



DEEP LEARNING IN COSMOLOGY AT THE INSTITUT D'ASTROPHYSIQUE DE PARIS (IAP)

Alessandro Manzotti for the IAP team



ILP Day on Machine Learning

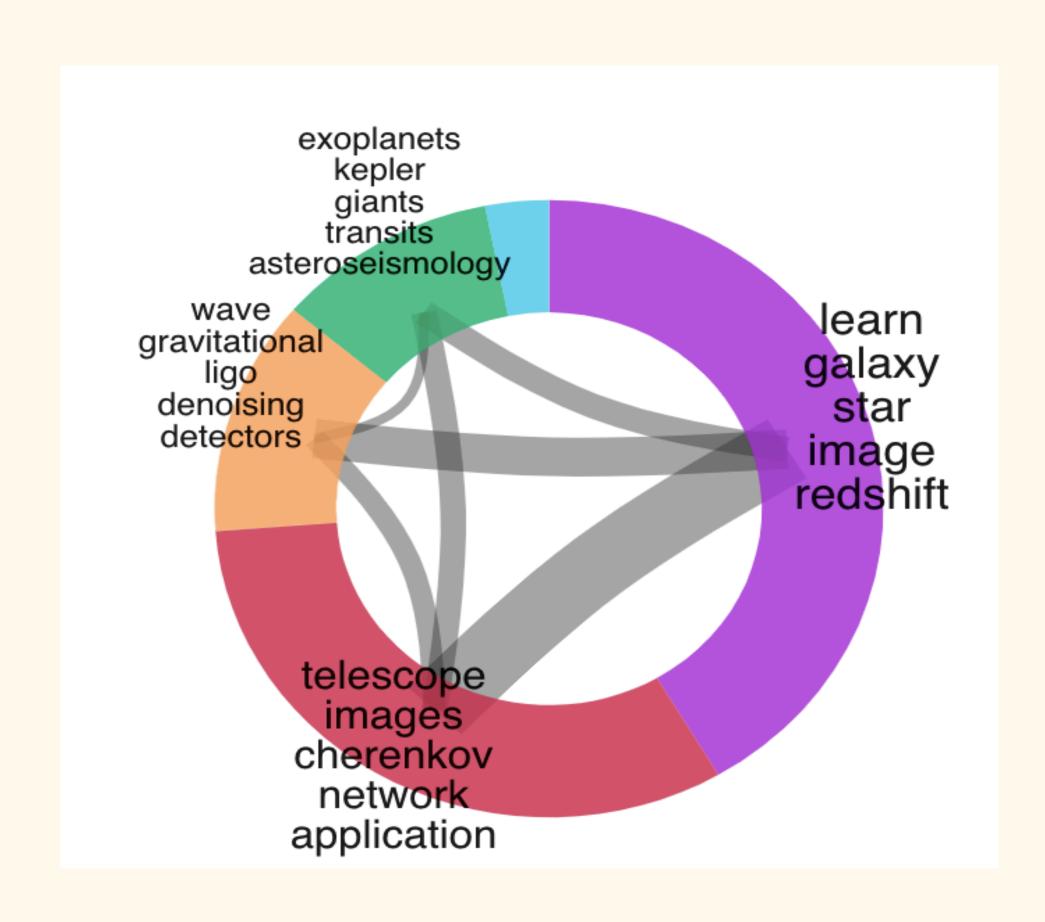


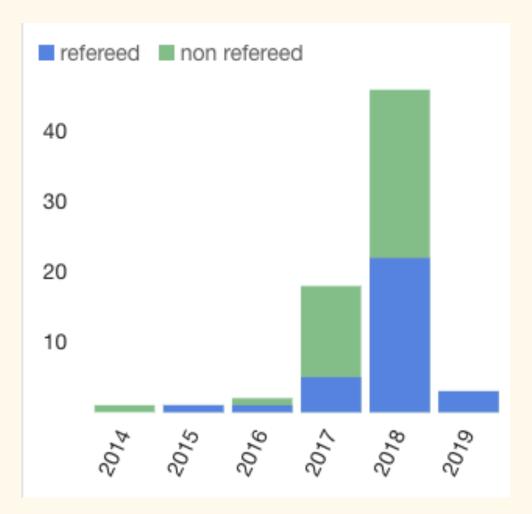


WHERE DOES IAP STAND IN THESE EXCITING PANORAMAS?



ALREADY HARD TO KEEP TRACK OF ALL ASTRO-APPLICATIONS





Arxiv cosmo or methods with "deep learning" in abstract

Practically use them: to automate/improve thing we do not care to understand.

Detect GWs, radio showers, strong lenses. Classify: pulsar or estimate photo-zs

Extract information when the physics/ statistics is too hard.

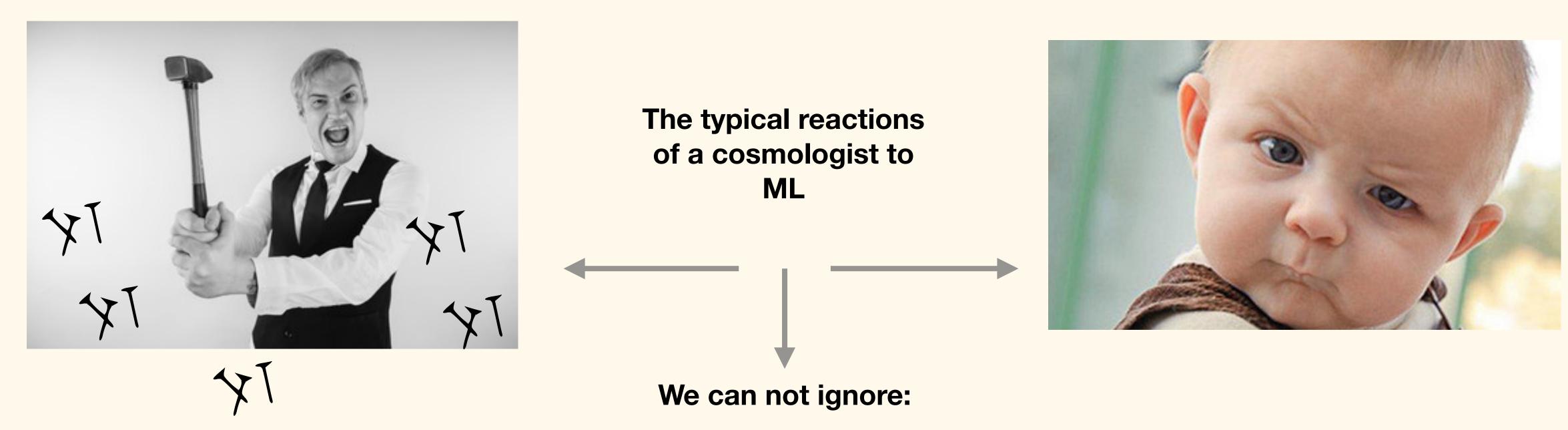
Extract information from weak lensing maps, 21 cm, strong lenses.

Debate inference with deep learning: should we add an extra step?

And use canonical approaches for inference.

Sometimes we do learn physics

SO WHAT SHOULD WE USE MACHINE/DEEP LEARNING FOR?



We are entering the **big data / accuracy cosmology era. Hard** to keep up with good accurate **physical modeling** of physics (we care about)

Hard to model **the systematics** (we probably do not care)

Difference:we can "generate" our datasets.

How not to forget physics?

Can we learn physics? Or just emulator/marginalizator machine?

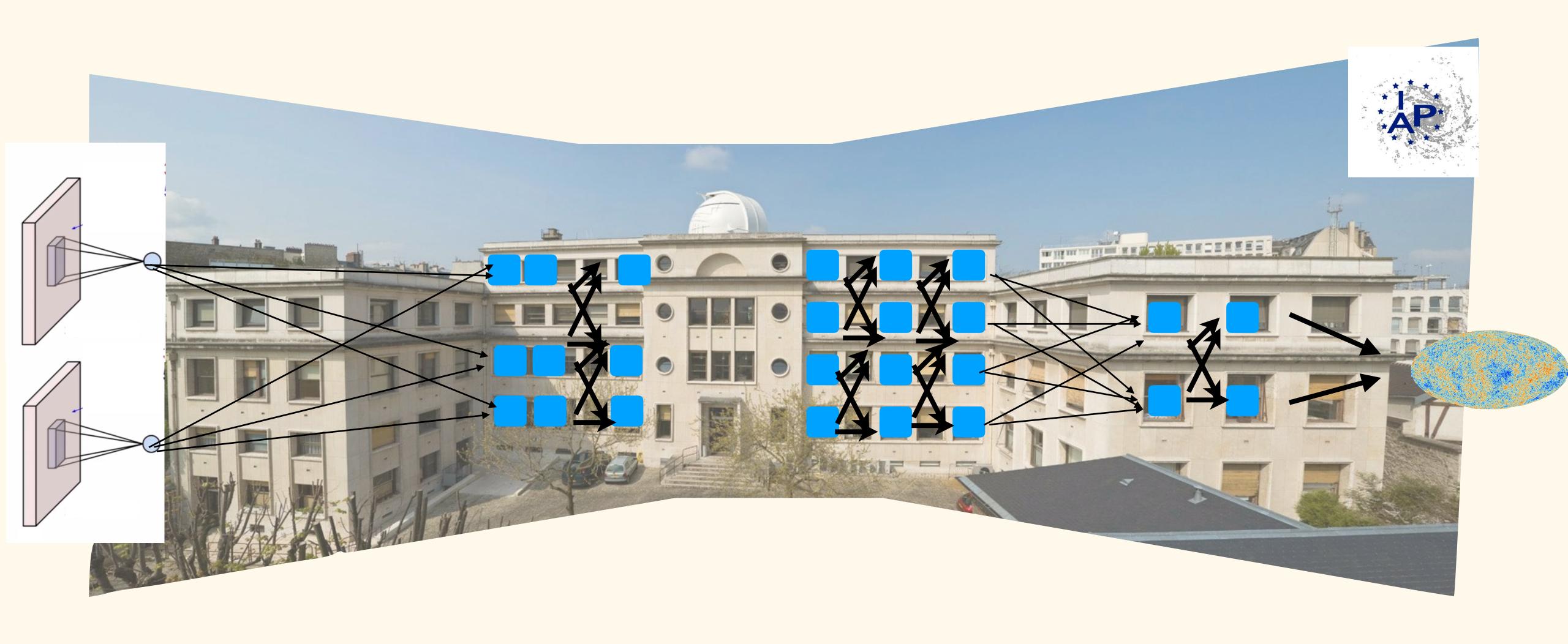
WHERE SHOULD WE GO?

