Differentiable Strong Lensing: Uniting Gravity and Neural Nets through Differentiable Probabilistic Programming

Marco Chianese,[1,*] Adam Coogan,[1,†] Paul Hofma,[1,‡] Sydney Otten,[1,2,§] and Christoph Weniger[1,¶]

[1] *Gravitation Astroparticle Physics Amsterdam (GRAPPA), Institute for Theoretical Physics Amsterdam and Delta Institute for Theoretical Physics, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands*

[2] *Institute for Mathematics, Astrophysics and Particle Physics (IMAPP), Radboud University, Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands*

(Dated: March 11, 2020)

The careful analysis of strongly gravitationally lensed radio and optical images of distant galaxies can in principle reveal DM (sub-)structures with masses several orders of magnitude below the mass of dwarf spheroidal galaxies. However, analyzing these images is a complex task, given the large uncertainties in the source and the lens. Here, we leverage and combine three important computer science developments to approach this challenge from a new perspective. (a) Convolutional deep neural networks, which show extraordinary performance in recognizing and predicting complex, abstract correlation structures in images. (b) Automatic differentiation, which forms the technological backbone for training deep neural networks and increasingly permeates 'traditional' physics simulations, thus enabling the application of powerful gradient-based parameter inference techniques. (c) Deep probabilistic programming languages, which not only allow the specification of probabilistic programs and automatize the parameter inference step, but also the direct integration of deep neural networks as model components. In the current work, we demonstrate that it is possible to combine a deconvolutional deep neural network trained on galaxy images as source model with a fully-differentiable and exact implementation of the gravitational lensing physics in a single probabilistic model. This does away with hyperparameter tuning for the source model, enables the simultaneous optimization of nearly one hundred source and lens parameters with gradient-based methods, and allows the use of efficient gradient-based Hamiltonian Monte Carlo posterior sampling techniques. We consider this work as one of the first steps in establishing *differentiable probabilistic programming* techniques in the particle astrophysics community, which have the potential to significantly accelerate and improve many complex data analysis tasks.

I. INTRODUCTION

Deep learning has advanced dramatically over the past decade, with accurate image classifiers [**?**] and a host of generative methods such as variational autoencoders [**?** **?**], generative adversarial networks [**?** **?** **?**] and flow-based models [**?**] capable of producing novel, realistic images counting among its successes. These methods have also been applied to the physical sciences [**?**], with topics including deblending galaxy images [**?**], generating weak gravitational lens convergence maps [**?**], and classifying LHC jet events [**?**]. However, there has been considerably less scientific investigation into leveraging *automatic differentiation* (AD) [**?**], the core technology enabling training of neural networks with millions of parameters using gradient descent. AD libraries [**?** **?** **?**] make it possible to take exact derivatives of complex computational functions by using the chain rule to compose the gradients of individual mathematical operations. This paper focuses on how deep generative models can be combined with known physics using AD to create a new pipeline for analyzing images of galaxy-galaxy strong lensing systems.

The approach of creating automatically-differentiable mechanistic models that can be combined with deep neural networks is known as *differentiable programming* [**?**]. Differentiable programming has recently been applied to several engineering problems, with demonstrated benefits for challenging optimization problems in various domains. Constructing a differentiable ray-tracer [**?**] and image-processing algorithms [**?** **?**] simplifies the inverse problem of fitting parameters describing the lighting, materials and objects in a scene from a photograph, as well as the forward problem of optimizing image-processing pipelines. Differentiable rigid-body physics engines make it possible to train deep learning controllers for robots in accurate environments [**?**]. These simulators can also be combined with neural networks to model physical systems based on video and predict their future behavior [**?**].

The physical phenomenon in which we are interested, strong lensing, is a gravitational effect through which an astrophysical light source is observed in distorted, multiple images in the sky due to the deflection of its light by matter distributed along the line of sight [**?**]. It has become one of the main ways to probe the small-scale structure of dark matter halos, since subhalos with mass below $\sim 10^8\ M_\odot$ do not host stars and are thus

* m.chianese@uva.nl
† a.m.coogan@uva.nl
‡ paul.hofma@student.uva.nl
§ s.m.m.otten@uva.nl
¶ c.weniger@uva.nl

invisible [? ?]. The detection (or non-detection) of these subhalos is a critical tool for discriminating among different paradigms of dark matter (DM) [? ?]. In the standard ΛCDM cosmological model, the large-scale structures of the Universe form through the collapse of primordial density fluctuations. The matter content of the Universe is dominated by non-relativistic and almost collisionless substance dubbed cold dark matter (CDM). In this scenario, an abundance of small DM substructures is formed as confirmed by ab-initio N-body cosmological simulations [?]. On the other hand, alternative well-motivated particle DM scenarios such as warm dark matter (WDM) [? ?] and self-interacting DM particles [? ?] predict strongly suppression of the formation of low-mass DM substructures. Recent analyses of current strong gravitational lensing data of extended sources [? ? ? ? ?] and quasars [? ? ?] have already demonstrated sensitivity to DM haloes with masses larger than 10^8 $M_\odot$. In the near future, new observatories like DES [?], LSST [? ? ?], Euclid [?], and next-generation observatories like ELT [?] will observe thousands of strong lensing systems with very high precision, pushing the sensitivity of lensing probes of DM substructures to even lower masses. Moreover, these observations will be dominated by lenses at high redshift, increasing the likelihood of detecting small DM haloes along the line-of-sight [?].

In analyzing a galaxy-galaxy strong lens, the observed lensed image is reconstructed by simultaneously modeling the surface brightness of the source and the matter distribution of the lens galaxy. This requires parametrizing both of these components. While N-body simulations show that the density profiles of galactic DM halos are well-described by various analytic profiles [?], the distribution of the source's light is more complicated. The Sersic brightness profile [?] is a common choice (see e.g. Ref. [?]), but is inadequate for high-resolution observations and modeling high-redshift source galaxies, which generally have more complex morphologies than low-redshift ones. Another class of methods computes the source brightness profile on a grid by linearly inverting the observed lensed image given a fixed lensed model [? ?]. This requires choosing a specific prior to regularize the source, depending on two-point quantities calculated between pairs of pixels as well as hyperparameters. These methods can be cast in a fully Bayesian framework and performed on an adaptive grid [? ?]. The public code PyAutoLens implements this analysis strategy [? ?]. Extensions of these methods include grid-free approaches using radial basis functions centered on image pixels ray-traced back to the source plane [?] and methods decomposing the source as a sum of shapelets [?]. These methods are available in the SaWLens2 [?] and Lenstronomy [? ?] software packages.

Recently, deep learning methods have been brought to bear on strong lensing analyses. Refs. [? ?] applied a convolutional neural network (CNN) [?] to infer the parameters of the lens matter distribution with unprecedented speed. This CNN was later coupled to an optimizer controlled by another CNN that performs a linear inversion to recover the source's pixelated brightness profile without requiring regularization hyperparameters [?]. Both CNNs are trained using supervised learning on mock lensing datasets. Very recently, supervised CNNs have been trained using toy models of lensing systems containing substructure to differentiate between different DM models [?] and infer parameters in the subhalo mass function [?]. Moreover, in Ref. [?] a supervised CNN was trained to classify whether or not a lensing system observation contained detectable substructure, thus identifying images worthy of follow-up observations and analyses.

In this work, we construct a differentiable programming pipeline consisting of a deep generative model for the source galaxy brightness profile and a physics model for the lens. In particular, we use a variational autoencoder (VAE) [? ?] to learn a parametric description of the source galaxy's light. In contrast with the aforementioned deep learning strategies for lensing, the VAE is trained in an unsupervised manner on unlensed galaxies; a similar VAE was previously constructed in Ref. [?]. The lensing physics is implemented by solving the Poisson equation for analytical models of the lens and external shear. Since our analysis pipeline is modularized, it is straightforward to change parameter's priors or include additional lensing effects such as line-of-sight halos and subhalos without having to retrain the source VAE. The source model can also be improved independently from the rest of the code.

