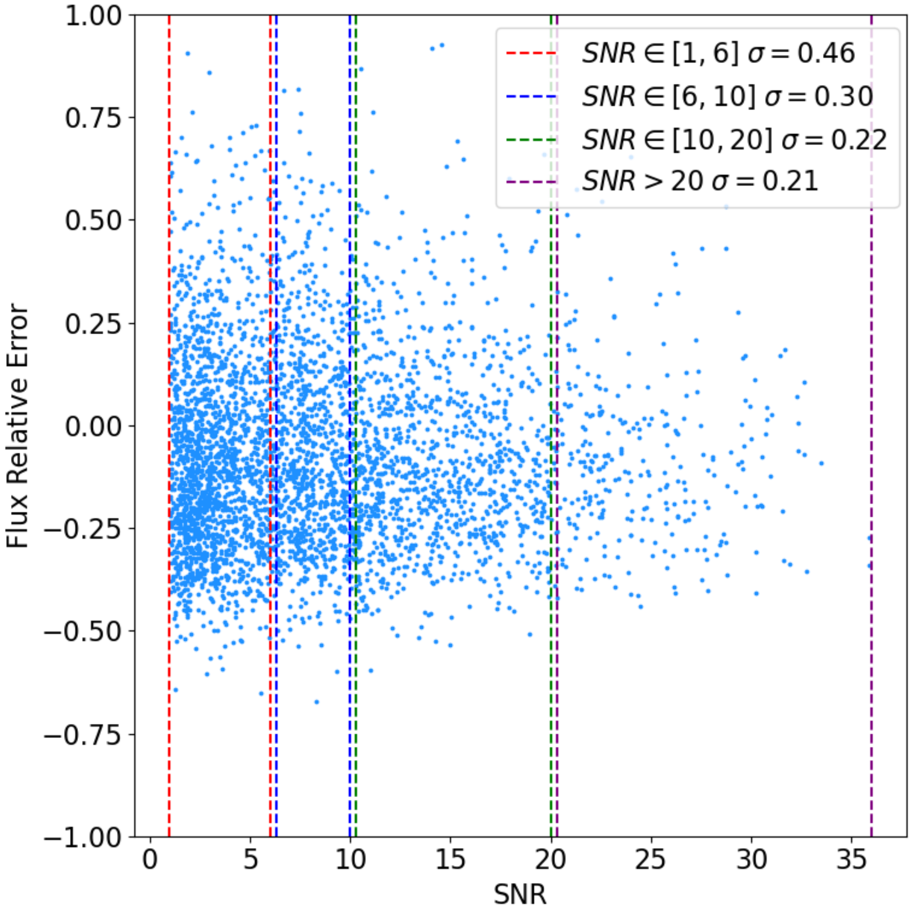

Figure 4.24. Scatter plot of the sources SNR against their flux densities relative errors. Vertical bars divide the plot in sections of SNR. The legend shows, for each SNR interval, the standard deviation of the relative errors.

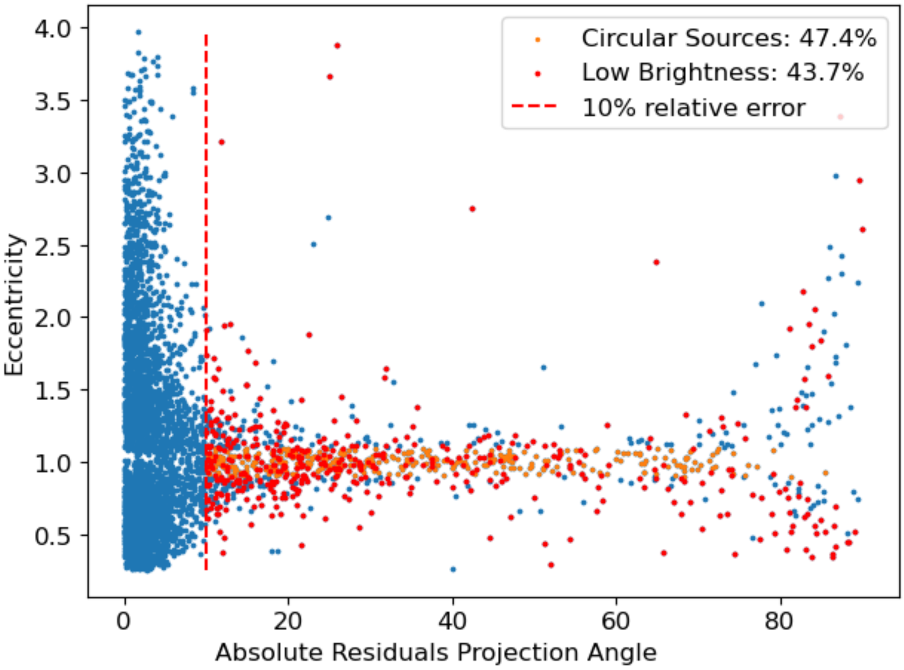

Figure 4.25. Scatter plot of the sources' absolute projection angle residual errors against their eccentricity, defined as the ratio of their FWHMs. The vertical bar delimits the 10% mark for the residual error, while the sources highlighted in orange are circular ($e \simeq 1$) and the ones in red have surface brightness lower than 30 mJy / beam. These sources account for respectively 47.4 and 43.7 of all sources with a relative error higher than 10%.

is less than 1% away from the true value, 80% less than 10% and 87% less than 20%. Regarding the projection angles, 53% of sources have a relative error which is less than 1% away from the true value, 73% less than 10% and 81% less than 20%. The scatter plot of true pa versus predicted ones (Fig. 4.23 last row, third panel from the left) shows two regions where the nets fail to make accurate predictions, namely, around the values of 0 and 90 degrees. This could be expected given the well-known degeneracy encountered in measuring projection angles for almost circular sources. Fig. 4.25 shows the scatter plot of the absolute values of the residuals on the projection angle estimation versus the source eccentricity (defined as $e = FWHM_x/FWHM_y$). 47.4% of all sources with a residual error higher than 10% are almost circular ($e \sim 1$) and 43.7% have a surface brightness less than 30 mJy / beam. Regarding the surface brightness estimation, the scatter plot of the true flux density versus the predicted one shows that at higher fluxes, the predictive error increases. This behaviour can be explained due to the fact that the brightest sources represent only 16% of the data set and that the ResNet is trained, by minimizing the l_1 loss, to predict the median of brightness distribution which is 51.2 in our Train set. The combination of the choice of the loss function and scarcity of bright sources in the data encourages the model to focus more on less bright sources while treating bright sources as outliers. Fig. 4.24 shows the scatter plot of the sources SNR against relative errors on the flux estimations. The standard deviation of the flux relative errors distribution halves for sources with SNR higher than 10 with respect to fainter sources.

4.4 Sky Model Reconstruction, comparison with tCLEAN

In order to compare the image reconstruction capabilities of Blobs Finder with respect to tCLEAN, we re-run our simulation code on the 1000 sky-model cubes setting the number of tCLEAN iterations to 200. This way, for each sky model cube, we obtain the corresponding sky-model reconstruction cube generated by tCLEAN. The obtained 1000 tCLEAN cleaned cubes are integrated along frequency to get the integrated images, and normalized to the $[0, 1]$ range. To measure and compare the true sky model reconstruction performance, we utilise two metrics: the mean struc-

Algorithm	mSSIM	MRS
BF	0.003	−0.0006
tCLEAN	0.45	0.05

Table 4.6. The table shows the mean structural similarity index (mSSIM) and the mean residual score (MRS) between the true sky model images, Blobs Finder (BF) and tCLEAN reconstruction over the Test set.

tural similarity index (mSSIM) and the mean residual score Mean Residual Score (MRS). To compute the mean residual score, first the residual image is computed as follows:

$$res_{ij} = (t_{ij} - p_{ij}) \ \forall i,j \in [256, 256] \tag{4.20}$$

where t_{ij} is the pixel value at index i, j of the sky-model image and p_{ij} is the pixel value of Blobs Finder or tCLEAN reconstructions. The mean residual is then computed as the mean of the residual image and the mean score is the average mean residual over the entire Test set.

Table 4.6 shows the mSSIM and MRS computed on the 1000 Test set images. An improvement of two orders of magnitudes on both metrics is reached when employing Blobs Finder. Figures 4.26, 4.27, 4.28, 4.29, 4.30, 4.31 show several examples of target Sky Model images reconstructions. In particular, Fig. 4.27, 4.28 show the residual images (see Eq. 4.20) relative to the example showcased in the first row of Fig. 4.26, and Fig. 4.30, 4.31 for the first row of Fig. 4.29. Blobs Finder shows better reconstruction capabilities characterised by residuals centred around zero and with a lower scatter with respect to that of tCLEAN residuals. Regarding computational times, Blobs Finder made its predictions on the entire Test set in 23 seconds employing a single NVIDIA Tesla K20, while tCLEAN took 4.3 minutes per cube utilising 8 Intel Xeon E5-2680 CPUs. Given the 400 CPUs at our disposal, we run it on 50 cubes at a time in parallel obtaining a total computational time of 1.5 hours. Employing BlobsFinder for the reconstruction task on the entire Test set results in a speed-up factor of 200 on our system with respect to tCLEAN. The speed-up is even more if tCLEAN is used in a serial fashion.

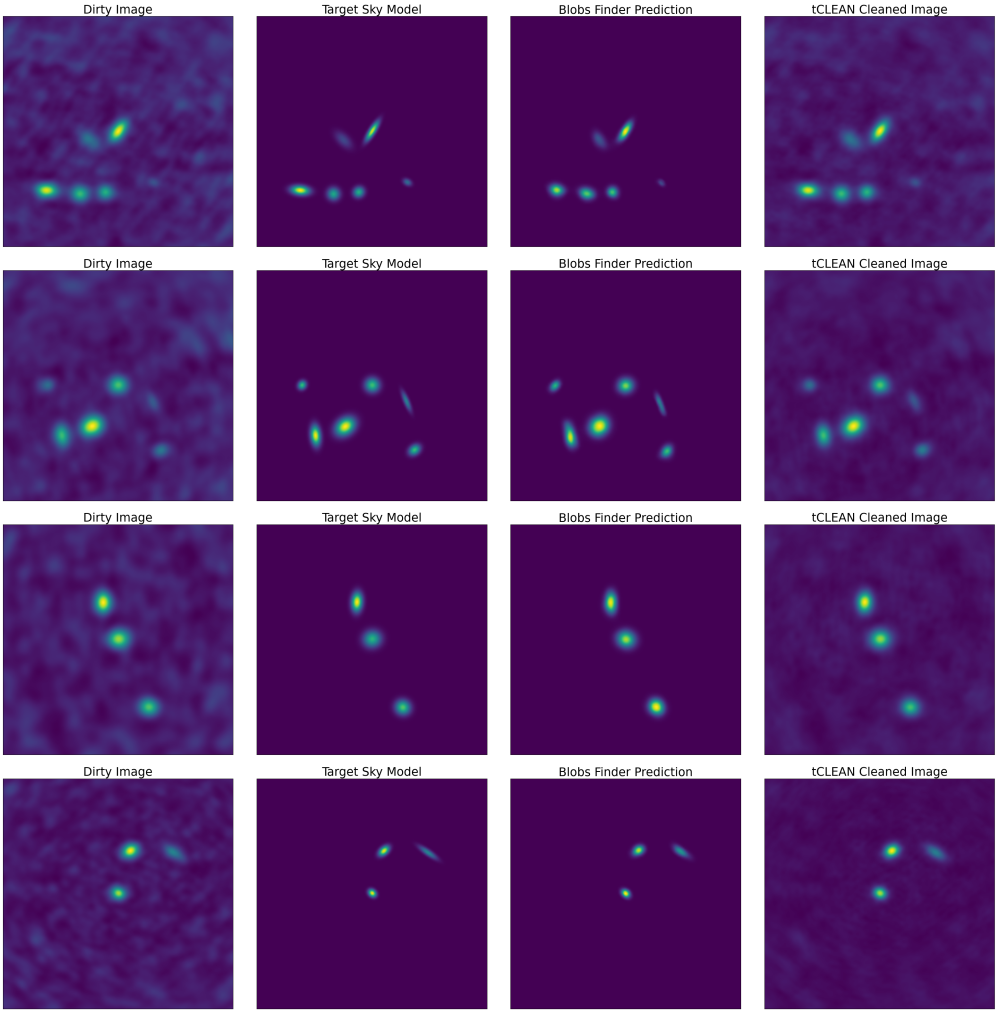

Figure 4.26. Examples of Dirty Images (first column), target Sky Models (second), Blobs Finder's reconstructions (third) and tCLEAN reconstructions with $n_{iter} = 200$ (forth). Fig. 4.27 and Fig. 4.28 show the residual images relative to the first row's images of this figure.

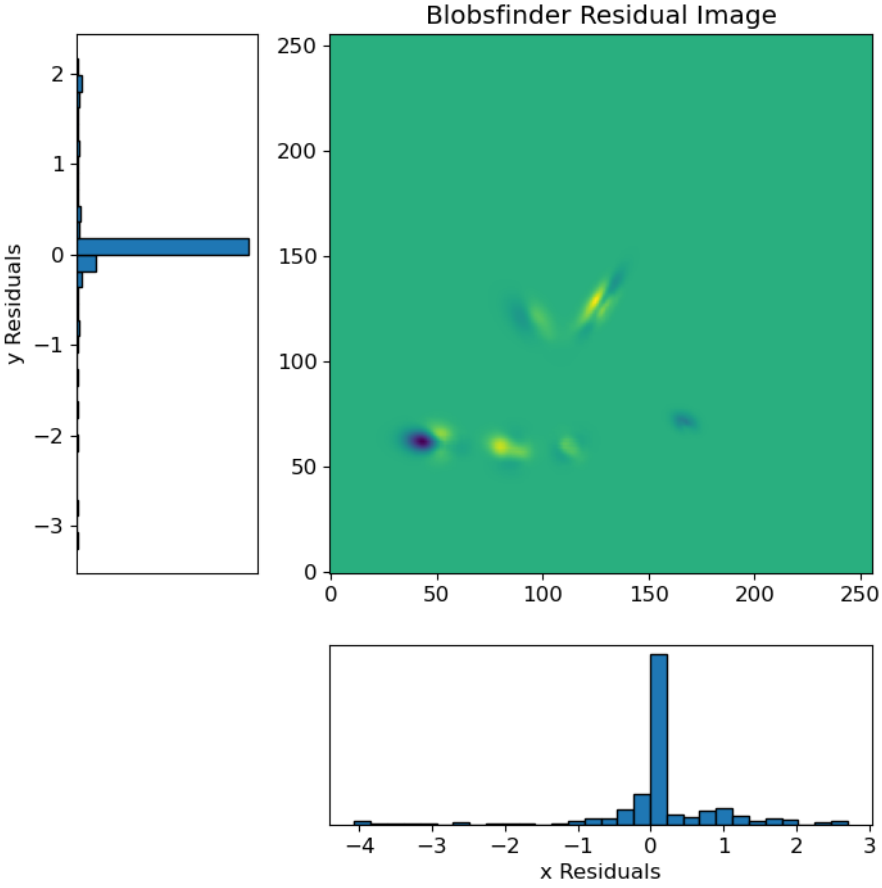

Figure 4.27. Residual Image created through Eq. 4.20 between the target Sky Model and Blobs Finder's predicted reconstruction. The two scatter plots show the marginal distributions obtained by summing pixel values along each row and column of the image, respectively. The Dirty image, The Sky Model image and Blobs Finder's prediction relative to the residual image shown in this figure are showcased in the first row of Fig. 4.26.

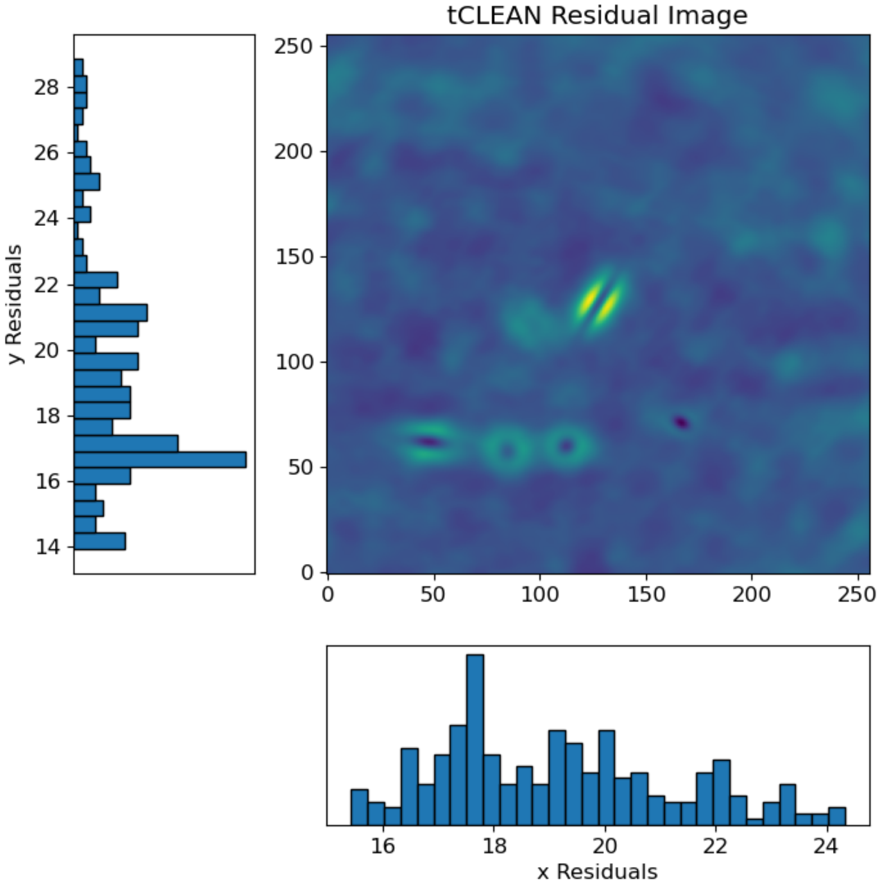

Figure 4.28. Residual Image created through Eq. 4.20 between the target Sky Model and tCLEAN reconstruction.The two scatter plots show the marginal distributions obtained by summing pixel values along each row and column of the image, respectively. The Dirty image, Sky Model image and tCLEAN reconstruction relative to the residual image shown in this figure are showcased in the first row of Fig. 4.26.

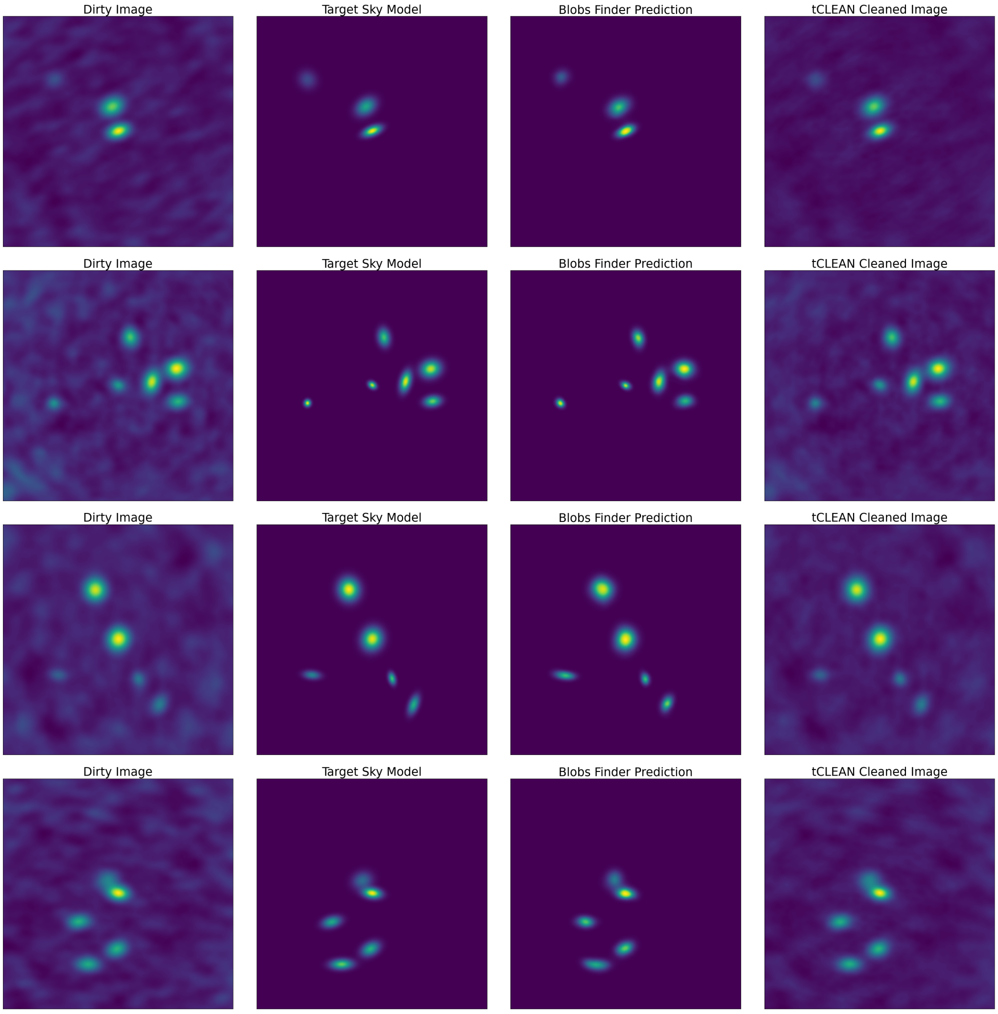

Figure 4.29. Examples of Dirty Images (first column), target Sky Models (second), Blobs Finder's reconstructions (third) and tCLEAN reconstructions with $n_{iter} = 200$ (fourth). Fig. 4.30 and Fig. 4.31 show the residual images relative to the first row's images of this figure.

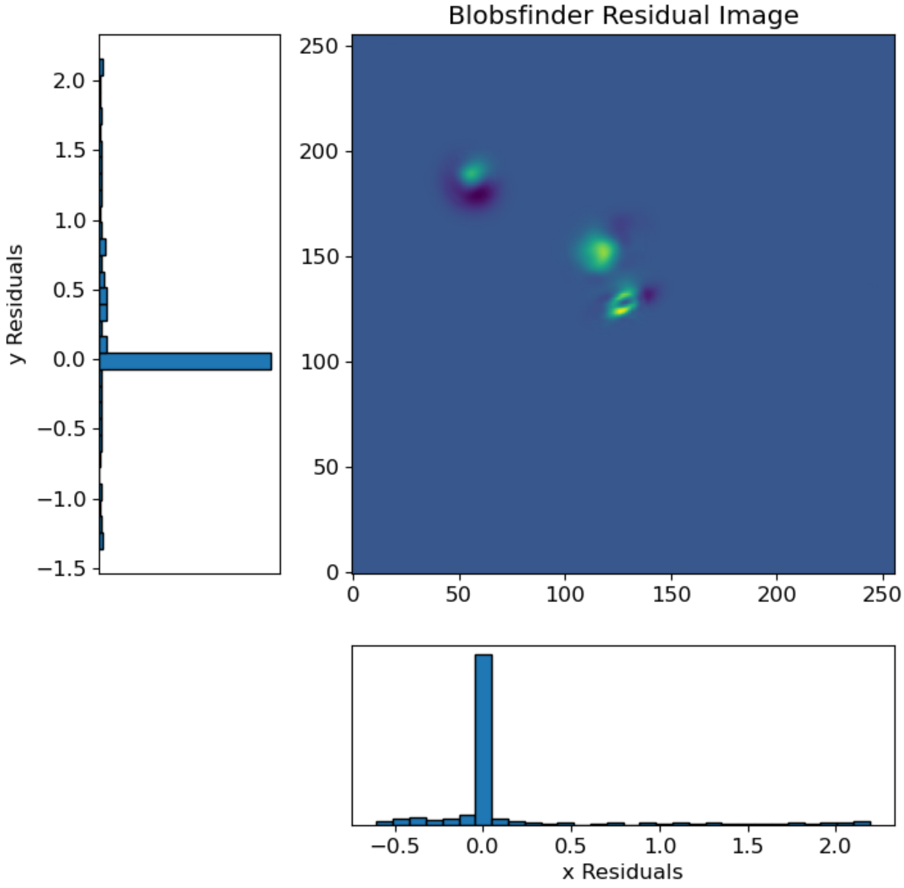

Figure 4.30. Residual Image created through Eq. 4.20 between the target Sky Model and Blobs Finder's predicted reconstruction. The two scatter plots show the marginal distributions obtained by summing pixel values along each row and column of the image, respectively. The Dirty image, The Sky Model image and Blobs Finder's prediction relative to the residual image shown in this figure are showcased in the first row of Fig. 4.29.

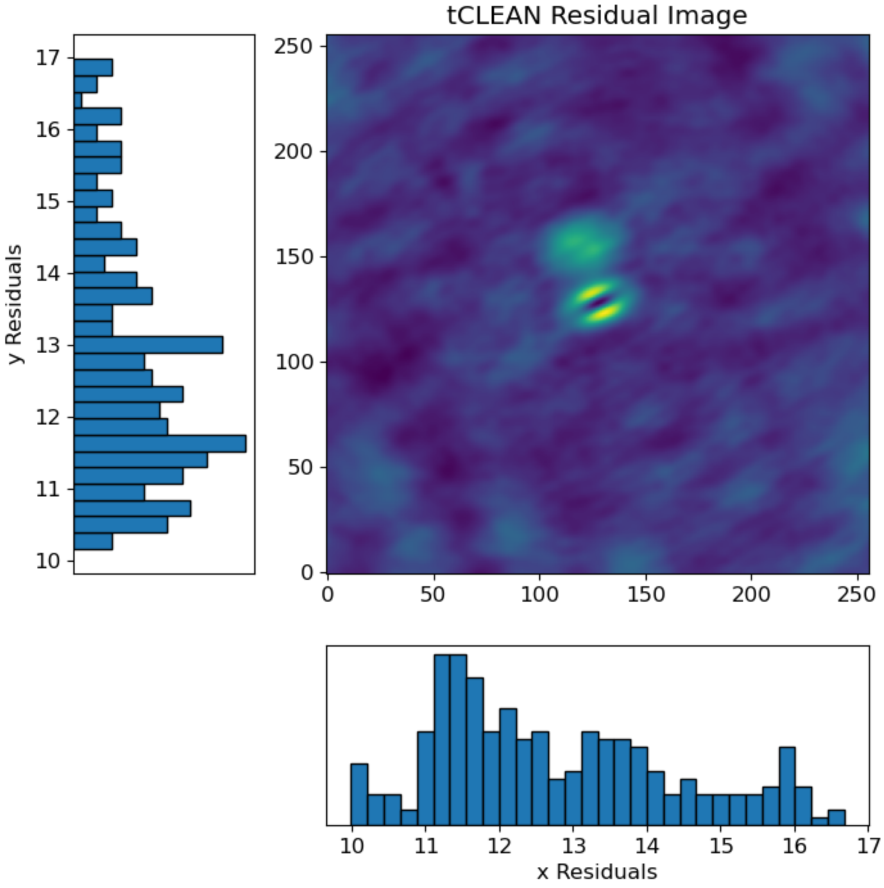

Figure 4.31. Residual Image created through Eq. 4.20 between the target Sky Model and tCLEAN reconstruction. The two scatter plots show the marginal distributions obtained by summing pixel values along each row and column of the image, respectively. The Dirty image, Sky Model image and tCLEAN reconstruction relative to the residual image shown in this figure are showcased in the first row of Fig. 4.29.

