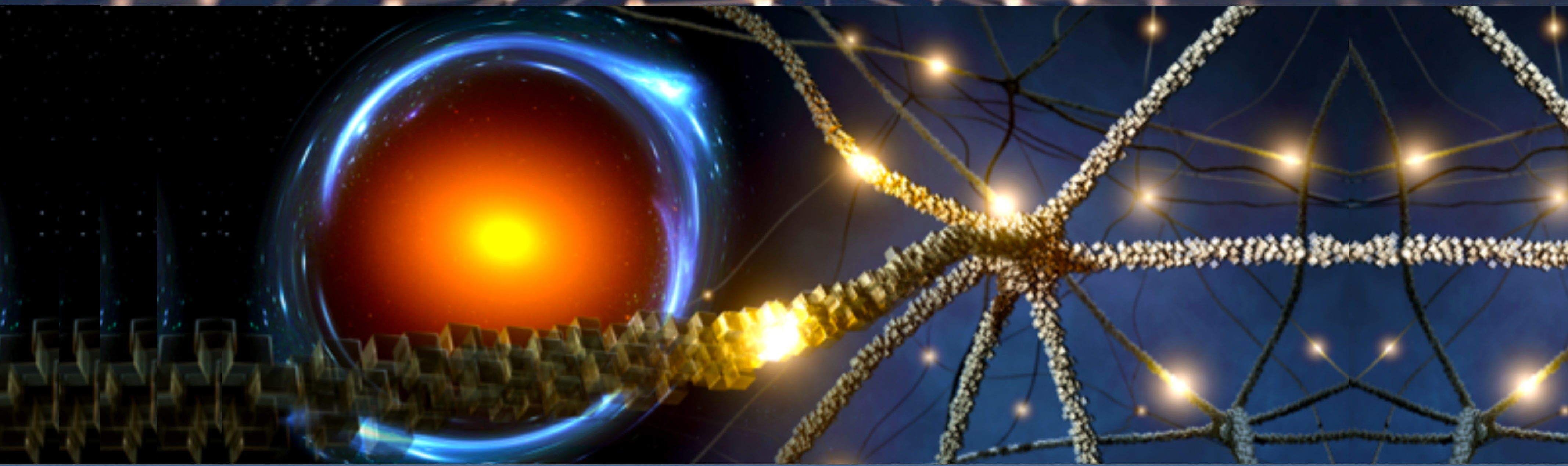
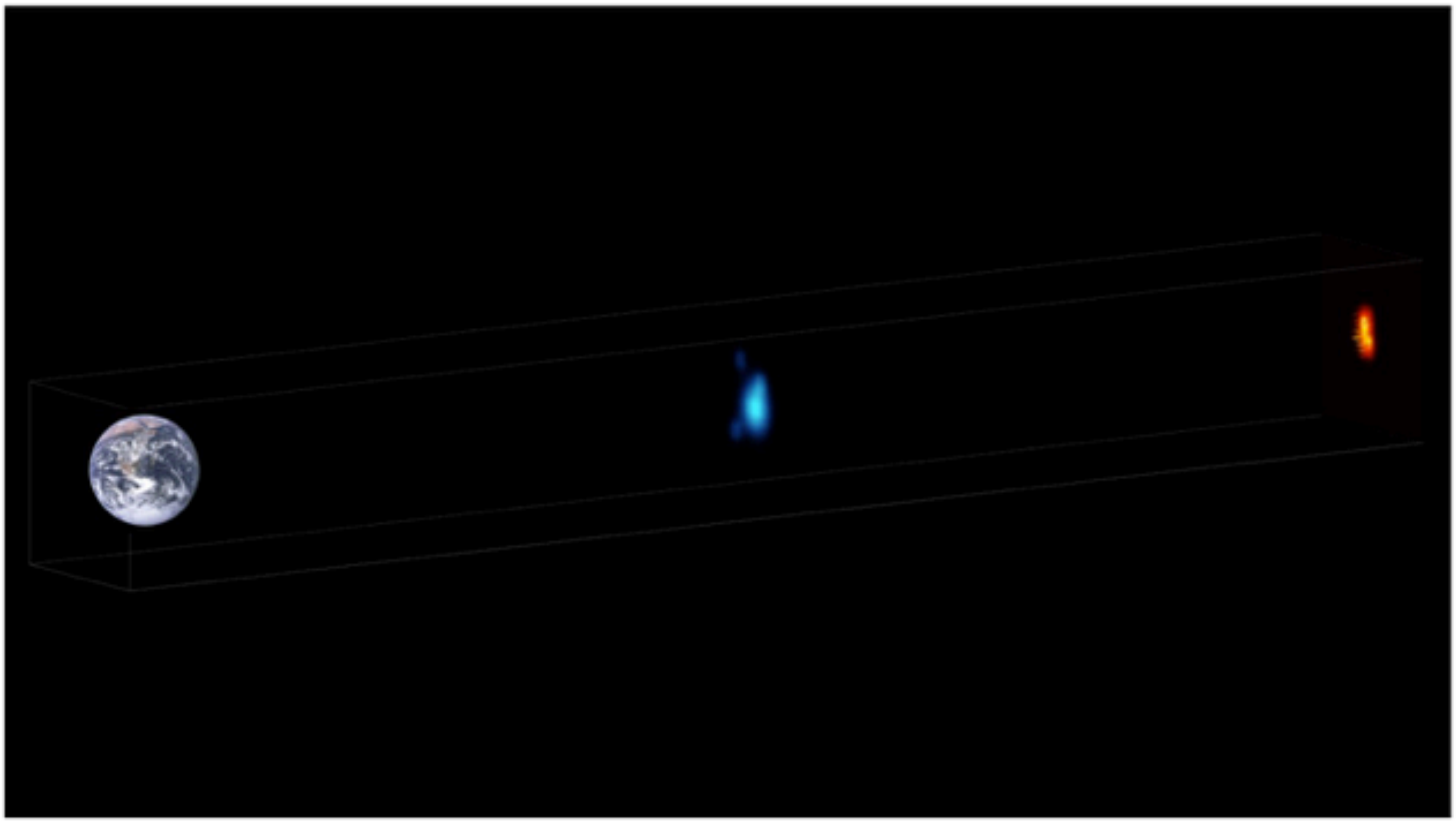
Fast automated analysis of strong gravitational lenses with convolutional neural networks



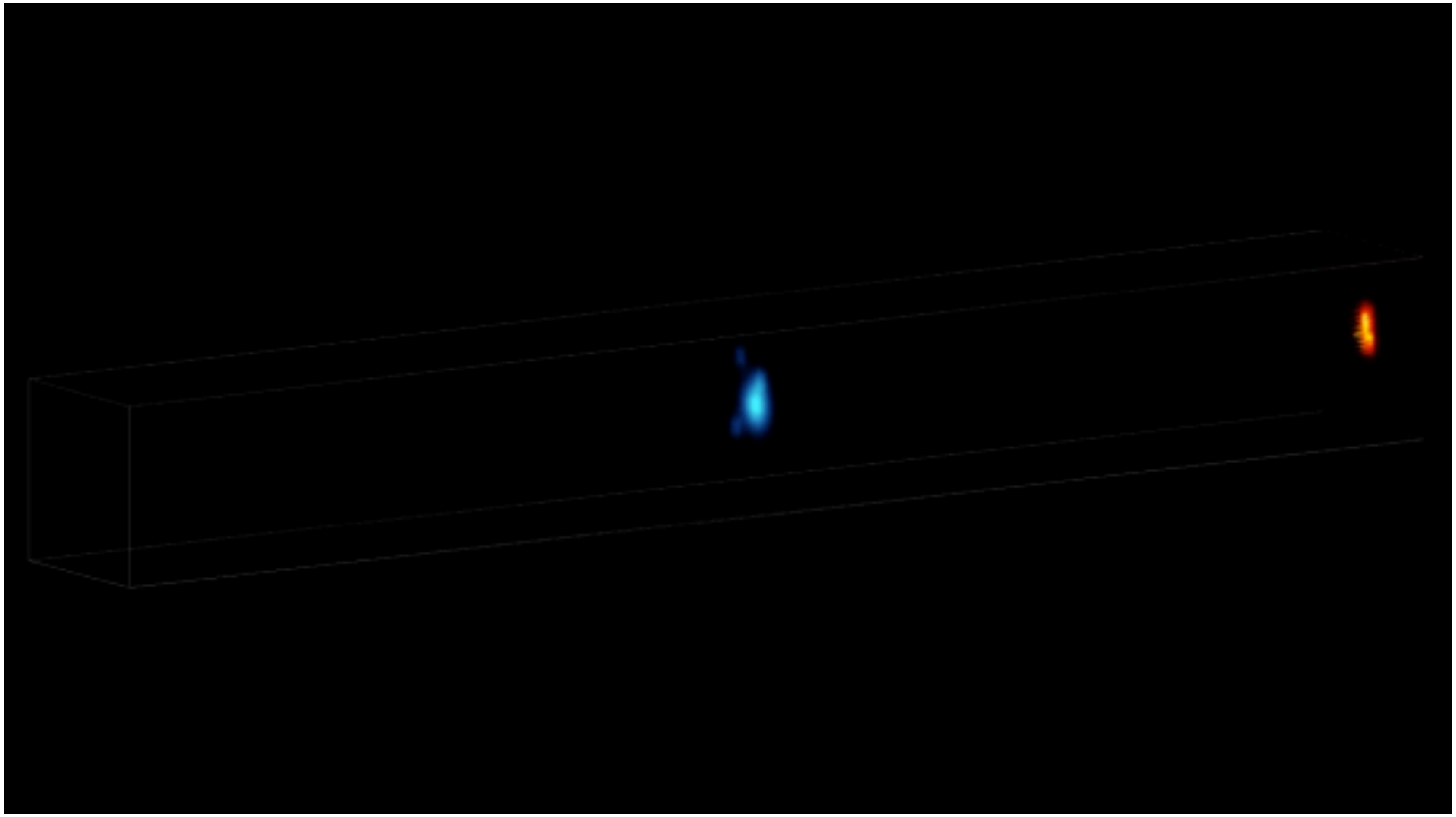
Laurence Perreault Levasseur

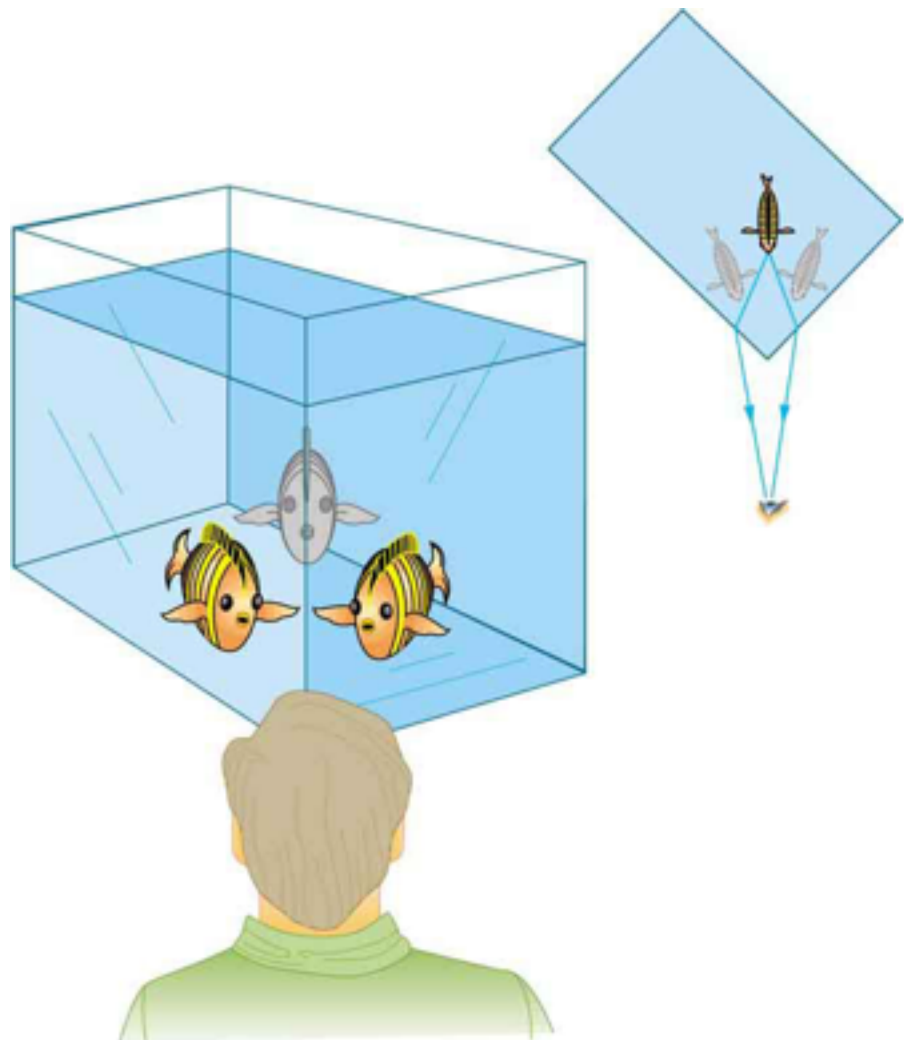
Hezaveh, Perreault Levasseur, Marshall, *Nature* 548, 555–557 (Aug. 2017)
Perreault Levasseur, Hezaveh, Wechsler, ApJL 850, L7-5 (Nov 2017)