Our pipeline is implemented in the PyTorch machine learning framework [?], which contains an automatic differentiation engine and graphical processing unit (GPU)-accelerated functions for array computations. As with the differentiable programming examples mentioned above, pervasive AD makes it straightforward to fit the high-dimension parameter vector of our lensing model using gradient-based methods. We also exploit the Pyro [?] probabilistic programming language to characterize the uncertainties of our parameter fits. Pyro is capable of sampling model parameters, computing likelihoods, and automatically performing inference with tools such as Markov chain Monte Carlo (MCMC) and variational inference [? ?] on arbitrary probabilistic models, even those with stochastic control flow. While probabilistic programming has existed for a long time [? ?], it has only recently been integrated with automatic differentiation, making parameter inference possible even for models including neural networks [? ? ? ? ? ?]. In the context of high energy physics, recent work has reframed the Sherpa event generator using probabilistic programming, enabling more efficient simulation of rare events [? ?]. In our present work, we again leverage the differentiability of our pipeline by employing Hamiltonian Monte Carlo (HMC) [? ?], a gradient-based MCMC method, to efficiently sample from the high-dimension posterior for the lens and source parameters. The unique meshing

of an exact physics model with a deep generative source model in an AD-compatible, probabilistic programming framework makes this paper as one of the first differentiable probabilistic programs to our knowledge.

We begin this paper by discussing the variational autoencoder model for source galaxies in Section **??**, and describing the training procedure and validation tests. In Section **??** we review the physics of strong gravitational lensing, and introduce all the ingredients required to generate lensed images. In particular, we define the lens and the source models considered in the present paper. Section **??** delineates our lensing inference pipeline. In Section **??** we test the pipeline on two mock galaxy-galaxy lensing systems, and discuss the parameter fitting and the posterior analysis. We draw our conclusions in Section **??**.

II. DEEP GENERATIVE MODELS FOR SOURCE GALAXIES

This section describes how we use a variational autoencoder (VAE) to construct a parametric model for source galaxy light. After reviewing VAEs and explaining the architecture we selected, we describe how ours is trained and present tests validating the performance of the parametric model we use it to construct.

A. Variational Autoencoders

Natural images (such as those of galaxies) lie on or near a low-dimensional submanifold in the space of all possible images [**?**]. This motivates the concept of *probabilistic latent variable models* [**?**], where a datum $\mathbf{x}$ (such as a galaxy image) is related to a latent variable $\mathbf{z}$ through a conditional probability density $p(\mathbf{x}|\mathbf{z})$, and the latent variables are assumed to have a prior density $p(\mathbf{z})$. We will use this to construct a parametric model for galaxy images, where $\mathbf{z}$ maps onto the mean of the decoding distribution and has prior $p(\mathbf{z})$.

The variational autoencoder (VAE) was constructed to efficiently approximate models of this form [**?** **?**]. It approximates the conditional density $p(\mathbf{x}|\mathbf{z})$ with a decoder $d_\theta(\mathbf{x}|\mathbf{z})$ whose parameters are functions represented by a neural network; θ represents the network's parameters. The VAE also includes an encoder, $e_\phi(\mathbf{z}|\mathbf{x})$, that similarly approximates the conditional distribution $p(\mathbf{z}|\mathbf{x})$ using a neural network with parameters ϕ. Figure **??** illustrates this structure. The decoder and encoder are typically both taken to be diagonal Gaussians:

$$d_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mu_d(\mathbf{z}), \sigma_d) \tag{1}$$
$$e_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_e(\mathbf{x}), \sigma_e(\mathbf{x})). \tag{2}$$

In our notation $\mathcal{N}(x|\mu, \sigma)$ denotes that x follows a normal distribution with mean μ and standard deviation σ. The functions μ_d, μ_e and σ_e are implemented using neural networks, and the decoder's standard deviation σ_d is a constant hyperparameter. We identify σ_d with the approximate standard deviation of the Gaussian noise in our training dataset.

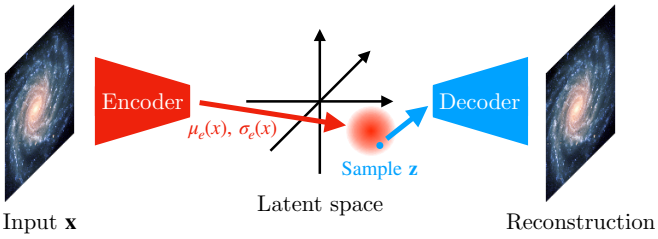

FIG. 1. **Diagram of a variational autoencoder.** The diagram shows how an input image $\mathbf{x}$ is passed through the encoder (red trapezoid) to yield an encoding distribution (red blob) with mean and standard deviation $\mu_e(\mathbf{x})$ and $\sigma_e(\mathbf{x})$. A point is then sampled from this distribution (blue dot) and passed through the decoder (blue trapezoid) to yield a reconstructed image.

Training VAE requires a dataset $\{\mathbf{x}\}$ as well as selecting the latent space's dimensionality and prior $p(\mathbf{z})$, which is typically taken to be $\mathcal{N}(0, I)$.[1] Ideally, the VAE would be trained by maximizing the marginal likelihood of the training dataset, which requires computing the integral

$$p_\theta(\mathbf{x}) = \int d\mathbf{z}\; d_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \tag{3}$$

for each training point. Note that the marginal likelihood only depends on the decoder. However, this integral is generally intractable. The difficulty is circumvented by defining an alternative objective function called the evidence lower bound (ELBO), whose derivation is reviewed in Appendix **??**. The reason the encoder network was introduced is because it is required to compute the ELBO, whose value for each training point is given by[2]

$$\log p_\theta(\mathbf{x}) \geq \sum_i \mathrm{ELBO}(\mathbf{x}; \theta, \phi) \tag{4}$$
$$\equiv \mathbb{E}_{e_\phi(\mathbf{z}|\mathbf{x})}\left[\log d_\theta(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left[e_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right].$$

The first term is related to the quality of the reconstruction obtained by passing an image through the encoder followed by the decoder: maximizing this term improves reconstruction quality. The function $D_{KL}[\cdot||\cdot]$ in the second term is the Kullback–Leibler divergence, which measures the difference between its two argument probability distributions. This term's maximization drives the averaged encoding distribution to look more like the prior $p(\mathbf{z})$. The VAE is trained to maximize the ELBO

[1] Several works have studied more complex prior distributions, including learnable ones [**?** **?** **?** **?** **?**].

[2] The objective eq. (**??**) is equivalent to the β-VAE [**?**] objective obtained by taking $\beta = \sigma^2$ and setting the decoder's standard deviation to 1.

with stochastic gradient descent by taking random minibatches of training images. The second term can often be computed analytically, which makes computing its gradient straightforward. A Monte Carlo estimate of the first term can be computed. Appendix **??** explains how the derivative of this estimate can be computed.

Our VAE architecture choice is influenced by three widespread trends in deep convolutional neural network design [**?**]: replacing pooling functions with strided convolutions, relying on convolutional layers rather than fully-connected ones when possible, and interleaving batch normalization layers [**?**]. In more detail, the encoder uses five blocks made up of strided convolutions, batch normalization layers and LeakyReLU (leaky rectified linear unit) activation functions [**?**]. The output from these blocks is processed by two separate dense layers, which give the $\mu_e(\mathbf{x})$ and $\sigma_e(\mathbf{x})$, the parameters of the encoding distribution. The decoder is similarly built from five blocks consisting of a transposed convolution, batch normalization layers and ReLU (rectified linear unit) [**?**] activation function. The output of last block is passed through a tanh activation function to produce $\mu_d(\mathbf{x})$, the pixel values of which are thus restricted to lie within $[-1, 1]$. The standard deviation of the decoding distribution was set to $\sigma_d = 1/50$, which is the approximate standard deviation of the noise in our training dataset. We also studied the impact of making σ_d a trainable parameter (as recommended in Ref. [**?**]), in which case we find it converges to approximately this value. We used 64 latent-space dimensions. Complete details of the architecture and weight initializations can be found in Appendix **??**, where we also describe other architectures with which we experimented.

