Draft version March 28, 2022
Typeset using LaTeX **twocolumn** style in AASTeX631

Free-Form Reconstruction of Gravitational Lenses using Recurrent Inference Machine

Alexandre Adam,[1,2] Laurence Perreault-Levasseur,[1,2,3] and Yashar Hezaveh[1,3]

[1] *Department of Physics, Université de Montréal, Montréal, Canada*
[2] *Mila - Quebec Artificial Intelligence Institute, Montréal, Canada*
[3] *Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, 10010, New York, NY, USA*

ABSTRACT

Modeling strong gravitational lenses in order to quantify the distortions of the background sources and reconstruct the mass density in the foreground lens has traditionally been a major computational challenge. As the quality of gravitational lens images increases, the task of fully exploiting the information they contain becomes computationally and algorithmically more difficult. In this work, we use a neural network based on the Recurrent Inference Machine (RIM) to simultaneously reconstruct an undistorted image of the background source and the lens mass density distribution as pixelated maps. The method we present iteratively reconstructs the model parameters (the source and density map pixels) by learning the process of optimization of their likelihood given the data using the physical model (a ray tracing simulation), regularized by a prior implicitly learnt by the neural network through its training data. When compared to more traditional parametric models, the method we propose is significantly more expressive and can reconstruct complex mass distribution, which we demonstrate by using realistic lensing galaxies taken from the hydrodynamical IllustrisTNG simulation.

Keywords: Gravitational lensing (670) — Astronomical simulations (1857) — Nonparametric inference (1903) — Convolutional Neural Networks (1938)

1. INTRODUCTION

A gravitational lens is composed of massive objects — or *deflectors*— in the line of sight that magnify and distort luminous background objects like early-type star-forming galaxies (Vieira et al. 2013; Marrone et al. 2018; Rizzo et al. 2020; Sun et al. 2021), otherwise too faint to study with our current ground and orbital telescope facilities. This distortion is a very good tracer of mass, independent of the electromagnetic signature of the foreground deflector. As such, it is one of the rare ways to study the properties of dark matter halos via its spatial distribution at the very small scales (Dalal & Kochanek 2002; Treu & Koopmans 2004; Hezaveh et al. 2016; Gilman et al. 2020, 2021). Gravitational lenses also act as cosmological rulers against which we can measure the expansion rate of the universe by monitoring the flickering light of multiply imaged quasars (Treu & Marshall 2016; Millon et al. 2020, and reference therein) or the dimming of multiply imaged supernovae (Refsdal 1964; Treu et al. 2016; Grillo et al. 2018). A central component of such cosmographic analysis is the careful mass modelling of the lensing galaxy (Chen et al. 2019; Wong et al. 2020) or lensing galaxy clusters (Kneib & Natarajan 2011; Hoekstra et al. 2013; Natarajan et al. 2017; Bergamini et al. 2021; Jauzac et al. 2021).

A common practice in galaxy mass modelling is to assume that the mass distribution of the main deflector follows a power law $\rho \propto r^{-\gamma'}$ (Keeton 2001). Following spectroscopic measurement of the velocity dispersion of early-type galaxies, a singular isothermal profile $\gamma' = 2$ can provide a good starting point for analysis (Koopmans et al. 2006; Barnabè et al. 2009; Auger et al. 2010). For cosmographic measurments, this assumption is relaxed by leaving the slope of the profile as a free parameter in the mass modelling stage since the isothermal approximation will induce a bias in the measurement of the Hubble constant (Treu & Koopmans 2004; Birrer et al. 2020). Composite models can also be constructed as in Millon et al. (2020), who uses a Navarro-Frenk-White profile (Navarro et al. 1997) to model the dark matter halo that host the lensing galaxy and a Sérsic profile (Sérsic 1963) to model the baryonic component of the galaxy. Even though these models could produce a large range of profiles, best fit models often have an average slope akin to an isothermal profile. This observation is dubbed the *bulge-halo conspiracy* (Dutton & Treu 2014).

Detailed modelling of high resolution images with high signal-to-noise ratio (SNR) will additionally require external perturbations to the main lensing galaxy coming from its local environment (Sluse et al. 2017; Wong et al. 2017; Birrer et al. 2019; Rusu et al. 2020) and from the line of sight (Rusu et al. 2017; Li et al. 2021) in order to fully capture the signal. But, this approach becomes unwieldy as the quality of images increases. More and more perturbations need to be added in order to account for fine details in the data that are only revealed in the high SNR regime. Famously, the Hubble Space Telescope (HST) Wide Field Camera 3 (WFC3) images of the Cosmic Horseshoe (J1148+1930) — initially discovered by Belokurov et al. (2007) — has many fine features that are hard to reproduce (e.g. Bellagamba et al. 2016; Cheng et al. 2019; Schuldt et al. 2019).

Free-form methods — also misleadingly called non-parametric — attempt to relax the assumptions about the smoothness and symmetries of the mass distribution by changing its parametric support. This includes regular (or adaptive) grid representations and meshfree representations (Saha & Williams 1997; Abdelsalam et al. 1998a,b; Diego et al. 2005; Birrer et al. 2015; Merten 2016). These methods are more expressive and make better use of the information contained in the arcs of the observed image in order to constrain the mass distribution of the lens. However, the widened range of distributions that can be represented include non-physical solutions that must be penalized via regularization.

An important example is semi-linear inversion, originally introduced by Warren & Dye (2003) for grid-based source brightness reconstruction. It was later improved to include linear perturbation to the lensing potential (Koopmans 2005). The regularization procedure was also formalized into a Bayesian framework (Suyu et al. 2006; Suyu & Blandford 2006; Vegetti & Koopmans 2009), such that it enabled detection of subhalos from distorted Einstein rings or giant arcs (Vegetti et al. 2010, 2012) and constrain the subhalo mass function (Vegetti et al. 2014; Li et al. 2016). This approach is generally limited to small perturbations from the analytical profile used in a preliminary step to approximate the solution. As such, it is difficult to extend and automate this framework to reconstruct lenses from hydrodynamical simulations.

Morningstar et al. (2019) observed that the regularization employed in semi-linear inversion only constrain a two-point prior, often resulting in noise leakage in the source brightness distribution due to the lack of knowledge on high order statistics. They showed that using a Recurrent Inference Machine (RIM, Putzky & Welling 2017), the neural network would implicitly incorporate a more complex prior from a dataset of galaxy images which resulted in better performance overall for the background source reconstruction.

In this paper, we provide a proof of concept for a free-form strong gravitational lensing reconstruction model that goes beyond analytical solutions to reconstruct complex mass distributions from hydrodynamical simulations of galaxies. Building on the work of Morningstar et al. (2019), we address the long standing problem of crafting a prior over a free-form mass distribution by casting it as a meta-learning problem in the context of a RIM. The problem is shifted from having to craft a functional form for the prior distribution to building a representative training set of gravitational lenses. We do this using HST images of galaxies from the COSMOS survey (Koekemoer et al. 2007; Scoville et al. 2007) for the background sources and projected mass density maps from the hydrodynamical simulation IllustrisTNG (Nelson et al. 2019) for the foreground lenses.

The paper is organised as follows. Section 2 details the inference pipeline. In Section 3, we details the data creation and augmentation for training. In Section 4, we report the training strategy of the different models used in the paper, as well as their parameters. In Section 5, we report our results on a held-out test set of gravitational lenses. Section 6 situates our finding within the larger context of studying gravitational lensing.

2. METHODS

In this section, we details the steps to build a free-form inference pipeline with a RIM, beginning with a general introduction about MAP inference with a Gaussian likelihood in Section 2.1. In Section 2.2, we motivate the use of a Recurrent Inference Machine to solve this problem and describe the computational graph of the RIM as well as the optimisation problem of learning the gradient model. The architecture of the gradient model is described in Section 2.3. We details the raytracing simulation in Section 2.4. Finally, we describe the fine-tuning procedure and transfer learning technique applied to reach noise level reconstructions in Section 2.5.

The source code, as well as the various scripts and parameters used to produce the model and results is available as open-source software under the package `Censai`[1].

2.1. *Maximum a posteriori*

The task of reconstructing a signal vector $\mathbf{x} \in \mathcal{X}$ given observed data $\mathbf{y} \in \mathcal{Y}$ is formulated as an ill-posed inverse problem with a known forward model F and additive

[1] https://github.com/AlexandreAdam/Censai

noise distribution. We assume a Gaussian distribution with known covariance matrix C, such that

$$\begin{aligned} \mathbf{y} &= F(\mathbf{x}) + \boldsymbol{\eta}; \\ \boldsymbol{\eta} &\sim \mathcal{N}(0, C). \end{aligned} \tag{1}$$

In our case study, F is a many-to-one non-linear mapping between the model space $\mathcal{X}$ and the data space $\mathcal{Y}$. Finding a unique solution for this ill-posed inverse problem requires strong inductive bias to be introduced in the inference procedure in order to favour certain hypotheses over others. This is often framed as regularization, although the choice of parametrization (or model space) is itself an inductive bias when such a choice is available. Maximum a posteriori (MAP) optimization is the Bayesian version of this, where a prior $p(\mathbf{x})$ is introduced as a probability distribution over $\mathcal{X}$ that will reduce the relevant space to explore during inference. The MAP solution is the hypothesis that maximizes the product of the likelihood $p(\mathbf{y} \mid \mathbf{x})$ and the prior:

$$\mathbf{x}_{\text{MAP}} = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \ \log p(\mathbf{y} \mid \mathbf{x}) + \log p(\mathbf{x}). \tag{2}$$

By restricting ourselves to Gaussian noise models, the likelihood can be calculated directly and takes the form

$$\log p(\mathbf{y} \mid \mathbf{x}) \propto -\big(\mathbf{y} - F(\mathbf{x})\big)^T C^{-1} \big(\mathbf{y} - F(\mathbf{x})\big) \tag{3}$$

However, the prior distribution is harder to define. It is problem-dependent and requires expert knowledge of the model domain.

2.2. *Recurrent Inference Machine*

Instead of handcrafting such a distribution, we attempt to build an inference machine with an implicit prior built in a deep architecture (Bengio 2009). We will use the notation G_φ to make reference to this machine, parametrized by a list of weights and biases φ. To learn such a machine, we turn to an optimisation problem that is made feasible through the inductive biases introduced by

($\mathcal{H}_1$) Defining a training dataset $\mathcal{D}^{\text{train}} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$;

($\mathcal{H}_2$) Choosing an architecture for G_φ;

($\mathcal{H}_3$) Choosing a loss function $\mathcal{L} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

The feasibility of the optimisation problem is determined almost entirely by the strength of these inductive biases. This follows from the no-free lunch theorem for machine learning (Wolpert & Macready 1995; Baxter 2011). The reader might also refer to a modern discussion on inductive bias for machine learning by (Goyal & Bengio 2020) for review and insights. In the context of this work, inductive biases can be defined as anything that might influence the trajectory of the optimizer in φ-space during learning.

We describe the dataset in section 3. The model architecture represents a family of function with properties that already encode some knowledge about the model domain. For example, translational equivariance is built in a convolutional neural network (CNN, Lecun & Bengio 1995). This manifests itself in the weight sharing and overall reduced weight connectivity of CNN compared to fully connected network. Practitioners also assume a certain locality to pixel covariance by using small convolution kernels. These induction biases makes CNN efficient and is one of the main reasons for their success in image recognition tasks (Krizhevsky et al. 2012). This was also clearly illustrated by Ulyanov et al. (2017), who showed that a randomly initialized CNN in a U-net architecture (Ronneberger et al. 2015) can learn the structure of natural images without the need for a training dataset. In other words, ($\mathcal{H}_2$) alone can be made strong enough with some neural network architectures in order to accomplish image denoising and some image inpainting tasks.

