Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **522,** 5442–5455 (2023) https://doi.org/10.1093/mnras/stad1325
Advance Access publication 2023 May 03

# LEMON: LEns MOdelling with Neural networks – I. Automated modelling of strong gravitational lenses with Bayesian Neural Networks

Fabrizio Gentile [1,2]★ Crescenzo Tortora [3] Giovanni Covone [3,4,5] Léon V. E. Koopmans [6] Rui Li,[7] Laura Leuzzi[1,2] and Nicola R. Napolitano [3,7]

[1]*Department of Physics and Astronomy (DIFA), University of Bologna, Via Gobetti 93/2, I-40129 Bologna, Italy*
[2]*INAF – Osservatorio di Astrofisica e Scienza dello Spazio, Via Gobetti 93/3, I-40129 Bologna, Italy*
[3]*INAF – Osservatorio Astronomico di Capodimonte, Salita Moiariello, 16, I-80131 Napoli, Italy*
[4]*Dipartimento di Fisica 'Ettore Pancini', Università di Napoli Federico II, Compl. Univ. Monte S. Angelo, Via Cinthia, I-80126 Napoli, Italy*
[5]*INFN, Sezione di Napoli, C.U. Monte S. Angelo, Via Cinthia, I-80126 Napoli, Italy*
[6]*Kapteyn Astronomical Institute, University of Groningen, P.O Box 800, NL-9700 AV Groningen, the Netherlands*
[7]*School of Physics and Astronomy, Sun Yat-sen University Zhuhai Campus, Daxue Road 2, 519082 Tangjia, Zhuhai, Guangdong, China*

Accepted 2023 April 24. Received 2023 April 21; in original form 2022 October 19

## ABSTRACT

The unprecedented number of gravitational lenses expected from new-generation facilities such as the ESA *Euclid* telescope and the *Vera Rubin Observatory* makes it crucial to rethink our classical approach to lens-modelling. In this paper, we present LEMON (Lens Modelling with Neural networks): a new machine-learning algorithm able to analyse hundreds of thousands of gravitational lenses in a reasonable amount of time. The algorithm is based on a *Bayesian Neural Network*: a new generation of neural networks able to associate a reliable confidence interval to each predicted parameter. We train the algorithm to predict the three main parameters of the singular isothermal ellipsoid model (the Einstein radius and the two components of the ellipticity) by employing two simulated data sets built to resemble the imaging capabilities of the *Hubble Space Telescope* and the forthcoming *Euclid* satellite. In this work, we assess the accuracy of the algorithm and the reliability of the estimated uncertainties by applying the network to several simulated data sets of $10^4$ images each. We obtain accuracies comparable to previous studies present in the current literature and an average modelling time of just $\sim$0.5 s per lens. Finally, we apply the LEMON algorithm to a pilot data set of real lenses observed with HST during the SLACS program, obtaining unbiased estimates of their SIE parameters. The code is publicly available on GitHub (https://github.com/fab-gentile/LEMON).

**Key words:** gravitational lensing: strong – methods: data analysis – software: data analysis – galaxies: elliptical and lenticular, cD.

## 1 INTRODUCTION

Initially predicted by Albert Einstein as a consequence of his General Relativity (Einstein 1915), gravitational lensing consists in the deflection of light caused by a gravitational field. This phenomenon can be very intense when a massive object (e.g. a galaxy or a cluster) is involved. In this case, lensing can produce multiple images of a distant compact source (see e.g. Huchra et al. 1985; Napolitano et al. 2020) or gravitational arcs can be formed in case of extended sources (Zwicky 1937). In both these cases, the phenomenon is called 'strong gravitational lensing', while the composite systems created are generally known as 'gravitational lenses'. The main observables related to this phenomenon (i.e. the position and shape of the lensed images) mainly rely on the matter distribution inside the lensing galaxy and on the relative *angular-diameter distances* involving the observer, the deflector, and the background source (e.g. Schneider, Ehlers & Falco 1992; Bartelmann 2010). An additional contribution (up to 15 per cent; see e.g. Schneider et al. 1992) can also derive from the line-of-sight matter distribution. Strong lensing can be successfully employed to study the dark matter distribution in galaxies (e.g. Treu & Koopmans 2004; Covone et al. 2009; Tortora et al. 2010; Auger et al. 2010b; Spiniello et al. 2011; Sonnenfeld et al. 2015; Shajib et al. 2021) and – since the distances involved in lensing are sensitive to the cosmological parameters – to measure the Hubble constant with the so-called 'time-delay technique (e.g. Refsdal 1964; Wong et al. 2020). Further interesting applications of lensing span from identifying dark matter substructures in galaxies (e.g. Mao & Schneider 1998; Dalal & Kochanek 2002; Koopmans 2005; Vegetti et al. 2014) to studying the universality of the Initial Mass Function (e.g. Treu et al. 2010; Auger et al. 2010a; Barnabè et al. 2013; Sonnenfeld et al. 2019). A more comprehensive review of the possible scientific applications of strong lensing in modern astrophysics and cosmology can be found in Treu (2010) and Congdon & Keeton (2018).

For decades, however, the main limitation to the scientific potentialities of strong lensing has been represented by the small number of known and analysed gravitational lenses. The need for massive objects acting as lenses and an almost perfect alignment between observer, foreground lens, and background source make gravitational

★ E-mail: fabrizio.gentile3@unibo.it

lenses rare (e.g. Schneider et al. 1992). Several studies estimated that less than one galaxy out of $10^{3-4}$ shows detectable lensing features (see e.g. Collett 2015). Therefore, the inspection of large samples of astronomical sources is needed to identify a statistically significant sample of lenses.

This scenario, however, is about to change dramatically in the near future. A new generation of telescopes such as the ESA *Euclid* satellite (Laureijs et al. 2011) and the *Vera Rubin Observatory* (LSST Science Collaboration et al. 2009) will be soon on the scene. Several forecasts claimed how these facilities will provide data on more than one billion galaxies, allowing the identification of $\sim 10^5$ new homogeneously selected gravitational lenses (see e.g. LSST Science Collaboration et al. 2009; Laureijs et al. 2011; Serjeant 2014; Collett 2015). These newly discovered systems will increase by several orders of magnitude the number of known gravitational lenses, allowing many astrophysical studies to rely on a stronger statistics, and to explore a wider range of redshifts and galaxy properties. Moreover, a whole new approach to lensing-based science – the so-called 'statistical lensing' – will be possible (e.g. Sonnenfeld & Cautun 2021; Sonnenfeld 2021a, b, c).

Although these are important advances, the massive amount of data produced by these instruments would to overwhelm our ability to analyse them with classical techniques. For this reason, significant effort has been afforded in the last years to rapidly search for gravitational lenses in vast data sets, mainly thanks to the employment of machine learning algorithms (see e.g. the noteworthy results of the first *strong gravitational lens finding challenge* in Metcalf et al. 2019). Several sky surveys have been systematically analysed with these techniques, allowing the identification of an unprecedented number of likely lensed objects (see some examples in Petrillo et al. 2017, 2019a, b; Jacobs et al. 2019a, b; Canameras et al. 2020; He et al. 2020; Li et al. 2020; Gentile et al. 2022). Never the less, our ability to model lenses – necessary to allow the scientific exploitation of the retrieved systems – remained almost unchanged in the required compute time. Currently, most lens-modelling algorithms rely on Bayesian analysis, such as *Monte Carlo Markov Chains* or *nested sampling* (see some noteworthy examples in Jullo et al. 2007; Vegetti & Koopmans 2009; Birrer & Amara 2018; Nightingale, Dye & Massey 2018; Lefor, Futamase & Akhlaghi 2013 for a comparison between the different methods). These techniques are computationally expensive (the analysis of a single lens can require up to several hours) and often require a non-trivial human intervention to select the lensing feature in the image. These properties make them less suitable to efficiently model large samples of lenses. As it happened for the search for strong lenses, an efficient approach to lens-modelling can come from machine-learning. Several algorithms have been proposed. These are mainly based on *Convolutional Neural Networks* (CNNs; see e.g. Hezaveh, Perreault Levasseur & Marshall 2017; Pearson, Li & Dye 2019; Schuldt et al. 2021) or *Bayesian Neural Networks* (BNNs; e.g. Perreault Levasseur, Hezaveh & Wechsler 2017; Bom et al. 2019; Park et al. 2021; Pearson et al. 2021; Schuldt et al. 2023): supervised-learning algorithms able to extract meaningful features from images and to convert them into useful parameters (LeCun, Bengio & Hinton 2015; Charnock, Perreault-Levasseur & Lanusse 2020). In this work, we focus on BNNs. This new generation of machine learning algorithms is able to account, in its predictions, for the different sources of uncertainty that can affect the analysed data and the algorithm itself, providing a full Bayesian treatment of them. Thanks to this property, BNNs can associate reliable confidence intervals to the estimated parameters, allowing a wide range of scientific applications (see some astrophysical examples in Cobb et al. 2019; Escamilla-Rivera, Carvajal Quintero & Capozziello 2020; Hortúa et al. 2020; Wagner-Carena et al. 2021).

In this paper, we propose a new lens-modelling algorithm called 'LEMON'.[1] (LEns MOdelling with Neural Networks). Based on a BNN, it can efficiently model large samples of lenses in a reasonable amount of time. This work – the first of the LEMON series – presents the algorithm and its training, assessing its performances in modelling both simulations and a pilot sample of real lenses. The paper is structured as follows: Section 2 presents the data sets employed to train and test the algorithm. Section 3 introduces BNNs, and the architecture implemented in the LEMON algorithm. In Section 4, we assess the algorithm's performance on a simulated data set and on a small set of real lenses from the *Sloan Lens ACS Survey* (SLACS; Bolton et al. 2006). Section 5 discusses the results obtained on all the data sets and measures the performance on the estimation of different lens parameters. Finally, in Section 6, we draw our conclusions and list the future perspective of this work.

## 2 BUILDING THE TRAINING SET

Training a supervised-learning algorithm, such as a BNN, consists in passing large numbers of 'labelled' examples (i.e. images for which the ground truth is known) to the network. When dealing with strong gravitational lensing, this task is particularly challenging because of the small number of known and modelled gravitational lenses. The largest homogeneously selected collections of confirmed lenses (e.g. Bolton et al. 2006; Gavazzi et al. 2012; Sonnenfeld et al. 2015; Shu et al. 2017) contain only a few hundreds objects: at least two orders of magnitude less than the number of examples required to find the optimal value for the millions of free parameters inside a BNN (see e.g. the sizes of the data sets employed by Perreault Levasseur et al. 2017 or Bom et al. 2019). In addition, these samples do not homogeneously cover all possible lensing configurations, being generally biased towards systems with more prominent lensing features. Since supervised-learning algorithms mainly act as interpolators (i.e. they are sensitive only in the range of parameters well-covered by the training set; see e.g. Goodfellow, Bengio & Courville 2016), often these samples are not well suited as training sets. For the above reasons, most studies focused on the search and modelling of gravitational lenses with machine-learning methods rely on simulated data.