4.5 Discussion and Conclusions

The transition of many existing and planned radio interferometers to the Terabyte data regime requires the community to develop automatized source detection and characterisation pipelines capable to cope with data streams of ever-increasing size and complexity. In recent years several attempts have been made to use deep learning in order to speed up procedures and to make the detection pipeline less subjective. In this Chapter we present a novel deep-learning-based source detection and characterisation pipeline for radio interferometric datacubes. This pipeline works directly on dirty interferometric calibrated data cubes which have not undergone any prior de-convolution (i.e., on the results of the Fourier transform applied to the calibrated visibility data) and combines spatial and frequency information to detect and characterize sources within the cubes. The pipeline was fine-tuned on the characteristics of ALMA data (commonly processed with the CASA software). Nonetheless, given the similarities existing among radio interferometric data from different instruments, the pipeline can be exported (in presence of a large enough number of simulations for training) to other instruments (such as LOFAR, SKA, VLBI, VLTI). Given the obtained reconstruction capabilities, the proposed pipeline has the potential to support CASA with a new design for image reconstruction and/or to provide a speed-up procedure for convergence purposes. Regarding the obtained performances, while Blobs Finder seems to be able to outperform tCLEAN in both speed and performance, this result was only obtained on simulated cubes containing simple point-like sources. Further investigations will be required on cubes containing sources with extended emissions and complex morphologies and on real data. Summarising: the pipeline is composed of six deep learning models interconnected through logical operations: Blobs Finder detects sources within the frequency-integrated data cubes, Deep GRU exploits the frequency domain and detects emission peaks in the spectra extracted from sources detected by Blobs Finder, and the ResNets regress the source parameters from 'spectrally focused' images created by cropping spatially around the sources, and integrating within their emission range found by Deep GRU, and the line emission images created by masking the cube with the 3D emission models found by combining Blobs Finder and Deep

GRU predicted emission ranges (in the spatial and frequency planes of the cube, respectively). In order to test the performances of the pipeline, an ALMA observation simulation code was developed. The simulation code is made available through GitHub allows the community to generate thousands of ALMA data cubes in parallel. The source detection capabilities of the pipeline can be summarised by the performances achieved on the test set: 92.3% of the simulated sources are detected with no false positives. While the achieved performances are promising for the prospect of applying the pipeline to real data, the low FPRs of the DL models and the subsequent removal of the remaining FPs through SNR and geometrical criteria (FP detection and source deblending step) are closely related to the simplified assumptions that were made to generate the mock data. Integration times and antenna configurations are kept constant resulting in a very low dirty beam variation across the data. Only single peak spectra are simulated without taking into consideration the effects of inclination with the observer's line of sight or velocity dispersion within the galaxy, resulting in simplified spectral profiles. Quality assessment is performed by comparing our results with three other methods, blobcat, Sofia-2 and decoras. We notice a substantial improvement in both precision and recall with respect to the first two methods and a smaller improvement with respect to decoras. Regarding the source characterisation performances, source positions are found with subpixel errors in spatial and frequency domains, while projection angles and flux densities estimations show a relative error within the standard amplitude calibration error of interferometric data ($\simeq$ 10% [36]) for respectively 73% and 80% of all sources. Regarding the projection angle regression, the ResNet is not able to correctly predict this parameter for circular sources only, being trivial. Given the insights obtained from the analysis of the ResNet performances, a way to avoid the futile regression of the projection angle on circular sources could be to first compute the source eccentricity (from the predicted $FWHM_x$ and $FWHM_y$) and then regress the projection angle only for non-circular sources. The flux relative error increases with source brightness and decreases with SNR. While explaining the latter trend is easy, the first appears to be counterintuitive. In fact, the brighter a source is, the easier it should be to measure its flux density. This trend comes from the fact that our simulations contain many more faint sources than bright ones (see Fig. 4.4) and from our choice of training the ResNet with an l_1 loss. Regarding the execution time, our pipeline can process a simulated data cube (67.3 MBs) in $\sim$ 10 ms. Although the DL pipeline has not yet been tailored for use within CASA, the technique is capable to speed up and improve over tCLEAN by leveraging correlations in all dimensions of the cube. In fact, we aim at improving our simulation code to include more complex galaxy morphologies, using physically-based models for the galaxy kinematics, and employ spectral catalogues to generate several spectral profiles for different classes of sources with the primary goal of improving the quality of our simulations, and the additional goal of having a publicly available and easy to use simulation code that the community may use to generate common data sets on which compare different architectures. Alongside we want to modify our pipeline to account for much more complex spectral profiles. The DL pipeline is currently tailored to detect single-peak emission lines, assuming that celestial sources have a single positively defined emission peak in their spectra. The above does not hold for absorption line detection or for sources with multi-peak spectral profiles. Both the peak detection (on Deep GRUs denoised spectra) and FP detection and removal (on spectrally focused images through SNR criteria) algorithms will thus need to be modified when dealing with real observations. I'm also planning to make an assessment on faint signal detection, especially in the presence of strong sidelobes in the cubes and asses the pipeline capabilities in the case of data containing several uv coverages and array configurations, which should result in a greater variation of the dirty beam within the data, thus posing a more complex image reconstruction problem. Furthermore, we also plan to make tests about incorporating the dirty beam within our pipeline in order to improve the image reconstruction capabilities and to test the FP removal pipeline against a DL classifier based on our ResNet architecture in case of the aforementioned more complex data. The DL pipeline is going to be further extended on continuum imaging and applied to real ALMA observations with the aim of finding new faint serendipitous galaxies in the neighbourhood of brighter companions. This task has been proven difficult for classical algorithms.

Data and Software Availability

The simulation code to generate the data used in this work is made publicly available through GitHub (ALMASim). Detailed instructions on how to set up a python environment with all the pip packages needed to run the code and generate the data are outlined in the repository. The simulation code has been tested only on Unix distributions (Ubuntu, CentOS, macOS) and is composed of two bash scripts and several python scripts which generate the sky models, run the CASA tasks to simulate ALMA observations and produce the cubes, control the noise properties, and record the source parameters. The bash scripts are written for the IBISCO architecture which uses the slurm workload manager to split computations among nodes, and thus they should be changed to reflect the architecture on which they are executed. On our system, $5,000$ cube pairs were generated on 400 Intel Xeon E5-2680 CPUs in around a day. The pipeline is also implemented in python and made publicly available through GitHub (DeepFocus). The Deep Learning models, the data loading, augmentation, training and testing routines are written through the pytorch library. While the simulation code is fully documented, at the time of writing of this paper, the documentation for the pipeline is still under construction. Basic instructions on how to run the training and testing of the pipeline are outlined on the GitHub page. Also in this case, the script parameters are tailored for the IBISCO architecture and should be changed accordingly. Blobs Finder' training lasted ~ 4 hours on a 2 NVIDIA Tesla K20 and it made the predictions on the test set in ~ 23 seconds including I/O operations. While Blobs Finder was trained on two GPUS, Deep GRU and the ResNets were trained on a single NVIDIA Tesla K20. Regarding the name of the pipeline, given its modularity, it will probably not hold for long and will be reformulated once more extensive tests are made over the several problems which are investigated in this paper: image reconstruction, source detection and source characterisation. Deep GRU's training lasted ~ 20 minutes on spectra produced from the truth catalogue and ~ 3 minutes on Blobs Finder's predictions on the training set, and it made the predictions on the test set in ~ 9 seconds including I/O operations. Each ResNet's training lasted ~ 2 hours on spectrally focused images (or line emission images for flux density regression) produced from the truth catalogue, ~ 30 minutes on spectrally focused images produced from Blobs Finder and Deep GRU predictions, and predictions on the test set were made in ~ 5 seconds including I/O operations.

Acknowledgements

I want to thank F. Guglielmetti, L. Tychoniec and E. Villard for their supervision of the work carried out in this Chapter and for discussing with me for endless hours. I'm especially grateful to F. Guglielmetti for her invaluable help and support for the development of this Chapter and for support during my stay in Garching as a guest scientist at European Southern Observatory (ESO). I want to thank The IBISCO HPC, the University Federico II supercomputing centre operated by the Department of Physics, for allowing us to extensively use their cluster to simulate the data and carry out all the experiments. Also, I want to thank the European Southern Observatory (ESO) and the Department of Electric Engineering and Information Technologies at the University Federico II for partial financial support. This work was also supported by an ESO internal ALMA development study investigating interferometric image reconstruction methods and partially sponsored by the EU in the framework of the Horizon 2020 SUNDIAL Initial Training Network.

Chapter 5

The SKA Data Challenge 2: Detection and Characterisation of SKA Sources through Deep Learning

> *I must not fear. Fear is the mind-killer. Fear is the little-death that brings total obliteration. I will face my fear. I will permit it to pass over me and through me. And when it has gone past I will turn the inner eye to see its path. Where the fear has gone there will be nothing. Only I will remain.*
>
> Frank Helbert, Dune

The Square Kilometre Array Observatory SKA Observatory (SKAO) is a next-generation radio interferometer currently being built by an international collaboration. The project is currently in a very advanced stage (with two prototypes: The Australian Square Kilometer Array Pathfinder (ASKAP) [72] and MERKAAT [5] already operational) and aims at building the world's largest radio telescope, with eventually over a square kilometre (one million square metres) of collecting area. SKA requires a huge leap forward in both engineering and research & development towards building and delivering a unique instrument which can be rightly considered as the largest scientific endeavour in history. SKA will eventually use thousands of dishes and up to a million low-frequency antennas to explore the radio sky to new depths marking a paradigm shift not only in the way we see the Universe but also in how we undertake scientific investigation. SKA will in fact produce up to 1.5 PBs of raw data/day and more than 100 TBs of processed data/day. This huge amount of data will need to be captured, transported, processed, stored, shared and analysed using automatic methods. Innovations developed in order to enable the SKA data journey will drive forward data technologies across software, hardware and logistics.

In order to support the community to prepare for such rich datasets, the SKAO has established a series of Science Data Challenges (SDCs). Each challenge involves some combination of real or simulated datasets designed to be as similar as possible to the future SKA data. More than 100 participants groups, including myself alongside other COIN Cosmostatistics Initiative (COIN) members, were invited to participate in the SKA Data Challenge 2[1] SKA Data Challenge 2 (SKADC2) which run between the 1st February of 2021 and the 31st of July.

This Chapter is arranged as follows: In Sec. 5.1 we outline relevant information about the SKADC2 and the simulated data product that was made available to its participants; In Sec. 5.2 we describe in details my team solution to the Challenge, a DL based pipeline, and its revision that I made, after the challenge completion, in order to address its shortcomings and greatly boost its performances, and other relevant architectures that participated to the challenge. In Sec. 5.3 we show the results of the challenge, and the improvements we have obtained after the challenge conclusion. In Sec. 5.4 we make our concluding remarks.

[1] https://sdc2.astronomers.skatelescope.org

5.1 The SKA Data Challenge 2

The Challenge consisted in the detection and characterisation of HI sources within a dataset consisting in a 5851 × 5851 × 6668 pixels[2] HI[3] imaging data cube simulating a future SKA MID spectral line observation with the following specifications:

1. 20 square degrees field of view.
2. 7 arcsec beam size, sampled with 2.8 × 2.8 arcsec pixels.
3. 950–1150 MHz bandwidth, sampled with a 30 kHz resolution. This corresponds to a redshift interval $z = 0.235$–0.495[4].
4. Noise consistent with a 2000-hour total observation.
5. Systematics including imperfect continuum subtraction, simulated RFI flagging and excess noise due to RFI.

The HI data cube was accompanied by a radio continuum data cube covering the same field of view at the same spatial resolution, with a 950-1400 MHz frequency range at a 50 MHz frequency resolution. Together with the full-size Challenge dataset, two smaller datasets were made available for development and testing purposes. Generated using the same procedure as the full-size dataset but with a different statistical realisation, the 'development' and 'large development' datasets were provided along with truth catalogues of HI sources. A further 'evaluation' dataset was provided without a truth catalogue, in order to allow teams to validate their methods in

[2] The dimension corresponding to two spatial directions on the plane of the sky and one spectral dimension corresponding to the different frequency channels.

[3] HI stands for Neutral Hydrogen. Neutral Hydrogen produces a hyperfine quantum transition which is detected at 21.1 cm wavelength and is the most used tracer to investigate the distribution of gas in the universe.

[4] Due to the expansion of the universe, distant galaxies recede from us at a speed which is proportional to their distance. As a result of the relativistic Doppler effect, this allows us to derive distances by measuring the wavelength shift of any specific spectral line with respect to its rest frame position. The redshift z is defined as: $z = \frac{\Delta\lambda}{\lambda_0} \simeq \frac{D}{c}H(t_0)$. Where $\Delta\lambda$ is the shift, λ_0 is the rest-frame wav., H is the Hubble constant, c is the speed of light and D is the distance of the galaxy. This simple relation allows once you have selected a specific line, to convert a given frequency range into an observability redshift range.

a blind way prior to application to the full dataset. The evaluation dataset was also used by teams to gain access to the full-size data cube hosted at a SKADC2 partner facility. Access was granted upon submission of a source catalogue based on the evaluation dataset and matching a required format. The development and evaluation datasets were made available for download prior to and during the Challenge. The Challenge organisers gave us computational resources through the *GENCI-IDRIS – Orsay, France* supercomputer. The challenge consists of two separate steps:

Source finding: defined as the detection of the pixels belonging to the galaxies in order to provide the coordinates of their dynamical centre as RA (degrees), Dec (degrees) and central frequency (Hz).

Source characterisation, defined as the recovery of the following properties:

1. Integrated line flux (Jy Hz): the total line flux integrated over the signal $\int F d_{\nu}$.
2. HI size (arcsec): the HI major axis diameter at 1 $M_{\odot}$ pc^{-2}.
3. Line width (km s^{-1}): the observed line width at 20% of its peak.
4. Position angle (degrees): the angle of the major axis of the receding side of the galaxy, measured anticlockwise from North.
5. Inclination angle (degrees): the angle between line-of-sight and a line normal to the plane of the galaxy.

The scoring of the Challenge also reflects the two separate steps:

1. Source Finding scoring: A cross match procedure is used to check for detected sources within the truth catalogue. The cross-match procedure considers the position of a source in the 3D cube, identified by RA, Dec and central frequency. All submitted sources with positions within which a truth catalogue source is in range are recorded as matches. For each submitted source, this range in the spatial and frequency dimensions is determined by the beam-convolved submitted HI size and the line width, respectively. Detections that do not have a true source within this range are recorded as false positives.

Matched detections are further filtered by considering the range of the matched truth sources. Detections which lie outside the beam-convolved HI size and the line width of the matched truth source are at this stage also rejected and recorded as false positives. It is possible that the cross-match returns multiple submitted sources per true source. In that case, all matches are retained and scored individually. The reasoning behind this choice is that components of HI sources, especially in the velocity field, could be correctly identified but interpreted as separate sources. If that were the case, classifying them as false positives would be too much of a penalty. All submitted sources matched to the same true source are inversely weighted by the number of multiple matches during the scoring step. During the cross-matching, it is also possible for more than one true source to be matched with a single submitted source. In these cases, only the match between the submitted source and truth source which yields the lowest multi-parameter error (eq. 5.1) is retained. This procedure ensures that matches in crowded regions will take into account the resemblance of a true source to a submitted source, in addition to its position. A final step is performed to compare the multi-dimensional error with a threshold value, above which any nominally matched submitted sources are discarded and counted as false positives. The multi-parameter error D is calculated using the Euclidean distance between truth and submitted sources in normalised parameter space:

$$D = (D^2_{pos} + D^2_{freq} + D^2_{HI\,size} + D^2_{line\,width} + D^2_{flux})^{\frac{1}{2}}, \qquad (5.1)$$

where the errors on parameters of spatial position, central frequency, line width and integrated line flux have been normalised following the definitions in Table 5.1. The error in HI size is at this stage normalised by the beam-convolved true HI size in order not to lead to the preferential rejection of unresolved sources. The multi-dimensional error threshold is set at 5, i.e. the sum in quadrature of unit normalised error values.

2. Source characterisation: For all detections that have been identified as a match, properties are compared with the truth catalogue and a score is assigned per property and per source. The following

> properties are considered for accuracy: sky position (RA, Dec), HI size, integrated line flux, central frequency, position angle, inclination angle and line width. Each attribute j of a submitted source i contributes a maximum weighted score w_i^j of 1/7, so that the maximum weighted score w_i for a single matched source is 1:
>
> $$w_i = \sum_{j=1}^{7} w_i^j. \tag{5.2}$$
>
> The weighted score of each property of a source is determined by
>
> $$w_i^j = \frac{1}{7} \min \left\{ 1, \frac{thr_j}{err_i^j} \right\}, \tag{5.3}$$
>
> where err_i^j is the error on the attribute and thr_j is a threshold applied to that attribute for all sources. Errors calculated in this step are detailed in Table 5.1, along with corresponding threshold values. Finally, the weighted scores of submitted sources are averaged over any duplicate matches with unique truth sources.

The final score is determined by subtracting the number of false positives N_{false} from the summed weighted scores w_i of all N_{match} unique matched sources:

$$final\ score = \sum_{i}^{N_{match}} w_i - N_{false}. \tag{5.4}$$

5.2 The Pipeline

In this chapter two Deep Learning pipelines are presented: the first one, referred as the COIN pipeline, is showcased in Fig. 5.1 and was developed in collaboration with other COIN scientists within the SKADC2 challenge; the second one, referred as the Revised pipeline, is showcased in Fig. 5.2 and has been developed after the completion of the challenge in order to improve on the obtained results. The second pipeline was built upon the first one in order to solve its two main shortcomings: a high number of false detections (FPs), and poor segmentation performances. The two

Property	Error term	Threshold
RA and Dec, x, y	$D_{pos} = \dfrac{(x-x')^2 + (y-y')^2}{S'}$	0.3
HI size, S	$D_{HI\ size} = \dfrac{\lvert S - S' \rvert}{S'}$	0.3
Integrated line flux, F	$D_{flux} = \dfrac{\lvert F - F' \rvert}{F'}$	0.1
Central frequency, ν	$D_{freq} = \dfrac{\lvert \nu - \nu' \rvert}{w'_{20,\,Hz}}$	0.3
Position angle, θ	$D_{PA} = \lvert \theta - \theta' \rvert$	10
Inclination angle, i	$D_{incl} = \lvert i - i' \rvert$	10
Line width, w_{20}	$D_{line\ width} = \dfrac{\lvert w_{20} - w'_{20} \rvert}{w'_{20}}$	0.3

Table 5.1. Definitions of errors and threshold values for the properties of sources. Prime denotes the attributes of the truth catalogue, x, y are the pixel coordinates corresponding to RA, Dec, ν is the central frequency, S is the HI major axis diameter, f is the source integrated line flux, θ is the position angle, i is the inclination angle, and w_{20} is the HI line width. Calculations of position angles take into account potential angle degeneracies by defining the angle difference as a point on the unit circle and taking the two-argument arctangent of the coordinates of that point: $|\theta - \theta'| = atan2[\sin(\theta - \theta'), \cos(\theta - \theta')]$

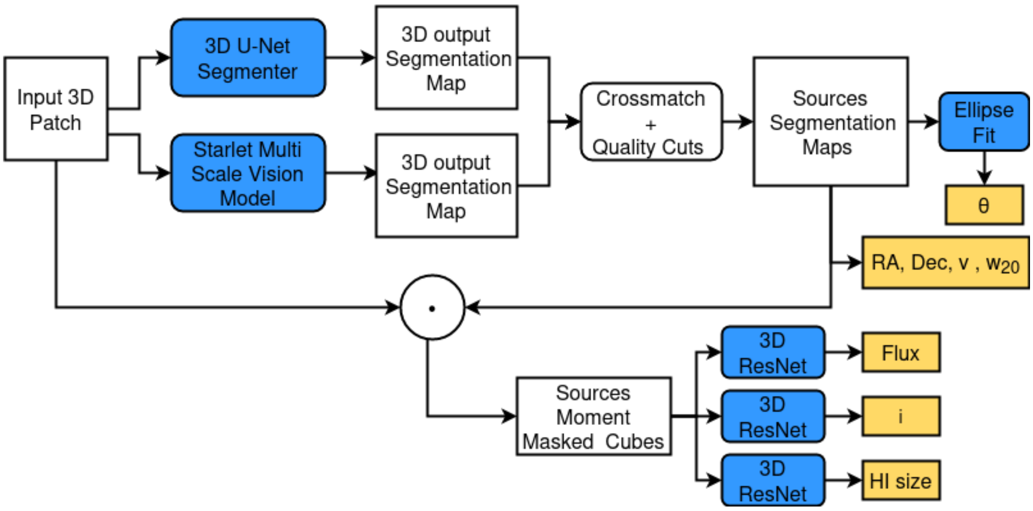
Input 3D Patch
3D U-Net Segmenter
Starlet Multi Scale Vision Model
3D output Segmentation Map
3D output Segmentation Map
Crossmatch + Quality Cuts
Sources Segmentation Maps
Ellipse Fit
θ
RA, Dec, v , w20
Sources Moment Masked Cubes
3D ResNet
3D ResNet
3D ResNet
Flux
i
HI size

Figure 5.1. Schema of the source detection pipeline which competed in the SKADC2 developed in collaboration with COIN members.

pipelines share the same working principles and some common elements. Both pipelines receive as input a 3D $64 \times 64 \times 128$ patch of the input cube and process it in order to predict a corresponding segmentation map masking with 1 pixels belonging to the sources within the cube and with 0 the rest. This operation is performed in parallel by two models: a DL model and our re-implementation of the Starlet Multi-Scale vision model. The Multi-Scale vision model, explained in 2 Sec. 2.3, has been extended to work with 3D data and details about its inner workings are explained later in this Chapter. The DL model is a 3D CAE in the COIN pipeline and 3D ResNet CAE in the revised pipeline. While the working principles of this kind of architectures are explained in Chapter 3 Sec. 3.1.3, their implementations for the purpose of this work are explained later in this Chapter. The output segmentation maps, one output of the DL model, and one output of the Starlet Multi-Scale Vision Model are normalized to the $[0, 1]$ range and combined. For each slice of the resulting cube, a threshold parameter t is used to select relevant pixels, and a friend-of-friend algorithm is employed to link pixels into 2D blobs (a combination of the *label* and *regionprops scikit-learn* [118] functions) which are recorded as a leaf of a graph. The resulting 3D graph is first pruned by removing all blobs containing less than t_p pixels, and then, for each pair of adjoining cube

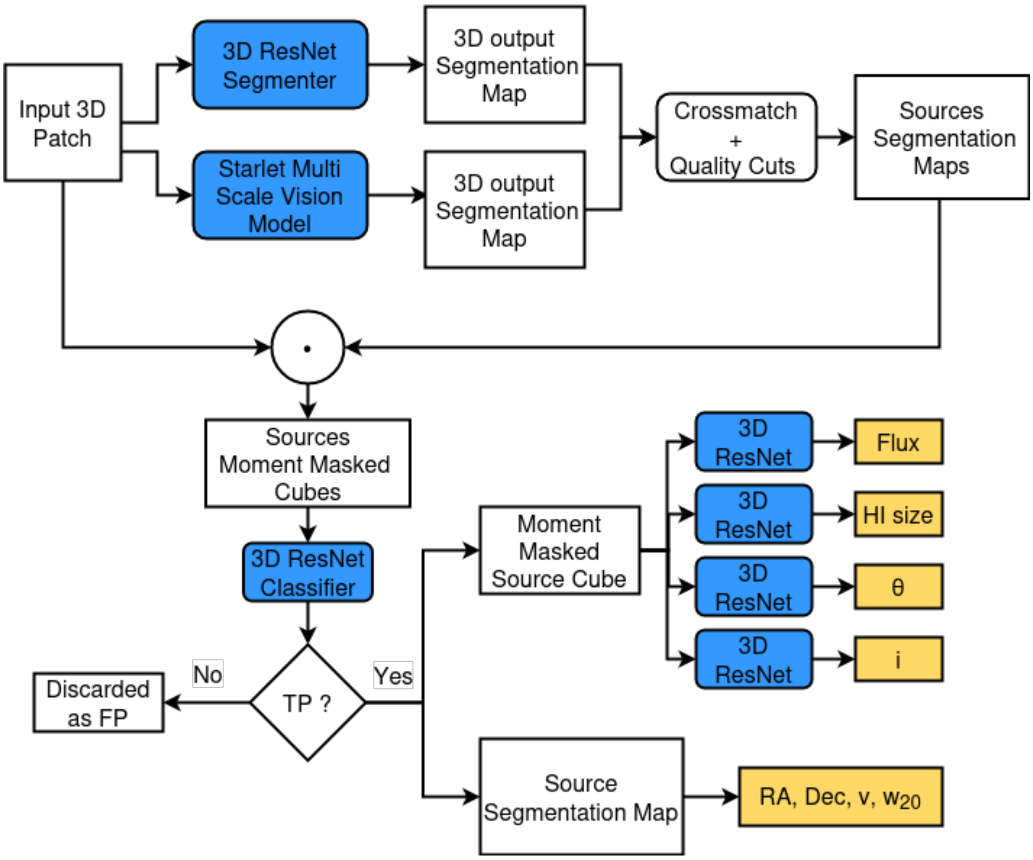
Input 3D Patch
3D ResNet Segmenter
Starlet Multi Scale Vision Model
3D output Segmentation Map
3D output Segmentation Map
Crossmatch + Quality Cuts
Sources Segmentation Maps
·
Sources Moment Masked Cubes
3D ResNet Classifier
TP ?
No
Yes
Discarded as FP
Moment Masked Source Cube
3D ResNet
3D ResNet
3D ResNet
3D ResNet
Flux
HI size
θ
i
Source Segmentation Map
RA, Dec, v, w20

Figure 5.2. Schema of the revised pipeline which I developed to address the shortcomings of the COIN pipeline.

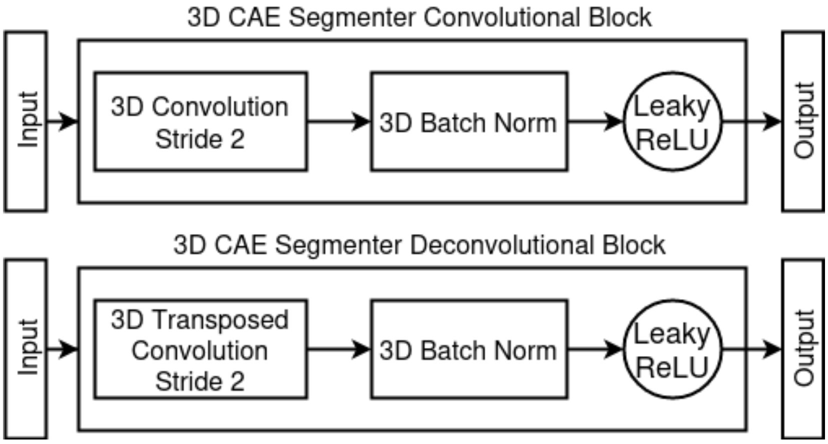
3D CAE Segmenter Convolutional Block
Input
3D Convolution Stride 2
3D Batch Norm
Leaky ReLU
Output
3D CAE Segmenter Deconvolutional Block
Input
3D Transposed Convolution Stride 2
3D Batch Norm
Leaky ReLU
Output

Figure 5.3. Architectures of the COIN CAE Convolutional and Deconvolutional blocks. The Convolutional block is made by a 3D Convolution with a stride of 2 and a kernel size of 3, a 3D Batch Norm Layer and a ReLU activation function. The Deconvolutional block is symmetric with respect to the convolutional block and its made by a 3D Transposed Convolution layer with a stride of 2 and a kernel size of 3, a 3D Batch Norm layer and a ReLU activation function.

slices, a link is set between blobs, if their spatial overlap is more than t_s. In practice, links are created between overlapping blobs in frequency in a fashion similar to that of how relevant sets of coefficients are linked between adjoin scales in the Multi-Vision model. The formed structures (linked blobs) are pruned again by eliminating all structures containing less than t_f links, or in practice, we are eliminating all source candidates which show relevant emission in less than t_f frequency slices. This step is represented by the *Crossmatch + Quality Cuts* step of the pipeline. The surviving candidates (or more precisely the pixels belonging to the candidates) are preserved in the *segmentation map* while the remaining pixels are set to 0. In order to account for the fact the Segmentation networks may have missed source boundaries which, given the nature of the simulated data and the spreading effect of the dirty beam, could be at or below the noise level, each source segmentation map is dilated by a factor t_d.The resulting processed segmentation map cube is multiplied with the input 3D patch to get the *moment masked cube* which contains all the input pixels values belonging to source candidates and the remaining pixels set to 0. For each

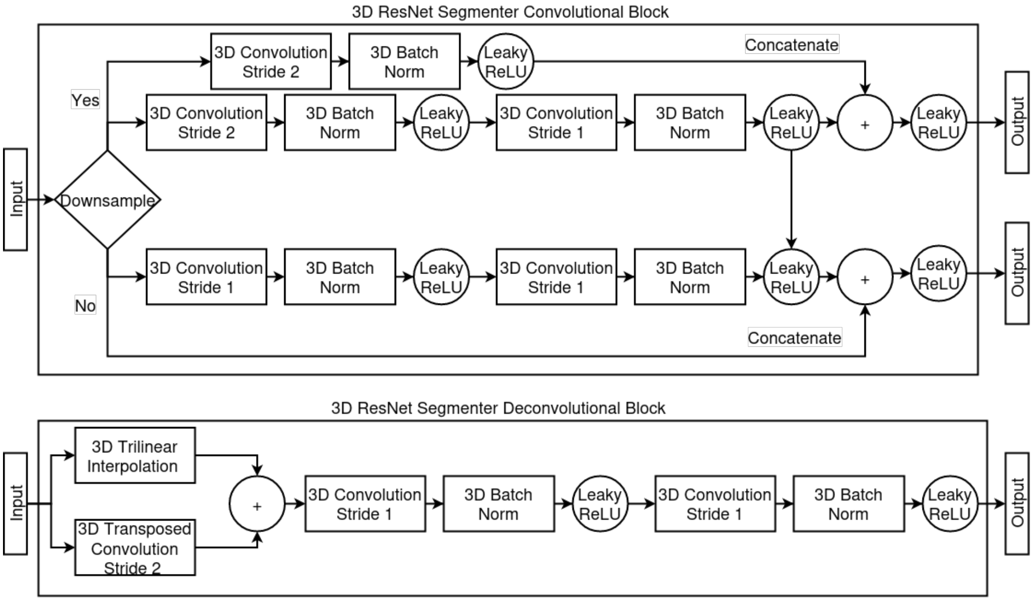
3D ResNet Segmenter Convolutional Block
Input
Downsample
Yes
No
3D Convolution Stride 2
3D Batch Norm
Leaky ReLU
3D Convolution Stride 1
Concatenate
+
Output
3D ResNet Segmenter Deconvolutional Block
3D Trilinear Interpolation
3D Transposed Convolution Stride 2

Figure 5.4. Architectures of the ResNets CAE Convolutional and Deconvolutional blocks. The Convolutional block is divided in two main pathways depending if downsampling must be applied in the layer. In the affirmative case, the 3D convolutions are applied with a stride of 2, otherwise with a stride of 1. As it can be seen, the output of the previous layer is brought forward through a skip-connection and concatenated with the output of the current layer before applying the final activation function. The deconvolutional block is made by two parallel pathways: the learnable upsampling made by a 3D Transposed Convolution with a stride of 2, and the upsampling made by a 3D trilinear interpolation. The output both pathways are concatenated and then passed along to two convolutional blocks with a stride of 1 followed by a 3D Batch Normalization layer and a Leaky ReLU activation function.