A RIM (Putzky & Welling 2017) is a form of learned gradient-based inference where the gradient of the likelihood is projected by a gradient model g_φ such that

$$\begin{aligned} \hat{\mathbf{x}}^{(t)} &= \hat{\mathbf{x}}^{(t-1)} + g_\varphi\big(\hat{\mathbf{x}}^{(t-1)}, \mathbf{y}, \boldsymbol{\nabla}_{\mathbf{y}|\hat{\mathbf{x}}^{(t-1)}}\big); \\ & t \in \{1, \ldots, T\}. \end{aligned} \tag{4}$$

We used Putzky & Welling (2017)'s notation to refer to the gradient of the likelihood ($\boldsymbol{\nabla}_{\mathbf{y}|\mathbf{x}} \equiv \boldsymbol{\nabla}_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})$). The parameters of the gradient model are shared across time (t). This choice reduces the computational complexity of optimizing φ. It also assumes that the governing rule of sequence (4) should be independent of time. The architecture of the gradient model is described in section 2.3.

In an earlier work related to the RIM by Andrychowicz et al. (2016), equation (4) was introduced as a form of optimization based meta-learning. The gradient model would only take as input the gradient of an objective function relative to another network's parameters. In the work of Putzky & Welling (2017), it was observed that (4) could be generalized to any kind of inference problems by adding as input the estimated solution $\hat{\mathbf{x}}^{(t)}$ to the gradient model. They interpreted this as giving the model an awareness of its absolute position in $\mathcal{X}$. It is also consistent with the idea of merging the prior in the implicit structure of the model parameters φ. In this work, we observe that inputting the observed datum $\mathbf{y}$

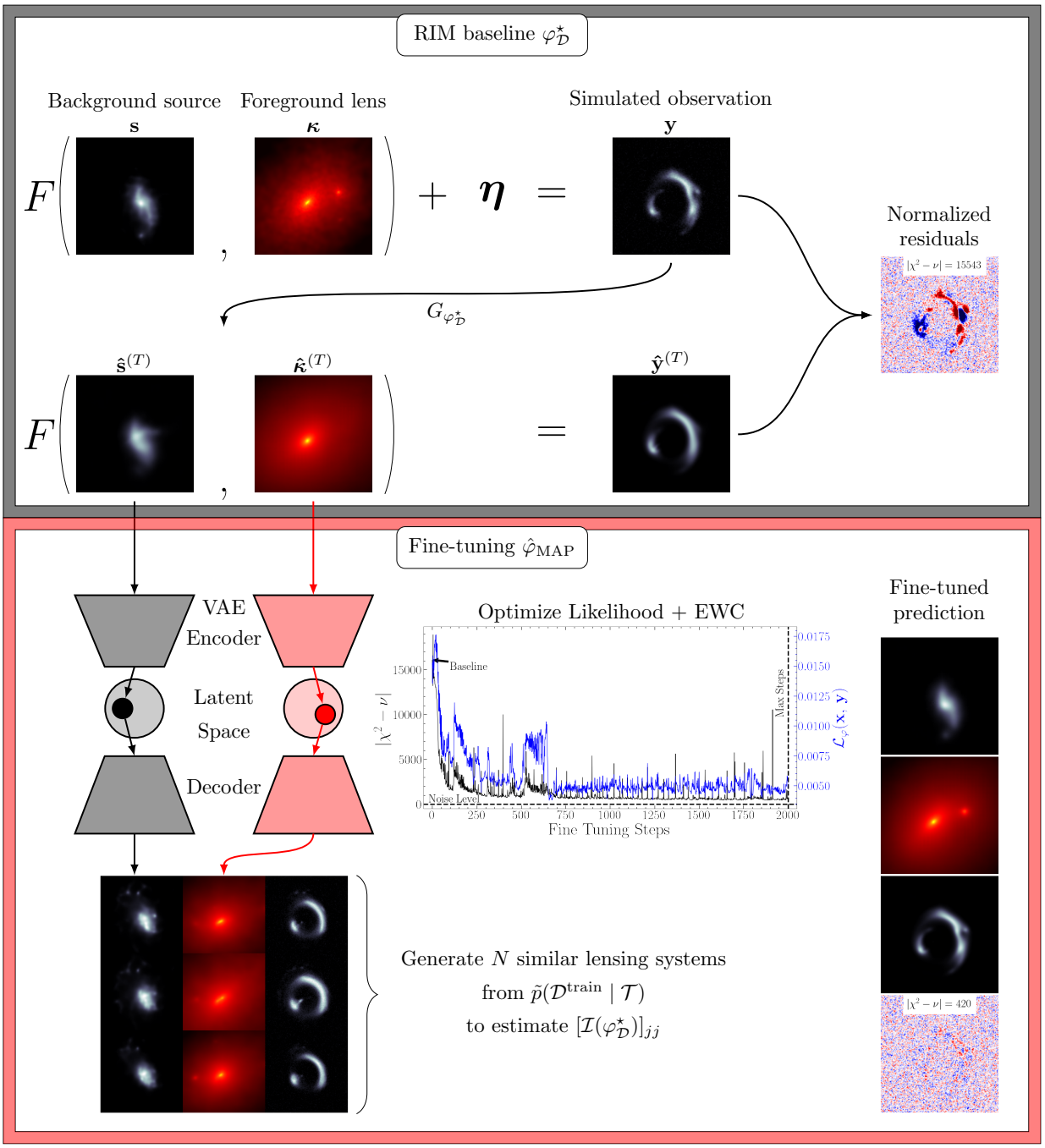

Figure 1. A summary of the main concepts explored in this paper. First, a source and convergence map are used to produce a noisy observation. This observation is the input of the RIM baseline model $G_{\varphi^\star_{\mathcal{D}}}$, which recovers the source and convergence map. The observed data is compared with the RIM prediction using normalized residual maps. To achieve noise level reconstruction ($\chi^2_\nu \simeq 1$), the model is fine-tuned by a likelihood optimization regularized by Elastic Weight Consolidation (EWC). We show the steps to generate a dataset from the conditional $\tilde{p}(\mathcal{D}^{\mathrm{train}} \mid \mathcal{T})$ using two VAE and the baseline prediction. These samples are used to compute the diagonal elements of the Fisher matrix $[\mathcal{I}(\varphi^\star_{\mathcal{D}})]_{jj}$.

in the gradient model adds flexibility to the inference by learning the initialization of the parameters as well.

Initialization can be performed by selecting a constant value (e.g. Morningstar et al. 2018, 2019) or by using an approximate inverse of the forward model ($\hat{F}^{-1}$) to estimate $\hat{\mathbf{x}}_0$ from the observed data (e.g. Lønning et al. 2019). With our approach, the initialization takes the form

$$\hat{\mathbf{x}}^{(0)} = g_\varphi(0, \mathbf{y}, 0). \tag{5}$$

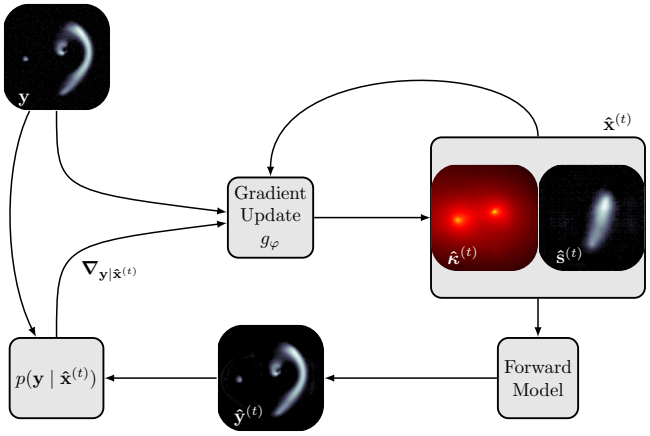

Figure 2. Rolled computational graph of the RIM.

This is a *learned* approximate inverse to the forward model. We note that merging the inverse $\hat{F}^{-1}$ inside the gradient model is not strictly necessary. A separate architecture could be learned before or during the optimisation of the gradient model. Such a design choice might be preferable in the case where the dimensionality of the observation and the parameters differs substantially.

Crucially, this adds an inductive bias that makes our MAP optimization problem more manageable. The learned initialization will favour an explicit mapping between $\mathcal{Y}$ and the region of $\mathcal{X}$ with a high prior probability density — that is, the region of the parameter space where the training dataset labels are found. Without this, information about the observation would only be filtered through the gradient of the likelihood. Since the likelihood is not a globally convex function, a first order method like the RIM will often fail to converge to the MAP if the starting solution $\hat{\mathbf{x}}^{(0)}$ is far from the MAP. The gradient of the likelihood, in that situation, would point toward a local minima that will conflict with the loss function $\mathcal{L}_\varphi$ minima, thus making the optimisation of the gradient model more difficult. If we suppose that an approximate inverse $\hat{F}^{-1}(\mathbf{y})$ exists — given a dataset $\mathcal{D}$ — and bake this function into the neural net architecture, then the RIM can take full advantage of the gradient information in the neighborhood of the MAP in $\mathcal{X}$.

When optimizing the gradient model on the training dataset, we use a standard recurrent neural net (RNN) objective where the loss at each step is accumulated and gradients are backpropagated through time (BPTT). The loss for each individual step is a weighted mean squared error over the M pixels of the labels, such that the total loss is

$$\mathcal{L}_\varphi(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{M} \mathbf{w}_i ([G_\varphi(\mathbf{y})]_i^{(t)} - \mathbf{x}_i)^2. \quad (6)$$

We use the notation $[G_\varphi(\mathbf{y})]_i^{(t)}$ to refer to i^{th} pixel of the RIM prediction at step t of the recurrent relation (4). When omitted, we assume $G_\varphi(\mathbf{y}) = [G_\varphi(\mathbf{y})]^{(T)}$. Throughout this paper, we will make a distinction between the loss function $\mathcal{L}_\varphi$ and the cost (also called empirical risk), which is defined as the expectation of the loss over a dataset $\mathcal{D}$. The RNN optimisation problem is to minimize the cost:

$$\hat{\varphi} = \underset{\varphi}{\operatorname{argmin}}\ \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\big[\mathcal{L}_\varphi(\mathbf{x}, \mathbf{y})\big]. \quad (7)$$

We follow previous work in setting a uniform weight over the time steps ($\mathbf{w}^{(t)} = \frac{\mathbf{w}}{T}$). The choice of the pixel weights $\mathbf{w}_i$ is informed by our physical intuition about the problem. This specific choice is discussed in section 4.2.

In figure 3, we show the unrolled computational graph of the RIM. During training of the gradient model g_φ, operations along the solid arrows are being recorded for BPTT. The recording is stopped along the dashed arrow since these operations are part of the forward modelling process. By avoiding the computation of these gradients, training time is reduced and knowledge about the inner workings of a specific likelihood (and forward model) is insulated from the optimization algorithm. This is analogous to a common RNN use-case like text generation, where the process responsible for producing the next element in a time series is a black box to the optimization algorithm.

The gradient of the likelihood is computed using automatic differentiation. Following (Modi et al. 2021), we preprocess the gradients using the Adam algorithm (Kingma & Welling 2013). For clarity, we only illustrated this step in Figure 4.

2.3. *The Gradient Model*

The neural network architecture is illustrated in Figure 4, which shows a single time step of the unrolled computation graph of the RIM. We use a U-net (Ronneberger et al. 2015) architecture with Gated Recurrent Units (GRU: Cho et al. 2014) placed in each skip connections.