In this work, we simulate two different training sets. The first one is composed by *Euclid mocks*: it is built to resemble the observational characteristics expected from the *Euclid*'s VIS imager as forecasted by Laureijs et al. (2011) and Cropper et al. (2018). The second training set (composed by the *HST mocks*) mimics the imaging capabilities of the *Advanced Camera for Surveys* (ACS; Ryon 2022) mounted on the *Hubble Space Telescope* (HST). Both the data sets are composed by 40 000 simulated images each. Some examples of these images are reported in Fig. 1. We underline that in this work we assume that is possible to retrieve a completely pure sample of strong lenses (without any contaminant, quite common in samples selected through visual inspection or machine-learning techniques; see, e.g. Petrillo et al. 2017, 2019a, b; Metcalf et al. 2019; Gentile et al. 2022). The impact of contaminants on the algorithm's accuracy will be investigated in the forthcoming papers of the LEMON series.

[1] https://github.com/fab-gentile/LEMON

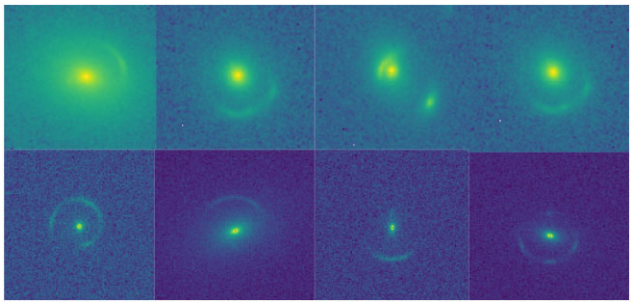

**Figure 1.** Some examples of the mock lenses employed in this work. The images in the top row are *HST mocks* and are simulated to recall the imaging capabilities of the ACS mounted on the HST. The images in the bottom row are *Euclid mocks* and mimic the observational characteristics of the systems observed with the forthcoming VIS imager of the *Euclid* telescope. Further details in Section 2.

**Table 1.** Distribution employed in Section 2 to randomly sample the parameters used during the simulations of the different training sets. All the parameters are uniformly sampled, except where indicated. Further details in Section 2.

| Parameter | Range | Units |
|---|---|---|
| | Lens (SIE) | |
| Einstein radius[(a)] | 0.5–3.0 | arcsec |
| Axial ratio | 0.3–1.0 | – |
| Major-axis angle | −90–90 | degrees |
| | Source (Sérsic) | |
| Effective radius ($R_{\rm Eff}$) | 0.2–0.6 ($\log_{10}$) | arcsec |
| Axial ratio | 0.3–1.0 | – |
| Major-axis angle | −90–90 | degree |
| Sérsic index | 0.5–5.0 | – |
| | Sérsic Blobs (1 up to 5) | |
| Effective radius | (1% to 10%)$R_{\rm Eff}$ | arcsec |
| Axial ratio | 1.0 | – |
| Major-axis angle | 0 | degrees |
| Sérsic index | 0.5–5.0 | – |

*Note.* [(a)]The Einstein radii for the *HST-like* data sets are sampled from a smaller distribution with maximum value 2 arcsec to account for the different resolution of the instrument and avoid gravitational arcs escaping from the cutouts.

### 2.1 Simulating gravitational arcs

Both simulation procedures start with a set of mock gravitational arcs. These are generated from scratch through the simulation code presented in Chatterjee (2019) and previously employed in several studies on strong lensing (Petrillo et al. 2017, 2019a, b; Gentile et al. 2022). The code first simulates a high-$z$ galaxy through a Sérsic profile (Sérsic 1963), whose parameters are randomly sampled from the distributions reported in Table 1. Uniform sampling is adopted for three parameters (axial ratio, position angle, and Sérsic index), while the effective radius is sampled from a logarithmic distribution. This choice is necessary to account for the likely smaller size of high-$z$ sources, as discussed in Petrillo et al. (2017). The code also adds a random number of Sérsic components (up to five) to the main light distribution to crudely mimic star-forming regions in the lensed galaxy (Petrillo et al. 2019a). The parameters of these components are sampled from the distributions reported in Table 1.

Once the background source is generated, the code simulates the foreground deflector's matter distribution through a singular isothermal ellipsoid model (SIE; Kormann, Schneider & Bartelmann 1994). All the model parameters are uniformly sampled from the distributions in Table 1. It is important to underline that, in order to save memory during the training phase, we sample the Einstein radii for the *HST mocks* from a smaller distribution than for the *Euclid mocks*. This step is crucial to prevent the most asymmetrical arcs from being cut out from the stamp due to the better resolution of the HST detector. To increase the complexity of the lensing galaxy and to account for the likely presence of a matter distribution along the line of sight, the code also adds a Gaussian Random Field to the gravitational field generated by the SIE model. More details on this step and the parameters employed in the simulation (i.e. the form of the power-spectrum and the variance of the field) can be found in Hezaveh et al. (2016) and Petrillo et al. (2019a). Accounting for the line-of-sight matter distribution can substantially contribute to the realism of the simulations and produce a significant change in the accuracy of the lens-modelling algorithms (see e.g. Pearson et al. 2021).

Once simulated both the background source and the foreground deflector, the code performs the ray-tracing and generates the gravitational arc. For doing so, the software randomly poses the lensed source within a distance given by the tangential caustics of the SIE plus one effective radius of the source Sérsic profile from the centre of the model. This step is needed to populate the training set with a higher percentage of highly-magnified arcs with respect to lensed systems with less-magnified two or four lensed images. Simulations are performed on a 121 × 121 pixel grid for the *Euclid mocks* (corresponding to a 12 arcsec side at the 0.1 arcsec pixel$^{-1}$ resolution expected from the *Euclid* VIS imager; Cropper et al. 2018). Similarly, the simulations of *HST mocks* are performed on a 131 × 131 grid (corresponding to a 3.9 arcsec side at the 0.03 arcsec pixel$^{-1}$ resolution of the ACS camera; Ryon 2022).

The final step of our simulation procedure consists of a convolution with a Gaussian 2D-kernel. This step produces more realistic images by accounting for the PSF blurring. The full width at half maximum (FWHM) values of the kernels are chosen as 0.16 and 0.08 arcsec to mimic the PSF expected from the *Euclid* and HST images, respectively (Cropper et al. 2018; Ryon 2022).

### 2.2 Simulating the lens light

A realistic image of a gravitational lens must include the light distribution of the foreground deflector. Moreover, since the possible blending between the arc and the central galaxy can make it harder to detect the lensing features, it can sensibly affect the modelling accuracy of our algorithm. In the current literature, two different strategies have been followed to simulate the deflectors. In some studies (e.g. Pourrahmani, Nayyeri & Cooray 2018; Metcalf et al. 2019), the deflector is generated from scratch through an analytical brightness profile. In others, mock gravitational arcs are stacked on real galaxy images employed as likely deflectors (e.g. Petrillo et al. 2017; Gentile et al. 2022). In this work, we employ both strategies.

#### 2.2.1 *HST mocks*

For the *HST mocks*, we employ as deflectors real galaxies observed in the *Cosmic Evolution Survey* (COSMOS) field by the HST (Koekemoer et al. 2007; Massey et al. 2010). First, we select the brightest galaxies in the COSMOS2020 catalogue[2] (Weaver et al.

[2]https://cosmos2020.calet.org/

2022), collecting all the sources with a F814W magnitude brighter than 21.5 (Gentile et al. 2022). Among these sources, we select the early-type galaxies, which are known to represent the majority of the gravitational lenses (e.g. Eisenstein et al. 2001; Oguri et al. 2006). We employ the 'FARMER' version of the COSMOS2020 catalogue to perform this second selection. Since this version contains the photometry extracted through the profile-fitting code 'THE FARMER' (Weaver et al, in preparation), it also includes a value named 'SOLUTION_MODEL' describing the best-fitting brightness profile for each source. We select all the galaxies with SOLUTION_MODEL equal to '*DevGalaxy*' (i.e. galaxies with a de Vaucouleurs brightness profile; de Vaucouleurs 1948) or '*SimpleGalaxy*' (partially resolved galaxies with a circular exponential profile). In doing so, we collect a sample of ∼4000 likely early-type galaxies and retrieve HST imaging for these sources from the public COSMOS archive.[3] A visual inspection of a significant fraction of these sources confirmed a contamination rate (i.e. a fraction of spirals and irregular galaxies) lower than 10 per cent. Since this tiny percentage of contaminants is not expected to sensibly affect the training, these images are not removed from the data set.

To finally simulate *HST mocks*, we follow a slightly modified version of the strategy employed in Petrillo et al. (2017, 2019a, b) and Gentile et al. (2022):

(i) We simulate a gravitational arc, as described in Section 2.1;
(ii) We randomly select a galaxy from the aforementioned sample of likely deflectors. To avoid excessive contamination from the central galaxy light, we choose only galaxies with an FWHM[4] in the range $[0.1, 1.1]\theta_E$. These values are broadly consistent with the findings by Bolton et al. (2008);
(iii) We randomly rotate the deflector image by an angle in the range [0,360] degrees and flip it on the horizontal axis with a probability of 50 per cent. Moreover, to account for the possible offset between the center of the SIE and that of the deflector's light, we shift the image by an amount of pixels randomly sampled in the range [−2,2] in both the $x$- and $y$-axis.
(iv) We stack the two images and rescale the brightness of the gravitational arc to $\alpha B$, where $B$ is the maximum brightness of the central galaxy and the $\alpha$-factor is uniformly sampled in the range [0.03,0.2]. The $\alpha$-factor accounts for the typical brightness ratio between lenses and arcs (e.g. Petrillo et al. 2017).
(v) We perform a square-root stretching of the co-added image to enhance the low-brightness lensing features;

We underline that – by employing real galaxies as deflectors – we do not need to simulate the background noise or add additional sources in the lens environment to account for the likely presence of nearby galaxies.

#### 2.2.2 *Euclid mocks*

The procedure exposed in the previous paragraph cannot be employed to simulate *Euclid mocks*. Since – at the moment – we do not have data on real galaxies observed by the *Euclid* satellite, we simulate from scratch the light distribution of the deflector by employing a Sérsic profile (Sérsic 1963). The forthcoming papers of the LEMON series will use a more accurate method by employing more realistic deflectors. The brightness profile's axial ratio and position angle are uniformly sampled from the same distributions employed in Section 2.1 and summarized in Table 1. The Sérsic index is fixed to the value $n = 4$ to account for the typical elliptical galaxies in the gravitational lenses population (e.g. Eisenstein et al. 2001; Oguri et al. 2006). The effective radius is uniformly sampled from the distribution $[0.1, 1.1]\theta_E$ to obtain images similar to those produced for the *HST mocks*. As for the *HST mocks*, we allow a small offset in the range [−2,2] pixels between the deflector center and the SIE.

