CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

Strong Lensing Inverse Modeling using Neural Posterior Estimation

Vitor Souza Ramos

Orientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2025

Vitor Souza Ramos

Strong Lensing Inverse Modeling using Neural Posterior Estimation

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física.

CBPF – Centro Brasileiro de Pesquisas Físicas

Supervisor: Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2025

Souza Ramos, Vitor
Strong Lensing Inverse Modeling using Neural Posterior Estimation/ Vitor Souza Ramos. - 2025
64 f. : il.

Dissertação de Mestrado – CBPF – Centro Brasileiro de Pesquisas Físicas ,Rio de Janeiro, RJ, 2025.
Supervisor: Dr. Clécio Roque De Bom

1. Lentes Gravitacionais Fortes 2. Simulation-Based Inference 3. Inteligência Artificial 4. Astrofísica Extragaláctica
CDU 02:141:005.7

"STRONG LENSING INVERSE MODELING USING NEURAL POSTERIOR ESTIMATION"

VITOR SOUZA RAMOS

Dissertação de Mestrado em Física apresentada no Centro Brasileiro de Pesquisas Físicas do Ministério da Ciência Tecnologia e Inovação. Fazendo parte da banca examinadora os seguintes professores:

Documento assinado digitalmente
gov.br
CLECIO ROQUE DE BOM
Data: 22/10/2025 17:07:07-0300
Verifique em https://validar.iti.gov.br

Clécio Roque de Bom - Orientador/CBPF

Documento assinado digitalmente
gov.br
CRISTINA FURLANETTO
Data: 26/06/2025 14:07:48-0300
Verifique em https://validar.iti.gov.br

Cristina Furlanetto - UFRGS

Laerte Sodré Junior – IAG/USP

Rio de Janeiro, 05 de junho de 2025.

Dedico este trabalho à memória de Simone.
Obrigado por tudo.

Acknowledgements

Em primeiro lugar, gostaria de agradecer a meus pais, Renato e Simone, e a meus irmãos, Pedro e Thais, pelo apoio constante. Agradeço também a Jayana, por estar sempre ao meu lado, mesmo com toda a distância. Sem vocês, este caminho teria sido muito mais difícil de percorrer.

Agradeço também ao meu orientador, Clécio De Bom, por acreditar na ciência feita no Brasil e por me incentivar a fazer um trabalho de qualidade. Não posso deixar de mencionar também sua paciência, que foi testada por quase um ano a mais do que o esperado. Agradeço também aos meus colegas de grupo, Gabriel, Phelipe, André, Juan, Viviane e Bernardo, pelas discussões, pelos almoços na Unirio e por todos os cafés que eu não tomei com vocês.

Não posso deixar de agradecer também a Paula e Francisco, por todas as noites assistindo filmes de terror e todos os jogos de sinuca.

Por fim, agradeço ao CBPF e ao CNPq pela oportunidade e pelo apoio para que eu pudesse desenvolver este trabalho.

"[Se] alguém condenado à morte diz ou pensa
que se ele tivesse que viver em alguma rocha alta,
em uma saliência tão estreita que só teria espaço para ficar de pé,
e o oceano, escuridão eterna, solidão eterna, tempestade eterna ao seu redor,
se tivesse que permanecer de pé em um metro quadrado de espaço toda a sua vida,
mil anos, eternidade, seria melhor viver assim do que morrer de uma vez!
Viver e viver! Vida, seja ela qual for.
O homem é uma criatura vil. E vil é aquele que o chama de vil por isso."

— Adaptado de Fiódor Dostoiévski em *Crime e Castigo*

Resumo

Com novas e futuras pesquisas esperando uma quantidade sem precedentes de dados de lentes gravitacionais fortes, o desenvolvimento de métodos de modelagem rápidos e automatizados que sejam capazes de prever incertezas confiáveis é crucial para a ciência derivada desses fenômenos. Recentemente, métodos baseados em Redes Neurais têm atraido muito interesse devido a alta performance e alta aplicabilidade em análise de grande volume de dados. Neste trabalho, relatamos a aplicação da Neural Posterior Estimation, um método de Simulation-Based Inference para inferir o raio de Einstein, dispersão de velocidade da galáxia de lente e redshifts de ambas as galáxias lente e fonte em imagens simuladas realistas baseadas na DEcam. Também é analizada a confiabilidade dos posteriores gerados pelos modelos. Nós discutimos algumas das limitações inerentes ao uso de inteligência artificial como asubstituto aos métodos tradicionais de modelagem, e aplicamos o modelo a conjuntos de dados reais para avaliar sua adequação em situações de mundo real. Nossos resultados sugerem que a Simulation-Based Inference é uma técnica promissora para realizar modelagem inversa em Lentes Gravitacionais, e pode se mostrar uma ferramenta essencial para futuras investigações nesta área.

Key-Words: Lentes Gravitacionais fortes, Simulation Based Inference, Inteligência Artificial, Astrofísica Extragaláctica.

Abstract

With new and future surveys expecting an unprecedented amount of strong lensing data, the development of fast and automated modeling methods that are able to predict reliable uncertainties is crucial to the science derived from these phenomena. Recently, methods based on Neural Networks have been raising a lot of interest due to their performance and suitability when analyzing large volumes of data. In this work, we report on the application of Neural Posterior Estimation, a Simulation-Based Inference method to infer Einstein radius, lens galaxy velocity dispersion, and redshift of both lens and source objects on realistic wide-field DECam-based simulated griz-band images. We also analyze the reliability of the posteriors generated by the models. We discuss some of the caveats involved in using Artificial Intelligence to replace traditional modeling and apply our model to real datasets to gauge their suitability in real-world scenarios. Our results suggest that Simulation-Based Inference is a promising technique for performing Bayesian inference in astrophysics and cosmology and could prove an essential tool for future investigations in this field.

Key-Words: Strong Gravitational Lensing, Bayesian Inference, Simulation-Based Inference Artificial Intelligence.

List of Figures

List of Tables

List of abbreviations and acronyms

CBPF	Centro Brasileiro de Pesquisas Físicas
DES	Dark Energy Survey
LSST	Legacy Survey of Space and Time
CCD	Charge-Coupled Device
SIS	Singular Isothermal Sphere
SIE	Singular Isothermal Ellipsoid
AI	Artificial intelligence
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
LLM	Large Language Model
MSE	Mean Squared Error
SGD	Stochastic Gradient Descent
CNN	Convolutional Neural Network
GPU	Graphical Processing Unit
BNN	Bayesian Neural Network
SBI	Simulation-Based Inference
NPE	Neural Posterior Estimation
MCMC	Markov Chain Monte Carlo

NF	Normalizing Flow
NSF	Neural Spline Flow
DELVE	DECam Local Volume Exploration
DECam	Dark Energy Camera
ADU	Analog-to-Digital Unit
PSF	Point spread function
ReLU	Rectified Linear Unit
C2ST	Classifier Two-Sample Test
KS	Kolmogorov-Smirnov
LaStBeRu	Last Stand Before Rubin

Contents

Chapter 1

Introduction

Strong gravitational lensing is a phenomenon predicted by General Relativity, in which the deformation in space-time caused by a massive object such as a galaxy alters the paths of light rays emitted by a more distant source that is sufficiently aligned to the line of sight of the massive object [1, 2, 3], causing the appearance of multiple images of the more distant source. The lensing effect may also cause images to appear magnified and distorted.

Strong lensing has been successfully employed to study many different astrophysical phenomena. These systems can be used to probe mass distributions in galaxies, providing valuable insights into the physics of dark matter [4, 5, 6, 7, 8]. The effect of strong lensing in multiply-imaged time-varying sources such as quasars and supernovae has been used to measure the expansion rate of the universe (known as the Hubble constant) [9, 10, 11], contributing to a solution to the so-called "Hubble tension", a discrepancy in the measured values of this constant. Furthermore, since lensing phenomena conserve surface brightness, a magnification effect may be induced, causing these systems to act as "gravitational telescopes", aiding in the study of very distant source objects [12, 13].

Currently, the number of confirmed and modeled lenses lies in the hundreds, preventing robust statistical analyses that depend on strong lensing data. In the coming years, however, ongoing and future astronomical surveys, such as the Dark Energy Survey (DES) [14], the Vera Rubin Observatory Legacy Survey of Space and Time (LSST) [15], and Euclid [16], featuring closer to ideal observing conditions for the detection of strong lenses, are expected to increase this number by up to three orders of magnitude [17]. This volume of data will require fast and automated data analysis methods, marking a departure from the traditional maximum likelihood approach that would quickly become prohibitively time- and effort-intensive. In recent years, methods based on Convolutional Neural Networks have achieved remarkable results in parameter inference tasks, though with limited success in uncertainty estimation [18, 19], and given its significant role in

the applications of strong lensing in astrophysics and cosmology, a method that offers a Bayesian approach to parameter inference could prove advantageous to the science derived from strong lensing data.

In this work, we propose using Normalizing Flows [20, 21], a class of Neural Network-based density estimators, to perform Neural Posterior Estimation [22], a Simulation-Based Inference method, of strong gravitational lensing parameters. Although similar methods have been applied to this problem before [23, 24, 25], this work targets four parameters relevant to the Singular Isothermal Ellipsoid lens model, namely the Einstein Radius (θ_E), the velocity dispersion of the lens galaxy (σ_v), as well as the redshifts of both lens (z_l) and source (z_s) galaxies, and focuses on ground-based data by using DECam-like simulated strong lens images. We aim to approach the problem under a Bayesian framework to prioritize uncertainty estimation. The chosen method is computationally efficient and not time-intensive, proving suitable for the volume of data expected in the near future.

This work is divided as follows: In Chapter 2, we provide historical context and a brief mathematical introduction to strong gravitational lensing, discussing some of its applications. In Chapter 3, we focus on deep learning, addressing how modern techniques such as normalizing flows can be applied to perform Bayesian inference, highlighting their advantages over traditional methods. In Chapter 4, we detail the methodology used in this work. By combining the information in previous chapters, we propose an algorithm to perform fast and automated analysis of strong lensing systems. We discuss the technical aspects of the work, detailing the simulations as well as the deep learning algorithm used in this work. Finally, in Chapters 5 and 6, we evaluate the performance of our proposed method using different metrics. We also discuss some of the caveats in our results.

Chapter 2

Strong Gravitational Lensing

The idea of an interaction between light and gravity was first speculated by Sir Isaac Newton in 1704 in his book *Opticks* [26], although the first attempt at computing the deflection angle of light by a massive body was only carried out in 1783 by John Mitchell [27, 3, 1], prompted by correspondence with Henry Cavendish, assuming a corpuscular nature of light in Newtonian gravity. Notably, at about the same time, Pierre-Simon Laplace also speculated that the deflection of light due to gravity could be so large that light would not be able to flow out of it, an early hint at the idea of a black hole.

Shortly after these initial calculations, in 1801 Johann Von Soldner used Newtonian mechanics to estimate the deflection angle for light particles close to the surface of the Sun, arriving at a value of $\alpha = 0.83''$. Soldner's result did not lead to much discussion at the time, and in fact, even when Albert Einstein arrived at a very similar result using his recently developed theory of Special Relativity, the result was still met with indifference by the community. Although there was talk of observational confirmation of these results during a solar eclipse in 1914, the proximity of the predicted value to the limit of observational precision caused the community to meet this prospect with indifference.

It took another eight years and the advent of General Relativity to generate interest in this phenomenon. Einstein's new theory changed the prediction to approximately double the original value, now well within the limits of observational precision. The prediction was to be tested during the 1919 solar eclipse, with groups observing the phenomenon from the Island of Principe, in northern Africa, and in the city of Sobral, in the state of Ceará, Northern Brazil. In accordance with Einstein's theory, the teams obtained a value of $1.61'' \pm 0.30''$ [27], marking the first experimental confirmation of GR and ushering a new post-newtonian era in astrophysics.

This confirmation led to an increase in interest, and prompted the theoretical prediction of the appearance of multiple images of a lensed object. In fact, for systems where

the source object is perfectly aligned with the lens, one would expect to see a ring (later denoted an "Einstein Ring") around the lens, though Einstein himself believed the probability of observing this phenomenon would be very low due to the small separation between images caused by masses of stars.

In 1937, Fritz Zwicky was the first to propose that galaxies could act as lenses [28], and that given their much higher mass, the separation between images would be more resolvable and thus, the probability of observation would be much higher [29]. Zwicky also noted that since the lensing phenomenon conserves surface brightness, the magnification effect would allow the observation of distant object that would otherwise be too faint to observe. Despite the new outlook proposed by Zwicky, the first observation of a multiply imaged system would only happen in the late 70s, with the advent of Charge-Coupled Devices (CCDs) replacing photographic films playing a significant role in the confirmation of the lensing phenomenon. In the following decades, the interest in gravitational lensing increased, prompting the discovery of many of its applications in astrophysics and cosmology.

2.1 The Lens Equation

Assuming a weak-field regime, Einstein's GR predicts that the deflection angle for a light particle due to a singular point mass would be given by

$$\hat{\alpha} = \frac{4GM}{c^2\xi}, \tag{2.1}$$

where G is the gravitational constant, c is the speed of light in a vacuum, M is the mass of the point particle. The impact parameter ξ denotes the lowest distance between the incoming light ray's trajectory and the center of mass of the object. This equation is valid for an impact parameter much larger than the Swarzschild radius $R_S \equiv 2GM/c^2$. The condition for a deflected ray to reach an observer O can be derived from Euclidean geometry (see Fig. 2.1), noting that the small angle approximation $\sin(\theta) \approx \theta$ is valid for all angles involved:

$$\hat{\beta} D_{OS} = \hat{\theta} D_{OS} - \hat{\alpha} D_{LS}, \tag{2.2}$$

known as the Lens Equation. By introducing the reduced deflection angle

$$\hat{\alpha'} = \frac{D_{LS}}{D_{OS}} \hat{\alpha}, \tag{2.3}$$

Eq. 2.2 can be written as

$$\hat{\beta} = \hat{\theta} - \hat{\alpha'}. \tag{2.4}$$

This equation can be used to constrain the position of the lensed image given the other parameters of the scheme. Combining Eqs. 2.1 and 2.4, and using that $\hat{\xi} \approx D_{OL}\hat{\theta}$,

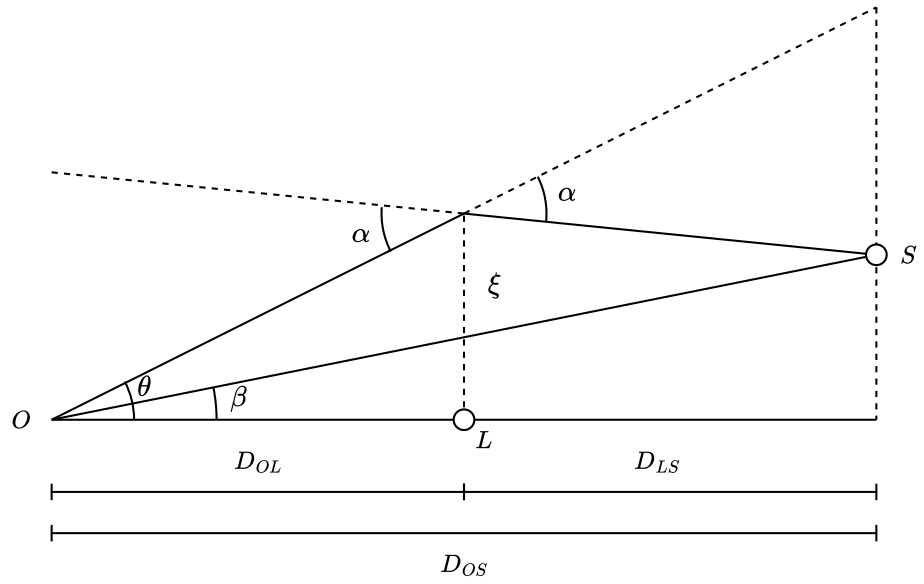

Figure 2.1 – Diagram of the Lens Equation.

we have

$$\hat{\beta} = \hat{\theta} - \left(\frac{4GM}{c^2}\frac{D_{LS}}{D_{OS}D_{OL}}\right)\frac{1}{\hat{\theta}}. \tag{2.5}$$

We can then define the Einstein Angle for a point-particle deflector as

$$\theta_E \equiv \sqrt{\frac{4GM}{c^2}\frac{D_{LS}}{D_{OS}D_{OL}}}. \tag{2.6}$$

By defining these angles in units of radians, the Einstein Angle (θ_E) remains an angular measurement, which is commonly used in observational studies. While θ_E is sometimes referred to as the "Einstein Radius" in observational contexts, it should not be confused with the physical Einstein Radius ($R_E = D_{OL}\theta_E$), which has units of length. For the remainder of this work, we will use "Einstein Radius" to refer to the angular measurement, following the convention in observational contexts.

Finally, this value can be applied to 2.4, yielding

$$\hat{\beta} = \hat{\theta} - \frac{\theta_E^2}{\hat{\theta}}. \tag{2.7}$$

When the lens and source objects are perfectly aligned ($\beta = 0$), given the symmetry of the problem, the source image appears as a ring to the observer, with radius θ_E.

2.1.1 Lensing by Extended Sources

In order to model the lensing effect caused by galaxies, instead of a point mass, one may consider a mass distribution. One of the simplest parametrizations of mass distributions in galaxies is the Singular Isothermal Sphere (SIS), given by

$$\rho(r) = \frac{\sigma_v^2}{2\pi G r^2}. \tag{2.8}$$

This model describes a collection of self-gravitating particles with velocities following a Maxwellian distribution, making the system behave like an isothermal gas. The SIS model presents two key issues: the central density is infinite (which is why it is called "singular"), and the density does not converge as the radius approaches infinity. The first issue can be addressed by introducing a core with a finite density within a certain radius, making the model more physically realistic. The second issue does not significantly affect lensing phenomena, as lensing is dominated by the mass at smaller radii ([3]).

By integrating the mass distribution along the line of sight, we obtain the surface density

$$\Sigma(\xi) = \frac{\sigma_v^2}{2G\xi}, \tag{2.9}$$

and the deflection angle is calculated using the relation for an arbitrary mass surface density in an axially symmetric mass distribution (see [30] for a more detailed explanation)

$$\hat{\alpha}(\xi) = \frac{4G}{c^2}\frac{2\pi\int_0^\xi \Sigma(\xi')\xi' d\xi'}{\xi}, \tag{2.10}$$

which yields

$$\hat{\alpha}(\theta) = \frac{4\pi\sigma_v^2}{c^2}. \tag{2.11}$$

The lens equation now reads

$$\hat{\beta} = \hat{\theta} - \left(\frac{4\pi\sigma_v^2}{c^2}\frac{D_{LS}}{D_{OS}}\right), \tag{2.12}$$

and the Einstein radius is then defined as

$$\theta_E \equiv 4\pi\frac{\sigma_v^2}{c^2}\frac{D_{LS}}{D_{OS}}. \tag{2.13}$$

As an alternative to the SIS model, one can opt for a slightly more realistic Singular Isothermal Ellipsoid (SIE) model [31, 32]. In this scenario, the density profile is given by

$$\rho(\tilde{r}) = \frac{\sigma_v^2}{2\pi G\tilde{r}^2}, \tag{2.14}$$

where $\tilde{r}$ is the elliptical distance from the center, given by $\tilde{r}^2 = x^2/q + qy^2$, with q representing the ratio between the axes, also know as the flattening of the ellipse. The Einstein radius for this density profile is also given by Equation 2.13.

The density profiles introduced in this chapter have two major characteristics. They present a divergence at the center and have infinite total mass as $r \to \infty$. Despite these peculiarities, they have been shown to be good approximations to both early- [33] and late-type galaxies [3], with the SIE model particularly appropriate for the modeling of early types, which account for the majority of lenses in strong lensing systems (see [34]).

Thus, the position of the lensed image (given by the Einstein Radius) is determined by three variables: the velocity dispersion of the lens galaxy, the distance between lens

and source and the distance between observer and source, which can be represented in observational measurements by their redshifts through the angular diameter distance and assuming a cosmological model. These parameters are traditionally measured from observations using spectroscopical information for the redshifts and the velocity dispersion combined with maximum-likelihood based methods for the Einstein Radius. While that method is a gold standard in parameter inference, new and future surveys are expected to detect an unprecedented number of SL systems, rendering traditional methods unsuitable. Thus, we investigate a method to perform fast and automated parameter inference using neural network-based methods.

Chapter 3

Deep Learning for Bayesian Inference

The field of artificial intelligence (AI) comprises a number of techniques intended to perform complex tasks. While the definition of intelligence is a matter of debate, it is generally agreed upon that machines that perform pattern recognition, natural language processing and many forms of decision making display some form of artificial intelligence.

Within the realm of AI methods, a subset that has garnered a lot of interest in the last few decades is Machine Learning (ML), a set of techniques that allow for training of a machine with the goal of performing a task. ML methods are defined by an inversion of the process of traditional programming. Instead of using a set of instructions to be applied to input data in order to arrive at a certain output, ML methods aim to learn the transformations that need to be applied to input data in order to arrive at a desired output. The methods that perform this task can use labeled data (that is, use both input and output data to learn the transformations), in which case they are referred to as Supervised Learning, or simply find patterns in input data, referred to as Unsupervised Learning.