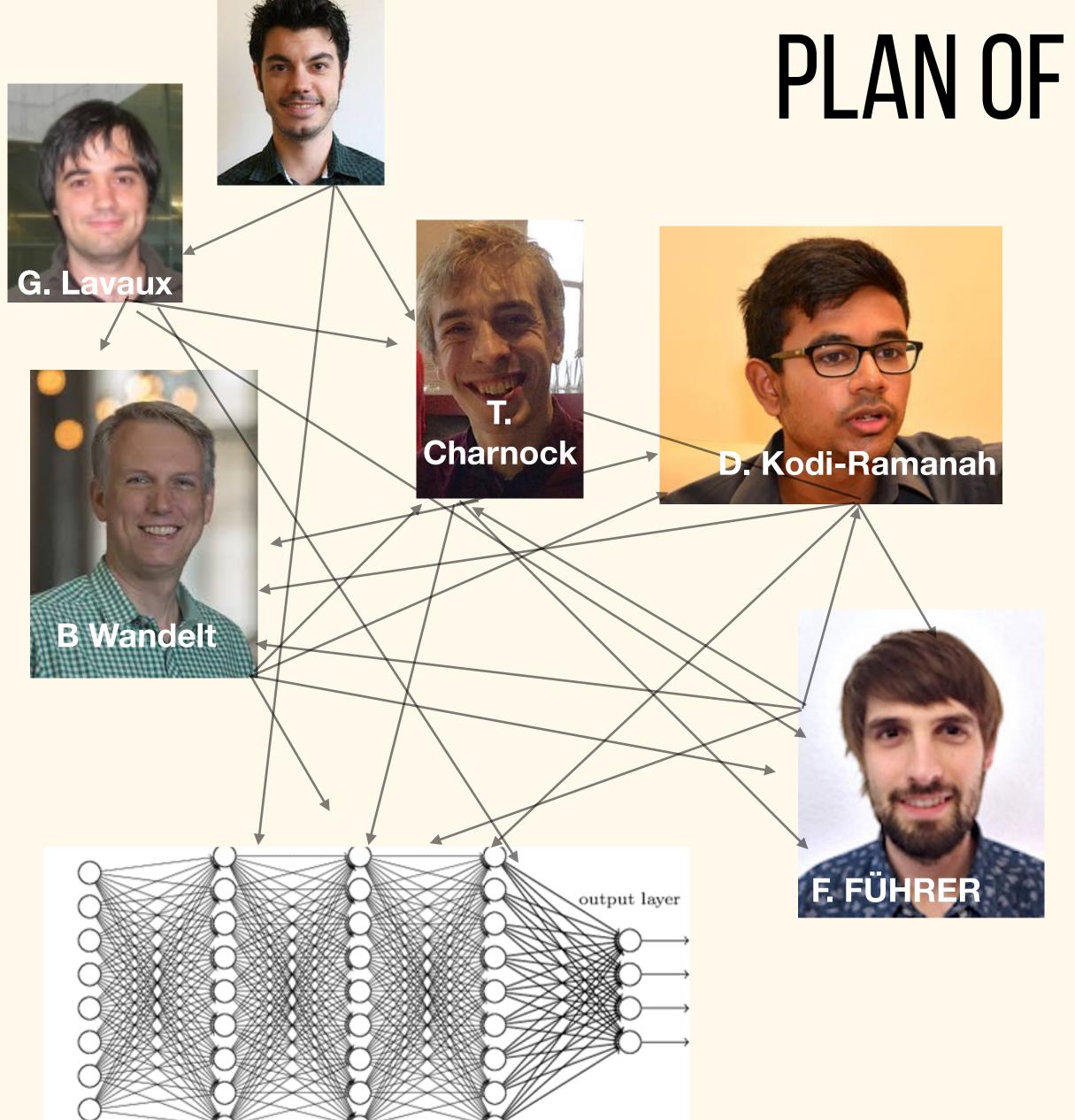




Meeting like this one should avoid this and trigger useful exchanges and collaborations

IAP IS HIGHLY INVOLVED IN DEEP LEARNING





PLAN OF THE TALK:

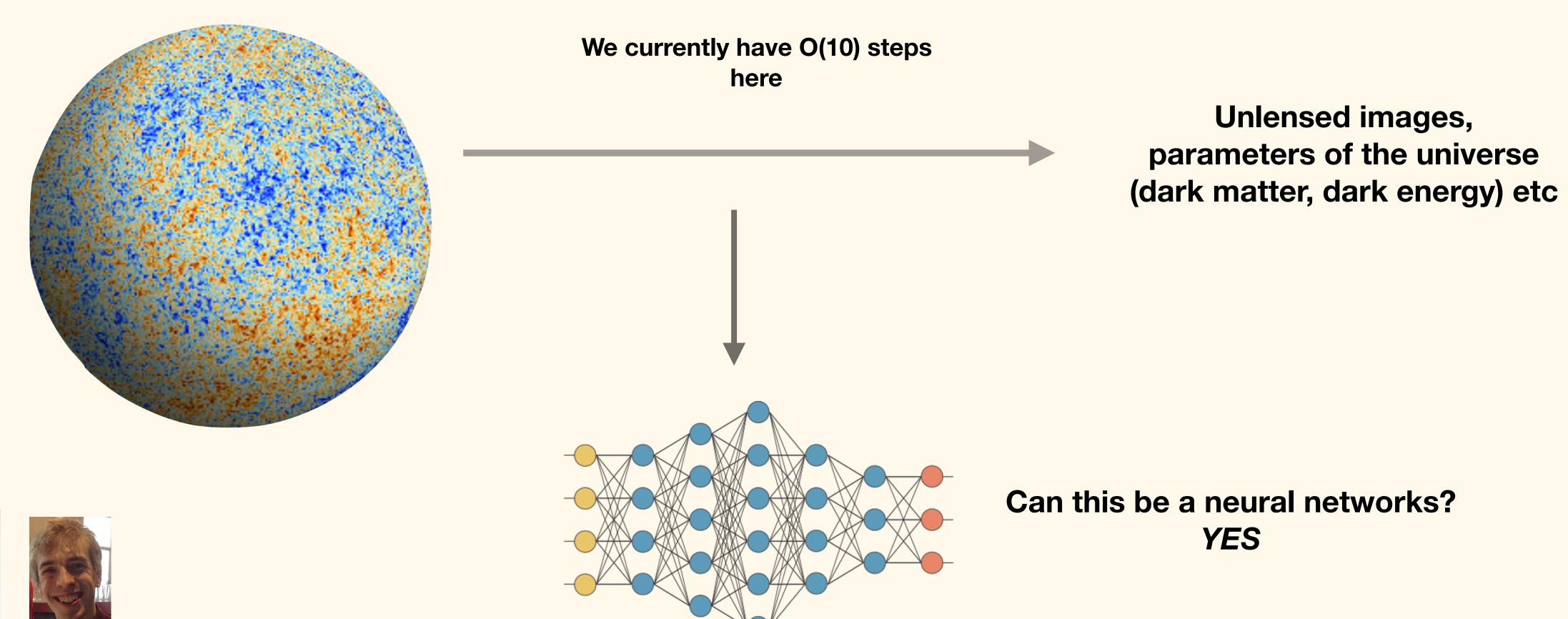
- Extracting Information from Cosmic Microwave data.
- Learn physical parameters from galaxy images and fluxes
- Trigger radio showers and classify type and direction of high energy particles
- Connect dark matter and galaxies

And many more I won't talk about ...

EXTRACTING INFORMATION FROM THE MICROWAVE SKY WITH NEURAL NETWORKS

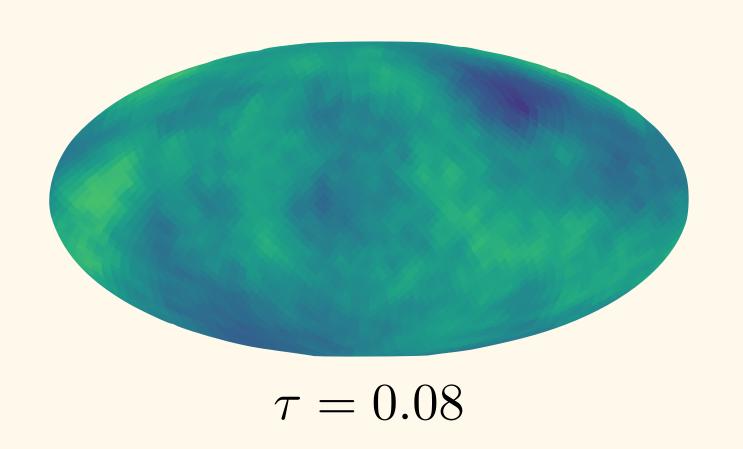
The idea is to avoid as many intermediate steps as possible. Learn directly from microwave anisotropies in the sky.

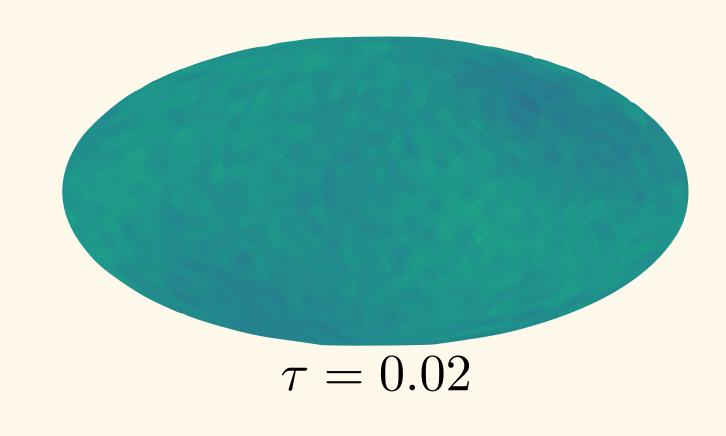
Raw data, noisy maps of the sky



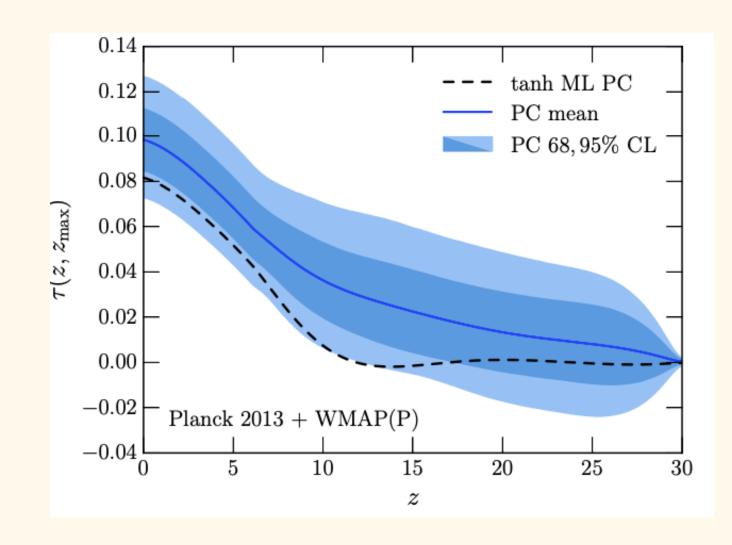
OPTICAL DEPTH I.E. HISTORY OF STARS FROM THE CAB

Learning the reionization history is important per se and crucial in order to constrain other parameters because of degeneracy (like neutrino mass)