Differently from the *HST mocks*, since the deflectors are generated from scratch, we also need to simulate the background noise. This is done through a Gaussian random-number generator: the mean value and the standard deviation of this normal distribution are computed starting from the average sky brightness (22.2 mag) and exposure time (1.610 s) expected from the *Euclid wide survey* (Laureijs et al. 2011; Scaramella et al. 2021). Once a noise map is simulated, we rescale the brightness of the arc to obtain an integrated SNR in the range [5,20] and rescale the brightness of the central galaxy to get an $\alpha$-factor in the same range employed in the previous paragraph.

## 3 BNNS

In a computational perspective, the problem of lens-modelling can be reduced to assigning a set of continuous values (i.e. the set of parameters describing the mass distribution in the lensing galaxy) to an image. From a machine-learning point of view, this task is a regression problem and – therefore – it can be successfully addressed with supervised-learning algorithms such as CNNs (e.g. LeCun et al. 2015). These algorithms can extract the most relevant features from an image and – once provided a proper training set – can approximate the complex relationship between the input images and the target values. For this reason, CNNs are nowadays employed in many astrophysical studies spanning from the morphological classification of galaxies (e.g. Dieleman, Willett & Dambre 2015; Aniyan & Thorat 2017; Domínguez Sánchez et al. 2018) to cosmological studies (e.g. Fluri et al. 2019; Canameras et al. 2020). Never the less, in the last years, some studies started to highlight several limits in the use of CNNs (see a review of the main ones in Kendall & Gal 2017 and Charnock et al. 2020). Among these, the difficulty to associate a confidence interval to the predicted parameters has probably the most significant impact on the scientific potentialities of these algorithms. Estimating the uncertainties affecting the scientific quantities is generally required to compare them with the theoretical predictions. Moreover, a confidence interval is generally useful in order to exclude statistical outliers (in our case, the systems for which the lens modelling is not reliable). A possible way to overcome this problem and obtain an estimation of the uncertainties is represented by BNNs (e.g. Charnock et al. 2020). This new generation of machine-learning algorithms is built by starting from a classical neural network (or a CNN when the analysis of images is needed) but it can also provide a fully Bayesian treatment of the different sources of uncertainty affecting the input data and the algorithm itself. Even though the concept itself of uncertainty is quite debated, especially in the field of machine-learning, there is an almost unanimous consensus about what a BNN is required to account for in its predictions. It mainly consists in what the algorithm 'does not know' (the so-called *epistemic uncertainty*) and what it 'cannot know' (i.e. the *aleatoric uncertainty*).

### 3.1 Epistemic uncertainty

The first kind of uncertainty that we have to model concerns all the features that a BNN cannot recognize because of a lack of proper training. Since – as discussed in Section 2 – we do not expect our

[3] https://irsa.ipac.caltech.edu/data/COSMOS/
[4] The value of the FWHM is the FWHM_F814W entry in the COSMOS2020 catalogue, obtained from the HST imaging.

algorithm to be sensitive on ranges of parameters insufficiently well-covered by our training set, we expect the predictions on images 'too different' from those on which we trained the algorithm to be 'more uncertain'. To translate this qualitative concept into a confidence interval, we have to pose the problem of uncertainty into a Bayesian framework.

It can be done by representing a 'standard' (convolutional) neural network as a highly complex parametric function

$$\mathbf{y} = f(\mathbf{x}, \omega), \tag{1}$$

providing a set of output values ($\mathbf{y}$) for each input datum ($\mathbf{x}$), once provided the weights ($\boldsymbol{\omega}$). Moreover, since the weights are adjusted during the training, they can be expressed as an (unknown) function of the data in the training set ($\mathbf{X}$) and their target values ($\mathbf{Y}$):

$$\omega = \omega(\mathbf{X}, \mathbf{Y}). \tag{2}$$

In a Bayesian probabilistic framework, the uncertainty on the output values can be evaluated through their posterior probability distribution:

$$p(\mathbf{y}) = p(\mathbf{y}|\mathbf{x}, \omega). \tag{3}$$

In principle, this term could be evaluated by marginalizing over all the possible values of the weights.

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int_{\Omega} p(\mathbf{y}|\mathbf{x}, \omega) p(\omega|\mathbf{X}, \mathbf{Y})\, d\omega. \tag{4}$$

However, due to the high dimensionality of the weights space $\Omega$ and the lack of knowledge about the posterior probability of the weights $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$, equation (4) cannot be evaluated in a computationally affordable way. A possible solution to this problem can reside in variational inference (e.g. Jordan et al. 1999). This technique consists of approximating the unknown posterior with an analytic and parametric function $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ with $\boldsymbol{\theta}$ variational parameters by minimizing the difference between these distributions through the Kullback–Leibler (KL) divergence (Kullback & Leibler 1951). In this way, equation (4) can be rewritten as

$$p(\mathbf{y}|\mathbf{x}) \approx \int p(\mathbf{y}|\mathbf{x}, \omega) q_{\theta}(\omega)\, d\omega, \tag{5}$$

allowing a more straightforward evaluation of the probability distribution on the output value. It can be shown (see e.g. Gal & Ghahramani 2015a, b) that minimizing the KL divergence is equivalent to maximizing the log-evidence lower bound with respect to the variational parameters $\boldsymbol{\theta}$:

$$\mathcal{L}_{VI} = \int q_{\theta}(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega)\, d\omega - KL(q_{\theta}(\omega)||p(\omega)). \tag{6}$$

A possible choice for the variational function $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ is the one employed in the so-called '*Monte Carlo dropout*' technique (Gal & Ghahramani 2015a):

$$\begin{cases} \omega_i = \theta_i \cdot \mathrm{diag}(\{\mathrm{s_{ij}}\}_{\mathrm{j=1}}^{\mathrm{i-1}}) \\ s_{ij} = \mathrm{Bernoulli(p_i)} \end{cases}, \tag{7}$$

where $s_{ij}$ are Bernoulli-distributed random variables with probability $p_i$. With this choice, the first term in equation (6) becomes the log-likelihood of the network's predictions, while the second term can be approximated with an $L_2$ regularization with a $\lambda$ parameter (Gal & Ghahramani 2015b):

$$\mathcal{L}_{VI} \sim \sum_{i=1}^{N} \mathcal{L}(\mathbf{y}_i, \mathbf{y}_i^{\star}(\mathbf{x}_i, \omega)) - \lambda \sum_{j} ||\omega_j||^2, \tag{8}$$

where $\mathcal{L}$ is the log-likelihood for the network's predictions $\mathbf{y}_i^{\star}$ on the input datum $\mathbf{x}_i$ with real values $\mathbf{y}_i$ and weights $\boldsymbol{\omega}$ randomly sampled from the variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$. Moreover, the choice in equation (7) allows us to evaluate the integral in equation (5) with a simple Monte Carlo integration. Specifically, it can be easily implemented with the '*dropout*' technique (Srivastava et al. 2014) consisting of randomly switching off some connections between the neurons in the neural network with a '$p$' probability ($1-p$ is the so-called *keep-rate*). The dropout is implemented both at training time and at testing time. In doing so, each prediction made on the same input datum with some connections randomly dropped out is equivalent to a sampling from the posterior probability presented in equation (3). With a sufficiently high number of predictions, we can reconstruct the probability distribution and, therefore, evaluate its 'epistemic' uncertainty by measuring its standard deviation.

### 3.2 Aleatoric uncertainty

A second kind of uncertainty that is crucial to evaluate properly is the so-called 'aleatoric uncertainty'. This value is mainly related to the intrinsic quality of the data analysed by the algorithm. Corruptions in the images (e.g. the presence of a masked region), PSF blurring, source blending and a low value of the SNR are among the most common sources of aleatoric uncertainty. It is noteworthy to underline that – differently from the epistemic uncertainty that, in principle, could be decreased by employing a more complete training set – the aleatoric uncertainty only depends on the quality of the analysed data (see e.g. Kendall & Gal 2017; Charnock et al. 2020). Since it is a function of the only input data, aleatoric uncertainty must be evaluated for each input image separately. This task is however challenging, since these values are not available for the images in the training set. Therefore, aleatoric uncertainties must be evaluated in an unsupervised framework. A common choice (but not the only one; see e.g. Fagin et al. in preparation) to estimate this quantity consists in employing a *Gaussian negative-log-likelihood* as the loss function to be minimized during the training (equation 8):

$$\log \mathcal{L}(\mathbf{y}_i, \mathbf{y}_i^{\star}(\mathbf{x}_i, \omega)) = \sum_{j=1}^{N_P} \left[ -\frac{1}{2\sigma_{i,j}^2} ||y_{i,j} - y_{i,j}^{\star}|| - \frac{1}{2} \log(\sigma_{i,j}^2) \right] \tag{9}$$

where $\sigma_{i,j}$ is the aleatoric uncertainty predicted for the $j$th parameter of the $i$th image. We underline that the second term in equation (9) prevents the algorithm to predict an infinite uncertainty for all the images, regardless of the input.

### 3.3 LEMON architecture

In this paper, we present LEMON (LEns MOdelling with Neural networks): a lens-modelling algorithm based on the BNN described in the previous subsection. Our network is able to model both the aleatoric and epistemic uncertainties affecting the data and the training process and to combine them into a single confidence interval for each predicted parameter.

We implement our BNN starting from a standard CNN with a *ResNet-34* architecture (He et al. 2015). The whole code is written in PYTHON 3.9, employing the open-source library KERAS (Chollet et al. 2015) with a TENSORFLOW back-end (Abadi et al. 2015). As described in the previous sections, we modify the architecture of the CNN by adding a dropout layer after each weight layer both in the convolutional blocks and in the final *fully connected* layer. While the first layer is designed to take as input a single-band image of a gravitational lens, the last layer of the network (i.e. the one that

performs the regression) is composed of six nodes. Three of them will predict the three parameters of the SIE model (Section 2) while the others will predict the relative logarithmic aleatoric uncertainties (Section 3.2). We decide to train our network to predict the Einstein radius ($\theta_E$) and the two components of the complex ellipticity ($\epsilon_x$ and $\epsilon_y$; see e.g. Kormann et al. 1994) instead of the more common axial ratio and position angle:

$$\epsilon_x = \frac{1-q^2}{1+q^2}\cos(2\varphi) \quad \epsilon_y = \frac{1-q^2}{1+q^2}\sin(2\varphi). \tag{10}$$

As discussed in Perreault Levasseur et al. (2017) and Hezaveh et al. (2017), this choice increases the algorithm's accuracy. Moreover, as shown in Perreault Levasseur et al. (2017) and Kendall & Gal (2017), we train the algorithm to predict $\log(\sigma^2)$ instead of directly $\sigma$: this choice improves the numerical stability of the algorithm and prevents it from predicting negative variances. A schematic representation of the LEMON architecture is reported in Fig. A1 in the Appendix.