After training the VAE as described in the next subsection, we obtain a parametric model for galaxy images where $\mathbf{z}$ corresponds to the image $\mu_d(\mathbf{z})$, and has prior $p(\mathbf{z})$. However, it is a well-known and difficult-to-solve problem that the assumed prior $p(\mathbf{z})$ does not actually match the "aggregate prior" obtained by encoding all the training dataset, [**?** **?**]

$$q_\phi(\mathbf{z}) \equiv \frac{1}{N}\sum_{i=1}^{N} e_\phi\left(\mathbf{z}|\mathbf{x}^{(i)}\right). \tag{5}$$

This mismatch can cause problems when performing maximum a posteriori (MAP) estimation of $\mathbf{z}$ for a given galaxy image. The assumed prior can drag $\mathbf{z}$ into unrealistic regions of parameter space far from the location preferred by the likelihood, where the corresponding decoded image $\mu_d(\mathbf{z})$ does not look like a galaxy.

A variety of methods have been proposed to address this problem, including constructing simple priors using $q_\phi(\mathbf{z})$ [**?** **?**], fitting $q_\phi(\mathbf{z})$ with a second VAE after training the primary one [**?**], or using normalizing flows [**?** **?** **?** **?**]. Here we follow the simpler approach of creating a weakly-informative prior for $\mathbf{z}$ by fitting a multivariate normal distribution to the set of encoded means of the training data $\{\mu_e\left(\mathbf{x}^{(i)}\right)\}_{i=1}^{N}$ and rescaling its covariance matrix by a factor of 9. This prior roughly confines the latent variable to realistic regions of the latent space while remaining diffuse enough that the likelihood drives MAP estimates of $\mathbf{z}$ from observations.

B. Training

We construct a dataset for training the VAE starting from 56,062 images of galaxies in the COSMOS field taken by the Hubble Space Telescope [**?** **?** **?**]. These were used for the GREAT3 weak gravitational lensing challenge [**?** **?**]. An estimate of the pixel noise is included for each image, which is assumed to be Gaussian and uncorrelated. Parameters for Sersic profiles fit to each of the galaxies are also provided. Details about the image processing and parameter fits can be found in Appendix E of Ref. [**?**].

We discard the small number of galaxies for which the Sersic fits failed or gave unphysical parameters. The dimensions of the images in the dataset vary, so we remove any that are smaller than 64×64 pixels. Images with unequal width and height are cropped into squares, and all are then downscaled to 64×64 pixels. Finally, many of the galaxies are extremely bright and small, and some are nearly indistinguishable from the pixel noise. In our experiments both of these degraded the quality of the VAE's reconstructions, in the former case by biasing it towards producing only compact, bright galaxies and in the later case by degrading the fidelity of the reconstructions. We find a useful, very heuristic way of removing these is to make a cut on the image signal-to-noise quantity $\max(\mathcal{I}_{1/4})/\sigma_{\rm noise}$, where $\sigma_{\rm noise}$ is the standard deviation of the pixel noise and $\mathcal{I}_{1/4}$ is the image downscaled by a factor of 4. Restricting this quantity to lie between 15 and 50 reduces the dataset to 17,543 images. We then split these into a training, test and validation datasets consisting of 15,500, 500 and 1,543 images respectively. This is a fairly small training dataset by industrial machine learning standards [**?**], but could be greatly augmented by future astronomical observations.

The training and validation sets are preprocessed by dividing each image by its maximum pixel value. The VAE's parameters are optimized to minimize the ELBO of the training using the Adam optimizer [**?**], with a learning rate of 10^{-6}, minibatch size of 32 and momentum parameters $(\beta_1, \beta_2) = (0.5, 0.999)$. The mean ELBO for the images in the validation set is monitored during training. While this decreases at first, it inevitably increases as the VAE starts overfitting the training data. We terminate training at this point ($\sim$ 450 epochs in practice). The test-set images are not seen by the VAE during training.

C. Validation tests

To assess the quality of our parametric model $\mu_d(\mathbf{z})$ for galaxy images, we present the reconstructions of galaxies from the test set that the model has not seen during training in Fig. **??**. We can observe that the reconstructions of our VAE model are denoised versions of the original images that reliably contain galaxy substructure. This indicates the latent space learned by the VAE contain a point corresponding to each of these galaxies, even though they have complex and varied morphologies.

It is also important that the VAE's reconstructions are equivariant under transformations such as rotations, since the parameter inferences by the analysis pipeline should be as well. In Fig. **??**, a galaxy image from the test set is rotated by various angles. Each of the rotated images is then derotated for comparison. The reconstructed images are once again denoised versions of the input images, and it can be seen by inspection that the derotated reconstructions are nearly identical. From this we conclude that our generative model has learned an approximate rotational equivariance, even though the training dataset was not augmented with rotated images to teach this explicitly.

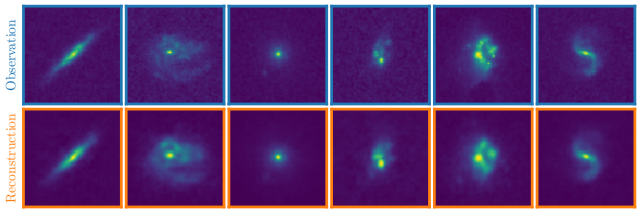

FIG. 2. **Reconstructions of galaxies from the test set.** The input images are shown in the top row and the reconstructions in the bottom row. The reconstructions were obtained by passing the observations $\mathbf{x}$ through the encoder, sampling $\mathbf{z} \sim \mathcal{N}(\mu_e(\mathbf{x}), \sigma_e(\mathbf{x}))$ from the encoded distribution, and taking the mean of the decoded distribution $\hat{\mathbf{x}} = \mu_d(\mathbf{z})$.

III. STRONG LENSING

Here we review the physics of strong gravitational lensing, describe how we model the main lens and external shear, and explain how we construct the source model using the variational autoencoder from the previous section.

A. Strong lensing physics

A gravitational galaxy-galaxy lensing system mainly consists of a background galaxy, playing the role of the source, and a foreground galaxy, acting as the main lens that deflects the light through its gravitational potential.

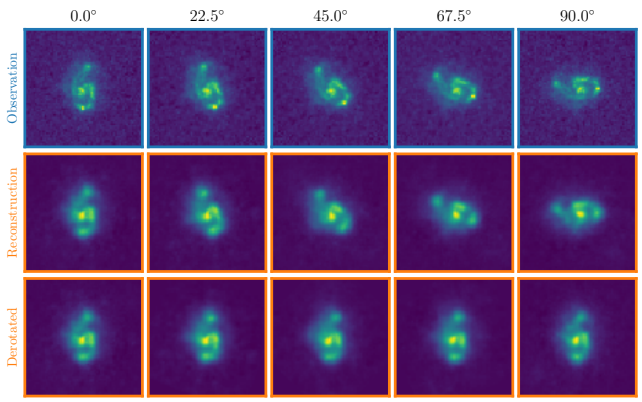

FIG. 3. **Approximate rotational invariance of reconstructions.** The first row shows a galaxy image from the test set and versions rotated counterclockwise by the amount indicated above each column, with new pixels filled using the known noise distribution. The second row shows the corresponding VAE reconstructions, with derotated versions shown in the third row to make comparison simpler. The same color scale is used in each subplot.

In the thin lens approximation, the relation between the two-dimensional angular coordinates of the lens plane $\boldsymbol{\theta}$ and the ones in the source plane $\boldsymbol{\beta}$ is encoded by the lens equation [**?** **?**]

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta}) \,, \tag{6}$$

where $\boldsymbol{\alpha}$ is the displacement field, which defines the deflection experienced by the light ray. The geometry of the system is displayed in Fig. **??**. The displacement field depends on the Newtonian gravitational potential related to the mass distribution of the foreground galaxy. By means of the Poisson equation, it can be expressed as

$$\boldsymbol{\alpha} = \frac{4\,G}{c^2} \frac{D_{\mathrm{OL}} D_{\mathrm{LS}}}{D_{\mathrm{OS}}} \int \Sigma(\boldsymbol{\theta}') \frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{|\boldsymbol{\theta} - \boldsymbol{\theta}'|^2} d^2\theta' \,, \tag{7}$$

where Σ is the surface mass density of the lens, and the quantities D_{OL}, D_{OS}, D_{LS} are the angular diameter distances between the observer and the lens, the observer and the source, the lens and the source, respectively. Moreover, G is the gravitational constant while c is the speed of light.