Each GRU cell has it's own memory tensor that is updated through time at each iteration of equation 4. The shape of a memory tensor is set to match the feature tensor fed into it from the parent layer in the network graph. Instead of learning a compressed representation like in the hourglass architecture (i.e. autoencoder), the U-net architecture naturally separates the spatial frequency components of the signal into its vertical levels. The first level generally encodes high frequency features while the lower level encodes low frequency features (due

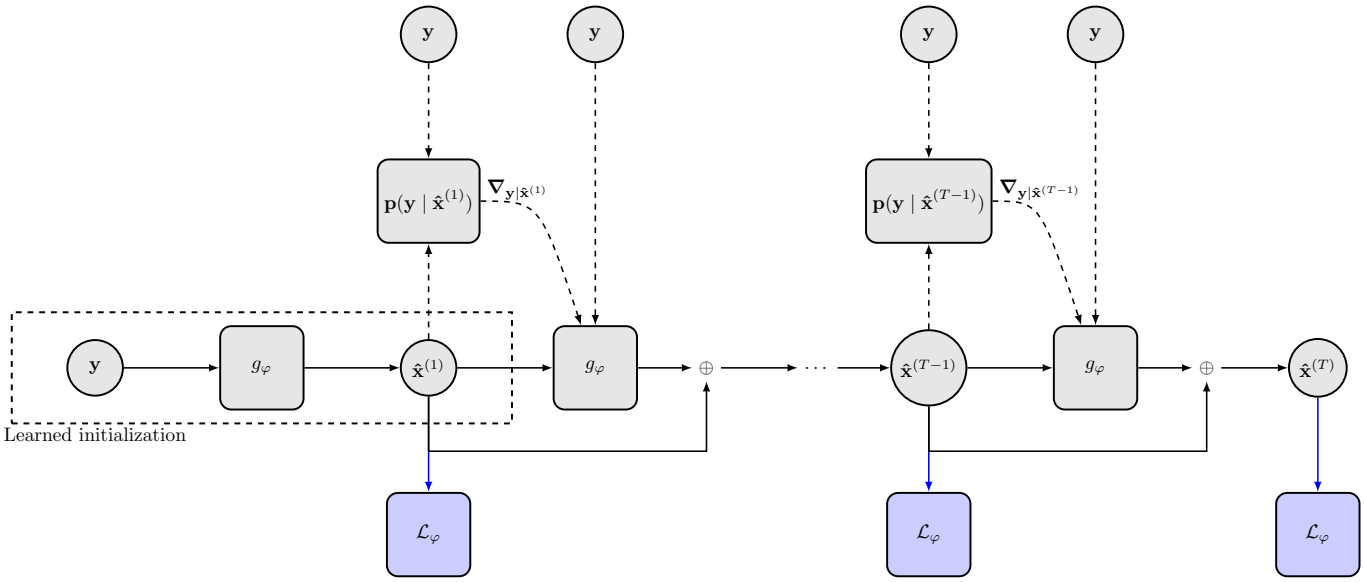

Figure 3. Unrolled computational graph of the RIM. Operations along solid arrows are being recorded for BPTT, while operations along dashed arrows are not. The blue arrows are only used for optimisation during training. During fine-tuning or testing, the loss is computed only as an oracle metric to validate that our methods can recover the ground truth.

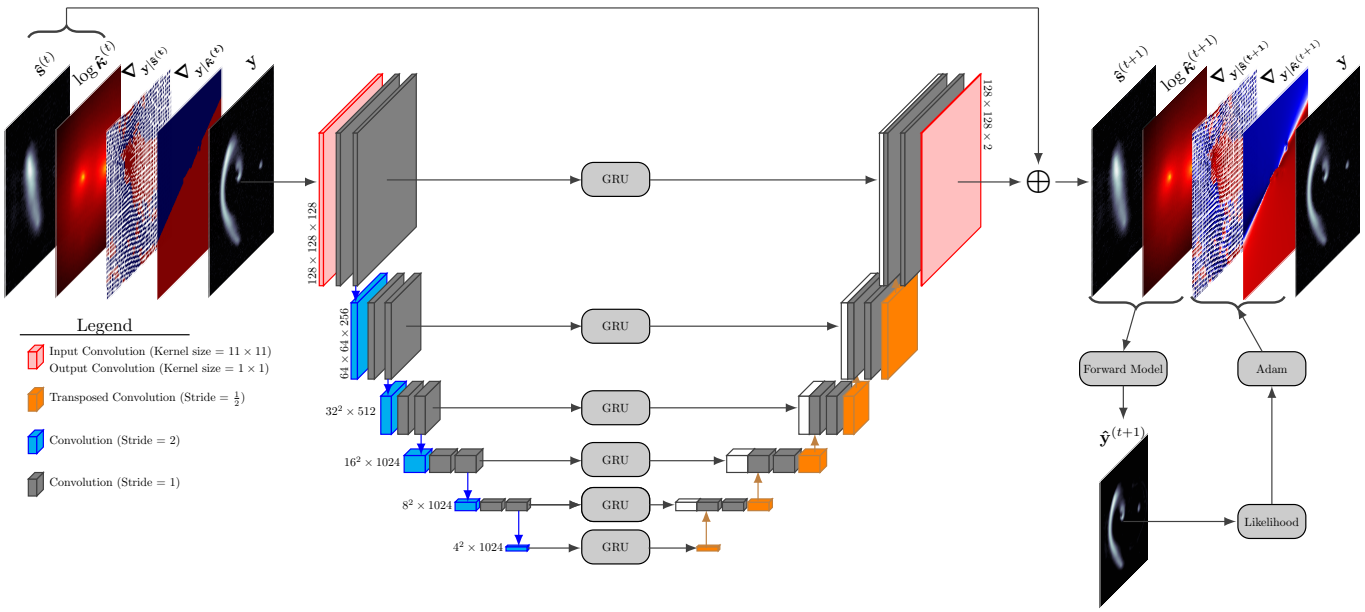

Figure 4. A single time step of the unrolled computation graph of the RIM. GRU units are placed in the skip connections to guide the reconstruction of the source and convergence. A schematic of the steps to compute the likelihood gradients is shown in the bottom right of the figure, including the Adam processing step. The $\oplus$ symbol refer to an addition operation. See the recurrent relation 4.

to downsampling of the feature maps). Adding an independent memory unit at each level preserve this property.

Convolutional layers with a stride of 2 are used for downsampling, while stride of $\frac{1}{2}$ are used for upsampling of the feature maps (identified in blue and orange respectively in figure 4). Most layer use a kernel size of 3×3, except the first and last layer. The first layer has larger receptive field (11×11) to capture more details in the input tensor, while the last layer has kernels of size 1×1. A tanh activation function is used for each convolutional layer, including strided convolutions, except for the output layer.

The U-net outputs an image tensor with two channels, one dedicated for the update of the source and the other to the update of the convergence (see figure 4).

Finally, we must address the notion of preprocessing in the context of a RIM. Since the prediction of the neural network are processed by a forward model during the inference, preprocessing must be implemented as a part of the RIM architecture. We use the notion of link function $\Psi : \Xi \to \mathcal{X}$ introduced by Putzky & Welling (2017), which is defined as an invertible transformation between the network prediction space $\boldsymbol{\xi} \in \Xi$ and the forward modelling space $\mathcal{X}$. The loss $\mathcal{L}_\varphi$ must be computed in the Ξ space in order to avoid gradient vanishing problems when Ψ is a non-linear mapping. The choice for Ψ is mentioned in section 4.2.

2.4. *The Forward Model*

An observation is simulated by ray tracing the brightness distribution of the background source to the foreground coordinate system. In our case, the coordinate systems have discretized representations. Each pixel of an image is labeled with a subscript index i, which we distinguish from a parenthesized superscript index (i) that refers to the member of a set or list of tensors. For clarity, we omit the superscript index in what follows.

Each pixel is associated with an intensity value and a coordinate vector. The foreground pixel coordinates $\boldsymbol{\theta}_i$ and the source pixel coordinates $\boldsymbol{\beta}_i$ are related by the lens equation

$$\boldsymbol{\beta}_i = \boldsymbol{\theta}_i - \boldsymbol{\alpha}(\boldsymbol{\theta}_i), \tag{8}$$

where $\boldsymbol{\alpha}$ is a deflection angle. It is obtained from the projected surface density field κ — also referred to as convergence — by the integral

$$\boldsymbol{\alpha}(\boldsymbol{\theta}_i) = \frac{1}{\pi} \int_{\mathbb{R}^2} \kappa(\boldsymbol{\theta}') \frac{\boldsymbol{\theta}_i - \boldsymbol{\theta}'}{\|\boldsymbol{\theta}_i - \boldsymbol{\theta}'\|^2} d^2\boldsymbol{\theta}' \tag{9}$$

The intensity of a pixel in a simulated observation is obtained by bilinear interpolation of the source brightness distribution at the coordinate $\boldsymbol{\beta}_i$. In this work, the convergence also has a discrete representation. Thus, we approximate this integral by a discrete global convolution. Taking advantage of the convolution theorem, this operation can be computed in near-linear time using the Fast Fourier Transform (FFT).

Assuming the observation has M^2 pixels, the convolution kernel would have $(2M+1)^2$ pixels. Both the convergence tensor and the kernel tensor are zero-padded to a size of $(4M+1)^2$ pixels in order to approximate a linear convolution and significantly reduce aliasing.

A blurring operator — convolution by a point spread function — is then applied to the lensed image to replicate the response of an imaging system. This operator is implemented as a GPU-accelerated matrix operation since the blurring kernels used in this paper have a significant proportion of their energy distribution encircled inside a small pixel radius.

2.5. *Fine-Tuning*

2.5.1. *Objective function*

Once the gradient model is trained, the RIM is a baseline estimator of the parameters $\mathbf{x}$ given a noisy observation $\mathbf{y}$. We now concern ourselves with a strategy to improve this estimator without having to retrain the gradient model from scratch. This is important for high SNR observation, where attaining noise level reconstruction gets exponentially more difficult.

The metric for the goodness of fit is the reduced chi squared $\chi^2_\nu = \frac{\chi^2}{\nu}$. ν is the total number of degrees of freedom. In this work, ν is the amount of pixels in $\mathbf{y}$. Generally, our goal will be to reach $\chi^2_\nu = 1$, where the residuals have reached noise level. We will also use the chi squared difference $|\chi^2 - \nu|$ when the reduced chi squared is very close to 1 to distinguish between reconstructions that reach noise level and those that still need improvement.

The χ^2 metric by itself is not sufficient to judge the quality of a reconstruction. This is why the full residual maps will be provided in the result section as well.

The fine-tuning objective is to minimize directly the χ^2:

$$\hat{\varphi}_{\text{MAP}} = \underset{\varphi}{\text{argmax}} \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{y} \mid [G_\varphi(\mathbf{y})]^{(t)}) + \log p(\varphi). \tag{10}$$

Unlike the squared loss (6), this objective function is implicit and makes no use of labels. This allows us to use this objective at test time.

2.5.2. *Transfer Learning*

We now address the issue of transferring knowledge from a training task (problem (7)) to a specific test task (problem (10)). The reader might refer to reviews on transfer learning (Pan & Yang 2010; Zhuang et al. 2019) for a broad overview of the field. The strategy we outline falls into the category of inductive transfer learning.

Since the data likelihood $p(\mathbf{y} \mid \mathbf{x})$ does not contain *a priori* information about the solution $\hat{\varphi}_{\text{MAP}}$, inductive biases must be introduced to make the problem (10) well-posed.

($\mathcal{H}_4$) Initializing φ with a pretrained set of weights that minimizes the cost over $\mathcal{D}^{\text{train}}$;

($\mathcal{H}_5$) Early stopping when a maximum number of steps is reached or $\chi^2_\nu \leq 1$;

($\mathcal{H}_6$) Using a small learning rate.

($\mathcal{H}_5$) and ($\mathcal{H}_6$) encode the assumption that the optimal estimator is to be found *near* the initialization.

As it turns out, ($\mathcal{H}_4$) is not strong enough to preserve the knowledge learned during pretraining. This has long been observed in the literature and was coined as the catastrophic interference phenomenon in connectionist networks (McCloskey & Cohen 1989; Ratcliff 1990). In summary, a sequential learning problem exhibits catastrophic forgetting of old knowledge when confronted with new examples (possibly from a different distribution or process), in a manner

(CF_1) proportional to the amount of learning;

(CF_2) strongly dependant to the disruption of the parameters involved in representing the old knowledge.

While ($\mathcal{H}_5$) and ($\mathcal{H}_6$) can potentially alleviate (CF_1), (CF_2) is not trivially addressed by the inductive biases introduced so far.

We follow the work of Kirkpatrick et al. (2016) to define a prior distribution over φ that address this issue. We denote the pretrained weights with $\varphi^\star_{\mathcal{D}}$ and the Fisher information matrix with $\mathcal{I}$:

$$\log p(\varphi) = -\frac{\lambda}{2}\sum_j [\mathcal{I}(\varphi^\star_{\mathcal{D}})]_{jj}(\varphi_j - \varphi^\star_{\mathcal{D},j})^2. \tag{11}$$

We've included a derivation of this term in the appendix C. The elements of the Fisher matrix are computed using examples from the training dataset that are similar to the observed lensing system $\mathbf{y}$.

