University of Amsterdam

NWO

GRAPPA
GRavitation AstroParticle Physics Amsterdam

# Hunting small dark matter halos in strongly lensed images with backpropagation

Image credits: DeepIdeas


## Christoph Weniger

1909.xxxxx: Marco Chianese, Adam Coogan, Paul Hofma, Sydney Otten, CW

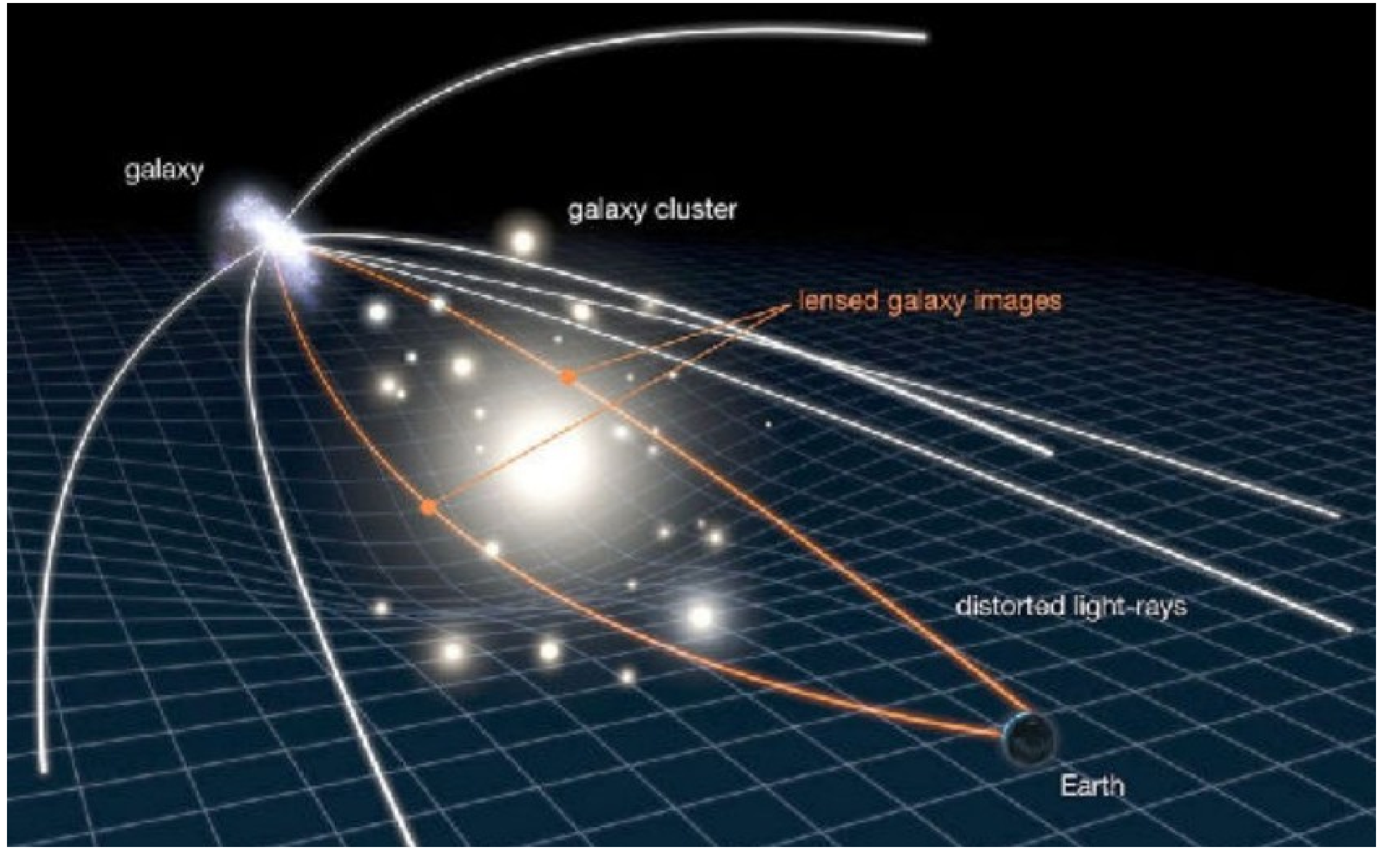

8 Oct 2019, 4th IBS-MultiDark-IPPP Workshop

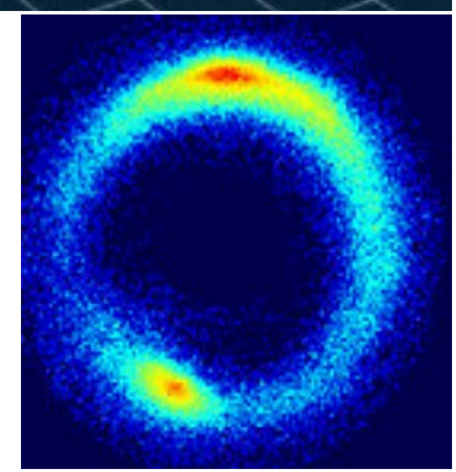
# Outline

- Motivation
- What are deep neural networks?
- What can one do with auto-grad?
- Using deep neural networks for strong lensing image analysis

# Strong gravitational lensing

- Light from distant galaxies is deflected by DM halos long the line of sight
- Leads to multiple images, arcs, near-perfect Einstein rings
- Careful analysis of lensed images reveals information about DM halos

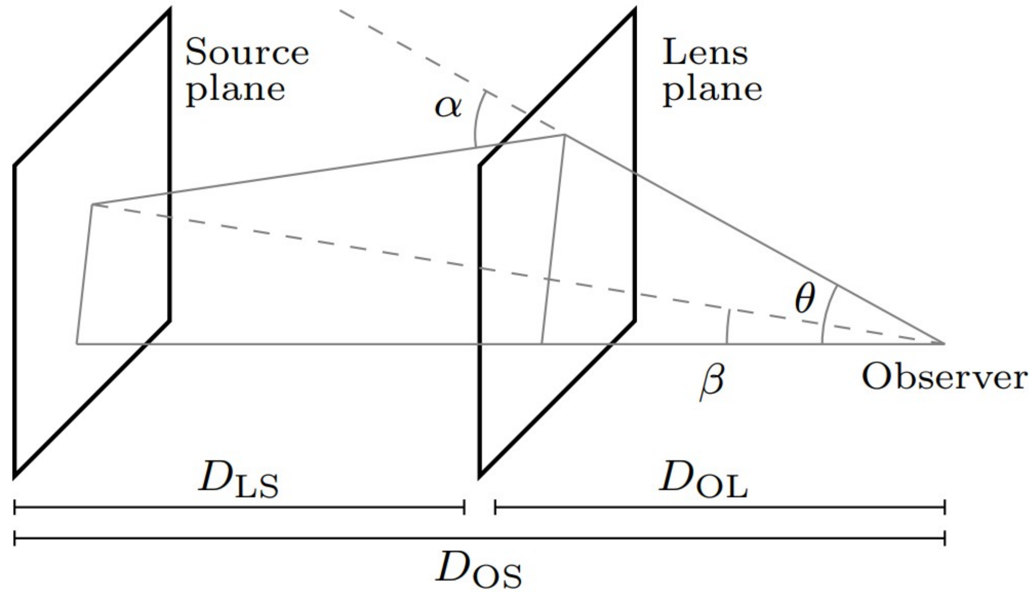


# Strong lensing basics

**Basic geometry**

Thin lense approximation




**Displacement field**
from Poisson kernel convolution

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

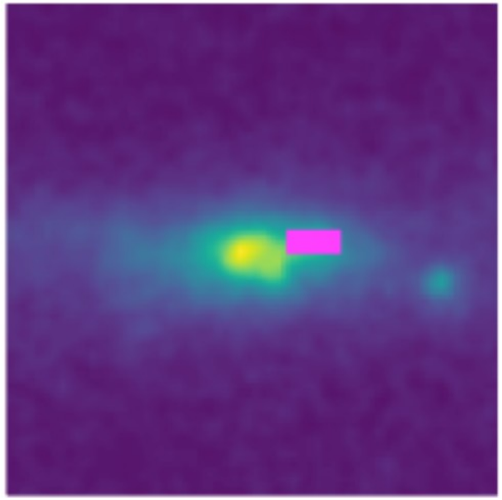
**Lensed image**

(surface brightness conserved)

$$\mathcal{I}_{\rm lens}(\boldsymbol{\theta}) = \mathcal{I}_{\rm src}(\boldsymbol{\theta}-\boldsymbol{\alpha}(\boldsymbol{\theta}))$$

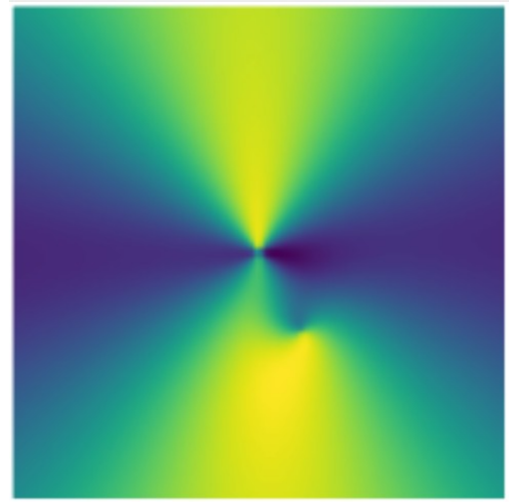
# Problem setup

**Source**

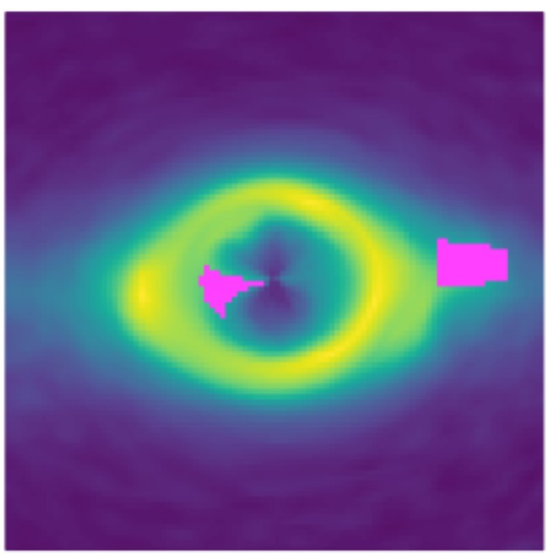


**Lens**

Displacement field



$$\mathcal{I}_{\rm lens}(\boldsymbol{\theta}) = \mathcal{I}_{\rm src}(\boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta}))$$