Deep Learning (DL) [35, 36] is a special subset of ML methods characterized by the use of multi-layer neural networks (NNs), a class of models that act as universal function approximators. With the goal of mimicking the behavior of neurons in the human brain, NNs have been widely employed in recent years in a vast number of fields, showing great performance in many different tasks, such as classification and regression, with applications in Natural Language Processing, Computer Vision, and many other tasks that rely on some form of decision making.

The notion that machines could demonstrate intelligence was first posed in its modern form in 1950 by Alan Turing [37]. Since then, the field went through periods of high interest, with advancements such as the artificial neuron [38], introduced in 1943 by

McCullough and Pitts. This idea was later developed by Rosenblatt in 1958 [39], sparking great interest, only to be followed by the discovery of technical limitations [40], leading to periods of low interest. Such periods came to be known as as AI winters (see [41]).

Some important milestones in the history of AI include the introduction of the Perceptron [39], a precursor to modern day neural networks, based in the artificial neuron introduced by McCullough and Pitts in 1943, the successful employment of AI to recognize handwritten digits by Yan LeCun in 1990 [42], the big AI boom of the 2010s, where Deep Learning models surpassed human performance in image classification tasks, and more recently, the development of Large Language Models (LLMs) powered by the transformer architecture [43].

As has always been the case, technological advancements are fueled by scientific discovery, and in turn, these advancements are then used accelerate new scientific discovery. In the current era of Big Data in astronomy and cosmology, AI methods can prove a valuable, if not essential tool to analyze large datasets. The paradigm of AI fueling scientific discovery extends beyond astronomy, with its increasing importance for science being recognized with the 2024 Nobel prize, awarded to physicists John Hopfield and Geoffrey Hinton for their foundational work which paved the way for modern Machine Learning.

3.1 Neural Networks

Inspired by the behavior of neurons in the human brain, the artificial neuron is the building block of neural networks. A neuron works by taking a set of inputs, each with a respective weight associated to it, and its output is determined by an activation function acting on the linear combination of weights and inputs (along with a bias term). Neurons are combined in layers, with all neurons in a layer receiving the same set of inputs, and layers are stacked in order to form complete networks, where the outputs of the neurons in a layer are used as the set of inputs for the neurons in the subsequent layer, a pattern that is repeated all the way to the output layer of the network.

For a given input $X = \{x_1, x_2, ..., x_n\}$, an artificial neuron processes this data by taking a linear combinations of the input data with a set of weights $W = \{w_1, w_2, ..., w_n\}$, and adding a bias term b. The output of a neuron is the value of an activation function evaluated at the result of this sum. A visual representation of this process can be seen in Figure 3.1. A Perceptron is a neural network built using a distribution of these artificial neurons.

The output of a neural network is directly dependent on the weights of the neurons in the network, such that in order for a network to perform a given task, it must have appropriate weights. In supervised learning, a network can be trained to perform a certain task given a set of labeled input data, that is, a set of inputs along with the desired output of the network for those inputs. The loss function $L(\hat{y}, y)$ compares the network's output

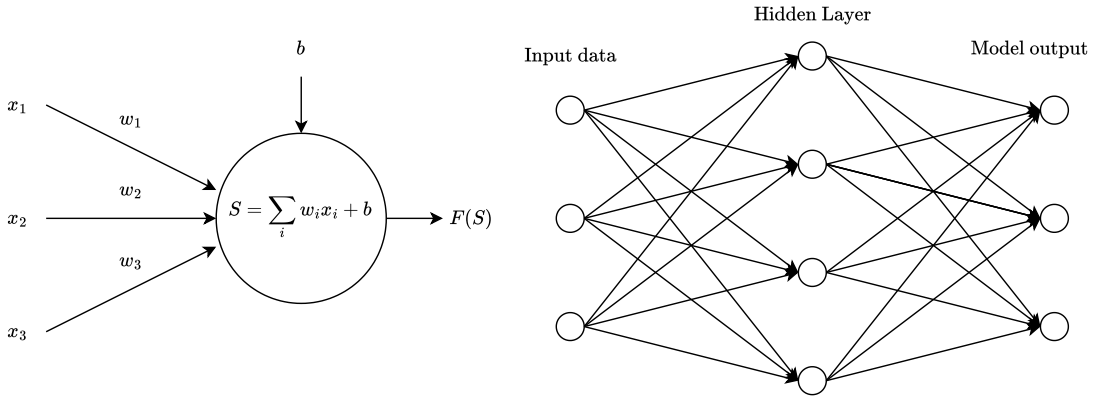

Figure 3.1 – On the left: Model of an artificial neuron. The output of the neuron is the activation function F applied on the linear combination of the inputs x and their respective weights, as well as the bias term. On the right: A model of a single layer Perceptron. In the case of a Multi-layer Perceptron, the outputs of the neurons in a hidden layer serve as inputs to the neurons in the subsequent hidden layer.

$\hat{y}$ to the true label y, and measures how well the network is performing. For example, in a classification task, a common choice for L is the cross-entropy loss, defined as

$$L_{\text{CE}}(\hat{y}, y) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{3.1}$$

where C is the number of classes, y_i is the true probability (1 for the correct class, 0 otherwise), and $\hat{y}_i$ is the predicted probability for class i. In regression tasks, the mean squared error (MSE) is often used as the loss function. It is defined as

$$L_{\text{MSE}}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2, \tag{3.2}$$

where N is the number of data points, $\hat{y}_i$ is the predicted output, and y_i is the true value.

The training process consists of comparing the output of a network for a given set of inputs with the label associated with these inputs. By computing the derivative of the loss function with respect to the weights, the gradient of the loss function is obtained. This gradient indicates the direction of the fastest decrease of the loss function. The weights of the network are then updated in the direction that minimizes the loss function, a process known as backpropagation.

In the case of Stochastic Gradient Descent (SGD), a common optimization algorithm, the rule for updating weights is given by:

$$W' = W - \eta \nabla_W L(W), \tag{3.3}$$

where W represents the current weights used by the network, W' represents the updated weights, and η is the learning rate. The learning rate controls how large a step is taken in the direction of the gradient during each update. A small learning rate leads to slow convergence, while a large learning rate can cause the algorithm to overshoot the optimal solution.

The process of backpropagation allows the network to learn how to adjust the weights at each layer to reduce the value of the loss function. By applying this process iteratively over a dataset, a network is able to learn the weights that better relate the input data and their expected outputs, and can then be used to create labels for unlabeled data.

Neural networks are often employed to perform regression and classification tasks. Regression tasks consist in predicting a value in a continuous space, whereas classification tasks predict values in a discrete space, where each point represents a class. By appropriately choosing a loss function, backpropagation can be used to perform both tasks, and works particularly well with low dimensional data.

While networks comprised of fully-connected layers can be used to virtually any kind of application, tasks that involve more complex or high dimensional data often benefit from more complex architectures[1] that allow for different operations to be performed. For instance, tasks involving time series data are often better modeled by Recurrent Neural Networks [44], algorithms that allow the information to flow in two directions within the network. For image analysis, the complexities in high-dimensional, often multi-channel images are usually analyzed using Convolutional Neural Networks (CNNs) [35, 36, 45].

3.1.1 Convolutional Neural Networks

With the goal of using neural networks to automate tasks of handwritten digit recognition, Yann LeCun developed the modern formulation of the Convolutional Neural Network in a work that became a milestone for AI development [42]. This special type of neural network uses a convolution operation to recognize patterns in two-dimensional data, making it well suited to image analysis.

A convolutional layer in a CNN works by sweeping an image, which can be thought of as a matrix of pixel values, with a smaller matrix called a filter, which analyses a region of the image and outputs the result of a convolution between the filter and that region of the image. The result of the convolution process between a filter and a region of an image is a single value that represents the existence of a certain feature in that region of the image. By sliding the filter over each possible region of the image, a map is created, showing the parts of the image that contain the feature that the filter is looking for. Thus, specially tuned filters sweep through an image and check for the existence of different features in each region of the image. The trainable parameters of convolutional layers are the pixel

[1] Architecture is the term used to refer to the distribution of neurons in a network.

values in the filters, such that the training process finds the filters that look for the most relevant features in the image. A diagram describing a convolutional kernel can be seen in Figure 3.2.

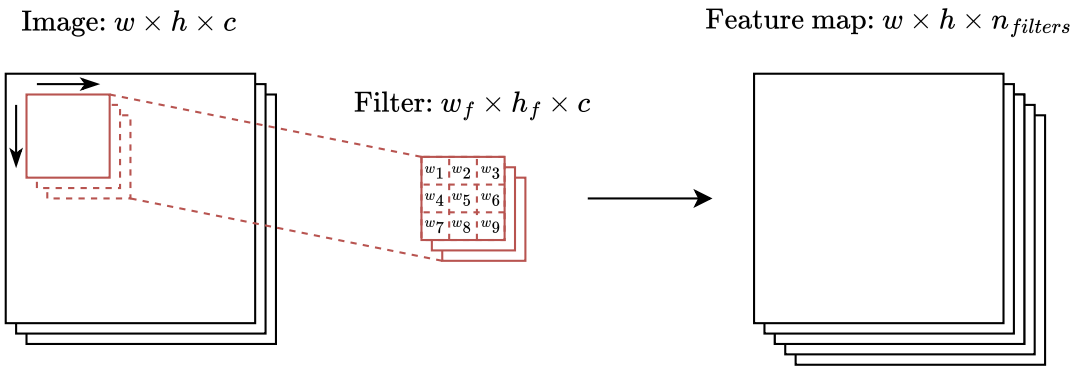

Figure 3.2 – Example of a convolutional layer. A filter sweeps through an image and returns a feature map describing the location of a certain feature in an image. A usual layer applies several filters to an image, returning a block of data with as many channels as the number of filters used.

In addition to convolutional layers, pooling layers are also widely used in CNNs. These layers also work by sweeping through the image, but instead of applying convolution with a filter, they condense the values in that region to a single value according to a specific function. Usual forms of pooling include max pooling, outputting the max value of a region, and average pooling, resulting in the average value of a region. Pooling layers are often applied to the feature maps generated by a convolutional layer, and are able to reduce the dimensions of the maps by analyzing non-intersecting regions of the maps and summarizing the information in these regions to a single value. This process makes a network more robust to variations in the position of features in the image. A diagram can be seen in Figure 3.3.

In most image analysis applications, convolutional Layers are the first layers of the networks, and combined with pooling layers, are used to summarize the information in images. This information is usually flattened to a single vector that is then used as an input to a fully-connected network, which is trained together with the convolutional part, sharing the same loss function. The convolutional layers are often referred to as the feature extractor part of the network, while the fully connected part is responsible for analysing the summarized information.

Throughout the 2010s, competitions such as the ImageNet Large Scale Visual Recognition Challenge [46], active between 2010 and 2017, fostered the development of highly specialized CNN architectures that very quickly surpassed the performance of humans in image classification tasks. In addition, the development and popularization of Graphical Processing Units (GPUs) played a major role in the AI boom of this era. As CNNs often rely on matrix multiplication, the use of GPUs, initially developed for image rendering

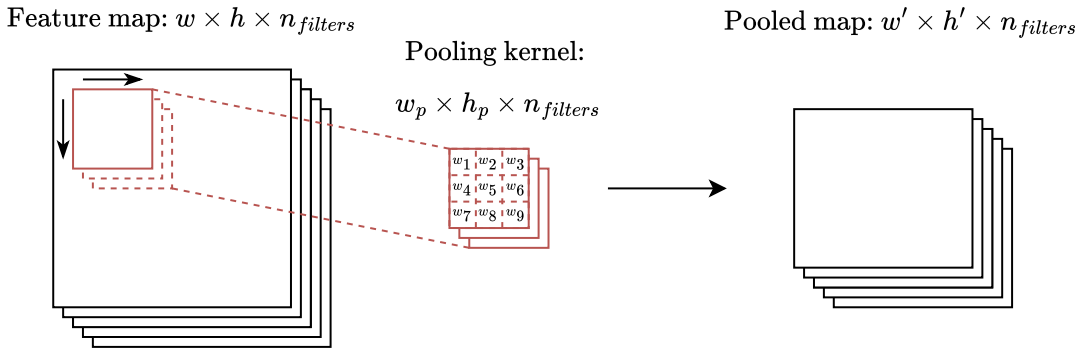

Figure 3.3 – Example of a pooling layer. Like in the convolutional layer, the pooling kernel sweeps through a feature map performing an operation using the pixels within the kernel. The shape of the output is generally smaller than the input (i.e. $w > w'$, $h > h'$).

for video games and Computer-Aided Design software, proved to be very well suited, drastically reducing training times for large CNN models. As a result, CNNs received a lot of attention from researchers in many different fields, including physics and astronomy [18, 19, 47, 48, 49, 50].

Although very capable in regression tasks, Neural Network based models are often criticized over their suitability to estimate uncertainties [51]. While methods such as cross-validation [36] can be employed to estimate *epistemic* uncertainties (those introduced by the Networks imperfect modeling of the data, often due to sub-optimal training or insufficient data), modeling *aleatoric* uncertainties, which arise from quality of the training data is a more daunting task.

In recent years, many models of Variational Inference [52] have been proposed to address this issue. Methods such as Bayesian Neural Networks (BNNs) [53] offer a probabilistic approach to parameter inference by learning probability distributions for weights instead of fixed values. For each forward pass of the network, the weights are sampled from their probability distributions, making the output of the network non-deterministic. Thus, by repeatedly showing a given input to a network, the probability distribution of the inferred value can be constructed.

Some of the major drawbacks of BNNs include the fact that they rely on the assumption that sampling the weights from simple probability distributions, often Gaussian, offers enough variation to appropriately model the uncertainties in the data. Furthermore, employing a BNN to analyze training data with no regard for its prior distribution fails to account for its influence on the posterior values, and therefore does not ensure a genuinely Bayesian approach. To address this limitation within a truly Bayesian framework, we turn to a novel method known as Simulation-Based Inference (SBI).

3.2 Simulation-Based Inference

Uncertainty estimation is an important aspect of statistical analysis. While traditional frequentist statistics are able to provide point estimates and confidence intervals, many problems require fully characterized posterior distributions in order to make reliable predictions from data. Bayesian statistics addresses this limitation by modeling unknown variables as probability distributions rather than fixed values. These distributions are refined from data, and incorporate both prior knowledge about the variable as well as its observed uncertainty. This approach allows for a more nuanced estimation of uncertainty, making Bayesian methods particularly valuable in complex inference tasks.

The task of parameter estimation can be posed in terms of Bayes Theorem, given by

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}. \tag{3.4}$$

We are interested in estimating the posterior distribution $P(\theta|x)$ of a parameter θ given an observation (or, in this case, a simulation) x. For this, we depend on the prior knowledge about the parameter $P(\theta)$ and on the likelihood distribution $P(x|\theta)$, which is a probability density on x, but a function of θ. While there is a dependence on the marginal term $P(x)$, that only serves as a normalization factor.

In many applications involving images and other types of high dimensional data, the likelihood term becomes intractable due to the complexity involved in constructing an analytical form for it. Simulation-Based Inference [54] methods offer a likelihood-free approach to parameter inference by replacing the likelihood term with a stochastic simulator. Instead of sampling values from an analytical likelihood distribution, in SBI, simulators are used to generate synthetic data modelling the effect of the likelihood. By using simulators that provide intrinsically uncertain data, SBI methods are able to account for both epistemic and aleatoric uncertainties without relying on an analytic form for the likelihood distribution. SBI can be performed using a number of different workflows, with advantages and disadvantages to each based on the characteristics of their intended use.

Neural Posterior Estimation (NPE) [55] is an SBI workflow that offers a fast and scalable approach to parameter inference. Unlike traditional approaches that rely on Markov Chain Monte Carlo (MCMC) sampling to reconstruct distributions, NPE uses a neural network to directly sample the posterior distribution. The direct sampling of the posterior distribution makes NPE more suitable to inference of multiple parameters (in other words, high-dimensional posteriors), avoiding costly MCMC calculations that would become prohibitively expensive in such scenarios.

In NPE, a simulator generates synthetic data using parameters drawn from a prior distribution. A neural network, acting as a density estimator, is then trained to approximate the posterior distribution. The training process minimizes a loss function based on the probability density of the true value (i.e. the label) under the learned posterior.

In practice the loss function used is the negative log-probability of the true value. The negative sign ensures that minimizing the loss means maximizing the probabilities, while using the logarithm of the probability helps with numerical stability.

For image analysis tasks, the network usually consists on two parts. A CNN employed to summarize the features in the images, referred to as the embedding, and the density estimator, a neural network-based algorithm that can be trained to model a complex probability distribution.

3.2.1 Density Estimation with Normalizing Flows

Normalizing Flows (NFs) [56, 57] are a class of neural network-based algorithms that can be employed to perform density estimation. In a normalizing flow, the input data is used to define the parameters of a set of invertible and differentiable transformations that are applied to a base distribution, with the goal of modeling a complex arbitrary distribution.

Considering a random variable in D dimensions $Z = \{z_1, z_2, ..., z_D\}$ with an associated probability distribution $p_Z(Z)$[2], and a transformation f that is both invertible and differentiable, the effect of this transformation on the variable creates a map $Z' = f(Z)$, such that the probability distribution on the new variable is given by

$$p_{Z'}(Z') = p_Z(Z) \left| det \frac{\partial f^{-1}}{\partial Z'} \right| = p_Z(Z) \left| det \frac{\partial f}{\partial Z} \right|^{-1}. \tag{3.5}$$

Thus, an arbitrarily complex probability distribution $p_{Z_n}(Z_n)$ can be constructed by repeatedly applying transformations on a base variable Z_0,

$$Z_n = f_n \circ f_{n-1} \circ ... \circ f_1(Z_0), \tag{3.6}$$

$$p_n(Z_n) = p_{n-1}(Z_{n-1}) \left| det \frac{\partial f_n}{\partial Z_n} \right|^{-1}, \tag{3.7}$$

$$= p_{n-2}(Z_{n-2}) \left| det \frac{\partial f_{n-1}}{\partial Z_{n-1}} \right|^{-1} \left| det \frac{\partial f_n}{\partial Z_n} \right|^{-1}, \tag{3.8}$$

$$= ... = p_0(Z_0) \prod_{i=1}^{n} \left| det \frac{\partial f_i}{\partial Z_i} \right|^{-1}. \tag{3.9}$$

In principle, this means that a distribution of interest (such as a posterior) can be obtained from any arbitrarily simple distribution by applying a series of appropriate transformations. To determine these transformations, a functional form is chosen, and its parameters are learned from data using neural networks. To be used as a trainable neural density estimator, a transformation f_i acting on the variable Z_{i-1} has its internal parameters Θ_i

[2] Note that $p_Z(Z) : \mathbb{R}^n \to \mathbb{R}$, that is, it maps an n-dimensional variable to a single dimensional output, as is expected of a probability distribution

determined by a neural network that takes the variable Z_{i-1} as input:

$$z_i = f_i(\Theta_i; Z_{i-1}), \tag{3.10}$$

$$\Theta_i = NN(Z_{i-1}). \tag{3.11}$$

The weights of this neural network affect how the transformation changes the base distribution, which in turn, affects the shape of the final distribution. This means that the weights have an effect on the value of the loss function of a network containing these transformations, and thus, the ideal weights can be learned through backpropagation. A diagram showing the workflow of Neural Posterior Estimation using Normalizing Flows can be seen in Figure 3.4.