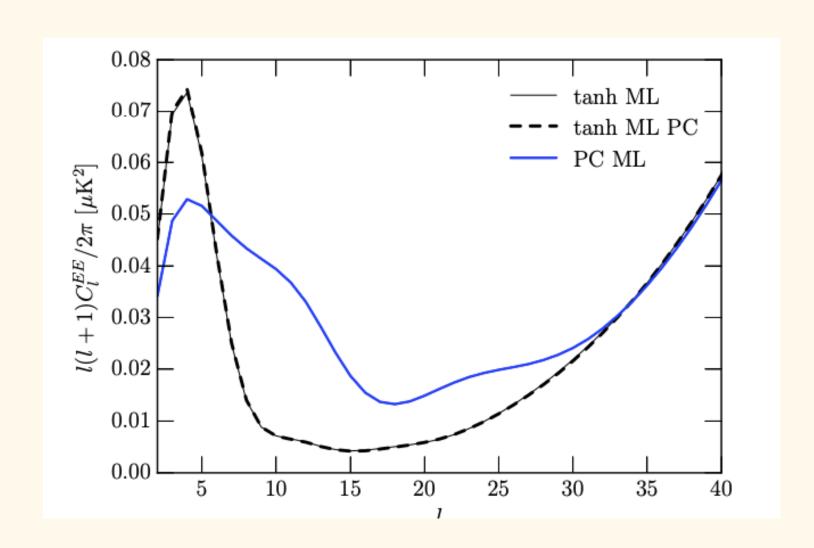




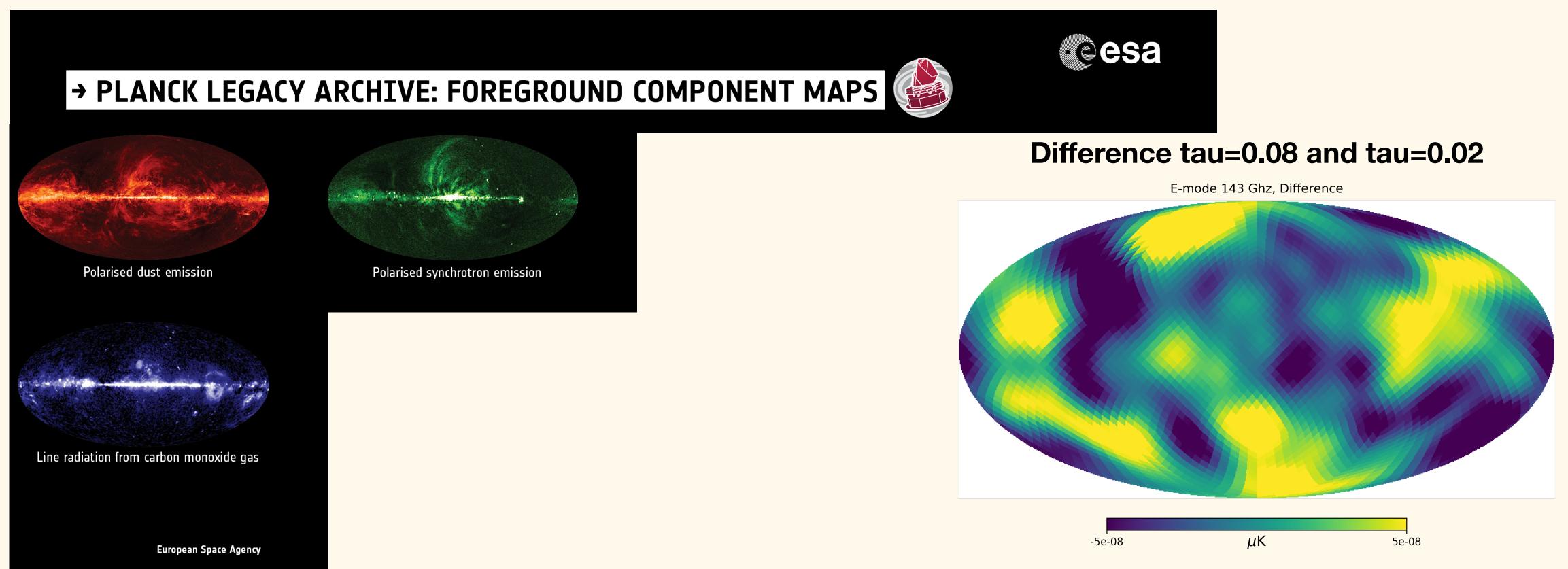
Large scale CMB is very sensitive to the history of deionisation!



Different history of reionisation, Gives different large scale CMBs



CONTAMINATED BY SYSTEMATICS, HARD TO MODEL THE CLEANED DATA



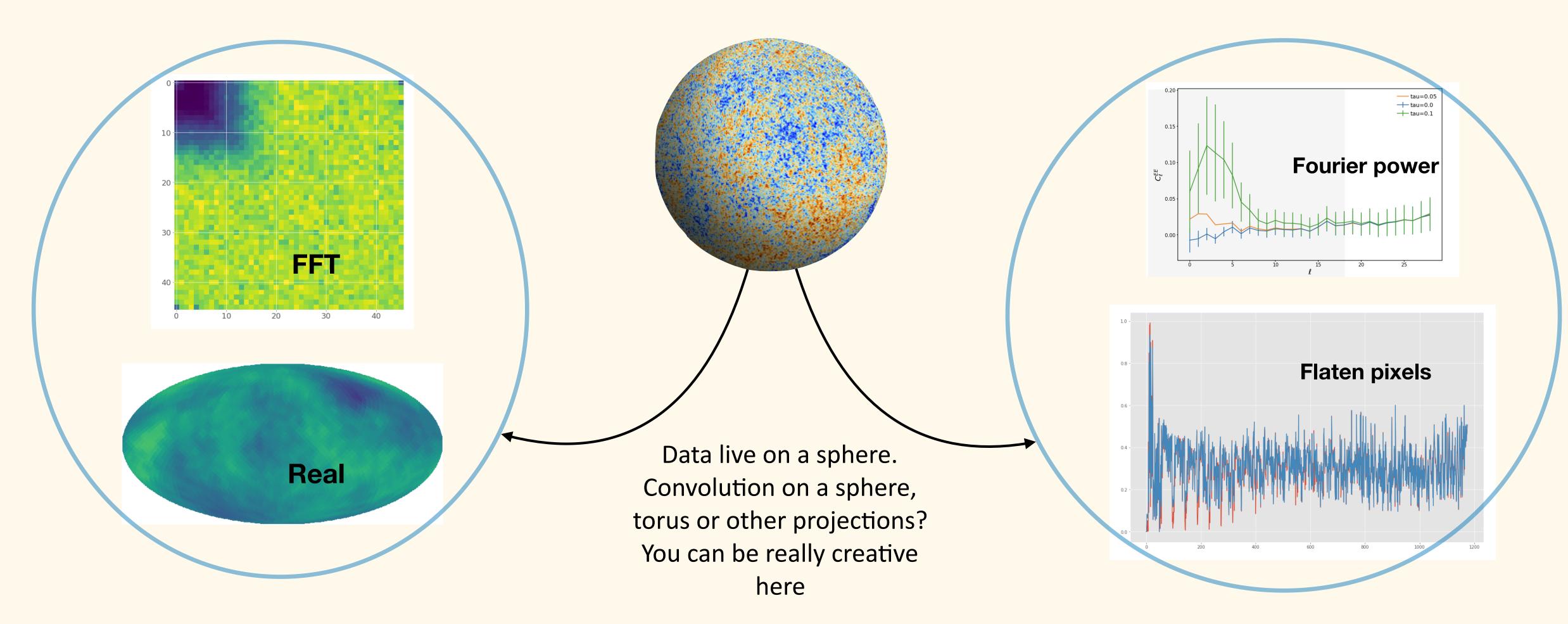
You have to infer the likelihood using simulations. A good application for machine learning.

Also large scale —> low resolution, less computational power needed

WE CAN GENERATE OUR TRAINING SET: WHAT FORM IS THE BEST? TIME FOR CREATIVITY

A lot can be tweaked in the preprocessing part.

The input/training data can be under various form. For this project we had 300 realisation of noise (expensive to generate) and we can generate as many signal realisation as we want.

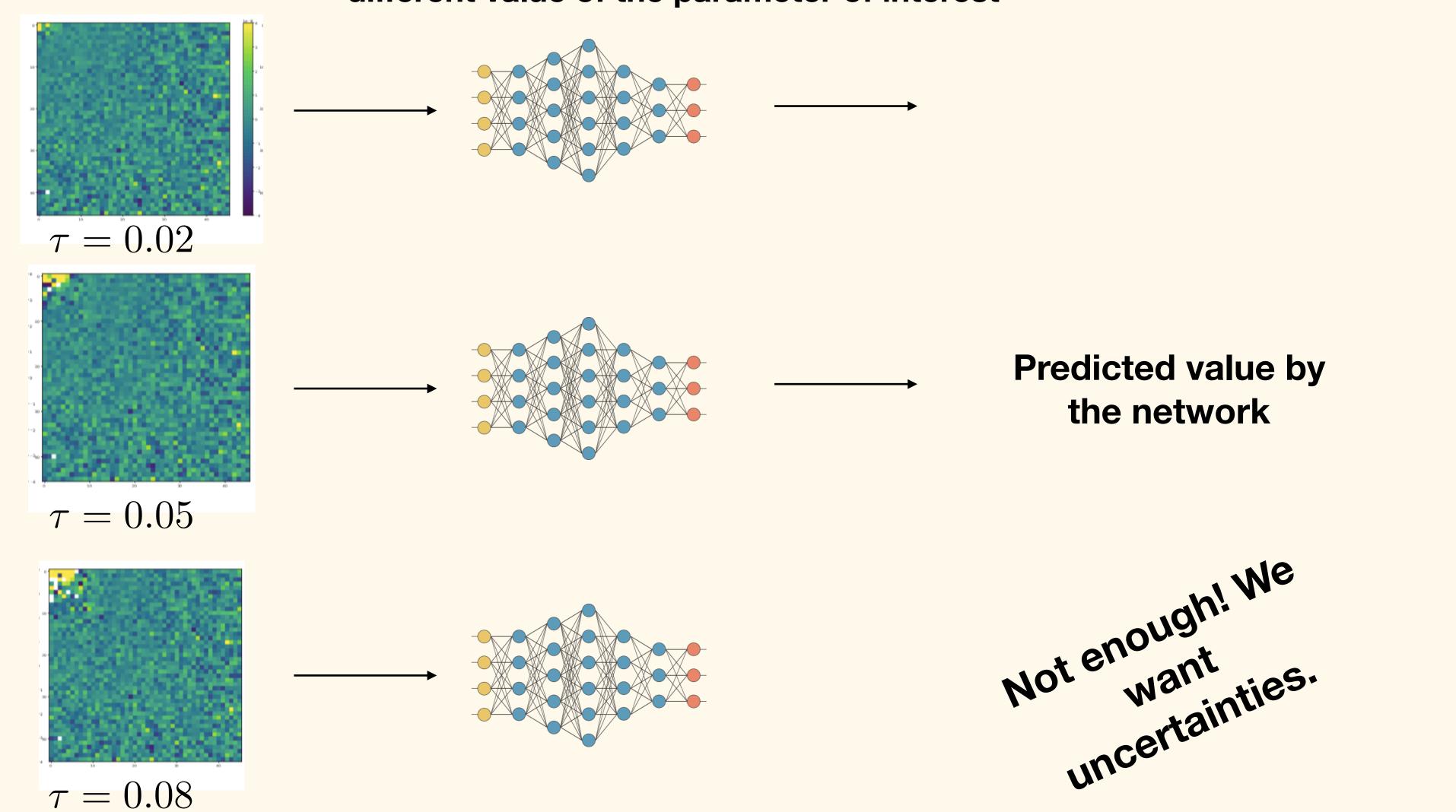


2D images in Fourier and real space

1D input

SIMPLE APPROACH: SIMULATE DATA AND LEARN.