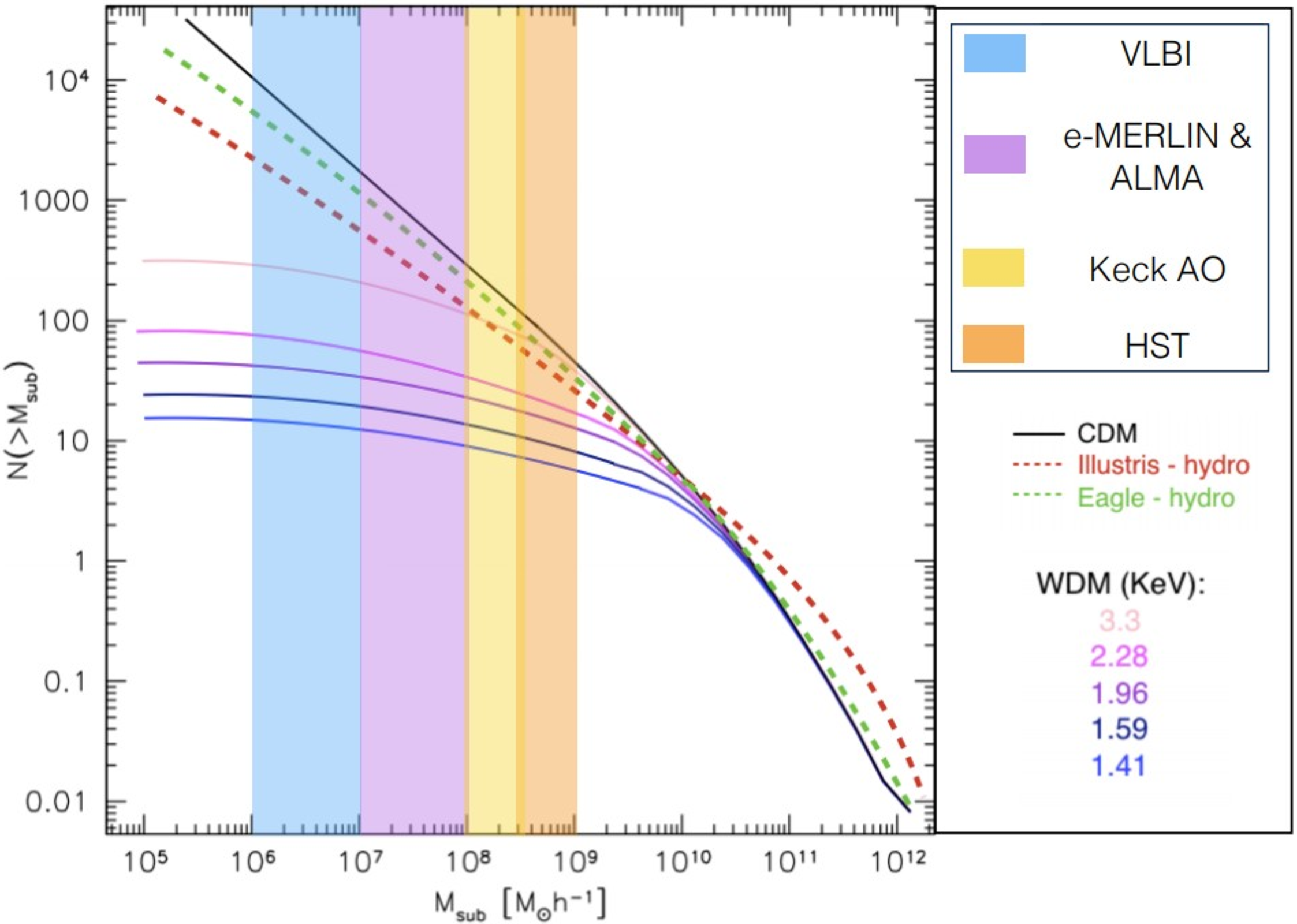
**Lensed image**

With multiply lensed regions

**Challenge**: Reconstruct both lens and source from a single image

Codes: *Lenstronomy*, Birrer & Amara 2018
*AutoLens*, Nightingale 2017

# Probe for DM temperature & mass



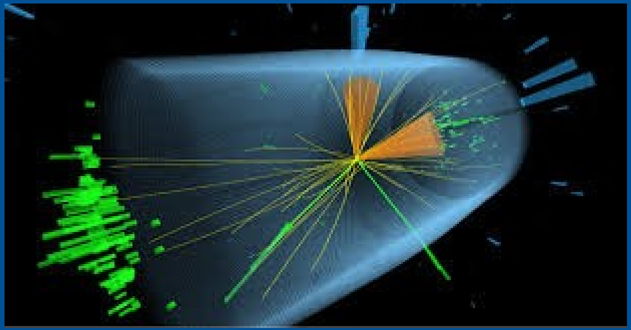

The cut-off in the mass function is directly related to the model for dark matter.

Slide credit: John McKean (ASTRON) 2018

# A.I. technology

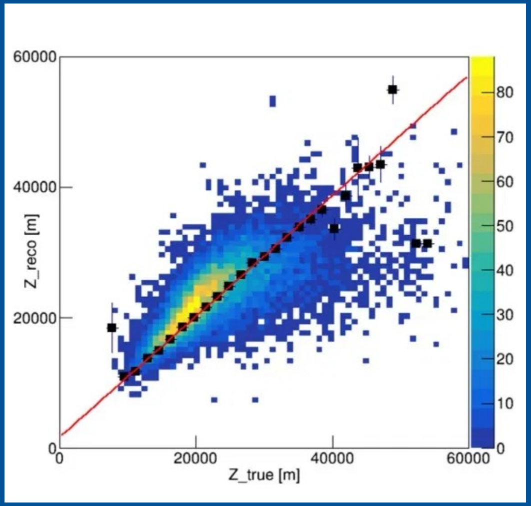
# A.I. technology

**LHC event reconstruction**



https://arxiv.org/abs/1806.11484

**CTA event reconstruction**




http://arxiv.org/abs/1810.00592

**A.I.**

**Super-fast & approximate** — **Deep Learning**

**Computer Science** — **Fast & exact**

**automatic differentiation**

probabilistic programming

high-dimensional optimization

**Unrealized potential**

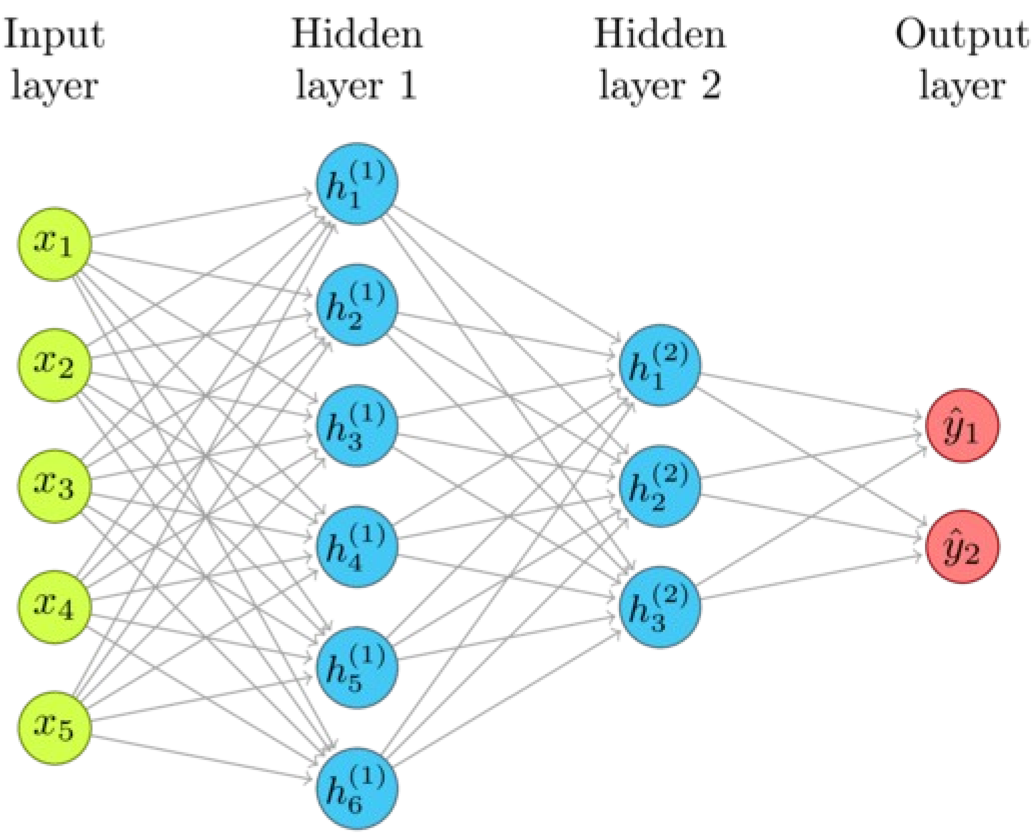
# Deep neural networks

**Deep neural networks are extremely flexible function approximators**

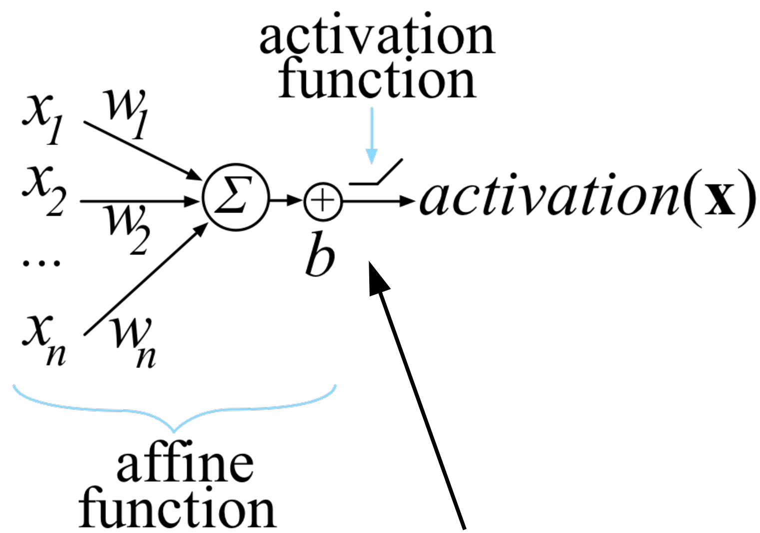
Input: $\vec{x}$ $\qquad f_\phi(\vec{x}) = \vec{y}$ $\qquad$ Output : $\vec{y}$