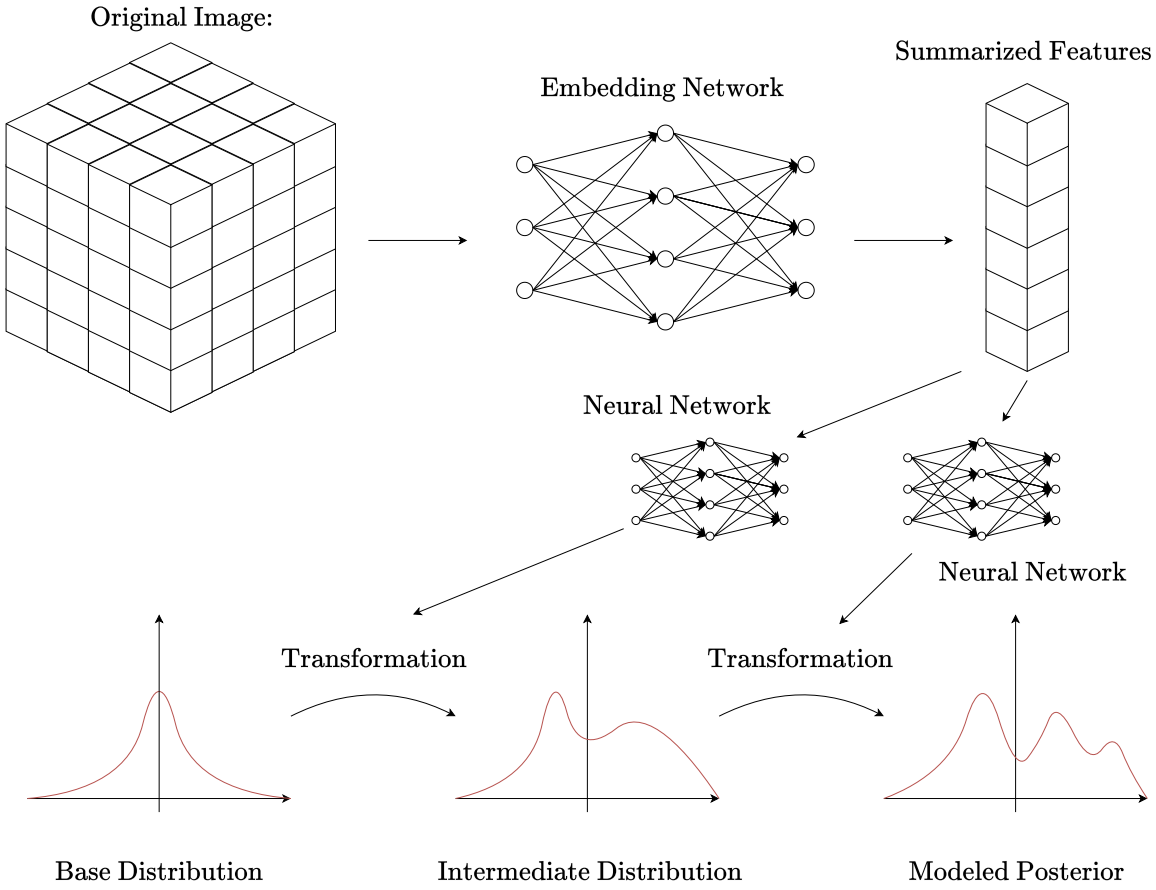

Figure 3.4 – Workflow of Normalizing flows used for Neural Posterior Estimation. The image is passed through an Embedding Network (usually a CNN), which returns a vector of summarized features. This vector is used as inputs for dense Neural Networks with a single hidden layer, and the outputs of those networks are the parameters of the transformations between different variables. Then by sampling from the base distribution and applying the transformations, it is possible to sample from the arbitrary modeled posterior.

There remains a question of what functional form should be used. This choice must take into account two main factors. As the dimension of Z increases, computing the

determinant of the Jacobian matrix becomes increasingly difficult. To deal with this problem, the transformation can be constructed in a way that minimizes the complexity of the calculation, usually by masking inputs in a way that causes the Jacobian to be a lower triangular matrix, with the determinant being simply the product of the elements in the main diagonal. Secondly, in order to mimic the intricacies of complex distributions, the transformations need to be complex enough to sufficiently alter the form of the original distribution. In other words, the transformations should be both *tractable* and *expressive*.

A typical transformation that acts elementwise, that is, alters a single dimension of the vector $Z = \{z_i\}_{i=0}^{d}$, is the *affine* transformation, given by

$$z_i' = g_{\Theta_i}(z_i) = \alpha_i z_i + \beta_i. \tag{3.12}$$

In this transformation, the parameters $\Theta_i = \{\alpha_i, \beta_i\}_{i=0}^{d}$ are determined by the output of a neural network that, in the general case, takes the vector Z as input. In order to increase the tractability of the transformation, the input of the neural network can be restricted, resulting in a simpler computation of the jacobian matrix.

3.2.1.1 Coupling Flows

One of the most commonly used ways to increase the tractability of a transformation without losing expressivity is by using a coupling transformation [58]. These transformations create two complementary subsets of the dimensions of the input vector $Z^A = \{z_1, ..., z_d\}$ and $Z^B = \{z_{d+1}, ..., z_D\}$, and use one of them to find parameters of a transformation h that acts on the dimensions of the other subset, following:

$$z^{A'}{}_i = h_{\theta_i}(z^A{}_i), \tag{3.13}$$

$$\theta_i = (\alpha_i, \beta_i) = \mathrm{NN}_i(Z^B), \tag{3.14}$$

$$z^{B'}{}_i = z^B{}_i. \tag{3.15}$$

In other words, each dimension of the first part of the transformed vector $Z^{A'}$ is given by a transformation h_{θ_i} whose parameters depend on the second input subset Z^B. The second part of the transformed vector is just a copy of the input vectors.

Coupling flows have a useful property: by using a subset of dimensions to generate the transformations h, the Jacobian matrix of these transformations is block triangular, and therefore, easy to calculate. Furthermore, when multiple transformations are chained together, the expressive power of the coupling flow can be greatly improved by applying a random permutation of the input dimensions, such that the subsets A and B of each intermediary transformation correspond to different dimensions of the original vector Z.

3.2.1.2 Autoregressive Flows

In autoregressive transformations [59], instead of using all dimensions of the vector Z as the input of a neural network that determines all transformations g_θ, the transformation that acts on each dimension is determined by its own neural network that takes as input only the dimensions lower than i.

$$z'_i = g_{\theta_i}(z_i), \tag{3.16}$$

$$\theta_i = \{\alpha_i, \beta_i\} = \mathrm{NN}_i(Z_{1:i-1}). \tag{3.17}$$

Thus, an autoregressive affine transformation employs equation 3.12 using parameters α_i and β_i following 3.16. The jacobian matrix for any autoregressive transformation is lower diagonal, and its determinant is simply the product of the elements in the main diagonal. As previously discussed, this property becomes increasingly useful as the dimension of Z increases.

3.2.1.3 Neural Spline Flows

One way to increase the the expressivity of a transformation is to replace the affine transformation with a more complex form. A Neural Spline Flow (NSF) [60] is a transformation that uses rational-quadratic polynomial splines as replacements for the affine transformation. A polynomial spline is a function that is defined piecewise, with each piece being a polynomial that connects to its neighbors at the edges of the spline. For a function to be rational quadratic, it must take the form of a quotient of two quadratic polynomials. By dividing the space in splines, and defining different elementwise rational quadratic transformations for each spline, NSFs offer increased expressivity compared to affine transformations.

In a NSF, the input is first mapped to the interval $[0, 1]$, which is then divided into K bins, each associated with a transformation $g_{\theta_{i_K}}$. The input vector Z (or limited sections of it, when using coupling or autoregressive flows) is processed by a neural network, which outputs a vector of $3K - 1$ parameters, partitioned as $\theta_i = [{\theta_i}^w, {\theta_i}^h, {\theta_i}^d]$.

To construct the piecewise transformation, the interval edges (knots) in the (z, z') space are determined by the parameters ${\theta_i}^w$ and ${\theta_i}^h$, which represent the widths and heights of the bins, respectively. The knots are the division points between adjacent bins, and their coordinates are computed using the cumulative sums of these width and height parameters. The derivative constraints are given by ${\theta_i}^d$, which specifies the derivative of the spline function at each knot. The two outermost knots have their derivatives fixed at 1 to ensure smooth boundary conditions.

By fixing the knots and the derivatives of the spline function at the knots, the quadratic polynomial functions of each bin are defined, and act as a transformation $g_{\theta_{i_K}}$ that maps z to z'. A visual example of this kind of transformation is shown in Figure 3.5.

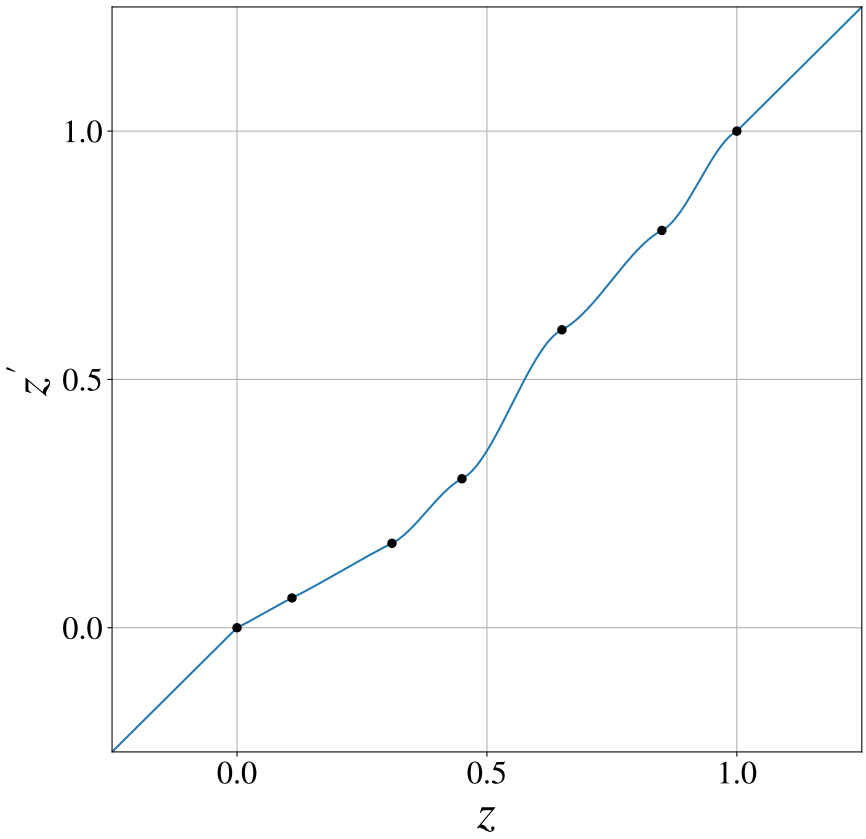

Figure 3.5 – Example of a transformation using a Neural Spline Flow. This transformation divides the interval $[0, 1]$ in 6 subspaces, where the positions of the knots (black dots) are the cumulative sums of ${\theta_i}^w, {\theta_i}^h$ and the derivatives of the transformation are fixed at the knots, and given by ${\theta_i}^d$. That information is enough to fit the rational quadratic polynomial splines that describe the transformation from z to z'.

By combining the method described above with either the coupling or autoregressive approach (i.e. defining each $\theta_i = \mathrm{NN}_i(Z_{1:i-1})$), NSFs show very high expressivity while maintaining tractability, and as such, become very powerful tools to perform tasks that depend on density estimation such as Bayesian inference, probabilistic prediction algorithms and even data generation. These transformations act as drop-in replacements to the affine transformation, and show much higher expressivity.

Chapter 4

Modeling Strong Lensing with Deep Learning

The deep learning methods discussed in the previous chapter have had a great impact on statistical analysis across many fields, including astrophysics and cosmology [61, 62]. In the context of strong lensing parameter inference, deep learning methods are a promising alternative to traditional likelihood-based methods, which would become prohibitively expensive in face of the volume of data expected in the near future.

In the past, CNNs have successfully been employed to extract features from lensing images [18, 19, 23], this approach provides limited capabilities for uncertainty estimation. To this end, methods relying on Bayesian parameter inference have been used in the past [23, 24, 25] using fully synthetic data to perform SBI, thus obtaining more reliable uncertainty estimation. However, the use of fully synthetic data provides limited variation in the image generating process, and could introduce biases related to the simulation algorithms.

In this work, we incorporate real galaxy images in the simulation process in order to increase the realism of the synthetic data generated. This approach ensures that the images better reflect real-world observing conditions. Unlike past work, we train our models to target parameters of the SIE lens model, including the redshifts of the lens and source galaxies. One of the challenges in modeling strong lensing systems is that redshifts are often not well modeled by currently available simulators, and may not accurately reflect real observations. In this work, we address this issue by incorporating real observational data for the lens galaxy in our simulations, which not only improves performance on real data but also enhances the reliability of our inferred parameters by better capturing the underlying cross-correlations between redshift and other lens model parameters. Lastly, our methodology is validated by applying our models to real images of strong lensing systems.

4.1 Simulating Strong Lensing images

Realistic simulations of astrophysical phenomena have become a key component of modern research, enabling scientists to test theories and interpret observations. These simulations are not only useful for understanding complex astrophysical phenomena, but can also be used to develop machine learning models tailored to analyze astrophysical and cosmological data. As described in the previous chapter, simulations can be used to train highly capable models to perform Bayesian inference. However, the effectiveness of these models is intrinsically linked to the realism and diversity of the simulations. The use of unrealistic simulations as training data could lead to models that show poor performance in real data, and could lead to biased or otherwise incorrect conclusions when interpreting results. Thus, it is necessary to ensure that the data used to train models is not only realistic, but also diverse.

4.1.1 Raw image simulation

In this work, we develop a method to simulate strong lensing images using real images of galaxies as a starting point. Our goal is to simulate data that resembles observations obtained by the DECam Local Volume Exploration (DELVE) survey [63], a wide-area optical survey designed to map over 21,000 square degrees of the southern sky using the Dark Energy Camera (DECam). It reaches a depth of $g \sim 23.5$ mag, enabling the detection of faint stellar systems and low-surface-brightness structures in the nearby universe. Thus, we generate images in the g, r, i and z bands, and include the instrumental and observational characteristics of DELVE, such as noise properties, pixel scale, and flux calibration.

To this end, we make use of two different packages to simulate each strong lensing system. First, we employ LensPop[1] [17] to simulate the parameters of a population of DECam-observable single-source galaxy-galaxy lensing systems. We apply a number of cuts to ensure detectability of lensing features by DECam. In particular, we make sure that the magnitude of the lens galaxy and the observed magnitude of the lensed features in the g band are below 21. We also employ cuts in the population in order to avoid creating regions of low sample size in the parameter space, which could lead to difficult training and inconsistent predictions by the network. We use Einstein radii (θ_E) between 0.6 and 2.2 arcsec, lens galaxy redshift (z_l) between 0.05 and 0.6 and source galaxy redshift (z_s) between 0.6 and 3, with the aforementioned condition of magnitude <21 for lensed features being the most restrictive. These cuts serve as limits for the prior distributions of the parameters. The values for the selection criteria are described in Table 1.

Since LensPop reports the magnitudes of the source object but not the lensed features, we model the magnification effect by first computing the flux f in each band using the

[1] Available at `https://github.com/tcollett/LensPop`.

Parameter	Cut Applied
Lens Galaxy Magnitude (g band)	< 21
Lensed Feature Magnitude (g band)	< 21
θ_E	[0.6, 2.2] arcsec
z_l	[0.05, 0.6]
z_s	[0.6, 3.0]

Table 1 – Parameter cuts applied to the simulated population to ensure detectability and avoid low-sample regions in parameter space.

object's absolute magnitude M and the zero-point magnitude of the observation M_0, multiplying the flux by the source magnification ratio μ reported by LensPop to obtain the magnified flux f^*, and then converting the lensed flux back to magnitude, following

$$\begin{aligned} f &= 10^{\frac{\mathrm{M}-\mathrm{M}_0}{-2.5}}, \\ f^* &= \mu f, \\ \mathrm{M}_{lensed} &= \mathrm{M}_0 - 2.5\log_{10}(f^*). \end{aligned} \tag{4.1}$$

The process of simulating an image for a given system consists of sampling a set of values for θ_E, z_l and z_s for a uniform prior using the intervals mentioned above, and comparing it to the population generated by LensPop. We assign a score to each member of the population by summing the squared difference between the sampled value and the population value over these three parameters[2]. We then randomly pick one of the systems with the ten highest scores and use its parameters to generate a simulation.

We compare the redshift of the lens galaxy with the redshifts available in a dataset of real galaxy images taken by the DELVE survey. We initially attempted to use spectroscopic redshifts, but the network failed to capture a relationship between the images and the values. We switched to photometric redshifts, for which we had measurements for a larger number of galaxies. We make a list of the three galaxies with redshifts closest to our chosen value and pick randomly from it. Finally, we perform a data augmentation step in order to increase the diversity of the simulated data. This step consists of choosing one of eight possible permutations of a square image (i.e., 0°, 90°, 180° and 270° rotations as well as a horizontally flipped version for each). We use the augmented image as a starting point to the simulation.

We then employ Lenstronomy [3] [64, 65, 66], a state-of-the-art strong lensing simulation package, to simulate the lensed features according to the parameters of the system chosen from the population. We use observational parameters (read noise, gain, sky brightness, zero-point magnitude) for each band taken from the real galaxy image, and generate an

[2] I.E. the score S_i of the i-th member of the population is given by $S_i = \sum_\Theta(\Theta_{\text{sampled}} - \Theta_i)^2$ for parameters $\Theta = [\theta_E, z_l, z_s]$.

[3] Available at https://lenstronomy.readthedocs.io/en/latest/.

image of a dark galaxy acting as lens. The dark galaxy is simulated using magnitude 200, Sérsic index 4 and Sérsic radius taken from the value indicated in the population generated by LensPop. Finally, the parameters of the source galaxy are taken from the chosen member of the population of strong lensing systems generated by LensPop.

The simplifying assumptions used are that all images in all bands use 7 exposures and have seeing sampled from a normal distribution with mean 1 and sigma 0.25. This value is chosen empirically comparing lenstronomy simulations with real images. The output generated by lenstronomy is a dark image where the only visible features pertain to the source galaxy (i.e. the lensed image and the source galaxy itself, when visible), which is then added to the original real galaxy image. It is worth noting that Lenstronomy uses the magnitude of the lensed features to generate a simulation, which is obtained following Equation 4.1. The fixed values used in the simulation process are shown in Table 2.

Parameter	Value
Simulated lens galaxy magnitude	200
Simulated lens galaxy Sérsic Index	4
Number of Exposures	7
Seeing	$\mathcal{N}(1, 0.25)$

Table 2 – Fixed parameters used in Lenstronomy to simulate the images. The parameters of the simulated lens galaxy (i.e., the dark galaxy).

The simulation process takes a parameter sampled from a prior distribution and matches it with a catalog in order to generate the final image. The ground-truth parameter for the image can be interpreted as either the original sample from the prior distribution or the value that is accepted during the matching process. As this choice can bias the inference process, this work investigates the results using both sets separately. We refer to the values taken from the prior distribution as the *tentative* set, and the values resulting from the matching process, which are ultimately passed to lenstronomy to generate the images, as the *effective* set. Both sets are reported in Figure 4.1. A diagram with the full simulation procedure can be seen in Figure 4.2.

4.1.2 Image preparation

To prepare the both real and simulated images for use in deep learning applications, we employ a number of steps that aim to improve visibility of relevant features, as well as improve training stability and overall network performance. We follow the image preparation described in [67] with some changes to better suit our application. First we apply a cut to pixel values in the galaxy images used to create the simulations, with a minimum allowed value of -400 and a maximum value of 6000 analog-to-digital units (ADU). Then, after the images are simulated and the lensed features are added to the galaxy images, we convert pixel values to flux units using each simulation's zero-point

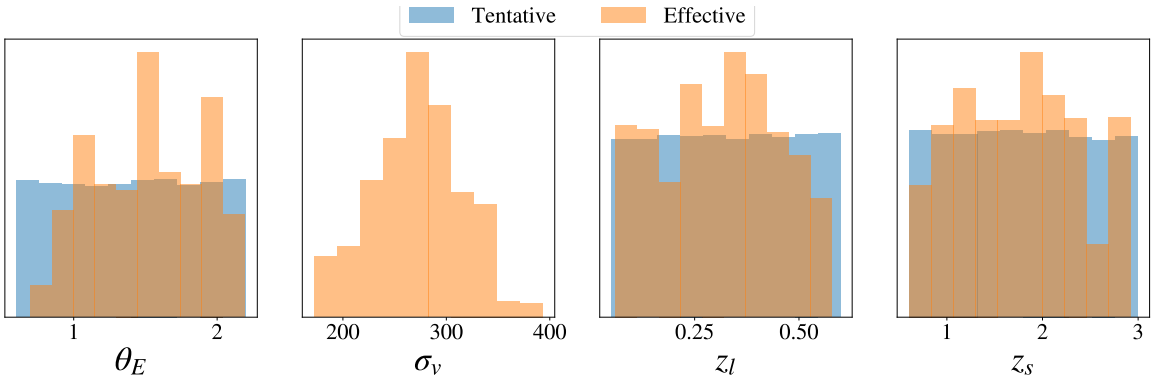

Figure 4.1 – Tentative (blue) and Effective (orange) sets for θ_E (arcsec), σ_v (km/s), z_l and z_s. Note that the velocity dispersion is not sampled in the tentative set, so we only report the effective distribution for that parameter.

magnitudes and exposure times, following [68]. Then, we rescale the images by subtracting the mean value of pixel counts and dividing by the standard deviation. Although this step does not alter the final image, it effectively rescales the values shown to the network and is known to help with training stability and model convergence. Additionally, the four parameters used as labels to train the network are also normalized using the minimum and maximum values in the simulated population for each parameter respectively, following

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}. \tag{4.2}$$

This effectively constrains all values to the interval $[0, 1]$, which also helps to improve network performance. A diagram of the image preparation process can be seen in Figure 4.3.

The simulation and preparation processes yields a dataset of 30000 griz-band images used for training as well as a test set consisting of 3000 images. These images are ready to be shown to a neural network. Some examples of images are shown in Figure 4.4.

