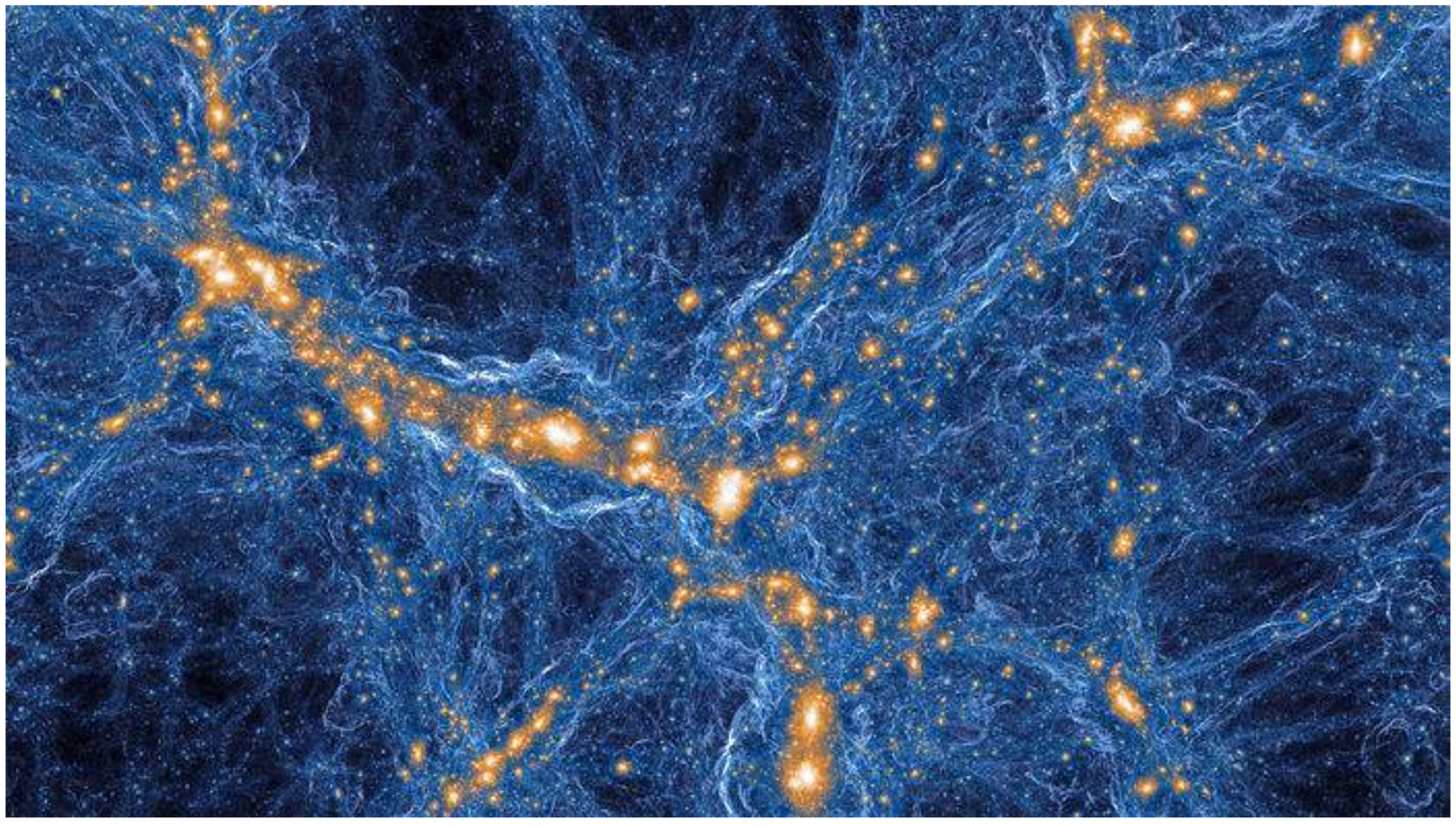
Deep learning the large scale structure of the Universe