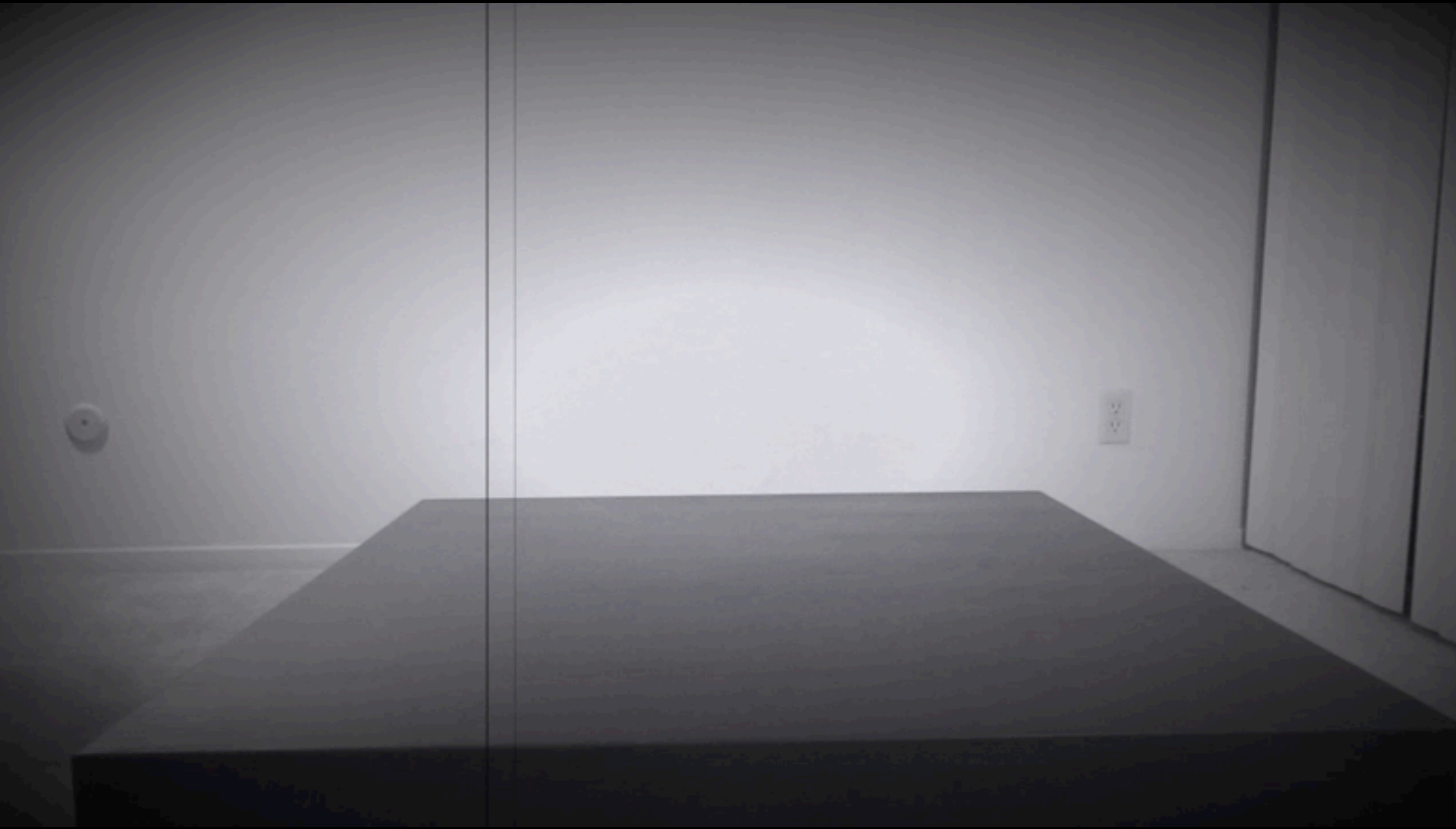
Strong Gravitational Lensing

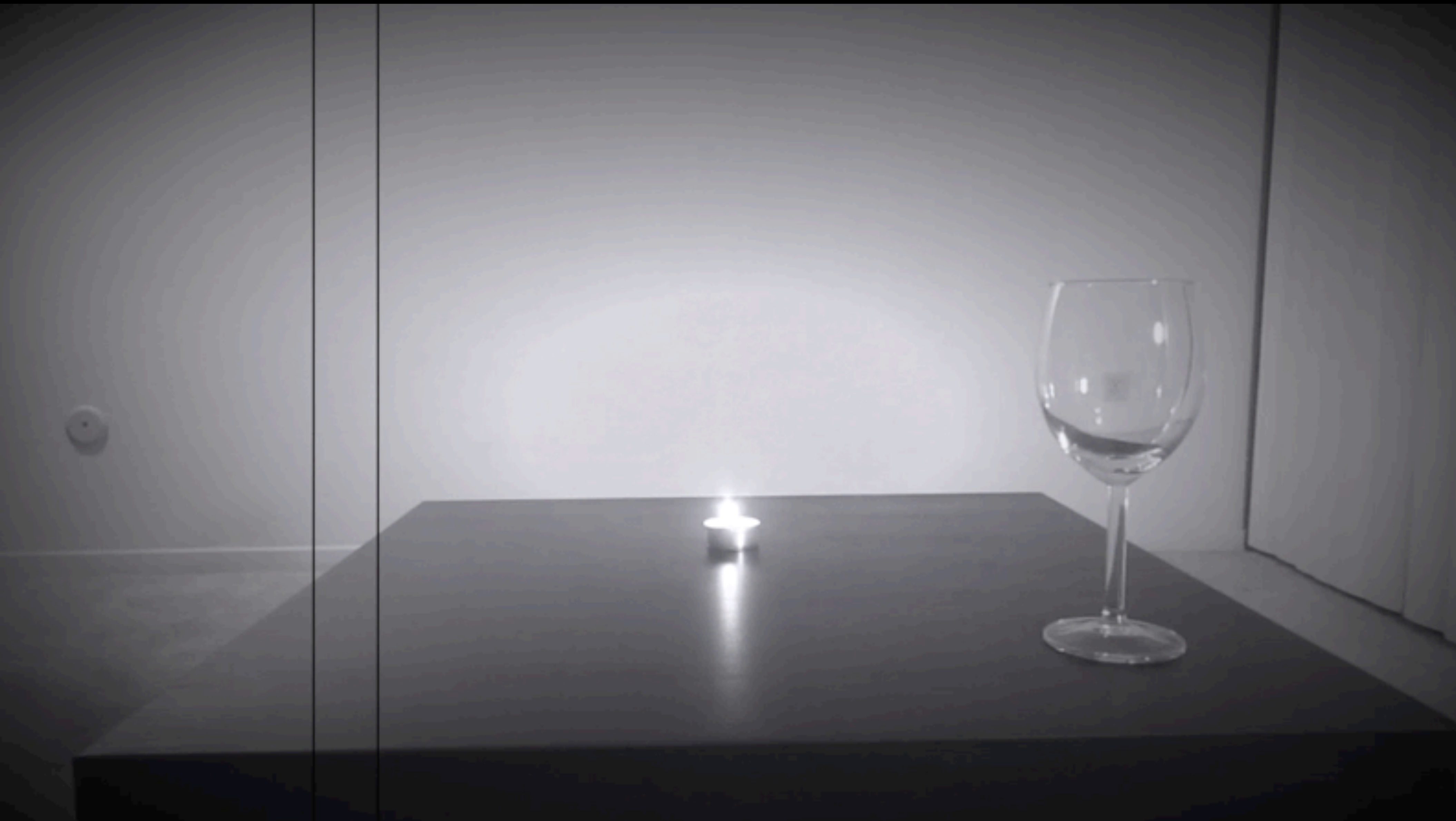


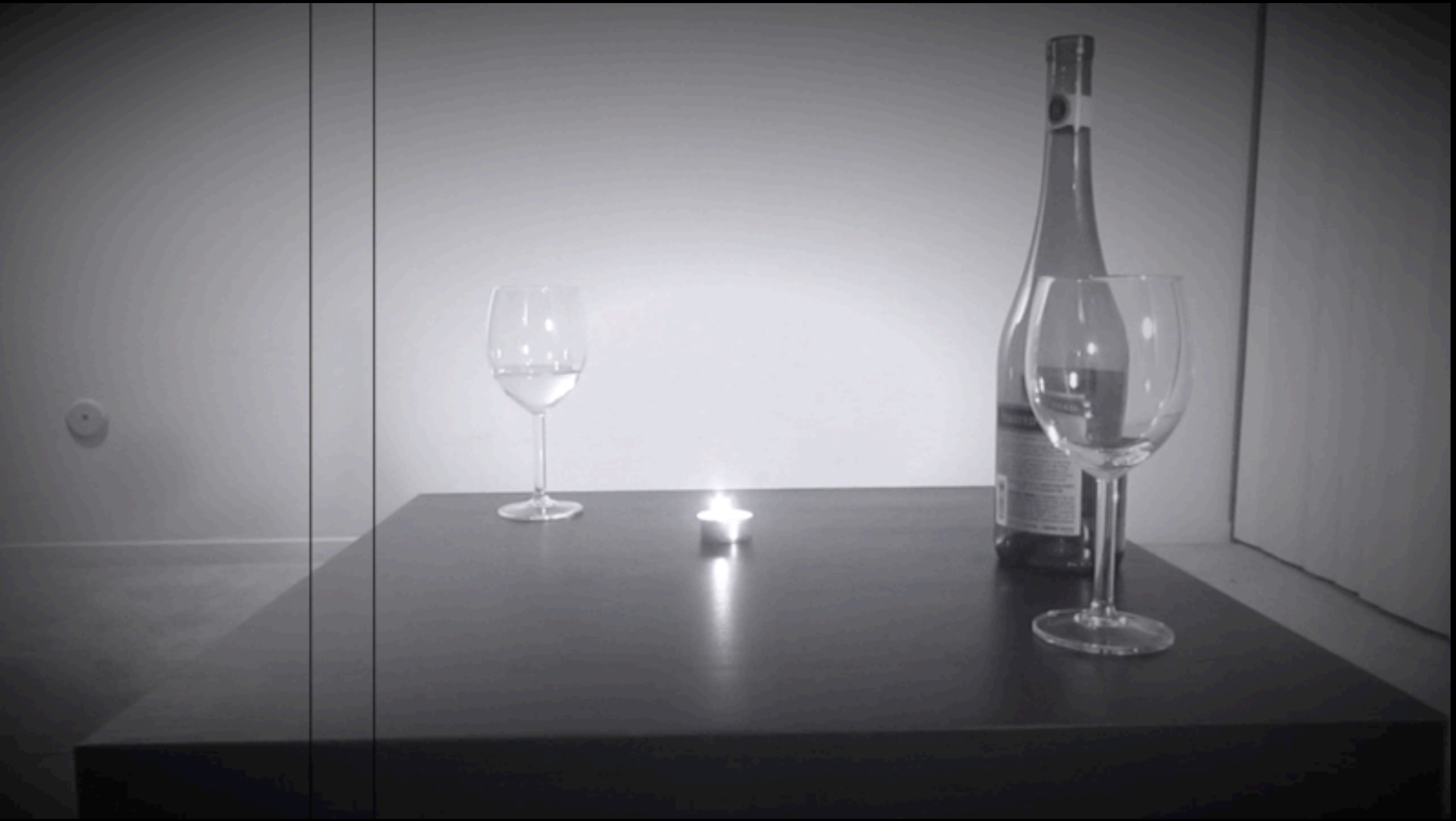
Strong Gravitational Lensing











Science motivations for strong lensing

- **Background source**

Use strong lensing as a **cosmic telescope**.

- **Foreground structure**

Use lensing to probe the **distribution of matter** in the lensing galaxies.

- **Cosmology**

Use time delays to measure $\mathbf{H_0}$

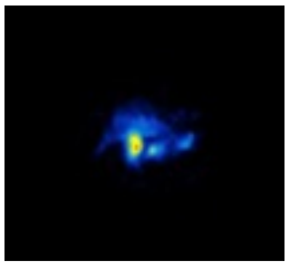
Lens Modeling:
The key to all these sciences

1- How does the background source truly look like? What is the undistorted image?

2- How is matter distributed in the lensing structure?

Lens Modeling

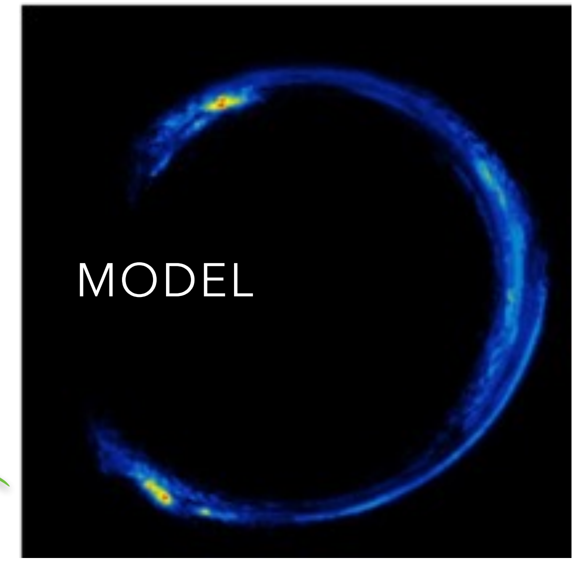
Postulate a Source Morphology (with parameters Ps)

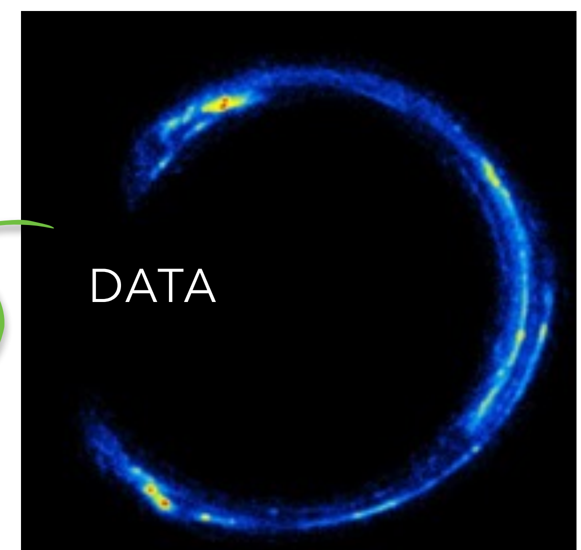


Postulate a Mass distribution in the lens (with parameters Pm)