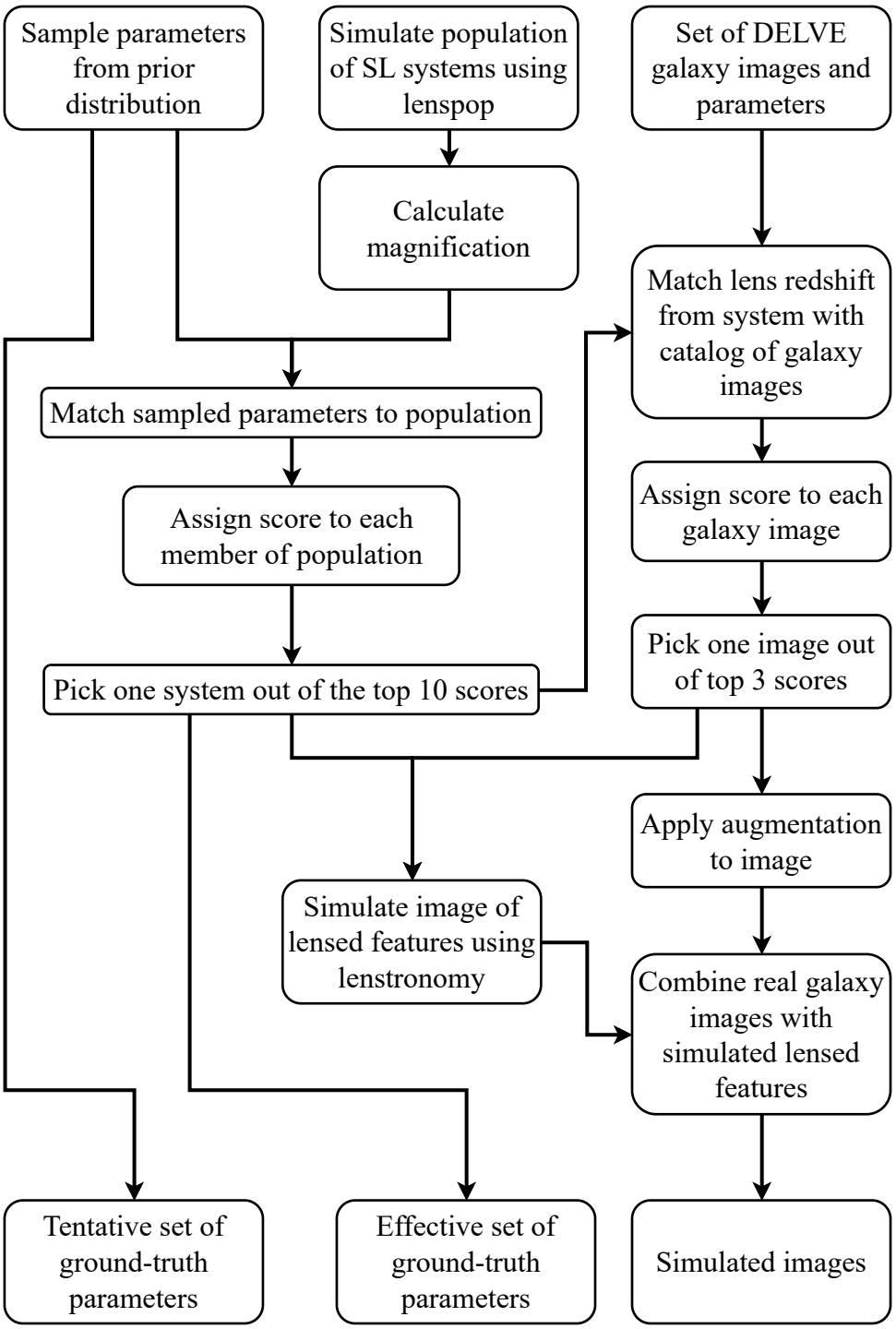

Figure 4.2 – Diagram of the simulation process. The final products are the images and the ground-truth parameters, which are used as labels to train the neural networks. Both the images and the labels undergo additional preparations steps before being shown to the networks.

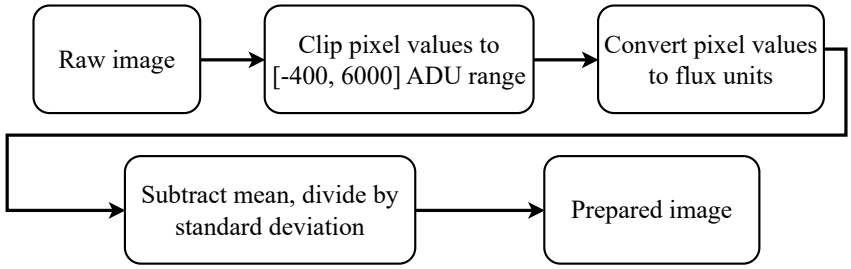

Figure 4.3 – Diagram of the image preparation process. The prepared image is used to train a the neural networks, along with the normalized ground-truth parameters.

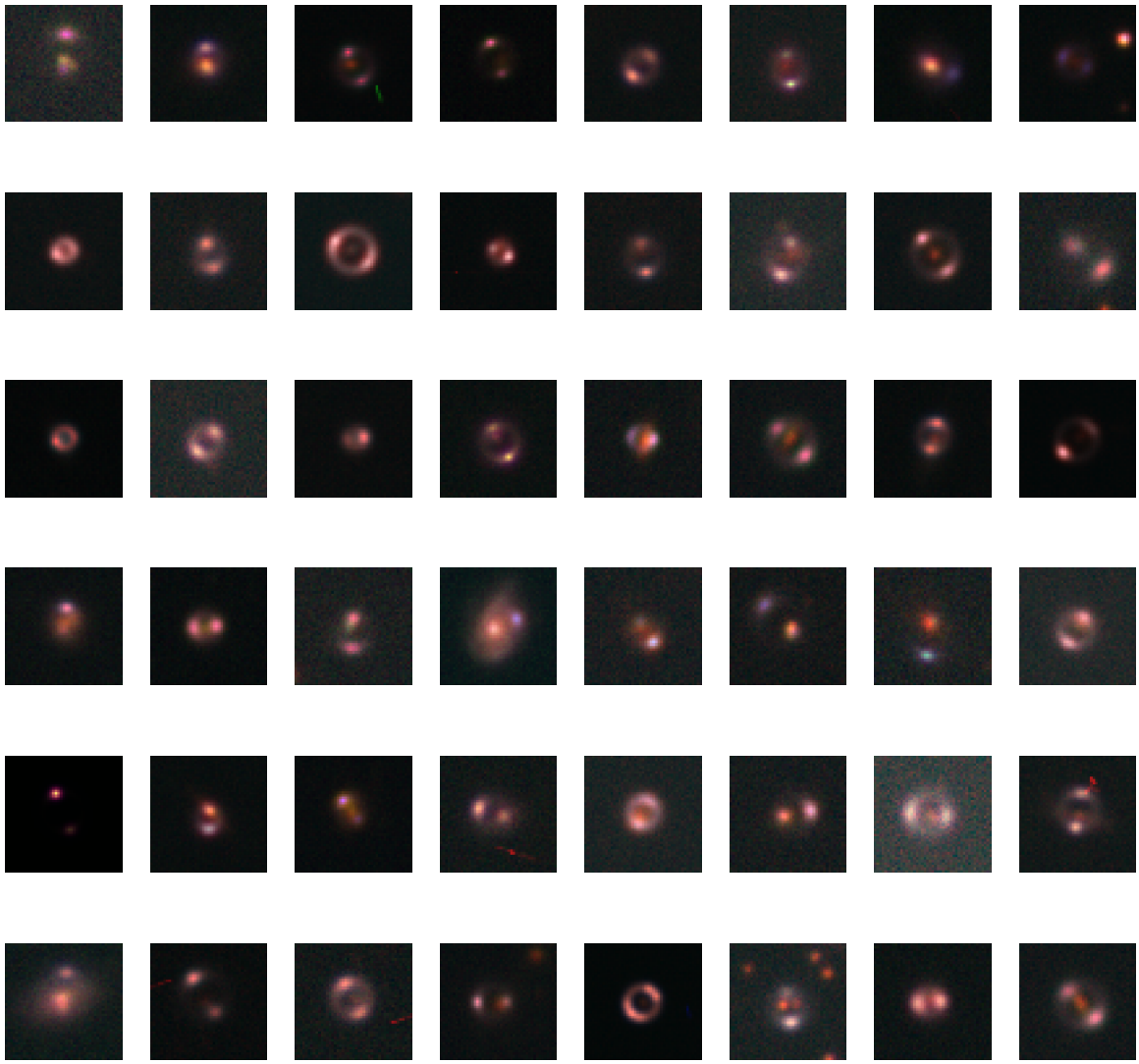

Figure 4.4 – Examples of simulations generated by the method described in this chapter. Aside from the described preparation, in order to display RGB images, we perform a rescaling of the pixel values in each band separately and map the bands $i + z$, r and g to R, G and B respectively.

4.2 The Deep Learning Model

The goal of the training process is to obtain a model capable of inferring the four-dimensional posterior distribution given an image, where each dimension corresponds to one of the four parameters of interest. As explained in Chapter 3, neural networks are well suited to that application. Thus, our training data is fed to a network that performs Neural Posterior Estimation targeting the four-dimensional posterior distribution. We use an inception-based architecture [69] as an embedding network that reduces the dimensionality of the data. A diagram of the architecture can be seen in Figures 4.5 and 4.6. The density estimation part of the network is done using a Neural Spline Flow. We also explored using Mixture Density Networks and Gaussian Mixture Models for the density estimation, but found that the NSF outperformed the other two methods.

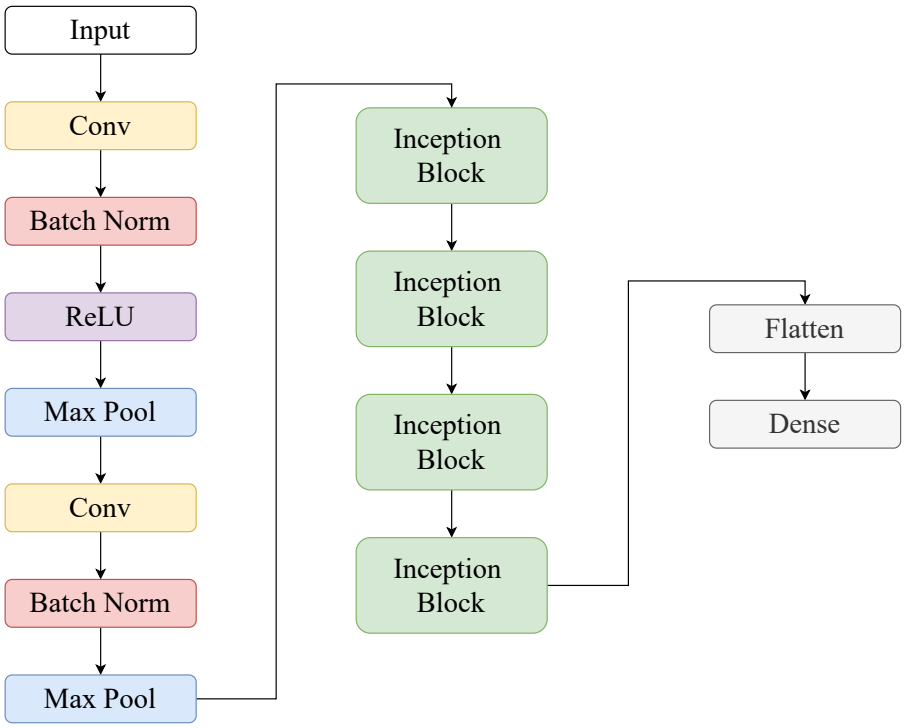

Figure 4.5 – Representation of the embedding network using the Inception-based architecture. The Conv and MaxPool layers are explained in Chapter 3. Relu (Rectified Linear Unit) is the activation function used in these layers, given by $\text{ReLU}(x) = \max(0, x)$. The size of the final output layer is determined by the grid search. A detailed diagram of the Inception block can be seen in Figure 4.6.

To determine the ideal values for some architecture choices of the network, a grid search was performed over three key parameters: the number of output neurons in the embedding network (n_{out}), the number of transformations in the Neural Spline Flow ($n_{\text{transforms}}$), and the number of hidden features in the one-layer neural network that

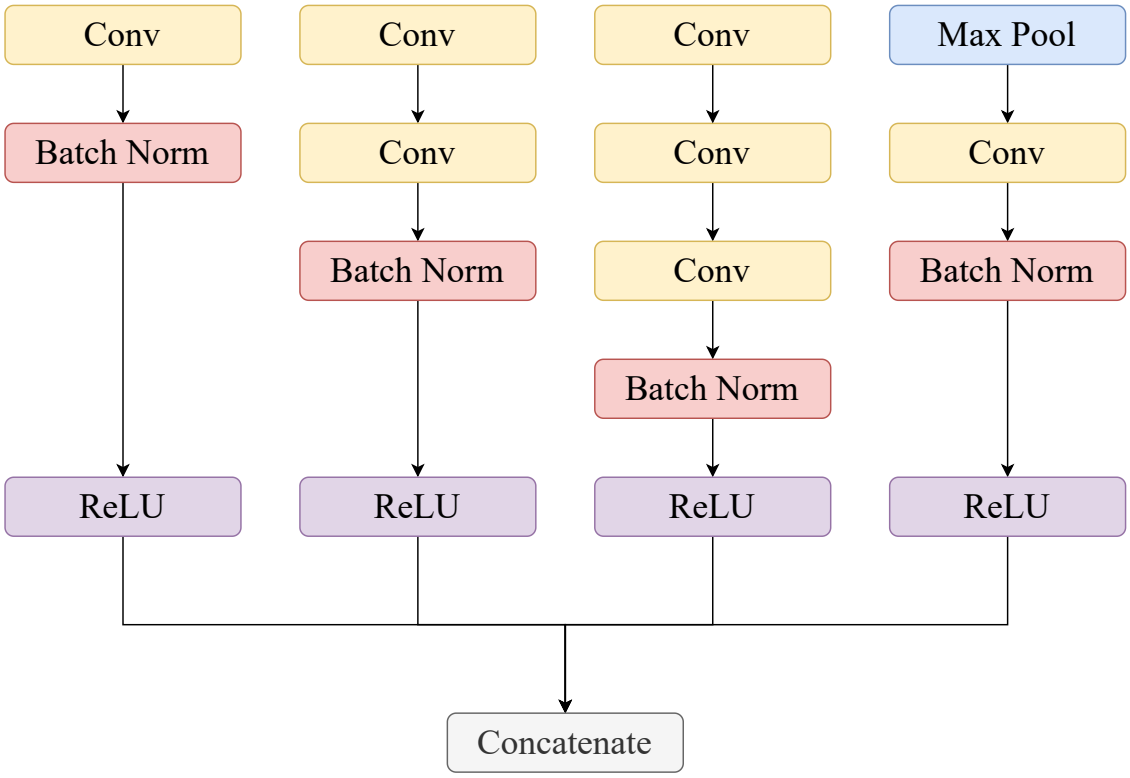

Figure 4.6 – Diagram of an Inception block.

determines the parameters of each transformation in the NSF (n_{hidden}). The explored values for each of these choices are summarized in Table 3.

Architecture Choice	Explored Values
n_{out}	{16, 32}
$n_{\text{transforms}}$	{4, 6}
n_{hidden}	{24, 32, 48}

Table 3 – Explored architecture choices in the grid search. The parameter n_{out} represents the number of output neurons in the embedding network. The parameter $n_{\text{transforms}}$ corresponds to the number of transformations in the Neural Spline Flow. Finally, n_{hidden} denotes the number of hidden features per transformation.

The training process is maintained for a maximum of 3000 epochs. We set aside 10% of the data as a validation set, used to monitor the performance of the model in data that is not used for weight optimization. This set is evaluated at the end of every epoch. With the goal of avoiding excessive training that could lead to overfitting, we introduce an early stopping protocol. If during training, the value of the loss function calculated with the validation data does not improve for 100 epochs, the training training is stopped. We use a batch size of 256 image-label pairs, and a learning rate of 10^{-3}. The optimization is handled by an Adam optimizer [70]. The values of the hyperparameters of the model are summarized in Table 4.

Parameter	Value
Maximum Epochs	3000
Patience	100 epochs
Batch Size	256
Learning Rate	10^{-3}
Optimizer	Adam

Table 4 – Model hyperparameters used during the training process.

4.3 Analyzing the output of the model

The evaluation of the test set consists of applying the trained model to an unseen image. To evaluate the test set, the image is passed through the network, which is equivalent to taking a sample from the posterior distribution of the parameters given that image. To estimate each posterior distribution, this process is repeated 10000 times for each image, yielding an empirical distribution of values. Then, a statistical analysis can be performed on each empirical distribution in order to derive an estimate for the value of each parameter, as well as their confidence intervals.

The predicted values can be compared directly with the values used to generate that simulation to ensure that they are consistent with each other. This can be done using the Pearson Correlation Coefficient. It is also useful to use precision, defined as

$$P = 1 - \frac{|(\text{pred}_{+1\sigma} - \text{pred}_{-1\sigma})|}{\text{pred}}, \tag{4.3}$$

and fractional deviation, defined as

$$D = \frac{|(\text{true} - \text{pred})|}{\text{true}}, \tag{4.4}$$

as evaluation metrics. Since these are obtained for each image individually, we report the median value obtained over the entire test set.

In some situations, simulation-based inference algorithms can produce unreliable uncertainties [71]. The expectation for a well-calibrated model is that the percentage of true values covered by a certain confidence interval should increase linearly as the confidence interval increases. This expectation follows from the definition of confidence intervals: if a model is well-calibrated, a $p\%$ confidence interval should contain the true value in approximately $p\%$ of cases over many repeated samples. As a result, comparing the empirical coverage of true values relative to their confidence interval offers a sense of how well-calibrated the model is.

To assess the validity of the posterior distributions obtained by the model, a number of Simulation-Based Calibration [72, 73] checks were employed, based on the analysis of rank statistics. The rank of a posterior is given by the number of posterior samples that fall under its true value. As such, for a well-calibrated model, the rank distribution over a set of predicted posteriors is expected to be uniform. We employ two methods

to summarize rank statistics: a Classifier Two-Sample Test (C2ST) [74] comparing the ranks distribution to a uniform distribution and a Kolmogorov-Smirnov p-values test (KS), testing the null-hypothesis that these two distributions are the same.

Chapter 5

Results

The methods discussed in the previous chapter were used to design and train a neural network using the simulated data described in Section 4.1. The training process was carried out in a multi-GPU cluster using 8 Nvidia RTX 3090 GPUs. While each model was trained on a single GPU, utilizing nearly all of the available 24 GB of VRAM, the grid search over architecture choices was performed in parallel, making use of multiple GPUs simultaneously. The hardware specifications are summarized in Table 5.

Component	Specification
CPU	Intel(R) Xeon(R) Platinum 8260
GPU	8 x NVIDIA GeForce RTX 3090 24 GB
RAM	1 TB
OS	Ubuntu 24.04.1 LTS

Table 5 – Hardware specifications for the machine used to train the models discussed in this work.

To decide on the best model, we evaluated the results obtained by the different architectures considering both precision and accuracy. As described in the previous chapter, we used a combination of metrics such as median precision, median fractional deviation, and Pearson correlation to evaluate accuracy, alongside uncertainty calibration metrics like C2ST and KS-p-values to assess precision. We present results from models that achieve the best balance between the aforementioned metrics.

The simulator described in Section 4.1 gives rise to two sets of ground-truth parameters, referred to as the *tentative* set and the *effective* set. As the choice of set can result in different behavior of the posterior distributions obtained by the model, the results for both sets are presented separately.

5.1 Tentative Set

5.1.1 Simulated Data

The training process for the best performing model took approximately 64 minutes, which translates to 181 epochs. A summary of the training process can be seen in Figure 5.1. The evaluation of the test set took 1 minute and 24 seconds (0.03 seconds per image) and yields all of the metrics discussed in the previous chapter. We report these results in Figure 5.2, which shows a comparison of true and predicted values for all four parameters. In these plots, the results are divided in bins over the true values, and the plot reports the mean predicted value (yellow line), as well as the confidence intervals (blue for 1-sigma, dark and light green for 2- and 3-sigma, respectively) calculated for each bin. We also report the calculated values of the Pearson coefficient, precision and fractional deviation in Table 6. We note that the use of the tentative set of ground-truth parameters results in a mostly uniform distribution of ground-truth values for all parameters except for the velocity dispersion, which is not contemplated in the tentative set and was trained using ground-truth values from the effective set. The parameters of the architecture used to obtain these results are shown in Table 7.

Metric	θ_E	σ_v	z_l	z_s
Median Precision	85.73%	98.05%	85.99%	81.29%
Median Fractional Deviation	11.95%	5.09%	29.06%	23.40%
Pearson	0.756	0.761	0.409	0.484

Table 6 – Evaluation metrics for the best-performing model.