Train the network with simulated (labelled) data for different value of the parameter of interest



Simulate

data with

different

parameter

values

HOW DO WE GET UNCERTAINTIES:

- 1. STANDARD METHOD?
- 2. MIXTURE DENSITY NETWORKS
- 3. THINK DIFFERENTLY, NN AS AUTO ENCODER
- 4. ??? BAYESIAN NETWORKS?

PREDICTION IS NOT ENOUGH. UNCERTAINTIES WITH MIXTURE DENSITY NETWORKS

Mixture density networks (Baysian neural network might work too)

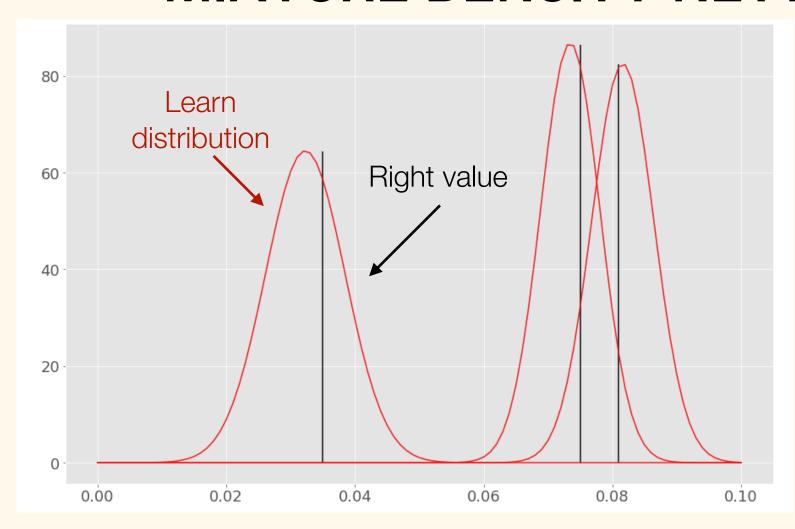
We train the network to **learn not a parameter** from the data, but **a mixture of density distributions**. We are now trying to learn a probability distribution $P(\theta)$ instead of the parameter θ itself.

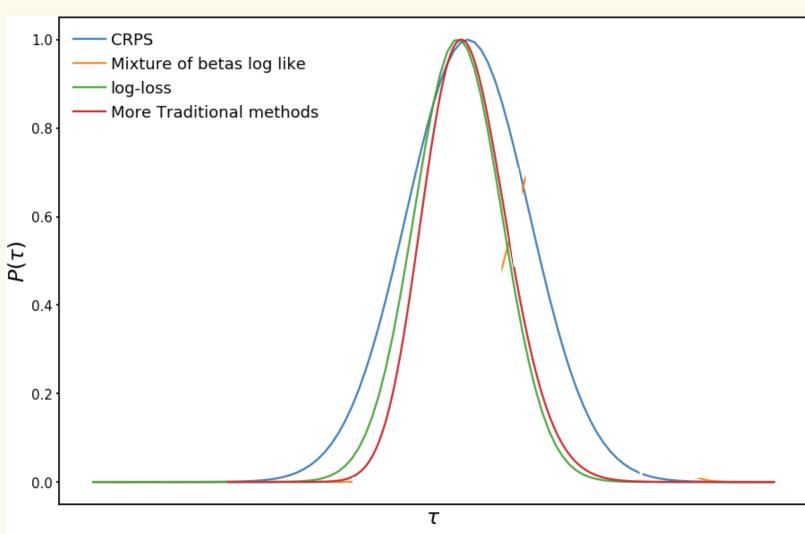
$$P(au) = \sum_{i=0}^N lpha_i \mathcal{G}(\mu_i, \sigma_i, au)$$
 $\left\{ lpha_i, \mu_i, \sigma_i
ight\} \longrightarrow$ Learn by the network

The last activation layer need to enforce the mathematical constrains of these parameters.

The Loss Function used is important CRPS vs log-like but we got similar results.

MIXTURE DENSITY NETWORK WORKS BUT ARE THEY ROBUST STATISTICALLY?



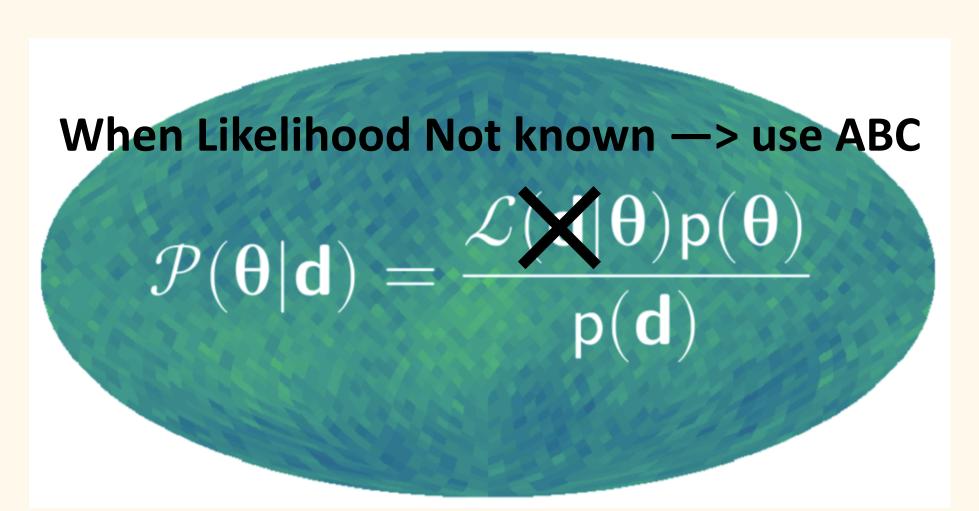


On the test set,
distributions are rightly
peaked and width is in
agreement with
expectation

Comparable with more standard techniques. But faster and more statistically motivated.

It worked nicely but still the statistical properties of the final distributions are unclear. More work and research need to be done.

A STEP BACK: APPROXIMATE BAYESIAN COMPUTATION



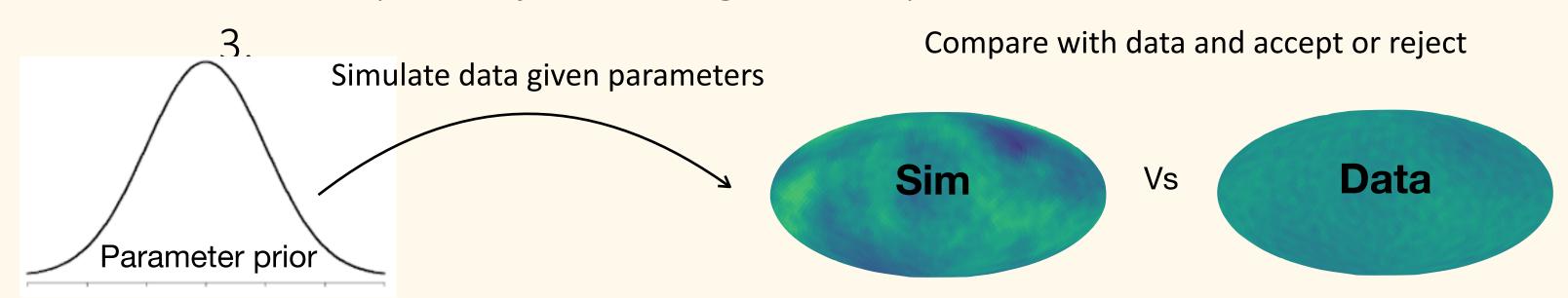
A step back.

If cannot write the likelihood, the probability of the data given the model parameters we can apply the

Approximate Bayesian computation (ABC)

HOW does ABC work

- 1. Simulate data at different θ drawn from $p(\theta)$.
- 2. Accept or reject θ using similarity of simulation and real data.



The space is too big. $O(10^3)$ pixels x several parameters —> impossible: we need a compressed summary statistics.

Not enough for future and in the presence of complex noise

A DIFFERENT APPROACH: NEURAL NETWORK AS A DATA

ENCODER

Starting from 10³ space (or Fourier equivalent)

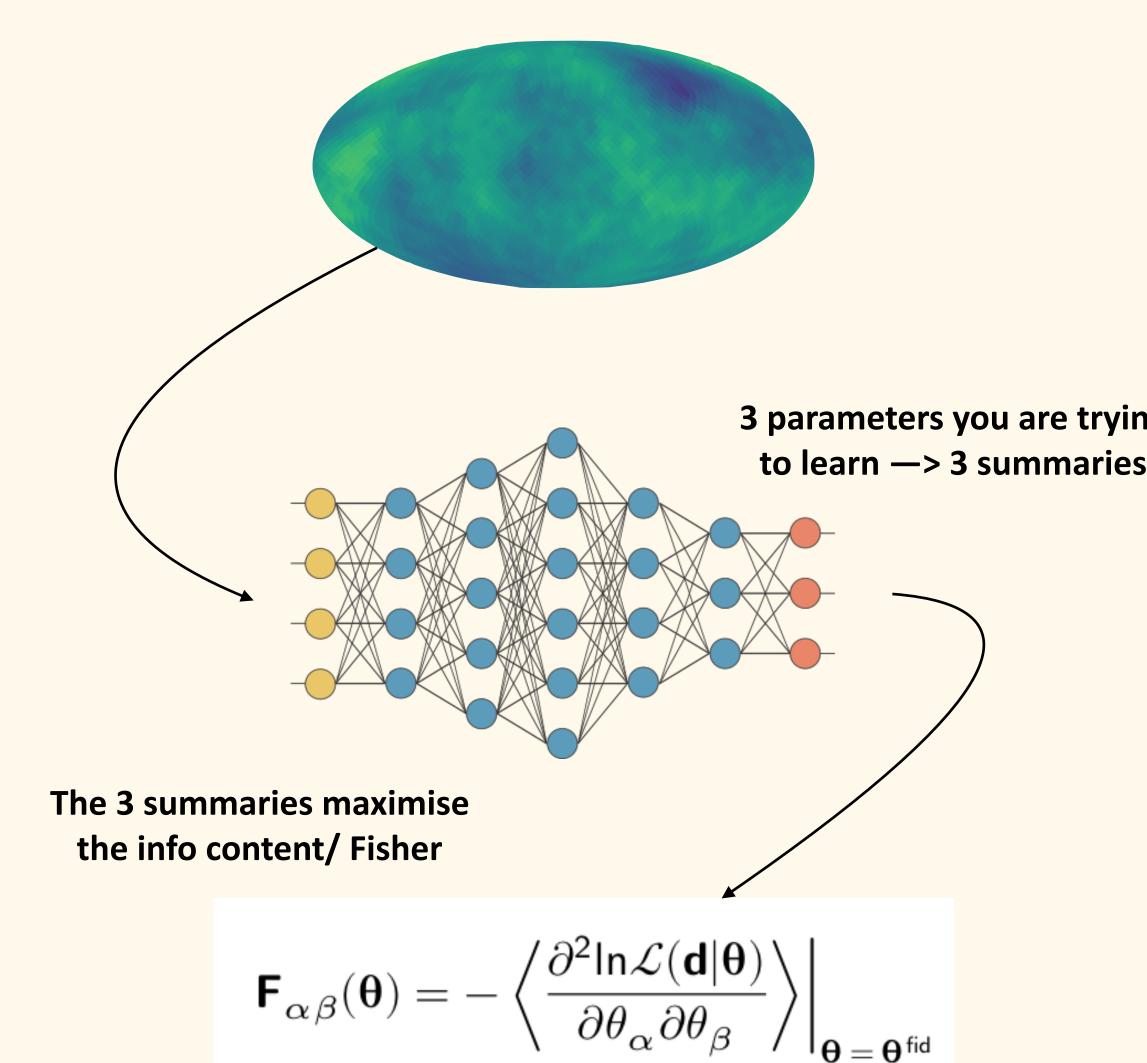
Find non linear combination of pixels that maximise the information about the parameter of interest.