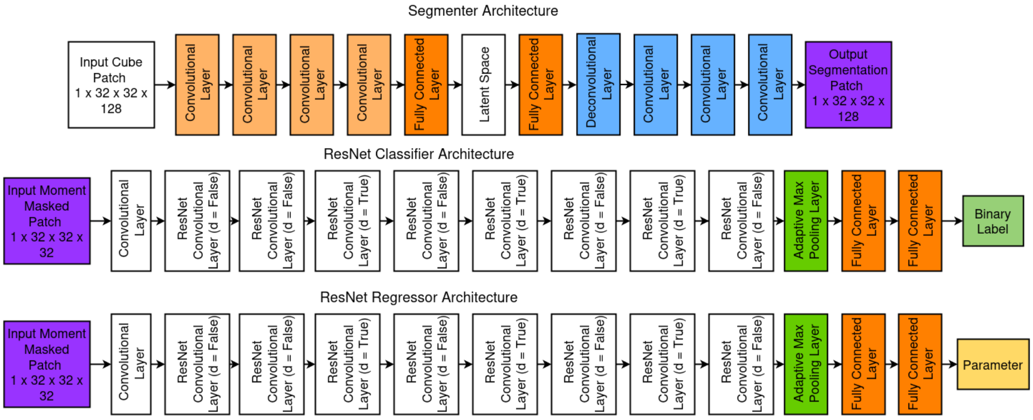
Segmenter Architecture
Input Cube Patch 1 x 32 x 32 x 128
Convolutional Layer
Convolutional Layer
Convolutional Layer
Convolutional Layer
Fully Connected Layer
Latent Space
Fully Connected Layer
Deconvolutional Layer
Convolutional Layer
Convolutional Layer
Convolutional Layer
Output Segmentation Patch 1 x 32 x 32 x 128
ResNet Classifier Architecture
Input Moment Masked Patch 1 x 32 x 32 x 32
Convolutional Layer
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
Adaptive Max Pooling Layer
Fully Connected Layer
Fully Connected Layer
Binary Label
ResNet Regressor Architecture
Input Moment Masked Patch 1 x 32 x 32 x 32
Convolutional Layer
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
ResNet Convolutional Layer (d = True)
ResNet Convolutional Layer (d = False)
Adaptive Max Pooling Layer
Fully Connected Layer
Fully Connected Layer
Parameter

Figure 5.5. From top to bottom: top) architecture of the CAE Segmenter (COIN Pipeline). The architecture in the Revised Pipeline is obtained by switching the convolutional layers with ResNet convolutional layers; middle) the architecture of the ResNet Classifier; bottom) the architecture of the ResNet parameter regressor.

detected source, we extract two $32 \times 32 \times 128$ pixel patches centred around the source bounding box, one from the segmentation-masked cube and one from the moment-masked cube. These are the source 3D segmentation map and the source 3D source moment-masked cube. In both pipelines, the source 3D segmentation map is used to estimate the source position in the cube (RA, Dec and ν) as the moment-masked weighted barycenters of the pixels belonging to the source in the x, y and z axes of the cube. The source line width w_{20} is estimated as half the length in the frequency of the source segmentation map. The moment-masked cubes are then fed to specialised ResNet regressors each one tasked to predict one of the source morphological parameters, i.e. the integrated line flux F, the HI size S, the inclination angle i and the position angle θ in the Revised pipeline. Details about the ResNet regressors architectures are described later in this Chapter. Two are the main differences between the COIN and the Revised pipeline:

1. the COIN pipeline estimates the projection angle θ by first integrating the source 3D segmentation map in order to get a 2D image which is then normalized to 1. An elliptical model (the Astropy [9] *ellipse* function) is fitted to the 2D image using as first estimates of the el-

lipse position, major and minor axes the coordinates and dimensions extracted from the segmentation map earlier in the pipeline. The ResNet utilises a specialized ResNet to predict the projection angle θ as the other morphological parameters;

2. in order to remove possible false detections, the Revised pipeline, before estimating the source parameters, employs a 3D ResNet Classifier. Only source candidates with a probability of being a true source higher than t_d are retained and passed along in order to estimate their morphological parameters, the remaining sources are discarded as false detections.

5.2.1 The CAE Segmenter

The CAE Segmenter is made by an Encoder and a Decoder network. The Encoder is made by four convolutional blocks that progressively reduce the spatial dimensions of the input while increasing the number of channels or feature maps and a final fully connected layer followed by a Leaky ReLU activation function. The Decoder has a symmetric architecture, i.e. a fully connected layer, followed by four deconvolutional blocks and a final identity layer which is constituted by a 3D Convolution followed by a Sigmoid activation function whose purpose is to normalize the output to the $[0, 1]$ range. The structures of the convolutional and deconvolutional blocks are showcased in Fig.5.3. Convolutional blocks are made by a 3D Convolution layer with a kernel size of 3 and a stride of 2 a 3D Batch Norm layer with a kernel size of 3 and a Leaky ReLU activation function. Deconvolutional blocks are made by a 3D Transposed Convolution layer with a kernel size of 3 and a stride of 2, a 3D Batch Norm layer with a kernel size of 3, and a Leaky ReLU activation function. The Identity layer is made by a 3D Convolution with a kernel size of 1 and a stride of 2 followed by a 3D Batch Normalization layer and a Sigmoid activation function. A detailed description of the spatial transformation performed by the CAE Segmenter can be seen in Tab. 5.2. The CAE is trained with the Dice loss [145] defined as follows:

$$DL(x, y) = 1 - \frac{2xy + 1}{x + y + 1} \tag{5.5}$$

Block Name	Input Size	Output Size
Conv Block 1	$[b, 1, 64, 64, 128]$	$[b, 8, 32, 32, 64]$
Conv Block 2	$[b, 8, 32, 32, 64]$	$[b, 16, 16, 16, 32]$
Conv Block 3	$[b, 16, 16, 16, 32]$	$[b, 32, 8, 8, 16]$
Conv Block 4	$[b, 32, 8, 8, 16]$	$[b, 64, 4, 4, 8]$
FC 1	$[b, 64 \times 4 \times 4 \times 8]$	$[b, 1024]$
FC 2	$[b, 1024]$	$[b, 64 \times 4 \times 4 \times 8]$
DeConv Block 1	$[b, 64, 4, 4, 8]$	$[b, 32, 8, 8, 16]$
DeConv Block 2	$[b, 32, 8, 8, 16]$	$[b, 16, 16, 16, 32]$
DeConv Block 3	$[b, 16, 16, 16, 32]$	$[b, 8, 32, 32, 64]$
DeConv Block 4	$[b, 8, 32, 32, 64]$	$[b, 1, 64, 64, 128]$
Final Block	$[b, 1, 64, 64, 128]$	$[b, 1, 64, 64, 128]$

Table 5.2. Input and Output shapes for each layer of the COIN and ResNet CAE Segmenter, where b indicates the batch size, and the horizontal line separates the Encoder from the Decoder network.

where 1 is added in the numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when $y = x = 0$.

5.2.2 The ResNet CAE Segmenter

The ResNet CAE Segmenter architecture is obtained by substituting the Convolutional and Deconvolutional block of the CAE Segmenter with the respective ResNet-like counterparts which are shown in Fig. 5.4. The Convolutional block is divided into two main pathways depending if a reduction of the spatial dimensions of the input wants to be achieved (*downsampling*). In the affirmative case, the input is first processed with a 3D Convolution with a kernel size of 3 and a stride of 2, a 3D Batch Normalization layer with a kernel size of 3, a Leaky ReLU activation function and then processed with a 3D Convolution with a kernel size of 3 and a stride of 1, a 3D Batch Norm layer and a Leaky ReLU activation function. The output of the previous operation is concatenated with the output of the second convolution and processed with a Leaky ReLU activation function. In the negative case, the input is first processed with two blocks made by a 3D Convolution with a kernel size of 3 and a stride of 1, a 3D Batch Normalization and a leaky ReLU activation function, and then concatenated with the output of the previous layer and processed with a Leaky ReLU activation function. Each deconvolutional block is constituted by two parallel pathways: a 3D Trilinear interpolation with stride of 2, followed by a 3D Batch Normalization layer and a Leaky ReLU activation function (upsampling block); and, a 3D Transposed Convolution layer with stride of 2 and a kernel size of 3, followed by a 3D Batch Normalization layer and a Leaky ReLU activation function (learnable upsampling block). The output of the upsampling block and learnable upsampling block are then concatenated and passed along to two convolutional blocks constituted by a 3D Convolution layer with stride 1 and a kernel size of 3, a 3D Batch Normalization layer and a Leaky ReLU activation function. The trilinear upsampling operation transforms the input layer in the desired spatial resolution without using any parameter (and thus it has a low impact on model complexity) while the resulting features should preserve most of the information of the original features. These upsampled features are concatenated to the output of the parametric upsampling performed through the Transposed Convolution in order to create a residual-like connection. This additive upsampling [161] should improve prediction capabilities and remove gridding artefacts produced by the subsequent Transposed Convolutions. The ResNet CAE Segmenter performs the same spatial transformations performed by the CAE Segmenter and showcased in Tab. 5.2 with the difference that it has almost double the amount of parameters (given that the number of convolutions is more than doubled). The network is also trained with the Dice loss.

5.2.3 The ResNet Classifier and Regressor

The ResNet Classifier and Regressor networks share the same architecture with the exception of the final activation function used in their last Fully Connected Layer and the choice of loss function we use to train them. The architectures of both networks are showcased in Fig. 5.5. Both models start with a Convolutional layer made by a 3D Convolution with a kernel size of 3 and a stride of 1 followed by a 3D Max Pooling layer, a 3D Batch Normalization layer and a ReLU activation function. This first layer preprocesses the input by halving its spatial dimensions and is followed by 8 ResNet Convolutional block (showcased in Fig. 5.4). Each pair of 2 consecutive blocks halves the spatial dimensions of its input and doubles the number of features. The input is then fed to a 3D Adaptive Max Pooling layer [94] which performs the same operation as that of Max Pooling (see Chapter 3 Sec. 3.1.2) but *stride* and *kernel size* are automatically computed in order to reduce kernels overlap. The three parameters are computed, for each dimension of the input, as follows:

- $stride = floor(S_i/S_o)$;
- $kernel\ size = S_i - (S_o - 1) \times stride$;

where S_i and S_o are the sizes in pixels of respectively input and output. The flattened feature vector is then passed to the first fully connected layer followed by a ReLU activation function and then a second fully connected layer. In the case of the Classifier, this latter fully connected layer maps the input to a mono-dimensional vector with a length equal to 2 which is then processed through a Sigmoid activation function in order to normalise it and give it a probabilistic interpretation. In the case of the Regressor, the fully connected layer maps the input to a single scalar which is the value of the parameter of interest for the ResNet. The ResNet Classifier is trained with the binary cross-entropy (BCE) loss between the output probabilities and the true class labels. The BCE loss is defined as follows:

$$BCE(x, y) = \frac{1}{N}\sum_{i=0}^{N}\Bigg(- y_i log(p_i) + (1 - y_i) log(1 - p_i)\Bigg) \tag{5.6}$$

where p_i is the probability that the ith input patch contains a true source, while $(1 - p_i)$ is the probability that it is empty. The ResNet Regressors are all trained with the $l - 1$ loss function defined as follows:

$$l_1(x, y) = \frac{1}{N}\sum_{i=0}^{N} |x_i - y_i| \tag{5.7}$$

5.2.4 The Multi-Vision Model

Our python implementation of the Multi-Vision Model (see Chapter 2 Sec. 2.3) is based on the *pymultiscale*[5] package from which we take their

[5] https://github.com/broxtronix/pymultiscale

implementation of the 3D Starlet Transform and Inverse Transform, and the *scipy optimize* [157] package *least_ squares* method. The method takes as arguments: a real function which computes, given the input patch and the output segmentation map and a set of parameters (the 3D array of coefficients weights, which are the thresholding operator P_w as described in the Multi-Vision model algorithm), the vector of residuals between the weighted reconstruction of the input patch and the target output segmentation map; a loss function which in our case is the l_1 loss between the reconstructed 3D patches and the target segmentation map; and a minimisation algorithm which in our case is the *scipy's* implementation of the Trust Region Reflective algorithm [20]. We also set the number of function evaluations or iterations to $10,000$ and optimised the noise threshold for the selection of significant wavelet coefficients K (see Eq. 2.43) and the object overlap threshold t_O through a grid-search strategy. K was optimised within the close interval $[1, 6]$ with a step size of 0.1 and t_O between 0.5 and 0.8 with a step size of 0.1. The purpose of the Multi-Vision model is that of complementing the DL models which for example, given the low frequency of bright sources in the data, may disregard them as outliers with respect ot the way more frequent low brightness sources in the data. On the contrary, these bright sources are the easiest to find with traditional algorithms such as The Multi-Vision model or SoFiA.

5.2.5 Training and Optimization Strategies

In both pipelines, the data flows from one DL model to the next. In particular, the moment masked source patches extracted through the Segmenters and Multi Vision model's segmentations of the input cube patch, are passed along to the ResNet classifier for false detections identification and removal and to the ResNet regressors and other pipeline components to regress the morphological parameters of the detected sources. Given that the challenge organisers made available to use both data cubes and the parameters of sources within them, all pipeline models can be trained at the same time in parallel. All networks are trained using the Adam Optimizer for a thousand epochs and with a *learning rate* of 10^{-3}. The Segmentation networks are trained with pairs of input cube patches and corresponding segmentation maps ($64 \times 64 \times 128$ pixels). The segmentation maps are produced from the source parameters through the following set of equations:

$$\begin{gathered}D_{HI_{pixel}} = D_{HI_{arcsec}}/(2 \times \Delta_{xy_{arcsec}}) \\ b = D_{HI_{pixel}} \times \sqrt{cos(i)^2 + \alpha^2 sin(i)^2} \\ n_{channels} = \nu_0 \times w_{20}/(c\Delta_z)\end{gathered} \tag{5.8}$$

$$e(x,y,z) = \begin{cases} 1, & \text{if } \left[\frac{(x-x_0)cos\theta+(y-y_0)sin\theta}{D_{HI}}\right]^2 + \left[\frac{-(x-x_0)sin\theta+(y-y_0)cos\theta}{b}\right]^2 \leq 1 \\ & \text{and} \quad z \in [z_0 - n_{channels}/2, z_0 + n_{channels}/2]. \\ 0, & \text{otherwise.} \end{cases} \tag{5.9}$$

where D_{HI} is the galaxy's major axis, b is the minor axis, i is the inclination angle, θ is the position angle, x_0, y_0 and z_0 are the source coordinates in pixels over the three cube axes, Δ_{xy} and Δ_z are the pixel sizes in arcseconds and Kms^{-1}, and α is a scaling parameter set to 0.2. The SKA segmentation problem is highly imbalanced with a ratio between voxels belonging to sources to the background of $\simeq 10^{-3}$. A naive subdivision of the development cube ($1286 \times 1286 \times 6668$ pixels) in patches of $64 \times 64 \times 128$ pixels would thus result in a respective highly imbalance dataset constituted of mostly of *empty* 3D patches. For this reason, we subdivide the development cube along the x-axis setting the upper 80% to train the networks and the lower 20% to validate their performances and, for each training iteration, we sample, from galaxies falling within the training and testing parts of the cube respectively, 128 patches centred around sources and 128 empty patches. To allow the networks to be rotationally invariant and also increase the parameter's sampling in the training set, through data augmentation. Each time a 3D patch is sampled, it is also randomly rotated in the xy plane. Source parameters are also modified to account for the rotation. Through the explained sampling technique, we provide the segmentation network with a balanced training set. The same training set is also used to train the ResNet Classifier, while only the patches containing sources and the corresponding parameters (F, i, S, θ) are used to train the ResNet Regressors. To prevent overfitting the data batches in the first training iterations, given the models' inherent initial instability due to the random initialisation of their weights, we adopt a *warm up* strategy for the learning rate [52] in which we start with a learning rate of 0, and we uniformly increase it to 1×10^{-3} in the first 10 iterations. The models are trained for a maximum of 1000 epochs, but we also employ an early stopping criterion based on the validation loss. If no improvement of validation loss with respect to the moving average of the last 10 validation losses is registered for 10 consecutive steps, then training is halted. The problem with this training strategy is that it would not take into account the fact that the Classification and Regression ResNets, in production, will not receive perfect source segmentation maps and thus moment masked source cubes but the product of the imperfect prediction of the previous models in the pipeline schema. The ResNet Classifier could receive (as input) source moment masked cubes which contain only partially the source emission, and the Regression ResNets misclassified false positives or correctly classified source moment masked cubes containing only partially the source emission. To account for this, we first train all models in parallel in order for them to learn how to solve their respective problems while optimising the pipeline total training time (which benefits from the fact that the models' training are carried on at the same time) following the strategies described above, and then we train each model (with the same training strategies that we have outlined, and with the exclusion of the Segmentation networks which are the first models in the two pipelines) on the un-augmented training set predictions of the previous model. In this way, each model should be able to correct for the mistakes (biases in the data) of the previous one. Each model is trained in this latter fashion for 300 iterations. The *Crossmatch and Quality Cuts* step of the pipeline utilises several threshold-based criteria to preselect and process worthy source candidates: the source probability threshold t, the minimum number of pixels within a blob t_p, the spatial overlap threshold t_s to link blobs along the frequency axis, the minimum number of slices within a source t_f, and the segmentation dilation factor t_d. Also the ResNet Classifier outputs, for each potential source, a probability score p and thus the threshold which we use to deem a source as True must be tuned to maximise the Challenge score. Clearly the number of detected sources and the number of false positives depends on these parameters and, in order to fine-tune them we

Table 5.3. Sampling intervals for the optimization of the *Crossmatch and Quality Cuts* thresholds. The first column shows the threshold name, the second the range from which the threshold values are sampled, the third the sampling interval.

Parameter Name	Range	
Source probability threshold t	[0.1 – 0.9]	0.05
Minimum number of pixels within a blob t_p	[3 - 10]	1
Blobs spatial overlap for frequency linking t_s	[0.3 – 0.9]	0.01
Minimum number of frequency slices t_f	[3 – 10]	1
Segmentation dilatation factor t_d	[1 – 5]	0.5

proceed as follows: once all the models have been trained to solve their respective problems, we employ them as a functional mapping between their respective inputs and outputs and optimise the parameters through a grid search strategy by randomly sampling each parameter value independently within the ranges and sampling intervals outlined in Table 5.3. We perform 1000 validation runs using both the training and validation sets, and the challenge final score (see. Eq. 5.4) as a loss function. The performances obtained on the full blind cube reported in Table 5.4 are obtained utilizing respectively the following configurations:

- COIN: $t = 0.65$, $t_p = 6$, $t_s = 0.55$, $t_f = 7$, $t_d = 1.5$;
- COIN Revised: $t = 0.7$, $t_p = 4$, $t_s = 0.72$, $t_f = 9$, $t_d = 1.5$, $p = 0.71$

5.2.6 Other Competing Pipelines

Several other DL-based pipeline participated to the challenge:

1. **FORSKA-Sweden**: their pipeline is made by a combination of a U-Net Segmenter and SoFiA [160]. The U-Net architecture [129] was used with an encoder of a ResNet architecture and the initial weights were pretrained from ImageNet and were provided by the PyTorch-based Segmentation Models package [75]. Each 2D $k \times k$ filter of the pretrained model was converted to a 3D filter through the procedure outlined in [163]. They aligned the 2D filters to the spatial plane, and repeated the same filter for k frequencies to

obtain a 3D $k \times k \times k$ filter. The Segmenter is trained on binary segmentation map produced in a similar fashion to those created by us and trained the model with a Dice loss. The training is carried with an Adam optimizer by sampling 128 $32 \times 32 \times 32$ pixel cube patches and corresponding 128 empty patches in order to oversample the number of sources within the dataset and speed-up training. Once the segmentation map is obtained, they feed it to the SoFiA package which is tasked to detect sources and regress their morphological parameters. SoFiA relies on several parametric choices, which they optimise by directly using the Challenge final score;

2. **MINERVA**: their pipeline is composed by two parallel pipelines whose results are combined together: the first pipeline is a highly customized version of a YOLO (You Only Look Once) network [124] which is a regression-based CNN developed for image segmentation. The network is trained on patches of $48 \times 48 \times 192$ pixels and performs detection based on a grid of size $6 \times 6 \times 12$. Each grid element is associated to a candidate detection with the following parameters: the positions x, y, z of the object within the grid element, the widths of the 3D bounding box w, h, d encompassing the object, and an objectness score O that expresses the probability of said object being a true source. YOLO networks are usually made to predict a class for each object, but the MINERVA team modified it to instead regress the source morphological parameters.

3. **Team SoFiA**: team SoFiA made use of the homonym package to detect and characterize sources. After flagging of bright continuum sources and standardization of each cube channel, the SoFiA S+C finder was run with a detection threshold of 3.8 times the noise level, spatial filter sizes of 0, 3, and 6 pixels and spectral filter sizes of 0, 3, 7, 15 and 31 channels, a linking radius of 2 pixels and a minimum size requirement of 3 pixels/channels. Based on tests using the development cube, they optimized the challenge score by removing all detections with $n_{pix} < 700$, $s < -0.00135 \times (n_{pix} - 942)$ or $f > 0.18 \times SNR + 0.17$ where N_pix is the number of pixels within the 3D source mask, s is the skewness of the flux density values within the mask and f is the filling factor of the source mask within its rect-

angular bounding box, and SNR is the signal to noise ration of the detection. As the calculation of the D_{HI} and i requires the spatial deconvolution of the sources, they employ a constant value of 8.5 arcseconds and 57.3 degrees for all unresolved detections. The second pipeline called the Convolutional Hybrid Ad-Hoc pipeline (CHADHOC) is made by a CNN to classify true sources among the detections and a set of CNNs to regress the source parameters. For the detection step, they employ a classical filtering + SNR thresholding algorithm. First, the cube is pre-processed by convolving it with a smoothing Gaussian filter with a width of 600 kHz, and then each frequency slice is converted into an SNR map. All pixels below an SNR of 2.2 are filtered out, while remaining pixels are aggregated into sources through a friend-of-friends algorithm with a linking radius of 2 pixels. The position of each source is computed by averaging its pixel coordinates. The selection step is performed by a CNN made by 5 convolutional layers (8, 16, 32, 32 and 8 filters) and 3 dense layers (96, 32 and 2 neurons). Batch normalization, dropout and pooling are inserted between every convolutional and dense layers. The network is trained with unsmoothed cutouts of $38 \times 38 \times 100$ pixels around the 10^5 brightest detections made in the development cube by assigning a True/False label to each detection. The CNN outputs the probability that the given input patch is a true source and a cut must be made in order to maximize the Challenge metric. They optimise this parameter, like us, outside their training. The CNN Regressors are trained with cutouts made around the 1300 brightest sources in the truth catalogue and data augmentation is performed by flipping the sampled patches. The catalogues made by the two pipelines are crossmatched together through a distance-based threshold and source properties are averaged together. This latter threshold is also optimized to maximise the Challenge score.

4. **NAOC-Tianlai**: their pipeline is based on Sofia-2 [159].

5. **HI-FRIENDS**: their pipeline is also based on Sofia-2 [159].

6. **EPFL**: their pipeline begins by subdividing the cube along its spatial dimensions and denoising each subcube using 3D wavelet filtering

(2D spatial filtering with the Isotropic Undecimated Wavelet Transform [142], 1D frequency filtering with the decimated 9/7 wavelet transform [158]). A joint-likelihood model is computed from the residual noise in the data cube and it is later used to identify HI source candidates through null hypothesis testing in a sliding window fashion along the frequency axis. Voxels with a likelihood higher than a given threshold are merged into islands and cuts are performed on the basis of the island's size and morphologies to reject false positives. A CNN is then used to classify source candidates and further reject false positives and then Inception CNNs are used to compute sources parameters.

7. **Spardha**: their pipeline is also based on SOFIA-2 [159];

8. **Starmech**: their pipeline is also based on SOFIA-2 [159];

9. **JLRAT**: their pipeline divides the whole dataset into 320 × 320 × 160 patches which are then fed to a CNN which works in the frequency domain to find signals of interest. The network is trained with spectra as input and the output is a binary mask with the candidate location in frequency. The detected spectra are correlated with known spectral in the development cube true dot product to get correlations in the spatial domain. A two-dimensional Gaussian function is then used to fit the moment zero map produced by multiplying the found 3D segmentation map with the original cube to produce a best-fit ellipse from which source morphological properties can be obtained. The flux integral is obtained by integrating pixels belonging to the ellipse both in space and in frequency;

10. **HIRAXers**: their pipeline is based on two DL models: U^2net [121] to detect sources, and a modified version of HighRes3DNet [92] to regress their morphological parameters. The U^2net model is trained with 3D segmentation map created from the truth catalogue. The 3D output of the U^2net is analysed with a peak finder algorithm and all pixels larger than its 27 neighbours are selected as potential sources. These are then passed along to a modified 8-layer HighRes3DNet which regresses all source parameters.

5.3 Results and Analysis

Table 5.4 shows the number of detections N_d, the number of matches N_m, the number of false positives N_f, the reliability R, completeness C, and accuracy A for all pipelines on the challenge *evaluation* full data cube. Reliability, completeness and accuracy of source properties measurements are defined as follows:

$$R = \frac{N_m}{N_d} = \frac{N_m}{N_m + N_f} \tag{5.10}$$

$$C = \frac{N_m}{N_t} \tag{5.11}$$

$$A = \frac{\sum_i^{N_m} w_i}{N_m} \tag{5.12}$$

where $N_t = 233,246$ is the total number of sources within the cube and w_i are the weighted scores for each source. We report, alongside those of the other participating pipelines, the performances obtained on the *evaluation* full cube dataset containing $233,246$ sources by three pipelines: 1. with the acronym COIN-DC we report the results obtained utilising the first pipeline created in collaboration with the other COIN team members as reported in the Challenge leaderboard and paper. Due to the time constraint of the challenge, we could only train on part of the smaller development cube and could not apply the pipeline in time on the entirety of the evaluation cube due to several issues: constraints in the job submission on *GENCI-IDRIS – Orsay, France* supercomputer which resulted in longer times to train and debug the pipelines, bugs in the final cross-matching code, and bugs in the ground truth segmentation maps. This resulted in the submission of results for only roughly 1/10th of the cube. Moreover, for the same aforementioned reason, we could not perform the optimization of the *Crossmatch and Quality Cuts* parameters resulting in a high number of false positives; 2. with the acronym COIN we report the results obtained using the COIN pipeline but this time after debugging, i.e. obtaining results for the full evaluation cube after performing the hyper-parameters optimization; 3. with the acronym COIN-REV we report the performances obtained using the Revised pipeline. As it can be seen, nevertheless being

Pipeline	Score	N_d	N_m	N_f	R	C	A
COIN-REV	23689	31232	30821	967	0.97	0.132	0.80
MINERVA	23254	32652	30841	1794	0.94	0.132	0.81
FORSKA-Sweden	22489	33294	31507	1798	0.94	0.135	0.77
Team SoFiA	16822	24923	23486	1446	0.94	0.100	0.78
NAOC-Tianlai	14416	29151	26020	3117	0.89	0.111	0.67
HI-FRIENDS	13903	21903	20828	1073	0.95	0.089	0.72
EPFL	8515	19116	16742	2369	0.87	0.071	0.65
COIN	5660	28974	26213	13214	0.66	0.11	0.72
Spardha	5615	18000	13513	4480	0.75	0.057	0.75
Starmech	2096	27799	17560	10224	0.63	0.075	0.70
JLRAT	1080	2100	1918	182	0.91	0.008	0.66
COIN-DC	-2	29	17	12	0.58	0.001	0.60
HIRAXers	-2	2	0	2	0.00	0.0	0.00
SHAO	-471	471	0	471	0.00	0.0	0.00

Table 5.4. Performances obtained by all competing pipelines on the full *evaluation* cube. Columns show the score obtained through Eq. 5.4 (Score), the number of detected sources (N_d), the number of matched sources (N_m), the number of false positives N_f, the reliability (R), completeness (C), and accuracy A. In bold are highlighted the results obtained with the COIN pipeline at the challenge's end without hyperparameters optimization, and with inference performed only on a subsample of the full *evaluation* cub (**COIN-DC**), the results obtained with the fully optimised COIN pipeline and with inference performed on the full *evaluation* cube (**COIN**), and the results obtained with the Revised pipeline on the full evaluation cube (**COIN-REV**).

in the top 10 scores of 100 participating teams, the COIN-DC pipeline performed poorly in both the source detection and source characterisation parts of the Challenge. The main culprit of the poor performance in detecting sources is to be found in a bug within the code which performed the creation of the ground truth segmentation maps which resulted in masks shifted (in frequency) with respect to the true source emissions for all but the brightest and biggest sources within the dataset. This resulted in both pour segmentation performances and wrongly constructed input moment masked cubes for the ResNets and other criteria by which source parameters are computed. Also, a high number of false positives is detected achieving roughly a 50:50 ratio between true and false positives for 0.25 deg^2 cutouts of the evaluation cube on which we performed the inference. The frequency shift bias in the pipeline had particularly a great effect on the integrated line flux estimation given that the moment-masked source cubes only containing partially the true source emissions. On the other side, the pipeline obtained competing results on the other morphological parameters regressions with competing pipelines [60]. While the removal of the bugs within the pipeline and the optimization of the source selection hyperparameters greatly increased the number of detected sources from 29 to $28,974$ and substantially increased reliability and completeness, we still obtained a high number of false positives which heavily penalised the final challenge score. The addition of the ResNet Classifier in the Revised pipeline greatly improves the number of false positives achieving the highest reliability among the competing pipelines. Regarding the number of detected sources and the performances in regressing source parameters, they are consistent with those of other high-scoring DL-based pipelines (MINERVA and FORSKA-Sweden) with the difference that our pipeline is more consistent with other non-ML-based methods (SoFiA, HIFRIENDS) in the brighter end of the integrated line flux range. This behaviour can be explained due to the fact that MINERVA and FORSKA-Sweden employ only DL-based model in the source detection parts of their respective pipelines. Our ResNet Segmenter, in fact, suffered from the same decrease in performance.