Since the lens equation (eq. (**??**)) preserves the surface brightness (photon flux density per unit angular area), the image of the system in the lens plane, denoted as $\mathcal{I}_{\mathrm{lens}}$, is simply obtained by evaluating the source light distribution $\mathcal{I}_{\mathrm{src}}$ on the lens plane. Hence, we have[3]

$$\mathcal{I}_{\mathrm{lens}}(\boldsymbol{\theta}) = \mathcal{I}_{\mathrm{src}}(\boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta})) \,. \tag{8}$$

[3] Note that we do not consider the light distribution of the foreground galaxy since it is in general subtracted in real data analyses.

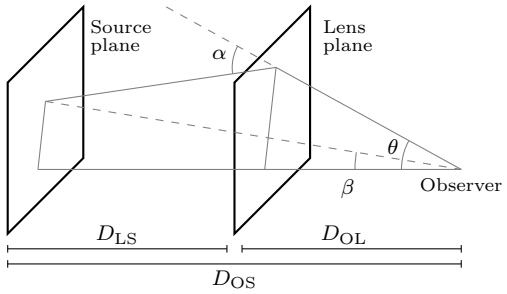

FIG. 4. **Diagram of a gravitational lensing system.** The simplest galaxy-galaxy lensing system is represented by a background and a foreground galaxy, which define the source (S) and the lens plane (L), respectively. The quantities D are the angular diameter distances between the different planes and the observer (O). The two-dimensional angular coordinates, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, are related through the displacement field $\boldsymbol{\alpha}$ by the lens equation (**??**).

This equation is solved on a squared pixel grid defined in the lens plane according to the observed image. Then, the predicted lensed image $\mathcal{I}_{\rm pred}$ is obtained by taking into account the Point Spread Function (PSF), which defines how a point-like source (a pixel) is spread due to atmospheric distortions and defects in the optics, and the noise from instrumental and astrophysical backgrounds. In the present paper, we consider a symmetric two-dimensional Gaussian PSF with a standard deviation of 0.05 arcsec. Moreover, to each pixel we add uncorrelated Gaussian noise with $\sigma_{\rm noise} = 0.333$, $\sigma_{\rm noise} = 0.1$ and $\sigma_{\rm noise} = 0.0333$ for mock images with low, medium and high signal-to-noise (S/N) ratio, respectively. Hence, we have

$$\mathcal{I}_{\rm pred} = \mathcal{N}\left(\mathrm{PSF} * \mathcal{I}_{\rm lens}, \sigma_{\rm noise}\right) , \tag{9}$$

where the symbol $*$ stands for the mathematical convolution between the Point Spread Function and the image in the lens plane.

The two main ingredients describing a gravitational lensing system are, therefore, the total mass distribution of the lens Σ (the lens model) and the surface brightness profile of the background source $\mathcal{I}_{\rm src}$ (the source model).

B. Lens model

In a typical lensing system, the dominant contributions to the total displacement field $\boldsymbol{\alpha}$ come from the smooth main halo (mh) of the foreground galaxy and the external shear (ext), namely

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\rm mh} + \boldsymbol{\alpha}_{\rm ext}. \tag{10}$$

The first contribution results from the sum of a spherically symmetric dark matter halo, described by a universal power-law dependence on the radial distance from the centre of the galaxy, the Navarro-Frenk-White profile [**?**], and a baryonic component in the bulge of the galaxy. The sum of these two mass components results in a isothermal power-law mass distribution with slope $\gamma = 2$. Such a behaviour, defined as *bulge-halo conspiracy* [**?**], has been confirmed for early-type galaxy by lensing studies and by local dynamics [**?** **?** **?**]. However, to model the variety of different galaxies, the surface mass distribution of the main halo is typically described by the so-called Singular Power-Law Ellipsoid (SPLE) profile where the slope is a free parameter. In this case, we have [**?** **?**]

$$\Sigma_{\rm mh} = \Sigma_{\rm cr} \frac{3-\gamma}{2} \left[\frac{\rho\left(\boldsymbol{\theta}', q\right)}{r_{\rm Ein}}\right]^{1-\gamma} , \tag{11}$$

where $\Sigma_{\rm cr} = c^2 D_{\rm OS}/(4\,\pi\,G\,D_{\rm OL}\,D_{\rm LS})$ is the critical surface mass density, γ is the slope, $r_{\rm Ein}$ denotes the Einstein radius, and ρ encodes the dependence on the position. In the special case $\gamma = 2$ this distribution reduces to the one for a singular isothermal ellipsoid [**?**]. In the coordinates system $\boldsymbol{\theta}'(\boldsymbol{\theta} - \boldsymbol{\theta}_{\rm lens}, \phi)$ with origin in the centroid of the foreground galaxy (denoted as $\boldsymbol{\theta}_{\rm lens}$) and axes aligned to the minor and major axes of the elliptical galaxy (after a rotation of an angle ϕ), we have

$$\rho(\boldsymbol{\theta}', q) = \theta_1'^2 + \theta_2'^2/q^2 , \tag{12}$$

with q being the minor to major axis ratio of the elliptical contours of equal surface mass density. Hence, the displacement field $\boldsymbol{\alpha}_{\rm mh}$ induced by the main halo mass distribution is simply given by plugging eq. (**??**) into eq. (**??**).

Since numerical integration is difficult to implement in an automatically-differentiable manner, we instead compute $\boldsymbol{\alpha}_{\rm mh}$ by interpolating over a precomputed grid, as described in detail in Appendix **??**. This enables computation of gradients of $\boldsymbol{\alpha}_{\rm mh}$ since the interpolation function is itself automatically differentiable.

The external shear contribution represents the additional angular structure provided by additional matter distribution in the cluster where the galaxy is located. The corresponding displacement field is parametrized as

$$\boldsymbol{\alpha}_{\rm ext} = \begin{pmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & -\gamma_1 \end{pmatrix} \boldsymbol{\theta}. \tag{13}$$

Our lens model is thus completely defined by a set of 8 parameters: $\boldsymbol{\Theta}_{\rm lens} \equiv \{\gamma_1, \gamma_2, \phi, q, \gamma, r_{\rm Ein}, \theta_{\rm lens,1}, \theta_{\rm lens,2}\}$.

C. Source model

The VAE's decoder $d_\theta(\mathbf{z})$ provides a parametrized model for 64×64-pixel galaxy images, which is the basis for our source model. One of its parameters is the 64-dimensional vector $\mathbf{z}$ specifying a point in the VAE's latent space. As described earlier, the prior for $\mathbf{z}$ is defined by fitting a multivariate normal distribution to the means of the encoded distributions of the training points and increasing its covariance by a factor of 9. We introduce four other parameters to complete the model: $\theta_{\rm src,1}$,

$\theta_{\rm src,2}$, s and ι. The first two of these are the position of the center of the decoded image $d(\mathbf{z})$. The second specifies the spatial scale of the image and the third is the normalization of the pixel intensities. In this work we adopt the priors

$$\theta_{\rm src,1}, \theta_{\rm src,2} \sim \mathcal{N}(0, 0.1) \quad (14)$$
$$s \sim \mathcal{N}(5, 1) \quad (15)$$
$$\iota \sim \mathcal{N}(1, 0.5) \quad (16)$$

for our mock data generation and analysis. Our overall model for the surface brightness of the source galaxy at a position $\boldsymbol{\theta} = (\theta_1, \theta_2)$ is thus

$$\mathcal{I}_{\rm src}(\boldsymbol{\theta}|\boldsymbol{\Theta}_{\rm src}) = \iota\, d(\mathbf{z}) \left(\frac{\theta_1 - \theta_{\rm src,1}}{s}, \frac{\theta_2 - \theta_{\rm src,2}}{s} \right), \quad (17)$$

where $\boldsymbol{\Theta}_{\rm src} \equiv \{\mathbf{z}, \theta_{\rm src,1}, \theta_{\rm src,2}, s, \iota\}$. We use bilinear interpolation to allow the right-hand side of this expression to be evaluated between adjacent pixels in the 64×64 output image from the decoder.

IV. THE LENSING PIPELINE

Our lensing pipeline is unique since it combines the VAE-based source and physical lens models detailed in the previous section in a fully-differentiable manner. As sketched in Fig. **??**, the pipeline consists of a *forward* flow (black solid lines) and a *backward* one (red dashed lines).