Finally, a reliable confidence interval should consider both the epistemic and aleatoric uncertainties. In this work, we combine the two sources of uncertainty through the procedure previously employed in Perreault Levasseur et al. (2017):

(i) For each input image, we predict the mean value ($\mu_i$) and the relative aleatoric uncertainty ($\sigma_i^A$) for the generic parameter $p$ ($\theta_e$, $\epsilon_x$, or $\epsilon_y$);
(ii) For each parameter $p$, we randomly sample a new value $p_i$ from a Gaussian distribution centred on the predicted value $\mu$ and with a standard deviation equal to $\sigma_i^A$. We assume that the Gaussian form better describes the statistical distribution of the aleatoric uncertainties since these were obtained by minimizing a *Gaussian negative log-likelihood* (Section 3.2);
(iii) We repeat the first two steps a thousand times for each image. In doing so, we sample the posterior probability of the parameters for each image;
(iv) Finally, we compute the mean ($\bar{p}$) and standard deviation ($\delta p$) of the distribution given by the $p_i$, obtaining the final value for the considered parameter and its combined uncertainties.

In the following – for the sake of brevity – we will employ the symbols $p$ and $\delta p$ to refer to the mean value of the $p_i$ distribution and the combined uncertainties for each parameter.

### 3.4 Training the algorithm

We perform two different trainings of the LEMON algorithm: one for each training set discussed in Section 2. We train the algorithm on 75 per cent of each training set, saving the lasting 25 per cent for cross-validation. We perform the training with the *stochastic gradient descent* technique, with a batch size of 32 images randomly chosen from the whole data set. During the training, the algorithm minimizes the *Gaussian negative-log-likelihood* introduced in equation (9) through the ADAM optimizer (Kingma & Ba 2014). We start the training with an initial learning rate of $10^{-3}$. We gradually update the learning rate during the training employing the REDUCEONPLATEAU callback provided by KERAS with a patience parameter concerning the validation loss of five epochs up to $10^{-5}$. An $L_2$ regularization is also employed to prevent overfitting and to account for the second term present in equation (8). Finally, during this phase, we employ data augmentation: a common choice in regression and classification problems involving images. It consists of feeding several times the same image to the network, realizing a different transformation every time. In this work, each image is translated in the up–down and left–right directions by an integer number of pixels in the range [−4,4]. This step reduces the risk of overfitting since it artificially increases the size of the training set and allows the BNN to learn the translational invariance of the lensing configurations. It is worth underlining that, differently from previous applications (e.g. Petrillo et al. 2017, 2019a, b; Gentile et al. 2022), we do not employ any augmentation involving rotations, flipping, and scaling of the analysed images. This choice is necessary since the predicted parameters are not invariant under these transformations. As discussed in the previous section, dropout layers are employed at both training and testing time, with a *keep-rate* of 0.97.

The whole trainings require on average 75 epochs and take ∼2.5 h each when performed on a single Tesla-K200 GPU freely offered by the cloud-computing platform GOOGLE COLAB.[5]

## 4 RESULTS

In this section, we assess the performances of the LEMON algorithm by applying the BNN to several data sets. After a brief introduction on the metrics employed to evaluate the accuracy of the algorithm and the reliability of the estimated uncertainties, we apply the network to two data sets of simulated images. The results obtained on the *Euclid mocks*, in particular, will be useful to forecast the performances of the LEMON algorithm when applied to the real data from the forthcoming *Euclid* satellite. These forecasts – however – strongly rely on the hypothesis that the simulations reasonably resemble the characteristics of real lenses. To test this assumption, we also apply the BNN to a data set of real lenses observed by the HST and compare the results obtained on these systems with those attained on the *HST-like* simulations. Finally, we perform additional tests about the relationship between the accuracy of the algorithm and the employed training sets.

### 4.1 Useful metrics

Throughout the following sections, we will define an 'ideal' lens-modeller as an algorithm able to supply the exact set of SIE parameters for each analysed lens. To compare our results with those expected from such an algorithm, we must introduce some statistical indices (or 'metrics' in the following). The choice of these quantities is not unique or trivial and can sensibly affect our description of the results or our ability to compare them with earlier studies present in the current literature. Since our algorithm can supply both a point estimate of the predicted parameters and their confidence intervals, we need more metrics to evaluate these aspects.

#### *4.1.1 Overall accuracy*

The first property to evaluate is the distance between the mean values predicted by the algorithm and the real ones. In a graphical perspective, the accuracy of a lens-modeller can be assessed through the scatter plots shown in Fig. 2. The accuracy can also be assessed by measuring the difference $\Delta p$ between each predicted parameter and its real value for each analysed image. Hence, we define the 'bias' ($\mu$) as the mean value of the $\Delta p$ distribution computed on the whole data set and the 'standard deviation' ($\sigma$) as the minimum $|\Delta p|$ having a 68 per cent statistical coverage of the analysed data set. In addition, we can consider non-symmetrical distributions by computing $\sigma^+$ and $\sigma^-$ as the 16th and 84th percentile of the $\Delta p$ distribution.

[5] https://colab.research.google.com/

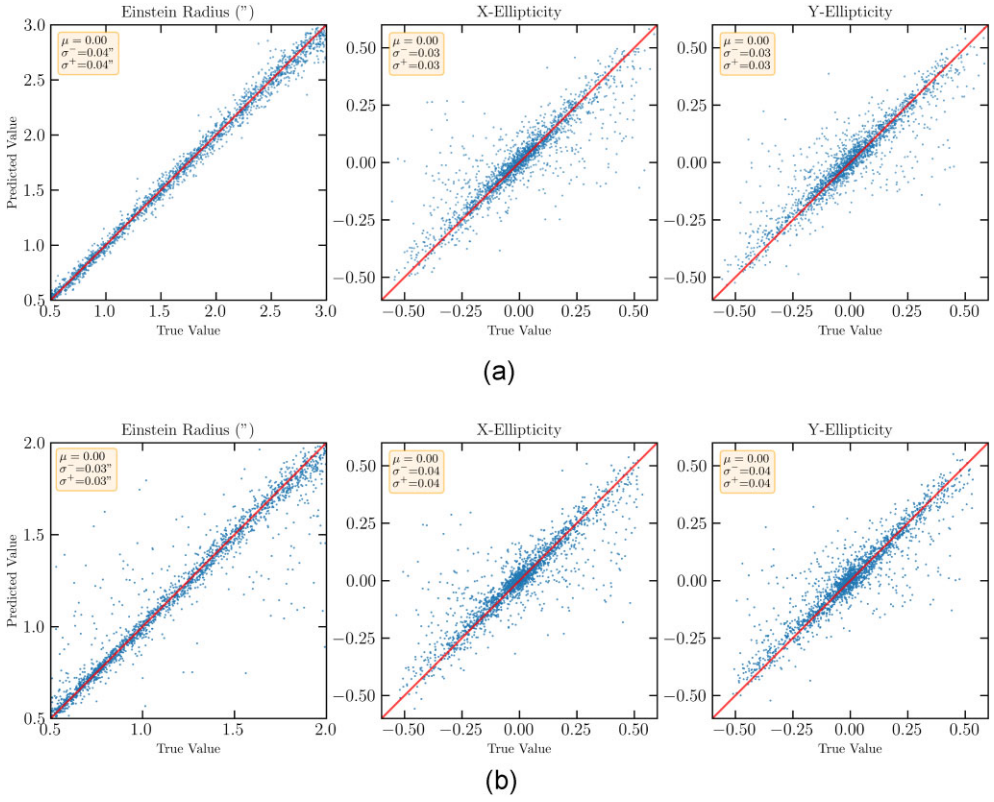


**Figure 2.** Comparison between the predictions of the LEMON algorithm and the real values of the SIE parameters for the simulated *Euclid-like* (top row) and *HST-like* (bottom row) data sets. For each plot, we also report in the yellow box the *bias* ($\mu$) and the 16th and 84th percentiles of the error distribution ($\sigma^+$ and $\sigma^-$, respectively). Further details in Section 4.3.

#### *4.1.2 Confidence intervals*

Evaluating the reliability of the estimated confidence intervals is a more challenging task. Since these values are predicted in an unsupervised framework (see the discussion in Section 3), we do not have any reference value to compare our results with. In the hypothesis that the uncertainties affecting our algorithm and data can be described with a Gaussian distribution (a reasonable assumption, given the likelihood introduced in Section 3.2), we expect that an ideal modeller would provide confidence intervals producing a Gaussian statistical coverage. Therefore, we expect that $\sim 68$ per cent of the predicted values lie within one estimated $\delta p$ from their actual value, $\sim 95$ per cent within $2\delta p$ and $\sim$99 per cent within $3\delta p$. In most of the earlier studies concerning BNN-based lens-modellers, the reliability of the confidence intervals has been assessed by comparing the statistical coverage at 1, 2, and $3\sigma$ computed on the data set with that expected from a Gaussian distribution. In this work, we employ a more sophisticated method relying on the so-called '*reliability plots*' (Fig. 3). These graphs are built by computing the cumulative statistical coverage $P(x)$ defined as

$$P(x) = \frac{N(|p_i - p_i^\star| < x\delta p_i)}{N_{\rm Tot}}, \tag{11}$$

where the numerator is the number of systems for which the $\Delta p$ is smaller than $x$ times the related uncertainty $\delta p$. $P(x)$ is evaluated for an ideal Gaussian distribution (on the $x$-axis) and for the analysed data set (on the $y$-axis) for $x$ in the range [0,5]. With this definition, an ideally calibrated algorithm would provide a graph with all the data points lying on the bisector. Similarly, a graph lying in the upper (lower) semiplane would represent a systematic overestimation (underestimation) of the uncertainties.

### 4.2 Calibrating the uncertainties

The choice to employ the reliability plots also allows us to perform a more efficient uncertainties calibration with respect to previous studies. BNNs, indeed, are generally not able to provide well-calibrated uncertainties, being affected by a systematic over or underestimation of the confidence intervals (see Guo et al. 2017 and references therein for some possible explanations of this phenomenon). Therefore, a successive 'calibration step' is often required. This can be performed following different methods (e.g. Zadrozny & Elkan 2001, 2002; Gal, Hron & Kendall 2017; Perreault Levasseur et al. 2017).