Cristiano Sabiu (Yonsei University)

James Jee, Stephen Appleby, David Parkinson, XiaoDong Li and KAML group



Korea Survey Science Meeting
High1 - 11/01/2020

Contents

- ★ **Motivation**
- ★ **Convolutional Neural Networks (CNN)**
- ★ **Information from CNN - Gaussian case**
- ★ **Realistic training data (model)**
- ★ **Future Plans**

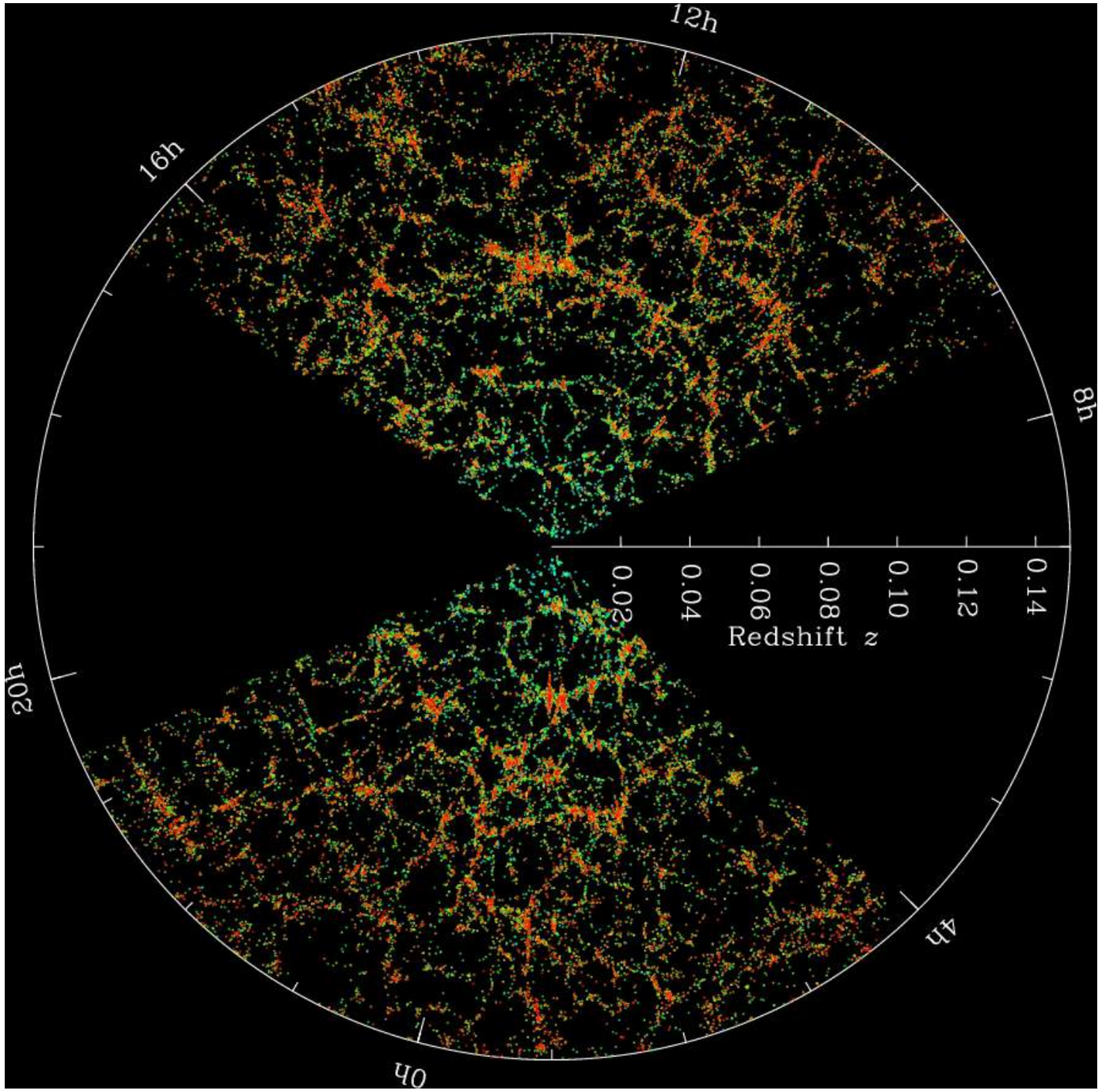
Background

We can now map large volumes of the Universe in 3D using galaxies as tracers of the underlying matter potential

But then what do we do with all these galaxy positions?