In the forward flow, the predicted lensed image $\mathcal{I}_{\rm pred}(\boldsymbol{\Theta}_{\rm lens}, \boldsymbol{\Theta}_{\rm src})$ is obtained once the displacement field $\boldsymbol{\alpha}(\boldsymbol{\theta}|\boldsymbol{\Theta}_{\rm lens})$ and the source surface brightness $\mathcal{I}_{\rm src}(\boldsymbol{\theta}|\boldsymbol{\Theta}_{\rm src})$ have been computed in the Lens Model (see sec. **??**) and the Source Model (see sec. **??**), respectively. This image is then compared with the observed one to estimate the likelihood function given the parameters of the lens and the source models. Thanks to the differentiable programming framework, it is then possible to compute the derivatives of the likelihood function with respect to all the parameters, $\boldsymbol{\Theta}_{\rm lens}$ and $\boldsymbol{\Theta}_{\rm src}$. This step represents the backward flow of the whole pipeline.

The pipeline is implemented in a differentiable probabilistic programming framework comprised of the PyTorch [**?**] machine learning library and Pyro [**?**] probabilistic programming library. PyTorch provides an automatic differentiation engine, enabling the backward flow of the pipeline. The likelihood calculations in the forward flow are made straightforward by Pyro. The VAE source model is constructed from neural network layers contained in PyTorch and trained using the variational inference module in Pyro. We employ Pyro's variational inference and Hamiltonian Monte Carlo modules for parameter fitting and posterior sampling our mock data analysis in the following section.

We stress that auto-differentiation is automatically guaranteed if all the pipeline is written in the differentiable programming language. Moreover, we note that the lensing pipeline is implemented so that the lens and the source models are independent building blocks. In the present paper, the former is fully based on physical models while the latter is provided by the VAE's decoder. However, thanks to the modularity of the pipeline, both can be easily modified or substituted, as can any of the priors on the lensing parameters. This fundamental feature allows one to generalize the present lensing pipeline to analyze more realistic systems and to include the gravitational effect of dark matter substructures in the lensing physics. This is left for future investigation.

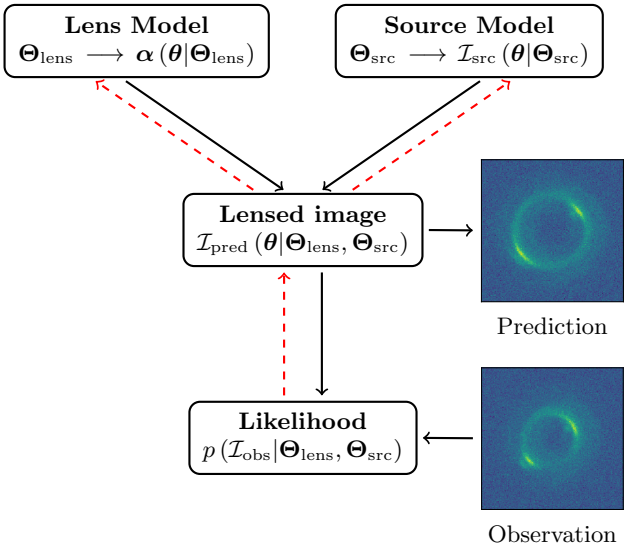

FIG. 5. **Lensing Pipeline.** The forward flow (black solid lines) estimates the likelihood by comparing the observation with the predicted lensed image. This is obtained through the Lens Model and the Source Model. The backward flow (red dashed lines) computes the derivatives of the likelihood with respect to all the parameters of the models, $\boldsymbol{\Theta}_{\rm lens}$ and $\boldsymbol{\Theta}_{\rm src}$. Each box represents an independent module.

V. RESULTS

In this section, we test the lensing pipeline on mock lensing system observations. We describe the generation of mock data, and discuss the results obtained by the parameter fitting and the posterior analysis for two different mock lensing systems.

A. Mock data

To test our pipeline, we generate mock lensing system observations. We create mock sources by first denoising images from the test dataset using the non-local means algorithm [**?** **?**]. These images are then rescaled to fill roughly the central third of the source image plane, which ensures the lensed image pixel intensities drop to

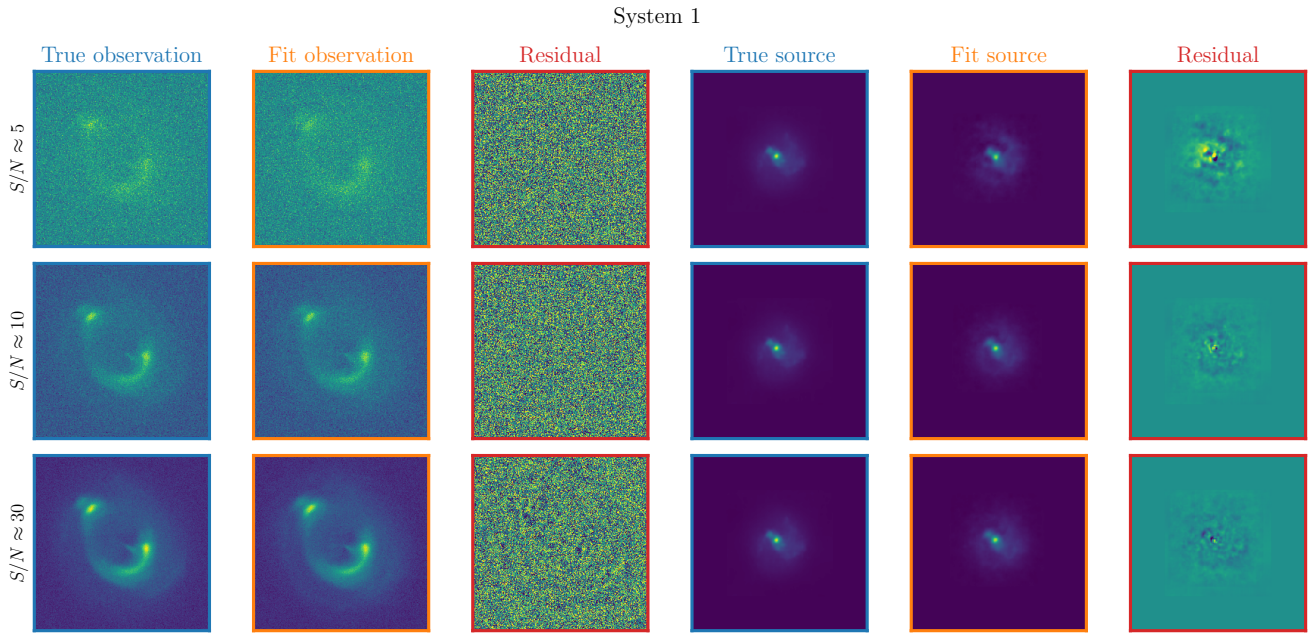

FIG. 6. **Results of MAP fit of mock lensing system 1.** The different columns show the true and fit observations, the residual between these, the true and fit sources and the residuals between those. The rows correspond to different observed signal-to-noise values. Within each row, the color scales for the true and fit observations are the same, as are the scales for the true and fit sources. The observation residuals are normalized by dividing by the standard deviation of the observation noise. The source residuals are normalized by dividing by 10% of the maximum value of the source. The color scale ranges from dark blue to bright yellow.

zero along the image boundaries. The mock lensing parameters are determined randomly by drawing from the following distributions:

$$\gamma_1, \gamma_2 \sim \mathcal{N}(0, 1) \tag{18}$$
$$\phi \sim \mathcal{N}(0, 1 \text{ rad}) \tag{19}$$
$$q \sim \mathcal{N}(0.5, 0.5) \tag{20}$$
$$r_{\text{Ein}} \sim \mathcal{N}(1.5 \text{ arcsec}, 0.5 \text{ arcsec}) \tag{21}$$
$$\gamma \sim \mathcal{N}(2, 0.5) \tag{22}$$
$$\theta_{\text{lens},1}, \theta_{\text{lens},2} \sim \mathcal{N}(0, 0.5). \tag{23}$$

The mock lensed images are then produced using the formalism from the previous section on a 256×256 pixel grid with angular size 10 arcsec $\times$ 10 arcsec.[4] Finally, we convolve with the PSF and add Gaussian pixel noise. To test how data quality impacts our pipeline's performance, we fix the pixel noise level σ_{noise} to give S/N ratios of approximately 5, 10 or 30, where this quantity is defined as $\max(\mathcal{I})/\sigma_{\text{noise}}$ for an image $\mathcal{I}$.