$\phi$ : Internal network parameters/weights

Typical feed-forward network with multiple hidden layers




Layer connection: Stacking affine and activation function

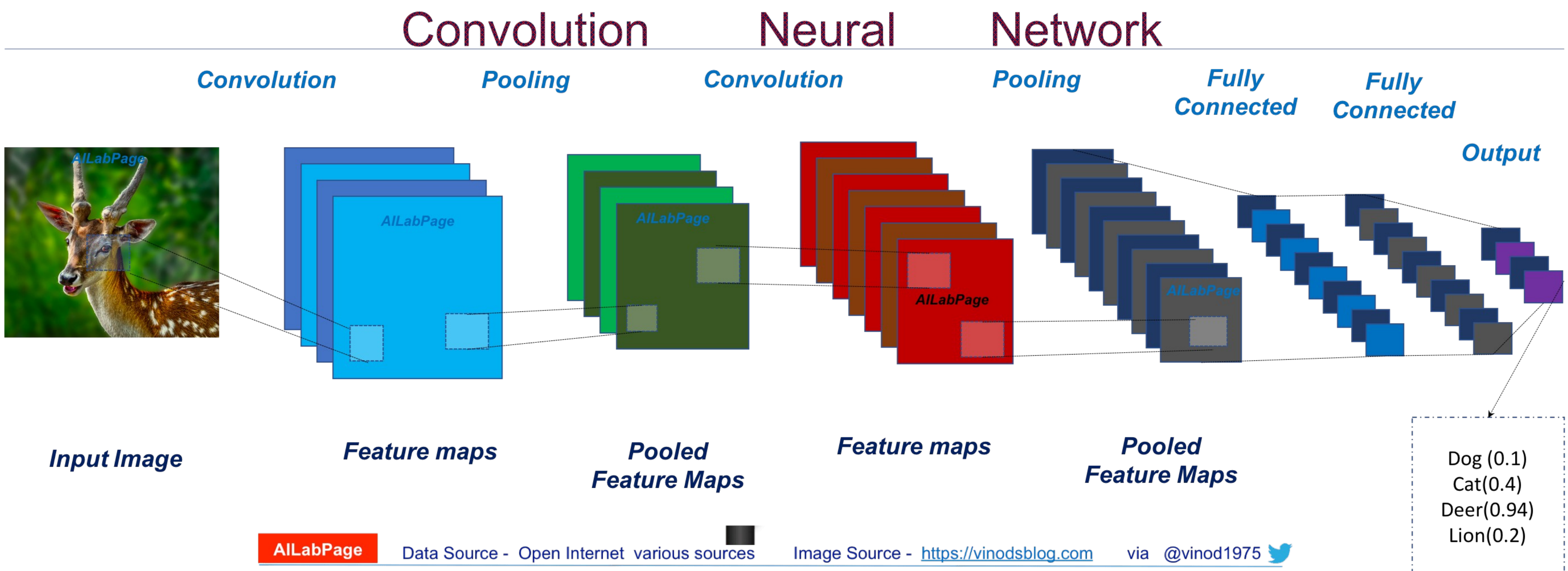



ReLU : $\max(0, x)$

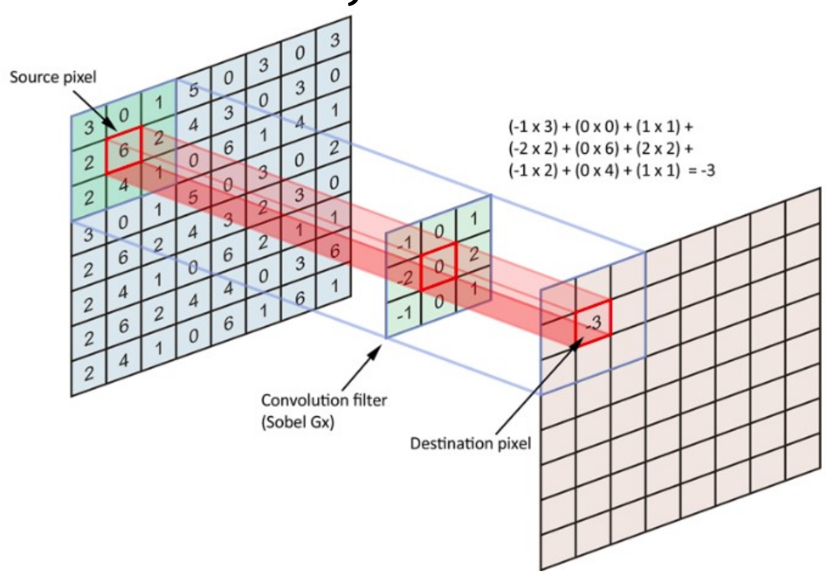
Network weights and biases $\qquad \phi = \vec{w}, \vec{b}$

Works theoretically because of "Universal approximation theorem"

# Example: Image classification network




## Convolutional layer




## Pooling layer

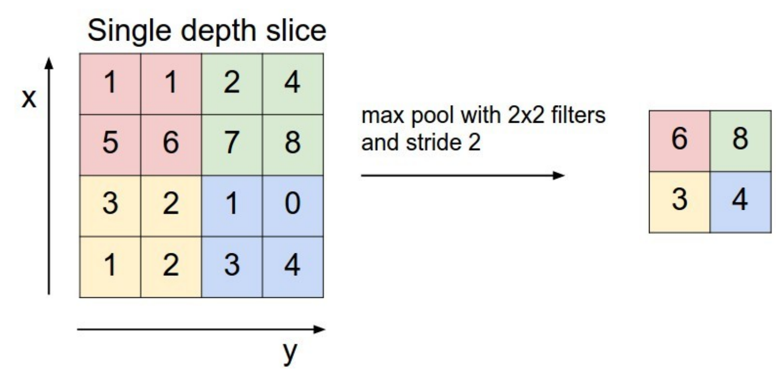
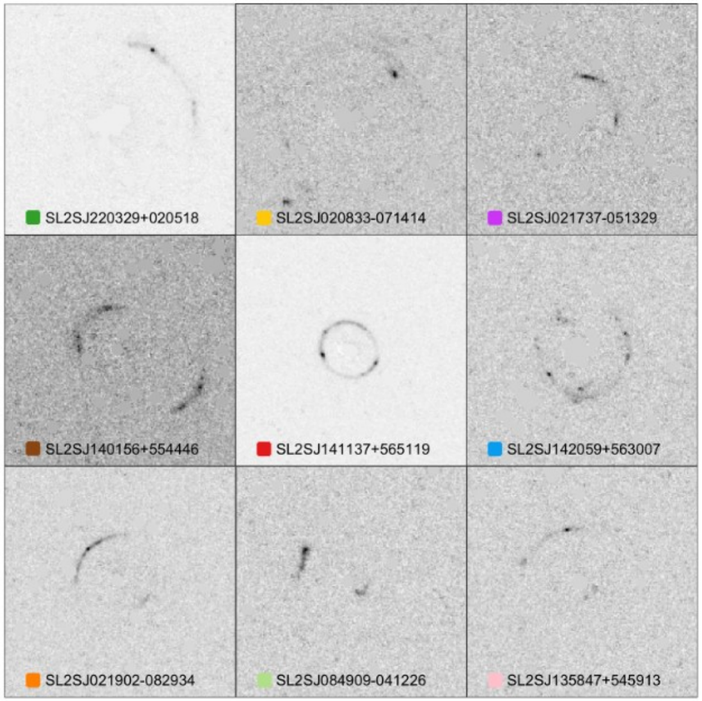


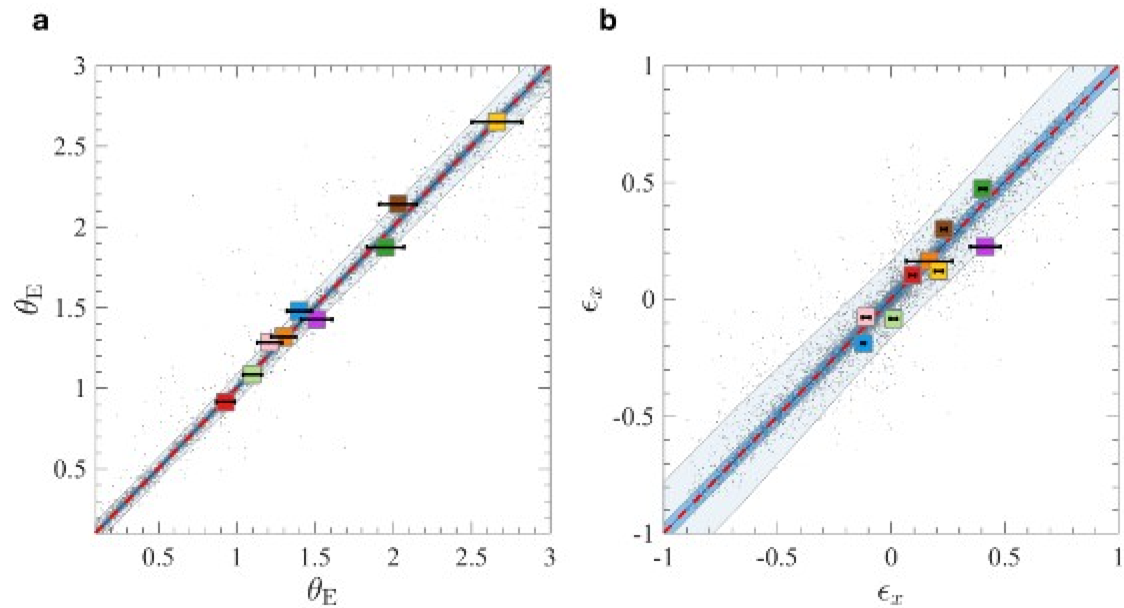

Image credit: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2, http://cs231n.github.io/convolutional-networks/

# Ex 1: Parameter inference with CNNs




Gravitational lensing data (HST)

Reconstructed parameters




- Amazing speed up: 1 day → 0.01 s
- Somewhat less accurate
- Challenge: Subhalos will show up as O(1%) perturbations of best-fit model, unclear if models can be trained for that as well?

Hezaveh, Y. D., Levasseur, L. P. & Marshall, P. J. Fast Automated Analysis of Strong Gravitational Lenses with Convolutional Neural Networks, Nature 2017.