In this work, we employ a slightly modified version of the so-called *platt-scaling* method (Kull, Filho & Flach 2017) employed in Hortúa et al. (2020). This method consists in re-scaling the predicted uncertainties by a factor '$s$' ($\sigma \rightarrow s\sigma$), in order to minimize the difference between the uncalibrated reliability plot and the ideal one. The procedure is the following:

(i) We divide the *validation set* in two parts: 30 per cent for 'calibration' and 70 per cent for 'test';

(ii) We build the *reliability plot* for the calibration set, employing the original uncertainties estimated by the BNN;

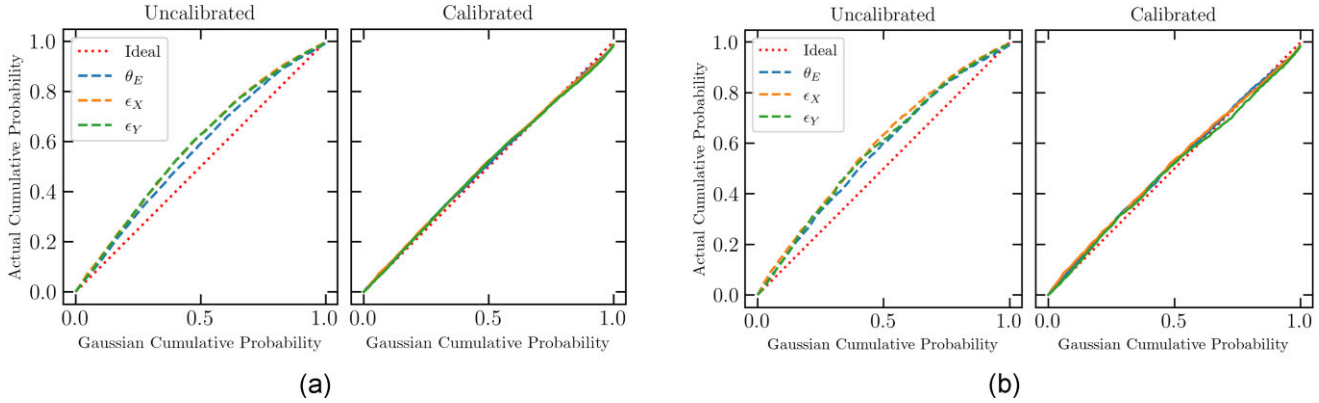


**Figure 3.** Reliability plots generated for the *HST-like* (a) and for the *Euclid-like* (b) validation sets before and after the calibration procedure described in Section 4.2. It is evident how the calibration improves the reliability of the estimated uncertainties and how these are able to reproduce a Gaussian statistical coverage of the analysed data sets. Further details in Section 4.

(iii) We fit the uncalibrated *reliability plot* with a $\beta$-function defined as

$$\beta(x, a, b, c) = \frac{1}{1 + \left(e^c \frac{x^a}{(1-x)^b}\right)^{-1}}, \tag{12}$$

with $a, b, c \in \mathbb{R}$ free parameters fixed during the fitting;

(iv) We find the value $s$ by minimizing[6] the difference between the $\beta$-function and the bisector of the *reliability plot*;

(v) We rescale the uncertainties of the test set predicted by the BNN by the found $s$-factor and build a new 'calibrated' reliability plot

Fig. 3 shows the difference between the uncalibrated reliability plot and the calibrated one for the parameters estimated by the LEMON algorithm. It is clear the improvement in the statistical coverage achieved by this method.

### 4.3 Application to simulated data

The first assessment of the LEMON algorithm's abilities to model vast samples of lenses is performed on two simulated validation sets. We assemble them following the same strategy discussed in Section 2, employing the same division between *HST* and *Euclid mocks*. To avoid possible biases and perform totally blind cross-validation, we reseed all the random generators used in the procedure: both those employed in the parameters' sampling and in the background noise generators. Since we are interested in the application to vast data sets, we simulate 10 000 lenses for each data set. The predictions are performed on the same hardware introduced in Section 3.4 and take ∼1.5 h for each data set, with an average modelling time of less than 0.5 s for each image. The scatter and reliability plots obtained on these data sets are reported in Figs 2 and 3.

### 4.4 Further analysis

The results achieved on these simulated data sets make it interesting to investigate the impact of the simulations employed to train the algorithm on its accuracy. We further investigate this point by simulating four additional data sets (two with *Euclid-like* properties and two with *HST-like* ones). Two of these data sets are generated following the same procedure described in Section 2, but without adding the light distributions of the deflectors. The importance of these '*arcs-only data sets*' is two-fold. On the one hand, these simulated lenses can help us to quantify the loss of accuracy due to the presence of the lens light and the possible improvement in the performances of our algorithm if we employed lens-subtracted systems. On the other hand, the results achieved on these systems can be compared with those obtained by previous studies based on analogous techniques but using lens-subtracted lenses (e.g. Hezaveh et al. 2017; Pearson et al. 2019) limiting the possible biases due to the different analysed objects.

The other two additional data sets explore the possible impact of the correlation between the light and mass distribution of the deflector. As highlighted by several studies, we do not expect the parameters of the brightness profile of the lens and its matter distribution to be independent. In particular, Koopmans et al. (2006) found an interestingly tight correlation between the parameters of the Sérsic model of the lens and its SIE model:[7]

$$\begin{cases} q_{SIE} = (0.99 \pm 0.11) q_L \\ \Delta\theta = \theta_{SIE} - \theta_L = (0 \pm 3)^\circ \end{cases} . \tag{13}$$

Inserting these correlations in the simulated data sets might – in principle – improve the accuracy of our algorithm by augmenting the number of features available to predict the SIE parameters. To further investigate this possibility, we simulate these additional '*correlated-lenses data sets*' through a slightly modified version of the procedure described in Section 2.2:

(i) *Euclid-like* lenses: the parameters of the Sérsic profile are not randomly sampled from the distribution in Table 1. Once the gravitational arc is simulated, the deflector's axial ratio and position angle are sampled from a Gaussian distribution centred on the same parameters of the SIE model and with the standard deviations reported in equation (13).

(ii) *HST-like* lenses: the deflector is not randomly chosen from the sample of likely deflectors described in Section 2.2. Once the

[6]The minimum is found numerically through the PYTHON SCIPY library (Virtanen et al. 2020).

[7]Actually, this correlation has been verified for galaxies with $q > 0.5$, for more elliptical galaxies some studies found different results (e.g. Treu et al. 2011). Since in this study we are discussing the role of a correlation in itself, we employ the result by Koopmans et al. (2006) for all the lenses in the data set.

**Table 2.** Mean accuracies achieved by the LEMON algorithm when applied to the several data sets introduced in Section 4.4. All the values reported in the table are the standard deviation of the error distribution computed as the 68% statistical coverage of the analysed data sets. Further details are given in Section 4.4.

| | *Standard data sets* | | *Arcs-only* | | *'Correlated' lenses* | |
|---|---|---|---|---|---|---|
| | *Euclid* mocks | HST mocks | *Euclid* mocks | HST mocks | *Euclid* mocks | HST mocks |
| $\theta_E('')$ | 0.04 | 0.03 | 0.02 | 0.02 | 0.04 | 0.03 |
| $\epsilon_X$ | 0.03 | 0.04 | 0.02 | 0.02 | 0.03 | 0.04 |
| $\epsilon_Y$ | 0.03 | 0.04 | 0.02 | 0.02 | 0.03 | 0.04 |

gravitational arc is simulated, the $q_L$ and $\varphi_L$ values for the axial ratio and position angle of the deflector, respectively, are sampled from the same Gaussian distribution mentioned above. Therefore, the galaxy with the closest $q$ value in the deflector sample is employed as lens. Analogously, the deflector is rotated to match its original position angle with the new one.

Once these new data sets are simulated (composed of 40 000 images each, as described in Section 2), we perform four additional trainings of the LEMON algorithm (one for each data set). Therefore, we apply each network to a corresponding validation set of 10 000 images, simulated following the same prescriptions described before and with a different random seed. The results of these additional cross-validations are reported in Table 2.

### 4.5 Application to real data

To independently assess the accuracy of the LEMON algorithm in modelling lenses, we apply the BNN trained on the *HST mocks* to a sample of real lenses discovered in the SLACS programme (Bolton et al. 2006, 2008; Shu et al. 2017). Since we do not expect our algorithm to be accurate out of the range of parameters covered by the training set, we only select the systems encompassing these characteristics:

(i) An elliptical central galaxy, as those employed in Section 2.2;
(ii) The SIE parameters estimated from lens-modelling (as reported in Shu et al. 2017; Bolton et al. 2008) within the range summarized in Table 1;
(iii) An SNR integrated on the full arc larger than 5;
(iv) A brightness ratio between the arc and the central galaxy (i.e. the $\alpha$-factor defined in Section 2.2) in the range [0.03,0.2].

We retrieve HST imaging in the F814W band for a pilot sample of 42 SLACS lenses having all the above characteristics[8] (our '*real-lenses sample*' in the following; see some examples in Fig. 4). We extract stamps of these systems with a 131-pixel side, apply a square-root stretch and pass these images to the algorithm. The results obtained by this algorithm application and their comparison with the estimates of the SLACS collaboration are reported in Figs 6 and 7. We underline that the calibration of the uncertainties is performed in a totally blind way, employing the same 's' factor obtained during the validation step on the simulated *HST mocks* (see Section 4.2).

[8]The HST images are retrieved from the Hubble Legacy Archive: https://hla.stsci.edu/.

## 5 DISCUSSION

This section discusses the main scientific findings obtained through the results described in Section 4.

(i) Modelling time: the first noteworthy result achieved by the LEMON algorithm resides in the time employed to model a vast sample of 10 000 simulated lenses. Pearson et al. (2021) reported – for a blind application of the PYAUTOLENS lens-modelling algorithm (Nightingale et al. 2018) – a mean modelling time of 30 min, with several outliers requiring up to 2 h to be modelled correctly. Our algorithm achieves an average modelling time of 0.5 s, independently of the morphology of the analysed system. The improvement of several orders of magnitude in the computational cost of lens modelling between traditional and machine-learning techniques is evident (also thanks to the possibility to train and test the latter algorithms on GPUs, much faster than CPUs). In addition, we underline that the whole procedure employed in this paper to train the algorithm and model the lensed systems is entirely automated, not requiring any human intervention. These results make BNN-based algorithms particularly suitable to model vast data sets of lenses in a computationally efficient way.