[4] This corresponds to an angular resolution of ~ 0.04 arcsec. The distributions above ensure the lensed images are generally well-contained in this grid. For reference, the angular resolutions of Hubble [**?**] and ELT [**?**] are approximately 0.1 arcsec and 0.005 arcsec, respectively.

In the rest of this paper, we focus on two particular mock systems, hereafter referred to as system 1 and system 2. Source 1 has a fairly simple spiral morphology. Source 2 has a significant amount of substructure, making it representative of complex, high-redshift source galaxies.

B. Parameter fitting

We first test our pipeline by finding the best-fit source and lens parameter values. For a given image $\mathcal{I}_{\text{obs}}$ we compute the maximum a posteriori parameter (MAP) estimates:

$$\hat{\boldsymbol{\Theta}}_{\text{lens}}, \hat{\boldsymbol{\Theta}}_{\text{src}} = \max_{\boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}}} p(\boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}} | \mathcal{I}_{\text{obs}}), \tag{24}$$

$$p(\boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}} | \mathcal{I}_{\text{obs}}) \propto p(\mathcal{I}_{\text{obs}} | \boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}}) p(\boldsymbol{\Theta}_{\text{lens}}) p(\boldsymbol{\Theta}_{\text{src}}).$$

The first term on the right-hand side of the posterior is the likelihood of the observed image for fixed source and lens parameters, which is Gaussian due to our noise assumption:

$$p(\mathcal{I}_{\text{obs}} | \boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}}) = \mathcal{N}(\mathcal{I}_{\text{obs}} | \mathcal{I}_{\text{pred}}(\boldsymbol{\Theta}_{\text{lens}}, \boldsymbol{\Theta}_{\text{src}}), \sigma_{\text{obs}}).$$

The second term is the prior on the lens parameters, for which we adopt eqs. (**??**)-(**??**). The source priors are

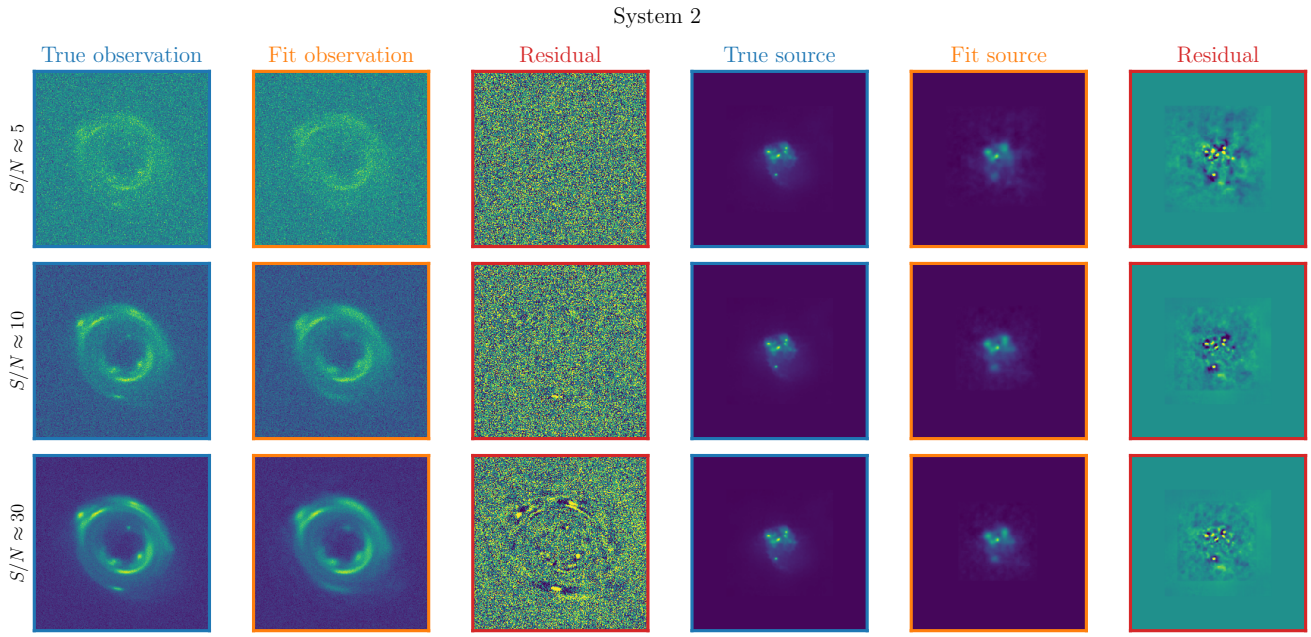

FIG. 7. **Results of MAP fit of mock lensing system 2.** The subplots and scales are explained in the caption of Fig. **??**.

specified in eq. (**??**)-(**??**), and the prior on $\mathbf{z}$ is described in sec. **??**.

Since it is possible to differentiate through the full lensing pipeline, we obtain the best-fit parameters using the gradient-based Adam optimizer [**?**]. Adam has been empirically shown to outperform other gradient-based methods in nonconvex optimization problems with large numbers of parameters. These fits converge after 10^4 iterations, which takes approximately 20 minutes on CPU.

The MAP reconstructions of the mock lensing systems 1 and 2 are shown in Figs. **??** and **??**. We report the true and fit images, and the residuals between these, in the lens and in the source planes, for three values of signal-to-noise ratio. In case of system 1, the residuals in the lens plane are at the noise level even for the smallest pixel noise level considered, i.e. $S/N \approx 30$. This is related to the fact that the reconstruction of the source improves as the signal-to-noise ratio increases, as can clearly seen in the last column in Fig. **??**. This does not occur for the complex source galaxy of system 2. Indeed, as shown in Fig **??**, even for the highest S/N value, the VAE's decoder is not able to reproduce all the substructures exhibited in the source surface brightness. This directly affects the image reconstruction in the lens plane, and the corresponding residuals (third column in Fig. **??**) increase and show substantial structure as the signal-to-noise ratio increases.

C. Posterior analysis

To study our pipeline's parameter inference capabilities, we ran Hamiltonian Monte Carlo [**?** **?** **?**] to sample from the parameters' posteriors. Hamiltonian Monte Carlo (HMC) is a Markov-chain Monte Carlo (MCMC) procedure that uses Hamiltonian dynamics based on the gradient of the posterior to efficiently traverse parameter space. HMC can take larger steps than other MCMC procedures such as Metropolis-Hastings while keeping the acceptance probabilities high, leading to more efficient exploration of parameter space. After 50 steps during which the internal HMC parameters are calibrated, we use it to sample 500 times from the posterior starting from the MAP parameter estimates. This takes about 10 hours on a CPU.

The HMC results for the two systems are reported in Figs. **??** and **??**, which show the marginalized posteriors of the lens parameters $(r_{\rm Ein}, \gamma, \phi)$. As expected, for both systems the posteriors shrink as the S/N ratio increases. In particular, the statistical error on the parameter estimates moves from $\sim 3\%$ to $\lesssim 1\%$ for the lowest and highest S/N ratio, respectively. However, especially for the system 1, we note that our lensing pipeline gives biased estimates typically at the level of 1%. We hypothesize this is because the VAE generally produces slightly blurred and hazy reconstructions (most clearly visible in the last row of residual plots in Fig. **??**). Improving the fidelity of reconstructions and samples from VAEs is an active area of research in machine learning [**?** **?** **?**]. In addition to improving the VAE architecture and training procedure, our source model would benefit from a larger,

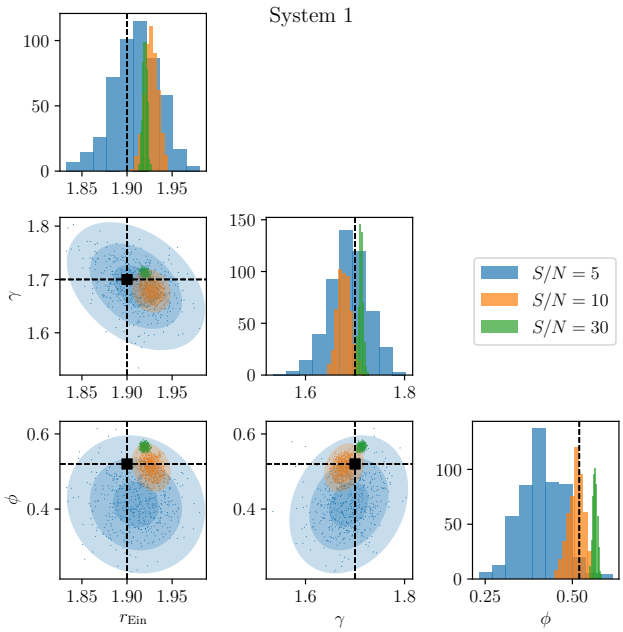

FIG. 8. **Results of HMC parameter posteriors of mock lensing system 1.** The different panels show the marginalized one-dimensional and two-dimensional posteriors for a subset of lens parameter, $(r_{\rm Ein}, \gamma, \phi)$. The different colors (blue, orange, green) refer to the different signal-to-noise ratios ($S/N \approx 5, 10, 30$), while the color shading in the ellipses corresponds to 68%, 95% and 99% confidence levels. The sampled points are also plotted. The black squares and dashed lines represent the true values of parameters.