We use the notation $\mathcal{T}$ to symbolize the random variable related to the event of observing an image $\mathbf{y}$, a noise covariance C and a PSF Π such that $(\mathbf{y}, C, \Pi) \sim \mathcal{T}$. Thus, the distribution of elements from the training dataset $\mathcal{D}^{\text{train}}$ that are similar to task $\mathcal{T}$ can be modeled by the conditional distribution $p(\mathcal{D}^{\text{train}} \mid \mathcal{T})$.

This distribution is intractable in general. Instead, we use the VAE trained on the training dataset labels as a surrogate distribution of the prior $\tilde{p}(\mathbf{x})$. After having observed $\mathcal{T}$, we can fully define the *noisy* forward model (equation (1)) using C and Π which gives us a surrogate of the conditional $(\tilde{\mathbf{x}}^{(i)}, F(\tilde{\mathbf{x}}^i) + \boldsymbol{\eta}) \sim \tilde{p}(\mathcal{D}^{\text{train}} \mid C, \Pi)$. The point estimate $\hat{\mathbf{x}}^{(T)}$ of the baseline RIM is used in conjonction with the encoder network of the VAE to constrain our sampling of the latent space to the region centered around the predicted mean of the latent code. This step conditions the surrogate prior on $\mathbf{y}$. These ideas are illustrated in figure 1.

3. DATA

3.1. *COSMOS*

Our sources brightness distributions are taken from the Hubble Space Telescope (HST) Advanced Camera for Surveys Wide Field Channel COSMOS field (Koekemoer et al. 2007; Scoville et al. 2007), a 1.64 deg^2 contiguous survey acquired in the F814W filter. A dataset of mag limited (F814W $<$ 23.5) deblended galaxy postage stamps (Leauthaud et al. 2007) was compiled as part of the GREAT3 challenge (Mandelbaum et al. 2014). The data is publicly available (Mandelbaum et al. 2012), and the preprocessing is done through the open source software `GALSIM` (Rowe et al. 2015).

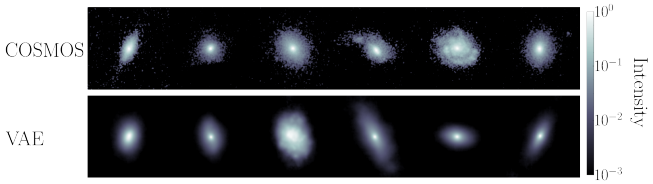

Figure 5. Examples of COSMOS galaxy images (top row) and VAE generated samples (bottom row) used as labels in $\mathcal{D}^{\text{train}}$.

We applied the `marginal` selection criteria (see the `COSMOSCatalog` class) and imposed a flux per image greater than 50 photons cm^{-2} s^{-1}. This final set has a total of 13 321 individual images. Each image is convolved with its original PSF and drawn into a postage stamps of 158^2 pixels. They are then background subtracted, randomly shifted and rotated by an angle of 90°. They are finally cropped to a 128^2 image. Random augmentations are applied only once per image, leaving the size of the dataset unchanged. The small amount of noise that is left in each image is removed using an autoencoder. Details regarding this procedure are found in the appendix B. Finally, pixel intensity is normalized in the range $[0, 1]$.

The final augmented set is then split into a training set (90%) and a test set (10%). The training set is used to train a VAE and produce simulated observations to train the RIM.

3.2. *IllustrisTNG*

3.2.1. *Smooth Particle Lensing*

To compute a convergence map from an N-body simulation, we follow Aubert et al. (2007) in treating each particle as flow tracers instead of describing their density as Dirac $\delta(\mathbf{r})$. Smoothing each particle density on a non-singular kernel reduces the particle noise affecting all important lensing quantities — most importantly the convergence. At the same time, the choice of the kernel size is important to preserve substructures in the lens that we might potentially be interested in. Following Rau et al. (2013), we use Gaussian smoothing with an adaptive kernel size determined by the distance of the

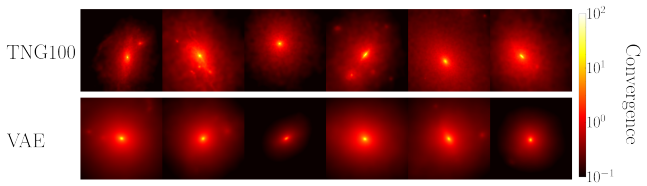

Figure 6. Examples of smoothed Illustris TNG100 convergence map (top row) and VAE generated samples (bottom row) used as labels in $\mathcal{D}^{\text{train}}$.

64^{th} nearest neighbours of a given particle $D_{64,i}$.

$$\kappa(\mathbf{x}) = \frac{1}{\Sigma_{\text{crit}}} \sum_{i=1}^{N_{\text{part}}} \frac{m_i}{2\pi\hat{\ell}_i^2} \exp\left(-\frac{1}{2}\frac{(\mathbf{x}-\mathbf{x}_i)^2}{\hat{\ell}_i^2}\right) \qquad (12)$$
$$\hat{\ell}_i = \sqrt{\frac{103}{1024}} D_{64,i}.$$

The nearest neighbours are found by fitting a k-d tree — implemented in `scikit-learn` (Pedregosa et al. 2011) — to the N_{part} particles in a cylinder centered on the centre of mass of the halo of interest. We canonically defined the critical surface density

$$\Sigma_{\text{crit}} = \frac{4\pi G}{c^2}\frac{D_\ell D_{\ell s}}{D_s}. \qquad (13)$$

D_ℓ, D_s and $D_{\ell s}$ are angular diameter distance to the lens, source and between the lens and the source respectively, G is the gravitational constant and c the speed of light.

3.2.2. *Data augmentation*

We used the last snapshot (redshift $z = 0$) of the IllustrisTNG-100 simulation (Nelson et al. 2019) to get physically plausible realizations of dark matter and baryonic matter halos. For our purposes, dark matter halos are synonymous with `subhaloes` in the data and galaxy clusters are associated with `halos`.

We select 1604 halos with the criteria that they have a total dark matter mass of at least $9 \times 10^{11} M_\odot$. We then collect all dark matter, gas, stars and black holes particles from the data associated to the galaxy cluster within which the halo resides in to create a smoothed projected surface density maps around the centroid of the halo as prescribed in section 3.2.1.

We adopt the ΛCDM cosmology from Planck Collaboration (2020) with $h = 0.68$ to compute angular diameter distances. We also fix the source redshift to $z_s = 1.5$ and the deflector redshift to $z_\ell = 0.5$. We note that changing the redshifts or the cosmology only amount in a rescaling of the κ map by a global scalar. The smoothed distributions from equation (12) are rendered into a regular grid of 188^2 pixels with a comoving field of view of 105 kpc/h. To avoid edge effects in the pixelated maps, we include particles outside of the field of view in the sum of equation (12).

Before applying augmentation or considering different projections, our dataset of halos is split into a training set (90%) and a test set (10%). We take 3 different projections (xy, xz and yz) of the 3D particle distribution, which amount to a total of 4 812 individual maps. Random 90° rotations and random shift to the pixel coordinates are applied to each image. The κ maps are then rescaled by a random factor to change their estimated Einstein radius to the range [0.5, 2.5] arcsec. The Einstein radius is defined as

$$\theta_E = \sqrt{\frac{4GM(\theta_E)}{c^2}\frac{D_{\ell s}}{D_\ell D_s}} \qquad (14)$$

where $M(\theta_E)$ is the mass enclosed inside the Einstein radius. In practice, we estimate this quantity by summing over the mass of pixels with a value greater than the critical density ($\kappa > 1$). For data augmentation purposes, this rough definition gives a good enough estimate of the size of the lensed image that will be produced by some κ map, circumventing the fact that the Einstein radius does not have a proper definition for most of the convergence fields of hydrodynamical simulations. We test multiple scaling factors for each κ map, then uniformly sample between those that produce an estimated Einstein radius within the desired range. This step is used to remove any bias in the Einstein radius that might come from the mass function of the simulation.

The final maps are cropped down to 128^2 pixels. Placed at a redshift $z_\ell = 0.5$, a κ map will thus span an angular field of view of 7.69″ with a resolution similar to HST. This field of view is wide enough to cover a typical gravitational lens observed in the sky, which partly justify our choice for the comoving field of view earlier.

With these augmentation procedures, a total of 50 000 maps are created from the training split and 5 000 from the test split. The training set is used to train a VAE and produce simulated observations to train the RIM.

3.3. *Simulated Observations*

Having defined a source map and a convergence map, we apply the ray tracing simulation prescribed in section 2.4 to produce an observation with observational effects that roughly corresponds to HST images.

For each observation, a Gaussian point spread function is created with a full width at half maximum (FWHM) randomly generated from a truncated normal distribution. The support of the distribution is truncated below by the angular size of a single pixel and

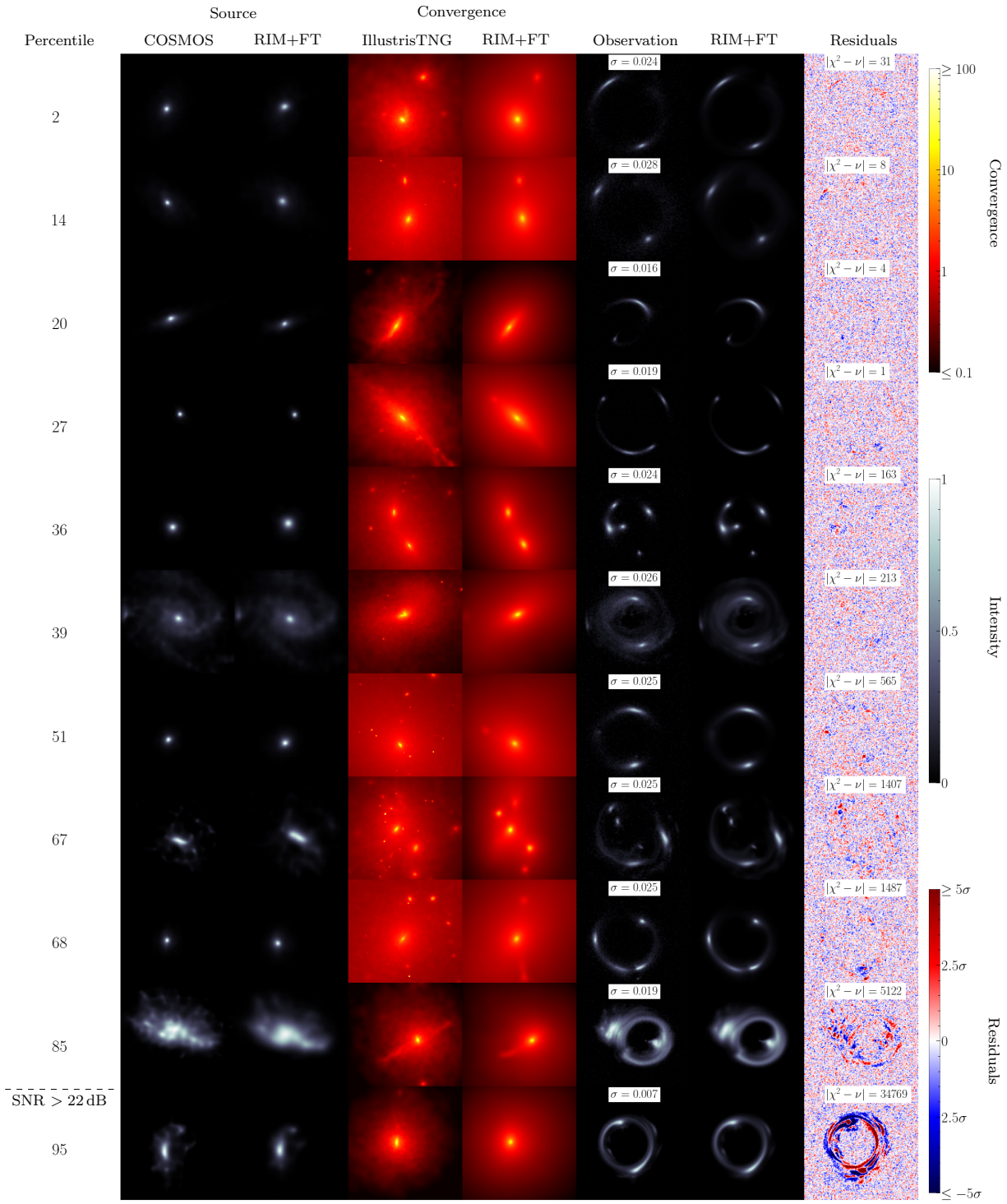

Figure 7. Cherry-picked sample of the fine-tuned RIM reconstructions on a test set of 3000 examples. Examples are ordered from the best χ^2 (top) to the worst (bottom). The percentile rank of each example is in the leftmost column. The last example shown has SNR above the threshold defined in Figure 9.

above by the angular size of 4 pixels. White noise with a standard deviation randomly generated from a truncated normal distribution is then added to the convolved observation to simulate SNR conditions between 10 dB and 30 dB. For simplicity, we define SNR $= \frac{1}{\sigma}$. This definition is equivalent to the peak signal-to-noise ratio.