Architecture choice	Chosen value
Output features	32
Hidden features	48
Number of transforms	6

Table 7 – Architecture chosen from the result of the grid search done with models using the tentative set.

For the Einstein radii, the model shows good performance, being able to accurately predict values in most regions. The predictions tend to show worse performance in the lower extreme (i.e. Einstein radius below 1 arcsec), which can be explained by the lower sample size in the effective prior, causing the model to be shown simulations with less diversity for this range of values, hurting its performance on unseen data. It is worth noting that it becomes increasingly difficult to resolve lensed features from the lens galaxy as the radius decreases, which could also lead to lower performance in these regions.

The predictions for the velocity dispersion tend to show very high deviation above 350 km/s, a region of critically low sample size in the prior distribution. Many regions also

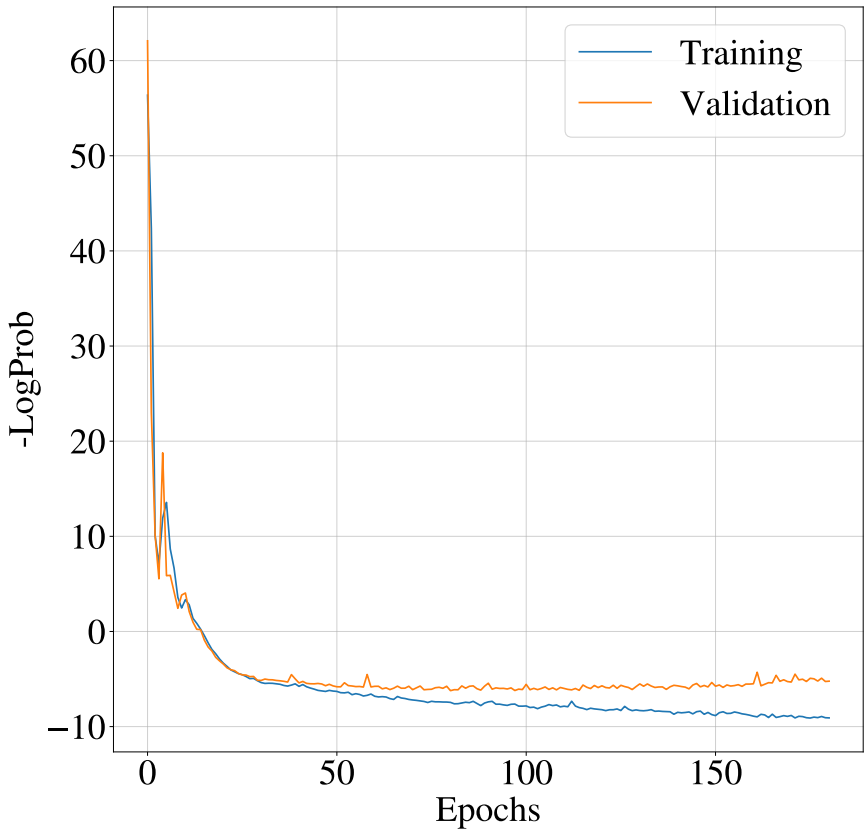

Figure 5.1 – Evolution of model performance over the epochs.

have true values lying outside of the 3 sigma intervals, showing low model performance. Nevertheless, the median fractional deviation values show that predictions are not too far from true values. The Pearson coefficient has a similar value to the one found for the Einstein Radius, showing that the model has a general sense of the relation between true values and predictions. The use of the effective prior to train the data could have led the model to generate predictions with increased precision, an effect that is systematically present in the results that will be discussed in Section 5.2.

For the redshift of the lens galaxy, the effective prior distribution has high variations in sample size. The predictions show low performance anywhere outside of the central region, which is likely to be a consequence of a network that is unsure what to predict, and minimizes the squared error by predicting central values. Nevertheless, the Pearson value of 0.5 indicates a moderate correlation between predictions and true values. The low performance can be attributed to the realism in the simulations - since our simulator uses real galaxy images to simulate the lens galaxy and we use a photometric estimate of the redshift of the lens galaxy as the true value, the model effectively tries to relate a real image to it's photometric redshift, which is an active area of research [75, 76], and often requires a more comprehensive approach.

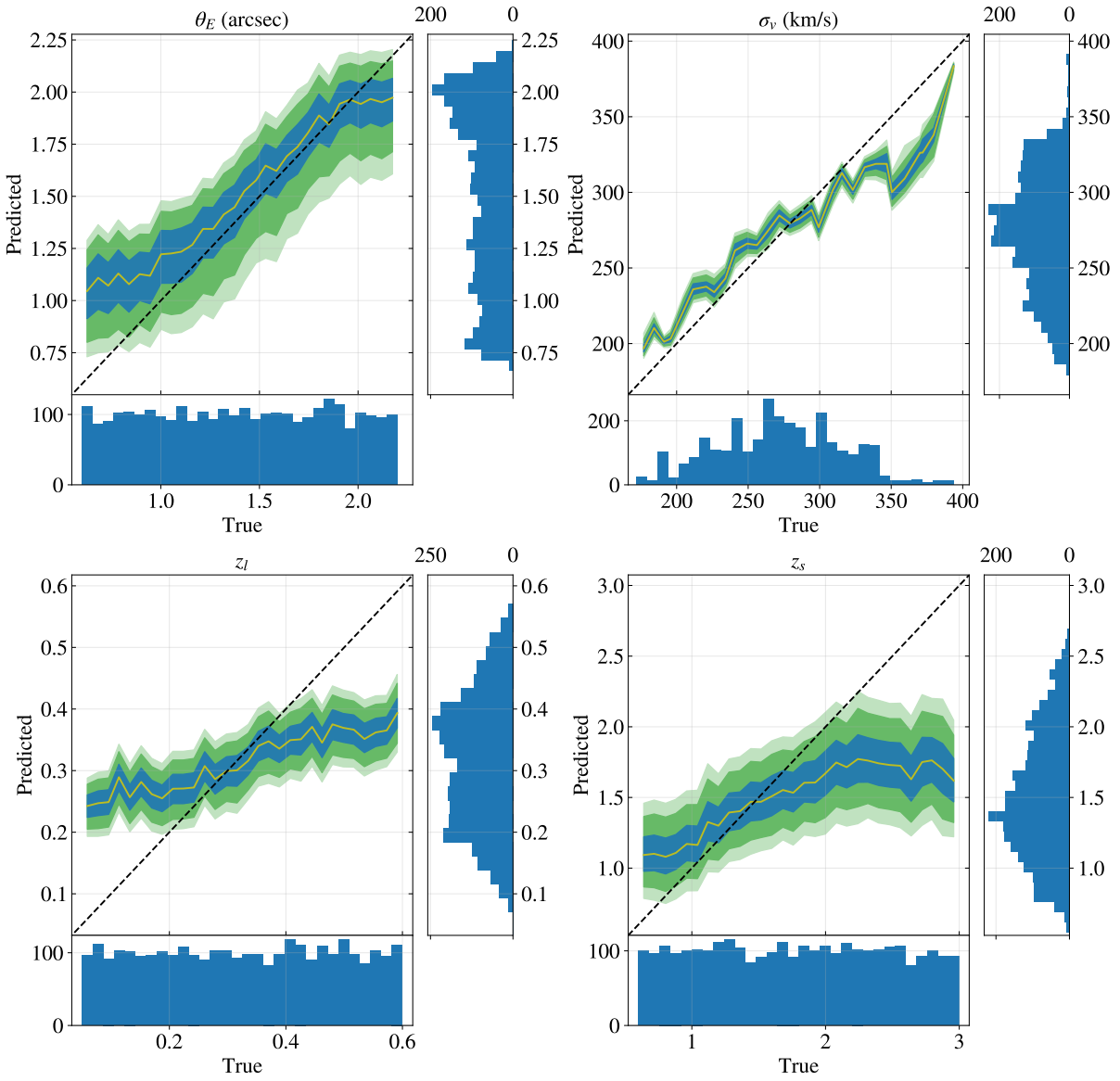

Figure 5.2 – Comparison of true and predicted values for the Einstein radius (top-left), lens velocity dispersion (top-right), lens redshift (bottom-left) and source redshift (bottom-right). The blue region corresponds to the one-sigma confidence interval, and the dark and light green regions correspond to the two- and three-sigma confidence intervals, respectively.

Finally, the predictions for the redshift of the source galaxy show adequate performance for redshift below 1.5, but tend to deviate from true values above that. This is again explained by the lower sample size of the effective prior distribution in those areas, but also a consequence of the fact that objects with higher redshift tend to have fainter magnitudes, thus are harder to detect and accurately model. These estimates also require the network to account for magnification effects on top of the magnitudes and colors of the lensed features, and as such, prove a more difficult task than other parameters.

As discussed in the previous chapter, we employ a number of metrics to verify the quality of the posterior distributions reported by the model. We present a posterior

coverage plot in Figure 5.3, where we report the empirical coverage of true values relative to confidence interval. We also report values for KS test and C2ST in Table 8. The values for the KS test indicate a vanishing probability that the rank distribution is sampled from a normal distribution, which indicate poorly-calibrated uncertainties. This diagnostic is further confirmed by the C2ST values, which represent the accuracy of a classifier tasked with differentiating points from the predicted posterior distributions and uniform distributions in the same range. The values that deviate from 0.5 indicated that the C2ST classifier is systematically able to differentiate the two distributions. However, the posterior coverage plots indicate that for all four parameters, the empirical coverage grows faster than the confidence interval, which indicates that the posterior distributions reported by the model are underconfident (i.e., wider than expected for a calibrated model). Thus, we can expect the true value of a parameter to be covered within one-sigma more frequently than the confidence interval suggests, providing a conservative safety margin for the uncertainties.

Metric	θ_E	σ_v	z_l	z_s
C2ST	76.0%	86.5%	81.9%	81.6%
KS-pvals	0	0	0	0

Table 8 – Uncertainty calibration metrics for the best-performing model.

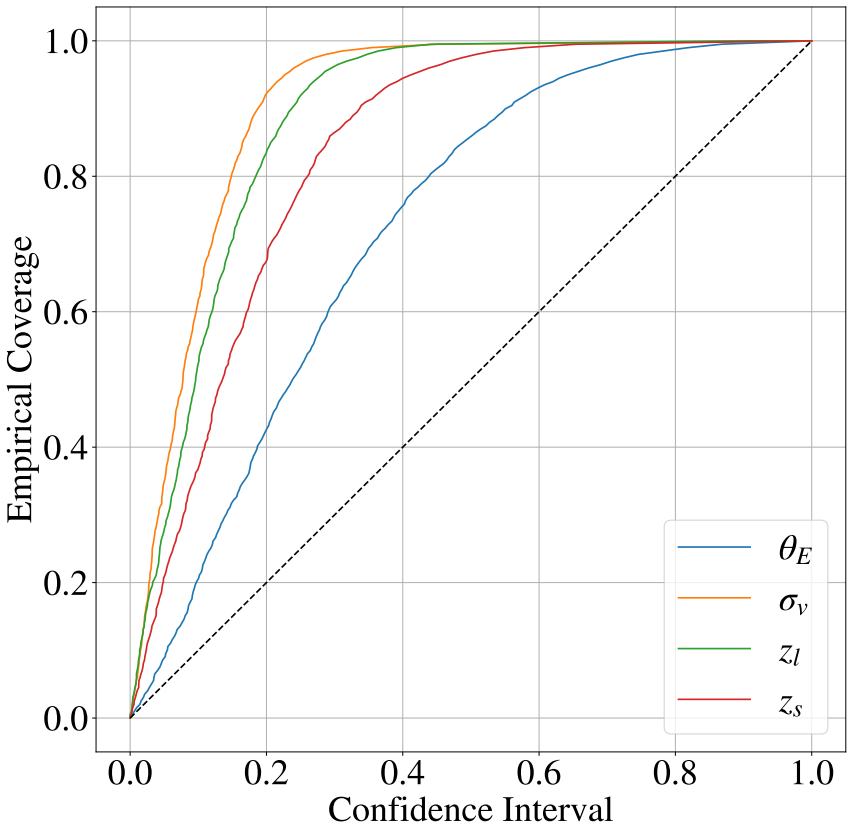

Figure 5.3 – Comparison of empirical and expected coverages for the four parameters.

5.1.2 Real Data

As the final goal of this work is to enable fast and automated analysis of lensing parameters on images generated by future surveys, we apply our model to available real Strong Lensing data as a way to gauge the preparedness of this approach to real world applications. As mentioned in Chapter 1, one of the biggest challenges with strong lensing today is the lack of real data. Strong Lensing systems are not only relatively rare occurrences, but also pose an observational challenge given the difficulty of identification. Thus, we report the results of using our model on two different datasets of images taken by the DELVE survey, demonstrating its ability to recover lensing parameters in real-world scenarios. These results highlight both the strengths and limitations of the model when applied to current observational data.

5.1.2.1 LaStBeRu Data

The Last Stand Before Rubin (LaStBeRu) [77] project aims to catalog all known strong lensing data before the start of LSST [15]. We apply cuts to the catalog in order to keep only single-source, galaxy-galaxy strong lensing systems that are located within

the footprint of the DELVE survey, since this is what the simulations are tailored to mimick. The resulting dataset contains 48 systems, with ground-truth values for all four parameters of interest. We use the astrometric data reported in the LaStBeRu catalog to generate image cutouts of the systems from the DELVE survey. As such, there is no guarantee that the lensing features are clearly observable using DECam. We present a comparison of our simulations and the cutouts in Figure 5.4. We deploy the model in these images, and we report the results in Figure 5.5. While the low sample size prevents any meaningful statistical analysis, we report the evaluation metrics in Table 9.

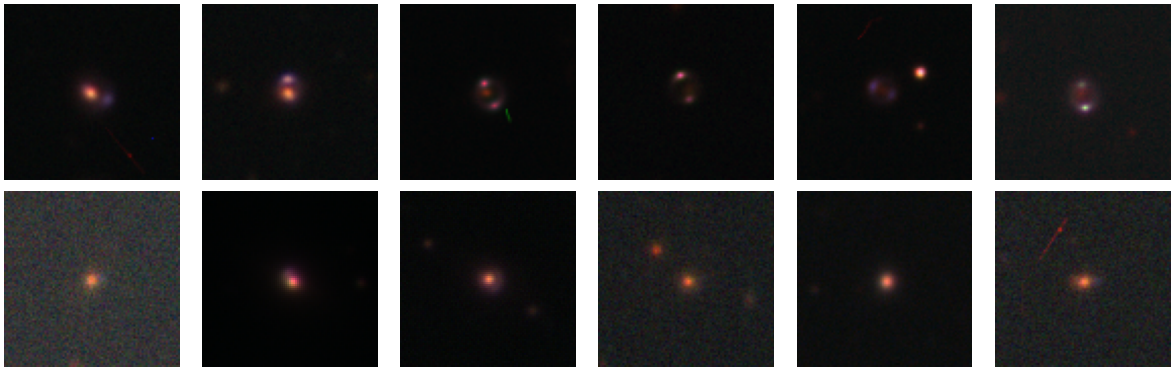

Figure 5.4 – Comparison between simulation (top row) and observations (bottom row) for the dataset generated using LaStBeRu data.

Metric	θ_E	σ_v	z_l	z_s
Median Precision	84.45%	93.30%	81.67%	75.98%
Median Fractional Deviation	35.64%	21.40%	42.38%	72.44%
Pearson	-0.063	0.152	0.434	0.233

Table 9 – Evaluation metrics for the best-performing model applied to the dataset generated using LaStBeRu data.

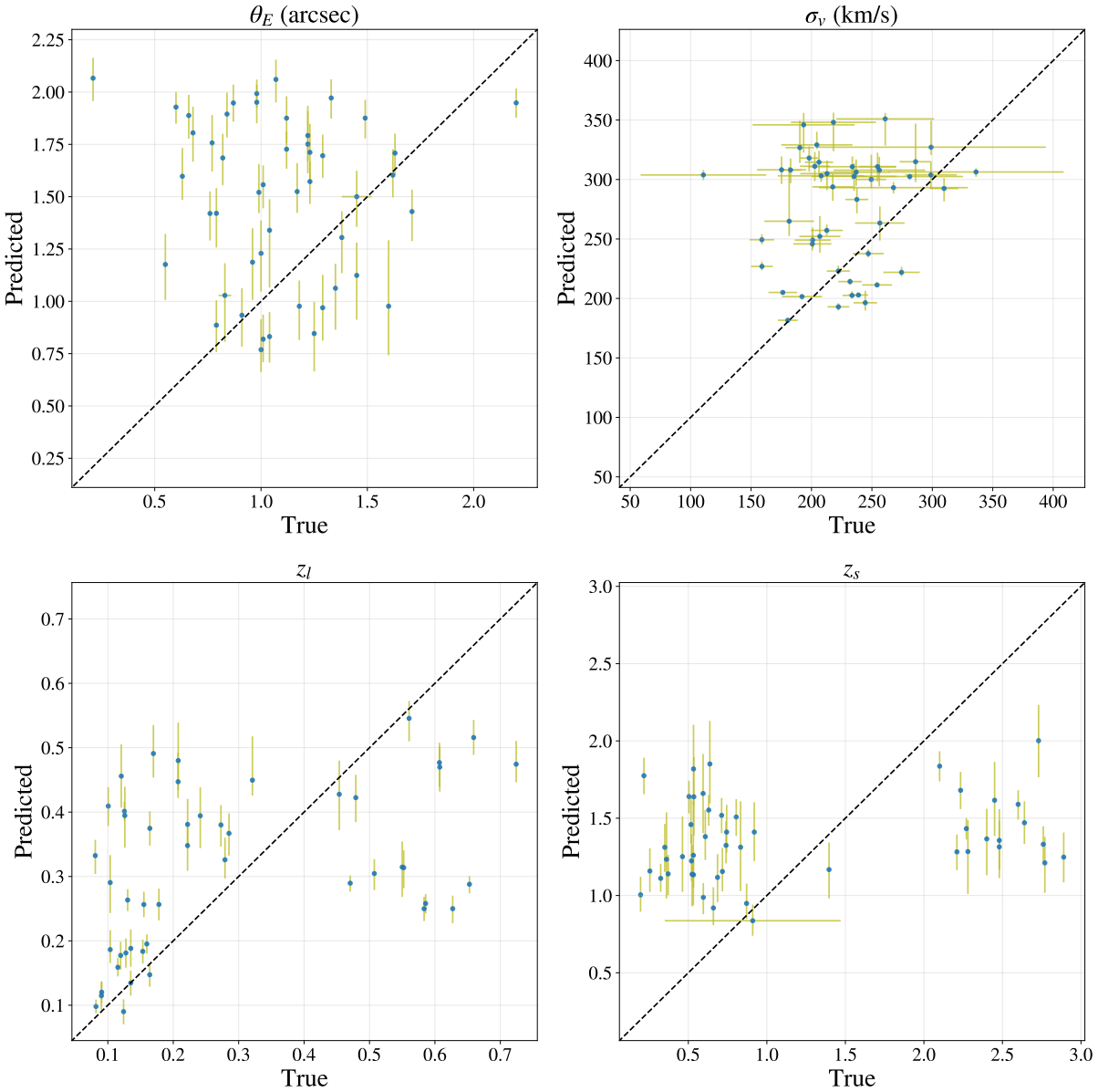

Figure 5.5 – Comparison of true and predicted values for the Einstein radius (top-left), lens velocity dispersion (top-right), lens redshift (bottom-left) and source redshift (bottom-right) for the dataset generated using LaStBeRu data. The vertical bars represent the 1-sigma confidence interval. The uncertainties in true values are reported when available.

5.1.2.2 DELVE Data

The second dataset is a subset of the systems identified in [67]. We limit ourselves to analyzing grade A candidates with a single source object, as this is the scenario used to generate the simulations, yielding a set of 22 observations. A comparison between some observations and simulation can be seen in Figure 5.6. For these images, the only ground truth parameters available are the Einstein Radii, obtained by measuring the separation between the lensed features and the lens galaxy in pixels and converting to angular distance using the pixel scale of the survey. This process is not as accurate as

traditional modeling, and tends to yield imprecise results, but is relatively fast to perform. The main results can be seen in Figure 5.7. As with the first dataset, the low sample size prevents a meaningful statistical analysis, but the evaluation metrics are reported in Table 10. It is worth mentioning that these results lie outside of the range of Einstein Radii used for training. This could explain the hesitance of the network in predicting values above 2 arcsec. Nevertheless, some of the results within the range of the simulations are compatible with the available estimates.