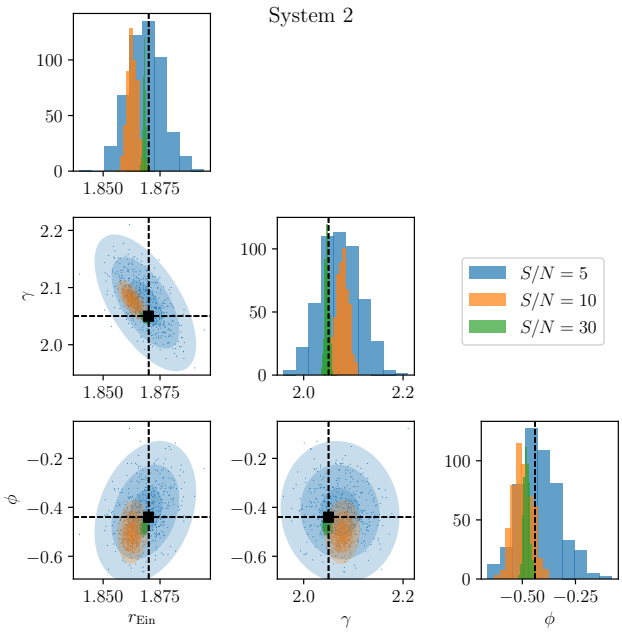

FIG. 9. **Results of HMC parameter posteriors of mock lensing system 2.** The panels are explained in the caption of Fig. **??**. Note the substantial overlap between the green ellipses and black squares.

higher-resolution training dataset, as will be made available by future astronomical surveys. Surprisingly, the level of bias is smaller for the system 2, even though the source-plane residuals are larger due to the presence of fine substructure.

VI. CONCLUSIONS

We have presented the first step towards a new inference pipeline to analyze present and future strong gravitational lensing systems. The main novelty of our approach is the use of the differentiable probabilistic programming framework in which all operations are automatically differentiable with respect to the input parameters. This powerful approach makes Bayesian inference feasible for complex models with hundreds (or thousands) of free parameters thanks to efficient sampling techniques utilizing gradient descent.

Our lensing pipeline, shown in Fig. **??**, consists of two independent blocks describing the surface brightness of the source galaxy (Source Model) and the mass distribution of the lens galaxy (Lens Model). They are combined to generate the lensed image, which is used to estimate the likelihood and perform the Bayesian inference. The advantages of this strategy are:

- Exact gradients of the pipeline's output can be computed with respect to its inputs using automatic differentiation. This makes it possible to use efficient gradient-based fitting and posterior sampling procedures.
- Using a differentiable probabilistic programming framework allows us to integrate the variational autoencoder source model learned from unlensed galaxy images directly with a physical lensing model. Learning the source model directly from data obviates for tuning hyperparameters regularizing the source model.
- We fully automatize the inference step using probabilistic programming. The lens and source parameters are fit and sampled simultaneously with arbitrary priors.
- By implementing our pipeline in the PyTorch framework, we automatically gain the ability to perform computations on graphical processing units.

From a quantitative perspective, the best-fit lens parameter values we obtained in our mock data tests were within $\sim 1\%$ of the true values, albeit with some bias at very high signal-to-noise ratios.

Our lensing model is well-suited to extensions improving its realism. Automatic differentiation makes the computational cost of adding parameters to our pipeline very

cheap. This will help with increasing the complexity of the lens model by including effects such as subhalos. For these higher-dimensional models, automated techniques such as variational inference [**?** **?**] have favorable scaling behavior, making it possible to analyze posterior distributions for hundreds or thousands of parameters. While our pipeline is an early example of differentiable probabilistic programming, we anticipate this approach will enable other challenging and exciting data analyses in the future by leveraging the advantages of deep learning and physics modeling.

ACKNOWLEDGMENTS

We would like to thank Simona Vegetti and Rajat Thomas for helpful conversations. This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. We acknowledge funding from the Netherlands Organization for Scientific Research (NWO) through the VIDI research program "Probing the Genesis of Dark Matter" (680-47-532).

Appendix A: The main halo model

In this appendix, we describe how the contribution to the displacement field due to the main halo, the Singular Power-Law Ellipsoid (SPLE) profile, is coded up in the Lens Model block of the lensing pipeline shown in Fig. **??**. It is worth observing that, in the absence of an analytical expression for $\boldsymbol{\alpha}_{\rm mh}$, one has to numerically compute the integral in eq. **??**. However, this is not feasible in our framework because the numerical integration is not coded up in a auto-differentiable way. For this reason, the displacement field is instead determined by means of an interpolation of a pre-computed numerical table of the corresponding integral, eq. **??**.

In case of a surface mass density profile with elliptical contours (like for example the SPLE profile), the two-dimensional integral of eq. **??** can be reduce to a simpler one-dimensional integral [**?** **?**]

$$\alpha_1\left(\theta_1,\theta_2\right) = 2\theta_1 q \int_0^{\rho(\theta_1,\theta_2)} \frac{\rho'\,\kappa(\rho')\,\omega}{\theta_1^2+\omega^4\,\theta_2^2} d\rho' \,, \tag{A1}$$

$$\alpha_2\left(\theta_1,\theta_2\right) = 2\theta_2 q \int_0^{\rho(\theta_1,\theta_2)} \frac{\rho'\,\kappa(\rho')\,\omega^3}{\theta_1^2+\omega^4\,\theta_2^2} d\rho' \,, \tag{A2}$$

with

$$\omega^2 = \frac{\Delta + r^2 + \rho'^2(1-q^2)}{\Delta + r^2 - \rho'^2(1-q^2)} \,, \tag{A3}$$

$$\Delta^2 = \left[\rho'^2(1-q^2) + \theta_2^2 - \theta_1^2\right]^2 + 4\theta_1^2\theta_2^2 \,. \tag{A4}$$

where $r^2 = \theta_1^2 + \theta_2^2$, $\rho\left(\theta_1,\theta_2\right)$ is defined in eq. **??**, and the quantity κ is the surface mass density Σ in units of the critical density $\Sigma_{\rm cr}$. For a given profile, the integrals in eqs. **??** and **??** can be tabulated for different values of a subset of the lens parameters $\boldsymbol{\Theta}_{\rm lens}$. In particular, in case of the SPLE lens, the displacement field has been evaluated on a unit circle ($r = 1$), for which the coordinates are $\theta_1 = \cos\eta$ and $\theta_1 = \sin\eta$ with the angle η being in the first quadrant ($0 \leq \eta \leq \pi/2$), for different values of the axis ratio q in the interval $[0, 1]$ and the slope γ. The Einstein radius is instead fixed to be $r_{\rm Ein} = 1$. Such a procedure provided a three-dimensional table in the variables $\{\eta,\, q,\, \gamma\}$. This numerical table is then interpolated to compute the SPLE displacement field for any values of q and γ, and at any position (θ_1, θ_2). Each component of the displacement field is indeed given by

$$\alpha_i\left(\theta_1,\theta_2\right) = \alpha_i\left(|\theta_1|,|\theta_2|\right)\,\mathrm{sign}\left(\theta_i\right) \,, \tag{A5}$$

$$\alpha_i\left(\theta_1,\theta_2\right)\big|_r = r^{2-\gamma}\;\alpha_i\left(\theta_1,\theta_2\right)\big|_{r=1} \,. \tag{A6}$$

Moreover, in case of the SPLE profile, these two components show a simple scaling relation as a function of the Einstein radius. We have

$$\alpha_i\left(\theta_1,\theta_2\right)\big|_{r_{\rm Ein}} = \left(\frac{1}{r_{\rm Ein}}\right)^{1-\gamma} \alpha_i\left(\theta_1,\theta_2\right)\big|_{r_{\rm Ein}=1} \tag{A7}$$

It is worth noticing that the accuracy in computing the displacement field by means of this procedure depends on the size of the interpolation table, which can be defined without any constraint.