Ray-tracing simulation

Generate the lensed image of the source



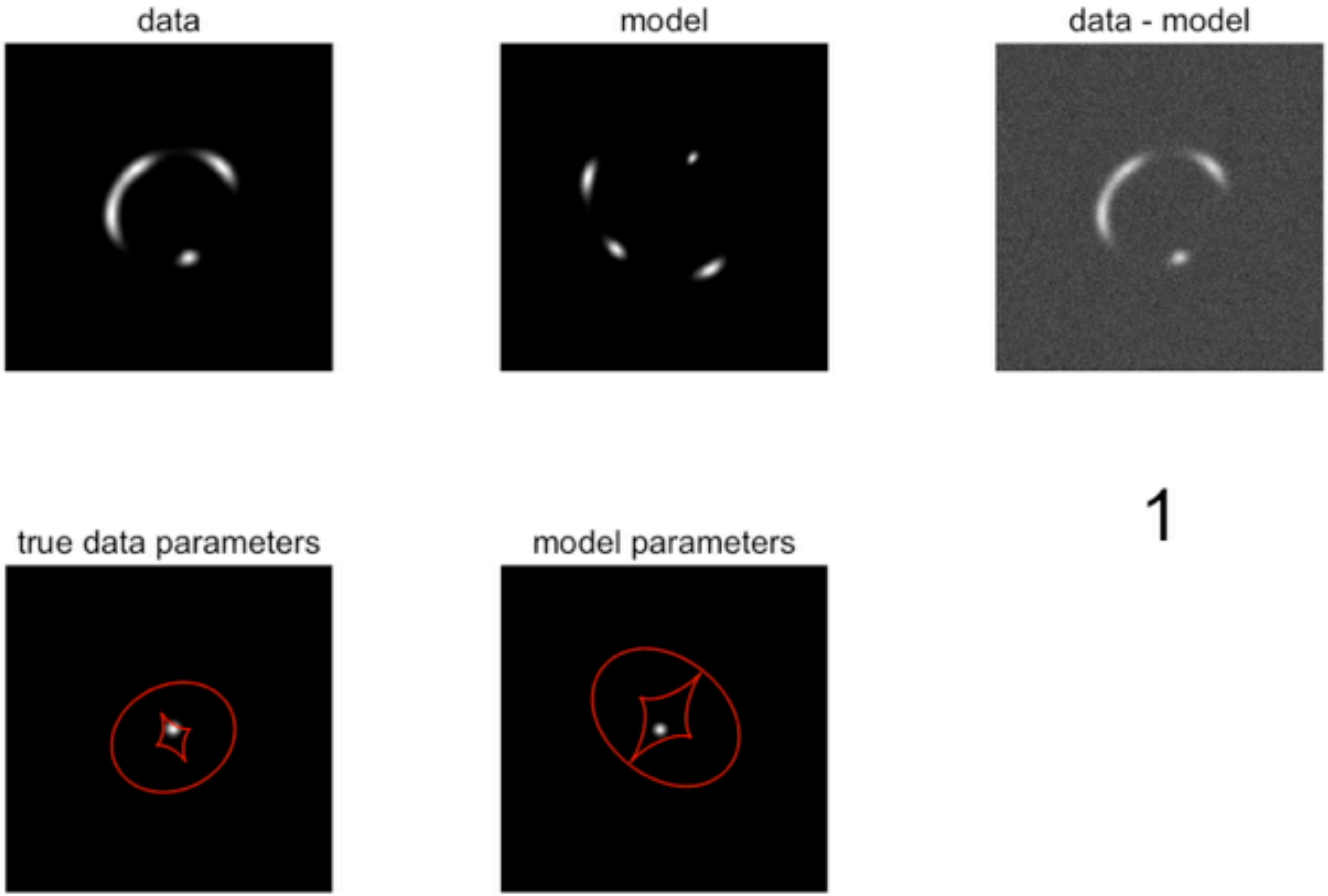


Maximize the likelihood of the model parameters given the data

The ugly: optimization

- Given the data, we need to find the parameters that optimize a goodness-of-fit function, (typically the parameter posterior).
- This is done using optimizers.
- The parameter space is full of crazy-looking local minima: optimizers get stuck regularly.
- Likelihood evaluations are very expensive.
- This makes the process both slow, and in need of constant baby sitting (not automated).

Sad story of a poor optimizer

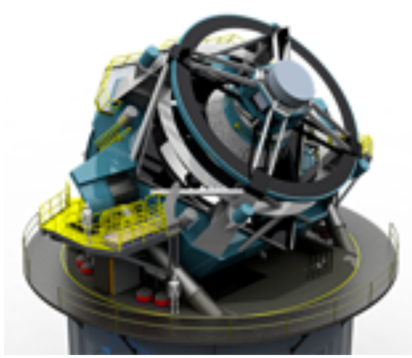


Looking into the future:

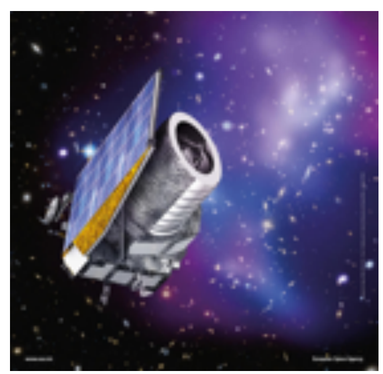
New **Lenses**

For future surveys we find that, assuming Poisson limited lens galaxy subtraction, searches of the DES, LSST, and Euclid data sets should discover **2400**, **120000**, and **170000** galaxy–galaxy strong lenses, respectively

Collett, ApJ. 2015



LSST



Euclid consortium

Looking into the future:

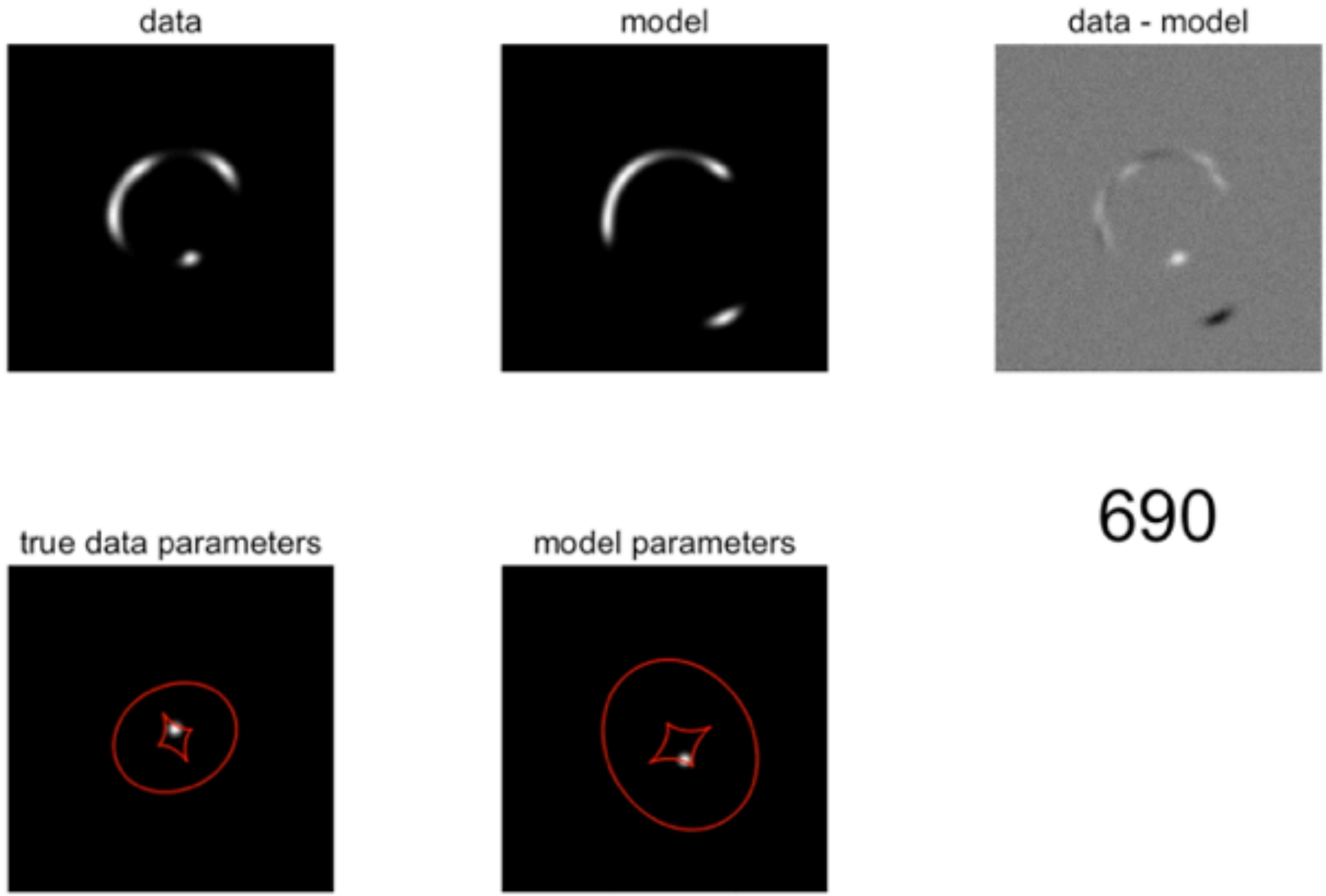
Methods?

How are we going to analyze 170,000 lenses?

- Lens modeling is **very slow**.
- Even a simple lens model can take 2-3 days of human and CPU time, translating to **1,400 years !!!**
- Even if we pay **100 people** to work on this, it'll be **14 years!!!** Old method are simply not feasible.

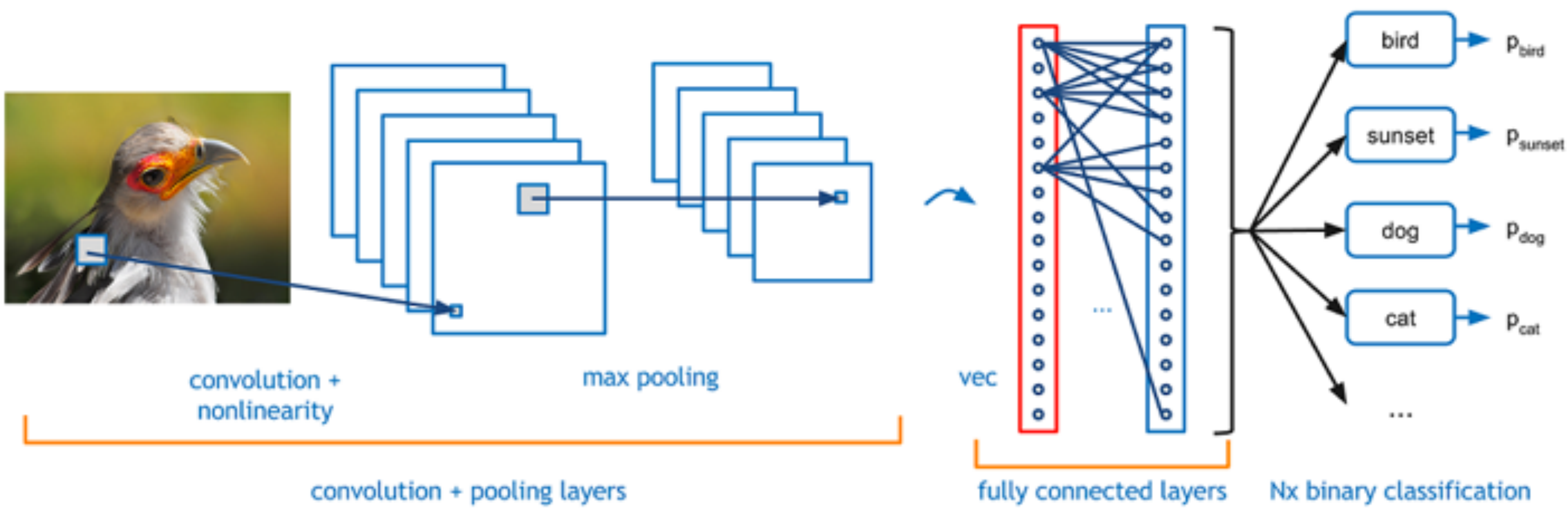
Lens modeling sweatshop of 2022

A bad local minimum is easily recognizable to human eye



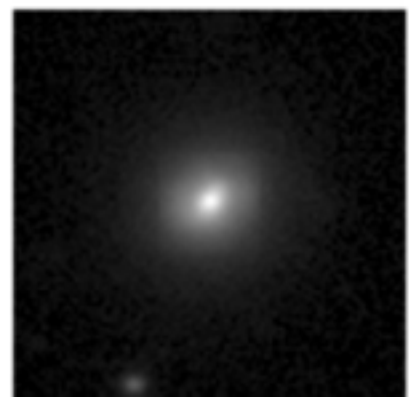
Computer Vision: Convolutional Neural Networks

Commonly used for image recognition and classification



Convolutional Neural Networks: Previously used to find lenses (Classification)

They can be trained to classify images:
Two classes: lenses vs. non-lenses

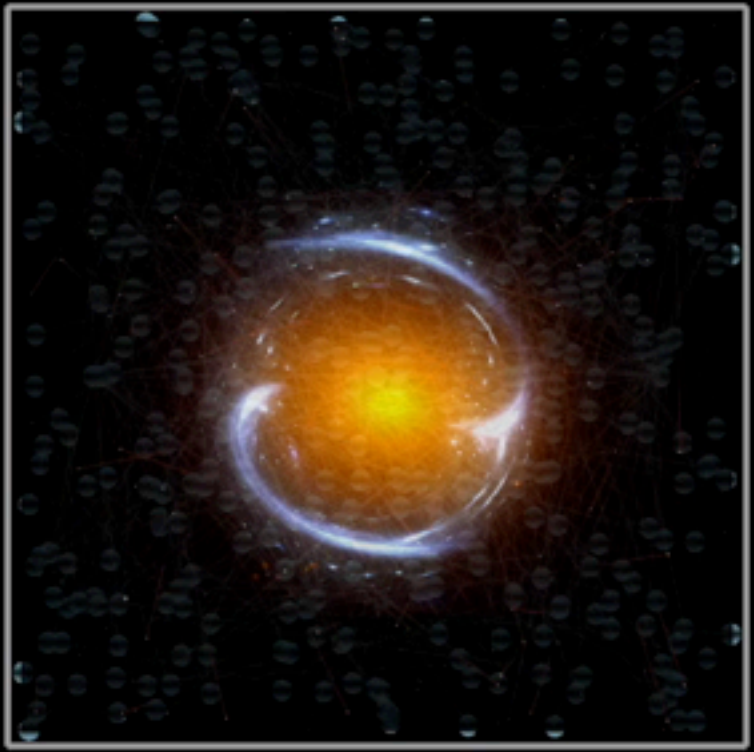


CMU DeepLens: Deep Learning For Automatic Image-based Galaxy-Galaxy Strong Lens Finding

François Lanusse,[1*] Quanbin Ma,[2] Nan Li,[3,4] Thomas E. Collett,[5] Chun-Liang Li,[2] Siamak Ravanbakhsh,[2] Rachel Mandelbaum[1] and Barnabás Póczos[2]
[1] McWilliams Center for Cosmology, Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] High Energy Physics Division, Argonne National Laboratory, Lemont, IL 60439, USA
[4] Department of Astronomy & Astrophysics, The University of Chicago, 5640 South Ellis Avenue, Chicago, IL 60637, USA
[5] Institute of Cosmology and Gravitation, University of Portsmouth, Burnaby Rd, Portsmouth, PO1 3FX, UK

Finding Strong Gravitational Lenses in the Kilo Degree Survey with Convolutional Neural Networks

C. E. Petrillo[1*], C. Tortora[1], S. Chatterjee[1], G. Vernardos[1], L. V. E. Koopmans[1], G. Verdoes Kleijn[1], N. R. Napolitano[2], G. Covone[3], P. Schneider[4], A. Grado[2], J. McFarland[1]
[1] Kapteyn Astronomical Institute, University of Groningen, Postbus 800, 9700 AV, Groningen, The Netherlands
[2] INAF - Osservatorio Astronomico di Capodimonte, Salita Moiariello, 16, 80131 Napoli, Italy
[3] Dipartimento di Scienze Fisiche, Università di Napoli Federico II, Compl. Univ. Monte S. Angelo, 80126 Napoli, Italy
[4] Argelander-Institut für Astronomie, Auf dem Hügel 71, D-53121 Bonn, Germany



Fast automated analysis of strong gravitational lenses with convolutional neural networks

Hezaveh, Perreault Levasseur, Marshall

Nature 548, 555–557, Aug 2017

Training Data

We need a large number of training images. There are only a couple of hundred of gravitational lenses known to date. But we can simulate these images very fast.