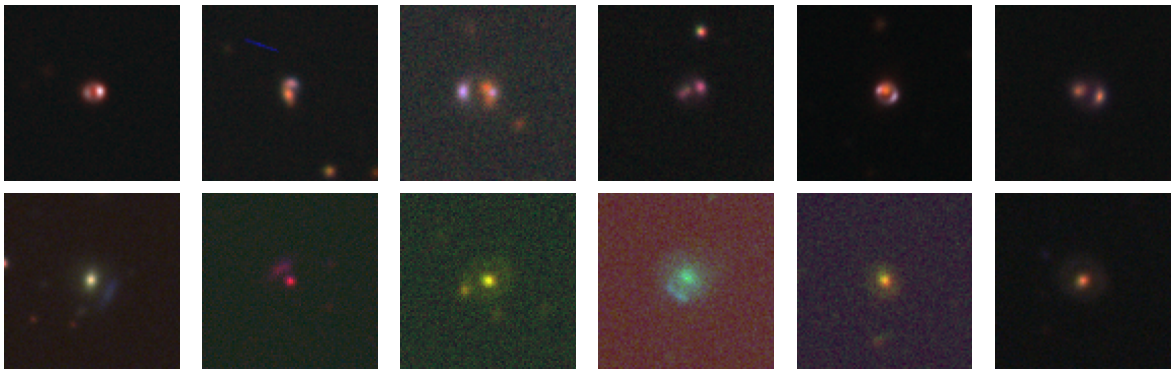

Figure 5.6 – Comparison between simulation (top row) and observations (bottom row) for the dataset generated using data from the DELVE survey.

Metric	θ_E
Median Precision	71.76%
Median Fractional Deviation	43.02%
Pearson	0.032

Table 10 – Evaluation metrics for the best-performing model applied to the data from the DELVE survey.

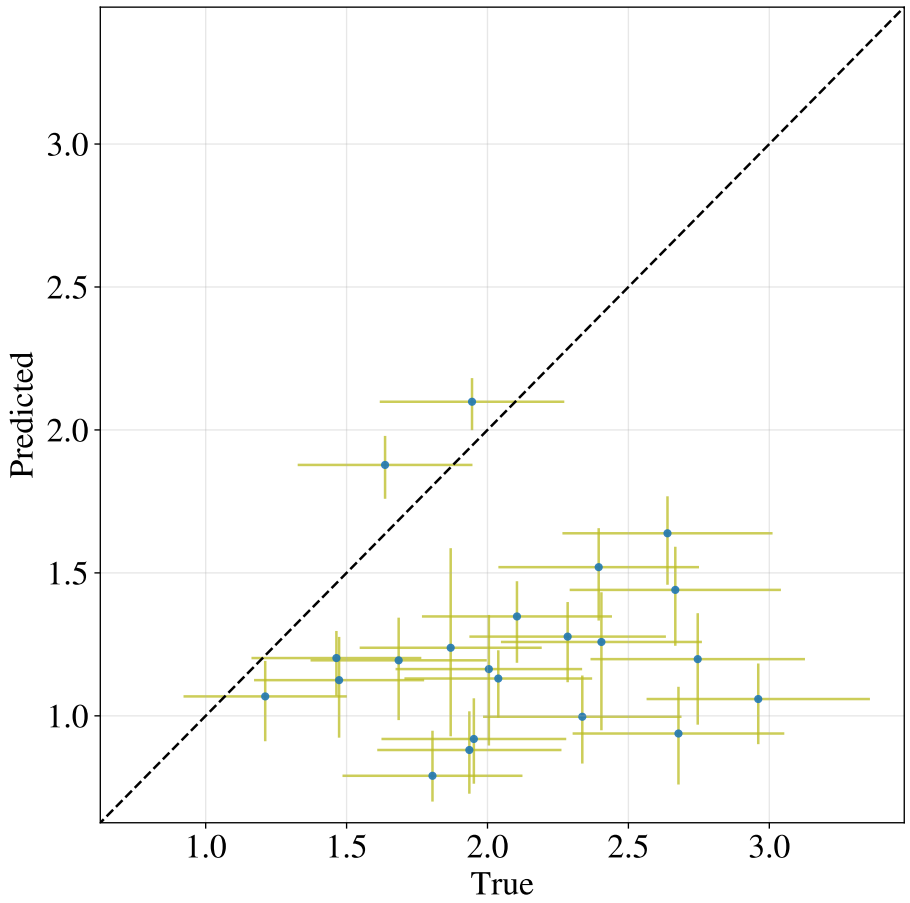

Figure 5.7 – Comparison of true and predicted values for the Einstein radius using real data from the DELVE survey. The vertical bars represent the 1-sigma confidence interval. The uncertainties in true values are reported also reported.

5.2 Effective Set

5.2.1 Simulated Data

The training process for the best performing model trained with the effective set of ground-truth parameters took approximately 75 minutes. This time translates to 241 epochs. The evaluation of the test set took 1 minute and 4 seconds (0.02 seconds per image). Like in the previous section, the evolution of the loss function during training is represented in Figure 5.8, and the evaluation metrics are shown in Table 11. The parameters of the architecture used to obtain these results are shown in Table 12.

The results for the effective set are more precise compared to the results for the tentative set. The matching process used for the tentative set introduces an extra layer of separation between the values used to simulate an image and the ground-truth values, which explains the wider uncertainty regions displayed by the models trained with the tentative set. This effect is not present for the velocity dispersion, for which the ground-truth values in both sets are the same.

Metric	θ_E	σ_v	z_l	z_s
Median Precision	98.69%	99.08%	96.66%	97.05%
Median Fractional Deviation	2.22%	3.6%	17.26%	10.02%
Pearson	0.964	0.892	0.679	0.817

Table 11 – Evaluation metrics for the best-performing model trained with ground-truth values taken from the effective prior.

Architecture choice	Chosen value
Output features	32
Hidden features	24
Number of transforms	4

Table 12 – Architecture chosen from the result of the grid search done with models using the effective set.

In general, the fractional deviations are smaller for the effective set. However, the high precision obtained by the model often causes the true value to fall outside of the 3-sigma confidence interval of the posterior distribution. These results are shown in Figure 5.9, and follow the same logic as those shown in 5.1.

An important difference between the two sets is the distribution of ground-truth values. The use of the effective set results in less control over the values, which can cause regions of low sample size. These regions can be attributed to the simulation of the population done by LensPop. There is a tendency for regions of high sample size to show lower deviations between true and predicted values. The results for the uncertainty calibration can be seen in Figure 5.10, as well as Table 13.

The predictions made by the model for the Einstein radius are generally in agreement with the true values. Despite the good correlation, evidenced by the Pearson coefficient values close to 1, some regions have true values falling outside the 3-sigma confidence region. This can be attributed to the high precision obtained by the model. However, the posterior coverage plots indicate that the uncertainty regions predicted by the model are overconfident, even more than those obtained for the tentative set. This effect could be a result of the model learning biases in the simulator, which become evident due to the lower uncertainty introduced by the use of the effective set.

With the tentative set, the results for the velocity dispersion show the importance of a uniform distribution of true values. The true values agree with predicted values in regions of higher sample size (i.e., between 220 and 340 km/s), but show high deviation above 340 km/s, a region of critically low sample size. The low fractional deviation combined with the relatively high Pearson coefficient values indicate that the model is able to predict reasonable point estimates, but the caveats related to uncertainty estimation mentioned for the Einstein radius also apply for the velocity dispersion. It is worth mentioning that although the ground-truth values for velocity dispersion are the same between the two sets, the models generate different predictions due to the different architecture, as well as

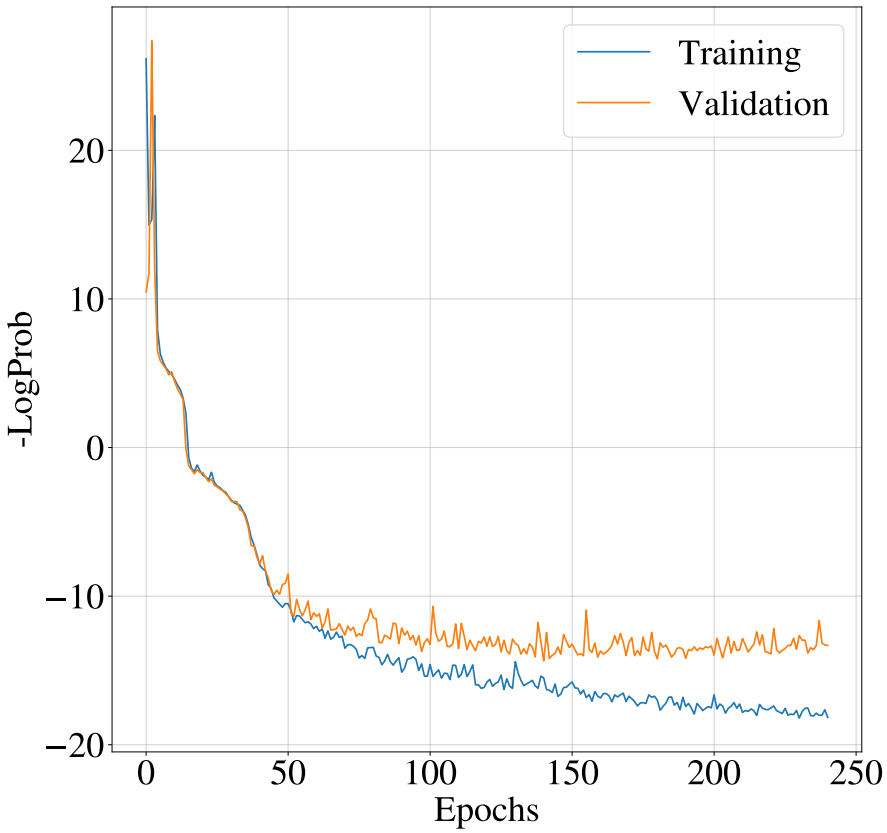

Figure 5.8 – Evolution of model performance over the epochs for the model trained with ground-truth values taken from the effective prior

the cross-correlation between the parameters.

For the lens redshift, the results are similar to those obtained with the tentative set, possibly due to the variations in sample size. For the effective set, the high precision obtained by the models causes true values to not agree with the predicted values for the majority of the parameters space. However, the Pearson coefficient indicates a marginally better correlation compared to the tentative set. This can be a consequence of the fact that the redshift of the lens galaxy is taken from a real image, and therefore, using the exact value for the redshift caused the network to perform better point estimates.

Lastly, the redshift of the source shows a behavior similar to the one obtained with the tentative set. There is reasonable agreement between true and predicted values for redshift below 1.5, but the results show a notable decrease in performance beyond that threshold. The same considerations about observational limitations discussed in Section 5.1 also apply.

A common trend between the parameters is the better correlation between true and predicted values compared to those obtained with the tentative set. The Pearson coefficient values are systematically higher for the effective set. However, the high precision

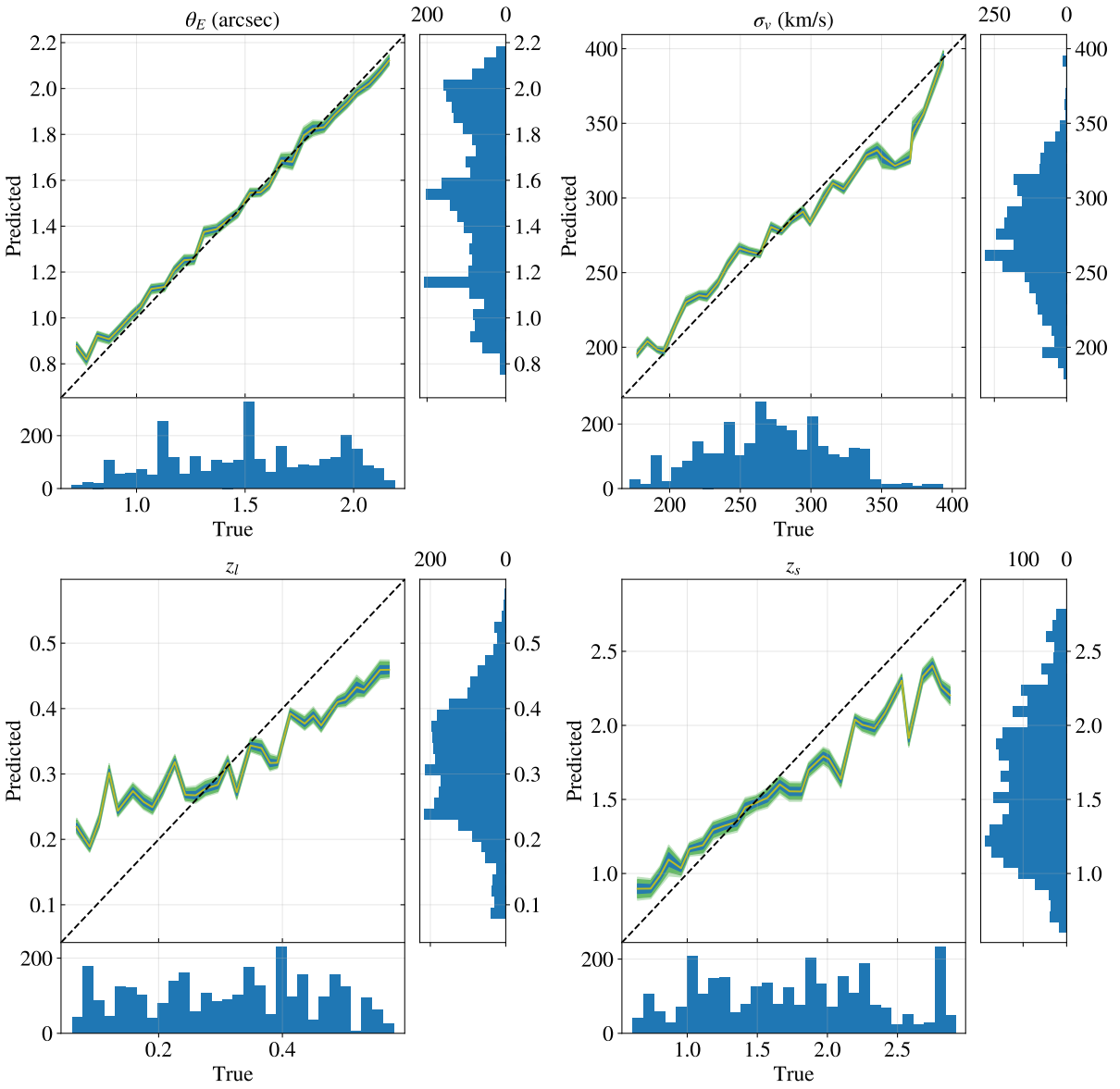

Figure 5.9 – Comparison of true and predicted values for the model trained with ground-truth values taken from the effective prior for the Einstein radius (top-left), lens velocity dispersion (top-right), lens redshift (bottom-left) and source redshift (bottom-right). The blue region corresponds to the one-sigma confidence interval, and the green regions correspond to the two- and three-sigma confidence intervals.

obtained by the model combined with a posterior coverage plot indicating overconfidence shows that the model might be learning biases in the simulation process. This effect is less noticeable with the tentative set due to the higher separation between the simulated images and the ground-truth values, causing the true values to be covered in the 3-sigma confidence intervals in a bigger part of the parameter space.

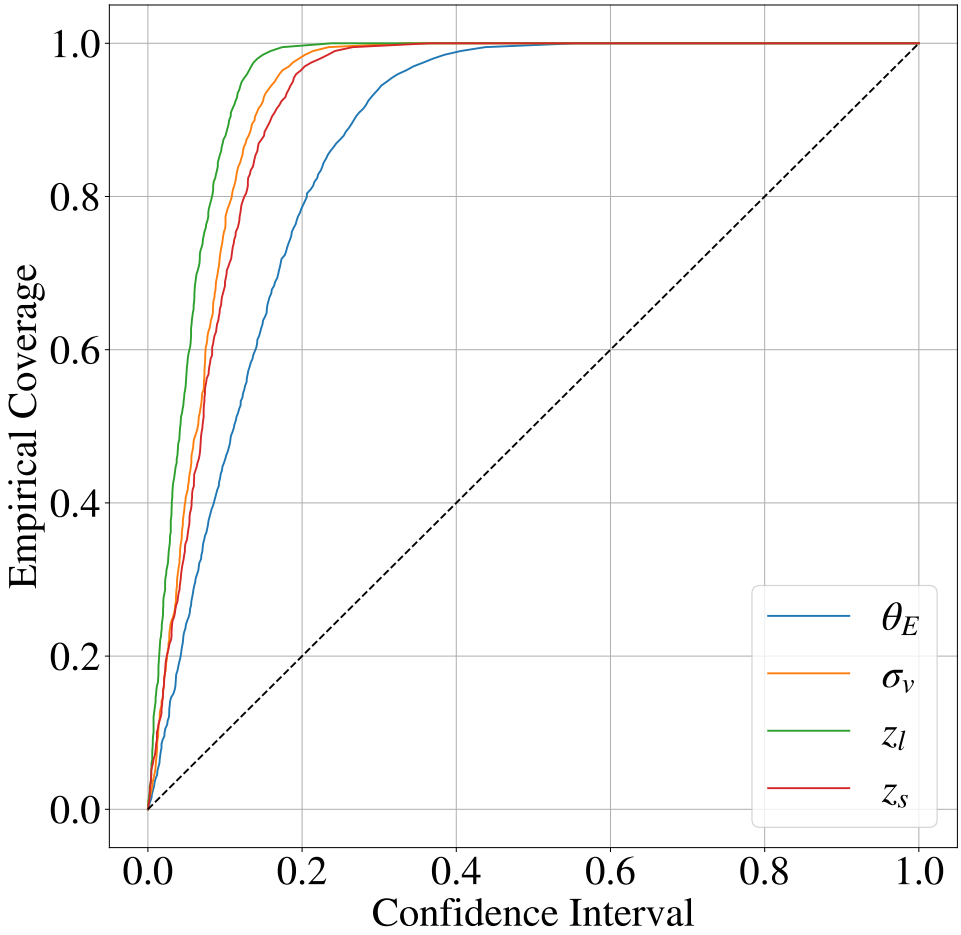

Figure 5.10 – Comparison of empirical and expected coverages for the model trained with ground-truth values taken from the effective prior across the four parameters.

Metric	θ_E	σ_v	z_l	z_s
C2ST	96.4%	93.8%	94.3%	95.0%
KS-pvals	0	0	0	0

Table 13 – Uncertainty calibration metrics for the best-performing model trained with ground-truth values taken from the effective prior.

5.2.2 Real Data

5.2.2.1 LaStBeRu Data

We report the results obtained by the model applied to the LaStBeRu dataset. The considerations made in section 5.1.2.1 still apply. The main results are shown in Table 14, as well as in Figure 5.11.

Metric	θ_E	σ_v	z_l	z_s
Median Precision	97.58%	98.24%	95.27%	94.30%
Median Fractional Deviation	27.74%	18.48%	47.96%	101.05%
Pearson	-0.048	0.175	0.380	0.116

Table 14 – Evaluation metrics for the best-performing model applied to the dataset generated using LaStBeRu data, with ground-truth values taken from the effective prior.

5.2.2.2 DELVE Data

Lastly, we report the results obtained by the model applied to the DELVE dataset. Like with the LaStBeRu data, The considerations made in section 5.1.2.2 still apply. The main results are shown in Table 15, as well as in Figure 5.12.

Metric	θ_E
Median Precision	98.19%
Median Fractional Deviation	40.84%
Pearson	0.166

Table 15 – Evaluation metrics for the best-performing model applied to the data from the DELVE survey, with ground-truth values taken from the effective prior.