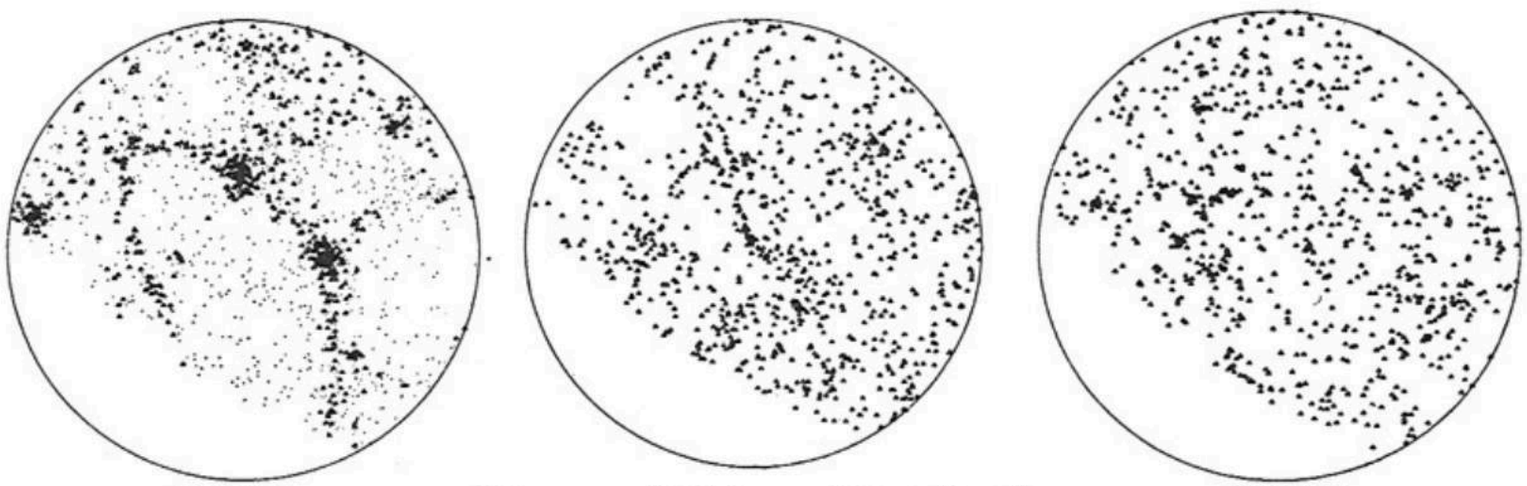
SDSS



Background

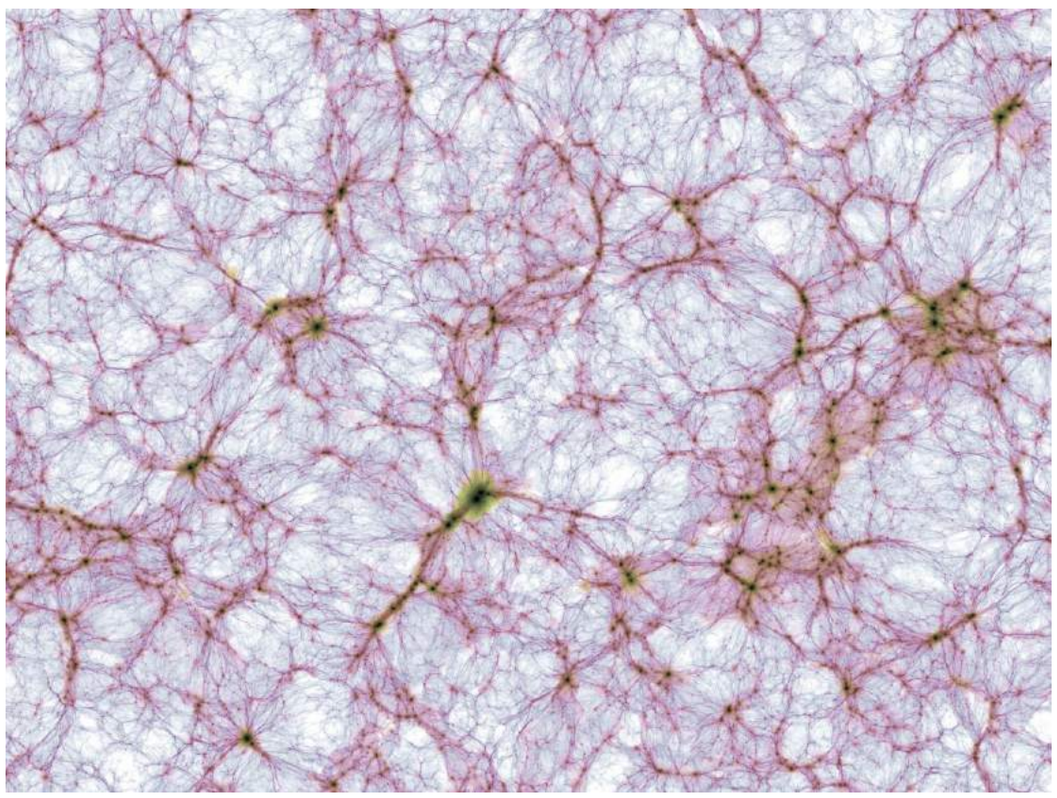
N-body Simulations

1980s
~1000 particles



Credit: S. D. M. White, 1986