# Training with gradient decent

1. Define loss function
   (here square error / chi-square)

$$\text{loss} = \sum_{\text{batch}} (f_\phi(\vec{x}_i) - \vec{y}_i)^2$$

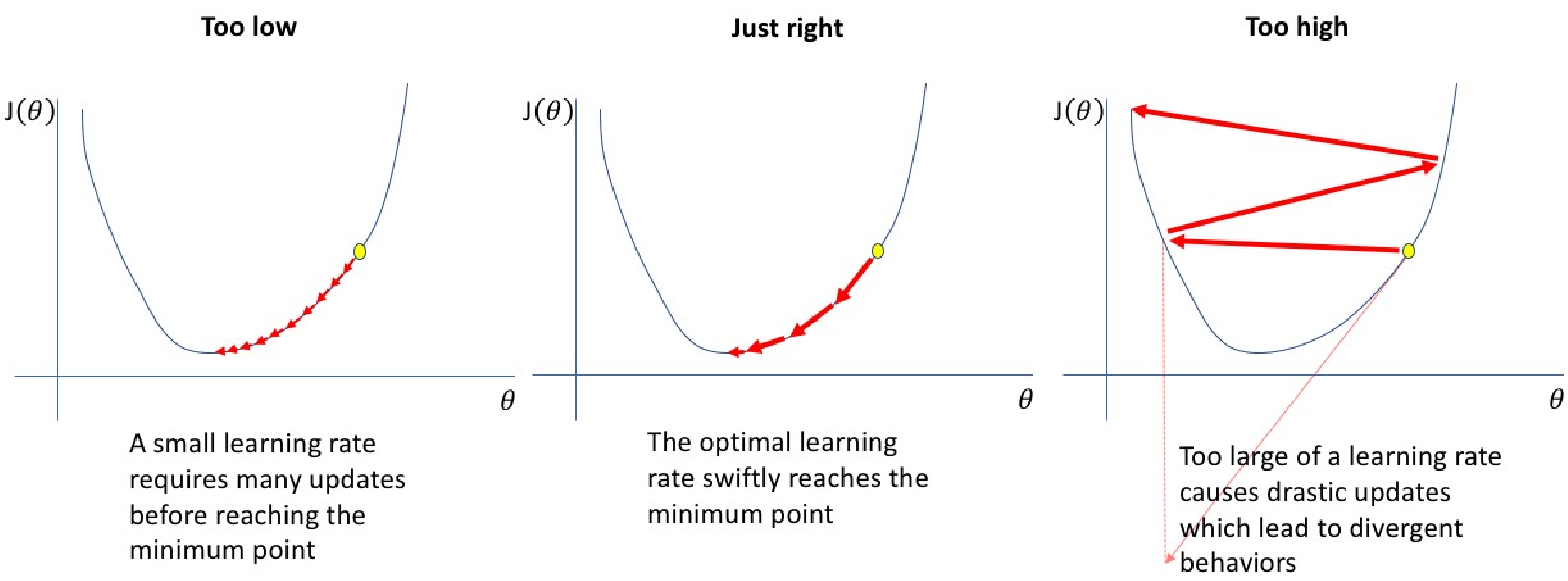
2. back-propagation

$$\text{grad} = \frac{\partial \text{loss}}{\partial \phi}$$

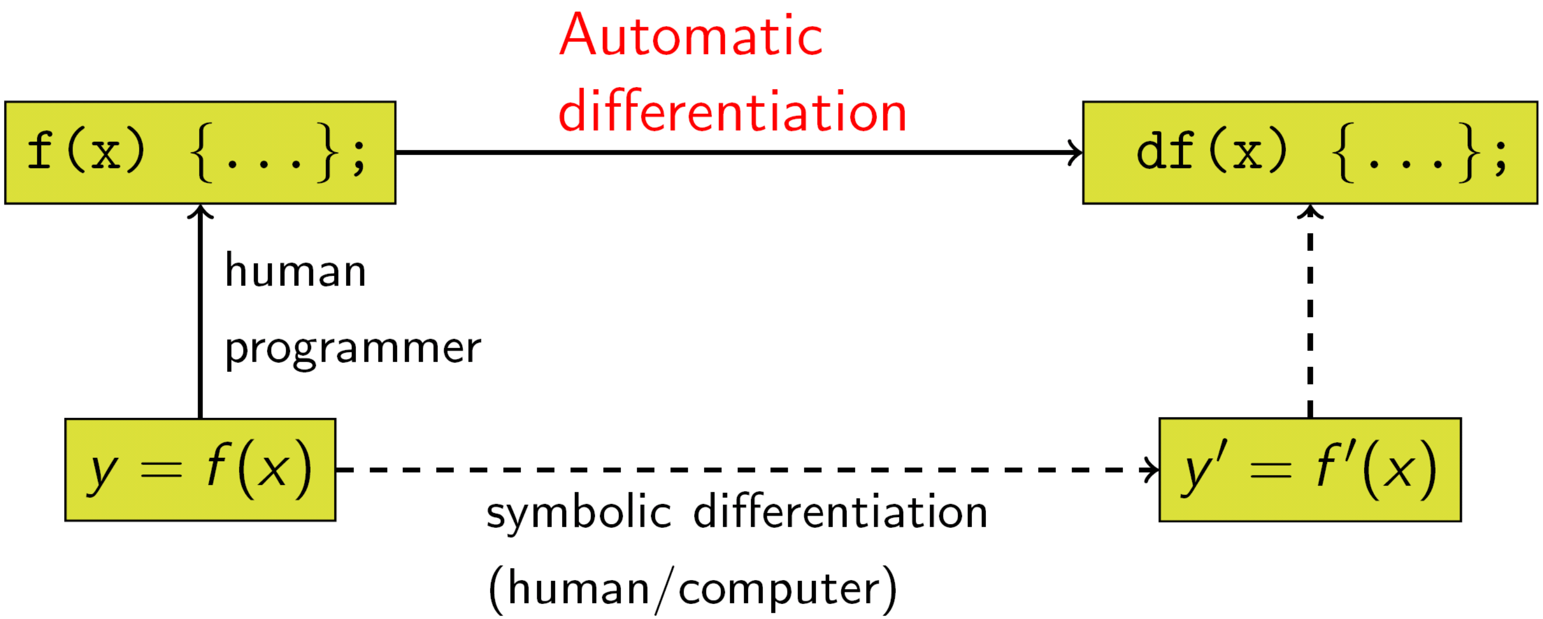
3. parameter update

$$\phi \to \phi - \text{LR} \cdot \text{grad}$$

Stop when it works well on separate *test data*



# Many recent auto-grad tools




Forward-mode Differentiation: $$h_i \equiv \frac{\partial}{\partial x} f_i(x)$$

Back-propagation: $$g_i \equiv \frac{\partial}{\partial x_i} f(\vec{x})$$

https://medium.freecodecamp.org/demystifying-gradient-descent-and-backpropagation-via-logistic-regression-based-image-classification-9b5526c2ed46

# What else to do with auto-grad?

Source: Deep Ideas

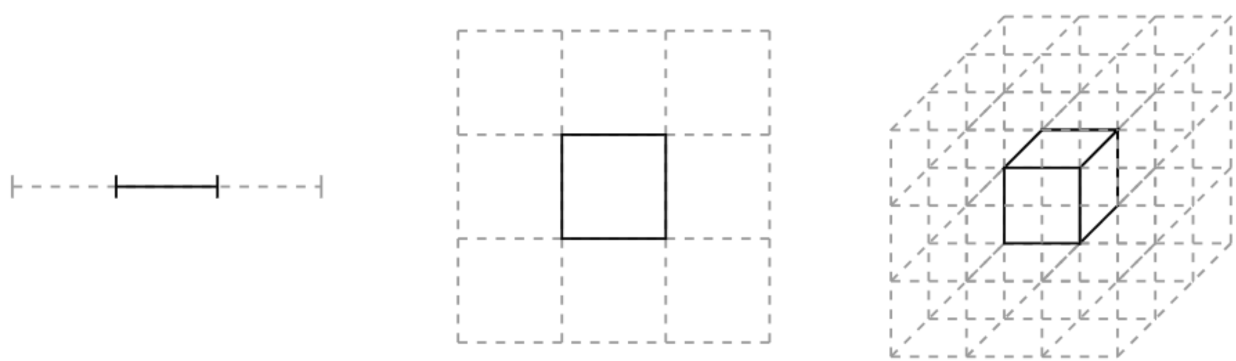
# 1) Hamiltonian Monte Carlo

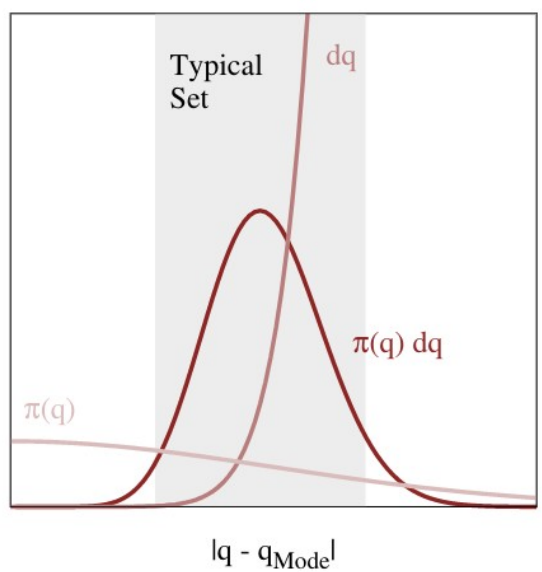
Typical goal of **Bayesian inference** is to evaluate expectation values of the form

$$\mathbb{E}_\pi[f] = \int_Q \mathrm{d}q\, \pi(q)\, f(q)$$

Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo (1701.02434).

**The course of dimensionality**

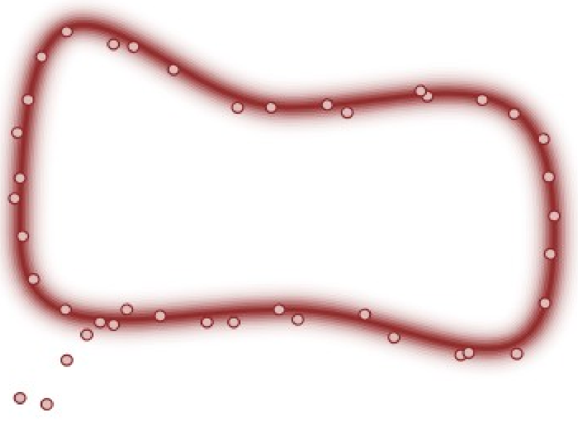




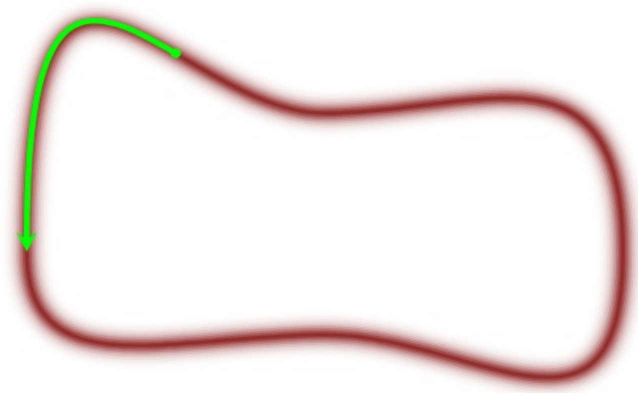

The integral is dominated by points in the "**typical set**", which is *not* conincident with the mode if dimensionality is large

# 1) Hamiltonian Monte Carlo

Goal is to explore the "typical set", using Monte Carlo techniques.

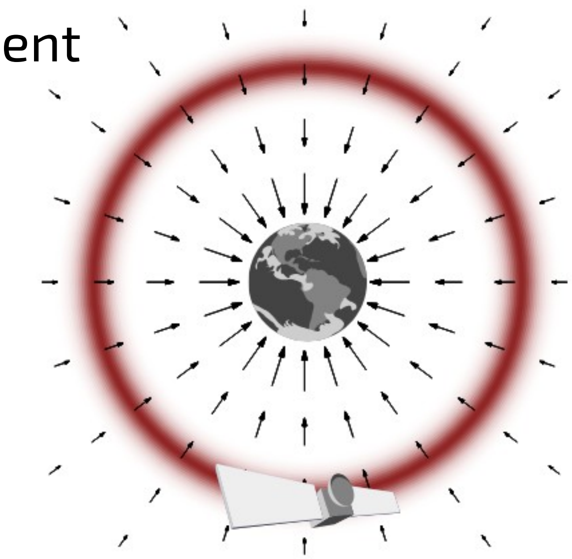


Knowing the direction of the typical set would greatly help.



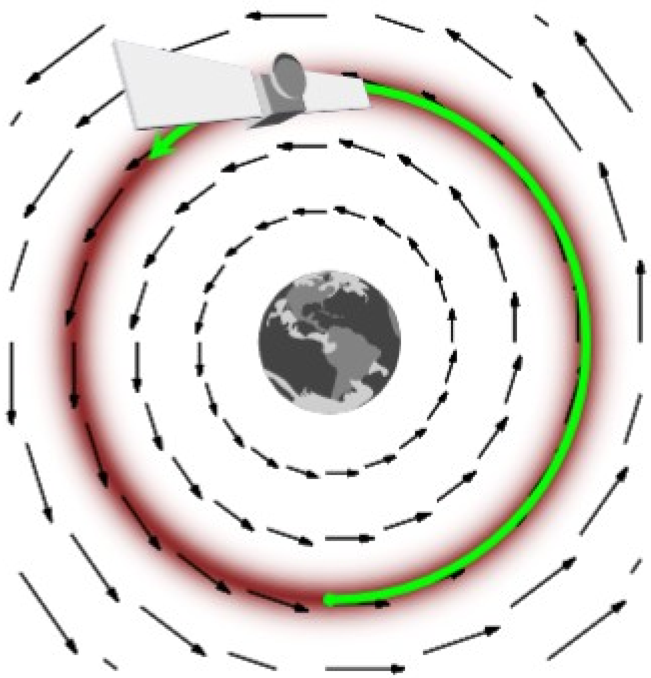
However, simply following the gradient would just lead to the mode

$$V(q) = -\ln \pi(q)$$



# 1) Hamiltonian Monte Carlo

Idea: Introduce momentum, *p*, such that MC is kept in the "orbit" of highest density mass (in the 'typical set').



Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo (1701.02434).

**New target function**

$$\pi(q,p) = e^{-H(q,p)}$$

$$\begin{aligned} H(q,p) &= -\log \pi(p \mid q) - \log \pi(q) \\ &\equiv \quad K(p,q) \quad + \; V(q)\,. \end{aligned}$$

**Sampling**

- Pick initial point
- follow Hamiltonian dynamics

$$\frac{\mathrm{d}q}{\mathrm{d}t} = +\frac{\partial H}{\partial p} = \frac{\partial K}{\partial p}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -\frac{\partial H}{\partial q} = -\frac{\partial K}{\partial q} - \frac{\partial V}{\partial q}$$

- resample momentum from canonical distribution, preserve position

# 2) Variational Inference

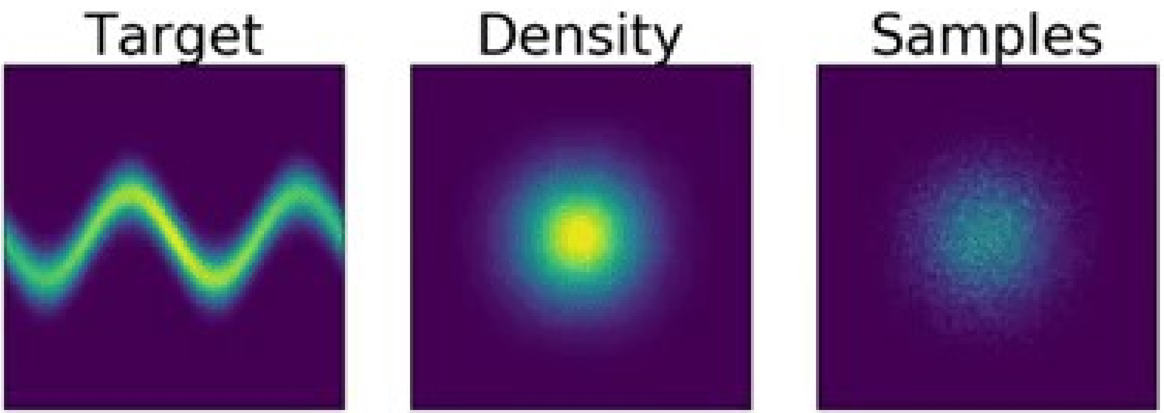
**Kullback-Leibler divergence:**

$$\mathrm{KL}(q(z)||p(z|x)) = \int dz q(z) \ln \frac{q(z)}{p(z|x)} = \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z}\,|\,\mathbf{x})]$$

Fit parameteric model for posterior, q(z), to true posterior p(z|x).