(ii) Overall accuracy: we applied the LEMON algorithm to two different simulated sets of 10 000 lenses each, retrieving the accuracy summarized in Section 4.3. These results are perfectly comparable with other studies present in the current literature and based on similar techniques (CNN or BNN-based lens-modeller, see e.g. Hezaveh et al. 2017; Perreault Levasseur et al. 2017; Pearson et al. 2019; Schuldt et al. 2021). It is noteworthy to underline how the scatter achieved on the *HST-mocks* is slightly larger than that obtained on the *Euclid-mocks*, albeit the higher resolution of the first simulations. This result is probably due to the more realistic deflector employed in the simulation procedure of the *HST-mocks*. Furthermore, as expected, machine learning algorithms are still not able to outperform or even achieve the same accuracy as classical Bayesian techniques (see Pearson et al. 2021 for a quantitative comparison between machine-learning and classical methods). Never the less, CNN- or BNN-based algorithms can still be employed to provide a first fast modelling of vast samples of lenses to identify the most interesting systems for future follow-up or more accurate modelling with standard techniques. Moreover, as highlighted by Pearson et al. (2021), machine-learning predictions can be employed as priors for classical Bayesian methods, significantly improving the modelling time with respect to a completely 'blind' analysis based on uninformative flat priors.

(iii) Calibrated uncertainties: as highlighted in Section 4.2, the calibration procedure followed to obtain reliable uncertainties is quite effective, producing a Gaussian statistical coverage of the analysed samples (Fig. 3). The goodness of the procedure is also independently assessed with the blind cross-validation performed in Sections 4.2 and 4.5. The possibility to predict consistent uncertainties is crucial to allow most of the scientific applications of lensing, where it is fundamental to compare the observational evidence with the theoretical predictions. In addition, high values of the uncertainties can also allow a selection of the contaminants (i.e. the systems for which the modelling procedure failed). Finally, once verified the Gaussian statistical coverage, the estimated uncertainties can be employed in classical Bayesian techniques as Gaussian priors, additionally improving the computation time with respect to flat priors.

(iv) Different data sets: the results achieved on the additional data sets employed in Section 4.4 translate into several findings concerning the relationship between the employed training sets and

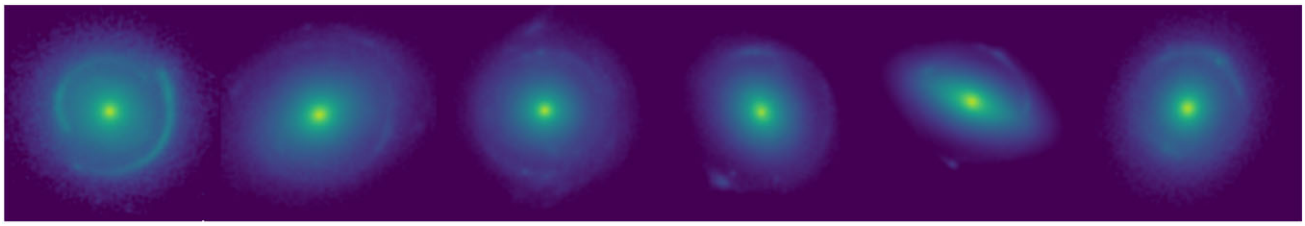

**Figure 4.** Some examples of the real lenses employed to test the accuracy of the LEMON algorithm. All the systems were observed by the HST during the *Sloan Lens ACS Survey* (SLACS; Bolton et al. 2006; Shu et al. 2017). Further details in Section 4.5.

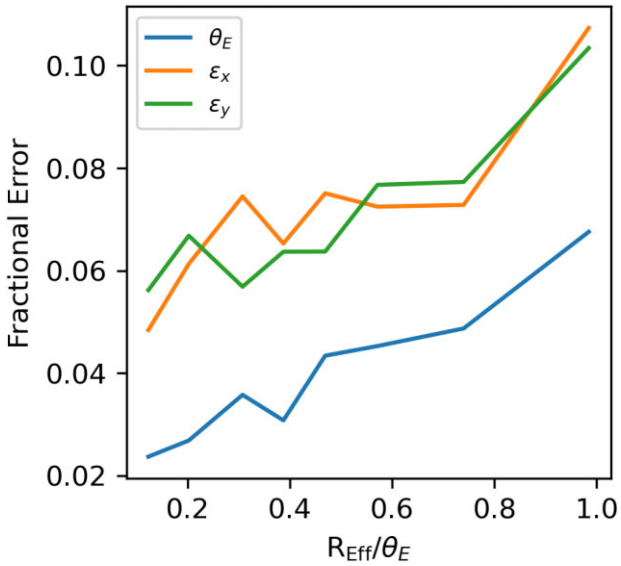


**Figure 5.** Median fractional accuracy achieved by the LEMON algorithm on the modelling of simulated lenses as a function of the ratio between the effective radius of the deflector and the Einstein radius of the lensed arc. As expected, the accuracy of the algorithm decreases with more significant blending between the central galaxy and the arc.

the algorithm's accuracy. First, as expected, the subtraction of the lens light increases the accuracy of the modelling (up to 100 per cent; see Table 2). This result is not surprising, since the presence of the central galaxy makes the identification of the lensing features more challenging for the algorithm. A confirmation of this hypothesis can be found in the analysis of the modelling accuracy as a function of the ratio between the $R_{\rm Eff}$ of the deflector and the Einstein radius of the lensed arc (Fig. 5). As expected, the median fractional error on the three predicted parameters increases when the blending between the two components is more significant. We underline, however, that the subtraction of the lens light is problematic for two main reasons. First, this process generally relies on parametric modelling of the deflector: wrong modelling can produce a severe bias in the deblending procedure and the inferred shape of the gravitational arc. Secondly, this procedure is not completely automatic but often requires a non-trivial human intervention to select the lensing features. The latter reason, in particular, strongly affects the possibility of modelling vast samples of lenses automatically. It is also worth noting that the different image quality achieved by the two instruments does not affect the accuracy of the algorithm when no lens light is included in the images.

A second – more counter-intuitive – finding consists in the identical accuracy obtained by the LEMON algorithm when applied to the correlated and uncorrelated lens data sets. A possible explanation of this result could reside in the ability of our algorithm to well discriminate between the gravitational arc and the lensing galaxy, basing its prediction only on the feature belonging to the first one and ignoring the shape and orientation of the latter. To study in detail this result, we perform a cross-study. Training the algorithm on the correlated data set and testing it on the uncorrelated one (and vice-versa) we obtain no significant changes in the accuracy of the predicted parameters. This result strengthens the hypothesis that the modelling performed by the LEMON algorithm is mainly driven by the gravitational arc.

(v) Real lenses: when applied to the set of real lenses from the SLACS sample (Section 4.5), the algorithm achieves the results summarized in Figs 6 and 7. We underline that – even though the bias is always consistent with zero – the algorithm produces a higher scatter between predicted and real values when passing from simulated to real data. This phenomenon is not new, as discussed in several studies concerning lens-finding with machine-learning algorithms (e.g. Petrillo et al. 2017, 2019a, b; Gentile et al. 2022). Our interpretation of this result resides in the still-not-sufficient realism of our simulations. To confirm our hypothesis, we perform an additional test on simulated data. We simulate an extra data set following the procedure discussed in Section 2, but without including the matter distribution on the line-of-sight. In doing so, we obtained a 'less-realistic' data set of simulated lenses. By training the LEMON algorithm on this new data set and by applying it to the original data set with 'more realistic lenses', we obtain – as expected – a small decrease in the accuracy of the algorithm (∼5 per cent on all the predicted parameters). This result agrees with that reported by Pearson et al. 2021 in an analogous test. Moreover, we obtain a correspondent increase in the predicted uncertainties (that, in principle, should be due to a higher epistemic uncertainty, see Section 3.1). Even though the calibration step described in Section 4.2 reproduces again a good Gaussian statistical coverage of the sample, this test clearly suggests that the realism of the simulations employed in during the training is crucial to achieve better accuracies with machine-learning based lens-modeller like LEMON. In the near future, it is possible to think of a training set composed only of real lenses observed in a wide survey such as those performed with *Euclid* or the *Vera Rubin Observatory*. At the moment, however, the only alternative consists in generating more reliable simulations. A possible improvement could be the employment of real galaxies as lensed sources (e.g. Pearson et al. 2021) or the increasing of the complexity of the deflector by employing more complex lens models (e.g. adding an external shear component, e.g. Keeton, Kochanek & Seljak 1997), or a better accounting for the matter distribution on the line-of-sight of the lens (e.g. Fleury, Larena & Uzan 2021) and likely perturbation to the lensing potential (e.g. Galan et al. 2022; Vernardos & Koopmans 2022). These improvements will be addressed in detail in the forthcoming papers of the LEMON series.

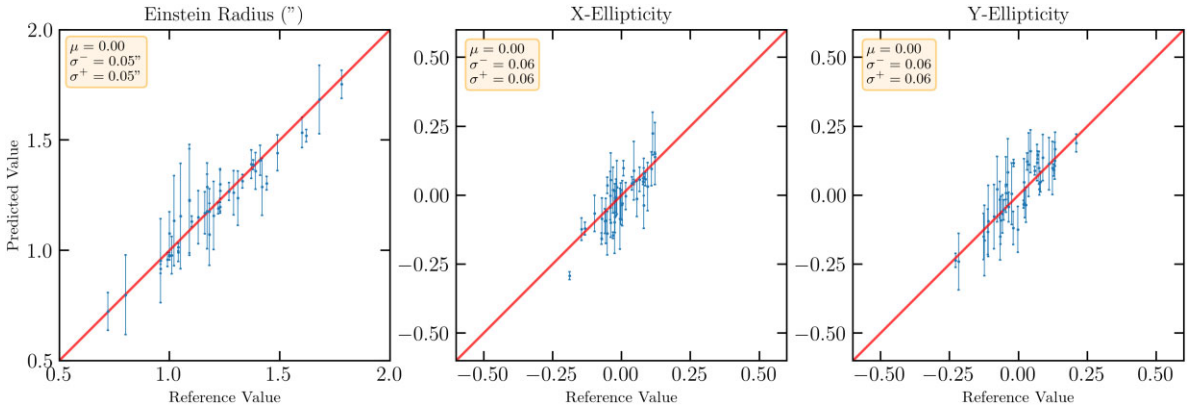


**Figure 6.** Comparison between the SIE parameters predicted by the LEMON algorithm and the values obtained by the SLACS collaboration through classical modelling (Bolton et al. 2008; Shu et al. 2017). The uncertainties included in the plot are calibrated as discussed in Section 4.2. For each plot we also report in the yellow box the *bias* ($\mu$) and the 16th and 84th percentiles of the error distribution ($\sigma^+$ and $\sigma^-$, respectively). Further details in Section 4.5.