Today
~10,000,000,000 particles

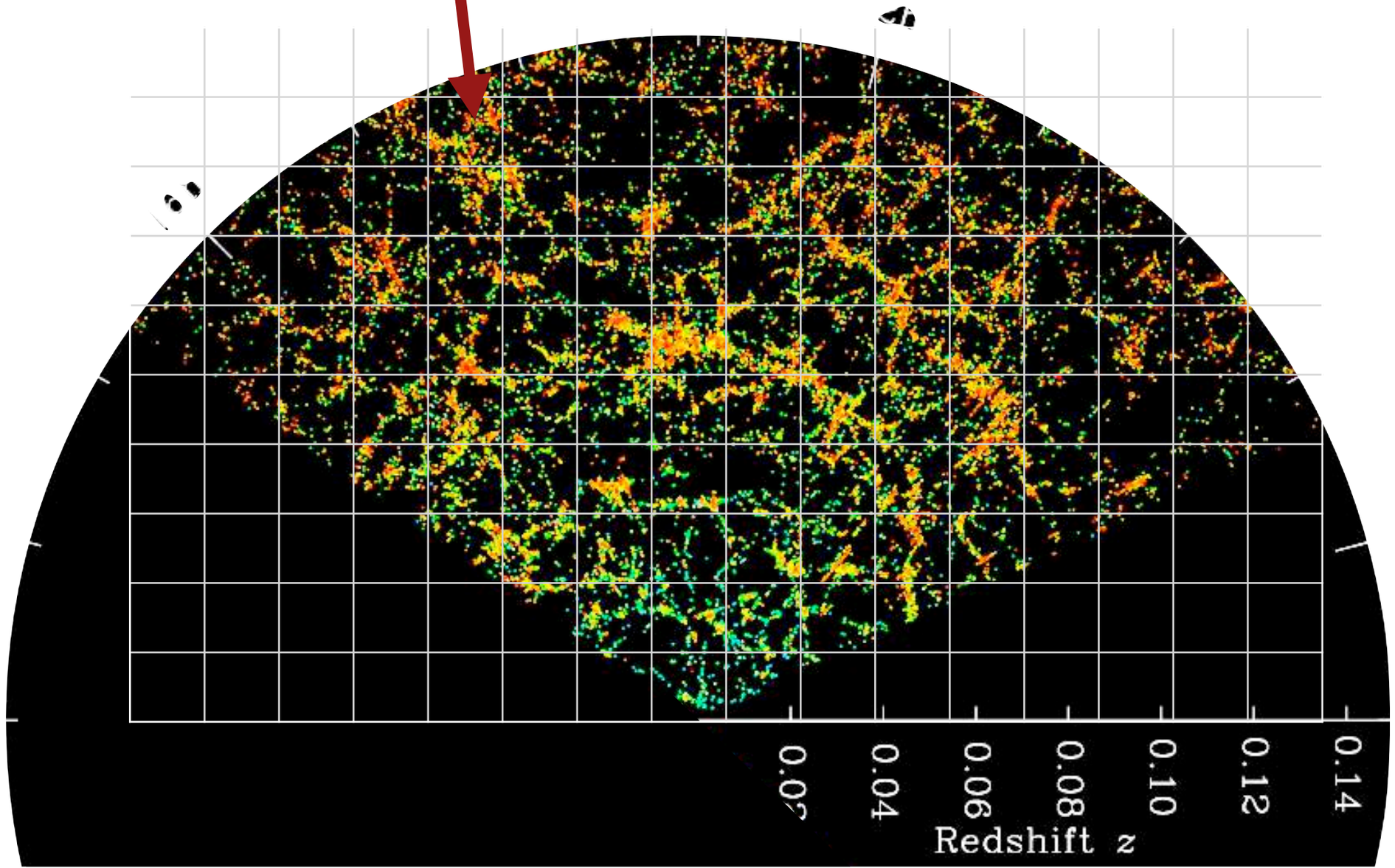


Illustris TNG Simulation

Background

Count galaxies in cells and compute:

$$\xi(r) = \langle \delta(x)\delta(x+r) \rangle_x$$



SDSS

Background

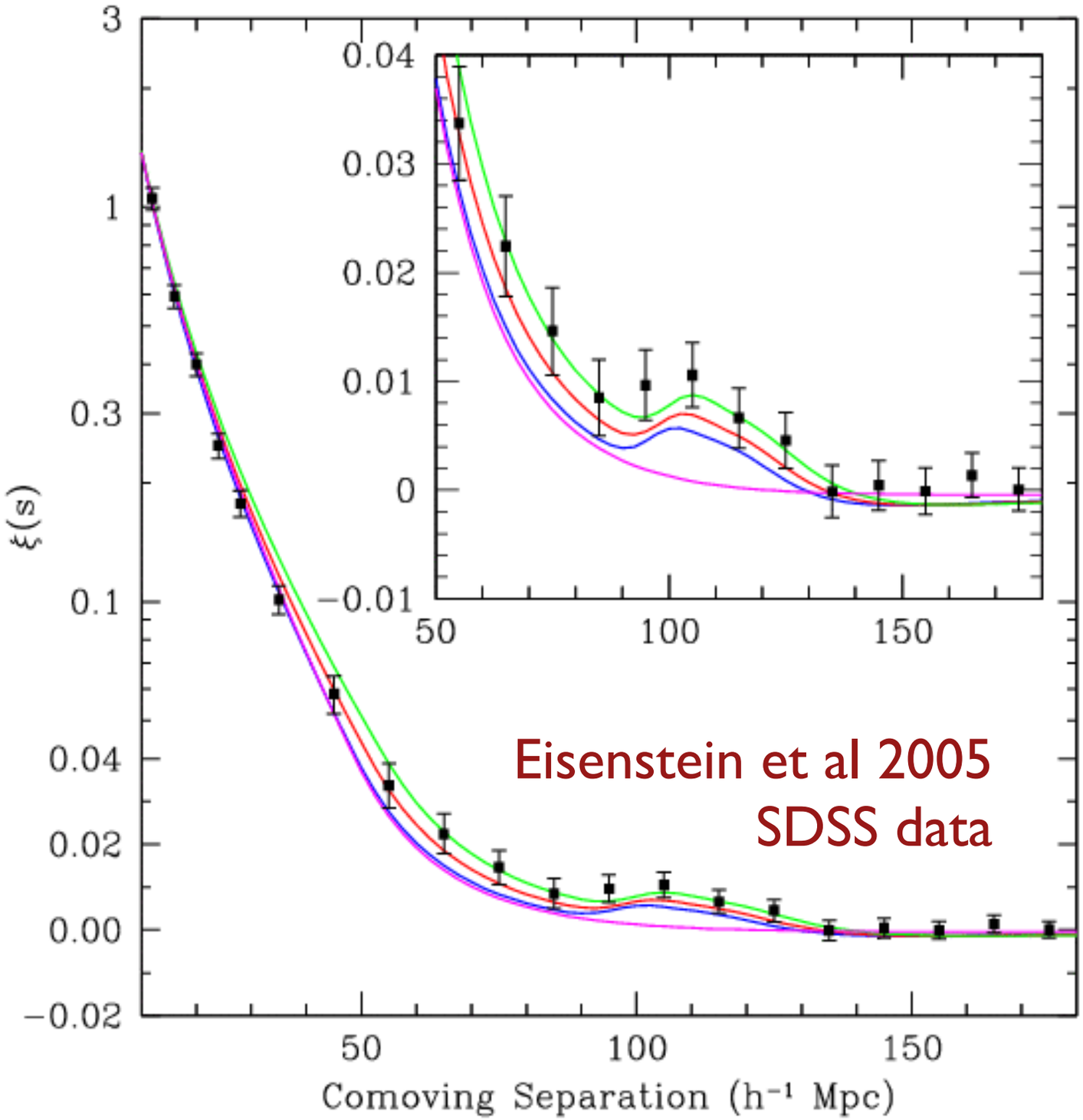
The measured 2-point correlation function (2PCF)

Large clustering strength at small separations

Almost zero on scales > 200Mpc/h

We can compare observation and theory/ simulations with the 2PCF