Sampling problem → Optimization problem

$$q^*(\mathbf{z}) = \arg\min_{q(\mathbf{z})\in\mathscr{Q}} \mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z}\,|\,\mathbf{x})\right)$$



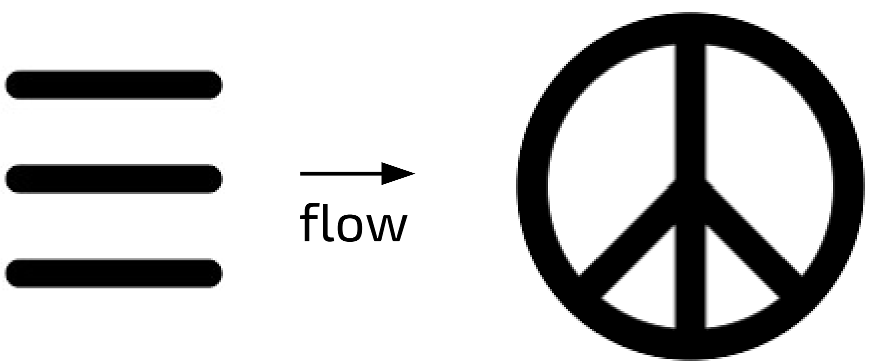

Video credit: Jesse Bettencourt

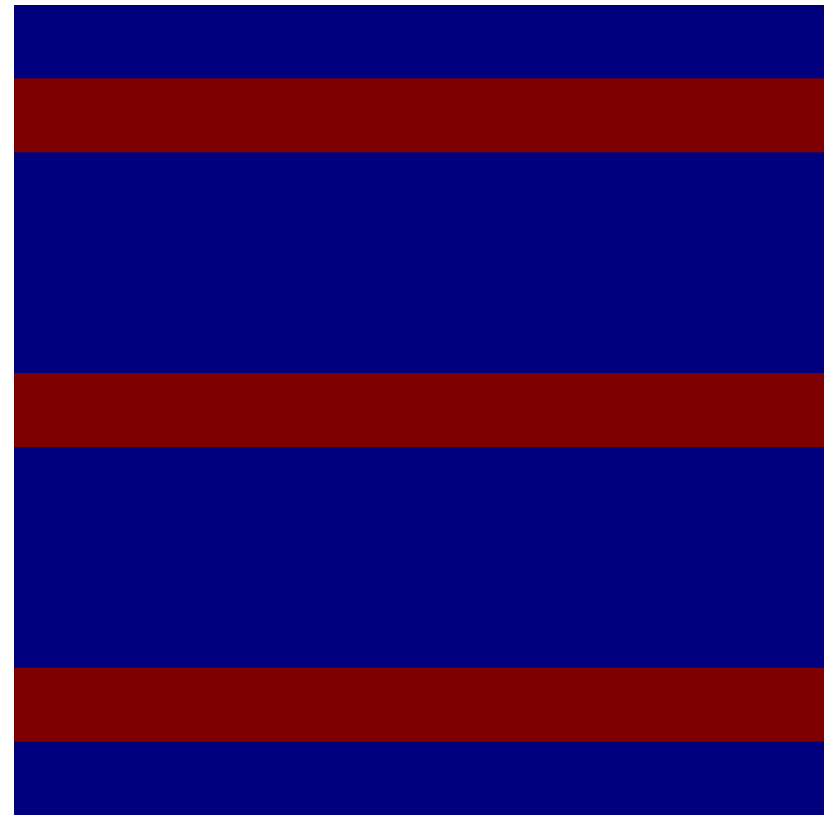
Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. Variational Inference: A Review for Statisticians (1601.00670).

# 3) Auto-grad through Euler fluid equations

**Optimization goal**

Find initial velocity field that does this



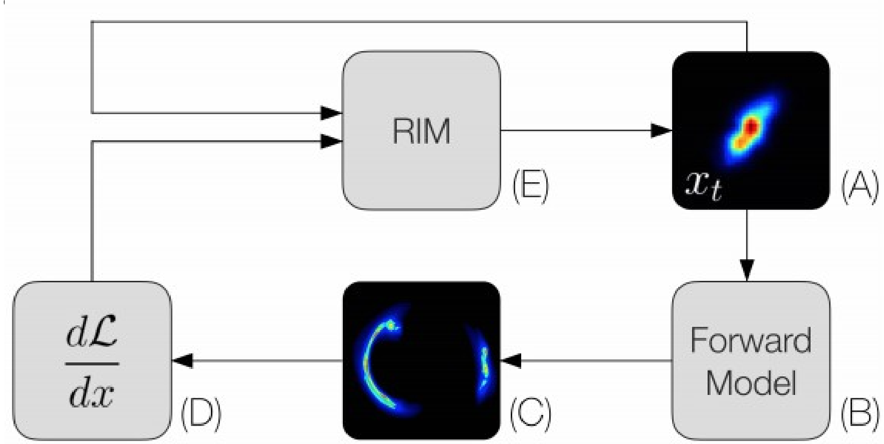
https://github.com/HIPS/autograd

# Back to strong lensing

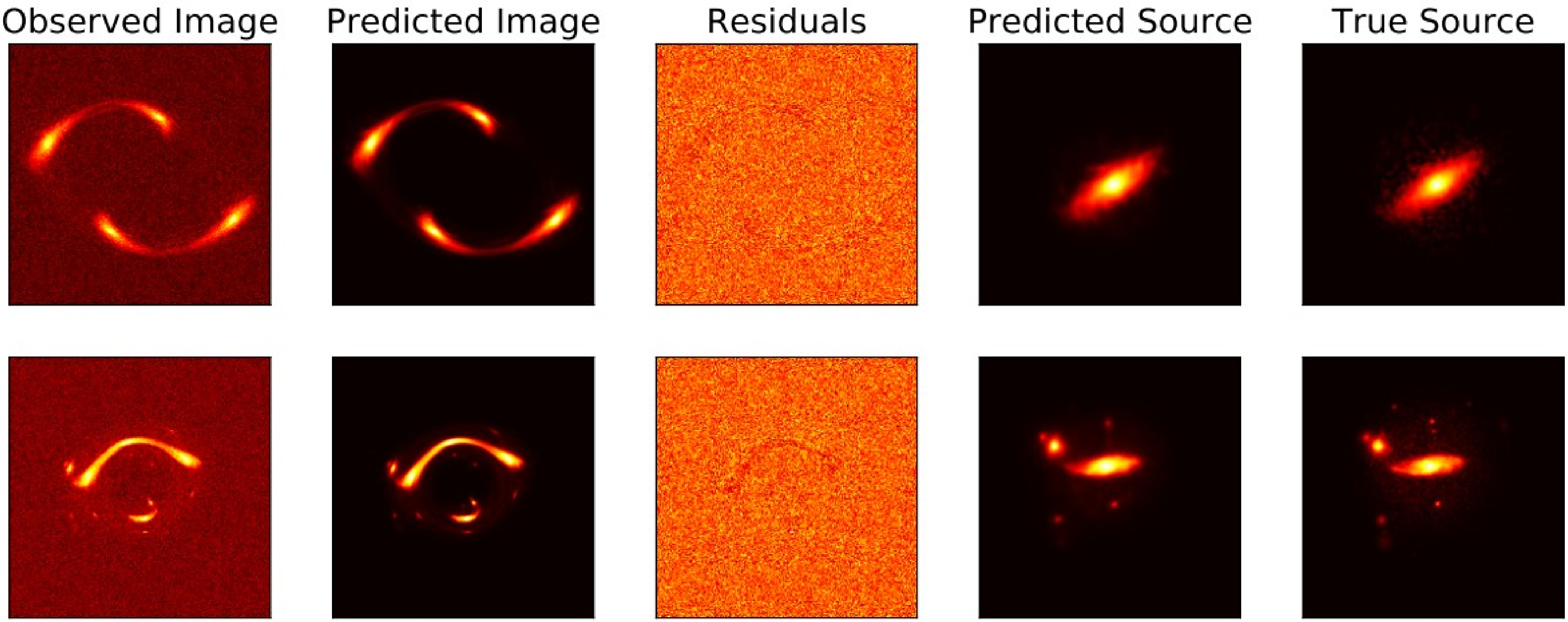
# Ex 2: Guiding source fit with RIM

**Recurrent inference machines for source modeling**

Morningstar+ 2019. https://arxiv.org/abs/1901.01359

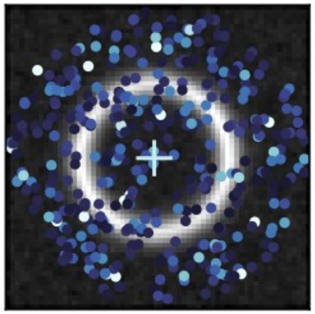



Guide source fit, integrating both gradient and galaxy prior information.



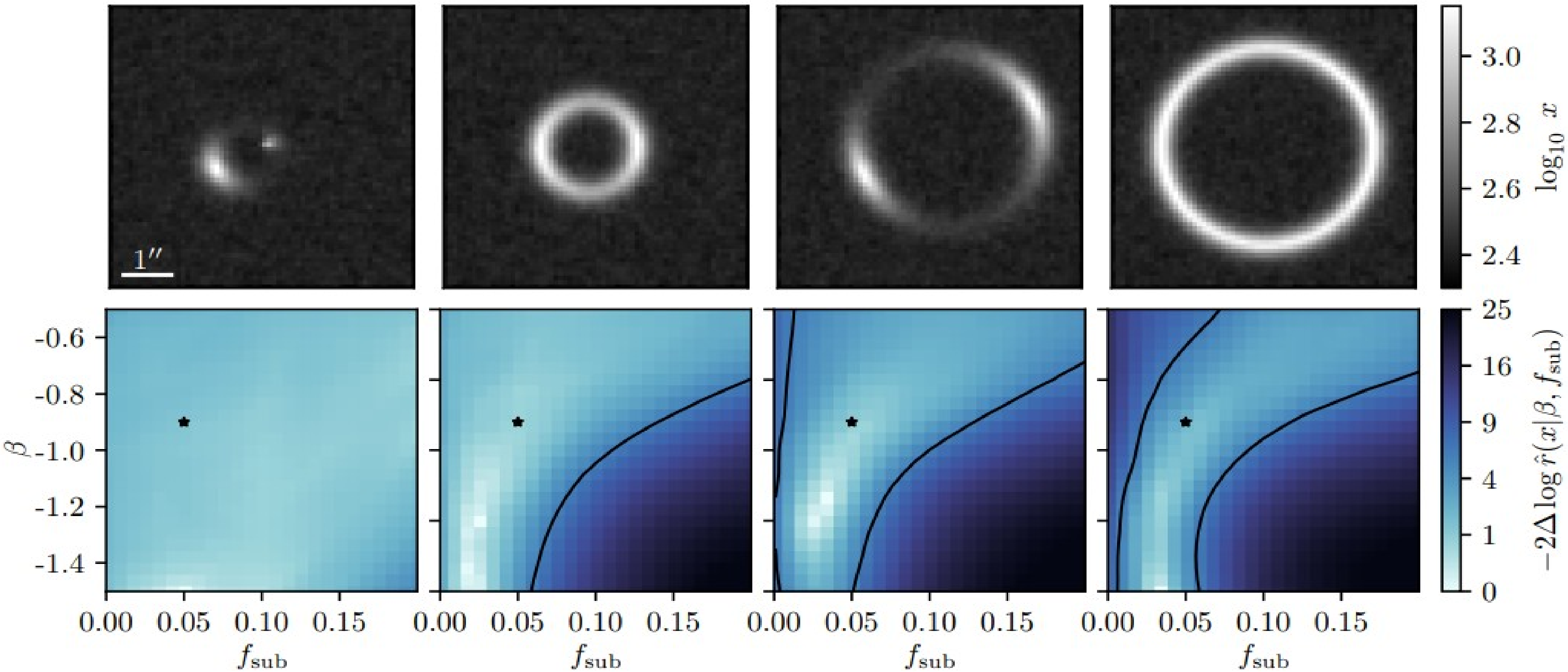
# Ex 3: Likelihood free inference for subhalos