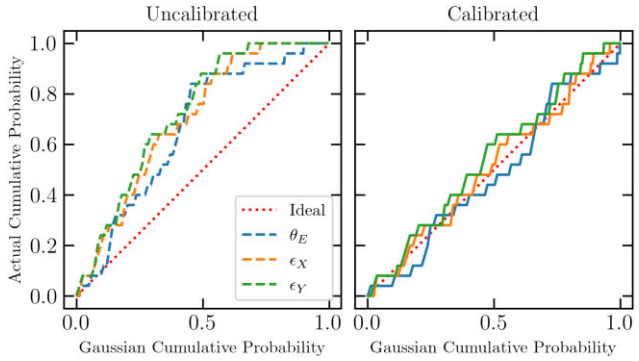


**Figure 7.** Reliability plots generated by applying the LEMON algorithm to the pilot sample of real lenses observed by the HST in the SLACS survey (Bolton et al. 2006; Shu et al. 2017). The plots are shown before and after the blind-calibration procedure. It is possible to observe the significant increase in the reliability of the estimated uncertainties. Further details are given in Section 4.5.

### 5.1 Comparison with the literature

The unprecedented number of lenses expected by next-generation facilities will represent a significant challenge to our analysis capabilities. The urgency of this issue justifies the presence, in the current literature, of at least two groups of studies aiming to automatize the lens-modelling procedure. The first of these is focused on existing 'classical' algorithms, trying to reduce their computational cost (e.g. Gu et al. 2022) or to decrease the needed human intervention (e.g. Etherington et al. 2022). A second group (also including this paper) tries to change the paradigm by involving machine learning algorithms in the task. This section focuses on comparing our code LEMON with the algorithms proposed by this second group. However, it is crucial to underline how a fair comparison between the different codes should consider the difference between the data sets employed to train and test the various algorithms. For instance, some studies (Hezaveh et al. 2017; Perreault Levasseur et al. 2017; Fagin et al., in preparation) focused on HST- or Euclid- like simulations, while others used ground-based simulated images (Bom et al. 2019; Schuldt et al. 2021). It is evident how the better image quality obtained by space-based facilities allows a more accurate lens-modelling (see e.g. the result by Pearson et al. 2019 employing both kinds of simulations). Similar effects are expected by the employment of more realistic simulations. Using real galaxies as deflectors and lensed sources (Hezaveh et al. 2017; Pearson et al. 2021; Schuldt et al. 2021) increases the realism of the simulations but also decreases the accuracy of the lens-modelling, as shown by Pearson et al. (2021). Analogously, the subtraction of the lens light generally produces higher accuracies with respect to studies employing the complete images (see e.g. Hezaveh et al. 2017; Pearson et al. 2019). Finally, it is worthwhile to underline that, in this study, we employed an uniform sampling of the Einstein radius for our validation set to assess the accuracy of the LEMON algorithm on all the possible lensing configurations. On the contrary, most of the aforementioned studies preferred a 'realistically sampled' data set, with a higher percentage of lenses with smaller Einstein radii (that represent the majority of the existing lenses, see e.g. Collett 2015). This choice can produce significant differences in the accuracy of the algorithms when these are measured through the 68th percentile of the scatter distribution commonly employed in this kind of studies. Since all these issues could be overcome only by training and testing all the codes on the same data set, we will neglect the likely effects due to the data sets to focus on other important differences.

(i) The first of these concerns the kind of algorithm employed. Hezaveh et al. (2017); Pearson et al. (2019), and Schuldt et al. (2021) used CNN-based lens-modeller. The most significant difference between these codes and LEMON is the possibility for our algorithm to estimate accurate uncertainties for the predicted parameters. Furthermore, the broad consistency of the accuracies reported by the different codes suggests that the estimation of the uncertainties does not sensibly affects the accuracy of the algorithm's point estimates.

(ii) For the algorithms based on a BNN (Perreault Levasseur et al. 2017; Bom et al. 2019; Pearson et al. 2021; Schuldt et al. 2023), one of the main differences is the calibration performed on the uncertainties predicted by the network. While the other codes changed the keep-rate of the dropout layers to achieve a Gaussian coverage of the test-set, the LEMON algorithm is the only one to employ the a posteriori procedure discussed in Section 4.2. As shown in Figs 3 and 7, our procedure allows an effective calibration of the uncertainties both on simulated and real data.

(iii) A further consideration concerns the kind of architecture employed by the different codes. While Pearson et al. (2019), Schuldt et al. (2021), and Pearson et al. (2021) used a 'classic' CNN, other studies employed a ResNet (i.e. the same architecture of LEMON, but with different amounts of layers) and others much more complex architectures (Hezaveh et al. 2017; Perreault Levasseur et al. 2017;

Bom et al. 2019; Schuldt et al. 2023). These differences strongly impact the dimension of the data sets required to successfully train the algorithms and – therefore – the possibility of training them only by employing the first data provided by Euclid or LSST. The slight differences in the accuracies achieved by these codes could suggest a minor role of the employed architecture in the effectiveness of the lens modelling.

(iv) Finally, it is noteworthy to underline how – among the aforementioned studies – only Hezaveh et al. (2017) validated the accuracies achieved on the simulations with an application to a pilot sample of nine real lenses observed in the SL2S program (Sonnenfeld et al. 2013). As discussed in Section 4.5, this procedure can highlight a significant difference between the accuracies, giving precious insights about the realism of the simulations employed in the training and test phases.

## 6 CONCLUSION

In this work, we presented LEMON: a new machine learning algorithm able to perform the fast automated analysis of strong gravitational lenses. The algorithm is based on a BNN: a new generation of machine learning algorithms able to associate reliable confidence intervals to the predicted parameters. This paper is the first of the LEMON series and is focused on the training of the algorithm and its first blind application to several data sets of both simulated and a pilot sample of real gravitational lenses observed with HST as a part of the SLACS program (Bolton et al. 2006; Shu et al. 2017). LEMON has been trained on two simulated data sets generated to resemble the imaging capabilities of the HST and the forthcoming ESA *Euclid* satellite. The main task of the algorithm is to estimate the three parameters of the SIE model ($\theta_E$, $\epsilon_x$, and $\epsilon_y$) and the related uncertainties. After the training, the algorithm has been applied to two simulated data sets generated following the same simulation procedure, to independently assess the accuracy of the estimates and the reliability of the modelled uncertainties.

The main scientific result consists in the modelling time of just 0.5 s for each image, at least two orders of magnitudes lower than classical techniques. Moreover, the results obtained on these data sets are perfectly comparable with the previous studies based on analogous machine-learning algorithms and present in the current literature. Finally, since the estimated uncertainties are able to produce a Gaussian statistical coverage of the analysed data set, we concluded that the confidence intervals are perfectly reliable and able to allow a vast set of scientific application. To provide an additional confirmation of the efficiency and accuracy of our algorithm, we applied the LEMON network to a pilot sample of real lenses discovered in the SLACS survey. We found that the algorithm is able to predict the parameters of each lens without any significant bias and to provide well-calibrated uncertainties. However, the application to real data provided a higher scatter between the real and predicted values when compared to the results obtained on the simulated data sets. This issue – most likely explainable with the not-sufficient realism of the simulations in the training set – will be afforded in detail in the forthcoming papers of the LEMON series.

## ACKNOWLEDGEMENTS

We thank the anonymous referee for his/her useful suggestions to improve the paper. FG and CT thank Georgios Vernardos, Jose Maria Diego, and Philip J. Marshall for the fruitful discussion on the paper. FG acknowledges the support from grant PRIN MIUR 2017–20173ML3WW_001. FG and CT acknowledge funding from INAF research grant 2022 LEMON. This research is based on observations made with the NASA/ESA Hubble Space Telescope obtained from the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5–26555. These observations are associated with programs 10563, 10494, 10886, and 10798.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, FG, upon reasonable request. The code employed in this study is freely available in a GitHub repository: https://github.com/fab-gentile/LEMON.

## REFERENCES

Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available at: https://www.tensorflow.org/
Aniyan A. K., Thorat K., 2017, ApJS, 230, 20
Auger M. W., Treu T., Gavazzi R., Bolton A. S., Koopmans L. V. E., Marshall P. J., 2010a, ApJL, 721, L163
Auger M. W., Treu T., Bolton A. S., Gavazzi R., Koopmans L. V. E., Marshall P. J., Moustakas L. A., Burles S., 2010b, ApJ, 724, 511
Barnabè M., Spiniello C., Koopmans L. V. E., Trager S. C., Czoske O., Treu T., 2013, MNRAS, 436, 253
Bartelmann M., 2010, Class. Quantum Gravity, 27, 233001
Birrer S., Amara A., 2018, Phys. Dark Univ., 22, 189
Bolton A. S., Burles S., Koopmans L. V. E., Treu T., Moustakas L. A., 2006, ApJ, 638, 703
Bolton A. S., Burles S., Koopmans L. V. E., Treu T., Gavazzi R., Moustakas L. A., Wayth R., Schlegel D. J., 2008, ApJ, 682, 964
Bom C., Poh J., Nord B., Blanco-Valentin M., Dias L., 2019, preprint (arXiv:1911.06341)
Canameras R. et al., 2020, A&A, 644, 27
Charnock T., Perreault-Levasseur L., Lanusse F., 2020, preprint (arXiv:2006.01490)
Chatterjee S., 2019, PhD thesis, Univ. Groningen/University of Groningen
Chollet F. et al., 2015, Keras. Available at: https://keras.io
Cobb A. D. et al., 2019, AJ, 158, 33
Collett T. E., 2015, ApJ, 811, 20
Congdon A. B., Keeton C. R., 2018, Principles of Gravitational Lensing: Light Deflection as a Probe of Astrophysics and Cosmology, Springer International Publishing, New York
Covone G. et al., 2009, ApJ, 691, 531
Cropper M. et al., 2018, in Lystrup M., MacEwen H. A., Fazio G. G., Batalha N., Siegler N., Tong E. C.eds, SPIE Conf. Ser., Vol. 10698, Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 1069828,
Dalal N., Kochanek C. S., 2002, ApJ, 572, 25
Dieleman S., Willett K. W., Dambre J., 2015, MNRAS, 450, 1441
Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018, MNRAS, 476, 3661
Einstein A., 1915, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften, Königlich Preußischen Akademie der Wissenschaften, Berlin, p. 831
Eisenstein D. J. et al., 2001, AJ, 122, 2267
Escamilla-Rivera C., Carvajal Quintero M. A., Capozziello S., 2020, J. Cosmol. Astropart. Phys., 2020, 008
Etherington A. et al., 2022, MNRAS, 517, 3275
Fleury P., Larena J., Uzan J.-P., 2021, J. Cosmol. Astropart. Phys., 2021, 024
Fluri J., Kacprzak T., Lucchi A., Refregier A., Amara A., Hofmann T., Schneider A., 2019, Phys. Rev. D, 100, 063514
Gal Y., Ghahramani Z., 2015a, preprint (arXiv:1506.02142)
Gal Y., Ghahramani Z., 2015b, preprint (arXiv:1506.02158)