We set the observed image field of view to match with the convergence field view ($7.69''$). We choose the back-

ground field of view to be $3''$. As a validation criteria for each simulated image, we impose a minimum magnification of 3. Thus, we make sure that most pixel coordinates in the image plane will be mapped inside the source coordinate system through the lens equation (8).

Table 1. Physical model parameters.

Parameter	Distribution/Value
Lens redshift z_ℓ	0.5
Source redshift z_s	1.5
Field of view (")	7.69
Source field of view (")	3
PSF FWHM (")	$\mathcal{TN}(0.06, 0.3; 0.08, 0.05)$ [a]
Noise amplitude σ	$\mathcal{TN}(0.001, 0.1; 0.01, 0.03)$

[a] We defined the parameters of the truncated normal in the order $\mathcal{TN}(a, b; \mu, \sigma)$, where $[a, b]$ defines the support of the distribution.

400 000 observations are simulated from random pairs of COSMOS sources and IllustrisTNG convergence training split in order to train the RIM. An additional 200 000 observations are created from pairs of COSMOS source and pixelated SIE convergence map are added to the dataset as well. The parameters for these κ maps are listed in table 2.

We found this addition to be beneficial to learning since it adds an inductive bias in the learning favoring isothermal profiles. We expect some lensing configurations like large Einstein rings or double images to poorly constrain the inner structure of the mass distribution. Building an inference pipeline with strong constraints on the slope of the profile goes beyond the scope of this work. As such, imposing a prior through the dataset is sufficient for our goal. It is also motivated by the *bulge-halo conspiracy* — the observation that most lensing configurations observed in the sky can be explained to first order approximation by an average slope consistent with an isothermal profile.

Table 2. SIE parameters.

Parameter	Distribution
Radial shift (arcsec)	$\mathcal{U}(0, 0.1)$
Azimutal shift	$\mathcal{U}(0, 2\pi)$
Orientation	$\mathcal{U}(0, \pi)$
θ_E (arcsec)	$\mathcal{U}(0.5, 2.5)$
Ellipticity	$\mathcal{U}(0, 0.6)$

1 600 000 simulated observations are generated from the VAE background sources and convergence maps as part of the training set. In principle, we could continuously generate examples from the VAE. However, having a fixed amount let us apply some validation check to each observation ($\mathbf{y}$) in order to avoid configuration like a single image of the background source or an Einstein ring cropped by the field of view.

4. TRAINING

4.1. *VAE*

As mentionned in Kingma & Welling (2019), direct optimisation of the ELBO can prove difficult because the reconstruction term $\log p_\theta(\mathbf{x} \mid \mathbf{z})$ is relatively weak compared to the Kullback Leibler (KL) divergence term. To alleviate this issue, we follow the work of Bowman et al. (2015) and Kaae Sønderby et al. (2016) in setting a warm-up schedule for the KL term in the ELBO (see appendix A), starting from $\beta = 0.1$ up to $\beta_{\max}$.

Usually, $\beta_{\max} = 1$ is considered optimal since it matches the original ELBO objective derived by Kingma & Welling (2013). But, we are more interested in the sharpness of our samples and accurate inference around small regions of the latent space for fine-tuning. Thus, setting $\beta_{\max} < 1$ allows us to increase the size of the information bottleneck (or latent space) of the VAE and improve the reconstruction cost of the model. This is a variant of the β-VAE (Higgins et al. 2017), where $\beta > 1$ was found to improve disentangling of the latent space (Burgess et al. 2018).

Table 3. Hyperparameters for the background source VAE.

Parameter	Value
Input preprocessing	$\mathbb{1}$
Architecture	
Levels (encoder and decoder)	3
Convolutional layer per level	2
Latent space dimension	32
Hidden Activations	Leaky ReLU
Output Activation	Sigmoid
Filters (first level)	16
Filters scaling factor (per level)	2
Number of parameters	3 567 361
Optimization	
Optimizer	Adam
Initial learning rate	10^{-4}
Learning rate schedule	Exponential Decay
Decay rate	0.5
Decay steps	30 000
Number of steps	500 000
$\beta_{\max}$	0.1
Batch size	20

Table 4. Hyperparameters for the convergence VAE.

Parameter	Value
Input preprocessing	$\log_{10}$
Architecture	
Levels (encoder and decoder)	4
Convolutional layer per level	1
Latent space dimension	16
Hidden Activations	Leaky ReLU
Output Activation	Linear
Filters (first level)	16
Filters scaling factor (per level)	2
Number of parameters	1 980 033
Optimization	
Optimizer	Adam
Initial learning rate	10^{-4}
Learning rate schedule	Exponential Decay
Decay rate	0.7
Decay steps	20 000
Number of steps	155 000
$\beta_{\max}$	0.2
Batch size	32

The value for $\beta_{\max}$ and the steepness of the schedule are grid searched alongside the architecture for the VAE. Our criteria for an optimal model is a VAE that achieve the lowest reconstruction error in order to produce sharp images. At the same time, the KL divergence should be below an empirically defined threshold to respect the latent space prior. This value is found in practice by manually looking at the quality of generated samples.

For the following architectural choice, we import the notion of *level* from the U-net architecture. However, we emphasize that there is no horizontal skip connection in the VAE since the encoder and decoder are fundamentally separate structures and must be trained as such. In each level, we place a block of convolutional layers before downsampling (encoder) or after upsampling (decoder). These operations are done with strided convolutions like in the U-net architecture of the gradient model.

A notable element of the VAE architecture is the use of a fully connected layer to reshape the features of the convolutional layer into the chosen latent space dimension. Following the work of Lanusse et al. (2021), we introduce an ℓ_2 penalty between the input and output of the bottleneck dense layers to encourage an identity mapping. This regularisation term is slowly removed during training.

4.2. *Baseline*

The choice for the pixel weight $\mathbf{w}_i$ of the loss (equation (6)) and the link function $\Psi(\boldsymbol{\xi})$ are now addressed.

For the source, we find that a linear link function ($\hat{\mathbf{s}} = \Psi(\boldsymbol{\xi}) = \mathbb{1}\boldsymbol{\xi}$) is better than an exponential or sigmoid link function. Because negative pixels are not excluded, a ReLU is applied to the predicted source at test time. The weights for the source pixels are uniform.

For the convergence, we use an exponential link function with base 10: $\hat{\boldsymbol{\kappa}} = \Psi(\boldsymbol{\xi}) = 10^{\boldsymbol{\xi}}$. This Ψ encodes the non-negativity of the convergence. Furthermore, it is a power transformation that leaves the linked pixel values $\boldsymbol{\xi}_i$ normally distributed, thus improving the learning through the non-linearities in the neural network.

The weights of the convergence loss function are chosen to encode the fact that the pixel with critical mass density ($\boldsymbol{\kappa}_i > 1$) have a stronger effect on the lensing configuration than other pixels. We find in practice that the weights

$$\mathbf{w}_i = \frac{\sqrt{\boldsymbol{\kappa}_i}}{\sum_i \boldsymbol{\kappa}_i}, \tag{15}$$

encode this knowledge in the loss function and improved both the empirical risk and the goodness of fit of the baseline model on early test runs.

The architecture of the gradient model was grid searched on smaller dataset ($\lesssim 10\,000$ examples) in order

to quickly identify a small grid of valid hyperparameters. Then, the best hyperparameters were identified using a two-stage training process on the training dataset. In the first stage, we trained 24 different architectures from this small hyperparameter set for approximately 4 days (wall time using a single Nvidia A100 gpu). Different architectures would have a training time much longer than others, and this was factored in the architecture selection process. For example, adding more time steps (T) to the recurrent relation (4) would yield better generalisation on the test set, but this would come at great costs to training time until convergence. Following this first stage, 4 architectures were deemed efficient enough to be trained for an additional 6 days. We report the hyperparameters for the best architecture after this second stage of training in table 5.

Table 5. Hyperparameters for the baseline RIM.

Parameter	Value
Source link function	$\mathbb{1}$
κ link function	10^{ξ}
Architecture	Figure 4
Recurrent steps (T)	8
Number of parameters	348 546 818
First Stage Optimisation	
Optimizer	Adamax
Initial learning rate	10^{-4}
Learning rate schedule	Exponential Decay
Decay rate	0.95
Decay steps	100 000
Number of steps	610 000
Batch size	1
Second Stage Optimisation	
Optimizer	Adamax
Initial learning rate	6×10^{-5}
Learning rate schedule	Exponential Decay
Decay rate	0.9
Decay steps	100 000
Number of steps	870 000
Batch size	1

Table 6. Hyperparameters for fine-tuning the RIM.

Parameter	Value
Optimizer	RMSProp
Learning rate	10^{-6}
Maximum number of steps	2 000
EWC Regularizer λ	0.4
Number of samples from VAE	200

5. RESULTS

In this section, we present the performance of our approach on the held out test set. A sample of 3000 reconstruction problems is generated from the held-out HST and IllustrisTNG data with noise conditions and PSFs similar to the training set.

5.1. *Goodness of Fit*

Figure 7 is a cherry picked sample of the reconstruction from the test set. Each reconstruction is performed by fine-tuning the baseline model on the observation for a maximum of 2000 steps ($\sim$ 20 minutes/reconstructions on a single Nvidia A100 GPU). Early stopping is applied when the χ^2 reaches noise level.

The samples are selected to showcase the wide range of lensing configuration that our approach can successfully solve at high SNR. We made a point to select mostly examples that have a lot of structure in their convergence map to distinguish our approach from existing analytical methods. We did not make an emphasis in selecting complicated sources since we judge that free-form reconstruction of the source is essentially a solved problem. Many methods can reconstruct the source to very high precision once the convergence map is known or well constrained.

To offset our selection bias, we selected samples in different percentile from the test set rank ordered by the χ^2 metric. We also show a randomly selected sample in Figure 11.

Figure 8 shows a comparison between the goodness of fit of the baseline model and the fine-tuned prediction. The left panel shows the loss difference between the fine-tune prediction and the baseline model. This is important context for the right panel, which shows the improvement for the χ^2 metric that is directly optimized by the fine-tuning procedure (10). We do not have a direct way to evaluate the prior, so we use the loss function $\mathcal{L}_\varphi$ as a surrogate measure. The fine-tuning procedure does not significantly deteriorate or improve the loss of the baseline prediction. This result is consistent with the claim that EWC regularisation preserves the prior learned during pretraining. We explore this in more details in Section 5.2.