**Likelihood free inference for modeling lenses with substructure**
Brehmer+ 2019, https://arxiv.org/abs/1909.02005



Use CNN to estimate likelihood ratio*

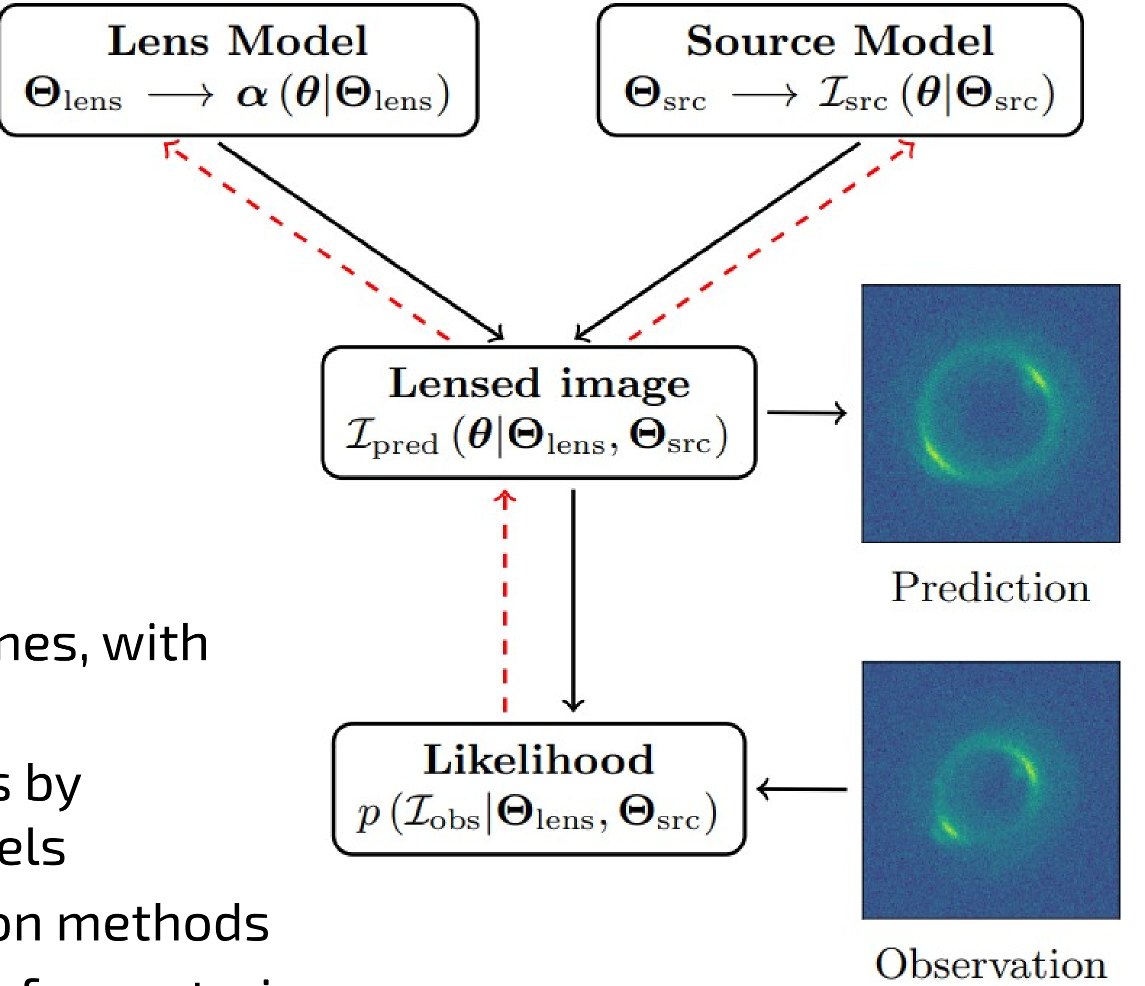
$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$




*what is learned is a binary classifier, whose results are then recalibrated to yield a likelihood ratio

Cranmer+ 1506.02169

# Differentiable probabilistic programming

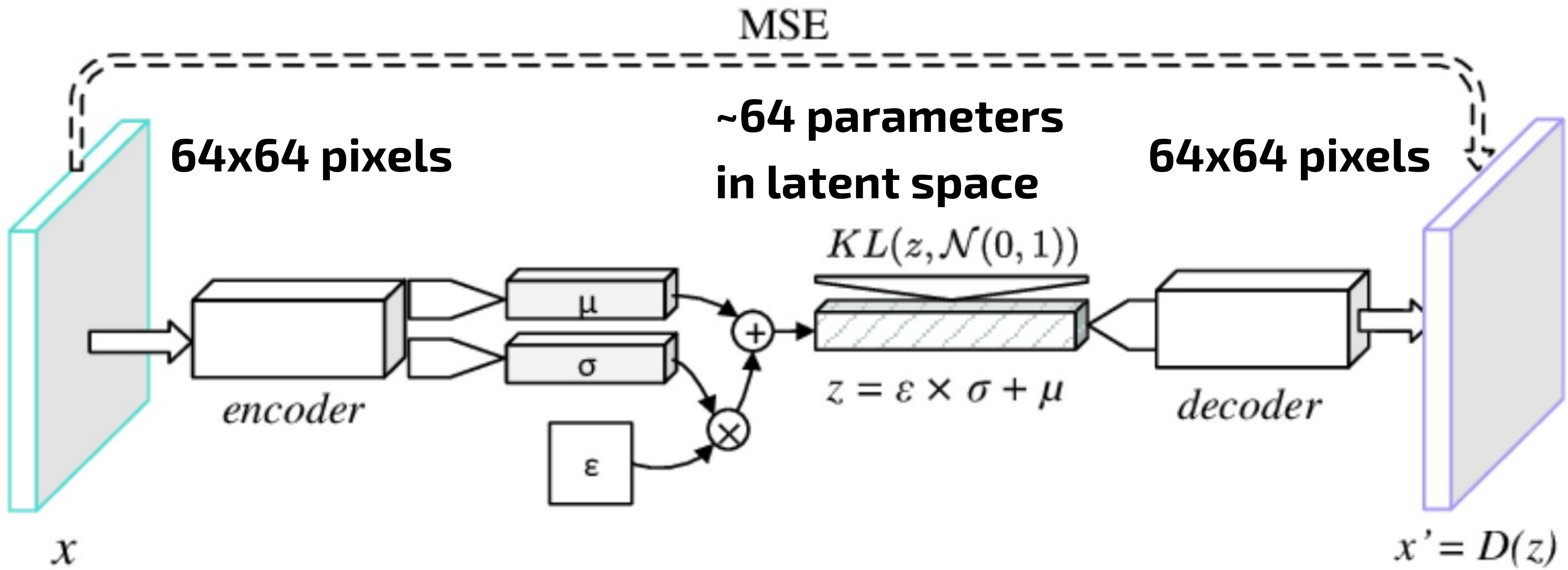



**Philosophy**

- Write traditional lensing pipelines, with back-propagation
- Replace individual components by deep/physical generative models
- Use gradient based optimization methods
- Use propabilistic programming for posterior estimates

1909.xxxxx: Marco Chianese, Adam Coogan, Paul Hofma, Sydney Otten, CW

# Variational Auto-Encoder as source model




**Components**

- Generative model
  $p(x, z) = p(x|z)p(z)$
- Inference model
  $q(z|x)$

**Training by ELBO maximization**

$$\mathrm{ELBO}(q) = \mathbb{E}[\log p(\mathbf{x} \,|\, \mathbf{z})] - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z}))$$

Marginal log-likelihood

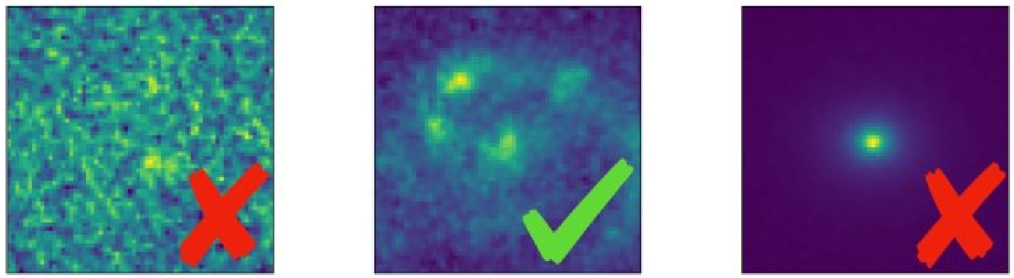
Difference w.r.t. prior

$$\log p(\mathbf{x}) \geq \mathrm{ELBO}(q)$$

Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. arXiv [stat.ML] (2013).