Gal Y., Hron J., Kendall A., 2017, preprint (arXiv:1705.07832)
Galan A., Vernardos G., Peel A., Courbin F., Starck J.-L., 2022, A&A, 668, A155
Gavazzi R., Treu T., Marshall P. J., Brault F., Ruff A., 2012, ApJ, 761, 170
Gentile F. et al., 2022, MNRAS, 510, 500
Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge
Gu A. et al., 2022, ApJ, 935, 49
Guo C., Pleiss G., Sun Y., Weinberger K. Q., 2017, in Proc. 34th International Conference on Machine Learning. Vol. 70, ICML'17. JMLR, Sydney, p. 1321
He K., Zhang X., Ren S., Sun J., 2015, preprint (arXiv:1512.03385)
He Z. et al., 2020, MNRAS, 497, 556
Hezaveh Y., Dalal N., Holder G., Kisner T., Kuhlen M., Levasseur L. P., 2016, J. Cosmol. Astropart. Phys., 2016, 048
Hezaveh Y. D., Perreault Levasseur L., Marshall P. J., 2017, Nature, 548, 555
Hortúa H. J., Volpi R., Marinelli D., Malagò L., 2020, Phys. Rev. D, 102, 103509
Huchra J., Gorenstein M., Kent S., Shapiro I., Smith G., Horine E., Perley R., 1985, AJ, 90, 691
Jacobs C. et al., 2019a, ApJS, 243, 17
Jacobs C. et al., 2019b, MNRAS, 484, 5330
Jordan M. I., Ghahramani Z., Jaakkola T. S., Saul L. K., 1999, An Introduction to Variational Methods for Graphical Models. MIT Press, Cambridge, p. 105
Jullo E., Kneib J. P., Limousin M., Elíasdóttir Á., Marshall P. J., Verdugo T., 2007, New J. Phys., 9, 447
Keeton C. R., Kochanek C. S., Seljak U., 1997, ApJ, 482, 604
Kendall A., Gal Y., 2017, preprint (arXiv:1703.04977)
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Koekemoer A. M. et al., 2007, ApJS, 172, 196
Koopmans L. V. E., 2005, MNRAS, 363, 1136
Koopmans L. V. E., Treu T., Bolton A. S., Burles S., Moustakas L. A., 2006, ApJ, 649, 599
Kormann R., Schneider P., Bartelmann M., 1994, A&A, 284, 285
Kull M., Filho T. S., Flach P., 2017, in Singh A., Zhu J., eds, Proc. 20th International Conference on Artificial Intelligence and Statistics. Vol. 54, PMLR, Lauderdale, p. 623
Kullback S., Leibler R. A., 1951, Ann. Math. Stat., 22, 79
LSST Science Collaboration et al., 2009, preprint (arXiv:0912.0201)
Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
LeCun Y., Bengio Y., Hinton G., 2015, Nature, 521, 436
Lefor A. T., Futamase T., Akhlaghi M., 2013, New A Rev., 57, 1
Li R. et al., 2020, ApJ, 899, 30
Mao S., Schneider P., 1998, MNRAS, 295, 587
Massey R., Stoughton C., Leauthaud A., Rhodes J., Koekemoer A., Ellis R., Shaghoulian E., 2010, MNRAS, 401, 371
Metcalf R. B. et al., 2019, A&A, 625, A119
Napolitano N. R. et al., 2020, ApJ, 904, L31
Nightingale J. W., Dye S., Massey R. J., 2018, MNRAS, 478, 4738
Oguri M. et al., 2006, AJ, 132, 999
Park J. W., Wagner-Carena S., Birrer S., Marshall P. J., Lin J. Y.-Y., Roodman A., LSST Dark Energy Science Collaboration, 2021, ApJ, 910, 39
Pearson J., Li N., Dye S., 2019, MNRAS, 488, 991
Pearson J., Maresca J., Li N., Dye S., 2021, MNRAS, 505, 4362
Perreault Levasseur L., Hezaveh Y. D., Wechsler R. H., 2017, ApJL, 850, L7
Petrillo C. E. et al., 2017, MNRAS, 472, 1129
Petrillo C. E. et al., 2019a, MNRAS, 482, 807
Petrillo C. E. et al., 2019b, MNRAS, 484, 3879
Pourrahmani M., Nayyeri H., Cooray A., 2018, ApJ, 856, 68
Refsdal S., 1964, MNRAS, 128, 307
Ryon J. E., 2022, ACS Instrument Handbook for Cycle 30 v. 21.0. Vol. 21, p. 21
Scaramella R. et al., 2021, A&A, 662, A112
Schneider P., Ehlers J., Falco E. E., 1992, Gravitational Lenses. Springer, New York
Schuldt S., Suyu S. H., Meinhardt T., Leal-Taixé L., Cañameras R., Taubenberger S., Halkola A., 2021, A&A, 646, A126
Schuldt S., Cañameras R., Shu Y., Suyu S. H., Taubenberger S., Meinhardt T., Leal-Taixé L., 2023, A&A, 671, A147
Serjeant S., 2014, ApJL, 793, L10
Sérsic J. L., 1963, Boletin de la Asociacion Argentina de Astronomia La Plata Argentina, 6, 41
Shajib A. J., Treu T., Birrer S., Sonnenfeld A., 2021, MNRAS, 503, 2380
Shu Y. et al., 2017, ApJ, 851, 48
Sonnenfeld A., 2021a, A&A, 659, 11
Sonnenfeld A., 2021b, A&A, 659, 9
Sonnenfeld A., 2021c, A&A, 656, A153
Sonnenfeld A., Cautun M., 2021, A&A, 651, A18
Sonnenfeld A., Gavazzi R., Suyu S. H., Treu T., Marshall P. J., 2013, ApJ, 777, 97
Sonnenfeld A., Treu T., Marshall P. J., Suyu S. H., Gavazzi R., Auger M. W., Nipoti C., 2015, ApJ, 800, 94
Sonnenfeld A., Jaelani A. T., Chan J., More A., Suyu S. H., Wong K. C., Oguri M., Lee C.-H., 2019, A&A, 630, A71
Spiniello C., Koopmans L. V. E., Trager S. C., Czoske O., Treu T., 2011, MNRAS, 417, 3000
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929
Tortora C., Napolitano N. R., Romanowsky A. J., Jetzer P., 2010, ApJL, 721, L1
Treu T., 2010, ARA&A, 48, 87
Treu T., Koopmans L. V. E., 2004, ApJ, 611, 739
Treu T., Auger M. W., Koopmans L. V. E., Gavazzi R., Marshall P. J., Bolton A. S., 2010, ApJ, 709, 1195
Treu T., Dutton A. A., Auger M. W., Marshall P. J., Bolton A. S., Brewer B. J., Koo D. C., Koopmans L. V. E., 2011, MNRAS, 417, 1601
Vegetti S., Koopmans L. V. E., 2009, MNRAS, 392, 945
Vegetti S., Koopmans L. V. E., Auger M. W., Treu T., Bolton A. S., 2014, MNRAS, 442, 2017
Vernardos G., Koopmans L. V. E., 2022, MNRAS, 516, 1347
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Wagner-Carena S., Park J. W., Birrer S., Marshall P. J., Roodman A., Wechsler R. H., LSST Dark Energy Science Collaboration, 2021, ApJ, 909, 187
Weaver J. R. et al., 2022, ApJS, 258, 11
Wong K. C. et al., 2020, MNRAS, 498, 1420
Zadrozny B., Elkan C., 2001, in Proc. Eighteenth International Conference on Machine Learning. ICML'01. Morgan Kaufmann Publishers Inc., San Francisco, p. 609
Zadrozny B., Elkan C., 2002, in Proc. Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '02. Association for Computing Machinery, New York, p. 694
Zwicky F., 1937, Phys. Rev., 51, 290
de Vaucouleurs G., 1948, Annales d'Astrophysique, 11, 247

## APPENDIX: ARCHITECTURE

The architecture of the LEMON algorithm is reported in Fig. A1.

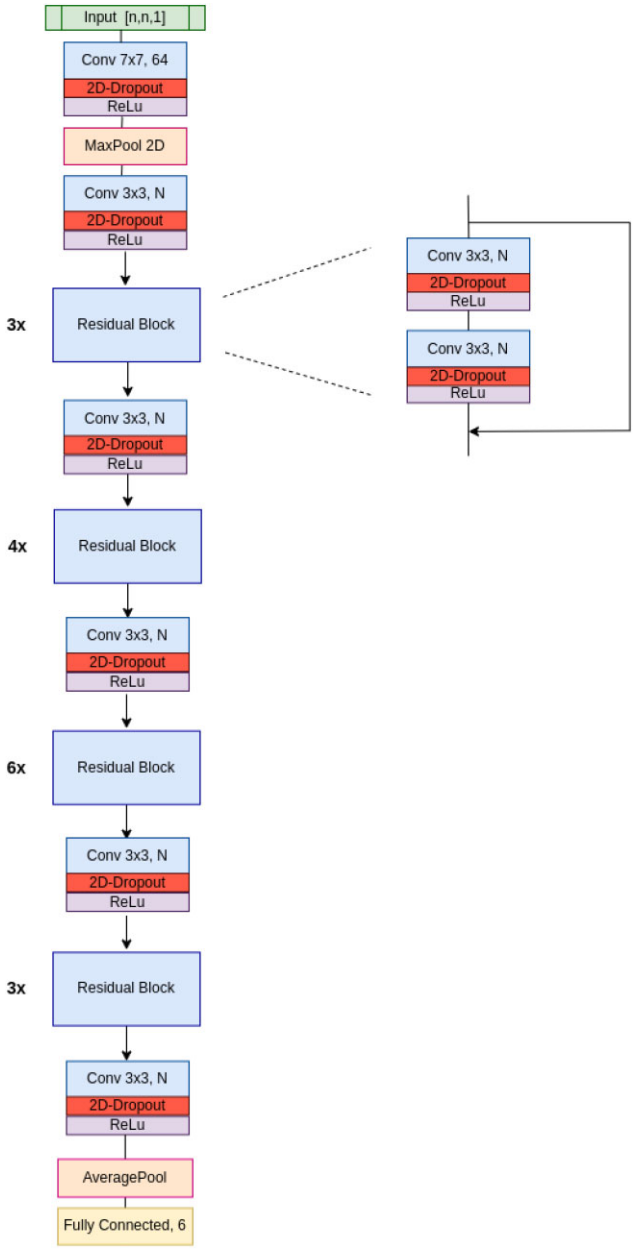


**Figure A1.** Schematic representation of the architecture employed in the LEMON algorithm. The number of feature maps 'N' doubles in each convolutional/residual block, while the shape of the input image 'n' is different for the Euclid and HST mocks.

This paper has been typeset from a TEX/LATEX file prepared by the author.