Recently some groups have used ML…

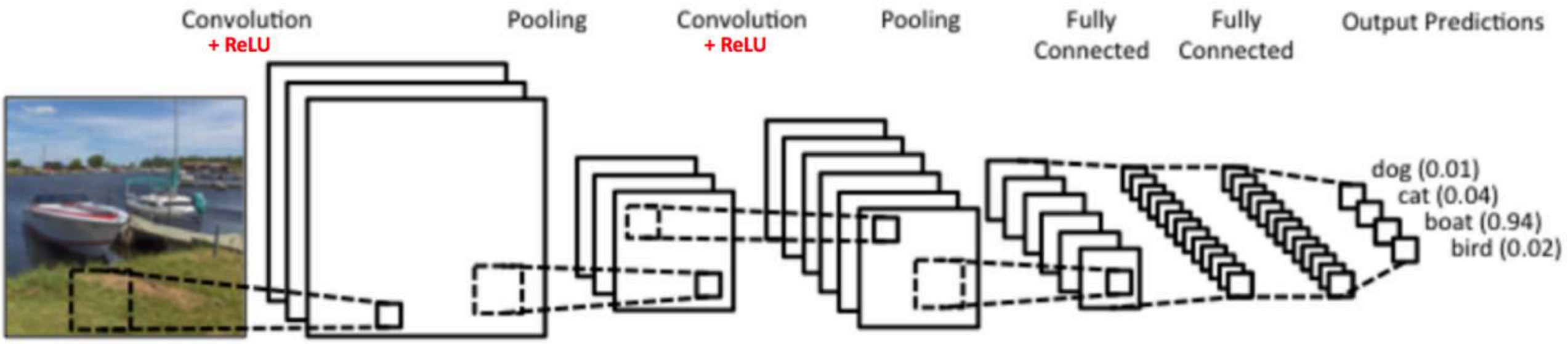


Convolutional Neural Networks

A convolutional neural network for visual classification

From a number of inputs obtain a meaningful output

By training CNN on a known set of data (labelled)