The training images need to be as realistic as possible, encompassing all optical effect and noise properties of real telescope images. These include:

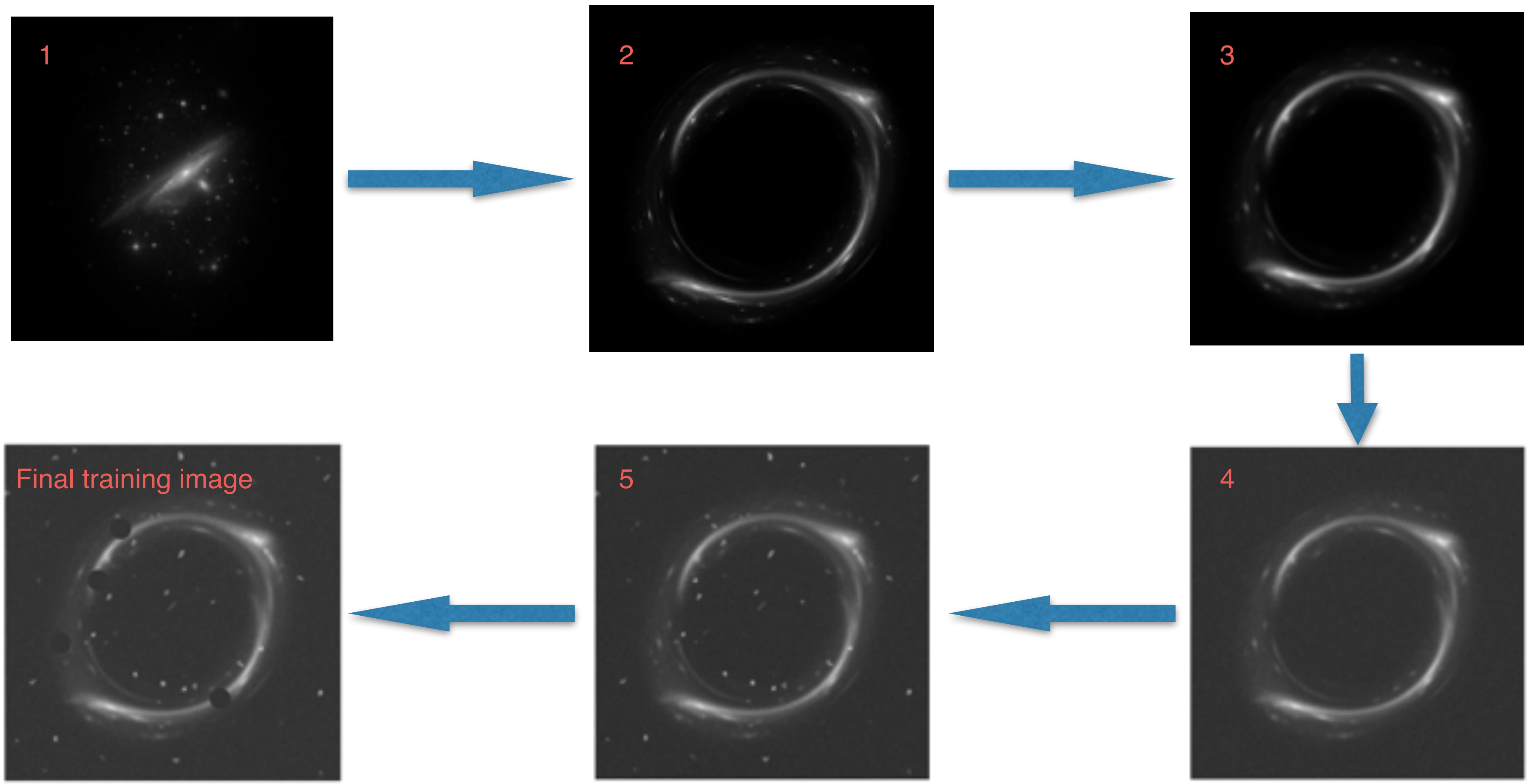
- Realistic images of galaxies
- Optical Blurring (telescope Point Spread Function)
- Addition of Poisson shot noise (discrete photon noise)
- Detector noise
- Cosmic rays, hot pixels, and other artifacts
- Zero bias

Producing the Training Data

Get a real image of a galaxy

Lens it

Blur it with a PSF

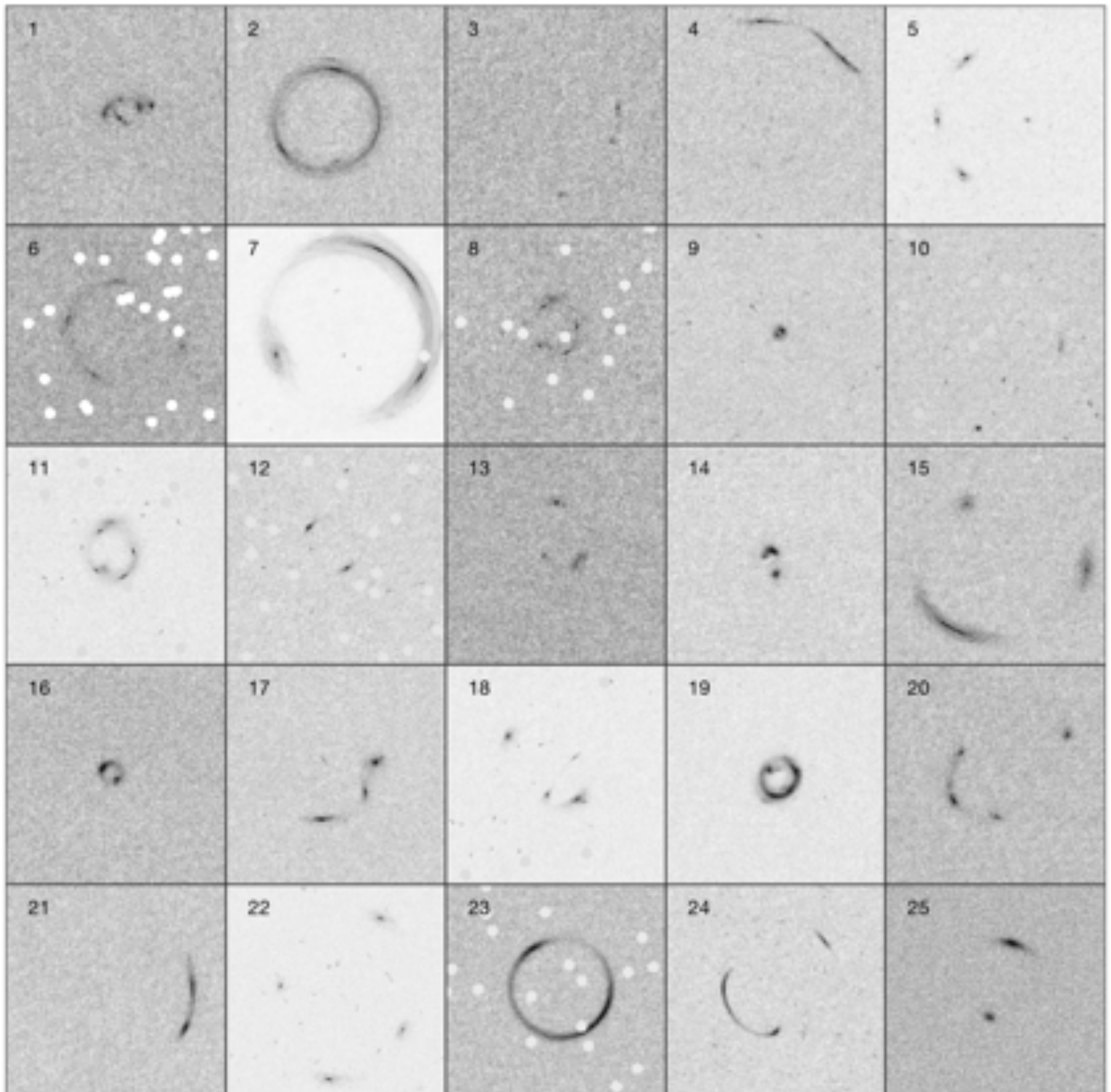


Apply random masks

Add cosmic rays

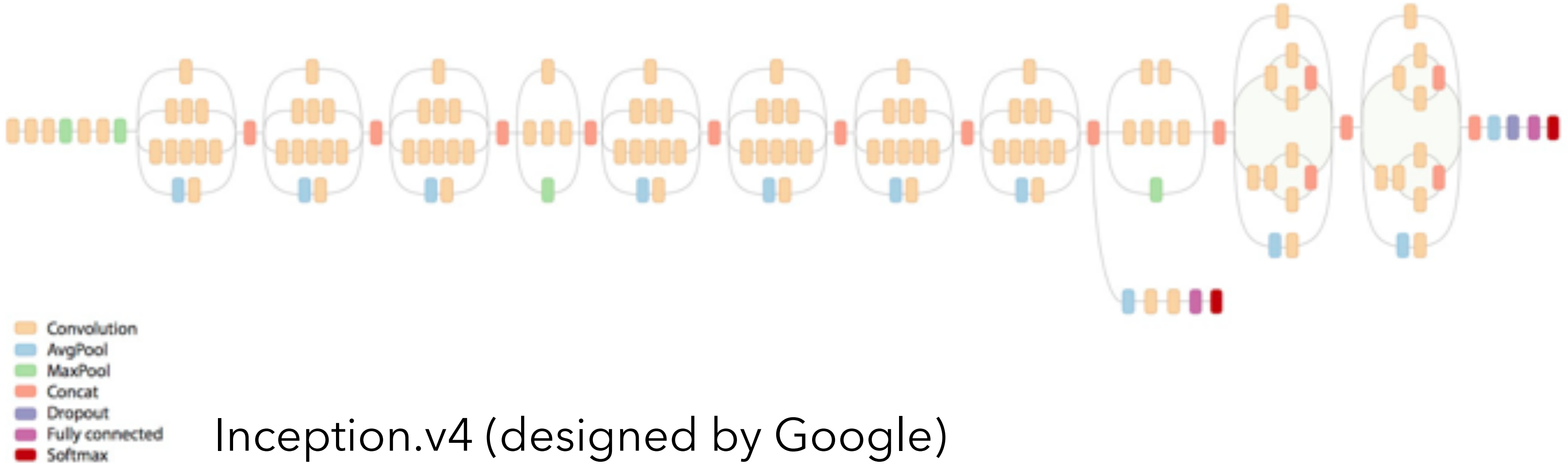
Add noise

EXAMPLES OF SIMULATED DATA



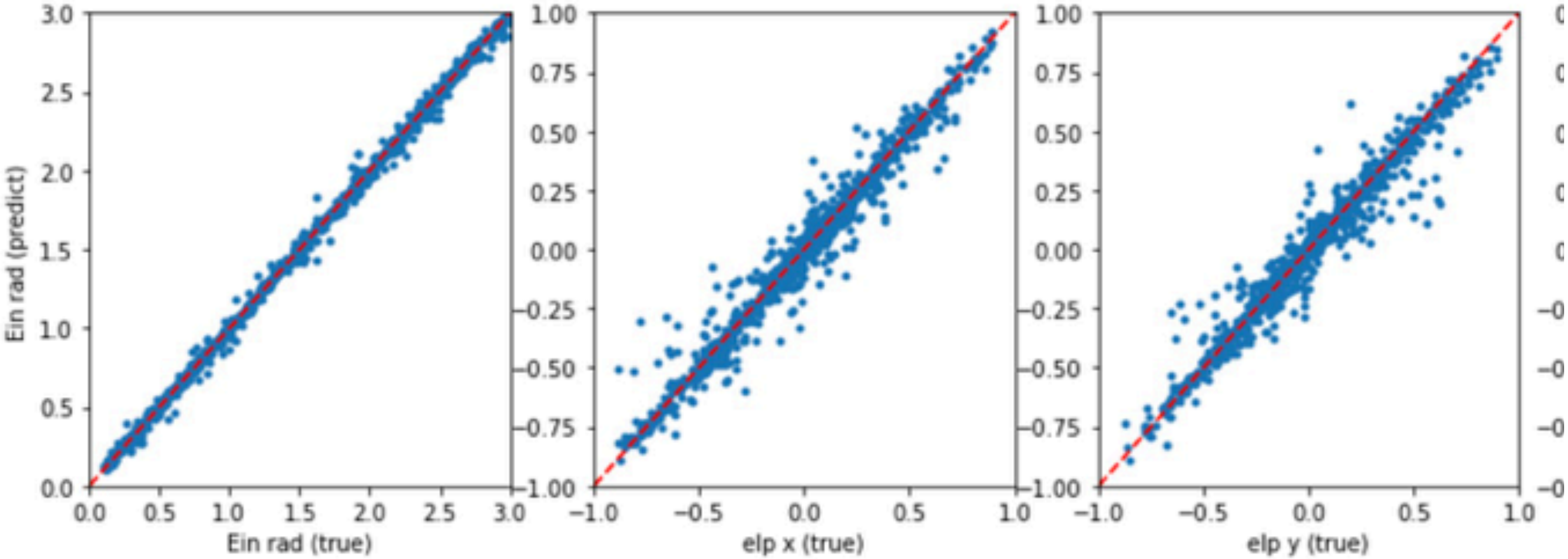
GENERAL INFORMATION ABOUT THE NETWORKS

- Predict the parameters of SIE and shear (5-8 parameters)
- Half a million (simulated) images for training.
- Trained multiple networks: e.g., Inception.v4 (hundreds of layers)
- Training time: About 1-2 day(s) on a single GPU



Inception.v4 (designed by Google)