Figure 5.6 show the number of matched sources plotted over the number of true sources per bin of integrated line flux integral. To plot the distribution of matched sources, true line flux values are employed. Fig.

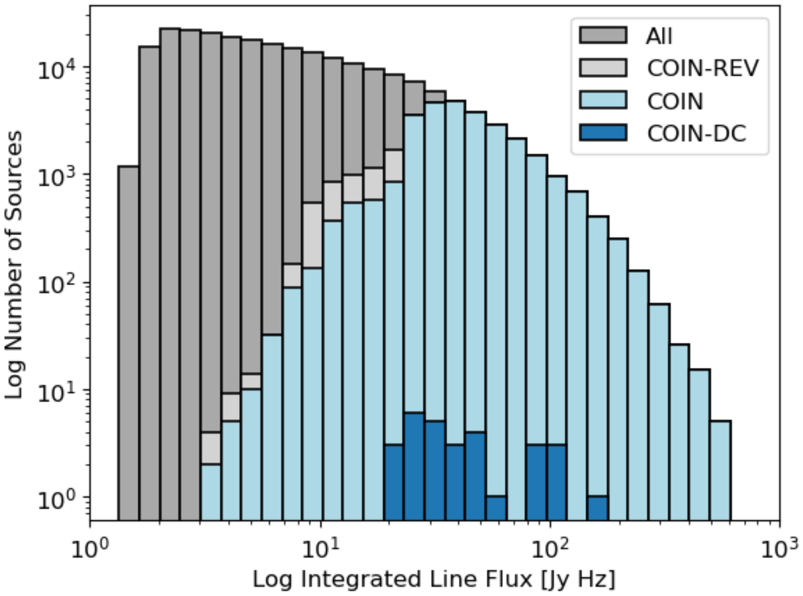

Figure 5.6. Sources in the full Challenge evaluation dataset are binned by their integrated line flux value. In dark grey we show the distribution of all sources within the cube, in light grey truth values of sources detected by the Revised Pipeline, in cyan those of sources detected by the COIN Pipeline, in blue those of sources detected by the COIN-DC pipeline and reported in the Challenge result paper [60].

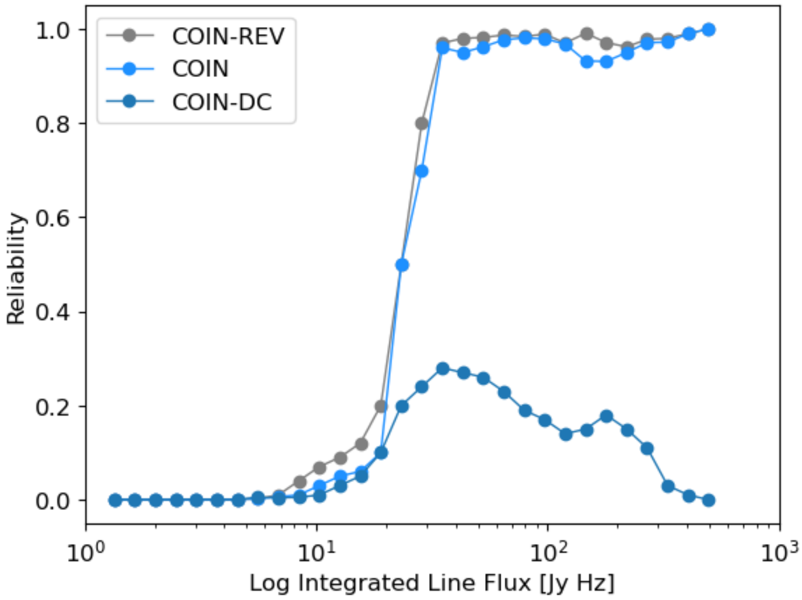

Figure 5.7. Reliability, defined as the number of matches divided by the number of detections, is plotted for the three pipelines as a function of the integrated line flux: the Revised pipeline (dark gray), the COIN pipeline (light blue), and the COIN-DC pipeline as reported in the Challenge result paper [60].

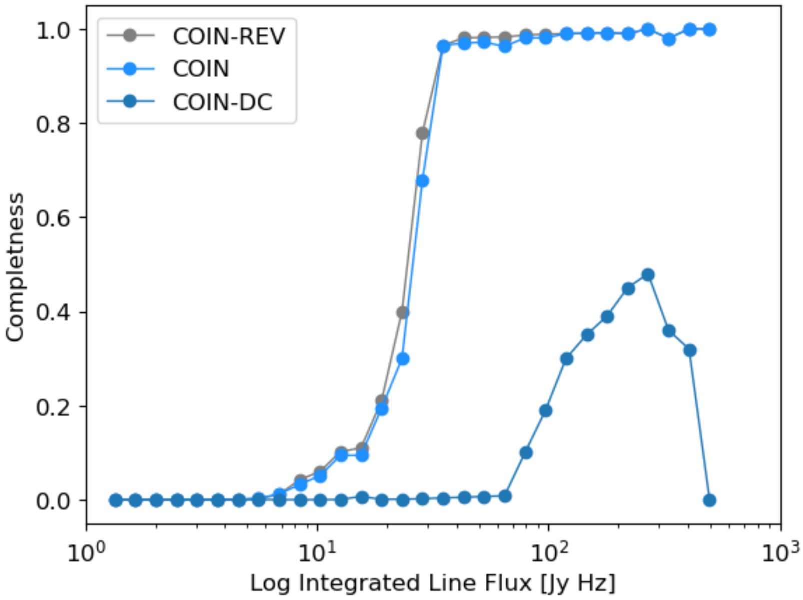

Figure 5.8. Completeness, defined as the number of matches divided by the number of truth catalogue sources, is plotted for the three pipelines as a function of the integrated line flux: the Revised pipeline (dark grey), the COIN pipeline (light blue), and the COIN-DC pipeline as reported in the Challenge result paper [60].

5.7 and Fig. 5.8 show, respectively, the reliability and completeness values as functions of the integrated line flux. Given the absence of truth values for false positives, only scores for matched sources are used.

5.4 Discussion and Conclusions

The application of pipelines based on DL has shown promise in the recovery and characterisation of HI sources as can be seen by the fact that our pipeline and the other two DL-based pipelines (MINERVA and FORSKA-Sweden) achieve the highest scores among all competing pipelines. These higher scores with respect to the several wrappers of SoFiA-2 which compete in the pipeline (Team SoFiA, NAOC-Tianlai, HI-FRIENDS, Spardha, Starmech), come with the caveat that all DL models have required extensive training on the simulated data which may differ substantially from the distribution of real data due to sample selection-bias and overfitting. As for the case of the detection and characterisation of ALMA sources (see Chapter 4), also in this setting, all DL-based methods show a fall in characterisation accuracy on the very brightest sources. Again this behaviour is related to the fact that very few of them were actually simulated in the data and thus they resulted in being severely undersampled in both training and validation sets. From the results obtained by improving the pipeline presented in the Challenge paper, we can confirm that under $\simeq 20$ Jy Hz no pipeline can reliably detect sources with completeness and reliability rapidly dropping to unreliable ranges. Through the combination of an ensemble of ML and non-ML models, our pipeline is capable of improving on both classes of pipelines competing in the challenge. While the COIN-DC pipeline was trained on the *GENCI-IDRIS – Orsay, France* supercomputer, all subsequent pipelines were trained and tested on the IBISCO-HPC (Infrastructure for Big data and Scientific Computing) at the University of Naples Federico II (IBISCO-HPC). In particular, the Segmenters were trained on two NVIDIA Tesla K20 GPUs, and the Classifier and the Regressor ResNets were trained in parallel each one on a single GPU. Each segmenter training lasted around 2 days, while the Classifier and Regressor were around 1 day. Subsequent training in cascade mode on the previous model output lasted around 6 hours for all combined models. The trained networks had an inference speed of 32 input cubes (64×64×128 pixels) per second using an NVIDIA Tesla K20 GPU on the IBISCO-HPC but due to overlaps in the cube patch sampling (to avoid missing sources due to boundary problems caused by accidentally splitting them within multiple patches by cutting in their emission range) and RAM limitations, inference on the full cube is made in around 30 hours. Moreover, the produced catalogues are matched together spatially and if overlapping sources are found, properties are averaged together. Classification scores are used to weight the average function after being normalized to the $[0, 1]$ range.

Acknowledgements

This work was partially produced in collaboration with members of COIN for the duration of the SKA Data Challenge 2. I want to thank all members of the COIN SKA Data Challenge 2 Team for the endless hours thinking together about the many difficulties and knowledge gaps which we had to overcome in order to find a solution to the Challenge problem. My gratitude goes to C. Heneka and B. Fraga for meeting every week for months in order to complete the challenge in the midst of the Covid pandemic and for sharing nights together coding. I want to thank The IBISCO HPC admin group for allowing us to extensively use their cluster to hold the Challenge data after the challenge completion and carry out all our experiments. I also wish to thank the European Southern Observatory (ESO) and the Dept. of electric Engineering and Information Technologies at the University Federico II for partial financial support. This work is supported by an ESO internal ALMA development study investigating interferometric image reconstruction methods and is partially sponsored by EU through the SUNDIAL ITN.

Chapter 6

The TOLIMAN Signal Detection Problem: Periodic Astrometric Signal Recovery through Deep Learning

> *In our time this search for extraterrestrial life will eventually change our laws, our religions, our philosophies, our arts, our recreations, as well as our sciences. Space, the mirror, waits for life to come look for itself there.*
>
> Ray Bradbury

6.1 The TOLIMAN Signal Detection Problem

In its simplest form, extracting the science signal arising from TOLIMAN data requires the exact registration of two overlapping point-spread functions (PSFs), one for each component of the binary star, in the image sensor plane of the orbiting space telescope. For an in-detail description of the scientific settings and characteristics of the TOLIMAN satellite see Chapter 2 Sec. 2.4. If the separation between these two stellar images can be monitored with sufficient precision, tiny perturbations due to the gravitational tug from an unseen planet can be detected. Given the configuration of the optical system, the scale of the shifts in the image plane are about one-millionth of a pixel (10^{-6} pix), thus exquisite stability is required: these motions are only manifest as a sinusoidal perturbation over year timescales. Although there are many potential sources of imperfection and error, this first study restricts itself to the most basic and fundamental one, with noise processes arising principally from photon noise and the spatial discretization of the signal. Additional terms, such as imperfect spacecraft pointing, jitter and roll stability, will be addressed in future work. For the present study, simulated and laboratory test-bed data were created to embody such error terms. During the 2^{nd} COIN-Focus: Toliman Event[1] (COIN-Focus #2) held in Rome, Italy, in November 2019, a data-challenge like event was held in which participants were asked to try and detect the astrometrical signal. A pictorial illustration of the basic challenge is shown in Figure 6.1: two patterns exist within the frame of data, in this case without the noise terms. High degrees of sharp image structure result in data for which accurate image registration is possible; however, the extreme measurement precision required to obtain the science signal moves this from a relatively routine exercise in image processing (at levels of 10^{-2} pixel) to an unsolved problem at signal fidelity levels never yet attempted within astronomy (at levels of 10^{-6} pixel). For this reason, six datasets with increasingly smaller signal amplitudes were simulated each one with an injected astrometric signal. For each dataset, the true signal period is considered as *recovered* if its residual error with the predicted period is within 10% of the true period value. Also, for each dataset, a corresponding dataset with no embedded signal is simulated. Given the unknown nature of the α Centauri AB system or of any other star system to which techniques such the ones tested in this work may be utilised, the presence or absence or the eventual number of orbiting bodies and thus astrometric signals cannot be a known parameter. For this reason, only unsupervised techniques which make no assumption about the presence or the number of embedded signals are tested and asked to make predictions on both datasets: the one containing the signal, and the one without it. For each of the six datasets, the challenge is considered surpassed if a cor-

[1] https://cosmostatistics-initiative.org/focus/toliman1/

rect detection is made on the dataset with the embedded signal, and no detection is made on the dataset containing no signal. The participants were not told which of the two dataset contained the signal.

Given that no optical test bed of the TOLIMAN satellite had been constructed at the time of this work, we had to rely on simulations. The following section briefly describes the data simulations problem and the properties of the corresponding data product.

6.2 Simulating the TOLIMAN data

The first challenge in developing a method capable of extracting a signal as small as one-millionth of a pixel is to develop a computational model capable of emulating such signal under varying conditions of noise. Although injecting a signal into an image may seem a rather trivial task, conventional approaches fall short when pushed to the limits of precision required by the TOLIMAN mission, often resulting in large computational costs. The traditional simulation approach consists of generating a super-sampled Point Spread Function (PSF). Since stars can be considered point sources, to simulate a stellar field as it would appear on the detector, we simply need to shift and downsample that PSF in the sensor grid. Thus, by assigning it to either random or specified positions within the image and repeating the procedure for many different point sources, we can recreate a stellar field. While this can be made computationally efficient today using the widespread GPU accelerators, such traditional methods, unfortunately, introduce errors orders of magnitude greater than the signal we expect to measure, thus requiring alternative approaches to the generation of the mock data.

6.2.1 Simulating the TOLIMAN PSF

Generating the TOLIMAN PSF is conceptually straightforward, requiring only the representation of the electric field at the aperture $E(x, y, \lambda) = A(x, y)e^{i\theta(x,y,\lambda)}$ as its amplitude $A(x, y)$ and phase $\theta(x, y)$, and combining these terms into a complex array. The PSF in the (u, v) focal plane is then found by taking the power of the resultant Fast Fourier Transform (FFT) of the complex array:

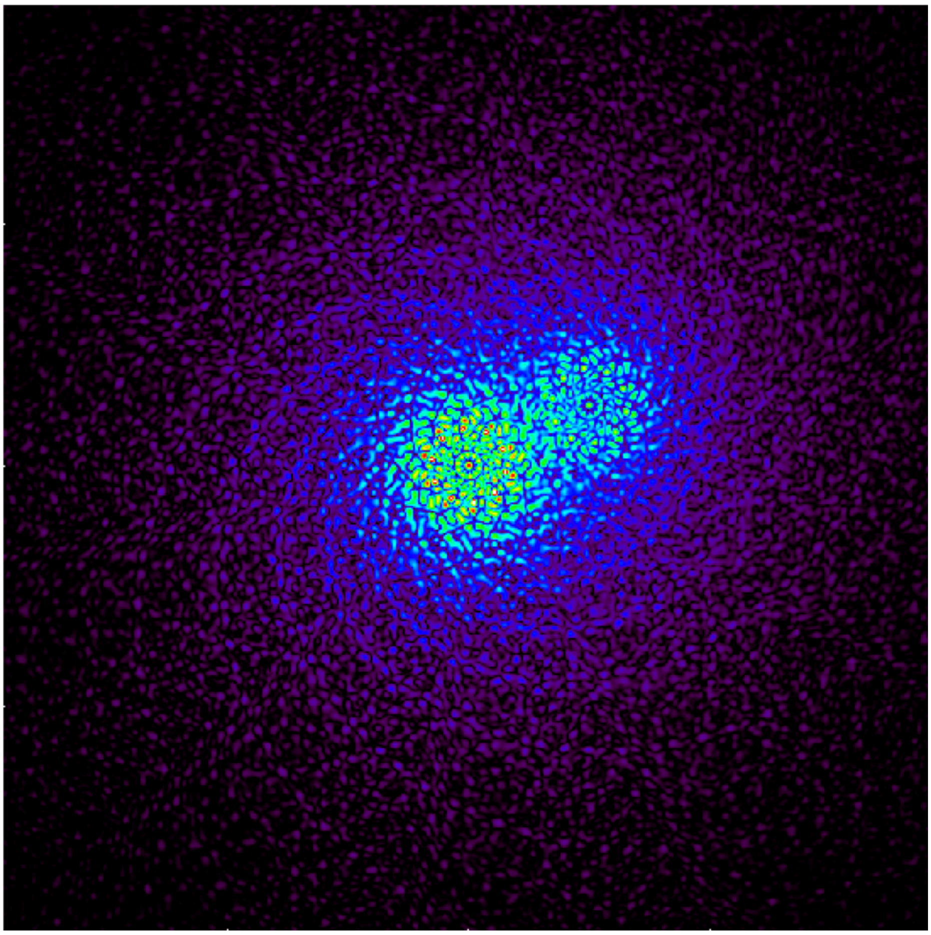

Figure 6.1. A simulated binary star as observed with the conceptual design TOLIMAN pupil discussed above.

$$PSF(u,v) = |\mathscr{F}\{E(x,y,\lambda)\}|^2 = |\mathscr{F}\{A(x,y)e^{i\theta(x,y,\lambda)}\}|^2. \tag{6.1}$$

Positional information can then be injected by applying a linear gradient to the phase θ. An Optical Path Difference (OPD) is introduced across the aperture by any source off-axis from the normal of the telescope pointing. Easily calculated through the angular offset from the normal, the OPD simply translates into phase as a function of the observation wavelength,

$$\theta_{slope}(x,y,\lambda) = \frac{2\pi}{\lambda}OPD(x,y). \tag{6.2}$$

This mathematical representation of the telescope spatial response function allows for arbitrary signals to be introduced. The TOLIMAN diffractive pupil (whose FFT is the PSF) $\theta_{Pupil}(x,y,\lambda)$ can be formulated as a mirror with 'steps' cut in which can be mathematically modelled through the following equation

$$\theta_{Pupil}(x,y,\lambda) = \frac{2\pi}{\lambda}2h(x,y) \tag{6.3}$$

where $h(x,y)$ is the height of each step in the mirror which can be translated to phase by taking the OPD as twice the height of the step. The total phase θ can be then reconstructed as a linear combination of these effects. Taking the field amplitude $A(x,y)$ as unity for all non-masked regions (steps within the mirror) of the aperture gives the full description of the electric field $E(x,y,\lambda)$. Having formulated the electric field response to the system, we must introduce a complete description of the optical architecture. This is described by a handful of parameters: aperture diameter D, effective focal length FL Focal Length (FL) and pixel size d_{pix}. Desiring computational efficiency through the inclusion of our optical system, we define some value N_{out} to be the size of the array that we pass to the Fast Fourier Transform Fast Fourier Transform (FFT). This is the primary driver behind the computational cost. Using this value and the previously described parameters, the size of the array N_E representing our electric field $E(x,y,\lambda)$ can be found. Not all arrays are taken to be of size $N \times N$. These two values necessarily differ as a way to encode optical parameters without focal plane interpolation. The ratio between N_{out} and N_E determines sampling in the focal plane matching that of our system,

$$N_E = N_{out} \frac{d_{pix} \times \lambda}{D \times \mathrm{FL}}. \tag{6.4}$$

Embedding this array representing the electric field into an N_{out} sized array, we use equation 6.1 to generate a PSF that requires no interpolation and can have positional signals of any size injected, limited only by floating-point precision. Further details and descriptions of these processes can be found in [125].

6.2.2 The data

With the tools to simulate PSF's through our optical system, we must now generate a data set. By adding basic noise processes, stellar spectra and astrometric signals we can create a comprehensive set of images that can be used to test the recovery and reconstruction abilities of signal detection techniques. Here a balance must be struck, as generating a truly comprehensive data set for the TOLIMAN mission is merely intractable. With a full signal period of order one year, any data set must present the fundamental challenges of the mission in an efficient way. Here we examine choices such as the number of wavelengths, stars and images to simulate, along with the included noise processes.

One of the first things to consider is the size of the data set, and the total number of images produced. The TOLIMAN signal is introduced to the α Cen system through the gravitational tug of an orbiting planet and so our signal is sinusoidal by nature. The orbital period that we are searching for is of order of a single year, and so producing a ‘frame by frame’ data set would be computationally intractable. Consequently, we need to generate each ‘image’ as a representation of a collection of multiple from the actual telescope. We chose to represent three full signal cycles over 1095 images, with each image representing approximately a full day. Observing in the visible spectrum over a 100 nm bandwidth, the choice of spectral resolution is essential. The wavelength dependence of the PSF demands that the image at each wavelength be computed individually. To represent the real world as closely as possible, a spectral resolution of 1 nm was chosen for several reasons:

1. the Toliman PSF is spread over many diffraction limits (10λ/D), so

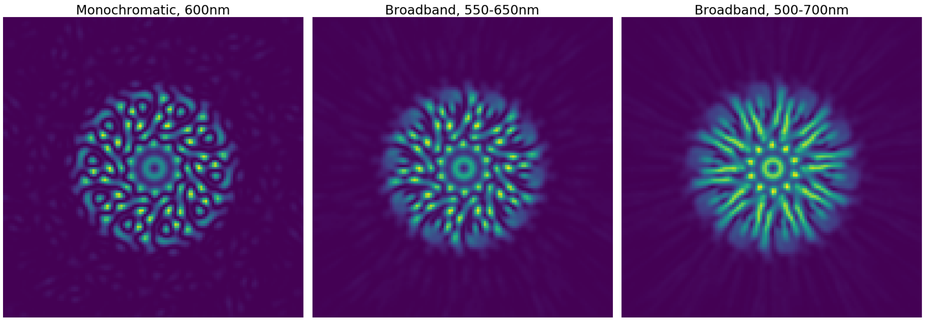

Figure 6.2. TOLIMAN PSF at different bandwidths. Left: Monochromatic 600nm. Centre: 550-600nm (best resembles actual mission). Right: 500-700nm.

at the outer reaches bandwidth smearing begins to have a substantial effect on the PSF shape (for an example see figure 6.2); 2);

2. by choosing to maintain the stellar alignment on the detector constant and keeping one of the stars stationary, we can massively reduce the number of PSFs we must compute. A stationary star only requires the calculation of a single broadband PSF. For the moving star, since the TOLIMAN signal is sinusoidal by nature and the stellar alignment is kept stationary, we only need to calculate the PSFs for a single signal cycle.

The result is a large overhead for small simulations, but with the benefit of being able to produce large and accurate simulations efficiently.

Given our chosen spectral resolution, we can use one of the many libraries available to generate spectra that reflect the true stellar parameters for each star. These libraries access existing stellar databases and recreate synthetic spectra for a host of variable stellar parameters such an effective temperature, metallicity and observational flux. We used *Pysynphot* [144] to generate stellar spectra and fluxes for our system. This package uses models built from HST observations across the HR diagram to simulate atmospheric emissions from different stars. Taking the relative fluxes and total photon counts output from this system, we can scale each monochromatic PSF by its relative power to recreate accurate PSFs. While real data will feature many varied noise processes, here we only consider two noise sources: photon and detector noise. These are dictated by Poisson and Gaussian statistics respectively. Detector noise is primarily driven by random thermal fluctuations of the discrete electrons that carry the signal through the detector. With available modern low-noise sensors, this noise is not expected to limit the extraction of the signal since it averages out to some constant value over many frames. The addition of even modest levels of this noise also serves a separate motivation: to allow for a smoother error space. This helps numerical algorithms converge faster as fine structures in the gradients are rounded and the algorithms can follow a smooth descent to the optimum. On the other hand, photon noise is an essential process that must be examined. Arising from the discrete nature of photons, this noise is simulated at each pixel by drawing from the Poisson distribution whose mean is dictated by the PSF. When performing image registration of small signals such as those anticipated in the TOLIMAN mission, the total number of photons that arrive in each image becomes an important factor. As shown in [55], there is a fundamental relationship between the number of photons received and the positional information carried by those photons. With insufficient photons, signals can not be extracted. Simulations proceeded with the production of a comprehensive batch of noisy image data sets, with sinusoidal signals in the separation of the binary star injected with decreasing amplitude to mimic increasingly more challenging planets, up to the limiting deflection of one-millionth of a pixel. These simulations closely resemble the expected response of the TOLIMAN optical system to the observation of the α Cen system. In particular six datasets were generated, respectively, with signals with amplitudes of 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6} the pixel size, constant flux PSFs (each of the two binaries' PSF presenting the same flux over time), an image peak value of 10^{9} photons and photon noise arising from the Poisson statistics. Due to the absence of any realistic noise components (jitter, rotations, aberrations, etc.), each image was cropped with a 256×256 pixels window centred around the image barycenter. This preprocessing was needed in order to eliminate any spurious shift in the image pixels which would result in adjoin noise components with amplitudes several orders of magnitudes higher than any of the injected signals. To summarise, the TOLIMAN signal registration problem consists in detecting signals with increasingly low levels of amplitude to capture the variation in separation between two overlapping PSFs within a noisy unevenly sampled time series

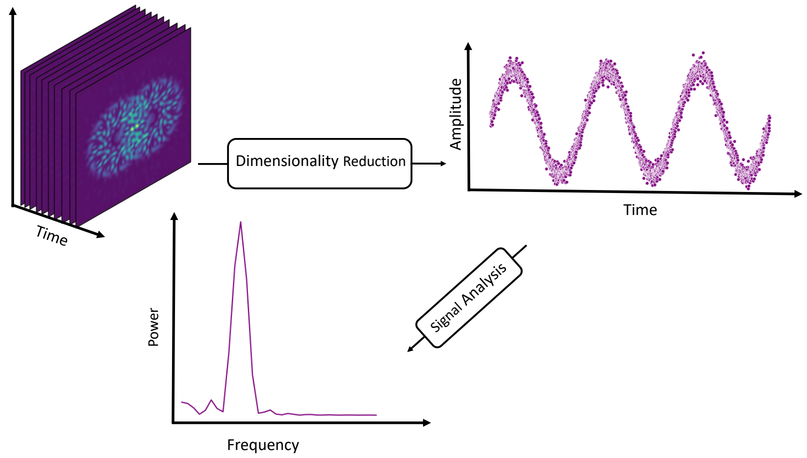

Figure 6.3. Concept workflow. The underlying concept of the proposed data analysis is based on finding a lower-dimensional representation (a compression) of the raw data which preserves periodic signals. Once a suitable representation is found, the effect caused by the presence of the planet can be detected using a time series analysis.

of images.

6.3 The Signal Detection Pipeline

The proposed pipeline developed for the detection of the astrometrical signal is a two-step process: a) a Convolutional Autoencoder (CAE) is used to compress the image sequence into a mono-dimensional latent space; and then b) the latent space is analysed through a Lomb-Scargle periodogram [96, 132, 156] in order to detect periodic components within the latent space and recover the period. An overview of the workflow of the pipeline can be seen in Fig. 6.3.

6.3.1 CAE Architecture

The architecture of the CAE implemented in this work, as shown in Fig. 6.4, is composed by five convolutional and five deconvolutional blocks. Each convolutional block is composed by a 2D Convolution layer with a kernel size of 3 and a stride of 1 followed by an ELU activation function,

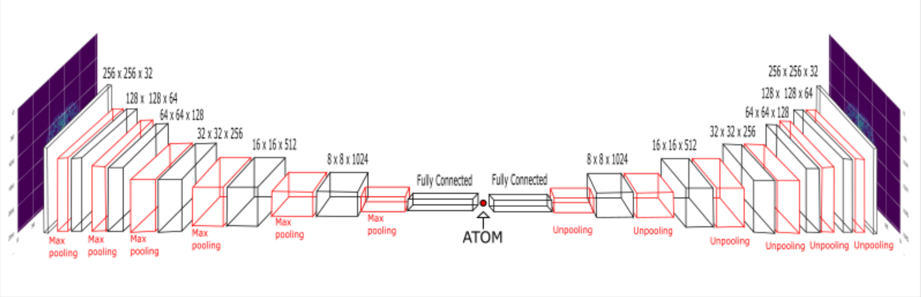

Figure 6.4. Architecture of the Deep Convolutional Auto-Encoder

a 2D Max Pool layer with a kernel size of 3 and a stride of 2, and a 2D Batch Normalization layer. Each deconvolutional block is composed by a 2D Transposed-Convolution layer witha a kernel size of 3 and a stride of 1 followed by an ELU activation function, a 2D Unpooling layer and a Batch Normalization layer. The Network latent space was chosen to be uni-dimensional (represented by the single ATOM in Figure 6.4), for the following reasons: (i) when a higher dimensional latent space was used, the Pearson correlation coefficient between the latent variables was found to be compatible with a value of 1.0; (ii) given that the separation of the star's PSFs is radial, and, given that the only varying feature in the images is the signal, it seems reasonable to think that the only information the network needs to recover from the latent space in order to decode, and thus reproduce, the images is the shift in phase. The remaining constant information (pixel luminosity and image geometry) can be stored in the network weights.