From O(10³) pixels to O(1) numbers!

Maximum information correspond to a maximum Fisher matrix.

Amount of information data d contains about parameters θ .

You want your summary statistic to maximally vary with the variation of the parameter, relative to the covariance.



OPTIMAL COMPRESSION TO 1 SUMMARY FOR EVERY PARAMETER OF INTEREST

Another way of seeing it:

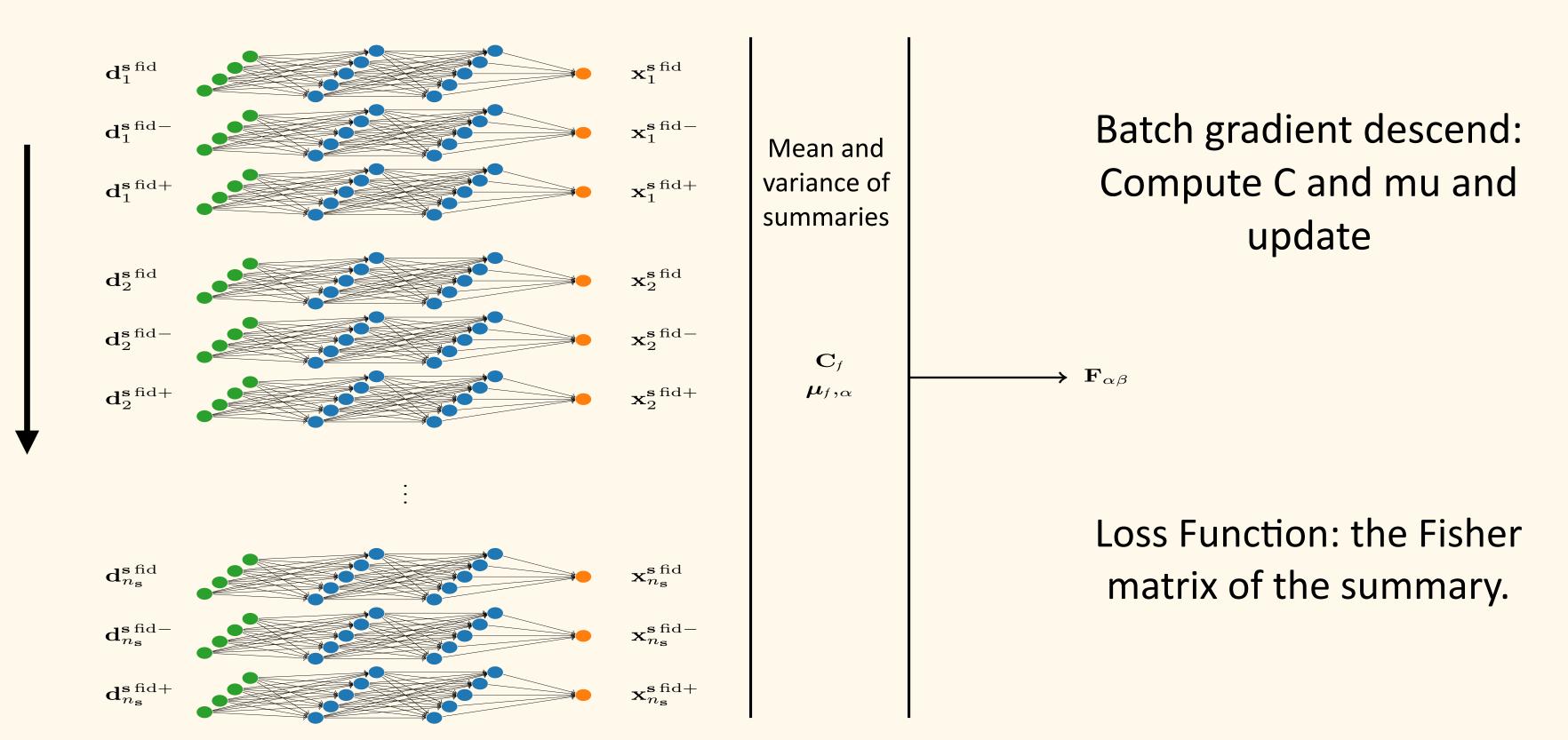
Data Transformation

Find a non-linear function, f: data → x(summary) which transforms the unknown likelihood into a multivariate gaussian

$$-2\log \mathcal{L}(x|\theta) = (x - \mu(\theta))C^{-1}(x - \mu(\theta))$$

Training examples

3 input needed to compute derivatives. We need numerical differentiation, the parameter appear only on the simulation side



IMNN Charnock et al. 1802.03537 on GitHub

LEARN FROM CMB MAPS WITH NN AND ABC:STEPS



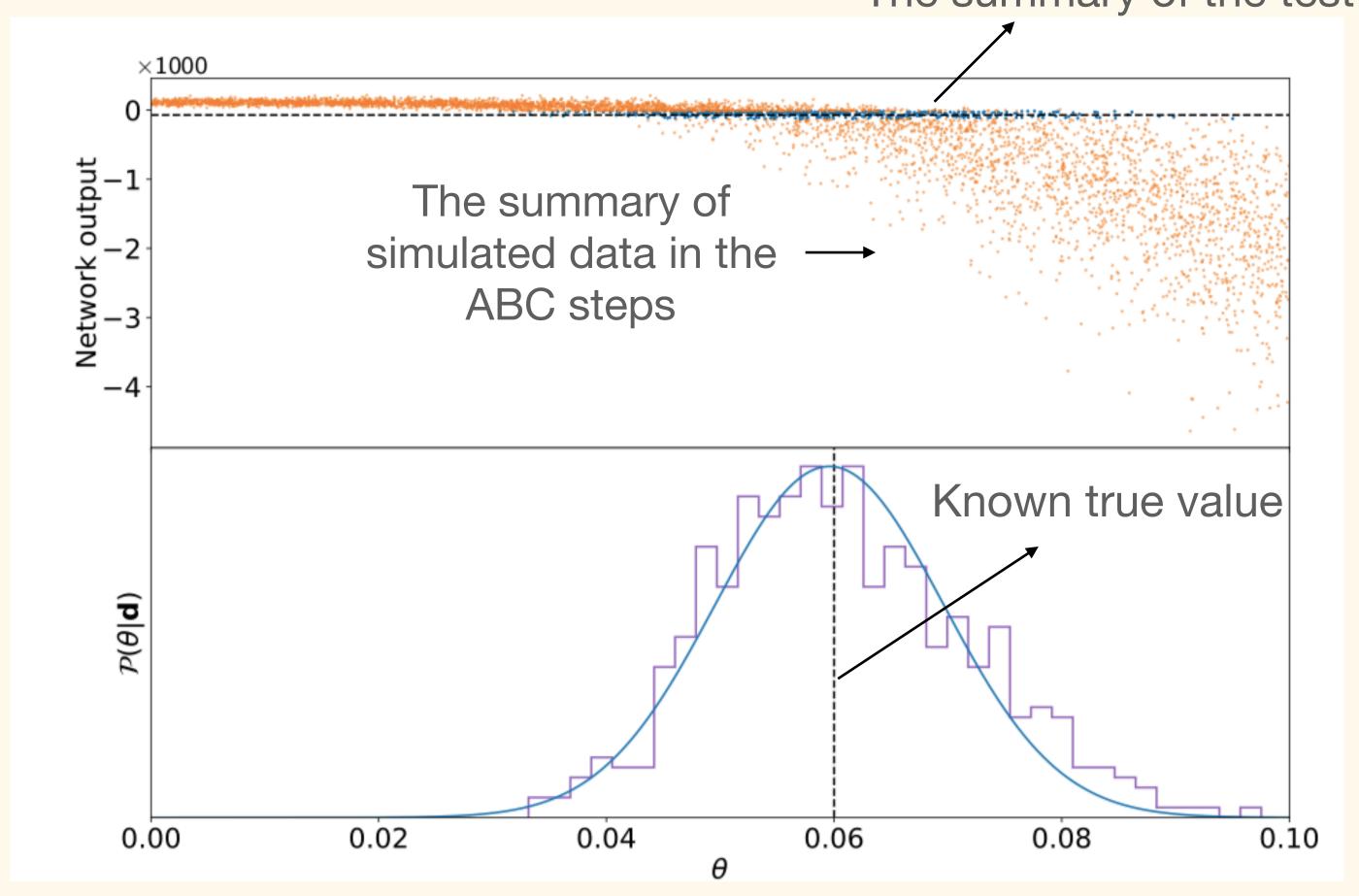
- Make several simulations of data for 3 values of each parameter ds_{fid}, ds _{fid}, ds _{fid+}. We need to compute derivatives
- Train the networks to maximise Fisher information. Test on train and data set looking for any possible problem before moving to data.
- Compress the real data using the trained network
- Apply the ABC: draw θ from p(θ) and create simulation
- Compress simulation
- Accept or reject proposed sample to build approximate posterior distribution

You have a distribution for the probability of the parameter given the data

WE CAN CONSTRAIN PARAMETERS!! TESTED ON SIMULATIONS.

Still working on best network architecture and preprocessing (1D,2D, 3D?) but results look promising.





The summary of simulated data with parameter=0.6 are closer to the test set.