Recovered parameters for simulated test data

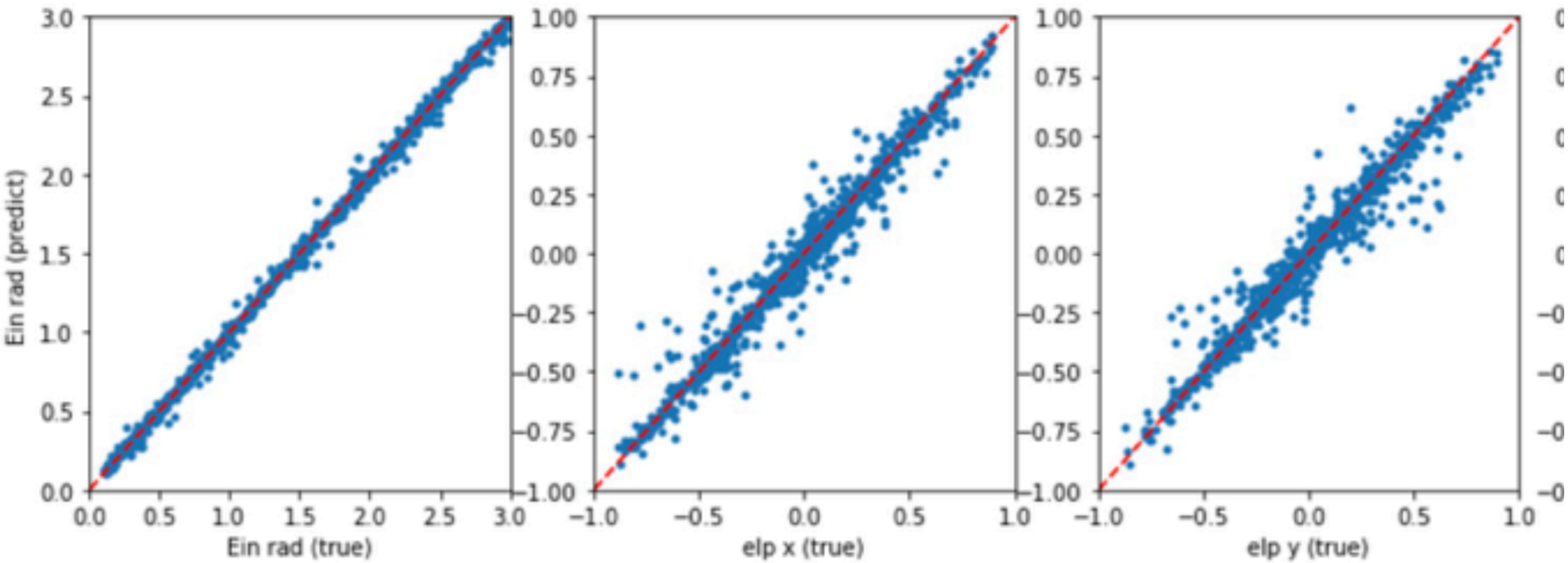


x-axis: true values

y-axis: estimated values

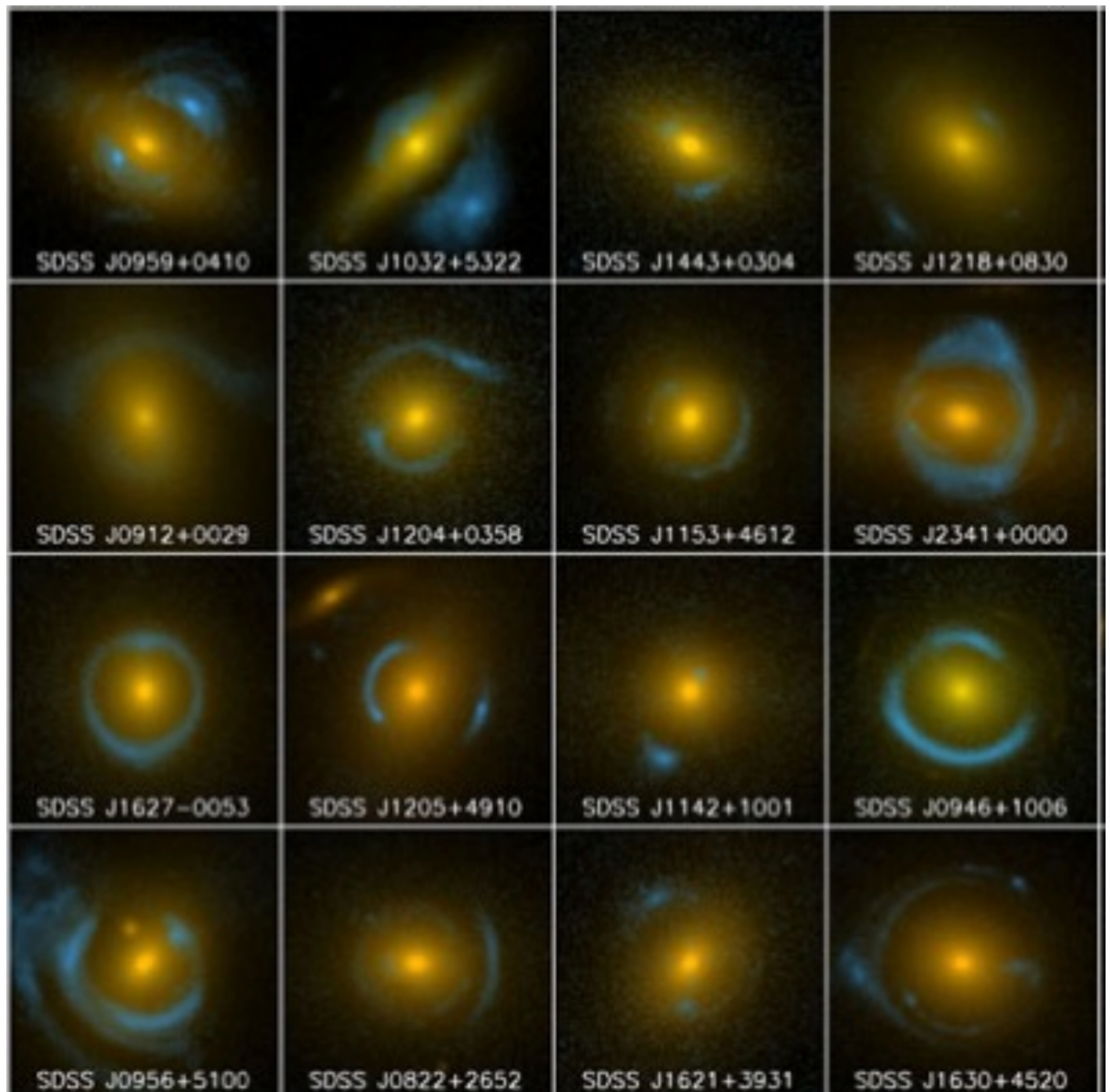
Recovered parameters for simulated test data

10 million times faster than ML lens modeling:
0.01 seconds on a single GPU

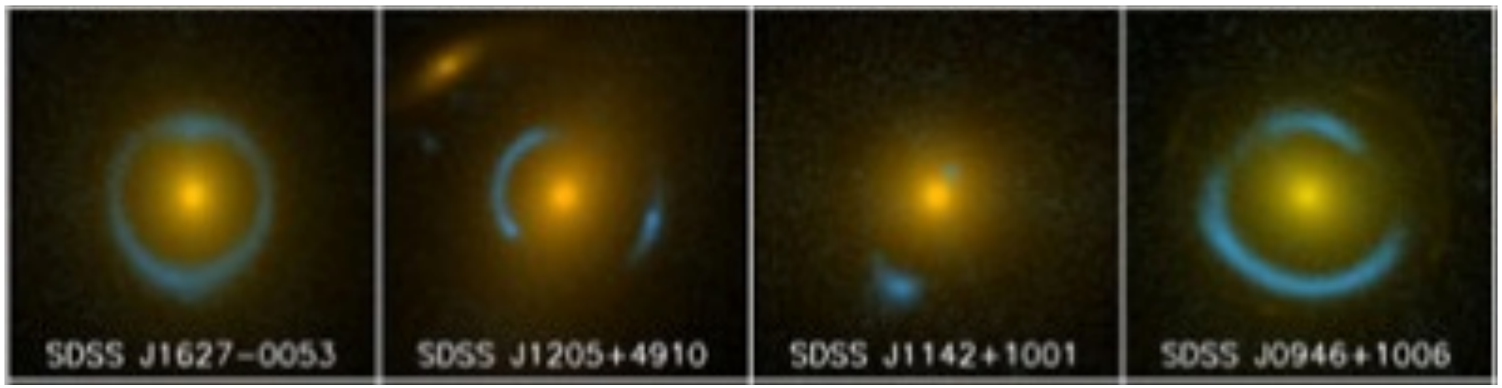


What would take **100 people 14 years** and **12 million CPU hours** can be done in **half an hour on a single GPU**

WHAT ABOUT THE LIGHT OF THE LENSING GALAXY?



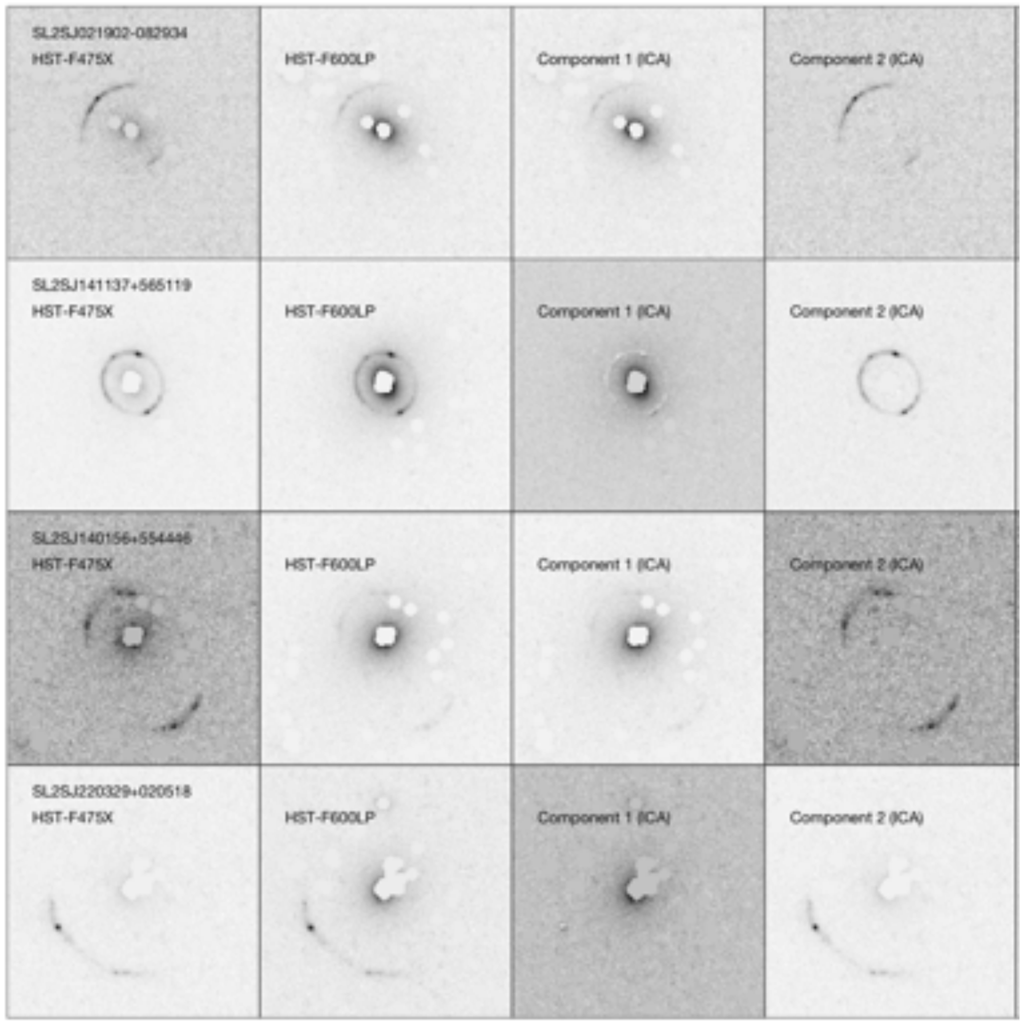
What about the light of the lensing galaxy?



Usually people fit a model (e.g., Sersic) to the light distribution of the lens galaxy and remove it from the data. This is:

1) Time consuming, requiring another non-linear optimization problem
2) Not automated, requiring guesses for starting points, a choice of an appropriate profile, etc.
3) Often leaves high residuals (Galaxies aren't exactly Sersic, or King, etc.)
4) They don't take advantage of color difference in the two sources.