6.3.2 The Lomb-Scargle Periodogram

The most commonly used tool for period components searching in irregular time series is the Lomb-Scargle periodogram Lomb-Scargle Periodogram (LSP) [96, 132]. It is a generalisation of the Schuster periodogram in Fourier analysis,

$$P_s(f) = \frac{1}{N} \left\| \sum_{n=1}^{N} g_n e^{-2\pi i f t_n} \right\|^2 , \tag{6.5}$$

but for irregular cadences. Apart from the normalization factor, Eq. 6.5 is the power spectrum computed for a continuous signal observed with uniform sampling defined by Dirac deltas and it follows that it should encode all the relevant frequency information within the data. In fact when the classical periodogram is applied to uniformly-sapled Gaussian noise, the values of the resulting periodogram show a chi-squared dstribution. This property, which in practice is used to distinguish between periodic and non-periodic components no longer holds in case of non-uniform sampling. To address this Scargle [132] considered a generalized form of the periodogram:

$$P(f) = \frac{A^2}{2}\left(\sum_n g_n cos(2\pi f[t_n - \tau])\right)^2 + \frac{B^2}{2}\left(\sum_n g_n sin(2\pi f[t_n - \tau])\right)^2 \tag{6.6}$$

where A, B and τ are functions of the frequency f and sampling times t_i and showed that by choosing A, B and τ as follows:

$$P_{LS}(f) = \frac{1}{2}\left\{\frac{\left\|\sum_{n=1}^{N} g_n \cos(2\pi f[t_n - \tau])\right\|^2}{\sum_{n=1}^{N} \cos^2(2\pi f[t_n - \tau])} + \frac{\left\|\sum_{n=1}^{N} g_n \sin(2\pi f[t_n - \tau])\right\|^2}{\sum_{n=1}^{N} \sin^2(2\pi f[t_n - \tau])}\right\}, \tag{6.7}$$

$$\tau = \frac{1}{4\pi f}\tan^{-1}\left(\frac{\sum_n \sin(4\pi f t_n)}{\sum_n \cos(4\pi f t_n)}\right). \tag{6.8}$$

the periodogram behaves as the classical one in case of uniformly sampled observations, is analytically computable and is insensitive to global time-shifts in the data. This modified periodogram differs from the classical periodogram only to the extent that the denominators $\sum_n \sin^2(2\pi f t_n)$ and $\sum_n \cos^2(2\pi f t_n)$ differ by $N/2$, which is the expected value of each of these quantities in the limit of complete phase sampling at each frequency. The results obtained through the Lomb-Scargle periodogram are identical to those obtained by fitting a sinusoidal model to the data at each frequency and by constructing a periodogram from the χ^2 goodness of fit at each frequency. If the problem is reformulated in this way, τ can be trough as a factor to orthogonalize the normal equations used in the least squares analysis.

6.3.3 Atom Time Series Analysis

To find periodic trends within the trained CAE latent space, we use the Lomb-Scargle periodogram and, to validate the goodness of the period estimation, we employ the following metrics:

- False Alarm Probability False Alarm Probability (FAP): encodes the probability of measuring a peak of a given height (or higher) conditioned on the assumption that the data consists of Gaussian noise with no periodic component;

- Full Width at Half Maximum FWHM: this expresses the extent of a function produced by the difference between the two extreme values of the independent variable at which the dependent variable is equal to half of its maximum value. Treating the FWHM as an error measure, we derive an error on the period through the following expression:

$$P = \frac{1}{f(\text{peak})}, \tag{6.9}$$

$$\Delta P = \frac{1}{f(\text{peak})^2} \times \left[f\left(\text{peak} + \frac{FWHM}{2}\right) - f\left(\text{peak}\frac{FWHM}{2}\right)\right], \tag{6.10}$$

where peak stands for the peak of the power spectrum and $f(\text{peak})$ its relative frequency.

6.3.4 Training Strategies

Each of the six simulated datasets contained 1095 images simulating three years of observations and three expected full planet orbits and a corresponding .csv file containing the mono-dimensional noiseless target astrometric signal. Each dataset was split into Train and Validation respectively containing 985 and 110 images. The Train and Validation sets are created by cutting time-contiguous chunks of the image time series. This is done in order to ensure that the Validation set contains an ordered sequence of images from which the embedded astrometric signal could be recovered. The CAE is trained on pairs of sampled Training set images and as a loss, we use a weighted combination of the l_1 loss and the $DSSIM$

loss (see Chapter 4 Sec. 4.2.1 for the mathematical details of the loss functions).

$$l(x, y, t) = a(t) \times l_1(x, y) + b(t) \times DSSIM(x, y) \tag{6.11}$$

where x and y are the prediction and target variables and t is the epoch counter. The weighting is performed with two variables a and b, which depend on the epoch counter t. The training begins with $a(0) = 1$ and $b(0) = 0$ in order to first allow the model to learn a *median* representation of the data (which should contain information about the TOLIMAN PSF and noise patterns, assumed roughly constant in the data) through the minimization of the l_1 loss. At each epoch, a is decreased by δ and b is increased by the amount in order to slowly transition to the $DSSIM$ loss. In the last epochs, only the $DSSIM$ loss is effectively used in order to learn the nuances in the data, such as the contours of the fringe patterns. The model is trained with an Adam Optimiser [81] and, in order not to overfit the data in the first training iterations, we adopt a *warm up* strategy for the learning rate [52] in which we start with a learning rate of 0, and we uniformly increase it to 1×10^{-4} in the first 100 iterations. The model is trained for $10,000$ epochs, but we also employ an early stopping criterion based on the validation loss. If no improvement of validation loss with respect to the moving average of the last 100 validation losses is registered for 10 consecutive steps, then training is halted.

6.4 Experiments

On each of the six couples of datasets, we run and compare our proposed DL-based pipeline, with Wasserstein dictionary learning [134] followed by LSP and PCA followed by LSP as a baseline. For details about the two techniques see Chapter. 2, Sec. 2.3.3. The two methodologies are utilised as follows:

1. Principal Component Analysis (PCA) + LSP (Baseline Method): each image is flattened resulting in an array of 256×256 values which is then compressed through PCA to a single scalar. The resulting monodimensional time series is analyzed with LSP to search for periodic patterns;

2. Wasserstein dictionary learning (WDL) + LSP: the algorithm is

Dataset	PCA + LSP	WDL + LSP	CAE + LSP
10^{-2} **pixels**	✓	✓	✓
10^{-3} **pixels**	×	✓	✓
10^{-4} **pixels**	×	✓	✓
10^{-5} **pixels**	×	×	✓
10^{-6} **pixels**	×	×	✓

Table 6.1. Hit or miss table showing whether the three compared methods are able to solve the signal detection problems. The five problems are ranked by the amplitude in pixels of the embedded astrometric signal. A problem is deemed as solved (✓) if the period of the astrometric signal is detected within 10% of its true value in the dataset containing the signal and no spurious signal is detected in the corresponding empty data. A failure may come from failing one of the two conditions or both of them.

trained on each full dataset using the Kullback-Leibler divergence [99] between the input images and those reconstructed from the dictionary.

$$L(y_{pred}, y_{true}) = y_{true} \cdot log\frac{y_{true}}{y_{pred}} = y_{true} \cdot (log y_{true} - log y_{pred}) \quad (6.12)$$

where y_{pred} and y_{true} are pixels from respectively the prediction and input image. The Kullback-Leibler loss is obtained by taking the mean over all image pixels. The method depends on three parameters: the number of iterations L, the γ coefficients for the choice of the Optimal Transport distance, and the number of atoms within each dictionary S. L was set to 1000, γ to 0 in order to use the Wasserstein distance, while several choices of S were tested and the best results have been obtained with $S = 3$.

To measure the CAE image reconstruction capabilities we utilise two metrics: the mean structural similarity index (SSIM) and the mean residual score (MRS) (for details on the mSSIM and MRS see respectively Chapter 4 Sec. 4.2.1 and Sec. 4.4).

Table 6.1 shows, for each dataset (row of the table) which models were able to solve the signal detection problem. A problem is deemed as solved if the period of the astrometric signal is detected within 10% of its true

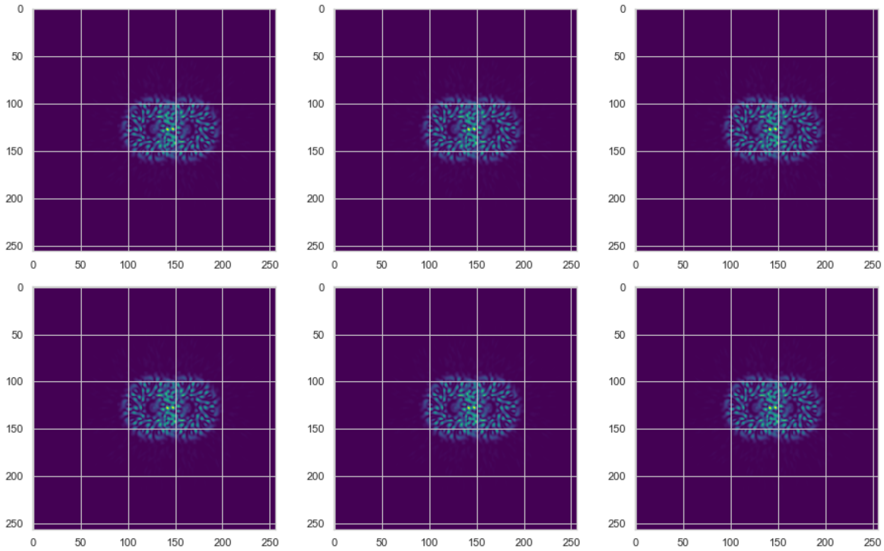

Figure 6.5. Example of CAE image reconstructions on the Validation set. The first row contains a random subset of Validation TOLIMAN images, the second row shows their respective reconstructions produced by the trained CAE model.

Time Series	**Period**	**FAP**
Target Noiseless Signal	0.33 ± 0.05	0
CAE ATOM Time Series	0.33 ± 0.06	7.2×10^{-68}

Table 6.2. Period and FAP obtained by analysing the LSP of the noiseless 10^{-6} pixels size (True Signal) and the CAE mono-dimensional latent-space time series.

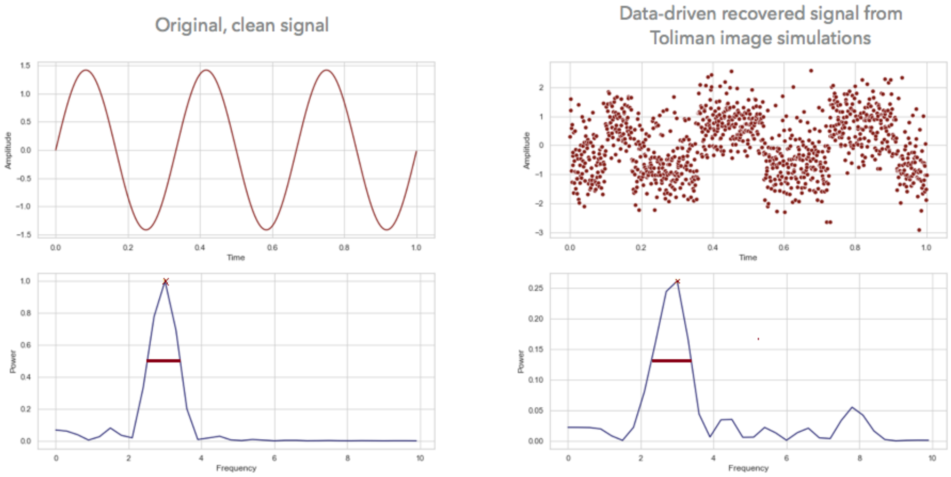

Figure 6.6. In the left panels, a perfect signal is represented at the top and the relative Lomb Scargle Periodogram obtained from its analysis is represented at the bottom. On the right, a time series from the atoms obtained with the deep convolutional auto-encoder applied to TOLIMAN simulation with a 10^{-6}-level astrometric shift is shown at the top, and its Lomb Scargle Periodogram is represented at the bottom. The power peaks and their relative FWHM are shown in red over the power spectrum.

value in the time series containing the embedded astrometric signal and no spurious signal is detected in the one containing no signal. As it can be seen, PCA is able to encode the periodic trend only for the 10^{-2} signal, while WDL is able to embed the periodic trend up to 10^{-4}. Our CAE architecture is able to embed the periodic trend up to 10^{-6}. We are thus focusing on the detection results of this latter, harder, detection problem. The final MRS and mSSIM on the Validation set are found to be respectively 4.4×10^{-8} and 2×10^{-6} and thus the images are reconstructed with a good enough precision (with respect to the accuracy needed) to recover the signal. Although the image reconstruction capability of the network directly correlates with these losses (as one should expect), we do not find any direct correlation with the signal reconstruction capabilities. Although after 1000 epochs the l_1 Loss gradient flattened, the DSSIM Loss keeps decreasing up to 5000 epochs while the latent space began showing an increasing sinusoidal trend in time. To have a loss function that correlates with the signal reconstruction in the latent space, we would need an architecture that makes use of the time dimension of the images: something not anticipated at the time of the publication of this work. For this reason, the network encodes only the detection of the signal and not its amplitude. The CAE latent space did not show any periodic trend for all the simulations with no signal injected. Table 6.2 shows the detected period and its relative error derived through Eq.6.10 and its False Alarm Probability obtaining analysing respectively the target noiseless signal and the CAE mono-dimensional Atom time series. The relative Lomb-Scargle Periodogram, are shown in Fig. 6.6. In yellow we highlighted the power peak and the FWHM of the power spectrum around that peak.

6.4.1 Discussion of Results

Sec. 6.4 describes both the CAE's reconstruction capabilities and the analysis on the atom time series to find its periodicity. We have shown that the recovered periodicity is compatible with the injected astrometric signal period and thus that the architecture is able to recover the signal directly from the TOLIMAN simulation images. One of the main current issues is the lack of correlation between the Network reconstruction of the TOLIMAN images and the presence of a periodic trend in the atom time series. Since the Network is only training with spatial information, in reality, there is no encoded reason why the latent space should present a sinusoidal trend with time. The only thing that the latent space should be encoding is how to reconstruct the images. That being said, given our knowledge of the sinusoidal nature of the astrometric signal, and since the astrometric shift is the only element changing through the images in time apart from uncorrelated Gaussian noise, we do not see any reason why the latent space should not present a sinusoidal trend with time, regardless of the fact that we did not apply any constraint (at the architectural level) to enforce it. As seen in Fig. 6.6, in fact, given enough epochs, the time series actually shows a sinusoidal behaviour.

A necessary step forward in this work is to produce simulations with increasing noise realism and complexity, in order to evaluate whether Deep Learning can still be used to recover the astrometric signal. It must be expected that this simple approach would fail to recover the signal if noise patterns which show correlations in time are simulated.

6.5 Conclusions

In this work, we have shown how Deep Learning, in particular Deep Convolutional Autoencoders, can be used to extract, in a completely unsupervised way, periodic astrometric signals with amplitudes of the order of 10^{-6} with respect to the size of a pixel in simplistic noise conditions. This is the magnitude of the signals that would be produced by an Earth-like planet at the habitable zone of a star in the Alpha Centauri binary system (see Chapter. 2 Sec. 2.4).

We presented a detailed explanation of the adopted network architecture (see Sec. 6.3) and of the simulations used, which were created using FFT techniques (see Sec. 6.2). Although the present simulations do not yet contain some realistic systematic noise components, such as telescope jitter, rotations and aberrations, they pose a significant challenge to classical unsupervised techniques, due to the small amplitude of the signal with respect to the pixel size. We have shown that, from the obtained CAE latent space, we can obtain a time-trend that can be analysed for periodicity, using any time-domain signal extraction technique. Here we used a standard Lomb-Scargle technique and were able to find a period consistent with that of the injected signal (see Sec. 6.4). Finally, we note that in this work we only explored a fully unsupervised method for the compression, although semi-supervised and hybrid methods can be a natural extension, by considering that we may constrain the problem's dimensionality – for instance, a first-order approximation of the shape of the PSF. A further step will be the generation of increasingly realistic systematic noise contributions, to design network architectures that can handle them and still allow for the detection of the planetary signal. This work opens an exciting path that we believe should be further studied, towards the extraction of periodic signals of binary systems at the milliarcsecond level, directly from times series of satellite imaging data.

Acknowledgements

The scientific results reached in this work were partially produced during the 2^{nd} COIN-Focus: Toliman Event[2] (COIN-Focus #2) held in Rome, Italy, in November 2019. The COIN-Focus: Toliman participants acknowledge the fundamental support of the Breakthrough Initiatives. The Breakthrough Watch initiative and committee (notably Olivier Guyon, Pete Klupar & Pete Worden) have supported and framed the problem. We also acknowledge input and ideas from people in the wider TOLIMAN collaboration including Ben Pope, Barnaby Norris, Bryn Jeffries, Anthony Horton and others. MB acknowledges financial contributions from the agreement *ASI/INAF 2018-23-HH.0, Euclid ESA mission - Phase D* and the *INAF PRIN-SKA 2017 program 1.05.01.88.04*. EEOI acknowledges financial support from CNRS 2017 MOMENTUM grant under project *Active Learning for Large Scale Sky Surveys*. AKM acknowledges the support from the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through grants SFRH/BPD/74697/2010, PTDC/FIS-AST/31546/2017 and from the Portuguese Strategic Programme UID/FIS/00099/2013 for CENTRA.

[2] https://cosmostatistics-initiative.org/focus/toliman1/

Chapter 7

Conclusions and Future Prospects

So Long, and Thanks for All the Fish.

Douglas Adams

In this Thesis, I have developed a set of Deep Learning based pipelines to solve several ill-conditioned inverse problems within the fields of Radio Interferometry and Narrow-Angle Astrometry. The purpose of this conclusive Chapter is to showcase the main scientific conclusions obtained addressing the three problems tackled in the Thesis and outline future prospects and directions. The chapter is divided into three sections, one for each problem, in which a short summary of the relative Chapter is given, before outlining future prospects.

7.1 Detection and Characterisation of ALMA Sources through Deep Learning

The planned upgrades to correlators and broadening of the receiver's bandwidth will bring ALMA in the Terabyte regime. As a result ALMA cube imaging will become a very demanding task necessitating novel automated pipelines capable of tackling the interferometric deconvolution problem and performing source detection and characterization with matching speeds and with the performances required to reliably use the obtained processed data products (such as sources morphological parameters and properties) for scientific purposes. To address this, in collaboration with several ESO researchers, I developed a Deep Learning source detection and characterization pipeline which, by working directly on dirty interferometric calibrated data cubes which have not undergone any prior de-convolution, is able to combine spatial and frequency information to reliably detect and characterize embedded Astronomical sources. The performed experiments and the developed pipeline are optimized to address two separate problems:

1. can Deep Learning be used to solve the ALMA image deconvolution problem in order to speed up and improve on the performance of tCLEAN [70]?

2. Can Deep Learning be used to reliably detect and characterise sources in the prevision of its use to search for serendipitous high-redshift sources within targeted ALMA observation or in a blind-search fashion on the full ALMA database?

To train and test the capabilities of the proposed pipeline, we needed thousands of ALMA models and dirty cube pairs and, for that reason, we created our own simple but realistic simulations of ALMA observations by combining *Python* and bash scripting with the Common Astronomy Software Application (CASA) v. 6.5.0.15 [105] Python libraries. The need to use simulated data instead of real ALMA observations arises from several practical necessities: a) in order to evaluate the pipeline's performance and reliability and its dependence on sources' observational and morphological parameters, we need full control over the data properties; b) in order to assess the reconstruction quality of our pipeline and its ability to solve the deconvolution problem without relying on CLEAN deconvolution, we need noiseless sky observations which are unattainable for real observations. Given that the foreseen scientific application of the pipeline is the search of serendipitous sources within high redshift ($z > 3$) ALMA observations, we simulate, in each cube, a central primary source with a given SNR, surrounded by less bright serendipitous sources that can occupy any position in the cube. The simulation code is made available through

GitHub allows the community to generate thousands of ALMA data cubes in parallel. Regarding the architecture of the proposed architecture, it is composed of six deep learning models interconnected through logical operations: Blobs Finder, a Convolutional Autoencoder, detects sources within the frequency-integrated data cubes, Deep GRU a Deep Recurrent Neural Network, exploits the frequency domain and detects emission peaks in the spectra extracted from sources detected by Blobs Finder, and the ResNets regress the source parameters from 'spectrally focused' images created by cropping spatially around the sources, and integrating within their emission range found by Deep GRU, and the line emission images created by masking the cube with the 3D emission models found by combining Blobs Finder and Deep GRU predicted emission ranges (in the spatial and frequency planes of the cube, respectively). $5,000$ simulated cube pairs were generated containing $22,532$ simulated sources and randomly divided into a train, validation, and test sets using the rather usual 60%, 20%, 20% splitting criterion. The proposed pipeline successfully detected 92.3% of the simulated test set sources with no false positives. While the achieved performances are promising for the prospect of applying the pipeline to real data, the low FPRs of the DL models and the subsequent removal of the remaining FPs through SNR and geometrical criteria (FP detection and source deblending step) are closely related to the simplified assumptions that were made to generate the mock data. Integration times and antenna configurations are kept constant resulting in a very low dirty beam variation across the data. Only single peak spectra are simulated without taking into consideration the effects of inclination with the observer's line of sight or velocity dispersion within the galaxy, resulting in simplified spectral profiles. Quality assessment is performed by comparing our results against three other methods, BLOBCAT, SOFIA-2 and DECORAS. The first two are widely known traditional methods used for source detection and characterisation in Radio Astronomy, while the latter is another DL pipeline constructed to solve a similar problem for another Radio Interferometer, the Very Long Baseline Array (VLBA). We notice a substantial improvement in both precision and recall with respect to the first two methods and a smaller improvement with respect to DECORAS in both the number of the detected sources and the quality of the reconstructed true sky models. Blobs Finders' and DECORAS' low numbers of FPs with respect to SOFIA-2 and BLOBCAT are explainable due to the DL models capabilities of approximating the dirty beam. While Blobs Finder improved performances over DECORAS are due to architectural choices and training strategies, we believe that if dirty beam variations are increased by simulating multiple antenna configurations and observing conditions (integration time or azimuth), the number of FPs detected by both models will probably increase due to the increased complexity required to approximate a more realistic variating PSF. Regarding the source characterisation performances, source positions are found with subpixel errors in spatial and frequency domains, while projection angles and flux densities estimations show a relative error within the standard amplitude calibration error of interferometric data ($\simeq$ 10% [36]) for respectively 73% and 80% of all sources. A direct comparison with TCLEAN is performed on the radio-interferometric deconvolution task over the 1000 simulated test set cubes and shows that Blobs Finder is able to outperform TCLEAN in both speed and sky model reconstruction performance. This result was only obtained on simulated cubes containing simple point-like sources and further investigations will be required on cubes containing sources with extended emissions and complex morphologies and on real data before definitive conclusions can be drown. Several interesting scientific questions rise up from the results obtained so far on the solution of the ALMA image reconstruction and source detection and characterisation problems through Deep Learning: can we maintain the obtained performances on the detection and characterisation problem in case of more complex and realistic beam variation and source morphology? Can we outperform TCLEAN in case of extended emission? In future work, we plan to answer both questions. In fact, we aim to improve our simulation code to include more complex galaxy morphologies and extended emission simulation which requires the use of physically-based models for the galaxy kinematics and employ spectral catalogues to generate several spectral profiles for different classes of sources with the primary goal of improving the quality of our simulations, and the additional goal of having a publicly available and easy to use simulation code that the community may use to generate common data sets on which compare different architectures. Alongside we want to modify our pipeline to account for much more complex spectral profiles. The DL pipeline is currently tailored to detect single-peak emission lines, assuming that celestial sources have a single positively defined emission peak in their spectra. The above does not hold for absorption line detection or for sources with multi-peak spectral profiles. Both the peak detection (on Deep GRUs denoised spectra) and FP detection and removal (on spectrally focused images through SNR criteria) algorithms will thus need to be modified when dealing with real observations. For example, the latter could be substituted by a similar ResNet Classifier to that that we have employed in the Revised SKA pipeline. We are also planning to make an assessment on faint signal detection, especially in the presence of strong sidelobes in the cubes and asses the pipeline capabilities in the case of data containing several uv coverages and array configurations, which should result in a greater variation of the dirty beam within the data, thus posing a more complex image reconstruction problem. Furthermore, we also plan to make tests about incorporating the dirty beam within our pipeline in order to improve the image reconstruction capabilities. Regarding the solution of the ALMA deconvolution problem, given the obtained performances on simplified simulations with Deep Learning and the improvements obtained on both speed and reconstruction quality with respect to tCLEAN, we think that this should be explored in greater detail through the use of the planned improved simulations and real data. Regarding the foreseen scientific application of the developed pipeline, i.e. the detection of high-redshift galaxies, we are already carrying out the necessary operations of data retrieval from the archive in search of high-redshift observations which may contain star-forming galaxies. In fact, among the many key drivers of galaxy evolution (AGNs feedback, mergers, clumps), the Star Formation Rate (SFR) is one of the key points necessary to achieve a complete galaxy evolution theory. While up to $z \simeq 3$, we have a robust understanding of the star formation history thanks to several multiwavelength surveys, constraints at $z > 3$ are mostly based on the ultraviolet rest-frame, which is very sensitive to dust reddening. In fact, when young stars and their surrounding HII regions coexist with dust, a small amount of their short-wavelength emission is absorbed by the dust and re-emitted in the far infrared (FIR) and, at high redshifts, these wavelengths are observed in the sub-millimetre regime [68]. For such reason, as well as for their faintness, the demographic of dusty galaxies at $z > 3$ is still very uncertain, thus undermining our understanding of the initial phases of galaxy evolution. ALMA offers the possibility to tackle this problem and in fact, the frequency range covered by the ALMA bands probes the thermal dust spectrum of high-redshift galaxies, covering it from the long-wavelength Rayleigh-Jeans tail to the Spectral Energy Distribution (SED) peak features [68]. The logical answer to fill the described observational bias at high redshift is to perform dedicated surveys with ALMA. However, in this case, two main problems arise: to resolve such high redshift galaxies long exposure times are needed (2 - 10 hrs); and, given the small field of view of ALMA, scans contain a single scientific target. These limitations imply that any high redshift survey would require a serious commitment in telescope time. Still, the 'Origins of Galaxies' is one of the three fundamental science drivers for ALMA in the next decades, and the community has devoted and will devote telescope time to perform targeted observations of high redshift galaxies (Large Programs such ASPECS, ALPINE, ALCS, REBELS, CRISTAL, ASPIRE). To increase the number of detectable galaxies, the community has also searched for serendipitous sources within known high redshift ALMA observations and performed blind detection on the available data [95, 48]. To my knowledge, all these source detection studies employ similar methodologies for data cleaning and source detection: CASA (tCLEAN) for data cleaning and S/N threshold-based algorithms that utilise convolving kernels to search for statistically meaningful correlations within the data (findclumps, clumpfind), for source detection. In future work, I plan to optimise and employ the developed pipeline to search in a blind search fashion for serendipitous high redshift galaxies within data collected in the above-mentioned ALMA Large Programs and in other high redshift targets already within the ESO Science Archive Facility. For that reason, I'm beginning the retrieval of targeted high redshift Quasi Stellar Objects (QSO) observations with known serendipitous sources in order to have the first test on real ALMA data.