# Training data set

- Dataset: ~56,000 galaxies, redshifts ~ 1

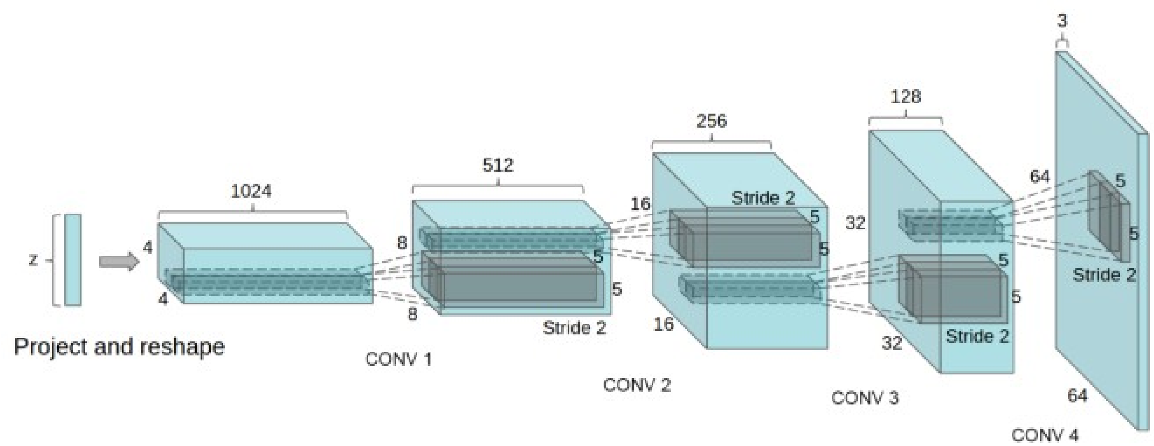


S/N < 10 S/N ~ 20 S/N > 100

**This talk: train on ~10,000 images with S/N = 15 - 50**

- Encoder, decoder: deep convolutional neural networks

**Eg, decoder:**



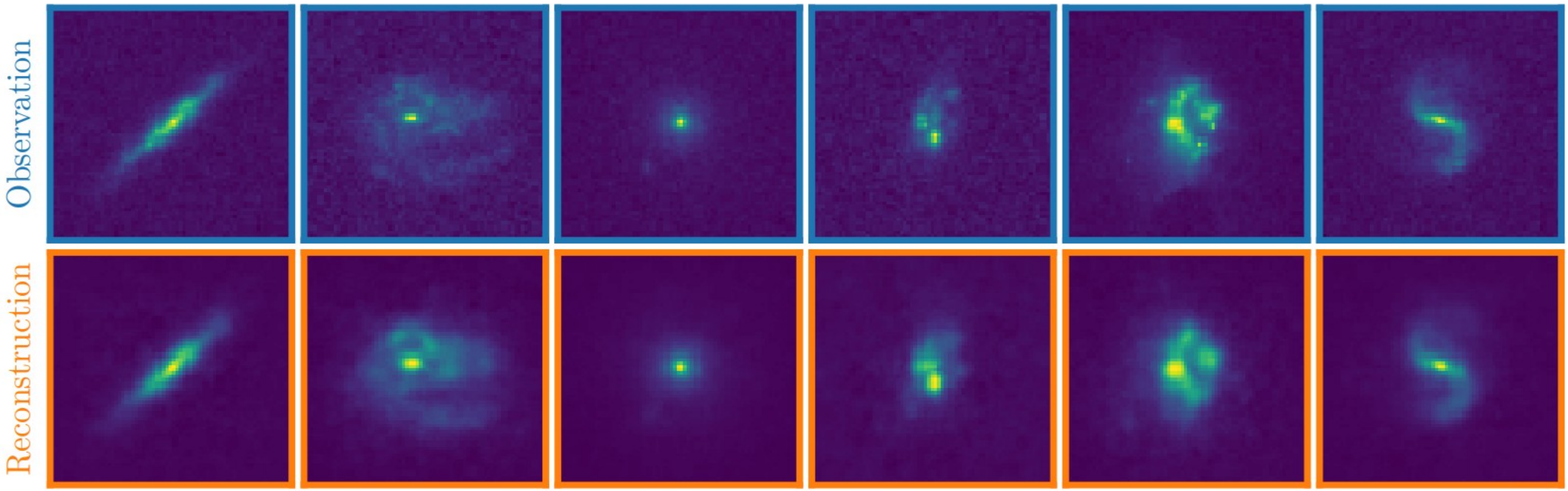

http://great3.jb.man.ac.uk/

Radford et al 2015 (DCGAN)

Slide credit: Adam Coogan

# Source galaxy reconstruction w/o lensing

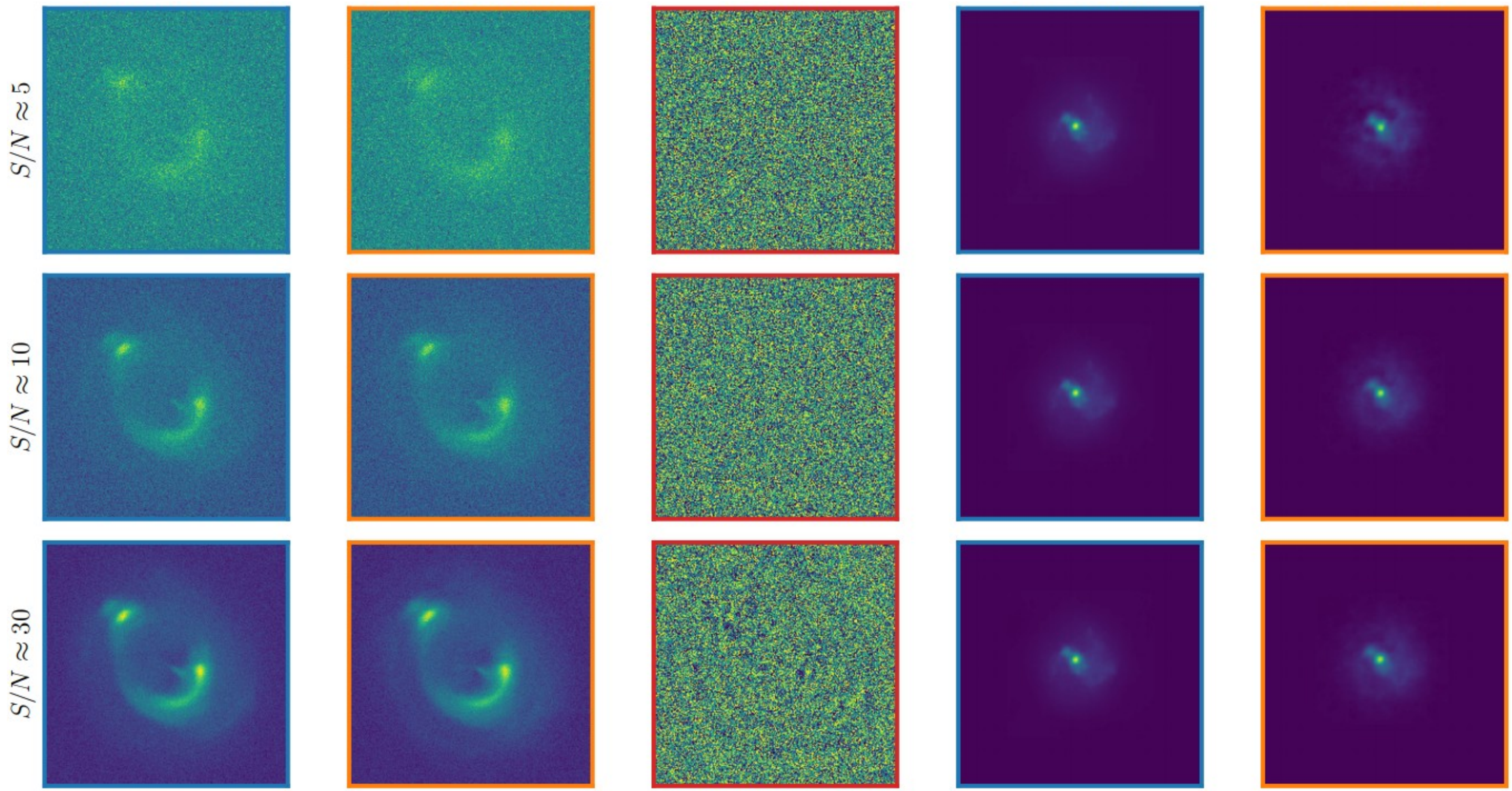
Galaxy image → (Encoder) → Latent space → (Decoder) → Reconstruction




Generative model (or "decoder") seems to be expressive enough to model real galaxies (though somewhat blured).

Can we use this in a "traditional" fit to lensed images?