As expected since true value fo r test set is 0.6

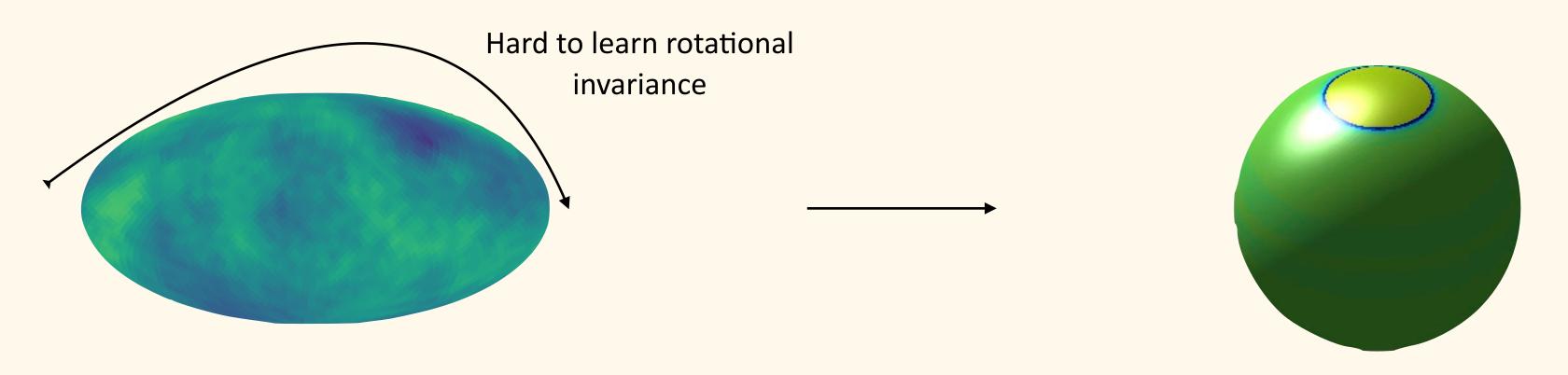
A new questions for cosmologist: how to test a nn based process before applying to the data??

SPHERICAL CONVOLUTIONAL LAYER WOULD BE COOL FOR ASTRONOMY!!

For satellite images our data naturally lives on a sphere. While the signal is rotationally symmetric most of the noise is not.

Also every **projection** of a sphere into 2D **lead to distortions** that reduce the performance of the network

The **natural way** to approach this is too develop a **spherical convolutional networks**. Unfortunately the Fourier theorem that helps in 2D does not hold on a sphere



Implemented in Tensorflow. Not memory/computational efficient.

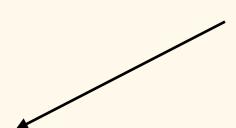
Still the future of this field.

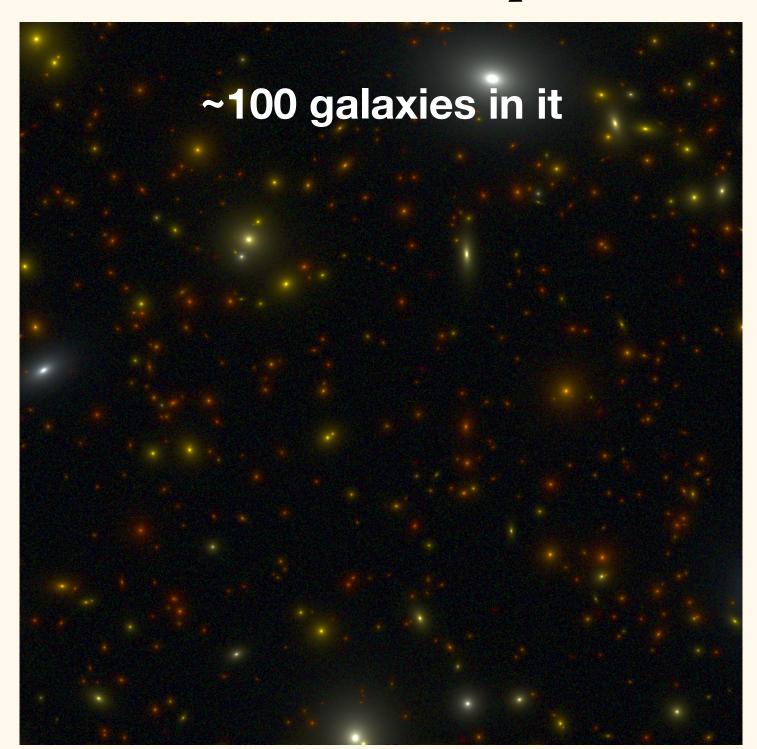
EXTRACTING PARAMETER MODELS DIRECTLY FROM A GALAXY FIELD

Florian Livet, PhD students @ IAP

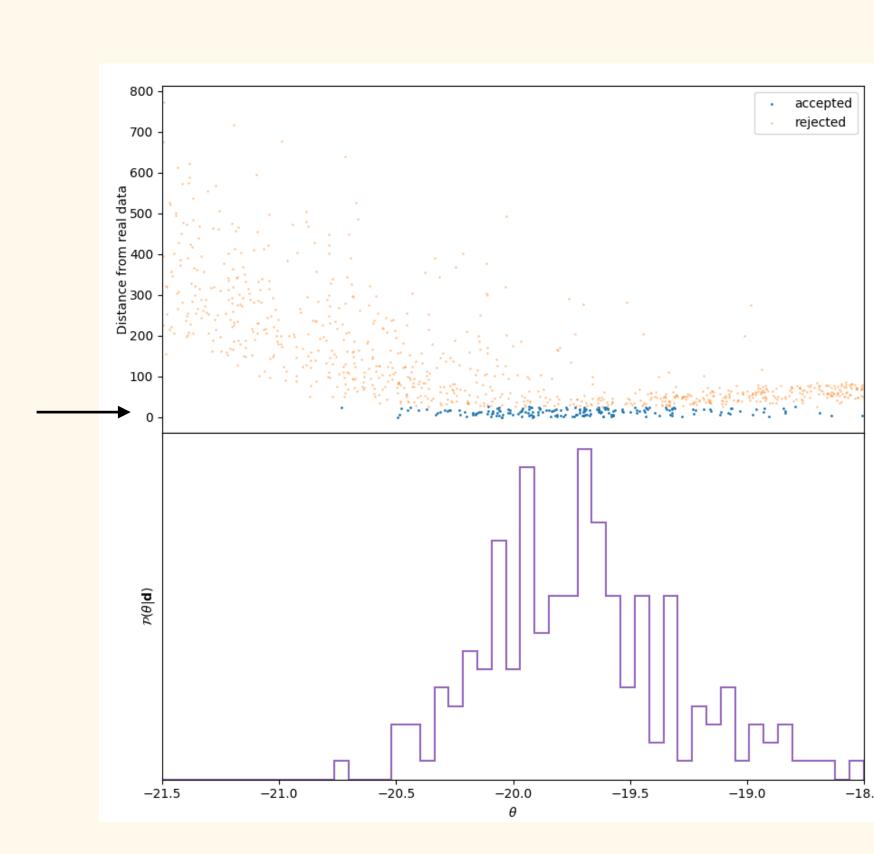
Can we learn about Luminosity functions and other physical properties directly from this?







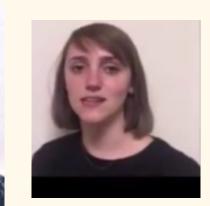
Yes!! with the IMNN we can compress the information and use standard ABC approached on a reduced parameter space.



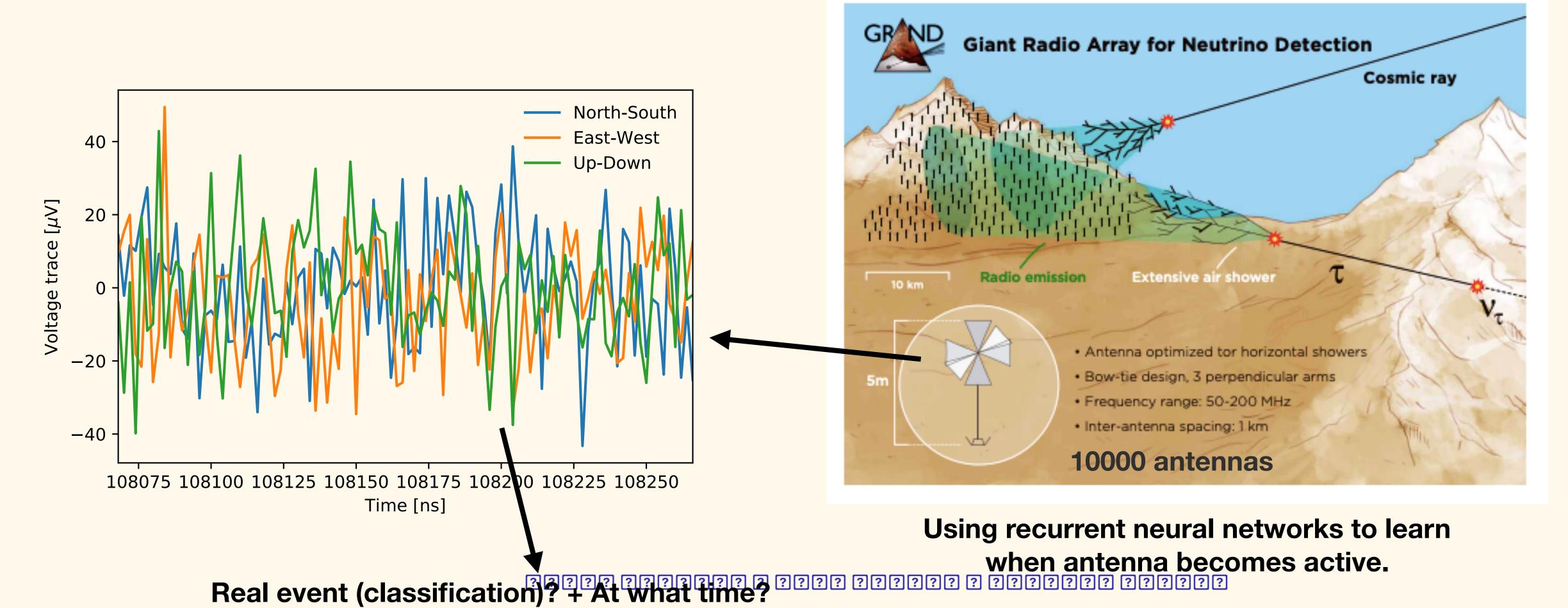


GRAND PROJECT DETECT NEUTRINOS WITH DEEP LEARNING: RADIO SHOWER TRIGGER.





See 1810.09994



Accuracy improved by 10% with first attempt.