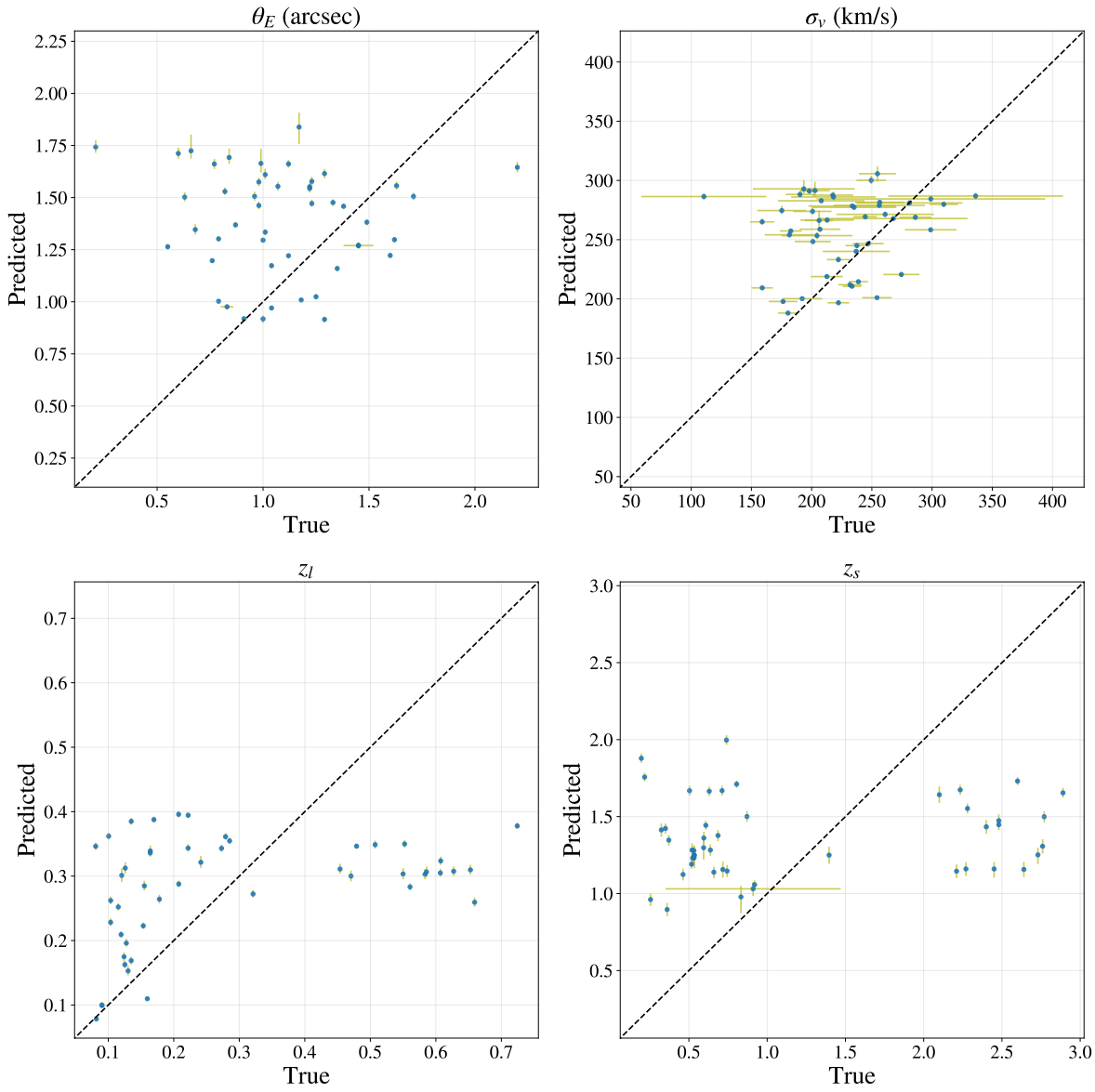

Figure 5.11 – Comparison of true and predicted values for the model trained with ground-truth values taken from the effective prior for the Einstein radius (top-left), lens velocity dispersion (top-right), lens redshift (bottom-left) and source redshift (bottom-right) for the dataset generated using LaStBeRu data. The vertical bars represent the 1-sigma confidence interval. The uncertainties in true values are reported when available.

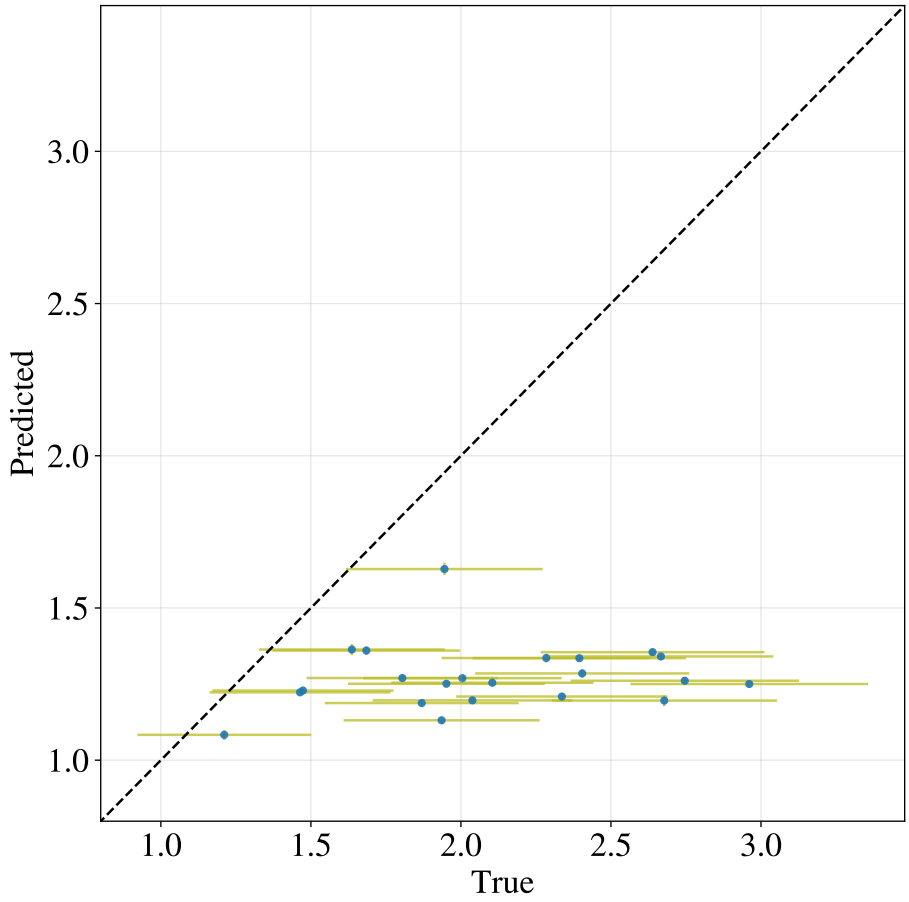

Figure 5.12 – Comparison of true and predicted values for the model trained with ground-truth values taken from the effective prior for the Einstein radius using data from the DELVE survey. The vertical bars represent the 1-sigma confidence interval. The uncertainties in true values are also reported.

Chapter 6

Concluding Remarks

In the next decade, the number of observed strong lensing systems is expected to increase by up to three orders of magnitude, with current estimates placing this number in the hundreds of thousands. This translates to the detection of dozens of lenses every day. The current methods for parameter estimation show reliable results, but are not suited to the new volume of data that future surveys can provide. Neural network based models such as the one discussed in this work are possible solutions, showing promising results. Our best performing model is capable of modeling a lensing system in 0.03 seconds, proving a suitable method of data analysis for the next generation of ground-based surveys.

Despite the suitability to the task, there are still challenges to the large-scale adoption of these methods as reliable parameter estimators. The models developed in this work are not yet at a point where they can be used to analyze real data and provide meaningful results. Nevertheless, this work serves as a step in the direction of the adoption of AI based methods to perform fast and automated Bayesian parameter estimation in Strong Lensing.

One of the most important factors that affect the performance of any SBI-based approach is the realism of the simulations. We have made an effort to go beyond the current available simulators, generating simulations that present many real-world effects, including survey-specific read noise, gain, sky brightness, zero-point magnitudes. Using real images of galaxies as a base for a simulated lens also introduces effects that are harder to simulate such as cosmic ray absorption. These effects must be present in the training data for any model robust enough to analyze real data. The difference between the results obtained in the training data and the real data show that there is still a significant difference between the simulations and the real data, an issue that must be addressed in future investigations.

Additionally, the results obtained in the training data are also not perfect. During the

development of the work, a significant trade off could be noticed between the realism of the simulations and the performance of the deep learning models. As the simulations were made mode complex, more advanced architectures were required to maintain the same level of performance. By utilizing more sophisticated deep learning models, it is likely that the performance could be further improved, potentially leading to more accurate predictions and better generalization across different simulation scenarios, even reducing the impact of imperfect simulations.

Finally, the impact of the sample size in the prior distribution shows that simulations that are significantly different from the average are unlikely to yield accurate classifications. This highlights one of the limitations of AI-based approaches when applied to large datasets, where rare or exceptional cases that deviate from the norm may suffer from reduced performance. Moreover, the influence of the sample size emphasizes the need for larger and more diverse datasets in future studies in order to enhance model reliability and performance.

In the coming decade, as the number of observed Strong Lensing systems increase, the need for a fast and automated approach to parameter inference will become very evident. While this work does not provide a fully developed, working model, it provides an important step towards the adoption of AI-based approaches to this problem. Significant challenges remain, especially regarding the realism of simulations and the need for more sophisticated models to improve accuracy. Harnessing the benefits of AI-based methods is crucial for the future of scientific discovery, but it is necessary to ensure that their results are reliable. While this work is not yet a final solution, it lays the foundation for the development of more robust, reliable, and scalable models to tackle parameter inference in Strong Gravitational Lensing.

References

[1] P. Schneider, J. Ehlers, and E.E. Falco. Gravitational Lenses. Astronomy and Astrophysics Library. Springer Berlin Heidelberg, 2013.

[2] A.O. Petters, H. Levine, and J. Wambsganss. Singularity Theory and Gravitational Lensing. Progress in Mathematical Physics. Birkhäuser Boston, 2012.

[3] Peter Schneider, Christopher Kochanek, and Joachim Wambsganss. Gravitational Lensing: Strong, weak and micro. Saas-Fee Advanced Course. Springer, Berlin, Germany, 2006 edition, May 2006.

[4] Tommaso Treu and Léon V. E. Koopmans. The Internal Structure and Formation of Early-Type Galaxies: The Gravitational Lens System MG 2016+112 at z = 1.004*. The Astrophysical Journal, 575(1):87, aug 2002.

[5] T. Treu and L. V. E. Koopmans. The internal structure of the lens PG1115+080: breaking degeneracies in the value of the Hubble constant. Monthly Notices of the Royal Astronomical Society, 337(2):L6–L10, 12 2002.

[6] Tommaso Treu and Léon V. E. Koopmans. Massive Dark Matter Halos and Evolution of Early-Type Galaxies to z ≈ 1*. The Astrophysical Journal, 611(2):739, aug 2004.

[7] M. W. Auger, T. Treu, A. S. Bolton, R. Gavazzi, L. V. E. Koopmans, P. J. Marshall, L. A. Moustakas, and S. Burles. The Sloan Lens Acs Survey. X. Stellar, Dynamical, and Total Mass Correlations of Massive Early-Type Galaxies. The Astrophysical Journal, 724(1):511, nov 2010.

[8] Andrew B. Newman, Richard S. Ellis, and Tommaso Treu. Luminous and Dark Matter Profiles From Galaxies to Clusters: Bridging the Gap With Group-Scale Lenses. The Astrophysical Journal, 814(1):26, nov 2015.

[9] S. H. Suyu, P. J. Marshall, M. W. Auger, S. Hilbert, R. D. Blandford, L. V. E. Koopmans, C. D. Fassnacht, and T. Treu. Dissecting the gravitational lens B1608+656. II. Precision measurements of the Hubble constant, spatial curvature, and the dark energy equation of state. The Astrophysical Journal, 711(1):201, feb 2010.

[10] Kenneth C Wong, Sherry H Suyu, Geoff C-F Chen, Cristian E Rusu, Martin Millon, Dominique Sluse, Vivien Bonvin, Christopher D Fassnacht, Stefan Taubenberger, Matthew W Auger, Simon Birrer, James H H Chan, Frederic Courbin, Stefan Hilbert, Olga Tihhonova, Tommaso Treu, Adriano Agnello, Xuheng Ding, Inh Jee, Eiichiro Komatsu, Anowar J Shajib, Alessandro Sonnenfeld, Roger D Blandford, Léon V E Koopmans, Philip J Marshall, and Georges Meylan. H0LiCOW – XIII. A 2.4 per cent measurement of H0 from lensed quasars: 5.3 σ tension between early- and late-Universe probes. Monthly Notices of the Royal Astronomical Society, 498(1):1420–1439, 09 2019.

[11] A J Shajib, S Birrer, T Treu, A Agnello, E J Buckley-Geer, J H H Chan, L Christensen, C Lemon, H Lin, M Millon, J Poh, C E Rusu, D Sluse, C Spiniello, G C-F Chen, T Collett, F Courbin, C D Fassnacht, J Frieman, A Galan, D Gilman, A More, T Anguita, M W Auger, V Bonvin, R McMahon, G Meylan, K C Wong, T M C Abbott, J Annis, S Avila, K Bechtol, D Brooks, D Brout, D L Burke, A Carnero Rosell, M Carrasco Kind, J Carretero, F J Castander, M Costanzi, L N da Costa, J De Vicente, S Desai, J P Dietrich, P Doel, A Drlica-Wagner, A E Evrard, D A Finley, B Flaugher, P Fosalba, J García-Bellido, D W Gerdes, D Gruen, R A Gruendl, J Gschwend, G Gutierrez, D L Hollowood, K Honscheid, D Huterer, D J James, T Jeltema, E Krause, N Kuropatkin, T S Li, M Lima, N MacCrann, M A G Maia, J L Marshall, P Melchior, R Miquel, R L C Ogando, A Palmese, F Paz-Chinchón, A A Plazas, A K Romer, A Roodman, M Sako, E Sanchez, B Santiago, V Scarpine, M Schubnell, D Scolnic, S Serrano, I Sevilla-Noarbe, M Smith, M Soares-Santos, E Suchyta, G Tarle, D Thomas, A R Walker, and Y Zhang. STRIDES: a 3.9 per cent measurement of the Hubble constant from the strong lens system DES J0408-5354. Monthly Notices of the Royal Astronomical Society, 494(4):6072–6102, 03 2020.

[12] H. Ebeling, M. Stockmann, J. Richard, J. Zabl, G. Brammer, S. Toft, and A. Man. Thirty-fold: Extreme gravitational lensing of a quiescent galaxy at z = 1.6. The Astrophysical Journal Letters, 852(1):L7, dec 2017.

[13] Johan Richard, Tucker Jones, Richard Ellis, Daniel P. Stark, Rachael Livermore, and Mark Swinbank. The emission line properties of gravitationally lensed $1.5 < z < 5$ galaxies. Monthly Notices of the Royal Astronomical Society, 413(1):643–658, 04 2011.

[14] Dark Energy Survey Collaboration:, T. Abbott, F. B. Abdalla, J. Aleksić, S. Allam, A. Amara, D. Bacon, E. Balbinot, M. Banerji, K. Bechtol, A. Benoit-Lévy, G. M. Bernstein, E. Bertin, J. Blazek, C. Bonnett, S. Bridle, D. Brooks, R. J. Brunner, E. Buckley-Geer, D. L. Burke, G. B. Caminha, D. Capozzi, J. Carlsen,

A. Carnero-Rosell, M. Carollo, M. Carrasco-Kind, J. Carretero, F. J. Castander, L. Clerkin, T. Collett, C. Conselice, M. Crocce, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, T. M. Davis, S. Desai, H. T. Diehl, J. P. Dietrich, S. Dodelson, P. Doel, A. Drlica-Wagner, J. Estrada, J. Etherington, A. E. Evrard, J. Fabbri, D. A. Finley, B. Flaugher, R. J. Foley, P. Fosalba, J. Frieman, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, T. Giannantonio, D. A. Goldstein, D. Gruen, R. A. Gruendl, P. Guarnieri, G. Gutierrez, W. Hartley, K. Honscheid, B. Jain, D. J. James, T. Jeltema, S. Jouvel, R. Kessler, A. King, D. Kirk, R. Kron, K. Kuehn, N. Kuropatkin, O. Lahav, T. S. Li, M. Lima, H. Lin, M. A. G. Maia, M. Makler, M. Manera, C. Maraston, J. L. Marshall, P. Martini, R. G. McMahon, P. Melchior, A. Merson, C. J. Miller, R. Miquel, J. J. Mohr, X. Morice-Atkinson, K. Naidoo, E. Neilsen, R. C. Nichol, B. Nord, R. Ogando, F. Ostrovski, A. Palmese, A. Papadopoulos, H. V. Peiris, J. Peoples, W. J. Percival, A. A. Plazas, S. L. Reed, A. Refregier, A. K. Romer, A. Roodman, A. Ross, E. Rozo, E. S. Rykoff, I. Sadeh, M. Sako, C. Sánchez, E. Sanchez, B. Santiago, V. Scarpine, M. Schubnell, I. Sevilla-Noarbe, E. Sheldon, M. Smith, R. C. Smith, M. Soares-Santos, F. Sobreira, M. Soumagnac, E. Suchyta, M. Sullivan, M. Swanson, G. Tarle, J. Thaler, D. Thomas, R. C. Thomas, D. Tucker, J. D. Vieira, V. Vikram, A. R. Walker, R. H. Wechsler, J. Weller, W. Wester, L. Whiteway, H. Wilcox, B. Yanny, Y. Zhang, and J. Zuntz. The Dark Energy Survey: more than dark energy – an overview. Monthly Notices of the Royal Astronomical Society, 460(2):1270–1299, 03 2016.

[15] Željko Ivezić, Steven M. Kahn, J. Anthony Tyson, Bob Abel, Emily Acosta, Robyn Allsman, David Alonso, Yusra AlSayyad, Scott F. Anderson, John Andrew, James Roger P. Angel, George Z. Angeli, Reza Ansari, Pierre Antilogus, Constanza Araujo, Robert Armstrong, Kirk T. Arndt, Pierre Astier, Éric Aubourg, Nicole Auza, Tim S. Axelrod, Deborah J. Bard, Jeff D. Barr, Aurelian Barrau, James G. Bartlett, Amanda E. Bauer, Brian J. Bauman, Sylvain Baumont, Ellen Bechtol, Keith Bechtol, Andrew C. Becker, Jacek Becla, Cristina Beldica, Steve Bellavia, Federica B. Bianco, Rahul Biswas, Guillaume Blanc, Jonathan Blazek, Roger D. Blandford, Josh S. Bloom, Joanne Bogart, Tim W. Bond, Michael T. Booth, Anders W. Borgland, Kirk Borne, James F. Bosch, Dominique Boutigny, Craig A. Brackett, Andrew Bradshaw, William Nielsen Brandt, Michael E. Brown, James S. Bullock, Patricia Burchat, David L. Burke, Gianpietro Cagnoli, Daniel Calabrese, Shawn Callahan, Alice L. Callen, Jeffrey L. Carlin, Erin L. Carlson, Srinivasan Chandrasekharan, Glenaver Charles-Emerson, Steve Chesley, Elliott C. Cheu, Hsin-Fang Chiang, James Chiang, Carol Chirino, Derek Chow, David R. Ciardi, Charles F. Claver, Johann Cohen-Tanugi, Joseph J. Cockrum, Rebecca Coles, Andrew J. Connolly, Kem H. Cook, Asantha Cooray, Kevin R. Covey, Chris Cribbs, Wei Cui, Roc Cutri, Philip N. Daly, Scott F. Daniel, Felipe Daruich, Guillaume Daubard, Greg Daues, William Dawson, Francisco

Delgado, Alfred Dellapenna, Robert de Peyster, Miguel de Val-Borro, Seth W. Digel, Peter Doherty, Richard Dubois, Gregory P. Dubois-Felsmann, Josef Durech, Frossie Economou, Tim Eifler, Michael Eracleous, Benjamin L. Emmons, Angelo Fausti Neto, Henry Ferguson, Enrique Figueroa, Merlin Fisher-Levine, Warren Focke, Michael D. Foss, James Frank, Michael D. Freemon, Emmanuel Gangler, Eric Gawiser, John C. Geary, Perry Gee, Marla Geha, Charles J. B. Gessner, Robert R. Gibson, D. Kirk Gilmore, Thomas Glanzman, William Glick, Tatiana Goldina, Daniel A. Goldstein, Iain Goodenow, Melissa L. Graham, William J. Gressler, Philippe Gris, Leanne P. Guy, Augustin Guyonnet, Gunther Haller, Ron Harris, Patrick A. Hascall, Justine Haupt, Fabio Hernandez, Sven Herrmann, Edward Hileman, Joshua Hoblitt, John A. Hodgson, Craig Hogan, James D. Howard, Dajun Huang, Michael E. Huffer, Patrick Ingraham, Walter R. Innes, Suzanne H. Jacoby, Bhuvnesh Jain, Fabrice Jammes, M. James Jee, Tim Jenness, Garrett Jernigan, Darko Jevremović, Kenneth Johns, Anthony S. Johnson, Margaret W. G. Johnson, R. Lynne Jones, Claire Juramy-Gilles, Mario Jurić, Jason S. Kalirai, Nitya J. Kallivayalil, Bryce Kalmbach, Jeffrey P. Kantor, Pierre Karst, Mansi M. Kasliwal, Heather Kelly, Richard Kessler, Veronica Kinnison, David Kirkby, Lloyd Knox, Ivan V. Kotov, Victor L. Krabbendam, K. Simon Krughoff, Petr Kubánek, John Kuczewski, Shri Kulkarni, John Ku, Nadine R. Kurita, Craig S. Lage, Ron Lambert, Travis Lange, J. Brian Langton, Laurent Le Guillou, Deborah Levine, Ming Liang, Kian-Tat Lim, Chris J. Lintott, Kevin E. Long, Margaux Lopez, Paul J. Lotz, Robert H. Lupton, Nate B. Lust, Lauren A. MacArthur, Ashish Mahabal, Rachel Mandelbaum, Thomas W. Markiewicz, Darren S. Marsh, Philip J. Marshall, Stuart Marshall, Morgan May, Robert McKercher, Michelle McQueen, Joshua Meyers, Myriam Migliore, Michelle Miller, David J. Mills, Connor Miraval, Joachim Moeyens, Fred E. Moolekamp, David G. Monet, Marc Moniez, Serge Monkewitz, Christopher Montgomery, Christopher B. Morrison, Fritz Mueller, Gary P. Muller, Freddy Muñoz Arancibia, Douglas R. Neill, Scott P. Newbry, Jean-Yves Nief, Andrei Nomerotski, Martin Nordby, Paul O'Connor, John Oliver, Scot S. Olivier, Knut Olsen, William O'Mullane, Sandra Ortiz, Shawn Osier, Russell E. Owen, Reynald Pain, Paul E. Palecek, John K. Parejko, James B. Parsons, Nathan M. Pease, J. Matt Peterson, John R. Peterson, Donald L. Petravick, M. E. Libby Petrick, Cathy E. Petry, Francesco Pierfederici, Stephen Pietrowicz, Rob Pike, Philip A. Pinto, Raymond Plante, Stephen Plate, Joel P. Plutchak, Paul A. Price, Michael Prouza, Veljko Radeka, Jayadev Rajagopal, Andrew P. Rasmussen, Nicolas Regnault, Kevin A. Reil, David J. Reiss, Michael A. Reuter, Stephen T. Ridgway, Vincent J. Riot, Steve Ritz, Sean Robinson, William Roby, Aaron Roodman, Wayne Rosing, Cecille Roucelle, Matthew R. Rumore, Stefano Russo, Abhijit Saha, Benoit Sassolas, Terry L. Schalk, Pim Schellart, Rafe H. Schindler, Samuel Schmidt, Donald P. Schneider, Michael D. Schneider, William Schoening, German

Schumacher, Megan E. Schwamb, Jacques Sebag, Brian Selvy, Glenn H. Sembroski, Lynn G. Seppala, Andrew Serio, Eduardo Serrano, Richard A. Shaw, Ian Shipsey, Jonathan Sick, Nicole Silvestri, Colin T. Slater, J. Allyn Smith, R. Chris Smith, Shahram Sobhani, Christine Soldahl, Lisa Storrie-Lombardi, Edward Stover, Michael A. Strauss, Rachel A. Street, Christopher W. Stubbs, Ian S. Sullivan, Donald Sweeney, John D. Swinbank, Alexander Szalay, Peter Takacs, Stephen A. Tether, Jon J. Thaler, John Gregg Thayer, Sandrine Thomas, Adam J. Thornton, Vaikunth Thukral, Jeffrey Tice, David E. Trilling, Max Turri, Richard Van Berg, Daniel Vanden Berk, Kurt Vetter, Francoise Virieux, Tomislav Vucina, William Wahl, Lucianne Walkowicz, Brian Walsh, Christopher W. Walter, Daniel L. Wang, Shin-Yawn Wang, Michael Warner, Oliver Wiecha, Beth Willman, Scott E. Winters, David Wittman, Sidney C. Wolff, W. Michael Wood-Vasey, Xiuqin Wu, Bo Xin, Peter Yoachim, and Hu Zhan. LSST: From Science Drivers to Reference Design and Anticipated Data Products. The Astrophysical Journal, 873(2):111, March 2019.