7.2 The SKA Data Challenge 2: Detection and Characterisation of SKA Sources through Deep Learning

The Square Kilometre Array (SKA) first observations will mark a paradigm shift not only in our knowledge of the primordial universe thanks to its sensibility but also in how scientific investigations are carried on to gain knowledge from such observations. In order to perform such sensitive observations and extract meaningful scientific information from them, an unprecedented amount of data will need to be captured, transported, processed, stored, shared and analysed. This will require extensive innovations in the way these tasks are traditionally carried out. With this goal in mind, the SKA Observatory (SKAO) has set up several data challenges, the latter of which, the SKA Data Challenge 2 consisted of the detection and characterisation of astronomical sources embedded in a 931 GBs data cube ($5851 \times 5851 \times 6668$ pixels) simulating an HI imaging datacube representative of future SKA MID spectral line observations. In the Chapter, we showcase our two attempts at solving the Challenge, two DL-based pipelines. The first one was devised in collaboration with other Cosmostatistic Initiative (COIN) members and directly participated in the Challenge (COIN pipeline), while the second one was developed by me and built upon the previous one in order to solve its main shortcomings and issues (Revised pipeline): a high number of false positive detections, poor segmentation performances, and several bugs within the data preprocessing. The two pipelines share the same working principles and some common elements. Both pipelines receive as input a 3D $64 \times 64 \times 128$ patch of the input cube and process it in order to predict a corresponding segmentation map masking with 1 pixels belonging to the sources within the cube and with 0 the rest. The operation is carried out in parallel by two models: a DL model and our 3D re-implementation of the Starlet Multi-Scale vision model. The DL model is a 3D Convolutional Autoencoder in the COIN pipeline and 3D ResNet CAE in the revised pipeline. The two predicted segmentation maps are combined together (averaged) and post-processed in order to detect meaningful emissions. This is the *Crossmatch + Quality Cuts* step of the pipeline. The surviving candidates (or more precisely the pixels belonging to the candidates) are preserved in the *segmentation map* while the remaining pixels are set to 0. In order to account for the fact the Segmentation networks may have missed source boundaries which, given the nature of the simulated data and the spreading effect of the dirty beam, could be at or below the noise level, each source segmentation map is dilated by a factor t_d. The resulting processed segmentation map cube is multiplied with the input 3D patch to get the *moment masked cube* which contains all the input pixels values belonging to source candidates and the remaining pixels set to 0. For each detected source, we extract two $32 \times 32 \times 128$ pixel patches centred around the source bounding box, one from the segmentation-masked cube and one from the moment-masked cube. These are the 3D source segmentation map and the 3D source moment-masked cube. In both pipelines, the source 3D segmentation map is used to estimate the source position in the cube. In the Revised pipeline a 3D ResNet Classifier is used to flag and eliminate false detections. In the COIN Pipeline, the projection angle θ was instead estimated to simple ellipse fitting to the integrated source segmentation map. Sources moment masked cubes are then passed along to a series of specialized ResNets which regress the embedded sources morphological parameters. The performances of the two DL pipelines are compared with the other 12 DL and traditional pipelines which competed in the challenge. Our revised pipeline achieves the highest score reaching similar results to the two winning DL-based pipelines MINERVA and FORSKA-Sweden with the advantage of being able to retain performances on the brightest end of the data-integrated line flux distribution. This is achieved through the combination of the DL models with a traditional compressed sensing technique such as the Multi Vision model. In fact, nevertheless, the DL Segmenter suffers from the under-representation of bright sources within the training set, the same bright sources are easier to detect for the Multi Vision model. The success obtained by pipelines combining multiple models to detect sources within SKA Data (Revised, MINERVA, FORSKA-Sweden) seems to suggest that ensembling may be the key to further pushing the overall performance. We believe that further increases may be obtaining increasing the amount of prior information used by the pipelines. As an example, the *dirty beam* should be known fairly well for a given observation with only minor fluctuations due to atmospheric and observational uncertainties to still be accounted for. In future work, we plan to introduce the dirty beam as a first approximation of the PSF, and to reformulate the problem in a residual fashion in order to ease the problem complexity and try and push down the flux detection limit of our pipeline.

7.3 The TOLIMAN Signal Detection Problem: periodic Astrometric Signal Recovery through Deep Learning

The TOLIMAN space telescope [152] is a low-cost mission which aims to push the boundaries of astrometric measurements in binary star systems and to enable the detection of Earth-like planets around α Centauri, our closest extra-solar system and other close candidate star systems. The mission is optimised to search for habitable-zone planets that, for α Centauri, imply deflections with amplitudes of the order of $\sim 1\mu$as over roughly 1-year orbital periods. The detection of such a small astrometric signal has never been reported before in the Astronomical literature. We have investigated the use of DL for the detection of such astronomical signals, expressed as shifts in the position of two overlapping Point Spread Functions (PSF), embedded in realistic simulations of the TOLIMAN's mission: a three-year observation of the α Centauri star system. Given the configuration of the optical system, the scale of the shifts in the image plane is about one-millionth of a pixel (10^{-6} pix), thus exquisite stability is required: these motions are only manifest as a sinusoidal perturbation over year timescales. Although there are many potential sources of imperfection and error, this first study restricts itself to the most basic and fundamental one, with noise processes arising principally from photon noise and the spatial discretisation of the signal. Additional terms, such as imperfect spacecraft pointing, jitter and roll stability, will be addressed in future work. The simulated data products consisted of six datasets containing, respectively, signals with amplitudes of 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6} times the pixel size, constant flux PSFs (each of the two binaries' PSF presenting the same flux over time), an image peak value of 10^9 photons and photon noise arising from the Poisson statistics. Due to the absence of any realistic noise components (jitter, rotations, aberrations, etc.), each image can be cropped with a 256×256 pixels window centred around the image barycenter. This preprocessing is needed in order to eliminate any spurious shift in the image pixels which would result in adjoining noise components with amplitudes several orders of magnitude higher than any of the injected signals. The proposed pipeline is based on a custom Convolutional Autoencoder (CAE) tasked with compressing the image sequence into a mono-dimensional latent space followed by the analysis of the latent space through a Lomb-Scargle periodogram (LSP) [96, 132, 156] in order to detect periodic components within the latent space and recover the target period. The signal compression performances obtained by the CAE and the correlated capability of extracting the signal through the LSP are compared with Principal Component Analysis (as baseline) and Wasserstein dictionary learning (WDL) [133] showing improved detection capabilities. While PCA was able to capture the periodic trend only for the easiest of problems (10^{-2}), and WDL managed to push it up to 10^{-4}, the CAE was the only model that was able to capture a periodic trend consistent with that of the injected signal up to the required amplitude of 10^{-6} times the size of the pixel. In future work, we plan to investigate the signal detection capabilities in case of further and increasingly realistic noise contributions, and we plan to modify the architecture by employing a first estimate of the TOLIMAN PSF as a prior to constraint thus regularising the problem.

Bibliography

[1] The European Space Agency. The HIPPARCOS and TYCHO catalogues. Astrometric and photometric star catalogues derived from the ESA HIPPARCOS Space Astrometry Mission. In *ESA Special Publication*, volume 1200 of *ESA Special Publication*, January 1997.

[2] A Akhazhanov, A More, A Amini, C Hazlett, T Treu, S Birrer, A Shajib, K Liao, C Lemon, A Agnello, B Nord, M Aguena, S Allam, F Andrade-Oliveira, J Annis, D Brooks, E Buckley-Geer, D L Burke, A Carnero Rosell, M Carrasco Kind, J Carretero, A Choi, C Conselice, M Costanzi, L N da Costa, M E S Pereira, J De Vicente, S Desai, J P Dietrich, P Doel, S Everett, I Ferrero, D A Finley, B Flaugher, J Frieman, J García-Bellido, D W Gerdes, D Gruen, R A Gruendl, J Gschwend, G Gutierrez, S R Hinton, D L Hollowood, K Honscheid, D J James, A G Kim, K Kuehn, N Kuropatkin, O Lahav, M Lima, H Lin, M A G Maia, M March, F Menanteau, R Miquel, R Morgan, A Palmese, F Paz-Chinchón, A Pieres, A A Plazas Malagón, E Sanchez, V Scarpine, S Serrano, I Sevilla-Noarbe, M Smith, M Soares-Santos, E Suchyta, M E C Swanson, G Tarle, C To, T N Varga, and J Weller and. Finding quadruply imaged quasars with machine learning – i. methods. *Monthly Notices of the Royal Astronomical Society*, 513(2):2407–2421, April 2022.

[3] Kazunori Akiyama, Shiro Ikeda, Mollie Pleau, Vincent L. Fish, Fumie Tazaki, Kazuki Kuramochi, Avery E. Broderick, Jason Dexter, Monika Mościbrodzka, Michael Gowanlock, Mareki Honma, and Sheperd S. Doeleman. Superresolution full-polarimetric imaging for radio interferometry with sparse modeling. *The Astronomical Journal*, 153(4):159, mar 2017.

[4] ALMA Partnership, C. L. Brogan, L. M. Pérez, T. R. Hunter, W. R. F. Dent, A. S. Hales, R. E. Hills, S. Corder, E. B. Fomalont, C. Vlahakis, Y. Asaki, D. Barkats, A. Hirota, J. A. Hodge, C. M. V. Impellizzeri,

R. Kneissl, E. Liuzzo, R. Lucas, N. Marcelino, S. Matsushita, K. Nakanishi, N. Phillips, A. M. S. Richards, I. Toledo, R. Aladro, D. Broguiere, J. R. Cortes, P. C. Cortes, D. Espada, F. Galarza, D. Garcia-Appadoo, L. Guzman-Ramirez, E. M. Humphreys, T. Jung, S. Kameno, R. A. Laing, S. Leon, G. Marconi, A. Mignano, B. Nikolic, L.-A. Nyman, M. Radiszcz, A. Remijan, J. A. Rodón, T. Sawada, S. Takahashi, R. P. J. Tilanus, B. Vila Vilaro, L. C. Watson, T. Wiklind, E. Akiyama, E. Chapillon, I. de Gregorio-Monsalvo, J. Di Francesco, F. Gueth, A. Kawamura, C.-F. Lee, Q. Nguyen Luong, J. Mangum, V. Pietu, P. Sanhueza, K. Saigo, S. Takakuwa, C. Ubach, T. van Kempen, A. Wootten, A. Castro-Carrizo, H. Francke, J. Gallardo, J. Garcia, S. Gonzalez, T. Hill, T. Kaminski, Y. Kurono, H.-Y. Liu, C. Lopez, F. Morales, K. Plarre, G. Schieven, L. Testi, L. Videla, E. Villard, P. Andreani, J. E. Hibbard, and K. Tatematsu. THE 2014 ALMA LONG BASELINE CAMPAIGN: FIRST RESULTS FROM HIGH ANGULAR RESOLUTION OBSERVATIONS TOWARD THE HL TAU REGION. *The Astrophysics Journal*, 808(1):L3, jul 2015.

[5] Justin Jonas and. The MeerKAT radio telescope. In *Proceedings of MeerKAT Science: On the Pathway to the SKA — PoS(MeerKAT2016)*. Sissa Medialab, February 2018.

[6] and A. G. A. Brown, A. Vallenari, T. Prusti, J. H. J. de Bruijne, C. Babusiaux, C. A. L. Bailer-Jones, M. Biermann, D. W. Evans, L. Eyer, F. Jansen, C. Jordi, S. A. Klioner, U. Lammers, L. Lindegren, X. Luri, F. Mignard, C. Panem, D. Pourbaix, S. Randich, P. Sartoretti, H. I. Siddiqui, C. Soubiran, F. van Leeuwen, N. A. Walton, F. Arenou, U. Bastian, M. Cropper, R. Drimmel, D. Katz, M. G. Lattanzi, J. Bakker, C. Cacciari, J. Castañeda, L. Chaoul, N. Cheek, F. De Angeli, C. Fabricius, R. Guerra, B. Holl, E. Masana, R. Messineo, N. Mowlavi, K. Nienartowicz, P. Panuzzo, J. Portell, M. Riello, G. M. Seabroke, P. Tanga, F. Thévenin, G. Gracia-Abril, G. Comoretto, M. Garcia-Reinaldos, D. Teyssier, M. Altmann, R. Andrae, M. Audard, I. Bellas-Velidis, K. Benson, J. Berthier, R. Blomme, P. Burgess, G. Busso, B. Carry, A. Cellino, G. Clementini, M. Clotet, O. Creevey, M. Davidson, J. De Ridder, L. Delchambre, A. Dell'Oro, C. Ducourant, J. Fernández-Hernández, M. Fouesneau, Y. Frémat, L. Galluccio, M. García-Torres, J. González-Núñez, J. J. González-Vidal, E. Gosset, L. P. Guy, J.-L. Halbwachs, N. C. Hambly, D. L. Harrison, J. Hernández, D. Hestroffer, S. T. Hodgkin, A. Hutton, G. Jasniewicz, A. Jean-Antoine-Piccolo, S. Jordan, A. J. Korn, A. Krone-Martins, A. C. Lanzafame, T. Lebzelter, W. Löffler, M. Manteiga, P. M. Marrese, J. M. Martín-Fleitas, A. Moitinho, A. Mora, K. Muinonen, J. Osinde, E. Pan-

cino, T. Pauwels, J.-M. Petit, A. Recio-Blanco, P. J. Richards, L. Rimoldini, A. C. Robin, L. M. Sarro, C. Siopis, M. Smith, A. Sozzetti, M. Süveges, J. Torra, W. van Reeven, U. Abbas, A. Abreu Aramburu, S. Accart, C. Aerts, G. Altavilla, M. A. Álvarez, R. Alvarez, J. Alves, R. I. Anderson, A. H. Andrei, E. Anglada Varela, E. Antiche, T. Antoja, B. Arcay, T. L. Astraatmadja, N. Bach, S. G. Baker, L. Balaguer-Núñez, P. Balm, C. Barache, C. Barata, D. Barbato, F. Barblan, P. S. Barklem, D. Barrado, M. Barros, M. A. Barstow, S. Bartholomé Muñoz, J.-L. Bassilana, U. Becciani, M. Bellazzini, A. Berihuete, S. Bertone, L. Bianchi, O. Bienaymé, S. Blanco-Cuaresma, T. Boch, C. Boeche, A. Bombrun, R. Borrachero, D. Bossini, S. Bouquillon, G. Bourda, A. Bragaglia, L. Bramante, M. A. Breddels, A. Bressan, N. Brouillet, T. Brüsemeister, E. Brugaletta, B. Bucciarelli, A. Burlacu, D. Busonero, A. G. Butkevich, R. Buzzi, E. Caffau, R. Cancelliere, G. Cannizzaro, T. Cantat-Gaudin, R. Carballo, T. Carlucci, J. M. Carrasco, L. Casamiquela, M. Castellani, A. Castro-Ginard, P. Charlot, L. Chemin, A. Chiavassa, G. Cocozza, G. Costigan, S. Cowell, F. Crifo, M. Crosta, C. Crowley, J. Cuypers†, C. Dafonte, Y. Damerdji, A. Dapergolas, P. David, M. David, P. de Laverny, F. De Luise, R. De March, D. de Martino, R. de Souza, A. de Torres, J. Debosscher, E. del Pozo, M. Delbo, A. Delgado, H. E. Delgado, P. Di Matteo, S. Diakite, C. Diener, E. Distefano, C. Dolding, P. Drazinos, J. Durán, B. Edvardsson, H. Enke, K. Eriksson, P. Esquej, G. Eynard Bontemps, C. Fabre, M. Fabrizio, S. Faigler, A. J. Falcão, M. Farràs Casas, L. Federici, G. Fedorets, P. Fernique, F. Figueras, F. Filippi, K. Findeisen, A. Fonti, E. Fraile, M. Fraser, B. Frézouls, M. Gai, S. Galleti, D. Garabato, F. García-Sedano, A. Garofalo, N. Garralda, A. Gavel, P. Gavras, J. Gerssen, R. Geyer, P. Giacobbe, G. Gilmore, S. Girona, G. Giuffrida, F. Glass, M. Gomes, M. Granvik, A. Gueguen, A. Guerrier, J. Guiraud, R. Gutiérrez-Sánchez, R. Haigron, D. Hatzidimitriou, M. Hauser, M. Haywood, U. Heiter, A. Helmi, J. Heu, T. Hilger, D. Hobbs, W. Hofmann, G. Holland, H. E. Huckle, A. Hypki, V. Icardi, K. Janßen, G. Jevardat de Fombelle, P. G. Jonker, Á. L. Juhász, F. Julbe, A. Karampelas, A. Kewley, J. Klar, A. Kochoska, R. Kohley, K. Kolenberg, M. Kontizas, E. Kontizas, S. E. Koposov, G. Kordopatis, Z. Kostrzewa-Rutkowska, P. Koubsky, S. Lambert, A. F. Lanza, Y. Lasne, J.-B. Lavigne, Y. Le Fustec, C. Le Poncin-Lafitte, Y. Lebreton, S. Leccia, N. Leclerc, I. Lecoeur-Taibi, H. Lenhardt, F. Leroux, S. Liao, E. Licata, H. E. P. Lindstrøm, T. A. Lister, E. Livanou, A. Lobel, M. López, S. Managau, R. G. Mann, G. Mantelet, O. Marchal, J. M. Marchant, M. Marconi, S. Marinoni, G. Marschalkó, D. J. Marshall, M. Martino, G. Marton, N. Mary, D. Massari, G. Matijevič, T. Mazeh, P. J. McMillan, S. Messina, D. Michalik, N. R. Millar, D. Molina, R. Molinaro, L. Molnár, P. Montegriffo, R. Mor, R. Morbidelli, T. Morel, D. Morris, A. F. Mulone, T. Muraveva, I. Musella, G. Nelemans, L. Nicastro, L. Noval, W. O'Mullane, C. Ordénovic, D. Ordóñez-Blanco, P. Osborne, C. Pagani, I. Pagano, F. Pailler, H. Palacin, L. Palaversa, A. Panahi, M. Pawlak, A. M. Piersimoni, F.-X. Pineau, E. Plachy, G. Plum, E. Poggio, E. Poujoulet, A. Prša, L. Pulone, E. Racero, S. Ragaini, N. Rambaux, M. Ramos-Lerate, S. Regibo, C. Reylé, F. Riclet, V. Ripepi, A. Riva, A. Rivard, G. Rixon, T. Roegiers, M. Roelens, M. Romero-Gómez, N. Rowell, F. Royer, L. Ruiz-Dern, G. Sadowski, T. Sagristà Sellés, J. Sahlmann, J. Salgado, E. Salguero, N. Sanna, T. Santana-Ros, M. Sarasso, H. Savietto, M. Schultheis, E. Sciacca, M. Segol, J. C. Segovia, D. Ségransan, I-C. Shih, L. Siltala, A. F. Silva, R. L. Smart, K. W. Smith, E. Solano, F. Solitro, R. Sordo, S. Soria Nieto, J. Souchay, A. Spagna, F. Spoto, U. Stampa, I. A. Steele, H. Steidelmüller, C. A. Stephenson, H. Stoev, F. F. Suess, J. Surdej, L. Szabados, E. Szegedi-Elek, D. Tapiador, F. Taris, G. Tauran, M. B. Taylor, R. Teixeira, D. Terrett, P. Teyssandier, W. Thuillot, A. Titarenko, F. Torra Clotet, C. Turon, A. Ulla, E. Utrilla, S. Uzzi, M. Vaillant, G. Valentini, V. Valette, A. van Elteren, E. Van Hemelryck, M. van Leeuwen, M. Vaschetto, A. Vecchiato, J. Veljanoski, Y. Viala, D. Vicente, S. Vogt, C. von Essen, H. Voss, V. Votruba, S. Voutsinas, G. Walmsley, M. Weiler, O. Wertz, T. Wevers, Ł. Wyrzykowski, A. Yoldas, M. Žerjal, H. Ziaeepour, J. Zorec, S. Zschocke, S. Zucker, C. Zurbach, and T. Zwitter. Gaia data release 2. *AStronomy & Astrophysics*, 616:A1, August 2018.

[7] and R. Abuter, M. Accardo, A. Amorim, N. Anugu, G. Ávila, N. Azouaoui, M. Benisty, J. P. Berger, N. Blind, H. Bonnet, P. Bourget, W. Brandner, R. Brast, A. Buron, L. Burtscher, F. Cassaing, F. Chapron, É. Choquet, Y. Clénet, C. Collin, V. Coudé du Foresto, W. de Wit, P. T. de Zeeuw, C. Deen, F. Delplancke-Ströbele, R. Dembet, F. Derie, J. Dexter, G. Duvert, M. Ebert, A. Eckart, F. Eisenhauer, M. Esselborn, P. Fédou, G. Finger, P. Garcia, C. E. Garcia Dabo, R. Garcia Lopez, E. Gendron, R. Genzel, S. Gillessen, F. Gonte, P. Gordo, M. Grould, U. Grözinger, S. Guieu, P. Haguenauer, O. Hans, X. Haubois, M. Haug, F. Haussmann, Th. Henning, S. Hippler, M. Horrobin, A. Huber, Z. Hubert, N. Hubin, C. A. Hummel, G. Jakob, A. Janssen, L. Jochum, L. Jocou, A. Kaufer, S. Kellner, S. Kendrew, L. Kern, P. Kervella, M. Kiekebusch, R. Klein, Y. Kok, J. Kolb, M. Kulas, S. Lacour, V. Lapeyrère, B. Lazareff, J.-B. Le Bouquin, P. Lèna, R. Lenzen, S. Lévêque, M. Lippa, Y. Magnard, L. Mehrgan, M. Mellein, A. Mérand, J. Moreno-Ventas, T. Moulin, E. Müller, F. Müller, U. Neumann, S. Oberti, T. Ott, L. Pallanca, J. Panduro, L. Pasquini, T. Paumard, I. Percheron, K. Perraut, G. Perrin, A. Pflüger, O. Pfuhl, T. Phan Duc, P. M. Plewa, D. Popovic, S. Rabien, A. Ramírez, J. Ramos,

C. Rau, M. Riquelme, R.-R. Rohloff, G. Rousset, J. Sanchez-Bermudez, S. Scheithauer, M. Schöller, N. Schuhler, J. Spyromilio, C. Straubmeier, E. Sturm, M. Suarez, K. R. W. Tristram, N. Ventura, F. Vincent, I. Waisberg, I. Wank, J. Weber, E. Wieprecht, M. Wiest, E. Wiezorrek, M. Wittkowski, J. Woillez, B. Wolff, S. Yazici, D. Ziegler, and G. Zins. First light for GRAVITY: Phase referencing optical interferometry for the very large telescope interferometer. *AStronomy & Astrophysics*, 602:A94, June 2017.

[8] and T. Prusti, J. H. J. de Bruijne, A. G. A. Brown, A. Vallenari, C. Babusiaux, C. A. L. Bailer-Jones, U. Bastian, M. Biermann, D. W. Evans, L. Eyer, F. Jansen, C. Jordi, S. A. Klioner, U. Lammers, L. Lindegren, X. Luri, F. Mignard, D. J. Milligan, C. Panem, V. Poinsignon, D. Pourbaix, S. Randich, G. Sarri, P. Sartoretti, H. I. Siddiqui, C. Soubiran, V. Valette, F. van Leeuwen, N. A. Walton, C. Aerts, F. Arenou, M. Cropper, R. Drimmel, E. Høg, D. Katz, M. G. Lattanzi, W. O'Mullane, E. K. Grebel, A. D. Holland, C. Huc, X. Passot, L. Bramante, C. Cacciari, J. Castañeda, L. Chaoul, N. Cheek, F. De Angeli, C. Fabricius, R. Guerra, J. Hernández, A. Jean-Antoine-Piccolo, E. Masana, R. Messineo, N. Mowlavi, K. Nienartowicz, D. Ordóñez-Blanco, P. Panuzzo, J. Portell, P. J. Richards, M. Riello, G. M. Seabroke, P. Tanga, F. Thévenin, J. Torra, S. G. Els, G. Gracia-Abril, G. Comoretto, M. Garcia-Reinaldos, T. Lock, E. Mercier, M. Altmann, R. Andrae, T. L. Astraatmadja, I. Bellas-Velidis, K. Benson, J. Berthier, R. Blomme, G. Busso, B. Carry, A. Cellino, G. Clementini, S. Cowell, O. Creevey, J. Cuypers, M. Davidson, J. De Ridder, A. de Torres, L. Delchambre, A. Dell'Oro, C. Ducourant, Y. Frémat, M. García-Torres, E. Gosset, J.-L. Halbwachs, N. C. Hambly, D. L. Harrison, M. Hauser, D. Hestroffer, S. T. Hodgkin, H. E. Huckle, A. Hutton, G. Jasniewicz, S. Jordan, M. Kontizas, A. J. Korn, A. C. Lanzafame, M. Manteiga, A. Moitinho, K. Muinonen, J. Osinde, E. Pancino, T. Pauwels, J.-M. Petit, A. Recio-Blanco, A. C. Robin, L. M. Sarro, C. Siopis, M. Smith, K. W. Smith, A. Sozzetti, W. Thuillot, W. van Reeven, Y. Viala, U. Abbas, A. Abreu Aramburu, S. Accart, J. J. Aguado, P. M. Allan, W. Allasia, G. Altavilla, M. A. Álvarez, J. Alves, R. I. Anderson, A. H. Andrei, E. Anglada Varela, E. Antiche, T. Antoja, S. Antón, B. Arcay, A. Atzei, L. Ayache, N. Bach, S. G. Baker, L. Balaguer-Núñez, C. Barache, C. Barata, A. Barbier, F. Barblan, M. Baroni, D. Barrado y Navascués, M. Barros, M. A. Barstow, U. Becciani, M. Bellazzini, G. Bellei, A. Bello García, V. Belokurov, P. Bendjoya, A. Berihuete, L. Bianchi, O. Bienaymé, F. Billebaud, N. Blagorodnova, S. Blanco-Cuaresma, T. Boch, A. Bombrun, R. Borrachero, S. Bouquillon, G. Bourda, H. Bouy, A. Bragaglia, M. A. Breddels, N. Brouillet, T. Brüsemeister, B. Bucciarelli, F. Budnik, P. Burgess, R. Burgon, A. Burlacu, D. Busonero, R. Buzzi, E. Caffau, J. Cambras,

H. Campbell, R. Cancelliere, T. Cantat-Gaudin, T. Carlucci, J. M. Carrasco, M. Castellani, P. Charlot, J. Charnas, P. Charvet, F. Chassat, A. Chiavassa, M. Clotet, G. Cocozza, R. S. Collins, P. Collins, G. Costigan, F. Crifo, N. J. G. Cross, M. Crosta, C. Crowley, C. Dafonte, Y. Damerdji, A. Dapergolas, P. David, M. David, P. De Cat, F. de Felice, P. de Laverny, F. De Luise, R. De March, D. de Martino, R. de Souza, J. Debosscher, E. del Pozo, M. Delbo, A. Delgado, H. E. Delgado, F. di Marco, P. Di Matteo, S. Diakite, E. Distefano, C. Dolding, S. Dos Anjos, P. Drazinos, J. Durán, Y. Dzigan, E. Ecale, B. Edvardsson, H. Enke, M. Erdmann, D. Escolar, M. Espina, N. W. Evans, G. Eynard Bontemps, C. Fabre, M. Fabrizio, S. Faigler, A. J. Falcão, M. Farràs Casas, F. Faye, L. Federici, G. Fedorets, J. Fernández-Hernández, P. Fernique, A. Fienga, F. Figueras, F. Filippi, K. Findeisen, A. Fonti, M. Fouesneau, E. Fraile, M. Fraser, J. Fuchs, R. Furnell, M. Gai, S. Galleti, L. Galluccio, D. Garabato, F. García-Sedano, P. Garé, A. Garofalo, N. Garralda, P. Gavras, J. Gerssen, R. Geyer, G. Gilmore, S. Girona, G. Giuffrida, M. Gomes, A. González-Marcos, J. González-Núñez, J. J. González-Vidal, M. Granvik, A. Guerrier, P. Guillout, J. Guiraud, A. Gúrpide, R. Gutiérrez-Sánchez, L. P. Guy, R. Haigron, D. Hatzidimitriou, M. Haywood, U. Heiter, A. Helmi, D. Hobbs, W. Hofmann, B. Holl, G. Holland, J. A. S. Hunt, A. Hypki, V. Icardi, M. Irwin, G. Jevardat de Fombelle, P. Jofré, P. G. Jonker, A. Jorissen, F. Julbe, A. Karampelas, A. Kochoska, R. Kohley, K. Kolenberg, E. Kontizas, S. E. Koposov, G. Kordopatis, P. Koubsky, A. Kowalczyk, A. Krone-Martins, M. Kudryashova, I. Kull, R. K. Bachchan, F. Lacoste-Seris, A. F. Lanza, J.-B. Lavigne, C. Le Poncin-Lafitte, Y. Lebreton, T. Lebzelter, S. Leccia, N. Leclerc, I. Lecoeur-Taibi, V. Lemaitre, H. Lenhardt, F. Leroux, S. Liao, E. Licata, H. E. P. Lindstrøm, T. A. Lister, E. Livanou, A. Lobel, W. Löffler, M. López, A. Lopez-Lozano, D. Lorenz, T. Loureiro, I. MacDonald, T. Magalhães Fernandes, S. Managau, R. G. Mann, G. Mantelet, O. Marchal, J. M. Marchant, M. Marconi, J. Marie, S. Marinoni, P. M. Marrese, G. Marschalkó, D. J. Marshall, J. M. Martín-Fleitas, M. Martino, N. Mary, G. Matijevič, T. Mazeh, P. J. McMillan, S. Messina, A. Mestre, D. Michalik, N. R. Millar, B. M. H. Miranda, D. Molina, R. Molinaro, M. Molinaro, L. Molnár, M. Moniez, P. Montegriffo, D. Monteiro, R. Mor, A. Mora, R. Morbidelli, T. Morel, S. Morgenthaler, T. Morley, D. Morris, A. F. Mulone, T. Muraveva, I. Musella, J. Narbonne, G. Nelemans, L. Nicastro, L. Noval, C. Ordénovic, J. Ordieres-Meré, P. Osborne, C. Pagani, I. Pagano, F. Pailler, H. Palacin, L. Palaversa, P. Parsons, T. Paulsen, M. Pecoraro, R. Pedrosa, H. Pentikäinen, J. Pereira, B. Pichon, A. M. Piersimoni, F.-X. Pineau, E. Plachy, G. Plum, E. Poujoulet, A. Prša, L. Pulone, S. Ragaini, S. Rago, N. Rambaux, M. Ramos-Lerate, P. Ranalli, G. Rauw,

A. Read, S. Regibo, F. Renk, C. Reylé, R. A. Ribeiro, L. Rimoldini, V. Ripepi, A. Riva, G. Rixon, M. Roelens, M. Romero-Gómez, N. Rowell, F. Royer, A. Rudolph, L. Ruiz-Dern, G. Sadowski, T. Sagristà Sellés, J. Sahlmann, J. Salgado, E. Salguero, M. Sarasso, H. Savietto, A. Schnorhk, M. Schultheis, E. Sciacca, M. Segol, J. C. Segovia, D. Segransan, E. Serpell, I-C. Shih, R. Smareglia, R. L. Smart, C. Smith, E. Solano, F. Solitro, R. Sordo, S. Soria Nieto, J. Souchay, A. Spagna, F. Spoto, U. Stampa, I. A. Steele, H. Steidelmüller, C. A. Stephenson, H. Stoev, F. F. Suess, M. Süveges, J. Surdej, L. Szabados, E. Szegedi-Elek, D. Tapiador, F. Taris, G. Tauran, M. B. Taylor, R. Teixeira, D. Terrett, B. Tingley, S. C. Trager, C. Turon, A. Ulla, E. Utrilla, G. Valentini, A. van Elteren, E. Van Hemelryck, M. van Leeuwen, M. Varadi, A. Vecchiato, J. Veljanoski, T. Via, D. Vicente, S. Vogt, H. Voss, V. Votruba, S. Voutsinas, G. Walmsley, M. Weiler, K. Weingrill, D. Werner, T. Wevers, G. Whitehead, Ł. Wyrzykowski, A. Yoldas, M. Žerjal, S. Zucker, C. Zurbach, T. Zwitter, A. Alecu, M. Allen, C. Allende Prieto, A. Amorim, G. Anglada-Escudé, V. Arsenijevic, S. Azaz, P. Balm, M. Beck, H.-H. Bernstein, L. Bigot, A. Bijaoui, C. Blasco, M. Bonfigli, G. Bono, S. Boudreault, A. Bressan, S. Brown, P.-M. Brunet, P. Bunclark, R. Buonanno, A. G. Butkevich, C. Carret, C. Carrion, L. Chemin, F. Chéreau, L. Corcione, E. Darmigny, K. S. de Boer, P. de Teodoro, P. T. de Zeeuw, C. Delle Luche, C. D. Domingues, P. Dubath, F. Fodor, B. Frézouls, A. Fries, D. Fustes, D. Fyfe, E. Gallardo, J. Gallegos, D. Gardiol, M. Gebran, A. Gomboc, A. Gómez, E. Grux, A. Gueguen, A. Heyrovsky, J. Hoar, G. Iannicola, Y. Isasi Parache, A.-M. Janotto, E. Joliet, A. Jonckheere, R. Keil, D.-W. Kim, P. Klagyivik, J. Klar, J. Knude, O. Kochukhov, I. Kolka, J. Kos, A. Kutka, V. Lainey, D. LeBouquin, C. Liu, D. Loreggia, V. V. Makarov, M. G. Marseille, C. Martayan, O. Martinez-Rubi, B. Massart, F. Meynadier, S. Mignot, U. Munari, A.-T. Nguyen, T. Nordlander, P. Ocvirk, K. S. O'Flaherty, A. Olias Sanz, P. Ortiz, J. Osorio, D. Oszkiewicz, A. Ouzounis, M. Palmer, P. Park, E. Pasquato, C. Peltzer, J. Peralta, F. Péturaud, T. Pieniluoma, E. Pigozzi, J. Poels, G. Prat, T. Prod'homme, F. Raison, J. M. Rebordao, D. Risquez, B. Rocca-Volmerange, S. Rosen, M. I. Ruiz-Fuertes, F. Russo, S. Sembay, I. Serraller Vizcaino, A. Short, A. Siebert, H. Silva, D. Sinachopoulos, E. Slezak, M. Soffel, D. Sosnowska, V. Straižys, M. ter Linden, D. Terrell, S. Theil, C. Tiede, L. Troisi, P. Tsalmantza, D. Tur, M. Vaccari, F. Vachier, P. Valles, W. Van Hamme, L. Veltz, J. Virtanen, J.-M. Wallut, R. Wichmann, M. I. Wilkinson, H. Ziaeepour, and S. Zschocke. The gaia mission. *AStronomy & Astrophysics*, 595:A1, November 2016.