# Source galaxy reconstruction with lensing




Mock data

Residuals

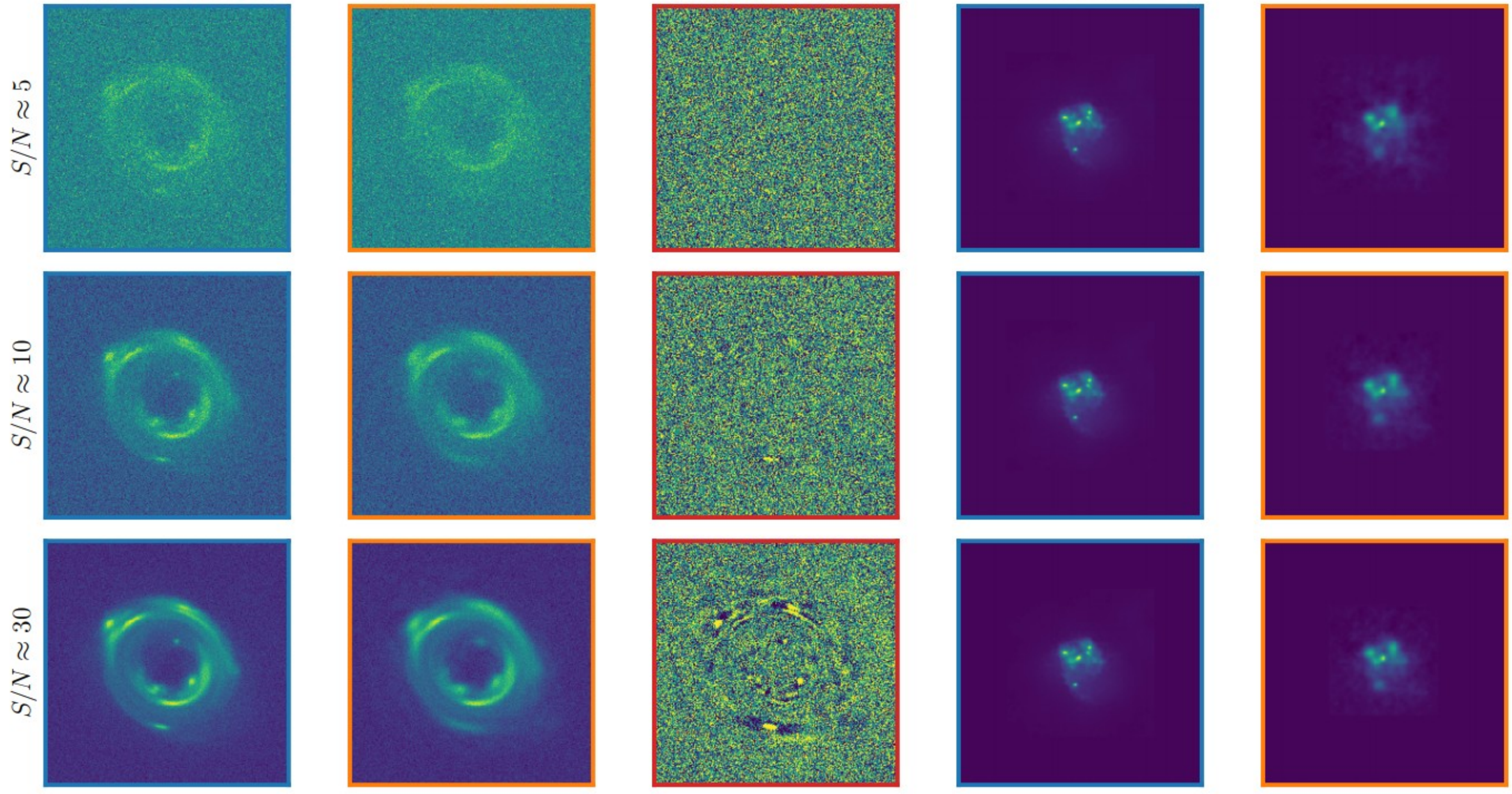
Best-fit source

Original source

Test system 1

Fit with 8 lens parameters
and 64 source paramters

# Source galaxy reconstruction with lensing




Mock data

Residuals

Best-fit source

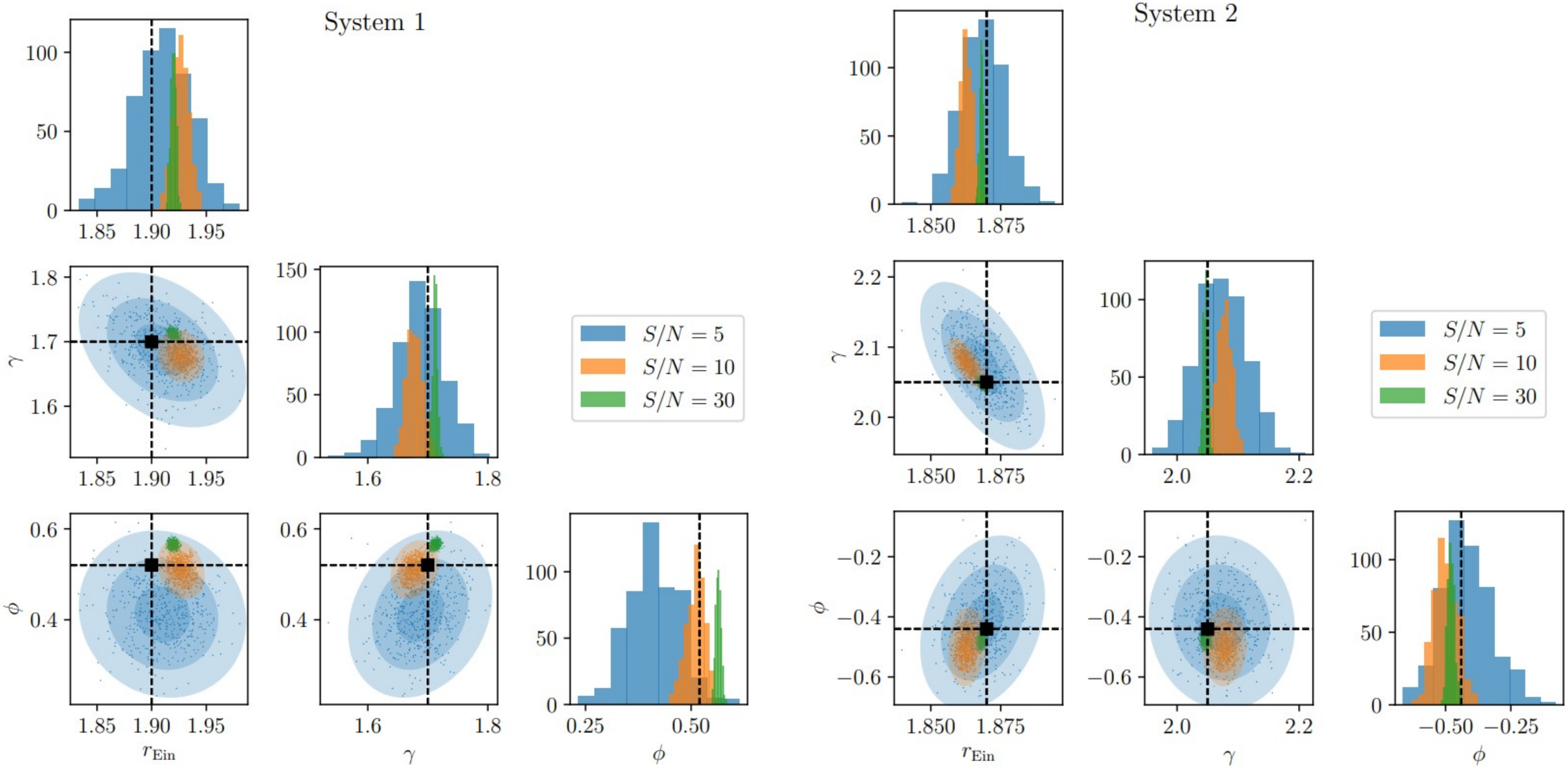
Original source

Test system 2

Fit with 8 lens parameters
and 64 source paramters

# Parameter reconstruction with HMC

We use Hamiltonian Monte Carlo to sample the ~75 dimensional posterior.




- Works excellent, but results are slightly biased for how S/N images
- Likely due to limited expressiveness of source model
- Estimating effect on subhalo searches is work in progress

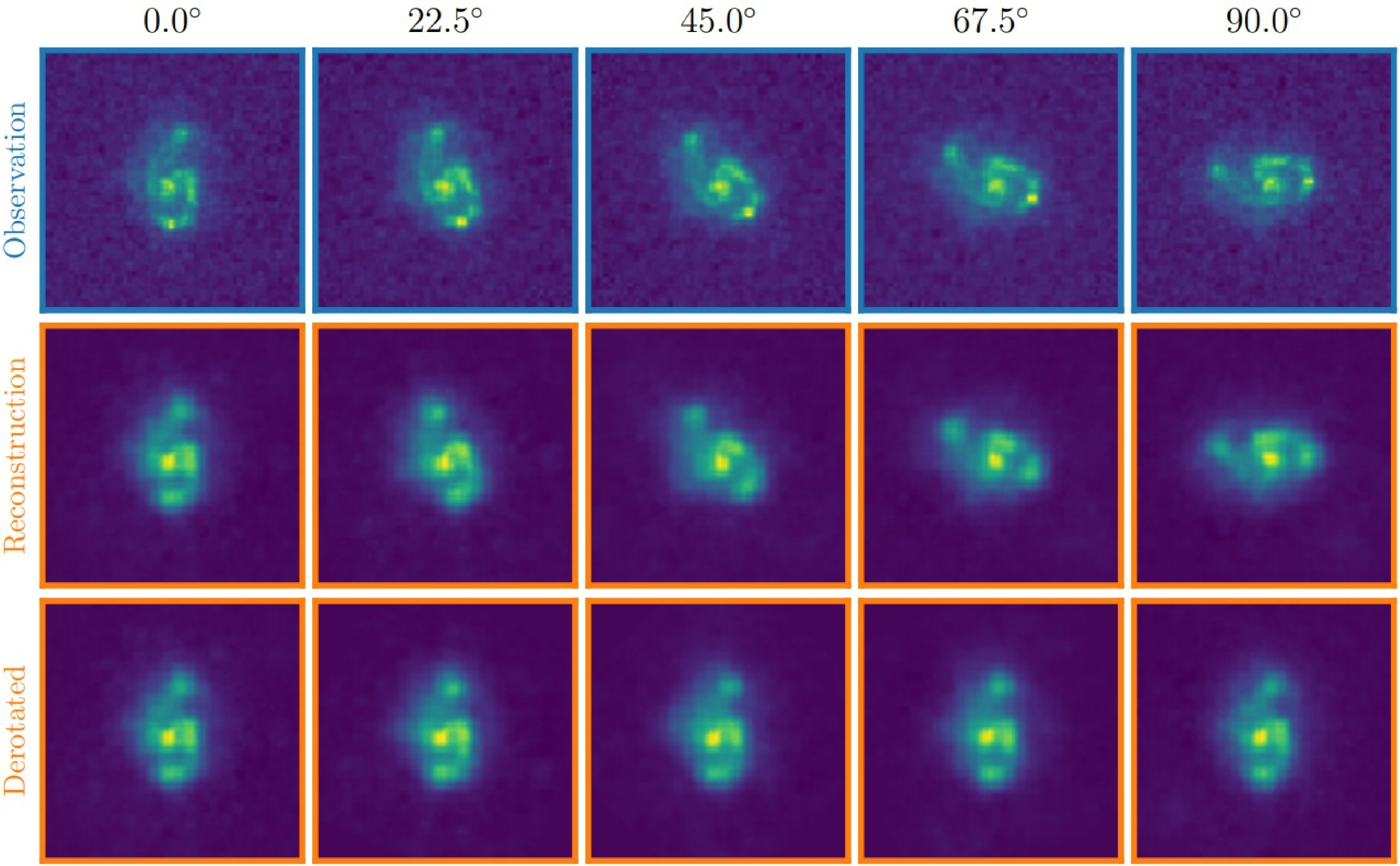
# Summary

- Deep neural networks are power flexible function approximators with many applications.
- Gradient decent is one of the key components of training neural networks. They are useful for high-dimensional optimization, sampling and variational inference.
- First steps towards gradient-based lensing pipeline that integrates deep generative models look very promising.
- A.I. technology provides tons of opportunites *for improving physics and data analysis*, still largely unchartered territory

**Thank you**

# Backup slides

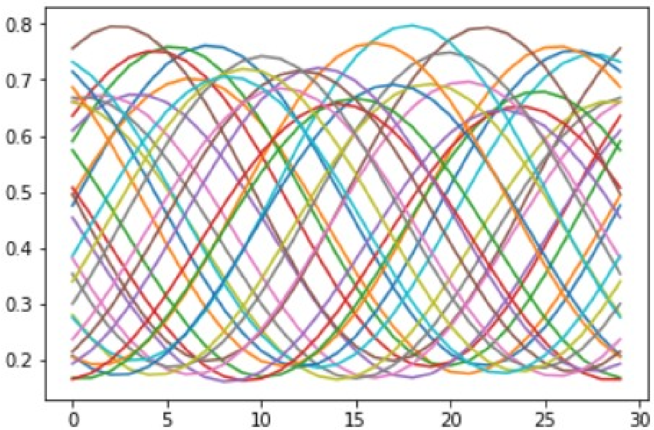
# Rotation test



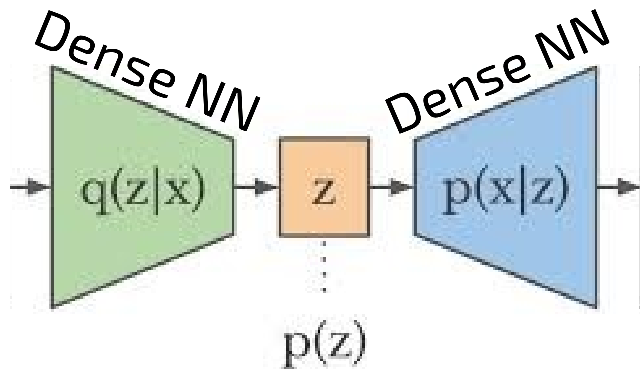
# A simple example for the latent space

**Training**

1000 sine curves

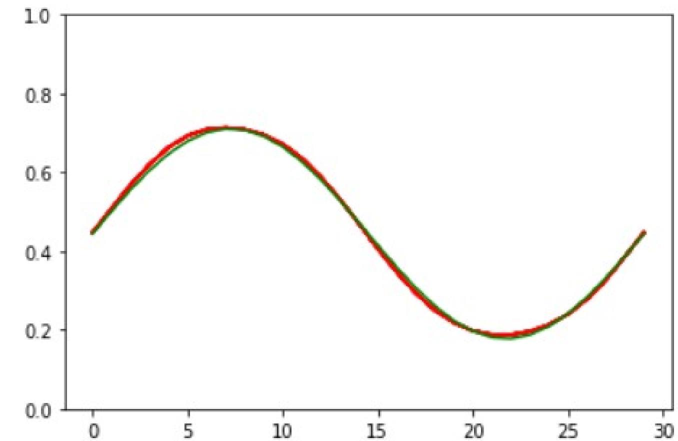


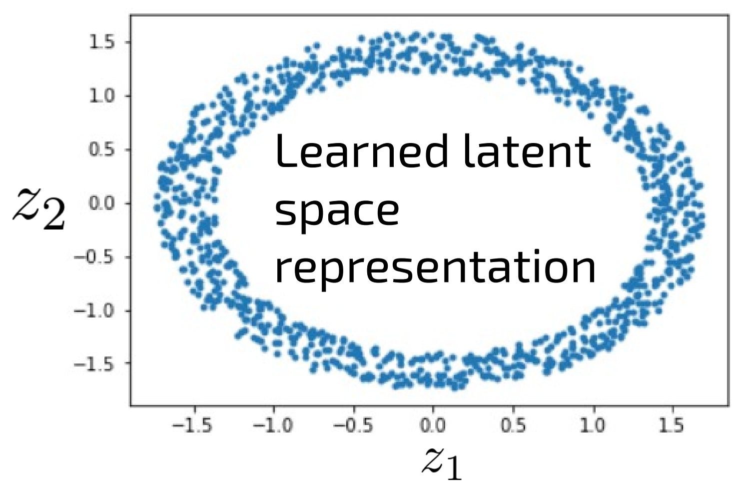
$$f(t) = A\sin(x + \phi)$$




**Reconstruction**

Works reasonbly well






**Learned latent space**

- Periodic variable (phase)
- Bounded variable (amplitude0