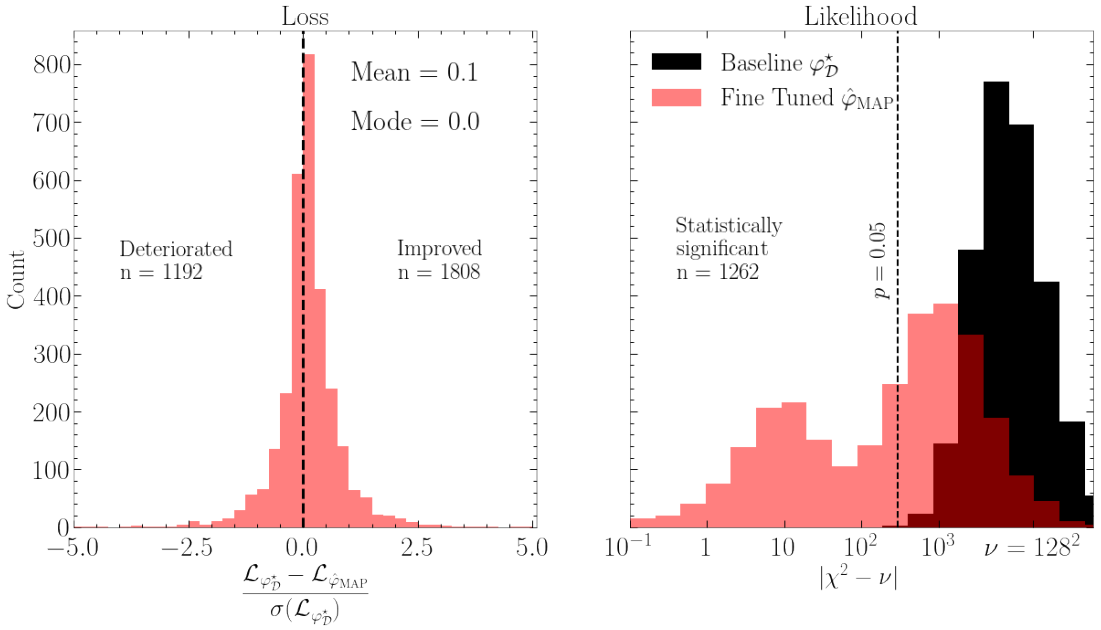

Figure 8. Summary of the goodness of fit (right panel) and loss (left panel) of the fine-tuned reconstructions compared with the baseline model reconstructions on the test set. The χ^2 is significantly improved by the optimisation. The loss, which is not directly optimized during fine-tuning, is bound within $\sim 2.5\sigma$ of the baseline loss because of the regularisation (EWC).

The χ^2 of the reconstruction is improved substantially compared to the baseline model as shown in the right panel. This is to be expected since the fine-tuning objective is to minimize the χ^2. Out of the 3000 reconstructions, 1262 have $\chi^2 < 296$. Those reconstructions have residual maps with a probability $p > 0.05$ of having been generated from a normal distribution $\mathcal{N}(0, \mathbb{1})$.

We now report the performance limit of our method. We are interested in the maximum SNR at which our method can still achieve noise level reconstruction. We purposefully ignore the low SNR regime since analytical methods perform well in that regime. Furthermore, RIM are generally robust against noise and our tests of the model suggested that it can perform meaningful reconstructions up to $\sigma \lesssim 0.2$. This upper bound is well above the regime at which our method is intended to operate.

We report the distribution of χ^2_ν against the noise level in Figure 9. Two behaviors can be identified. For SNR below a certain threshold, the goodness of fit of the fine-tuned model is essentially flat around the noise level. For SNR above the threshold, the goodness of fit follows the trend $\chi^2 \propto \sigma^{-2}$, which means the reconstructions have stopped improving on par with the noise level. We define the threshold at the data point with largest SNR value which still have statistically significant residuals ($|\chi^2 - \nu| < 296$). We find this threshold value to be 22 dB based on the test set. This is well above the peak SNR value of most HST data of known gravitational lenses

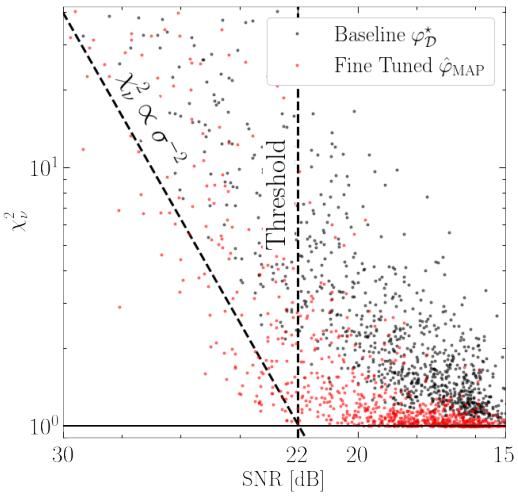

Figure 9. Goodness of fit as a function of SNR shows a threshold behavior where our method reaches its limit.

5.2. *Quality of the Reconstructions*

The loss $\mathcal{L}_\varphi$ by itself is not a strong enough metric to judge the quality of the reconstructions. Information about the spatial correlation of pixels is lost in that metric. The coherence spectrum encode this information via the cross correlation:

$$\gamma(k) = \frac{P_{12}(k)}{\sqrt{P_{11}(k) P_{22}(k)}}. \tag{16}$$

$P_{ij}(k)$ is the cross power spectrum of images i and j at the wavenumber k. We report the mean value and the 68% confidence interval of those spectrum in Figure 10

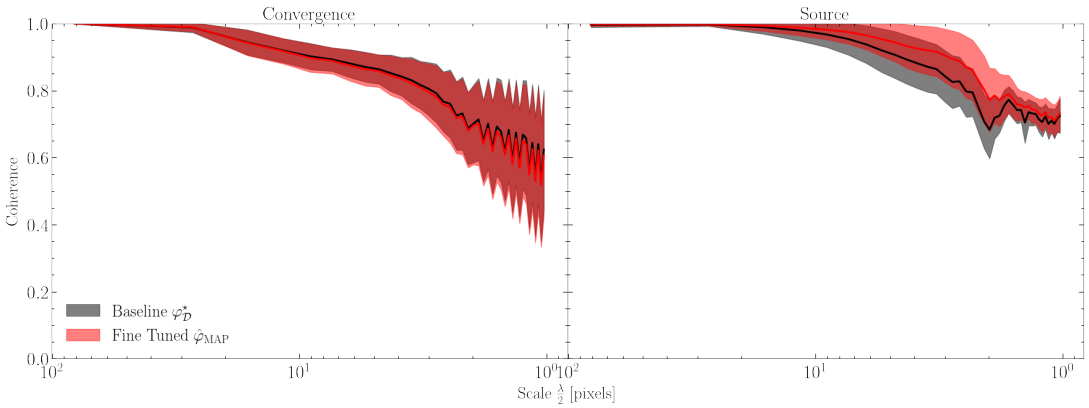

Figure 10. Statistics of the coherence spectrum on the test set. The solid line is the average coherence. The transparent region is the 68% confidence interval. The fine-tuning procedure yields a noticeable improvement on the coherence of the source at all frequencies.

for the convergence and source maps in the test set. The fine-tuning procedure is able to improve significantly the coherence of the background source at all scales. The coherence spectrum of the convergence remains unchanged however. This is to be expected since there is much fewer constraints on the convergence than there are on the background brightness distribution when using only the lensed image in the likelihood.

6. CONCLUSION

In this work, we introduced a framework for free-form gravitational lensing inference at high SNR. This is an improvement upon traditional modeling. Our framework can recover MAP point estimate of both the source and the convergence on pixelated grids with high precision and high accuracy, for a wide range of gravitational lensing configurations.

We believe that this framework will enable detailed modelling of present high SNR data and upcoming images from facilities like the James Webb Telescope, thereby pushing our understanding of dark matter and baryonic matter distribution at the very small scale.

7. ACKNOWLEDGEMENTS

REFERENCES

Abdelsalam, H. M., Saha, P., & Williams, L. L. R. 1998a, AJ, 116, 1541, doi: 10.1086/300546

—. 1998b, MNRAS, 294, 734, doi: 10.1046/j.1365-8711.1998.01356.x

Andrychowicz, M., Denil, M., Gomez, S., et al. 2016, arXiv e-prints, arXiv:1606.04474. https://arxiv.org/abs/1606.04474

Aubert, D., Amara, A., & Metcalf, R. B. 2007, MNRAS, 376, 113, doi: 10.1111/j.1365-2966.2006.11296.x

Auger, M. W., Treu, T., Bolton, A. S., et al. 2010, ApJ, 724, 511, doi: 10.1088/0004-637X/724/1/511

Barnabè, M., Czoske, O., Koopmans, L. V. E., et al. 2009, MNRAS, 399, 21, doi: 10.1111/j.1365-2966.2009.14941.x

Baxter, J. 2011, arXiv e-prints, arXiv:1106.0245. https://arxiv.org/abs/1106.0245

Bellagamba, F., Tessore, N., & Metcalf, R. B. 2016, Monthly Notices of the Royal Astronomical Society, 464, 4823, doi: 10.1093/mnras/stw2726

Belokurov, V., Evans, N. W., Moiseev, A., et al. 2007, ApJL, 671, L9, doi: 10.1086/524948

Bengio, Y. 2009, Found. Trends Mach. Learn., 2, 1–127, doi: 10.1561/2200000006

Bergamini, P., Rosati, P., Vanzella, E., et al. 2021, A&A, 645, A140, doi: 10.1051/0004-6361/202039564

Birrer, S., Amara, A., & Refregier, A. 2015, ApJ, 813, 102, doi: 10.1088/0004-637X/813/2/102

Birrer, S., Treu, T., Rusu, C. E., et al. 2019, MNRAS, 484, 4726, doi: 10.1093/mnras/stz200

Birrer, S., Shajib, A. J., Galan, A., et al. 2020, A&A, 643, A165, doi: 10.1051/0004-6361/202038861

Bowman, S. R., Vilnis, L., Vinyals, O., et al. 2015, arXiv e-prints, arXiv:1511.06349. https://arxiv.org/abs/1511.06349

Burgess, C. P., Higgins, I., Pal, A., et al. 2018, arXiv e-prints, arXiv:1804.03599. https://arxiv.org/abs/1804.03599

Chen, G. C. F., Fassnacht, C. D., Suyu, S. H., et al. 2019, MNRAS, 490, 1743, doi: 10.1093/mnras/stz2547

Cheng, J., Wiesner, M. P., Peng, E.-H., et al. 2019, ApJ, 872, 185, doi: 10.3847/1538-4357/ab0029

Cho, K., van Merrienboer, B., Gulcehre, C., et al. 2014, arXiv e-prints, arXiv:1406.1078. https://arxiv.org/abs/1406.1078

Dalal, N., & Kochanek, C. S. 2002, ApJ, 572, 25, doi: 10.1086/340303

Diego, J. M., Protopapas, P., Sandvik, H. B., & Tegmark, M. 2005, MNRAS, 360, 477, doi: 10.1111/j.1365-2966.2005.09021.x

Dutton, A. A., & Treu, T. 2014, Monthly Notices of the Royal Astronomical Society, 438, 3594, doi: 10.1093/mnras/stt2489

Gilman, D., Birrer, S., Nierenberg, A., et al. 2020, MNRAS, 491, 6077, doi: 10.1093/mnras/stz3480

Gilman, D., Bovy, J., Treu, T., et al. 2021, MNRAS, 507, 2432, doi: 10.1093/mnras/stab2335

Goyal, A., & Bengio, Y. 2020, arXiv e-prints, arXiv:2011.15091. https://arxiv.org/abs/2011.15091

Grillo, C., Rosati, P., Suyu, S. H., et al. 2018, ApJ, 860, 94, doi: 10.3847/1538-4357/aac2c9

Hezaveh, Y. D., Dalal, N., Marrone, D. P., et al. 2016, ApJ, 823, 37, doi: 10.3847/0004-637X/823/1/37

Higgins, I., Matthey, L., Pal, A., et al. 2017, in ICLR

Hoekstra, H., Bartelmann, M., Dahle, H., et al. 2013, SSRv, 177, 75, doi: 10.1007/s11214-013-9978-5

Jauzac, M., Klein, B., Kneib, J.-P., et al. 2021, MNRAS, 508, 1206, doi: 10.1093/mnras/stab2270