Use another machine learning tool: Independent Component Analysis (ICA)



Hezaveh, Perreault Levasseur, Marshall, *Nature* Aug. 2017

Images of nine systems

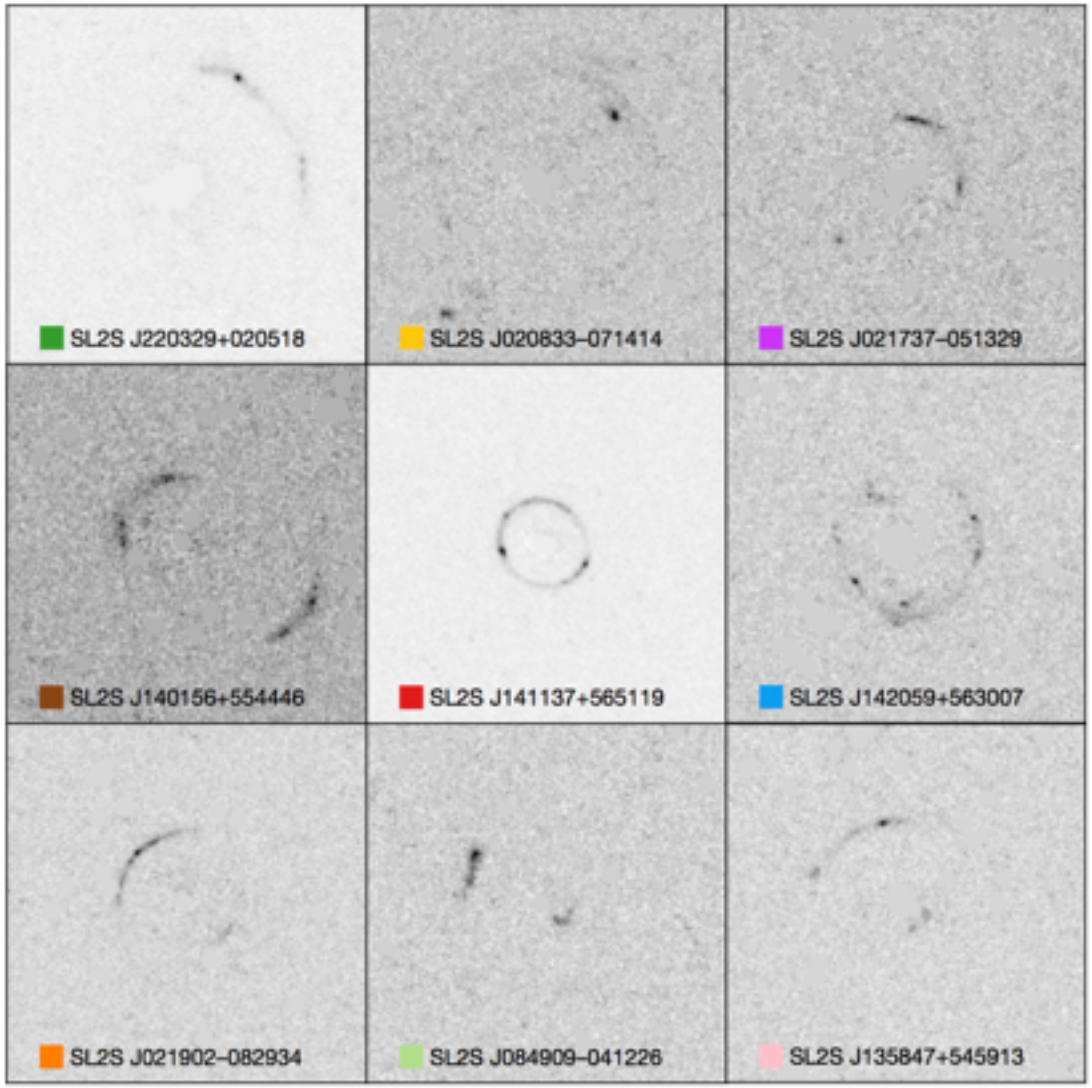


Figure 2 | Hubble Space Telescope images of strongly lensed galaxies from the SL2S survey. These images are used to demonstrate the performance of the network on real data. The light of the lensing galaxies has been removed using independent component analysis of two filters, and circular masks with radii of 0.2″ have been applied to bright cosmic rays and the lens centre. Each panel contains the object name in addition to the data marker used to show its parameters in Fig. 1.

Hezaveh, Perreault Levasseur, Marshall, *Nature* Aug. 2017

RECOVERED PARAMETERS FOR REAL DATA

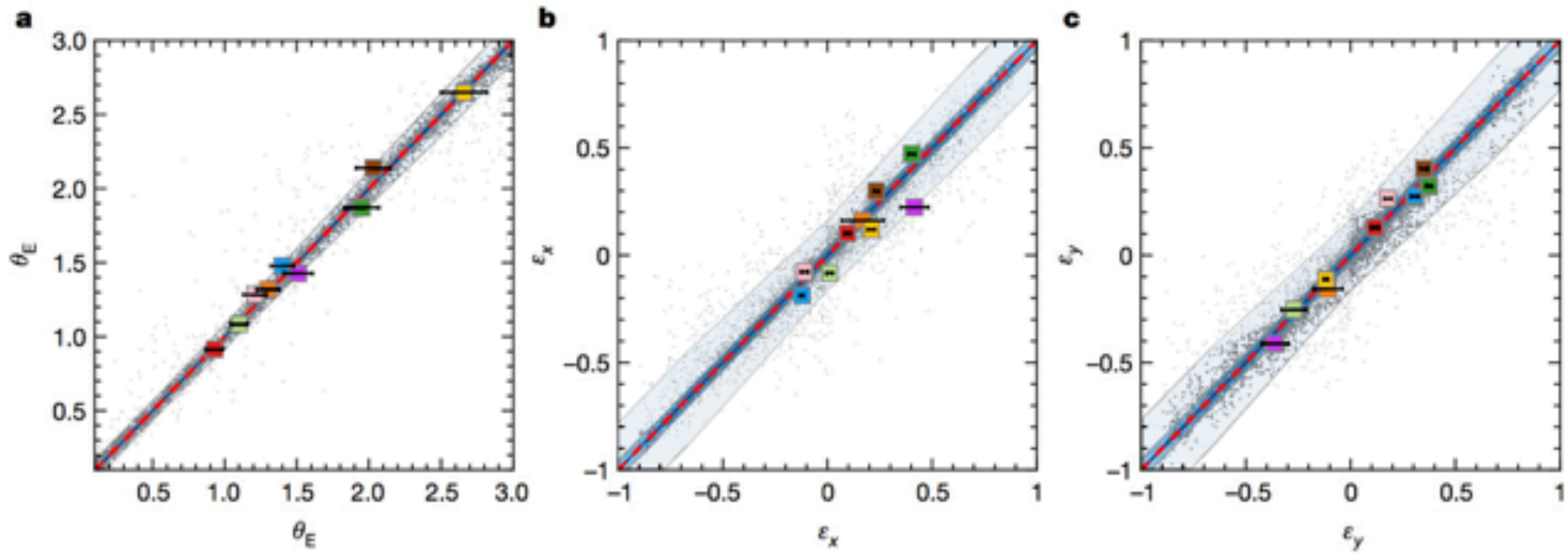


Figure 1 | Comparison of estimated parameters with their true values. The estimated values of the Einstein radius θ_E (**a**) and the x and y components of the complex ellipticity ε_x and ε_y (**b** and **c**) are shown on the y axis; the true values are shown on the x axis. The red dashed line marks the $y = x$ diagonal, on which perfectly recovered parameters should lie. The shaded blue areas represent the 68% and 95% intervals of the parameters recovered from a test set that the network has not been trained on. The small grey dots show the parameters of 10,000 test samples. The coloured data points and their error bars (95% confidence) correspond to real HST images of gravitational lenses, with the true parameters set to previously published values[17].

Hezaveh, Perreault Levasseur, Marshall, *Nature* Aug. 2017

WHAT ARE THE UNCERTAINTIES OF THE OUTPUT PARAMETERS?

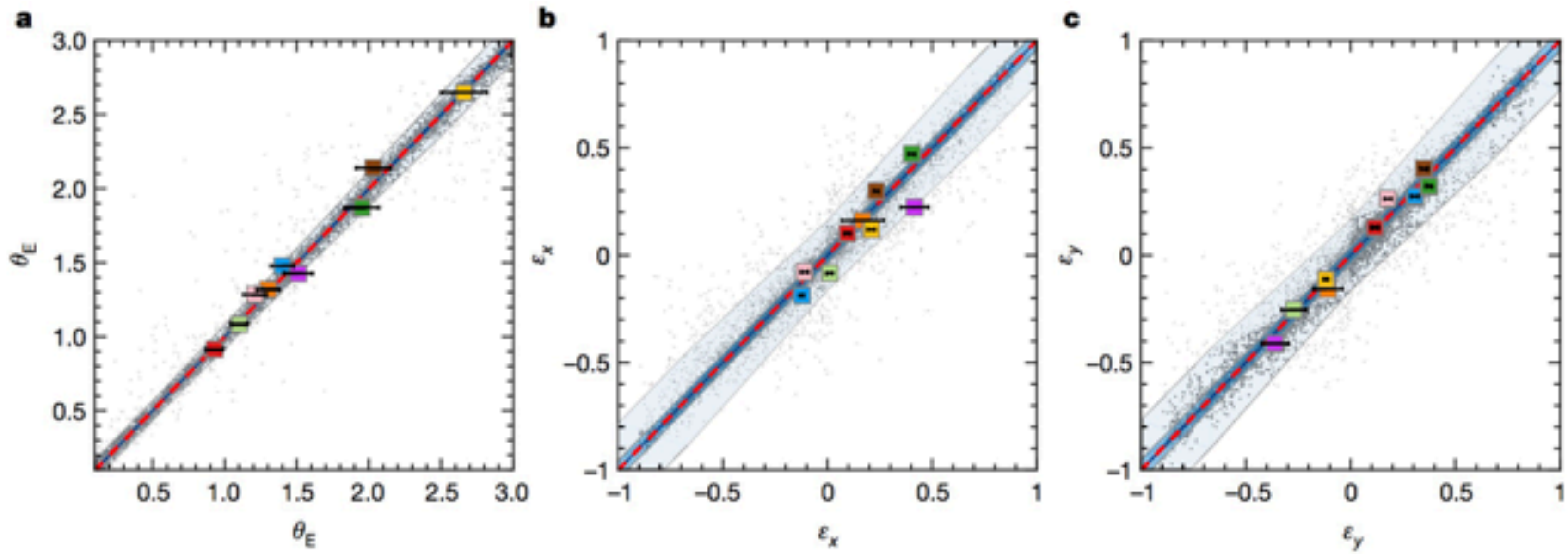


Figure 1 | Comparison of estimated parameters with their true values. The estimated values of the Einstein radius θ_E (**a**) and the x and y components of the complex ellipticity ε_x and ε_y (**b** and **c**) are shown on the y axis; the true values are shown on the x axis. The red dashed line marks the $y = x$ diagonal, on which perfectly recovered parameters should lie. The shaded blue areas represent the 68% and 95% intervals of the parameters recovered from a test set that the network has not been trained on. The small grey dots show the parameters of 10,000 test samples. The coloured data points and their error bars (95% confidence) correspond to real HST images of gravitational lenses, with the true parameters set to previously published values[17].

Hezaveh, Perreault Levasseur, Marshall, *Nature* Aug. 2017

What are the uncertainties of the output parameters?

Sources of errors in the predictions:

1- **Aleatoric**.
Inherent corruptions to the input data: noise, PSF blurring, etc.

2 -**Epistemic**.
Errors made by the networks: these could be due to insufficient training, network architecture, etc.

WHAT IS THE LOG-LIKELIHOOD OF THE NETWORK OUTPUT, $\mathcal{L}(\mathbf{y}_n, \hat{\mathbf{y}}_n(\mathbf{x}_n, \omega))$?

We approximate the likelihood with an analytic distribution.

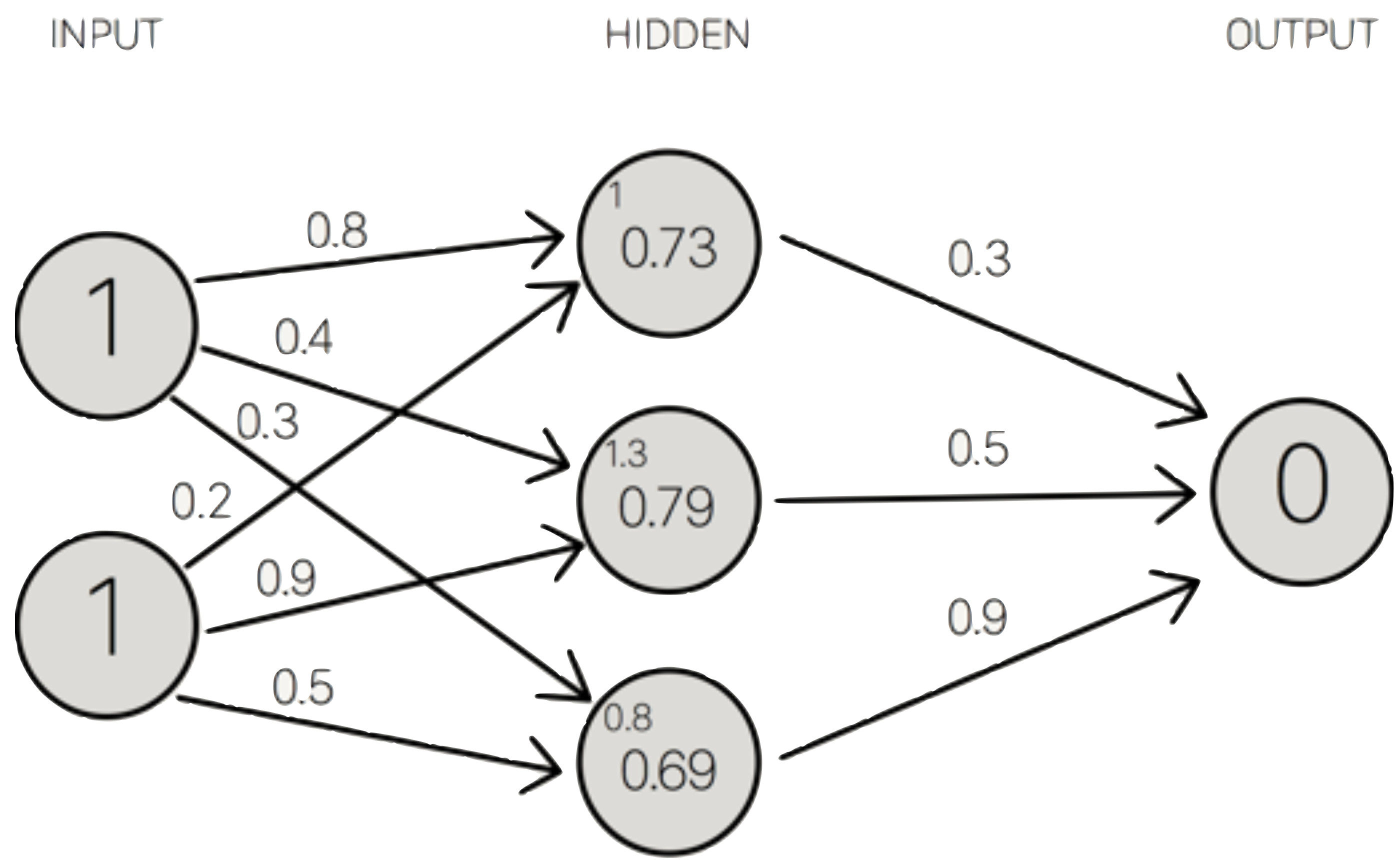
Assume Gaussian:

$$\mathcal{L}(\mathbf{y}_n, \hat{\mathbf{y}}_n(\mathbf{x}_n, \omega)) \propto \sum_k \frac{-1}{2\sigma_k^2} ||y_{n,k} - \hat{y}_{n,k}(\mathbf{x}_n, \omega)||^2 - \frac{1}{2} \log \sigma_k^2$$

Epistemic Uncertainties

Standard Neural Networks:

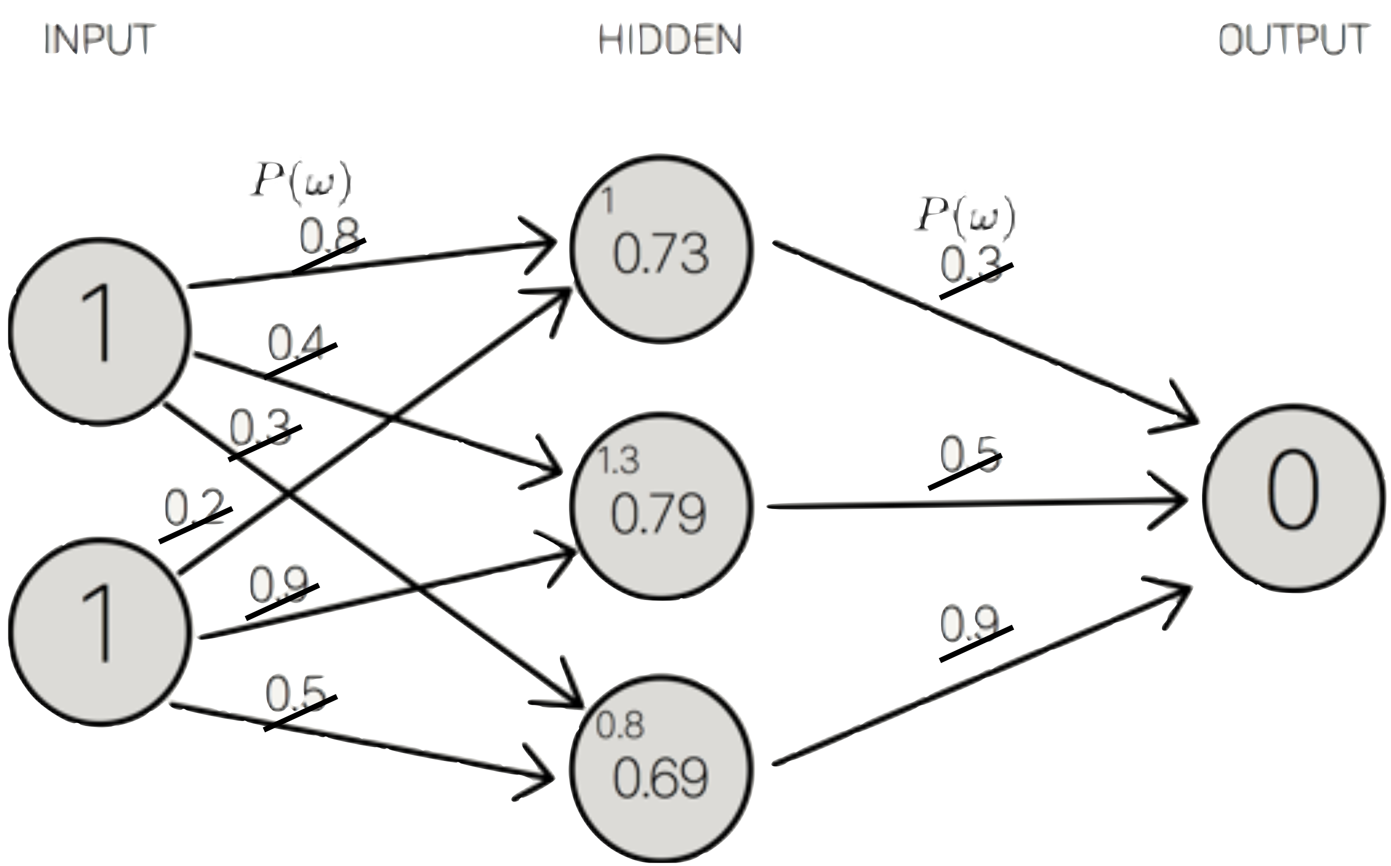
Weight have fixed, deterministic values



Epistemic Uncertainties

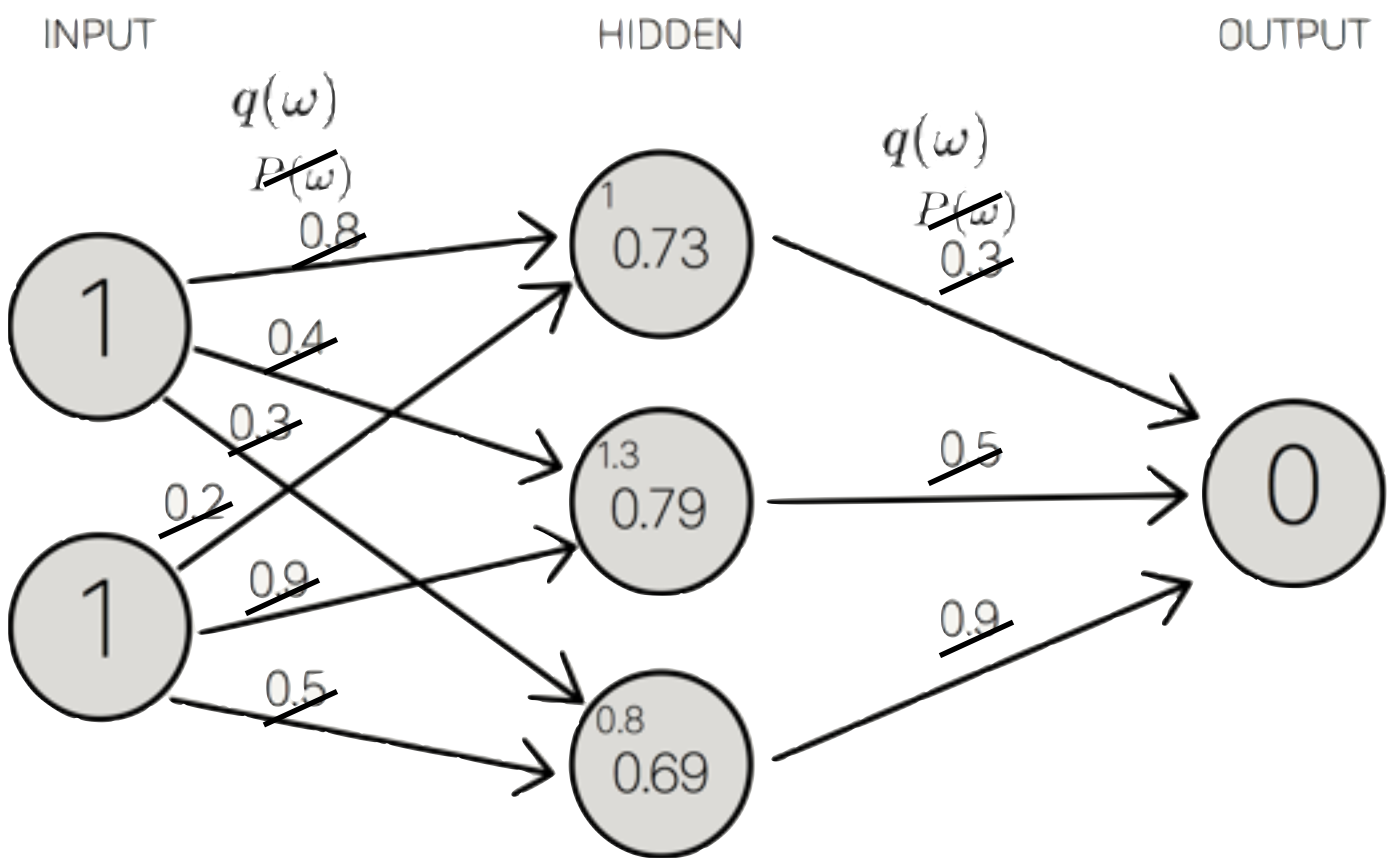
Bayesian Neural Networks:

Instead of fix values, weights are defined by probability distributions



VARIATIONAL INFERENCE

REPLACE $P(\omega)$ BY A DISTRIBUTION WITH A SIMPLE ANALYTIC FORM, $q(\omega)$, (E.G., A GAUSSIAN).



RECAP

1- PLACE DROPOUT BEFORE EVERY WEIGHT LAYER.

2- TRAIN WITH DROPOUT, OPTIMIZING THE LOG LIKELIHOOD

$$\mathcal{L}(\mathbf{y}_n, \hat{\mathbf{y}}_n(\mathbf{x}_n, \omega)) \propto \sum_k \frac{-1}{2\sigma_k^2} ||y_{n,k} - \hat{y}_{n,k}(\mathbf{x}_n, \omega)||^2 - \frac{1}{2} \log \sigma_k^2$$

3- AT TEST TIME, KEEP DROPOUT ON. PERFORM MONTE CARLO DROPOUT: INPUT THE DATA MULTIPLE TIMES, PERFORM DROPOUT AND COLLECT THE OUTPUTS.

4- ADD YOUR ALEATORIC UNCERTAINTY (THE SIGMA ABOVE) TO THE SAMPLE.

5- DONE

EXAMPLE

UNCERTAINTIES ON THE MAGNIFICATION OF LENSES

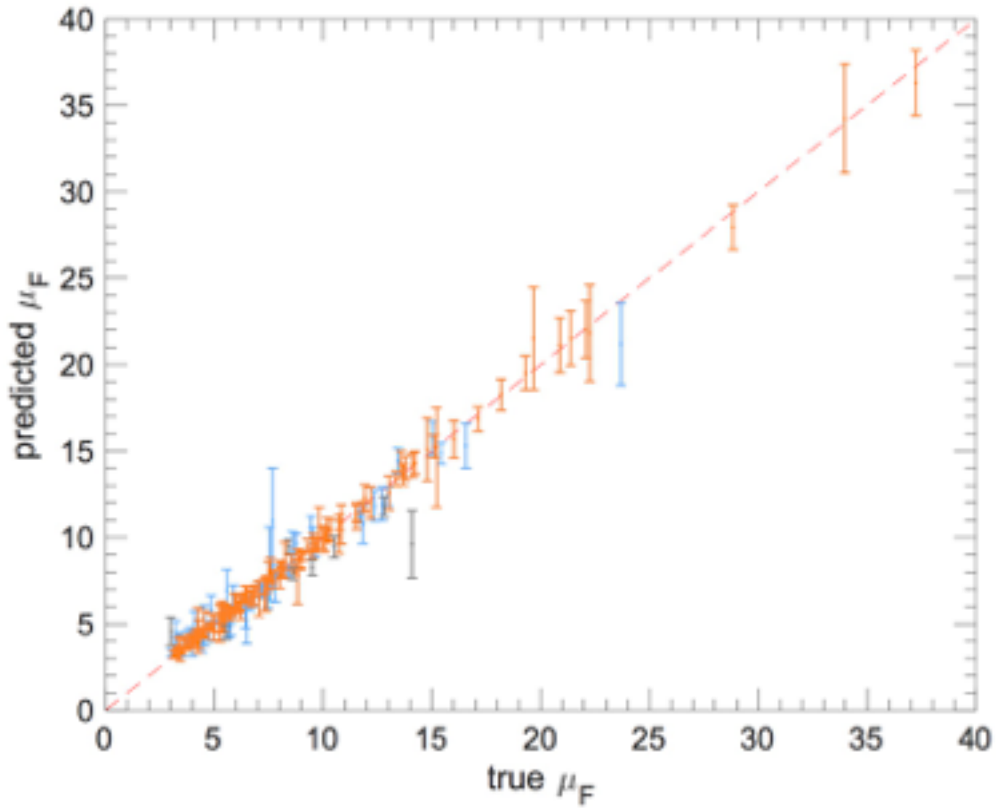
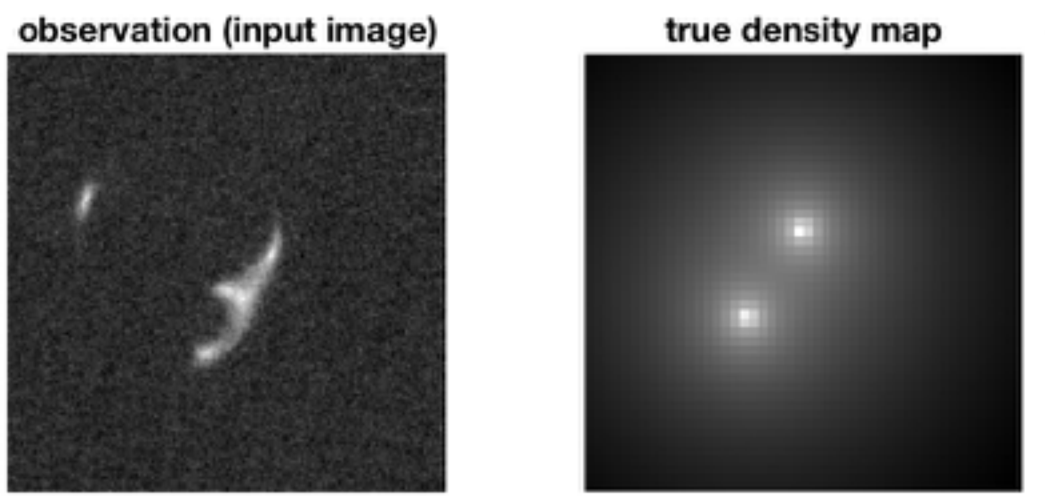


Figure 1. Predicted 68.3% uncertainties for lensing flux magnification, μ_F, as a function of the true value of this parameter. The orange, blue, and black correspond to examples where the true values fall within the 68.3, 95.5, and 99.7% confidence intervals respectively.

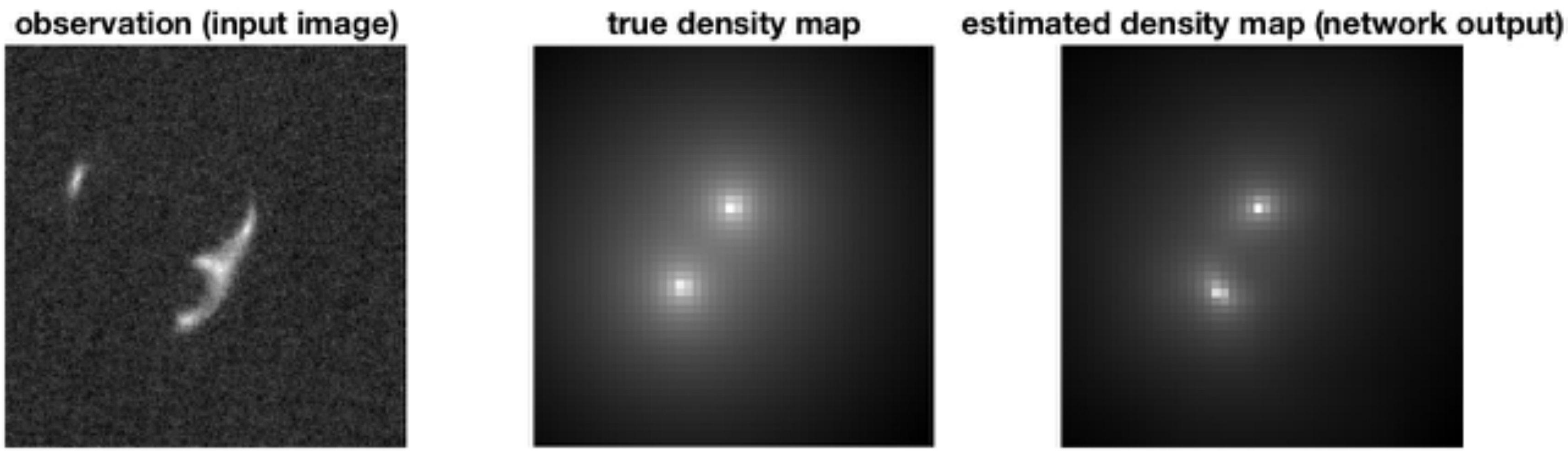
Perreault Levasseur, Hezaveh, Wechsler, ApJL, Nov 2017

ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?