[9] Astropy Collaboration, A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig,

N. Dencheva, A. Ginsburg, J. T. Vand erPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D'Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, 156(3):123, September 2018.

[10] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pages 1–5, 2016.

[11] Dalya Baron. Machine learning in astronomy: a practical overview, 2019.

[12] I Becker, K Pichara, M Catelan, P Protopapas, C Aguirre, and F Nikzat. Scalable end-to-end recurrent neural network for variable star classification. *Monthly Notices of the Royal Astronomical Society*, 493(2):2981–2995, 02 2020.

[13] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[14] M. Bertero and P. Boccacci. Image deconvolution. In *NATO Security through Science Series*, pages 349–370. Springer Netherlands, 2005.

[15] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astrophysics and Space Science*, 117:393–404, June 1996.

[16] Michael R. Blanton, Matthew A. Bershady, Bela Abolfathi, Franco D. Albareti, Carlos Allende Prieto, Andres Almeida, Javier Alonso-García, Friedrich Anders, Scott F. Anderson, Brett Andrews, Erik Aquino-Ortíz, Alfonso Aragón-Salamanca, Maria Argudo-Fernández, Eric Armengaud, Eric Aubourg, Vladimir Avila-Reese, Carles Badenes, Stephen Bailey, Kathleen A. Barger, Jorge Barrera-Ballesteros, Curtis Bartosz, Dominic Bates, Falk Baumgarten, Julian Bautista, Rachael Beaton, Timothy C. Beers, Francesco Belfiore, Chad F. Bender, Andreas A. Berlind, Mariangela Bernardi, Florian Beutler, Jonathan C. Bird, Dmitry Bizyaev, Guillermo A. Blanc, Michael Blomqvist, Adam S. Bolton, Médéric Boquien, Jura Borissova, Remco van den Bosch, Jo Bovy, William N. Brandt, Jonathan Brinkmann, Joel R. Brownstein, Kevin Bundy, Adam J. Burgasser, Etienne Burtin, Nicolás G. Busca, Michele Cappellari, Maria Leticia Delgado Carigi, Joleen K. Carlberg, Aurelio Carnero Rosell, Ricardo Carrera, Nancy J. Chanover, Brian Cherinka, Edmond Cheung, Yilen Gómez Maqueo Chew, Cristina Chiappini, Peter Doohyun Choi, Drew Chojnowski, Chia-Hsun Chuang, Haeun Chung, Rafael Fernando Cirolini, Nicolas Clerc, Roger E. Cohen, Johan Comparat, Luiz da Costa, Marie-Claude Cousinou, Kevin Covey, Jeffrey D. Crane, Rupert A. C. Croft, Irene Cruz-Gonzalez, Daniel Garrido Cuadra, Katia Cunha, Guillermo J. Damke, Jeremy Darling, Roger Davies, Kyle Dawson, Axel de la Macorra, Flavia Dell'Agli, Nathan De Lee, Timothée Delubac, Francesco Di Mille, Aleks Diamond-Stanic, Mariana Cano-Díaz, John Donor, Juan José Downes, Niv Drory, Hélion du Mas des Bourboux, Christopher J. Duckworth, Tom Dwelly, Jamie Dyer, Garrett Ebelke, Arthur D. Eigenbrot, Daniel J. Eisenstein, Eric Emsellem, Mike Eracleous, Stephanie Escoffier, Michael L. Evans, Xiaohui Fan, Emma Fernández-Alvar, J. G. Fernandez-Trincado, Diane K. Feuillet, Alexis Finoguenov, Scott W. Fleming, Andreu Font-Ribera, Alexander Fredrickson, Gordon Freischlad, Peter M. Frinchaboy, Carla E. Fuentes, Lluís Galbany, R. Garcia-Dias, D. A. García-Hernández, Patrick Gaulme, Doug Geisler, Joseph D. Gelfand, Héctor Gil-Marín, Bruce A. Gillespie, Daniel Goddard, Violeta Gonzalez-Perez, Kathleen Grabowski, Paul J. Green, Catherine J. Grier, James E. Gunn, Hong Guo, Julien Guy, Alex Hagen, ChangHoon Hahn, Matthew Hall, Paul Harding, Sten Hasselquist, Suzanne L. Hawley, Fred Hearty, Jonay I. Gonzalez Hernández, Shirley Ho, David W. Hogg, Kelly Holley-Bockelmann, Jon A. Holtzman, Parker H.

Holzer, Joseph Huehnerhoff, Timothy A. Hutchinson, Ho Seong Hwang, Héctor J. Ibarra-Medel, Gabriele da Silva Ilha, Inese I. Ivans, KeShawn Ivory, Kelly Jackson, Trey W. Jensen, Jennifer A. Johnson, Amy Jones, Henrik Jönsson, Eric Jullo, Vikrant Kamble, Karen Kinemuchi, David Kirkby, Francisco-Shu Kitaura, Mark Klaene, Gillian R. Knapp, Jean-Paul Kneib, Juna A. Kollmeier, Ivan Lacerna, Richard R. Lane, Dustin Lang, David R. Law, Daniel Lazarz, Youngbae Lee, Jean-Marc Le Goff, Fu-Heng Liang, Cheng Li, Hongyu Li, Jianhui Lian, Marcos Lima, Lihwai Lin, Yen-Ting Lin, Sara Bertran de Lis, Chao Liu, Miguel Angel C. de Icaza Lizaola, Dan Long, Sara Lucatello, Britt Lundgren, Nicholas K. MacDonald, Alice Deconto Machado, Chelsea L. MacLeod, Suvrath Mahadevan, Marcio Antonio Geimba Maia, Roberto Maiolino, Steven R. Majewski, Elena Malanushenko, Viktor Malanushenko, Arturo Manchado, Shude Mao, Claudia Maraston, Rui Marques-Chaves, Thomas Masseron, Karen L. Masters, Cameron K. McBride, Richard M. McDermid, Brianne McGrath, Ian D. McGreer, Nicolás Medina Peña, Matthew Melendez, Andrea Merloni, Michael R. Merrifield, Szabolcs Meszaros, Andres Meza, Ivan Minchev, Dante Minniti, Takamitsu Miyaji, Surhud More, John Mulchaey, Francisco Müller-Sánchez, Demitri Muna, Ricardo R. Munoz, Adam D. Myers, Preethi Nair, Kirpal Nandra, Janaina Correa do Nascimento, Alenka Negrete, Melissa Ness, Jeffrey A. Newman, Robert C. Nichol, David L. Nidever, Christian Nitschelm, Pierros Ntelis, Julia E. O'Connell, Ryan J. Oelkers, Audrey Oravetz, Daniel Oravetz, Zach Pace, Nelson Padilla, Nathalie Palanque-Delabrouille, Pedro Alonso Palicio, Kaike Pan, John K. Parejko, Taniya Parikh, Isabelle Pâris, Changbom Park, Alim Y. Patten, Sebastien Peirani, Marcos Pellejero-Ibanez, Samantha Penny, Will J. Percival, Ismael Perez-Fournon, Patrick Petitjean, Matthew M. Pieri, Marc Pinsonneault, Alice Pisani, Radosław Poleski, Francisco Prada, Abhishek Prakash, Anna Bárbara de Andrade Queiroz, M. Jordan Raddick, Anand Raichoor, Sandro Barboza Rembold, Hannah Richstein, Rogemar A. Riffel, Rogério Riffel, Hans-Walter Rix, Annie C. Robin, Constance M. Rockosi, Sergio Rodríguez-Torres, A. Roman-Lopes, Carlos Román-Zúñiga, Margarita Rosado, Ashley J. Ross, Graziano Rossi, John Ruan, Rossana Ruggeri, Eli S. Rykoff, Salvador Salazar-Albornoz, Mara Salvato, Ariel G. Sánchez, D. S. Aguado, José R. Sánchez-Gallego, Felipe A. Santana, Basílio Xavier Santiago, Conor Sayres, Ricardo P. Schiavon, Jaderson da Silva Schimoia, Edward F. Schlafly, David J. Schlegel, Donald P. Schneider, Mathias Schultheis, William J. Schuster, Axel Schwope, Hee-Jong Seo, Zhengyi Shao, Shiyin Shen, Matthew Shetrone, Michael Shull, Joshua D. Simon, Danielle Skinner, M. F. Skrutskie, Anže Slosar, Verne V. Smith, Jennifer S. Sobeck, Flavia Sobreira, Garrett Somers, Diogo Souto, David V.

Stark, Keivan Stassun, Fritz Stauffer, Matthias Steinmetz, Thaisa Storchi-Bergmann, Alina Streblyanska, Guy S. Stringfellow, Genaro Suárez, Jing Sun, Nao Suzuki, Laszlo Szigeti, Manuchehr Taghizadeh-Popp, Baitian Tang, Charling Tao, Jamie Tayar, Mita Tembe, Johanna Teske, Aniruddha R. Thakar, Daniel Thomas, Benjamin A. Thompson, Jeremy L. Tinker, Patricia Tissera, Rita Tojeiro, Hector Hernandez Toledo, Sylvain de la Torre, Christy Tremonti, Nicholas W. Troup, Octavio Valenzuela, Inma Martinez Valpuesta, Jaime Vargas-González, Mariana Vargas-Magaña, Jose Alberto Vazquez, Sandro Villanova, M. Vivek, Nicole Vogt, David Wake, Rene Walterbos, Yuting Wang, Benjamin Alan Weaver, Anne-Marie Weijmans, David H. Weinberg, Kyle B. Westfall, David G. Whelan, Vivienne Wild, John Wilson, W. M. Wood-Vasey, Dominika Wylezalek, Ting Xiao, Renbin Yan, Meng Yang, Jason E. Ybarra, Christophe Yèche, Nadia Zakamska, Olga Zamora, Pauline Zarrouk, Gail Zasowski, Kai Zhang, Gong-Bo Zhao, Zheng Zheng, Zheng Zheng, Xu Zhou, Zhi-Min Zhou, Guangtun B. Zhu, Manuela Zoccali, and Hu Zou. Sloan Digital Sky Survey IV: Mapping the Milky Way, Nearby Galaxies, and the Distant Universe. *The Astronomical Journal*, 154(1):28, July 2017.

[17] M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light.* Pergamon Press, 1959.

[18] Alexandre Boucaud, Marc Huertas-Company, Caroline Heneka, Emille E O Ishida, Nima Sedaghat, Rafael S de Souza, Ben Moews, Hervé Dole, Marco Castellano, Emiliano Merlin, Valerio Roscani, Andrea Tramacere, Madhura Killedar, Arlindo M M Trindade, and Collaboration for the COIN. Photometry of high-redshift blended galaxies using deep learning. *Monthly Notices of the Royal Astronomical Society*, 491(2):2481–2495, 12 2019.

[19] Micah Bowles, Anna M M Scaife, Fiona Porter, Hongming Tang, and David J Bastien. Attention-gating for improved radio galaxy classification. *Monthly Notices of the Royal Astronomical Society*, 501(3):4579–4595, December 2020.

[20] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, January 1999.

[21] G. Brand. XI. observations of altitude and azimuth of the great comet of 1843, made at st. helena. *Monthly Notices of the Royal Astronomical Society*, 6(11):136–136, December 1844.

[22] Massimo Brescia, S G Djorgovski, Eric D Feigelson, Giuseppe Longo, and Stefano Cavuoti, editors. *Proceedings of the International Astronomical*

Union Symposia and Colloquia: Astroinformatics (IAU S325). Cambridge University Press, Cambridge, England, June 2017.

[23] Guillermo Cabrera-Vives, Ignacio Reyes, Francisco Förster, Pablo A. Estévez, and Juan-Carlos Maureira. Deep-hits: Rotation invariant convolutional neural network for transient detection. *The Astrophysical Journal*, 836(1):97, feb 2017.

[24] John M. Carpenter, Andrea Corvilló n, Jennifer Donovan Meyer, Adele L. Plunkett, Robert Kurowski, Alex Chalevin, and Enrique Macías. Update on the systematics in the ALMA proposal review process after cycle 8. *Publications of the Astronomical Society of the Pacific*, 134(1034):045001, apr 2022.

[25] Rodrigo Carrasco-Davis, Guillermo Cabrera-Vives, Francisco Förster, Pablo A. Estévez, Pablo Huijse, Pavlos Protopapas, Ignacio Reyes, Jorge Martínez-Palomera, and Cristóbal Donoso. Deep learning for image sequence classification of astronomical events. *Publications of the Astronomical Society of the Pacific*, 131(1004):108006, sep 2019.

[26] S. Casertano, M. G. Lattanzi, A. Sozzetti, A. Spagna, S. Jancart, R. Morbidelli, R. Pannunzio, D. Pourbaix, and D. Queloz. Double-blind test program for astrometric planet detection with gaia. *AStronomy & Astrophysics*, 482(2):699–729, February 2008.

[27] D Chen. Step mission: high-precision space astrometry to search for terrestrial exoplanets. *Journal of Instrumentation*, 9(4):C04040–C04040, April 2014.

[28] XI Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 864–872. PMLR, 10–15 Jul 2018.

[29] Ting-Yun Cheng, Christopher J Conselice, Alfonso Aragón-Salamanca, Nan Li, Asa F L Bluck, Will G Hartley, James Annis, David Brooks, Peter Doel, Juan García-Bellido, David J James, Kyler Kuehn, Nikolay Kuropatkin, Mathew Smith, Flavia Sobreira, and Gregory Tarle. Optimizing automatic morphological classification of galaxies with machine learning and deep learning using dark energy survey imaging. *Monthly Notices of the Royal Astronomical Society*, 493(3):4209–4228, February 2020.

[30] Ting-Yun Cheng, Nan Li, Christopher J. Conselice, Alfonso Aragón-Salamanca, Simon Dye, and Robert B. Metcalf. Identifying strong

lenses with unsupervised machine learning using convolutional autoencoder. *Monthly Notices of the Royal Astronomical Society*, 494(3):3750–3765, April 2020.

[31] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[32] B. G. Clark. An efficient implementation of the algorithm 'clean'. *Astronomy and Astrophysiscs*, 89(3):377, sep 1980.

[33] Liam Connor, Katherine L Bouman, Vikram Ravi, and Gregg Hallinan. Deep radio-interferometric imaging with POLISH: DSA-2000 and weak lensing. *Monthly Notices of the Royal Astronomical Society*, 514(2):2614–2626, May 2022.

[34] Tim J. Cornwell. Multiscale CLEAN deconvolution of radio synthesis images. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):793–801, October 2008.

[35] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[36] Paulo Cortes, Anthony Remijan, Andy Biggs, Bill Dent, John Carpenter, Ed Fomalont, Antonio Hales, Seiji Kameno, Brian Mason, N. Philips, Kamaljeet Saini, Felix Stoehr, Baltasar Vila Vilaro, and Eric Villard. *ALMA Cycle 8 2021 Technical Handbook*, March 2021. Additional contributors to this edition: Tim Bastian, Crystal Brogan, John Carpenter, Chinshin Chang, Geoff Crew, Paul Fisher, Misato Fukagawa, Melissa Hoffman, Chat Hull, Hugo Messias, Todd Hunter, Ruediger Kneissl, Andy Lipnicky, Ryan Loomis, Sergio Martin, Lynn Matthews, Luke Maud, Yusuke Miyamoto, Tony Mroczkowski, Hiroshi Nagai, Kouichiro Nakanishi, Masumi Shimojo, Richard Simon, Jeremy Thorley, MCarmen Toribio, Catarina Ubach, Catherine Vlahakis, Martin Zwaan.

[37] Robert A. Crain, Joop Schaye, Richard G. Bower, Michelle Furlong, Matthieu Schaller, Tom Theuns, Claudio Dalla Vecchia, Carlos S. Frenk, Ian G. McCarthy, John C. Helly, Adrian Jenkins, Yetli M. Rosas-Guevara, Simon D. M. White, and James W. Trayford. The EAGLE simulations of galaxy formation: calibration of subgrid physics and model variations. *Monthly Notices of the Royal Astronomical Society*, 450(2):1937–1961, April 2015.

[38] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[39] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[40] A. D'Isanto and K. L. Polsterer. Photometric redshift estimation via deep learning. *Astronomy & Astrophysics*, 609:A111, January 2018.

[41] Lars Doorenbos, Olena Torbaniuk, Stefano Cavuoti, Maurizio Paolillo, Giuseppe Longo, Massimo Brescia, Raphael Sznitman, and Pablo Márquez-Neila. ulisse: A tool for one-shot sky exploration and its application for detection of active galactic nuclei. *Astronomy & Astrophysics*, 666:A171, October 2022.

[42] T.A. Driscoll and R.J. Braun. *Fundamentals of Numerical Computation.* Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2017.

[43] Roberta Duarte, Rodrigo Nemmen, and João Paulo Navarro. Black hole weather forecasting with deep learning: a pilot study. *Monthly Notices of the Royal Astronomical Society*, 512(4):5848–5861, March 2022.

[44] Thierry Dumas, Aline Roumy, and Christine Guillemot. Autoencoder based image compression: Can the learning be quantization independent? In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1188–1192, 2018.

[45] Torsten A. Enßlin, Mona Frommert, and Francisco S. Kitaura. Information field theory for cosmological perturbation reconstruction and nonlinear signal analysis. *Phys. Rev. D*, 80:105005, Nov 2009.

[46] F. Förster, G. Cabrera-Vives, E. Castillo-Navarrete, P. A. Estévez, P. Sánchez-Sáez, J. Arredondo, F. E. Bauer, R. Carrasco-Davis, M. Catelan, F. Elorrieta, S. Eyheramendy, P. Huijse, G. Pignata, E. Reyes, I. Reyes, D. Rodríguez-Mancini, D. Ruz-Mieres, C. Valenzuela, I. Álvarez-Maldonado, N. Astorga, J. Borissova, A. Clocchiatti, D. De Cicco, C. Donoso-Oliva, L. Hernández-García, M. J. Graham, A. Jordán, R. Kurtev, A. Mahabal, J. C. Maureira, A. Muñoz-Arancibia, R. Molina-Ferreiro, A. Moya, W. Palma, M. Pérez-Carrasco, P. Protopapas, M. Romero, L. Sabatini-Gacitua, A. Sánchez, J. San Martín, C. Sepúlveda-Cobo, E. Vera, and J. R. Vergara. The automatic learning for the rapid classification of events (ALeRCE) alert broker. *The Astronomical Journal*, 161(5):242, April 2021.

[47] Joana Frontera-Pons, Florent Sureau, Jerome Bobin, and Emeric Le Floc'h. Unsupervised feature-learning for galaxy seds with denoising autoencoders. *Astronomy & Astrophysics*, 2017.

[48] Seiji Fujimoto, Masamune Oguri, Gabriel Brammer, Yuki Yoshimura, Nicolas Laporte, Jorge González-López, Gabriel B. Caminha, Kotaro Kohno, Adi Zitrin, Johan Richard, Masami Ouchi, Franz E. Bauer, Ian Smail, Bunyo Hatsukade, Yoshiaki Ono, Vasily Kokorev, Hideki Umehata, Daniel Schaerer, Kirsten Knudsen, Fengwu Sun, Georgios Magdis, Francesco Valentino, Yiping Ao, Sune Toft, Miroslava Dessauges-Zavadsky, Kazuhiro Shimasaku, Karina Caputi, Haruka Kusakabe, Kana Morokuma-Matsui, Kikuchihara Shotaro, Eiichi Egami, Minju M. Lee, Timothy Rawle, and Daniel Espada. ALMA lensing cluster survey: Bright [c ii] 158 μm lines from a multiply imaged sub-l galaxy at z = 6.0719. *The Astrophysical Journal*, 911(2):99, April 2021.

[49] Simon Goode, Jeff Cooke, Jielai Zhang, Ashish Mahabal, Sara Webb, and Sarah Hegarty. Machine learning for fast transients for the deeper, wider, faster programme with the removal of BOgus transients (ROBOT) pipeline. *Monthly Notices of the Royal Astronomical Society*, 513(2):1742–1754, April 2022.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[51] John T Gosling. The solar wind. In *Encyclopedia of the Solar System*, pages 99–116. Elsevier, 2007.

[52] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[53] F. Guglielmetti, R. Fischer, and V. Dose. Background-source separation in astronomical images with bayesian probability theory - i. the method. *Monthly Notices of the Royal Astronomical Society*, 396(1):165–190, June 2009.

[54] Isabelle Guyon, Masoud Nikravesh, Steve Gunn, and Lotfi A. Zadeh, editors. *Feature Extraction*. Springer Berlin Heidelberg, 2006.

[55] Olivier Guyon, Eduardo A Bendek, Josh A Eisner, Roger Angel, Neville J Woolf, Thomas D Milster, S Mark Ammons, Michael Shao, Stuart Shaklan, Marie Levine, et al. High-precision astrometry with a diffractive pupil telescope. *The Astrophysical Journal Supplement Series*, 200(2):11, 2012.

[56] Olivier Guyon, Josh A. Eisner, Roger Angel, Neville J. Woolf, Eduardo A. Bendek, Thomas D. Milster, S. Mark Ammons, Michael Shao, Stuart Shaklan, Marie Levine, Bijan Nemati, Frantz Martinache, Joe Pitman, Robert A. Woodruff, and Ruslan Belikov. Simultaneous exoplanet characterization and deep wide-field imaging with a diffractive pupil telescope. *The Astrophysical Journal*, 767(1):11, March 2013.

[57] C. A. Hales, T. Murphy, J. R. Curran, E. Middelberg, B. M. Gaensler, and R. P. Norris. blobcat: software to catalogue flood-filled blobs in radio images of total intensity and linear polarization. *Monthly Notices of the Royal Astronomical Society*, 425(2):979–996, August 2012.

[58] G. Hallinan, V. Ravi, S. Weinreb, J. Kocz, Y. Huang, D. P. Woody, J. Lamb, L. D'Addario, M. Catha, J. Shi, C. Law, S. R. Kulkarni, E. S. Phinney, M. W. Eastwood, K. L. Bouman, M. A. McLaughlin, S. M. Ransom, X. Siemens, J. M. Cordes, R. S. Lynch, D. L. Kaplan, S. Chatterjee, J. Lazio, A. Brazier, S. Bhatnagar, S. T. Myers, F. Walter, and B. M. Gaensler. The dsa-2000 – a radio survey camera, 2019.

[59] Qin Hao-ran, Lin Ji-ming, and Wang Jun-yi. Stacked denoising autoencoders applied to star/galaxy classification. *Chinese Astronomy and Astrophysics*, 41(2):282 – 292, 2017.

[60] P. Hartley, A. Bonaldi, R. Braun, J. N. H. S. Aditya, S. Aicardi, L. Alegre, A. Chakraborty, X. Chen, S. Choudhuri, A. O. Clarke, J. Coles, J. S. Collinson, D. Cornu, L. Darriba, M. Delli Veneri, J. Forbrich, B. Fraga, A. Galan, J. Garrido, F. Gubanov, H. Håkansson, M. J. Hardcastle, C. Heneka, D. Herranz, K. M. Hess, M. Jagannath, S. Jaiswa, R. J. Jurek, D. Korber, S. Kitaeff, D. Kleiner, B. Lao, X. Lu, A. Mazumder, J. Moldón, R. Mondal, S. Ni, M. Önnheim, M. Parra, N. Patra, A. Peel, P. Salomé, S. Sánchez-Expósito, M. Sargent, B. Semelin, P. Serra, A. K. Shaw, A. X. Shen, A. Sjöberg, L. Smith, A. Soroka, V. Stolyarov, E. Tolley, M. C. Toribio, J. M. van der Hulst, A. Vafaei Sadr, L. Verdes-Montenegro, T. Westmeier, K. Yu7 L. Yu, L. Zhang, X. Zhang, Y. Zhang, A. Alberdi, M. Ashdown, C.R. Bom, M. Brüggen, J. Cannon, R. Chen, F. Combes, J. Conway36, F. Courbin, J. Ding, G. Fourestey, J. Freundlich, L. Gao, C. Gheller, Q. Guo, E. Gustavsson, M. Jirstrand, M. G. Jones, G. Józsa, P. Kamphuis, J.-P. Kneib, M. Lindqvist, B. Liu, Y. Liu, Y. Mao, A. Marchal, I. Márquez, A. Meshcheryakov, M. Olberg, N. Oozeer, M. Pandey-Pommier, W. Pe, B. Peng, J. Sabater, A. Sorgho, J.L.Starck, C. Tasse, A. Wang, Y. Wang, H. Xi, X. Yang, H. Zhang, J. Zhang, M. Zhao, and S. Zuo. Ska science data challenge 2: analysis and results. 2022.

[61] Mohammed Hassanin, Saeed Anwar, Ibrahim Radwan, Fahad S Khan, and Ajmal Mian. Visual attention methods in deep learning: An in-depth survey, 2022.

[62] Ryan Hausen and Brant E. Robertson. Morpheus: A deep learning framework for the pixel-level analysis of astronomical image data. *The Astrophysical Journal Supplement Series*, 248(1):20, may 2020.

[63] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[66] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[67] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[68] Jacqueline A. Hodge and Elisabete da Cunha. High-redshift star formation in the atacama large millimetre/submillimetre array era. *Royal Society Open Science*, 7, 2020.

[69] J. A. Högbom. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. *Astronomy & Astrophysics*, 15:417, June 1974.

[70] JA HÖGBOM. Aperture synthesis with a non-regular distribution of interferometer baselines. *ON THE COLLECTION AND ANALYSES OF ASTROPHYSICAL, DATA*, 15:417, 1972.

[71] Mareki Honma, Kazunori Akiyama, Makoto Uemura, and Shiro Ikeda. Super-resolution imaging with radio interferometry using sparse modeling. *Publications of the Astronomical Society of Japan*, 66(5), September 2014.