Kaae Sønderby, C., Raiko, T., Maaløe, L., Kaae Sønderby, S., & Winther, O. 2016, arXiv e-prints, arXiv:1602.02282. https://arxiv.org/abs/1602.02282

Keeton, C. R. 2001, arXiv e-prints, astro. https://arxiv.org/abs/astro-ph/0102341

Kingma, D. P., & Welling, M. 2013, arXiv e-prints, arXiv:1312.6114. https://arxiv.org/abs/1312.6114

—. 2019, arXiv e-prints, arXiv:1906.02691. https://arxiv.org/abs/1906.02691

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. 2016, arXiv e-prints, arXiv:1612.00796. https://arxiv.org/abs/1612.00796

Kneib, J.-P., & Natarajan, P. 2011, A&A Rv, 19, 47, doi: 10.1007/s00159-011-0047-3

Koekemoer, A. M., Aussel, H., Calzetti, D., et al. 2007, The Astrophysical Journal Supplement Series, 172, 196, doi: 10.1086/520086

Koopmans, L. V. E. 2005, MNRAS, 363, 1136, doi: 10.1111/j.1365-2966.2005.09523.x

Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, ApJ, 649, 599, doi: 10.1086/505696

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12 (Red Hook, NY, USA: Curran Associates Inc.), 1097–1105

Lanusse, F., Mandelbaum, R., Ravanbakhsh, S., et al. 2021, MNRAS, 504, 5543, doi: 10.1093/mnras/stab1214

Leauthaud, A., Massey, R., Kneib, J.-P., et al. 2007, The Astrophysical Journal Supplement Series, 172, 219, doi: 10.1086/516598

Lecun, Y., & Bengio, Y. 1995, Convolutional Networks for Images, Speech and Time Series, ed. M. A. Arbib (The MIT Press), 255–258

Li, N., Becker, C., & Dye, S. 2021, MNRAS, 504, 2224, doi: 10.1093/mnras/stab984

Li, R., Frenk, C. S., Cole, S., et al. 2016, MNRAS, 460, 363, doi: 10.1093/mnras/stw939

Lønning, K., Putzky, P., Sonke, J. J., et al. 2019, Medical Image Analysis, 53, 64, doi: 10.1016/j.media.2019.01.005

Mandelbaum, R., Lackner, C., Leauthaud, A., & Rowe, B. 2012, Zenodo. https://zenodo.org/record/3242143

Mandelbaum, R., Rowe, B., Bosch, J., et al. 2014, The Astrophysical Journal Supplement Series, 212, 5, doi: 10.1088/0067-0049/212/1/5

Marrone, D. P., Spilker, J. S., Hayward, C. C., et al. 2018, Nature, 553, 51, doi: 10.1038/nature24629

McCloskey, M., & Cohen, N. J. 1989in (Academic Press), 109–165, doi: 10.1016/S0079-7421(08)60536-8

Merten, J. 2016, MNRAS, 461, 2328, doi: 10.1093/mnras/stw1413

Millon, M., Courbin, F., Bonvin, V., et al. 2020, A&A, 640, A105, doi: 10.1051/0004-6361/202037740

Modi, C., Lanusse, F., Seljak, U., Spergel, D. N., & Perreault-Levasseur, L. 2021, arXiv e-prints, arXiv:2104.12864. https://arxiv.org/abs/2104.12864

Morningstar, W. R., Hezaveh, Y. D., Levasseur, L. P., et al. 2018, arXiv e-prints. https://arxiv.org/abs/1808.00011v1

Morningstar, W. R., Levasseur, L. P., Hezaveh, Y. D., et al. 2019, The Astrophysical Journal, 883, 14, doi: 10.3847/1538-4357/ab35d7

Natarajan, P., Chadayammuri, U., Jauzac, M., et al. 2017, MNRAS, 468, 1962, doi: 10.1093/mnras/stw3385

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493, doi: 10.1086/304888

Nelson, D., Springel, V., Pillepich, A., et al. 2019, MNRAS, 6, doi: 10.1186/s40668-019-0028-x

Pan, S. J., & Yang, Q. 2010, IEEE Transactions on Knowledge and Data Engineering, 22, 1345

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Planck Collaboration. 2020, A&A, 641, A6, doi: 10.1051/0004-6361/201833910

Putzky, P., & Welling, M. 2017, arXiv e-prints. https://arxiv.org/abs/1706.04008

Ratcliff, R. 1990, Psychological Review, 97, 285, doi: 10.1037/0033-295X.97.2.285

Rau, S., Vegetti, S., & White, S. D. 2013, Monthly Notices of the Royal Astronomical Society, 430, 2232, doi: 10.1093/mnras/stt043

Refsdal, S. 1964, MNRAS, 128, 307, doi: 10.1093/mnras/128.4.307

Rizzo, F., Vegetti, S., Powell, D., et al. 2020, Nature, 584, 201, doi: 10.1038/s41586-020-2572-6

Ronneberger, O., Fischer, P., & Brox, T. 2015, arXiv e-prints, arXiv:1505.04597. https://arxiv.org/abs/1505.04597

Rowe, B. T., Jarvis, M., Mandelbaum, R., et al. 2015, Astronomy and Computing, 10, 121, doi: 10.1016/j.ascom.2015.02.002

Rusu, C. E., Fassnacht, C. D., Sluse, D., et al. 2017, MNRAS, 467, 4220, doi: 10.1093/mnras/stx285

Rusu, C. E., Wong, K. C., Bonvin, V., et al. 2020, MNRAS, 498, 1440, doi: 10.1093/mnras/stz3451

Saha, P., & Williams, L. L. R. 1997, MNRAS, 292, 148, doi: 10.1093/mnras/292.1.148

Schuldt, S., Chirivì, G., Suyu, S. H., et al. 2019, A&A, 631, A40, doi: 10.1051/0004-6361/201935042

Scoville, N., Aussel, H., Brusa, M., et al. 2007, The Astrophysical Journal Supplement Series, 172, 1, doi: 10.1086/516585

Sérsic, J. L. 1963, Boletin de la Asociacion Argentina de Astronomia La Plata Argentina, 6, 41

Sluse, D., Sonnenfeld, A., Rumbaugh, N., et al. 2017, MNRAS, 470, 4838, doi: 10.1093/mnras/stx1484

Sun, F., Egami, E., Pérez-González, P. G., et al. 2021, ApJ, 922, 114, doi: 10.3847/1538-4357/ac2578

Suyu, S. H., & Blandford, R. D. 2006, MNRAS, 366, 39, doi: 10.1111/j.1365-2966.2005.09854.x

Suyu, S. H., Marshall, P. J., Hobson, M. P., & Blandford, R. D. 2006, MNRAS, 371, 983, doi: 10.1111/j.1365-2966.2006.10733.x

Treu, T., & Koopmans, L. V. E. 2004, ApJ, 611, 739, doi: 10.1086/422245

Treu, T., & Marshall, P. J. 2016, A&A Rv, 24, 11, doi: 10.1007/s00159-016-0096-8

Treu, T., Brammer, G., Diego, J. M., et al. 2016, ApJ, 817, 60, doi: 10.3847/0004-637X/817/1/60

Ulyanov, D., Vedaldi, A., & Lempitsky, V. 2017, arXiv e-prints, arXiv:1711.10925. https://arxiv.org/abs/1711.10925

Vegetti, S., & Koopmans, L. V. E. 2009, MNRAS, 392, 945, doi: 10.1111/j.1365-2966.2008.14005.x

Vegetti, S., Koopmans, L. V. E., Auger, M. W., Treu, T., & Bolton, A. S. 2014, MNRAS, 442, 2017, doi: 10.1093/mnras/stu943

Vegetti, S., Koopmans, L. V. E., Bolton, A., Treu, T., & Gavazzi, R. 2010, MNRAS, 408, 1969, doi: 10.1111/j.1365-2966.2010.16865.x

Vegetti, S., Lagattuta, D. J., McKean, J. P., et al. 2012, Nature, 481, 341, doi: 10.1038/nature10669

Vieira, J. D., Marrone, D. P., Chapman, S. C., et al. 2013, Nature, 495, 344, doi: 10.1038/nature12001

Warren, S. J., & Dye, S. 2003, ApJ, 590, 673, doi: 10.1086/375132

Wolpert, D. H., & Macready, W. G. 1995

Wong, K. C., Suyu, S. H., Auger, M. W., et al. 2017, MNRAS, 465, 4895, doi: 10.1093/mnras/stw3077

Wong, K. C., Suyu, S. H., Chen, G. C. F., et al. 2020, MNRAS, 498, 1420, doi: 10.1093/mnras/stz3094

Zhuang, F., Qi, Z., Duan, K., et al. 2019, arXiv e-prints, arXiv:1911.02685. https://arxiv.org/abs/1911.02685

APPENDIX

A. VARIATIONAL AUTOENCODER (VAE)

When working with limited data, data augmentation is crucial to insure that the trained model is robust against all sort of perturbations — like rotations of an image — that are not directly included as symmetries in the architecture of the model. In that sense, data augmentation is a way to induce (or remove) a bias via **??** that is not enforced by ($\mathcal{H}_2$). In section **??**, we discuss the different methods for augmentations applied to our data. In this section, we discuss a generative modelling approach to data augmentation that will complement the other ones.

VAEs were originally introduced by Kingma & Welling (2013) as a framework to do approximate inference on intractable posterior distributions with a latent variable graphical model. We aim here to briefly cover the most salient concepts related to our work, and refer the reader to the white paper of Kingma & Welling (2019).

A VAE is decomposed into two parts. On the one hand, an encoder q_ϕ is a stochastic function which approximate the posterior of a latent variable $\mathbf{z}$:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) \approx p_\theta(\mathbf{z} \mid \mathbf{x}). \tag{A1}$$

On the other hand, the decoder networks with parameters θ is the generative part of the VAE. It learns to decode the latent space into meaningful features of the data $\mathbf{x}$. In this sense, the generative model goal is to learn a posterior of the data $p_\theta(\mathbf{x} \mid \mathbf{z})$. The objective function for this problem is the evidence lower bound ($\mathcal{L}_{\phi,\theta}$: ELBO) which aims to satisfy (A1):

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{x})\right] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\big). \tag{A2}$$

To make this problem tractable, we assume the latent variable distributions should follow a normal distribution with a diagonal covariance matrix:

$$p(\mathbf{z}) \sim \mathcal{N}(0, \mathbb{1}) \tag{A3}$$

Under the reparameterization trick (Kingma & Welling 2013)

$$\begin{aligned} \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \mathbb{1}) \\ (\boldsymbol{\mu}, \log \boldsymbol{\sigma}) &= \mathrm{Encoder}_\phi(\mathbf{x}) \\ \mathbf{z} &= \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \end{aligned} \tag{A4}$$

the ELBO can then be differentiated w.r.t ϕ and θ since this choice yields a tractable ELBO with a functional form described by equation (10) of Kingma & Welling (2013):

$$\mathcal{L}_{\phi,\theta} \simeq \log p_\theta(\mathbf{x} \mid \mathbf{z}) + \frac{\beta}{2} \sum_{j=1}^{J} \big(1 + \log(\boldsymbol{\sigma}_j^2) - \boldsymbol{\mu}_j^2 - \boldsymbol{\sigma}_j^2\big). \tag{A5}$$

We've introduced the β parameters to balance the KL term with the reconstruction error as mentionned in section 4.1. Once trained, the generative model $\mathrm{Decoder}_\phi(\mathbf{z})$ can be used to generate new examples from the latent space $\mathbf{z} \sim \mathcal{N}(0, \mathbb{1})$.

B. AUTOENCODER

C. ELASTIC WEIGHT CONSOLIDATION

Suppose we are given a training set $\mathcal{D}$ and a test task $\mathcal{T}$. The posterior of the RIM parameters φ can be rewritten using the Bayes rule as

$$p(\varphi \mid \mathcal{D}, \mathcal{T}) = \frac{p(\mathcal{T} \mid \mathcal{D}, \varphi) p(\varphi \mid \mathcal{D})}{p(\mathcal{T} \mid \mathcal{D})}. \tag{C6}$$

We suppose that φ encode information about $\mathcal{D}$, while $\mathcal{T}$ was unseen by φ. It follows that $\mathcal{T}$ and $\mathcal{D}$ are conditionally independent when given φ. We do not make the stronger assumption that $\mathcal{D}$ and $\mathcal{T}$ are completely independent. In fact, such an assumption would contradict the premiss of our work that building a dataset $\mathcal{D}$ can inform a machine G_φ about task $\mathcal{T}$ — or that, more broadly, $\mathcal{D}$ contains information about $\mathcal{T}$.