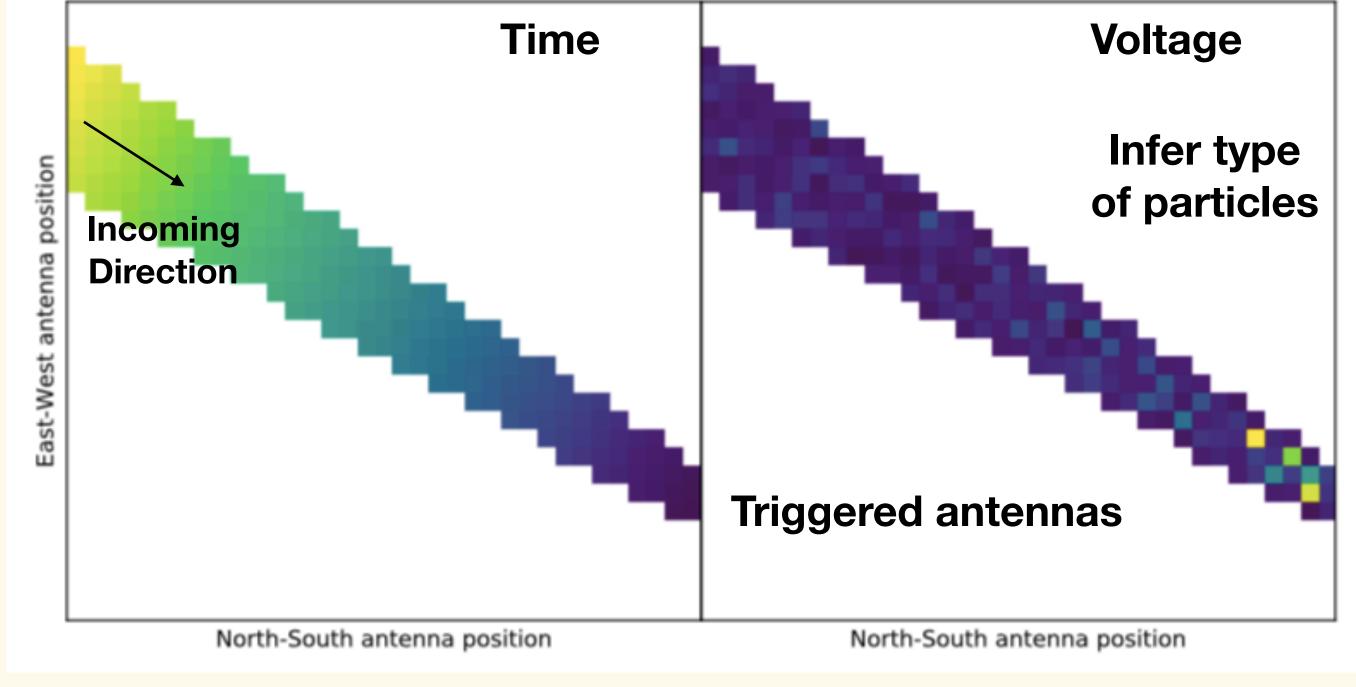


WE CAN DO MORE! WE CAN CLASSIFY THE INCOMING DIRECTION AND PARTICLES TYPE.

Near instant classification.

Use recurrent convolutional neural network to detect particle shower direction, particle type, event energy, etc.

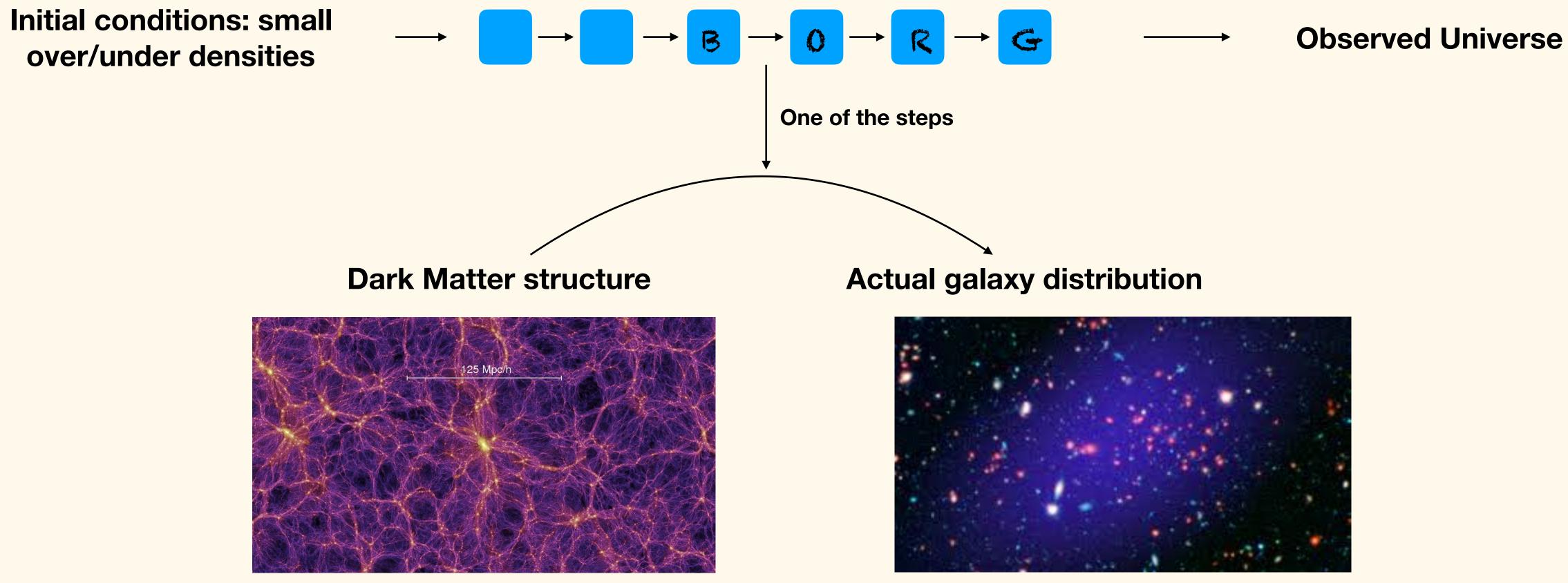




1D cons layers + Cons 2D +fully connected

NEURAL NETWORKS AS A STEP OF FULL FORWARD SIMULATIONS





2 approaches:

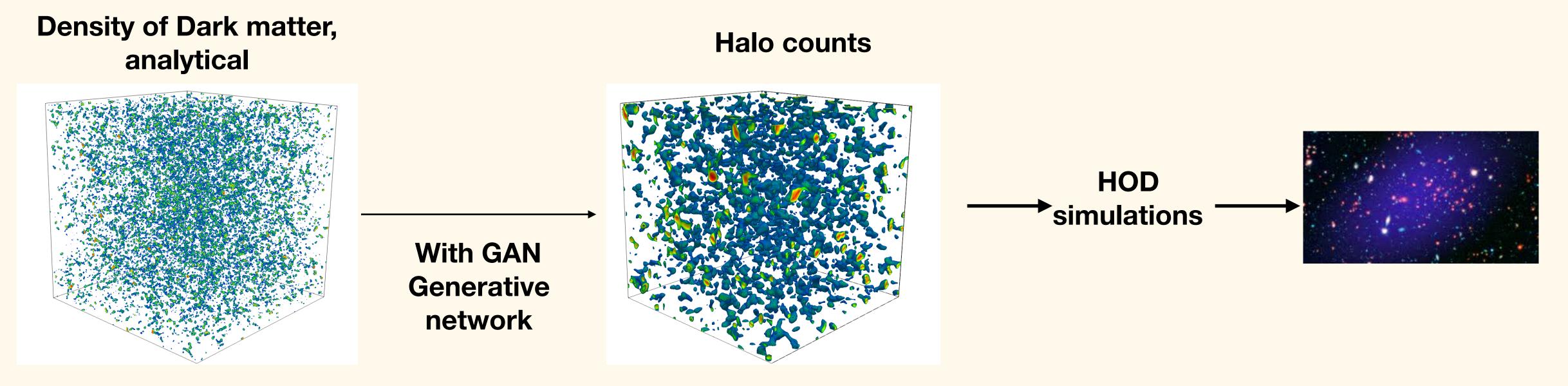
- Learning Halo Impainting with Neural nets
- 2. Use neural nets as a model and sample (with Hamiltonian Monte Carlo) over it

PAINTING HALOS FROM 3D DARK MATTER FIELDS USING GENERATIVE NETWORKS

Save the need to run full particle mesh simulations

Cosmological dependence of matter density field in N-body sim encoded in "cheap" LPT sim

Neural network → deterministic transformation to populate statistically the density field with halos