Appendix B: ELBO derivation

This appendix demonstrates one possible derivation of eq. **??**. Consider a latent variable model defined by the joint probability distribution $p(x, z)$, where x and z are the observed and latent variables, respectively. We start by rewritting an expression for the log of the evidence:

$$\log p(x) = \log \int_z p(x,z) dz \tag{B1}$$

$$= \log \int_z p(z|x)\,\frac{p(x,z)}{p(z|x)} dz \,, \tag{B2}$$

This can be recognized as an expectation value over $p(z|x)$:

$$\log p(x) = \log \mathbb{E}_{p(z|x)}\left[\frac{p(x,z)}{p(z|x)}\right] . \tag{B3}$$

By Jensen's inequality, which relates the expectation value of a convex function to that function applied to an expectation value, we have

$$\log \mathbb{E}_{p(z|x)}\left[\frac{p(x,z)}{p(z|x)}\right] \geq \mathbb{E}_{p(z|x)}\left[\log\left(\frac{p(x,z)}{p(z|x)}\right)\right] , \tag{B4}$$

and thus:

$$\log p(x) \geq \mathbb{E}_{p(z|x)}\left[\log\left(\frac{p(x,z)}{p(z|x)}\right)\right] . \tag{B5}$$

Using the definition of the Kullback-Leibler divergence for continuous random variables

$$D_{KL}[p||q] = -\mathbb{E}_{p(x)}\left[\log\frac{q(x)}{p(x)}\right], \tag{B6}$$

this can be manipulated to yield

$$\begin{aligned}\log p(x) &\geq \mathbb{E}_{p(z|x)}\left[\log\frac{p(x|z)p(z)}{p(z|x)}\right] && \text{(B7)}\\ &= \mathbb{E}_{p(z|x)}\left[\log p(x|z) + \log\frac{p(z)}{p(z|x)}\right] && \text{(B8)}\\ &= \mathbb{E}_{p(z|x)}\left[\log p(x|z)\right] - D_{KL}\left[p(z|x)||p(z)\right] && \text{(B9)}\\ &\equiv \text{ELBO}(\theta,\phi;x). && \text{(B10)}\end{aligned}$$

The VAE training objective is obtained by substituting the approximate distributions $p(x|z) \to d_\theta(x|z)$ and $p(z|x) \to e_\phi(z|x)$ for the true ones.

Appendix C: Optimizing the ELBO

Training the variational autoencoder requires taking the gradient of the ELBO for (batches of) training images $\{x^{(i)}\}_{i=1}^N$ with respect to the encoder and decoder's parameters θ and ϕ:

$$\begin{aligned}&\nabla_{\theta,\phi}\,\text{ELBO}\left(\theta,\phi;x^{(i)}\right)\\ &\quad= \nabla_{\theta,\phi}\,\mathbb{E}_{e_\phi\left(z|x^{(i)}\right)}\left[\log d_\theta\left(x^{(i)}|z\right)\right]\\ &\qquad- \nabla_{\theta,\phi} D_{KL}\left[e_\phi\left(z|x^{(i)}\right)||\,p(z)\right].\end{aligned} \tag{C1}$$

For the normal encoding distribution and latent space prior adopted in this work, the KL divergence term can be integrated analytically, which makes it simple to compute the second term on above. The first term is more challenging: while a Monte Carlo estimate of the derivative with respect to θ can be performed by sampling $\left\{z^{(j)} \sim e_\phi\left(z|x^{(i)}\right)\right\}_{j=1}^M$, it is not obvious how to compute the derivatives of the sampled latent variable values with respect to ϕ.

The solution is the reparameterization trick introduced in the two original papers on variational autoencoders [**?** **?**]. The insight is that (assuming a normal encoding distribution) the randomness and ϕ-dependent parts of the sampling process can be factored, allowing the sampled values to be written as

$$z^{(j)} = \mu_e\left(x^{(i)};\phi\right) + \epsilon^{(j)}\,\sigma_e\left(x^{(i)};\phi\right), \tag{C2}$$

with $\epsilon^{(j)} \sim \mathcal{N}(0, I)$. These can be used to construct the following Monte Carlo gradient estimator by treating the $\epsilon^{(j)}$ values as constants for the optimization epoch:

$$\begin{aligned}&\nabla_{\theta,\phi}\,\mathbb{E}_{e_\phi\left(z|x^{(i)}\right)}\left[\log d_\theta\left(x^{(i)}|z\right)\right]\\ &\qquad\approx \frac{1}{M}\sum_{j=1}^{M}\log d_\theta\left(x^{(i)}|z^{(j)}\right).\end{aligned} \tag{C3}$$

This estimator is stable; we set $M = 10$ in our work by using batches of 32 training images.

Appendix D: Variational Autoencoder Architecture

The architectural details of the encoder and decoder networks of our variational autoencoder are presented in Tables **??** and **??**, respectively. All weights were initialized to 0.02. The biases in the final linear layers of the encoder were initialized to 0.

We tried several experiments to see whether we could improve upon this VAE design. For example, since the decoder can have trouble saturating the final tanh activation, we tried exchanging this for a LeakyReLU, as well as removing it altogether. We also tested 32 and 128 latent-space dimensions. The former lead to blurry reconstructions while the later yielded little improvement relative to 64 latent-space dimension. Ref. [**?**] analytically demonstrated that making the hyperparameter σ_d a trainable parameter should lead to sharper reconstructions. We did not find this to be the case, and instead found that σ_d converged to roughly the value we selected by hand based on the signal-to-noise ratio of the training data. Finally, we experimented with a residual network-based architecture, as was used in Ref. [**?**], which did not yield any improvement relative to the architecture we eventually selected.

Conv2d(1, 64, 4, 2, 1)	
LeakyReLU(0.2)	
Conv2d(64, 128, 4, 2, 1)	
BatchNorm2d(128)	
LeakyReLU(0.2)	
Conv2d(128, 256, 4, 2, 1)	
BatchNorm2d(256)	
LeakyReLU(0.2)	
Conv2d(256, 512, 4, 2, 1)	
BatchNorm2d(512)	
LeakyReLU(0.2)	
Conv2d(512, 4096, 4, 1, 0)	
LeakyReLU(0.2)	
Linear(4096, 64)	Linear(4096, 64)
	Exp
$\mu_e(x)$	$\sigma_e(x)$

TABLE I. **Encoder neural network architecture.** The notation uses the same conventions as pytorch. The arguments of Conv2d indicate the number of input channels, number of output channels, kernel size, stride and zero padding; all convolutions are unbiased. The LeakyReLU argument is slope for inputs less than 0. The BatchNorm2d argument is the number of input channels. The output of the last convolutional block is flattened before being passed to the two separate Linear layers to produce the mean and standard deviation of the encoding distribution. Linear layers' arguments show the number of input and output channels.

ConvTranspose2d(64, 512, 4, 1, 0)
BatchNorm2d(512)
ReLU
ConvTranspose2d(512, 256, 4, 2, 1)
BatchNorm2d(256)
ReLU
ConvTranspose2d(256, 128, 4, 2, 1)
BatchNorm2d(128)
ReLU
ConvTranspose2d(128, 64, 4, 2, 1)
BatchNorm2d(64)
ReLU
ConvTranspose2d(64, 1, 4, 2, 1)
Tanh
$\mu_d(z)$

TABLE II. **Decoder neural network architecture.** The notation is described in the caption of Table **??**, and is the same for Conv2d and ConvTranspose2d. The input vector **z** is reshaped to have 64 channels and spatial dimensions equal to 1 along both axes.