We rewrite the marginal $p(\mathcal{T} \mid \mathcal{D})$ using the Bayes rule in order to extract the sampling distribution used to compute the Fisher diagonal elements $p(\mathcal{D} \mid \mathcal{T})$ s.t.

$$p(\varphi \mid \mathcal{D}, \mathcal{T}) = \frac{p(\mathcal{T} \mid \varphi) p(\varphi \mid \mathcal{D})}{p(\mathcal{D} \mid \mathcal{T})} \frac{p(\mathcal{T})}{p(\mathcal{D})}. \tag{C7}$$

The log-likelihood $\log p(\mathcal{T} \mid \varphi)$ is equivalent to the negative of the loss function for the particular task at hand. In this work, we assign a uniform probability density to $p(\mathcal{T})$ and $p(\mathcal{D})$ in order to ignore them.

We now turn to the prior $p(\varphi \mid \mathcal{D})$, which appears as a conditional relative to the training dataset. We use the Laplace approximation around the maxima $\varphi^\star_{\mathcal{D}}$ to evaluate the prior, where $\varphi^\star_{\mathcal{D}}$ are the trained parameters (learned with dataset $\mathcal{D}$) that minimize the training cost. The Taylor expansion of the prior around this maxima yields

$$\log p(\varphi \mid \mathcal{D}) \approx \log p(\varphi^\star_{\mathcal{D}} \mid \mathcal{D}) + \frac{1}{2} (\varphi - \varphi^\star_{\mathcal{D}})^T \underbrace{\left(\frac{\partial^2 \log p(\varphi \mid \mathcal{D})}{\partial^2 \varphi} \bigg|_{\varphi^\star_{\mathcal{D}}} \right)}_{\mathbf{H}(\varphi^\star_{\mathcal{D}})} (\varphi - \varphi^\star_{\mathcal{D}}). \tag{C8}$$

Since $\varphi^\star_{\mathcal{D}}$ is an extrema of the prior, the linear term vanishes. The empirical estimate of the negative hessian matrix is the observed Fisher information matrix which can be written as

$$\mathcal{I}(\varphi^\star_{\mathcal{D}}) = -\mathbb{E}_{\mathcal{D}|\mathcal{T}}[\mathbf{H}(\varphi^\star_{\mathcal{D}})] = \mathbb{E}_{\mathcal{D}|\mathcal{T}} \left[\left(\left(\frac{\partial \log p(\varphi \mid \mathcal{D})}{\partial \varphi} \right) \left(\frac{\partial \log p(\varphi \mid \mathcal{D})}{\partial \varphi} \right)^T \right) \bigg|_{\varphi^\star_{\mathcal{D}}} \right]. \tag{C9}$$

The expectation is taken over the sample space $p(\mathcal{D} \mid \mathcal{T})$ since the network parameters are held fixed during sampling. In order to compute the Fisher score, we apply once more the Bayes rule to extract a loss function:

$$\log p(\varphi \mid \mathcal{D}) = \log p(\mathcal{D} \mid \varphi) + \log p(\varphi) - \log p(\mathcal{D}). \tag{C10}$$

$p(\mathcal{D} \mid \varphi)$ is the negative of a loss function. We use a rescaled version of (6):

$$\log p((\mathbf{x}, \mathbf{y}) = \mathcal{D} \mid \varphi) = -TM \mathcal{L}_\varphi(\mathbf{x}, \mathbf{y}), \tag{C11}$$

where M is the total number of pixels in $\mathbf{x}$ and T is the total number of steps for the recurrent inference machine. The rescaling removes the artificial temperature added to the loss during training. The derivative of $\log p(\varphi)$ remains constant when taking the expectation of (C9). If needed, it can be modeled with a traditional ℓ_2 loss. But, we find that this term can safely be ignored.

Since the full Fisher matrix is intractable for a neural network, we approximate the quadratic term of the prior with the diagonal of the Fisher matrix following Kirkpatrick et al. (2016). For an optimisation problem, the first term of (C8) is constant. Thus, the posterior becomes proportional to

$$\log p(\varphi \mid \mathcal{D}, \mathcal{T}) \propto \log p(\mathcal{T} \mid \varphi) - \frac{\lambda}{2} \sum_j [\mathcal{I}(\varphi^\star_{\mathcal{D}})]_{jj} (\varphi_j - \varphi^\star_{\mathcal{D},j})^2. \tag{C12}$$

The Lagrange multiplier λ is introduced to weight the importance of the regularisation during fine-tuning.

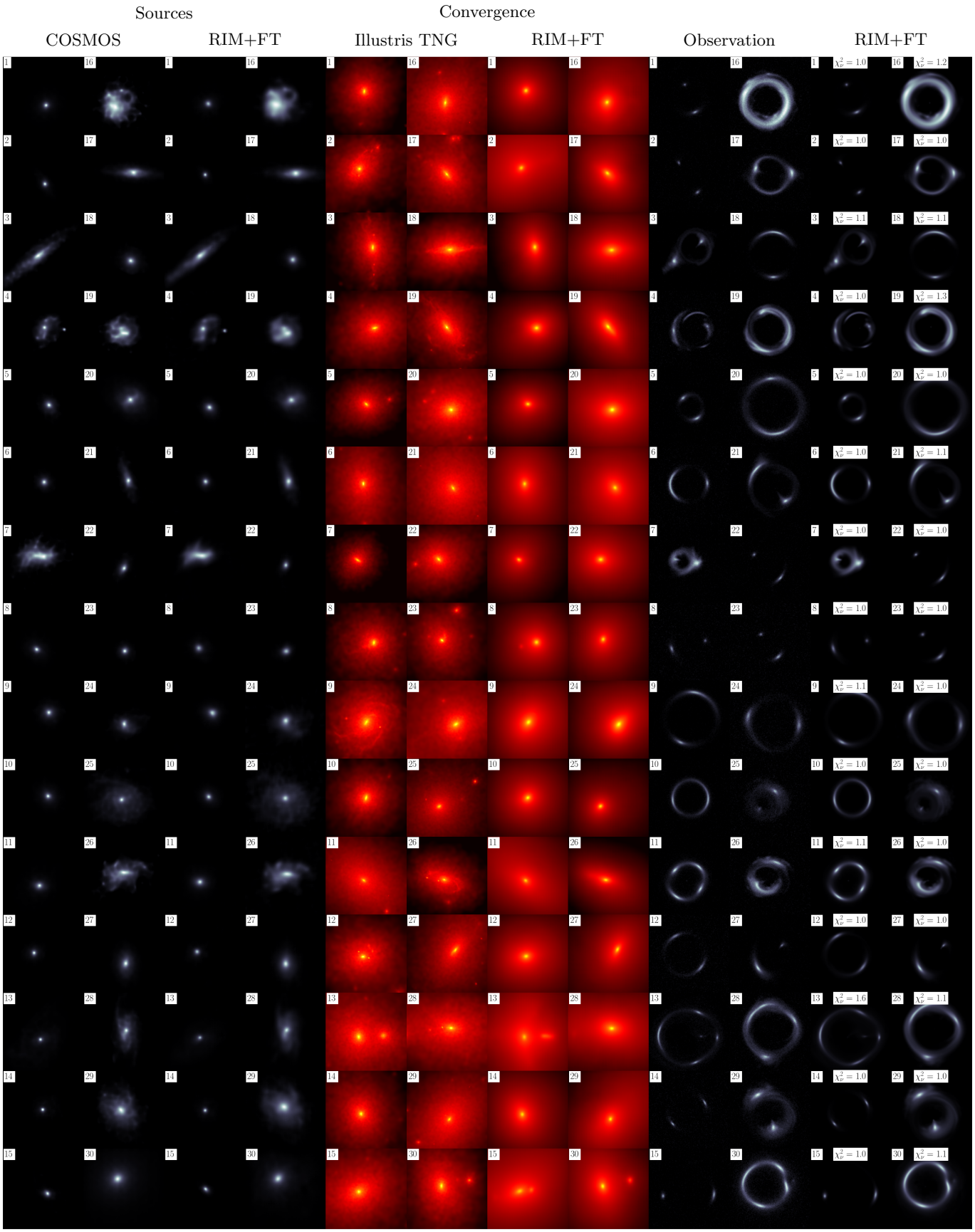

Figure 11. 30 reconstructions taken at random from the test set of 3000 examples simulated from COSMOS and IllustrisTNG data at high SNR. The colorscale are the same as in Figure 7.

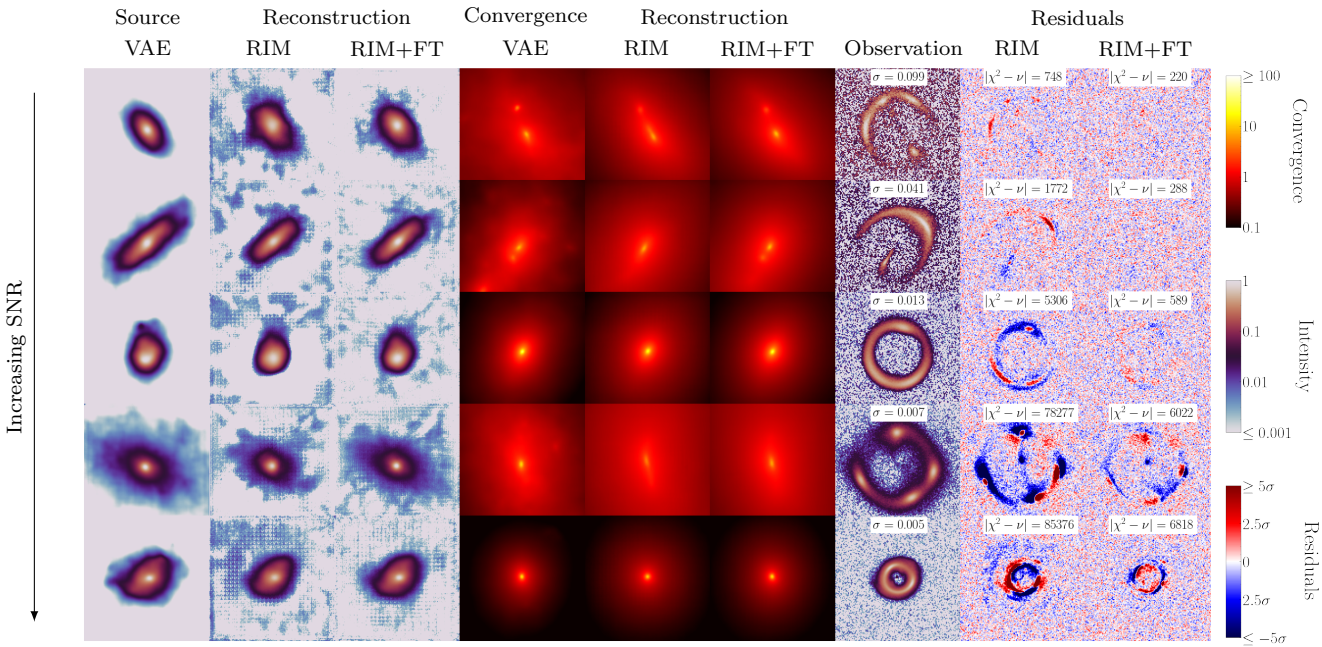

Figure 12. Comparison between baseline and MAP estimate of the weights of the RIM for VAE samples. The examples chosen are representative of the larger sample. From top to bottom, we increase SNR. The first 2 rows have noise level reconstruction, while the last 3 row show significant improvement over the baseline. The intensity color scale is chosen to show the reconstruction down to the third decimal place, where the baseline prediction breaks down.