[72] A. W. Hotan, J. D. Bunton, A. P. Chippendale, M. Whiting, J. Tuthill, V. A. Moss, D. McConnell, S. W. Amy, M. T. Huynh, J. R. Allison, C. S. Anderson, K. W. Bannister, E. Bastholm, R. Beresford, D. C.-J. Bock, R. Bolton, J. M. Chapman, K. Chow, J. D. Collier, F. R. Cooray, T. J. Cornwell, P. J. Diamond, P. G. Edwards, I. J. Feain, T. M. O. Franzen, D. George, N. Gupta, G. A. Hampson, L. Harvey-Smith, D. B. Hayman,

I. Heywood, C. Jacka, C. A. Jackson, S. Jackson, K. Jeganathan, S. Johnston, M. Kesteven, D. Kleiner, B. S. Koribalski, K. Lee-Waddell, E. Lenc, E. S. Lensson, S. Mackay, E. K. Mahony, N. M. McClure-Griffiths, R. McConigley, P. Mirtschin, A. K. Ng, R. P. Norris, S. E. Pearce, C. Phillips, M. A. Pilawa, W. Raja, J. E. Reynolds, P. Roberts, D. N. Roxby, E. M. Sadler, M. Shields, A. E. T. Schinckel, P. Serra, R. D. Shaw, T. Sweetnam, E. R. Troup, A. Tzioumis, M. A. Voronkov, and T. Westmeier. Australian square kilometre array pathfinder: I. system description. *Publications of the Astronomical Society of Australia*, 38, 2021.

[73] M. Huertas-Company, J. R. Primack, A. Dekel, D. C. Koo, S. Lapiner, D. Ceverino, R. C. Simons, G. F. Snyder, M. Bernardi, Z. Chen, H. Domínguez-Sánchez, C. T. Lee, B. Margalef-Bentabol, and D. Tuccillo. Deep learning identifies high-z galaxies in a central blue nugget phase in a characteristic mass range. *The Astrophysical Journal*, 858(2):114, may 2018.

[74] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2004.

[75] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[76] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[77] Mikaël Jacquemont, Thomas Vuillaume, Alexandre Benoit, Gilles Maurin, and Patrick Lambert. Deep learning for astrophysics, understanding the impact of attention on variability induced by parameter initialization. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 174–188, Cham, 2021. Springer International Publishing.

[78] Peng Jia, Xiyu Li, Zhengyang Li, Weinan Wang, and Dongmei Cai. Point spread function modelling for wide-field small-aperture telescopes with a denoising autoencoder. *Monthly Notices of the Royal Astronomical Society*, 493(1):651–660, 02 2020.

[79] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, September 2017.

[80] H. Junklewitz, M. R. Bell, M. Selig, and T. A. Enßlin. RESOLVE: A new algorithm for aperture synthesis imaging of extended emission in radio astronomy. *Astronomy & Astrophysics*, 586:A76, January 2016.

[81] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[82] M Kovačević, G Chiaro, S Cutini, and G Tosti. Classification of blazar candidates of uncertain type from the fermi LAT 8-yr source catalogue with an artificial neural network. *Monthly Notices of the Royal Astronomical Society*, 493(2):1926–1935, February 2020.

[83] Kazuki Kuramochi, Kazunori Akiyama, Shiro Ikeda, Fumie Tazaki, Vincent L. Fish, Hung-Yi Pu, Keiichi Asada, and Mareki Honma. Superresolution interferometric imaging with sparse modeling using total squared variation: Application to imaging the black hole shadow. *The Astrophysical Journal*, 858(1):56, may 2018.

[84] François Lanusse, Rachel Mandelbaum, Siamak Ravanbakhsh, Chun-Liang Li, Peter Freeman, and Barnabás Póczos. Deep generative models for galaxy image simulations. *Monthly Notices of the Royal Astronomical Society*, 504(4):5543–5555, May 2021.

[85] Francois Lanusse, Peter Melchior, and Fred Moolekamp. Hybrid physical-deep learning model for astronomical inverse problems, 2019.

[86] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[87] Chien-Hsiu Lee. Exoplanets: Past, present, and future. *Galaxies*, 6(2):51, April 2018.

[88] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[89] Di Li, Pei Wang, Lei Qian, Marko Krco, Alex Dunning, Peng Jiang, Youling Yue, Chenjin Jin, Yan Zhu, Zhichen Pan, and Rendong Nan. FAST in space: Considerations for a multibeam, multipurpose survey using china's 500-m aperture spherical radio telescope (FAST). *IEEE Microwave Magazine*, 19(3):112–119, May 2018.

[90] Rui Li, Nicola R. Napolitano, Haicheng Feng, Ran Li, Valeria Amaro, Linghua Xie, Crescenzo Tortora, Maciej Bilicki, Massimo Brescia, Stefano Cavuoti, and Mario Radovich. Galaxy morphoto-z with neural networks (GaZNets). *Astronomy & Astrophysics*, 666:A85, October 2022.

[91] Weitian Li, Haiguang Xu, Zhixian Ma, Ruimin Zhu, Dan Hu, Zhenghao Zhu, Junhua Gu, Chenxi Shan, Jie Zhu, and Xiang-Ping Wu. Separating the EoR signal with a convolutional denoising autoencoder: a deep-learning-based method. *Monthly Notices of the Royal Astronomical Society*, 485(2):2628–2637, 02 2019.

[92] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3d convolutional networks: Brain parcellation as a pretext task. In *Lecture Notes in Computer Science*, pages 348–360. Springer International Publishing, 2017.

[93] Sheng-Chieh Lin, Yuanyuan Su, Gongbo Liang, Yuanyuan Zhang, Nathan Jacobs, and Yu Zhang. Estimating cluster masses from SDSS multiband images with transfer learning. *Monthly Notices of the Royal Astronomical Society*, 512(3):3885–3894, March 2022.

[94] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[95] Federica Loiacono, Roberto Decarli, Carlotta Gruppioni, Margherita Talia, Andrea Cimatti, Gianni Zamorani, Francesca Pozzi, Lin Yan, Brian C. Lemaux, Dominik A. Riechers, Olivier Le Fèvre, Matthieu Bèthermin, Peter Capak, Paolo Cassata, Andreas Faisst, Daniel Schaerer, John D. Silverman, Sandro Bardelli, Médéric Boquien, Sandra Burkutean, Miroslava Dessauges-Zavadsky, Yoshinobu Fudamoto, Seiji Fujimoto, Michele Ginolfi, Nimish P. Hathi, Gareth C. Jones, Yana Khusanova, Anton M. Koekemoer, Guilaine Lagache, Lori M. Lubin, Marcella Massardi, Pascal Oesch, Michael Romano, Livia Vallini, Daniela Vergani, and Elena Zucca. The ALPINE–ALMA [c II] survey. *Astronomy & Astrophysics*, 646:A76, February 2021.

[96] N. R. Lomb. Least-Squares Frequency Analysis of Unequally Spaced Data. *Astrophysics and Space Science*, 39(2):447–462, February 1976.

[97] Giuseppe Longo, Erzsébet Merényi, and Peter Tiňo. Foreword to the focus issue on machine intelligence in astronomy and astrophysics. *Publications of the Astronomical Society of the Pacific*, 131(1004):1–6, 2019.

[98] Vesna Lukic, Francesco de Gasperin, and Marcus Brüggen. ConvoSource: Radio-astronomical source-finding with convolutional neural networks. *Galaxies*, 8(1):3, December 2019.

[99] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Copyright Cambridge University Press, 2003.

[100] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.

[101] Fabien Malbet, Céline Boehm, Alberto Krone-Martins, Antonio Amorim, Guillem Anglada-Escudé, Alexis Brandeker, Frédéric Courbin, Torsten Enßlin, Antonio Falcão, Katherine Freese, Berry Holl, Lucas Labadie, Alain Léger, Gary A. Mamon, Barbara McArthur, Alcione Mora, Mike Shao, Alessandro Sozzetti, Douglas Spolyar, Eva Villaver, Ummi Abbas, Conrado Albertus, João Alves, Rory Barnes, Aldo Stefano Bonomo, Hervé Bouy, Warren R. Brown, Vitor Cardoso, Marco Castellani, Laurent Chemin, Hamish Clark, Alexandre C. M. Correia, Mariateresa Crosta, Antoine Crouzier, Mario Damasso, Jeremy Darling, Melvyn B. Davies, Antonaldo Diaferio, Morgane Fortin, Malcolm Fridlund, Mario Gai, Paulo Garcia, Oleg Gnedin, Ariel Goobar, Paulo Gordo, Renaud Goullioud, David Hall, Nigel Hambly, Diana Harrison, David Hobbs, Andrew Holland, Erik Høg, Carme Jordi, Sergei Klioner, Ariane Lançon, Jacques Laskar, Mario Lattanzi, Christophe Le Poncin-Lafitte, Xavier Luri, Daniel Michalik, André Moitinho de Almeida, Ana Mourão, Leonidas Moustakas, Neil J. Murray, Matthew Muterspaugh, Micaela Oertel, Luisa Ostorero, Jordi Portell, Jean-Pierre Prost, Andreas Quirrenbach, Jean Schneider, Pat Scott, Arnaud Siebert, Antonio da Silva, Manuel Silva, Philippe Thébault, John Tomsick, Wesley Traub, Miguel de Val-Borro, Monica Valluri, Nicholas A. Walton, Laura L. Watkins, Glenn White, Lukasz Wyrzykowski, Rosemary Wyse, and Yoshiyuki Yamada. Faint objects in motion: the new frontier of high precision astrometry. *Experimental Astronomy*, 51(3):845–886, June 2021.

[102] Berta Margalef-Bentabol, Marc Huertas-Company, Tom Charnock, Carla Margalef-Bentabol, Mariangela Bernardi, Yohan Dubois, Kate Storey-Fisher, and Lorenzo Zanisi. Detecting outliers in astronomical images with deep generative networks. *Monthly Notices of the Royal Astronomical Society*, 496(2):2346–2361, June 2020.

[103] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[104] Michel Mayor and Didier Queloz. A jupiter-mass companion to a solar-type star. *Nature*, 378(6555):355–359, November 1995.

[105] J. P McMullin, B. Waters, D. Shiebel, W. Young, and K. Golap. CASA Architecture and Applications. *Astronomical Data Analysis Software and Systems XVI ASP Conference Series, Vol. 376, proceedings of the conference held 15-18 October 2006 in Tucson, Arizona, USA. Edited by Richard A. Shaw, Frank Hill and David J. Bell., p.127*, 376, October 2007.

[106] Michael Mesarcik, Albert-Jan Boonstra, Christiaan Meijer, Walter Jansen, Elena Ranguelova, and Rob V van Nieuwpoort. Deep learning assisted data inspection for radio astronomy. *Monthly Notices of the Royal Astronomical Society*, 496(2):1517–1529, 05 2020.

[107] Niruj R. Mohan and David A. Rafferty. Pybdsf: Python blob detection and source finder. In *https://www.astron.nl/citt/pybdsf/*, 2015.

[108] Robert Morgan, Brian Nord, Keith Bechtol, SJ González, E Buckley-Geer, A Möller, JW Park, AG Kim, S Birrer, M Aguena, et al. Deepzipper: A novel deep-learning architecture for lensed supernovae identification. *The Astrophysical Journal*, 927(1):109, 2022.

[109] Warren R. Morningstar, Laurence Perreault Levasseur, Yashar D. Hezaveh, Roger Blandford, Phil Marshall, Patrick Putzky, Thomas D. Rueter, Risa Wechsler, and Max Welling. Data-driven reconstruction of gravitationally lensed galaxies using recurrent inference machines. *The Astrophysical Journal*, 883(1):14, September 2019.

[110] Ali Mousavi, Ankit B. Patel, and Richard G. Baraniuk. A deep learning approach to structured signal recovery. In *In Proceeding of 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015.

[111] Daniel Muthukrishna, Gautham Narayan, Kaisey S. Mandel, Rahul Biswas, and Renée Hložek. Rapid: Early classification of explosive transients using deep learning. *Publications of the Astronomical Society of the Pacific*, 131(1005):118002, sep 2019.

[112] Steffen Neutsch, Caroline Heneka, and Marcus Brüggen. Inferring astrophysics and dark matter properties from 21 cm tomography using deep learning. *Monthly Notices of the Royal Astronomical Society*, 511(3):3446–3462, 01 2022.

[113] Isadora Nun, Pavlos Protopapas, Brandon Sim, Ming Zhu, Rahul Dave, Nicolas Castro, and Karim Pichara. Fats: Feature analysis for time series, 2015.

[114] 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):606–619, 2014.

[115] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[116] K Pearson. On lines of closes fit to system of points in space, london, e dinb. *Dublin Philos. Mag. J. Sci*, 2:559–572, 1901.

[117] W. J. Pearson, L. Wang, J. W. Trayford, C. E. Petrillo, and F. F. S. van der Tak. Identifying galaxy mergers in observations and simulations with deep learning. *Astronomy & Astrophysics*, 626:49, June 2019.

[118] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[119] Meyer Z. Pesenson, Isaac Z. Pesenson, and Bruce McCollum. The data big bang and the expanding digital universe: High-dimensional, complex and massive data sets in an inflationary epoch. *Advances in Astronomy*, 2010:1–16, 2010.

[120] Óscar Pimentel, Pablo A. Estévez, and Francisco Förster. Deep attention-based supernovae classification of multiband light curves. *The Astronomical Journal*, 165(1):18, December 2022.

[121] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, oct 2020.

[122] P. Ranalli, D. Hobbs, and L. Lindegren. Astrometry and exoplanets in the gaia era: a bayesian approach to detection and parameter recovery. *AStronomy & Astrophysics*, 614:A30, June 2018.

[123] Oleksandra Razim, Stefano Cavuoti, Massimo Brescia, Giuseppe Riccio, Mara Salvato, and Giuseppe Longo. Improving the reliability of photometric redshift with machine learning. *Monthly Notices of the Royal Astronomical Society*, 507(4):5034–5052, August 2021.

[124] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[125] Andrew Resnick. Fourier optics and computational imaging, by kedar khare. *Contemporary Physics*, 58(1):102–103, 2017.

[126] Esteban Reyes and Pablo A. Estévez. Transformation based deep anomaly detection in astronomical images. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[127] S Rezaei, J P McKean, M Biehl, and A Javadpour. DECORAS: detection and characterization of radio-astronomical sources using deep learning. *Monthly Notices of the Royal Astronomical Society*, 510(4):5891–5907, December 2021.

[128] Alejandra Rocha-Solache, Iván Rodríguez-Montoya, David Sánchez-Argüelles, and Itziar Aretxaga. Time-domain deep-learning filtering of structured atmospheric noise for ground-based millimeter astronomy. *The Astrophysical Journal Supplement Series*, 260(1):15, 2022.

[129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[130] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart and James L. Mcclelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[131] A Vafaei Sadr, Etienne E Vos, Bruce A Bassett, Zafiirah Hosenie, N Oozeer, and Michelle Lochner. Deepsource: point source detection using deep learning. *Monthly Notices of the Royal Astronomical Society*, 484(2):2793–2806, February 2019.

[132] J. D. Scargle. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *Astrophys. J.*, 263:835–853, December 1982.

[133] K. Schmidt, F. Geyer, S. Fröse, P.-S. Blomenkamp, M. Brüggen, F. de Gasperin, D. Elsässer, and W. Rhode. Deep learning-based imaging in radio interferometry. *Astronomy & Astrophysics*, April 2022.

[134] Morgan A. Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngolè , David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, jan 2018.

[135] Paolo Serra, Tobias Westmeier, Nadine Giese, Russell Jurek, Lars Flöer, Attila Popping, Benjamin Winkel, Thijs van der Hulst, Martin Meyer, Bärbel S. Koribalski, Lister Staveley-Smith, and Hélène Courtois. Sofia: a flexible source finder for 3d spectral line data. *Monthly Notices of the Royal Astronomical Society*, 448(2):1922–1929, 02 2015.

[136] Michael Shao, Geoff Marcy, Joseph H. Catanzarite, Stephen J. Edberg, Alain Léger, Fabien Malbet, Didier Queloz, Matthew W. Muterspaugh, Charles A. Beichman, Debra A. Fischer, Eric B. Ford, Rob P. Olling, Shrinivas R. Kulkarni, Stephen C. Unwin, and Wesley A. Traub. Astrometric detection of earthlike planets. *arXiv: Earth and Planetary Astrophysics*, 2009.

[137] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[138] Michael J Smith and James E Geach. Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning. *Monthly Notices of the Royal Astronomical Society*, 490(4):4985–4990, 10 2019.

[139] Charles Soussen, Jérôme Idier, David Brie, and Junbo Duan. From Bernoulli-Gaussian deconvolution to sparse signal restoration. technical report, 34 pages, January 2010.

[140] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks, 2015.

[141] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *CoRR*, abs/1507.06228, 2015.

[142] Jean-Luc Starck, Jalal Fadili, and Fionn Murtagh. The undecimated wavelet decomposition and its reconstruction. *IEEE Transactions on Image Processing*, 16(2):297–309, 2007.

[143] Jean-Luc Starck and Fionn Murtagh. *Astronomical Image and Data Analysis*. Springer Berlin Heidelberg, 2002.

[144] STScI Development Team. pysynphot: Synthetic photometry software package, March 2013.

[145] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer International Publishing, 2017.

[146] Jesse Sun, Fatemeh Darbehani, Mark Zaidi, and Bo Wang. Saunet: Shape attentive u-net for interpretable medical image segmentation. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 797–806, Cham, 2020. Springer International Publishing.

[147] Sam F. Sweere, Ivan Valtchanov, Maggie Lieu, Antonia Vojtekova, Eva Verdugo, Maria Santos-Lleo, Florian Pacaud, Alexia Briassouli, and Daniel Cámpora Pérez. Deep learning-based super-resolution and denoising for xmm-newton images, 2022.

[148] THE CASA TEAM, Ben Bean, Sanjay Bhatnagar, Sandra Castro, Jennifer Donovan Meyer, Bjorn Emonts, Enrique Garcia, Robert Garwood, Kumar Golap, Justo Gonzalez Villalba, Pamela Harris, Yohei Hayashi, Josh Hoskins, Mingyu Hsieh, Preshanth Jagannathan, Wataru Kawasaki, Aard Keimpema, Mark Kettenis, Jorge Lopez, Joshua Marvil, Joseph Masters, Andrew McNichols, David Mehringer, Renaud Miel, George Moellenbrock, Federico Montesino, Takeshi Nakazato, Juergen Ott, Dirk Petry, Martin Pokorny, Ryan Raba, Urvashi Rau, Darrell Schiebel, Neal Schweighart, Srikrishna Sekhar, Kazuhiko Shimada, Des Small, Jan-Willem Steeb, Kanako Sugimoto, Ville Suoranta, Takahiro Tsutsumi, Ilse M. van Bemmel, Marjolein Verkouter, Akeem Wells, Wei Xiong, Arpad Szomoru, Morgan Griffith, Brian Glendenning, and Jeff Kern. Casa, the common astronomy software applications for radio astronomy. *arXiv*, 2022.

[149] The Theia Collaboration, Celine Boehm, Alberto Krone-Martins, Antonio Amorim, Guillem Anglada-Escude, Alexis Brandeker, Frederic Courbin, Torsten Ensslin, Antonio Falcao, Katherine Freese, Berry Holl, Lucas Labadie, Alain Leger, Fabien Malbet, Gary Mamon, Barbara McArthur, Alcione Mora, Michael Shao, Alessandro Sozzetti, Douglas Spolyar, Eva Villaver, Conrado Albertus, Stefano Bertone, Herve Bouy, Michael Boylan-Kolchin, Anthony Brown, Warren Brown, Vitor Cardoso, Laurent Chemin, Riccardo Claudi, Alexandre C. M. Correia, Mariateresa Crosta, Antoine Crouzier, Francis-Yan Cyr-Racine, Mario Damasso, Antonio da Silva, Melvyn Davies, Payel Das, Pratika Dayal, Miguel de Val-Borro, Antonaldo Diaferio, Adrienne Erickcek, Malcolm Fairbairn, Morgane Fortin, Malcolm

Fridlund, Paulo Garcia, Oleg Gnedin, Ariel Goobar, Paulo Gordo, Renaud Goullioud, Nigel Hambly, Nathan Hara, David Hobbs, Erik Hog, Andrew Holland, Rodrigo Ibata, Carme Jordi, Sergei Klioner, Sergei Kopeikin, Thomas Lacroix, Jacques Laskar, Christophe Le Poncin-Lafitte, Xavier Luri, Subhabrata Majumdar, Valeri Makarov, Richard Massey, Bertrand Mennesson, Daniel Michalik, Andre Moitinho de Almeida, Ana Mourao, Leonidas Moustakas, Neil Murray, Matthew Muterspaugh, Micaela Oertel, Luisa Ostorero, Angeles Perez-Garcia, Imants Platais, Jordi Portell i de Mora, Andreas Quirrenbach, Lisa Randall, Justin Read, Eniko Regos, Barnes Rory, Krzysztof Rybicki, Pat Scott, Jean Schneider, Jakub Scholtz, Arnaud Siebert, Ismael Tereno, John Tomsick, Wesley Traub, Monica Valluri, Matt Walker, Nicholas Walton, Laura Watkins, Glenn White, Dafydd Wyn Evans, Lukasz Wyrzykowski, and Rosemary Wyse. Theia: Faint objects in motion or the new astrometry frontier, 2017.

[150] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.

[151] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5435–5443, 2017.

[152] Peter G. Tuthill, Eduardo Bendek, Olivier Guyon, Bryn Jeffries, Nemanja Jovanovic, Pete Klupar, Kieran Larkin, Barnaby Norris, Ben Pope, Michael Shao, and Anthony Horton. The TOLIMAN space telescope. In Antoine Mérand, Michelle J. Creech-Eakman, and Peter G. Tuthill, editors, *Optical and Infrared Interferometry and Imaging VI*. SPIE, July 2018.

[153] Stephen C. Unwin, Michael Shao, Angelle M. Tanner, Ronald J. Allen, Charles A. Beichman, David Boboltz, Joseph H. Catanzarite, Brian C. Chaboyer, David R. Ciardi, Stephen J. Edberg, Alan L. Fey, Debra A. Fischer, Christopher R. Gelino, Andrew P. Gould, Carl Grillmair, Todd J. Henry, Kathryn V. Johnston, Kenneth J. Johnston, Dayton L. Jones, Shrinivas R. Kulkarni, Nicholas M. Law, Steven R. Majewski, Valeri V. Makarov, Geoffrey W. Marcy, David L. Meier, Rob P. Olling, Xiaopei Pan, Richard J. Patterson, Jo Eliza Pitesky, Andreas Quirrenbach, Stuart B. Shaklan, Edward J. Shaya, Louis E. Strigari, John A. Tomsick, Ann E. Wehrle, and Guy Worthey. Taking the measure of the universe: Precision astrometry with sim planetquest. *Publications of the Astronomical Society of the Pacific*, 120(863):38–88, January 2008.

[154] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016. PMLR.

[155] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[156] Jacob T. VanderPlas. Understanding the Lomb-Scargle Periodogram. *The Astrophysical Journal Supplements Series*, 236(1):16, May 2018.

[157] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[158] Cédric Vonesch, Thierry Blu, and Michael Unser. Generalized daubechies wavelet families. *IEEE Trans. Signal Process.*, 55(9):4415–4429, 2007.

[159] T Westmeier, S Kitaeff, D Pallot, P Serra, J M van der Hulst, R J Jurek, A Elagali, B-Q For, D Kleiner, B S Koribalski, K Lee-Waddell, J R Mould, T N Reynolds, J Rhee, and L Staveley-Smith. Sofia-2: an automated, parallel hi source finding pipeline for the wallaby survey. *Monthly Notices of the Royal Astronomical Society*, 506(3):3962–3976, July 2021.

[160] T Westmeier, S Kitaeff, D Pallot, P Serra, J M van der Hulst, R J Jurek, A Elagali, B-Q For, D Kleiner, B S Koribalski, K Lee-Waddell, J R Mould, T N Reynolds, J Rhee, and L Staveley-Smith. sofia-2 an automated, parallel hi source finding pipeline for the WALLABY survey. *Monthly Notices of the Royal Astronomical Society*, 506(3):3962–3976, July 2021.

[161] Zbigniew Wojna, Vittorio Ferrari, Sergio Guadarrama, Nathan Silberman, Liang-Chieh Chen, Alireza Fathi, and Jasper R. R. Uijlings. The devil is in the decoder. *CoRR*, abs/1707.05847, 2017.

[162] Long Xu, Yihua Yan, and Xin Huang. *Deep Learning in Solar Astronomy*. Springer Nature Singapore, 2022.

[163] Jiancheng Yang, Xiaoyang Huang, Yi He, Jingwei Xu, Canqian Yang, Guozheng Xu, and Bingbing Ni. Reinventing 2d convolutions for 3d images. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3009–3018, 2021.

[164] Zhenping Yi, Jia Li, Wei Du, Meng Liu, Zengxu Liang, Yongguang Xing, Jingchang Pan, Yude Bu, Xiaoming Kong, and Hong Wu. Automatic detection of low surface brightness galaxies from sloan digital sky survey images. *Monthly Notices of the Royal Astronomical Society*, 513(3):3972–3981, March 2022.

[165] Andy B. Yoo, Morris A. Jette, and Mark Grondona. Slurm: Simple linux utility for resource management. In *Job Scheduling Strategies for Parallel Processing*, pages 44–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[166] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution, 2018.

[167] Ivan Zelinka, Massimo Brescia, and Dalya Baron, editors. *Intelligent Astrophysics*. Springer International Publishing, 2021.

[168] Qingguo Zeng, Xiangru Li, and Haitao Lin. Concat convolutional neural network for pulsar candidate selection. *Monthly Notices of the Royal Astronomical Society*, 494(3):3110–3119, April 2020.

[169] Hongyang Zhao, Jing Jin, Yi Liu, Yi Shen, and Yu Jiang. AdaBoost-MICNN: a new network framework for pulsar candidate selection. *Monthly Notices of the Royal Astronomical Society*, 513(2):2869–2883, 03 2022.

[170] Junbo Jake Zhao, Michaël Mathieu, Ross Goroshin, and Yann LeCun. Stacked what-where auto-encoders. *CoRR*, abs/1506.02351, 2015.

Author's Publications

1. M. Delli Veneri, L. Desdoigts, M. A. Schmitz, A. Krone-Martins, E. E. O. Ishida, P. Tuthill, R. S. De Souza, R. Scalzo, M. Brescia, G. Longo, A. Picariello, Periodic Astrometric Signal Recovery Through Convolutional Autoencoders, Intelligent Astrophysics pp 167 - 195, part of the Emergence, Complexity and Computation book series,
https://link.springer.com/chapter/10.1007/978-3-030-65867-0_8;

2. M. Delli Veneri, S. Cavuoti, R. Abbruzzese, M. Brescia, G. Sperli', V. Moscato, G. Longo, HyCASTLE: A Hybrid ClAssification System based on Typicality, Labels and Entropy, Knowledge-Based Systems, Volume 244, 2022,
https://www.sciencedirect.com/science/article/pii/S0950705122002507?via=ihub;

3. M. Delli Veneri, L. Tychoniec, F. Guglielmetti, G. Longo, E. Villard, 3D Detection and characterisation of ALMA sources through Deep Learning, Monthly Notices of the Royal Astronomical Society, Accepted on 12/11/2022,
https://doi.org/10.1093/mnras/stac3314;

4. V. Amaro, S. Cavuoti, M. Brescia, G. Riccio, C. Tortora, M. D'Addona, M. Delli Veneri, N. R. Napolitano, M. Radovich, G. Longo, Rejection criteria based on outliers in the KiDS photometric redshifts and PDF distributions derived by Machine Learning, Intelligent Astrophysics pp 245 - 265, part of the Emergence, Complexity and Computation book series,
https://link.springer.com/chapter/10.1007/978-3-030-65867-0_11

5. C. Donaddio, M. Brescia, A. Riccardo, G. Angora, M. Delli Veneri, G. Riccio, A novel approach to the classification of terrestrial drainage networks based on deep learning and preliminary results on solar system bodies, Sci-

entific Reports 11, Nature, a.n. 5875, 2021,
https://www.nature.com/articles/s41598-021-85254-x;

6. C. Patruno, G. Fabbrocini, G. Longo, G. Argenziano, S. M. Ferrucci, L. Stringendi, K. Peris, M. Ortoncelli, A. Offidani, G. F. Amoruso, M. Talamonti, G. Girolomoni, T. Grieco, M. Iannone, E. Nettis, C. Foti, F. Rongieletti, M. Corazza, M. Delli Veneri, M. Napoliatano, DADE Study Group, Effective and safety of long-term dupilumab treatment in elderly patients with atopic dermatitis: a multicenter real-life observational study, American Journal of Clinical Dermatology, 22, 481 - 586, 2021
https://link.springer.com/article/10.1007/s40257-021-00597-5;

7. F. Guglielmetti, P.Arras, M. Delli Veneri, T. Enßlin, G. Longo, Ł. Tychoniec, E. Villard, Bayesian and Machine Learning Methods in the Big Data era for astronomical imaging, proceedings International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, IHP, Paris, July 18-22, 2022,
https://doi.org/10.48550/arXiv.2210.01444;

8. M. Delli Veneri, R. S. De Souza, A. Krone-MArtins, E. E. O. Ishida, M. L. L. Dantas, N. Kennamer, COIN Collaboration, How have astronomers cited other fields in the last decade?, Research Notes of the American Astronomical Society, Volume 6, Number 6, June 2022, https://iopscience.iop.org/article/10.3847/2515-5172/ac74c7.