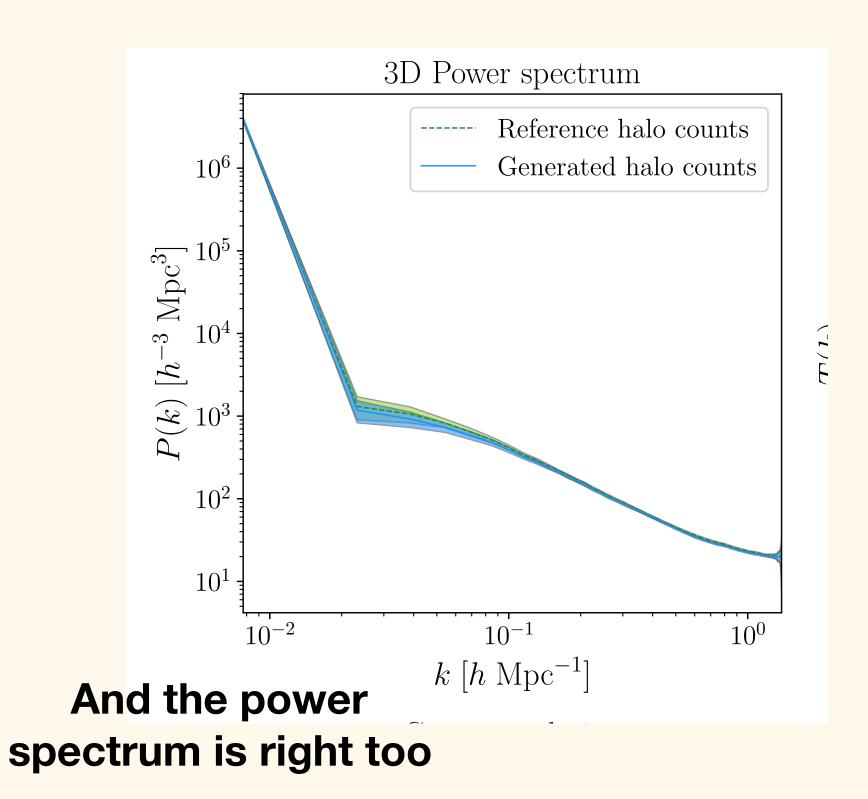


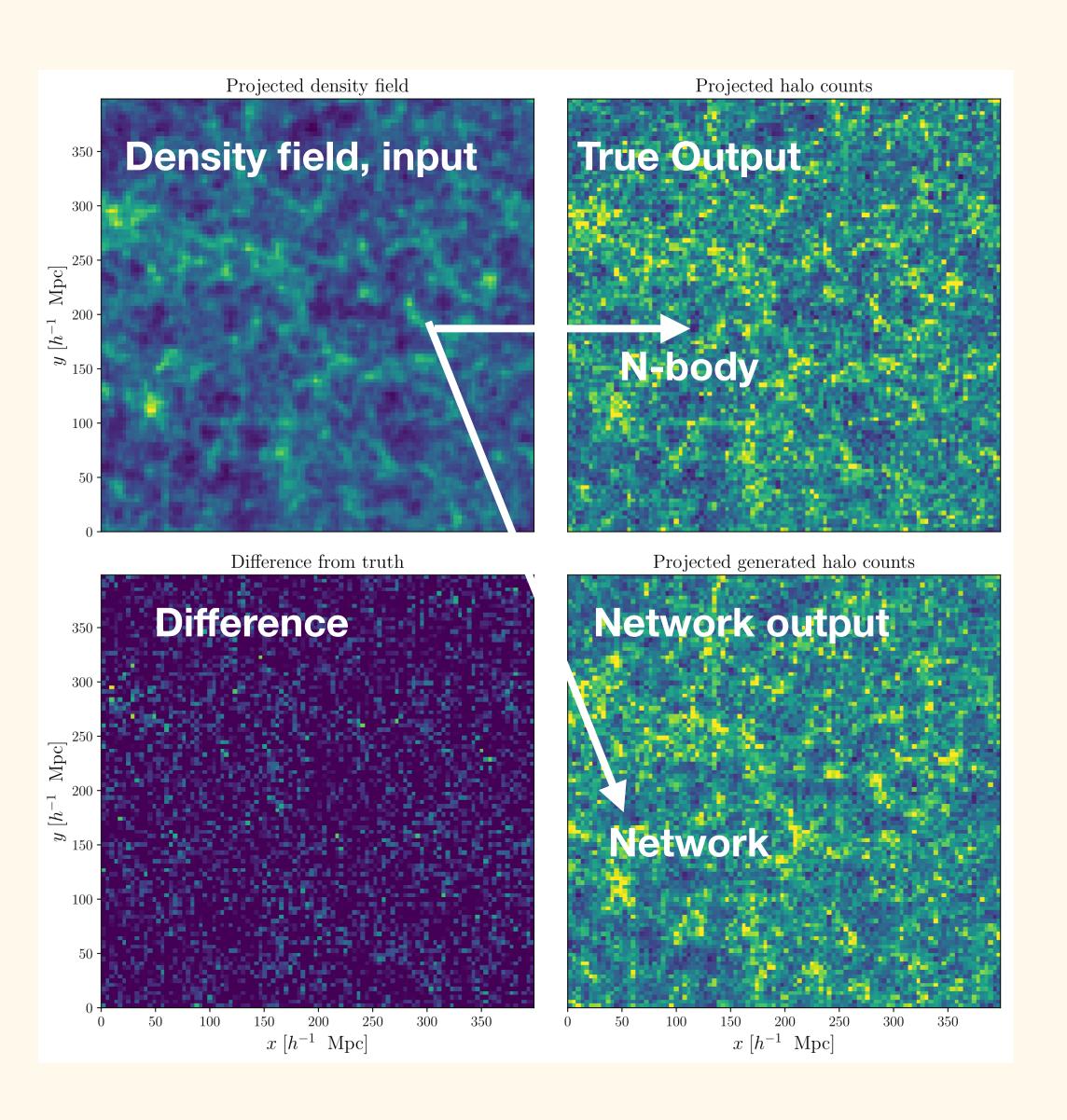


RESULTS: THE NETWORK CORRECTLY PREDICTS THE TRUE HALO FIELD.

Investigate performance of network qualitatively (visual comparison) and via summary statistics

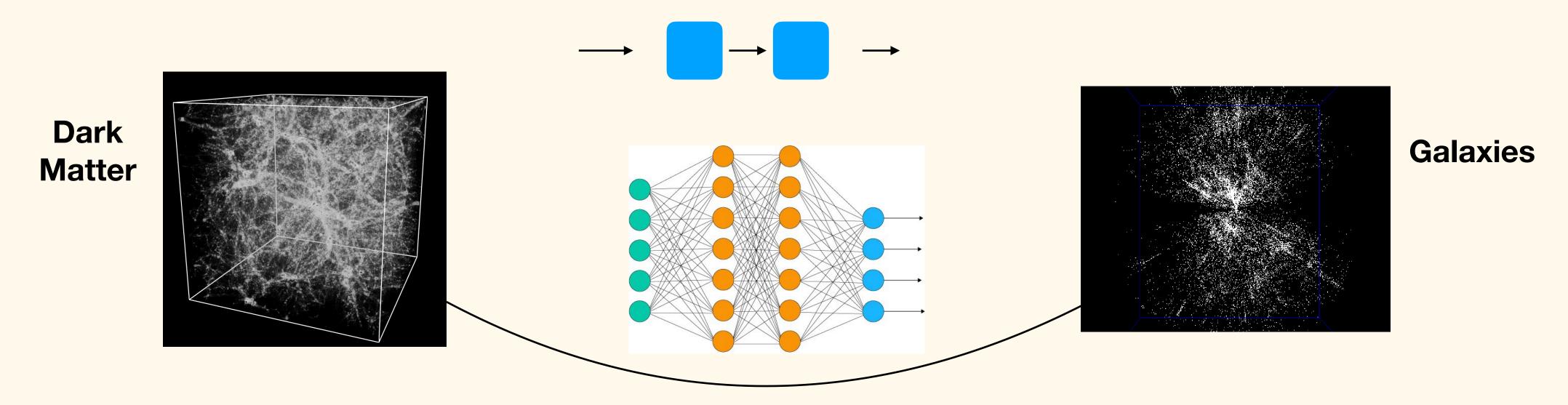
Generative network capable of mapping complex structures of cosmic web to halo distribution





SAMPLING ON A NETWORK AS ON AN EMPIRICAL BIAS MODEL





The network is now your bias model!

The bias model is parametrised by a network. Weights and biases are parameters of the full HMC process we sample over W, and b.

We are using it as the simplest non linear model we can built. Use symmetries to reduce size of NN and parameters.

Weight and biases are treated Bayesianly. We sample on them and we can marginalise.

It is crucial that network are differentiable

DEEP LEARNING IN COSMO AND @IAP, INTERDISCIPLINARY AND RAPIDLY GROWING

Deep learning is extremely powerful, and cosmologist are still trying to figure out how to optimally use it.

The feedback from expert and collaboration is crucial at this point, to support the effort and avoid bad habits to consolidate.

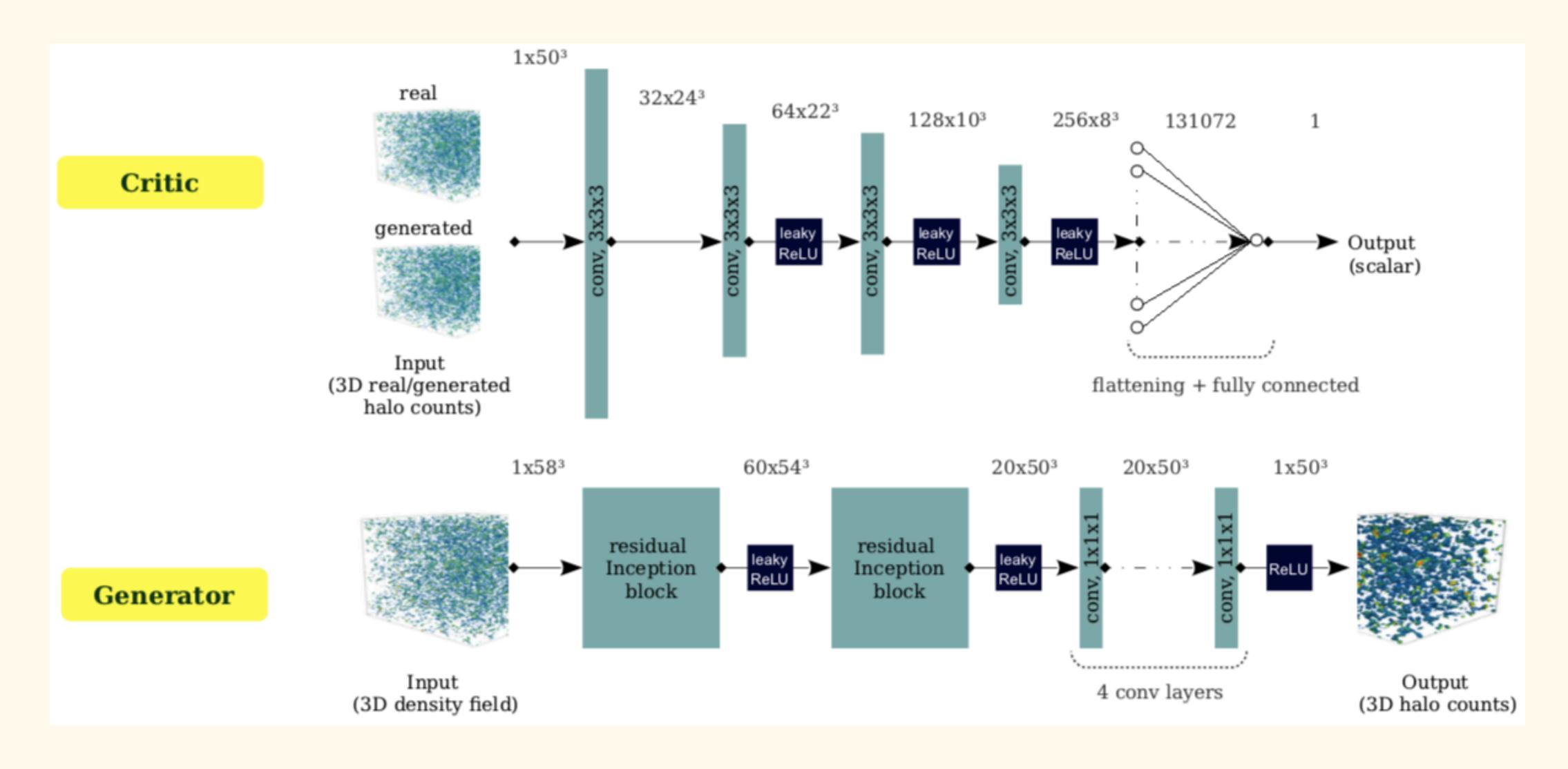
IAP is becoming quickly a big hub of deep learning development and application to cosmology.

We use it to extract model's parameters from data and to emulate automate part of galaxy simulations and triggering.

Are we using it right? What are other application?



WGAN GENERATIVE NETWORK



TESTING INFORMATION MAXIMISER

Consider an experiment which measures $n_{\mathbf{d}} = 10$ data points which are drawn from a zero-mean Gaussian where the variance, $\theta = \sigma^2$, is not perfectly known, $\mathbf{d} = \{d_i \cap \mathcal{N}(0, \theta) | i \in [1, n_{\mathbf{d}}]\}$. The likelihood is written

$$\begin{split} \mathcal{L}(\mathbf{d}|\vartheta) &= \prod_{i=1}^{n_{\mathbf{d}}} \frac{1}{\sqrt{2\pi\vartheta}} \exp\left[-\frac{1}{2\vartheta} d_i^2\right] \\ &= \frac{1}{(2\pi\vartheta)^{n_{\mathbf{d}}/2}} \exp\left[-\frac{1}{2\vartheta} \sum_{i=1}^{n_{\mathbf{d}}} d_i^2\right], \end{split}$$

Best summary statistics of the data

$$x = \sum_{i=1}^{n_{\mathbf{d}}} d_i^2,$$
 F=5