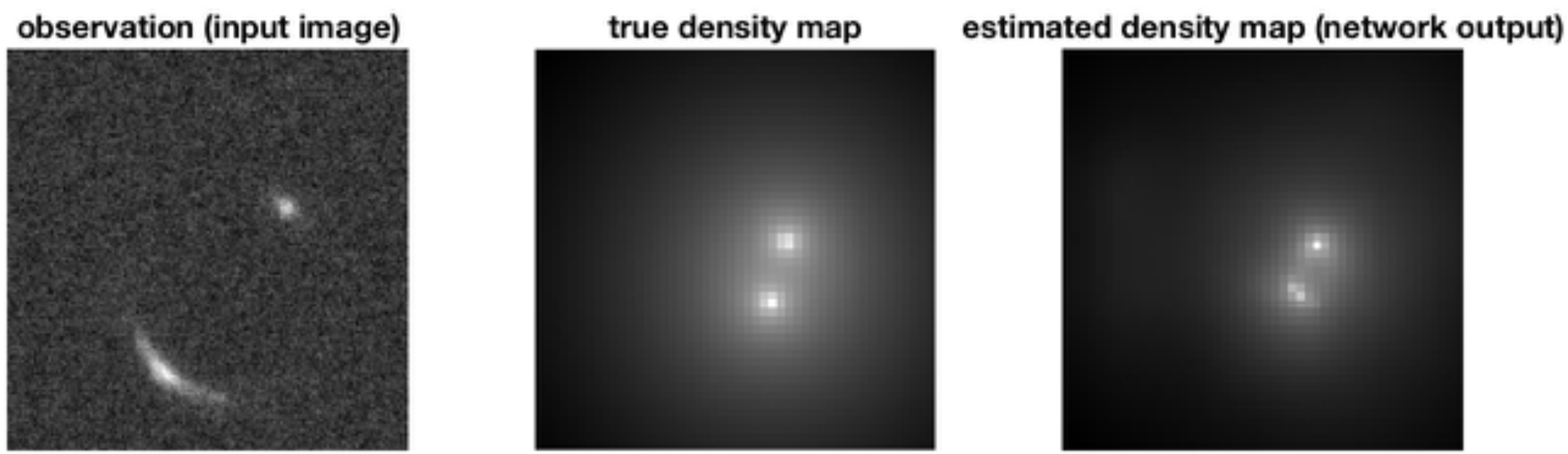
ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?



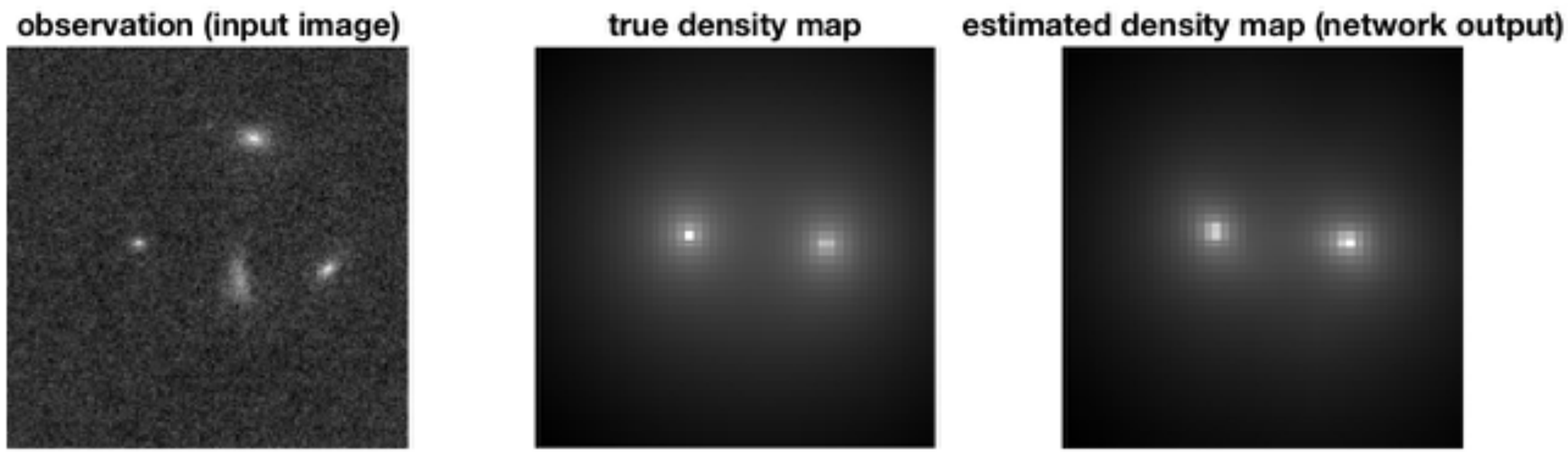
ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?



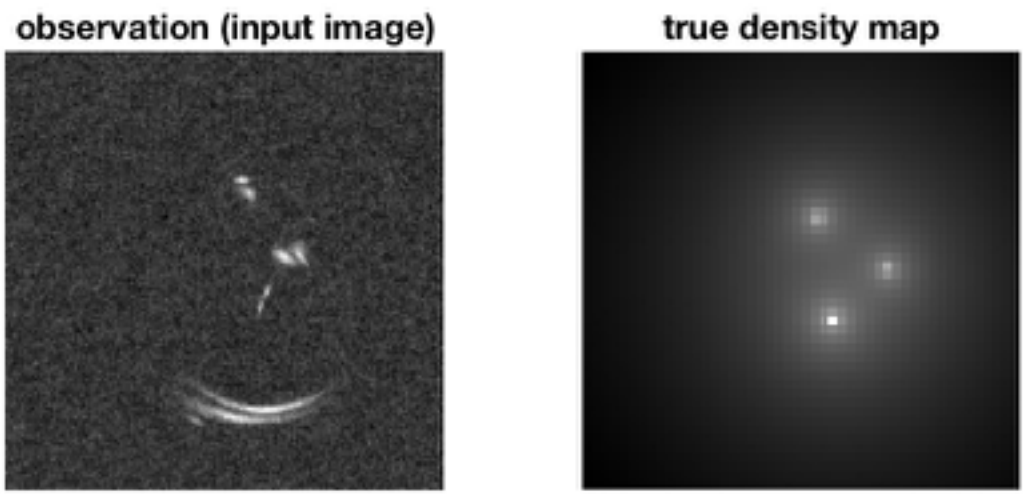
ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?



ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?

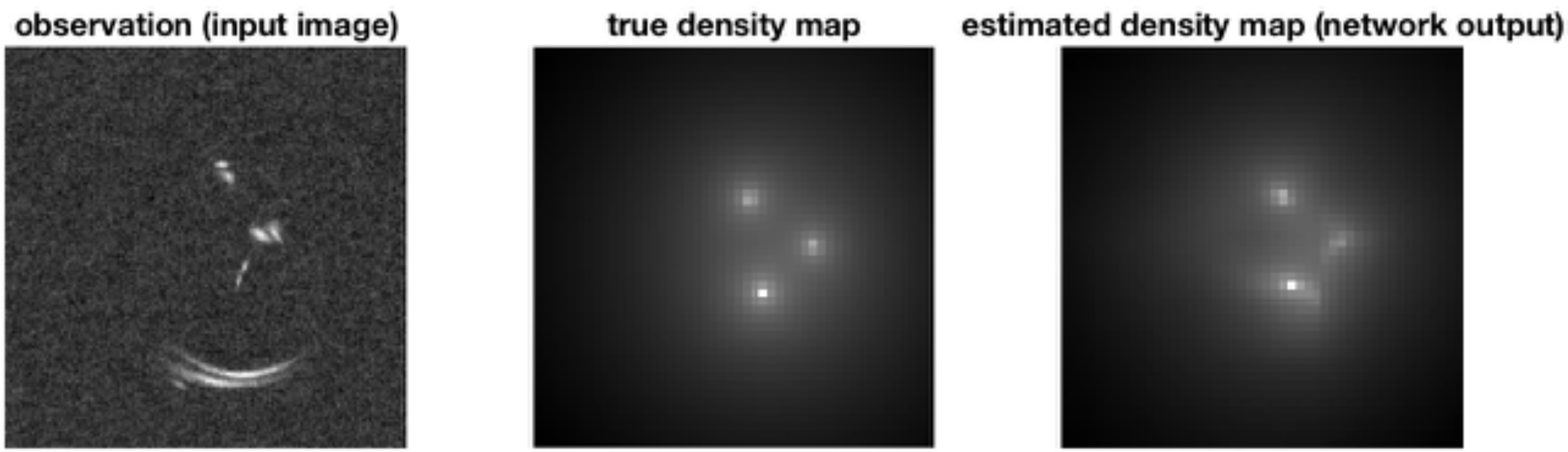


ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?



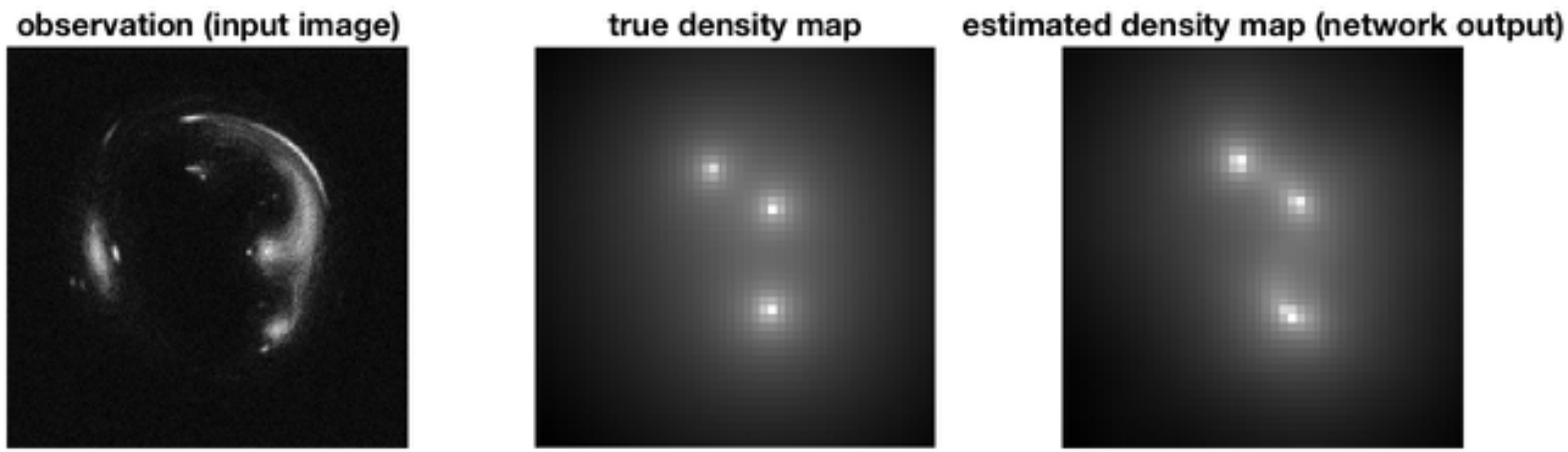
Are neural nets doing a dictionary look-up?

No



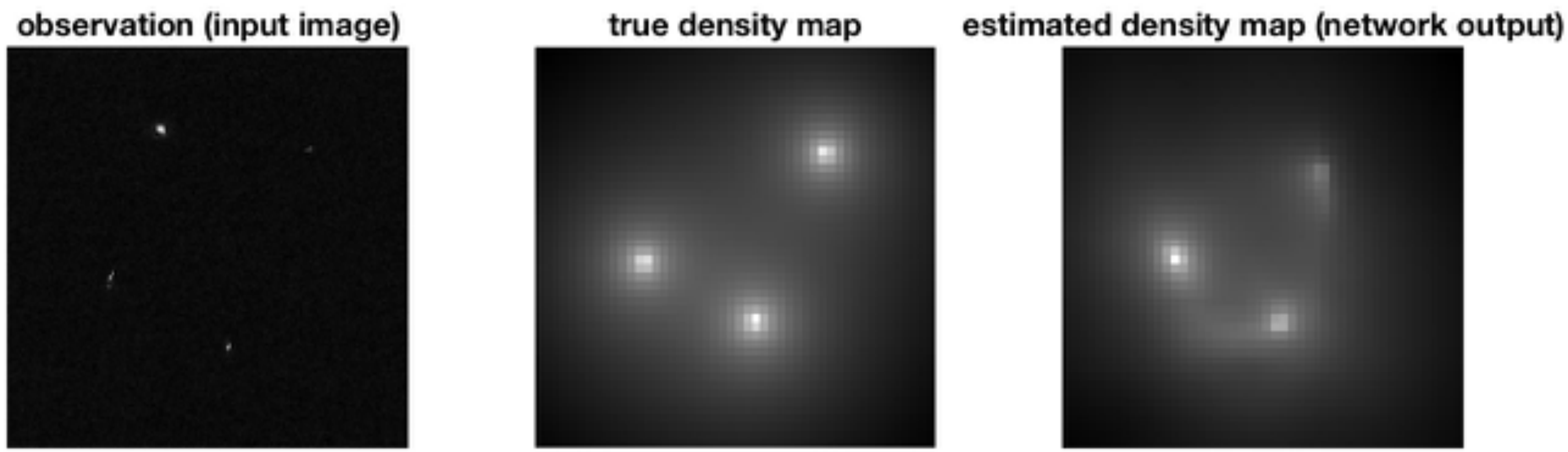
ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?

NO



ARE NEURAL NETS DOING A DICTIONARY LOOK-UP?

NO



Thank you