[16] R. Scaramella, J. Amiaux, Y. Mellier, C. Burigana, C. S. Carvalho, J.-C. Cuillandre, A. Da Silva, A. Derosa, J. Dinis, E. Maiorano, M. Maris, I. Tereno, R. Laureijs, T. Boenke, G. Buenadicha, X. Dupac, L. M. Gaspar Venancio, P. Gómez-Álvarez, J. Hoar, J. Lorenzo Alvarez, G. D. Racca, G. Saavedra-Criado, J. Schwartz, R. Vavrek, M. Schirmer, H. Aussel, R. Azzollini, V. F. Cardone, M. Cropper, A. Ealet, B. Garilli, W. Gillard, B. R. Granett, L. Guzzo, H. Hoekstra, K. Jahnke, T. Kitching, T. Maciaszek, M. Meneghetti, L. Miller, R. Nakajima, S. M. Niemi, F. Pasian, W. J. Percival, S. Pottinger, M. Sauvage, M. Scodeggio, S. Wachter, A. Zacchei, N. Aghanim, A. Amara, T. Auphan, N. Auricchio, S. Awan, A. Balestra, R. Bender, C. Bodendorf, D. Bonino, E. Branchini, S. Brau-Nogue, M. Brescia, G. P. Candini, V. Capobianco, C. Carbone, R. G. Carlberg, J. Carretero, R. Casas, F. J. Castander, M. Castellano, S. Cavuoti, A. Cimatti, R. Cledassou, G. Congedo, C. J. Conselice, L. Conversi, Y. Copin, L. Corcione, A. Costille, F. Courbin, H. Degaudenzi, M. Douspis, F. Dubath, C. A. J. Duncan, S. Dusini, S. Farrens, S. Ferriol, P. Fosalba, N. Fourmanoit, M. Frailis, E. Franceschi, P. Franzetti, M. Fumana, B. Gillis, C. Giocoli, A. Grazian, F. Grupp, S. V. H. Haugan, W. Holmes, F. Hormuth, P. Hudelot, S. Kermiche, A. Kiessling, M. Kilbinger, R. Kohley, B. Kubik, M. Kümmel, M. Kunz, H. Kurki-Suonio, O. Lahav, S. Ligori, P. B. Lilje, I. Lloro, O. Mansutti, O. Marggraf, K. Markovic, F. Marulli, R. Massey, S. Maurogordato, M. Melchior, E. Merlin, G. Meylan, J. J. Mohr, M. Moresco, B. Morin, L. Moscardini, E. Munari, R. C. Nichol, C. Padilla, S. Paltani, J. Peacock, K. Pedersen, V. Pettorino, S. Pires, M. Poncet, L. Popa, L. Pozzetti, F. Raison, R. Rebolo, J. Rhodes, H.-W. Rix, M. Roncarelli, E. Rossetti, R. Saglia, P. Schneider, T. Schrabback, A. Secroun, G. Seidel, S. Serrano, C. Sirignano, G. Sirri, J. Skottfelt, L. Stanco, J. L. Starck, P. Tallada-Crespí, D. Tavagnacco, A. N. Taylor, H. I. Teplitz, R. Toledo-Moreo,

F. Torradeflot, M. Trifoglio, E. A. Valentijn, L. Valenziano, G. A. Verdoes Kleijn, Y. Wang, N. Welikala, J. Weller, M. Wetzstein, G. Zamorani, J. Zoubian, S. Andreon, M. Baldi, S. Bardelli, A. Boucaud, S. Camera, D. Di Ferdinando, G. Fabbian, R. Farinelli, S. Galeotta, J. Graciá-Carpio, D. Maino, E. Medinaceli, S. Mei, C. Neissner, G. Polenta, A. Renzi, E. Romelli, C. Rosset, F. Sureau, M. Tenti, T. Vassallo, E. Zucca, C. Baccigalupi, A. Balaguera-Antolínez, P. Battaglia, A. Biviano, S. Borgani, E. Bozzo, R. Cabanac, A. Cappi, S. Casas, G. Castignani, C. Colodro-Conde, J. Coupon, H. M. Courtois, J. Cuby, S. de la Torre, S. Desai, H. Dole, M. Fabricius, M. Farina, P. G. Ferreira, F. Finelli, P. Flose-Reimberg, S. Fotopoulou, K. Ganga, G. Gozaliasl, I. M. Hook, E. Keihanen, C. C. Kirkpatrick, P. Liebing, V. Lindholm, G. Mainetti, M. Martinelli, N. Martinet, M. Maturi, H. J. McCracken, R. B. Metcalf, G. Morgante, J. Nightingale, A. Nucita, L. Patrizii, D. Potter, G. Riccio, A. G. Sánchez, D. Sapone, J. A. Schewtschenko, M. Schultheis, V. Scottez, R. Teyssier, I. Tutusaus, J. Valiviita, M. Viel, W. Vriend, and L. Whittaker. Euclid preparation: I. The Euclid Wide Survey. Astronomy & Astrophysics, 662:A112, June 2022.

[17] Thomas E. Collett. The population of galaxy-galaxy strong lenses in forthcoming optical imaging surveys. The Astrophysical Journal, 811(1):20, September 2015. arXiv:1507.02657 [astro-ph].

[18] Yashar D. Hezaveh, Laurence Perreault Levasseur, and Philip J. Marshall. Fast Automated Analysis of Strong Gravitational Lenses with Convolutional Neural Networks. Nature, 548(7669):555–557, August 2017. arXiv:1708.08842 [astro-ph].

[19] Clecio Bom, Jason Poh, Brian Nord, Manuel Blanco-Valentin, and Luciana Dias. Deep Learning in Wide-field Surveys: Fast Analysis of Strong Lenses in Ground-based Cosmic Experiments, November 2019. arXiv:1911.06341 [astro-ph].

[20] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43:3964–3979, November 2021. arXiv:1908.09257 [cs, stat].

[21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference, April 2021. arXiv:1912.02762 [cs, stat].

[22] David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic Posterior Transformation for Likelihood-Free Inference, May 2019. arXiv:1905.07488 [cs, stat].

[23] Laurence Perreault Levasseur, Yashar D. Hezaveh, and Risa H. Wechsler. Uncertainties in Parameters Estimated with Neural Networks: Application to Strong

Gravitational Lensing. The Astrophysical Journal, 850(1):L7, November 2017. arXiv:1708.08843 [astro-ph].

[24] Jason Poh, Ashwin Samudre, Aleksandra Ćiprijanović, Brian Nord, Gourav Khullar, Dimitrios Tanoglidis, and Joshua A. Frieman. Strong Lensing Parameter Estimation on Ground-Based Imaging Data Using Simulation-Based Inference, November 2022. arXiv:2211.05836 [astro-ph].

[25] Ronan Legin, Yashar Hezaveh, Laurence Perreault Levasseur, and Benjamin Wandelt. Simulation-Based Inference of Strong Gravitational Lensing Parameters, June 2022. arXiv:2112.05278 [astro-ph].

[26] Isaac Newton, G. W. Hemming, and donor DSI Burndy Library. Opticks: or, A treatise of the reflections, refractions, inflexions and colours of light : also two treatises of the species and magnitude of curvilinear figures. London : Printed for Sam. Smith, and Benj. Walford ..., 1704.

[27] Stanley L. Jaki. Johann georg von soldner and the gravitational bending of light, with an english translation of his essay on it published in 1801. Foundations of Physics, 8(11-12):927–950, 1978.

[28] F. Zwicky. Nebulae as gravitational lenses. Phys. Rev., 51:290–290, Feb 1937.

[29] F. Zwicky. On the Masses of Nebulae and of Clusters of Nebulae. ApJ, 86:217, October 1937.

[30] Massimo Meneghetti. Introduction to gravitational lensing.

[31] Aggeliki Kassiola and Israel Kovner. Elliptic mass distributions versus elliptic potentials in gravitational lenses. The Astrophysical Journal, 417(2):450–473, 1993.

[32] R. Kormann, P. Schneider, and M. Bartelmann. Isothermal elliptical gravitational lens models. AAP, 284:285–299, April 1994.

[33] Glenn van de Ven, Rachel Mandelbaum, and Charles R. Keeton. Galaxy density profiles and shapes - I. Simulation pipeline for lensing by realistic galaxy models. MNRAS, 398(2):607–634, September 2009.

[34] C. R. Keeton and C. S. Kochanek. Gravitational lensing by spiral galaxies. The Astrophysical Journal, 495(1):157, mar 1998.

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

[36] Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., 2nd edition, 2019.

[37] B J Copeland. The Essential Turing. Oxford University Press, 09 2004.

[38] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

[39] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 1958.

[40] M.L. Minsky and S. Papert. Perceptrons: An Introduction to Computational Geometry. MIT Press, 1969.

[41] Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach (4th Edition). Pearson, 2020.

[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[44] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 11 1997.

[45] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into Deep Learning. Cambridge University Press, 2023. https://D2L.ai.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[47] Daniel George and E. A. Huerta. Deep Neural Networks to Enable Real-time Multimessenger Astrophysics. Phys. Rev. D, 97(4):044039, 2018.

[48] Daniel George and E. A. Huerta. Deep Learning for Real-time Gravitational Wave Detection and Parameter Estimation: Results with Advanced LIGO Data. Phys. Lett. B, 778:64–70, 2018.

[49] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. Rev. Mod. Phys., 91(4):045002, 2019.

[50] C. R. Bom, A. Cortesi, U. Ribeiro, L. O. Dias, K. Kelkar, A. V. Smith Castelli, L. Santana-Silva, V. Silva, T. S. Gonçalves, L. R. Abramo, E. V. R. Lima, F. Almeida-Fernandes, L. Espinosa, L. Li, M. L. Buzzo, C. Mendes de Oliveira, L. Sodré Jr., A. Alvarez-Candal, M. Grossi, E. Telles, S. Torres-Flores, S. V. Werner, A. Kanaan, T. Ribeiro, and W. Schoenell. An extended catalogue of galaxy morphology using deep learning in southern photometric local universe survey data release 3, 2023.

[51] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks, 2022.

[52] Ankush Ganguly and Samuel W. F. Earp. An introduction to variational inference, 2021.

[53] Ethan Goan and Clinton Fookes. Bayesian Neural Networks: An Introduction and Survey, page 45–87. Springer International Publishing, 2020.

[54] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. Proceedings of the National Academy of Sciences, 117(48):30055–30062, 2020.

[55] David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic Posterior Transformation for Likelihood-Free Inference, May 2019. arXiv:1905.07488 [cs, stat].

[56] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43:3964–3979, November 2021. arXiv:1908.09257 [cs, stat].

[57] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference, April 2021. arXiv:1912.02762 [cs, stat].

[58] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

[59] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked Autoregressive Flow for Density Estimation, June 2018. arXiv:1705.07057 [cs, stat].

[60] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows, December 2019. arXiv:1906.04032 [cs, stat] version: 2.

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

[62] José-Víctor Rodríguez, Ignacio Rodríguez-Rodríguez, and Wai Lok Woo. On the application of machine learning in astronomy and astrophysics: A text-mining-based scientometric analysis. WIREs Data Mining and Knowledge Discovery, 12(5):e1476, 2022.

[63] A. Drlica-Wagner, J. L. Carlin, D. L. Nidever, P. S. Ferguson, N. Kuropatkin, M. Adamów, W. Cerny, Y. Choi, J. H. Esteves, C. E. Martínez-Vázquez, S. Mau, A. E. Miller, B. Mutlu-Pakdil, E. H. Neilsen, K. A. G. Olsen, A. B. Pace, A. H. Riley, J. D. Sakowska, D. J. Sand, L. Santana-Silva, E. J. Tollerud, D. L. Tucker, A. K. Vivas, E. Zaborowski, A. Zenteno, T. M. C. Abbott, S. Allam, K. Bechtol, C. P. M. Bell, E. F. Bell, P. Bilaji, C. R. Bom, J. A. Carballo-Bello, M.-R. L. Cioni, A. Diaz-Ocampo, T. J. L. de Boer, D. Erkal, R. A. Gruendl, D. Hernandez-Lang, A. K. Hughes, D. J. James, L. C. Johnson, T. S. Li, Y.-Y. Mao, D. Martínez-Delgado, P. Massana, M. McNanna, R. Morgan, E. O. Nadler, N. E. D. Noël, A. Palmese, A. H. G. Peter, E. S. Rykoff, J. Sánchez, N. Shipp, J. D. Simon, A. Smercina, M. Soares-Santos, G. S. Stringfellow, K. Tavangar, R. P. van der Marel, A. R. Walker, R. H. Wechsler, J. F. Wu, B. Yanny, M. Fitzpatrick, L. Huang, A. Jacques, R. Nikutta, and A. Scott. The DECam Local Volume Exploration Survey: Overview and First Data Release. The Astrophysical Journal Supplement Series, 256(1):2, September 2021. arXiv:2103.07476 [astro-ph].

[64] Simon Birrer, Adam Amara, and Alexandre Refregier. Gravitational Lens Modeling with Basis Sets. ApJ, 813(2):102, November 2015.

[65] Simon Birrer and Adam Amara. Lenstronomy: multi-purpose gravitational lens modelling software package, 2018.

[66] Simon Birrer, Anowar J. Shajib, Daniel Gilman, Aymeric Galan, Jelle Aalbers, Martin Millon, Robert Morgan, Giulia Pagano, Ji Won Park, Luca Teodori, Nicolas Tessore, Madison Ueland, Lyne Van de Vyvere, Sebastian Wagner-Carena, Ewoud Wempe, Lilan Yang, Xuheng Ding, Thomas Schmidt, Dominique Sluse, Ming Zhang, and Adam Amara. lenstronomy ii: A gravitational lensing software ecosystem. Journal of Open Source Software, 6(62):3283, 2021.

[67] E. A. Zaborowski, A. Drlica-Wagner, F. Ashmead, J. F. Wu, R. Morgan, C. R. Bom, A. J. Shajib, S. Birrer, W. Cerny, E. J. Buckley-Geer, B. Mutlu-Pakdil, P. S. Ferguson, K. Glazebrook, S. J. Gonzalez Lozano, Y. Gordon, M. Martinez, V. Manwadkar, J. O'Donnell, J. Poh, A. Riley, J. D. Sakowska, L. Santana-Silva, B. X. Santiago, D. Sluse, C. Y. Tan, E. J. Tollerud, A. Verma, J. A. Carballo-Bello, Y. Choi, D. J. James, N. Kuropatkin, C. E. Martínez-Vázquez, D. L. Nidever, J. L. Nilo Castellon, N. E. D. Noël, K. A. G. Olsen, A. B. Pace, S. Mau, B. Yanny, A. Zenteno, T. M. C. Abbott, M. Aguena, O. Alves, F. Andrade-Oliveira, S. Bocquet, D. Brooks, D. L.

Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, F. J. Castander, C. J. Conselice, M. Costanzi, M. E. S. Pereira, J. De Vicente, S. Desai, J. P. Dietrich, P. Doel, S. Everett, I. Ferrero, B. Flaugher, D. Friedel, J. Frieman, J. García-Bellido, D. Gruen, R. A. Gruendl, G. Gutierrez, S. R. Hinton, D. L. Hollowood, K. Honscheid, K. Kuehn, H. Lin, J. L. Marshall, P. Melchior, J. Mena-Fernández, F. Menanteau, R. Miquel, A. Palmese, F. Paz-Chinchón, A. Pieres, A. A. Plazas Malagón, J. Prat, M. Rodriguez-Monroy, A. K. Romer, E. Sanchez, V. Scarpine, I. Sevilla-Noarbe, M. Smith, E. Suchyta, C. To, N. Weaverdyck, and (DELVE & DES Collaborations). Identification of Galaxy–Galaxy Strong Lens Candidates in the DECam Local Volume Exploration Survey Using Machine Learning. The Astrophysical Journal, 954(1):68, aug 2023.

[68] Philip J. Marshall, Aprajita Verma, Anupreeta More, Christopher P. Davis, Surhud More, Amit Kapadia, Michael Parrish, Chris Snyder, Julianne Wilcox, Elisabeth Baeten, Christine Macmillan, Claude Cornen, Michael Baumer, Edwin Simpson, Chris J. Lintott, David Miller, Edward Paget, Robert Simpson, Arfon M. Smith, Rafael Küng, Prasenjit Saha, and Thomas E. Collett. Space Warps – I. Crowdsourcing the discovery of gravitational lenses. Monthly Notices of the Royal Astronomical Society, 455(2):1171–1190, 11 2015.

[69] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions, 2014.

[70] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[71] Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful, 2022.

[72] Samantha R Cook, Andrew Gelman, and Donald B Rubin. Validation of software for bayesian models using posterior quantiles. Journal of Computational and Graphical Statistics, 15(3):675–692, 2006.

[73] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, 2020.

[74] David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests, 2018.

[75] E.V.R. Lima, L. Sodré, C.R. Bom, G.S.M. Teixeira, L. Nakazono, M.L. Buzzo, C. Queiroz, F.R. Herpich, J.L. Nilo Castellon, M.L.L. Dantas, O.L. Dors, R.C.

Thom de Souza, S. Akras, Y. Jiménez-Teja, A. Kanaan, T. Ribeiro, and W. Schoennell. Photometric redshifts for the s-plus survey: Is machine learning up to the task? Astronomy and Computing, 38:100510, January 2022.

[76] G. Teixeira, C. R. Bom, L. Santana-Silva, B. M. O. Fraga, P. Darc, R. Teixeira, J. F. Wu, P. S. Ferguson, C. E. Martínez-Vázquez, A. H. Riley, A. Drlica-Wagner, Y. Choi, B. Mutlu-Pakdil, A. B. Pace, J. D. Sakowska, and G. S. Stringfellow. Photometric redshifts probability density estimation from recurrent neural networks in the decam local volume exploration survey data release 2, 2024.

[77] João Paulo C. França, Martin Makler, Ingrid Beloto, Eduardo Cypriano, Renan A. Oliveira, Thiago S. Gonçalves, and James Nightingale. The last stand before rubin: semi-automated inverse modelling of galaxy-galaxy strong lensing systems. Proceedings of the International Astronomical Union, 18(S381):31–34, 2022.