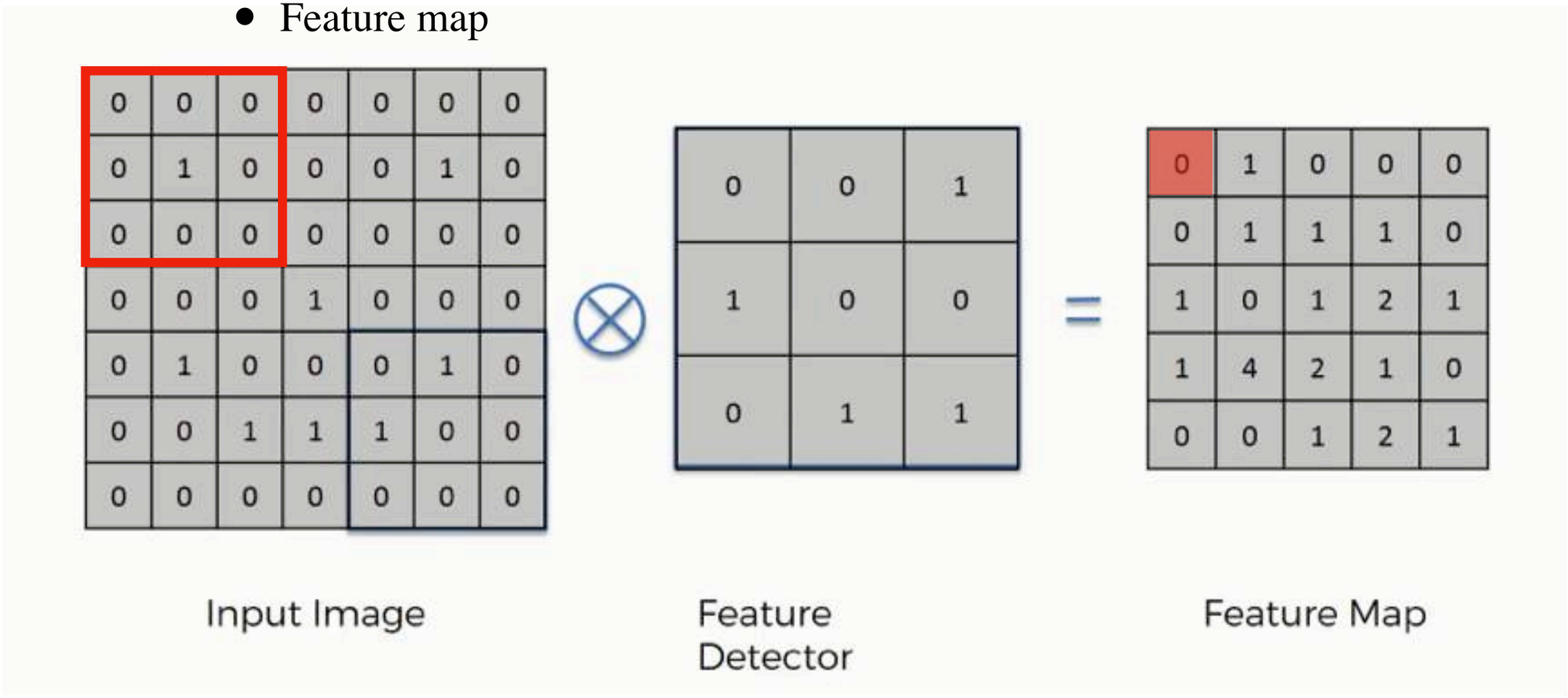
We will break down each step one by one…

Convolutional Neural Networks

Convolution Layer

Here are the three elements that enter into the convolution operation:

- Input image
- Feature detector
- Feature map

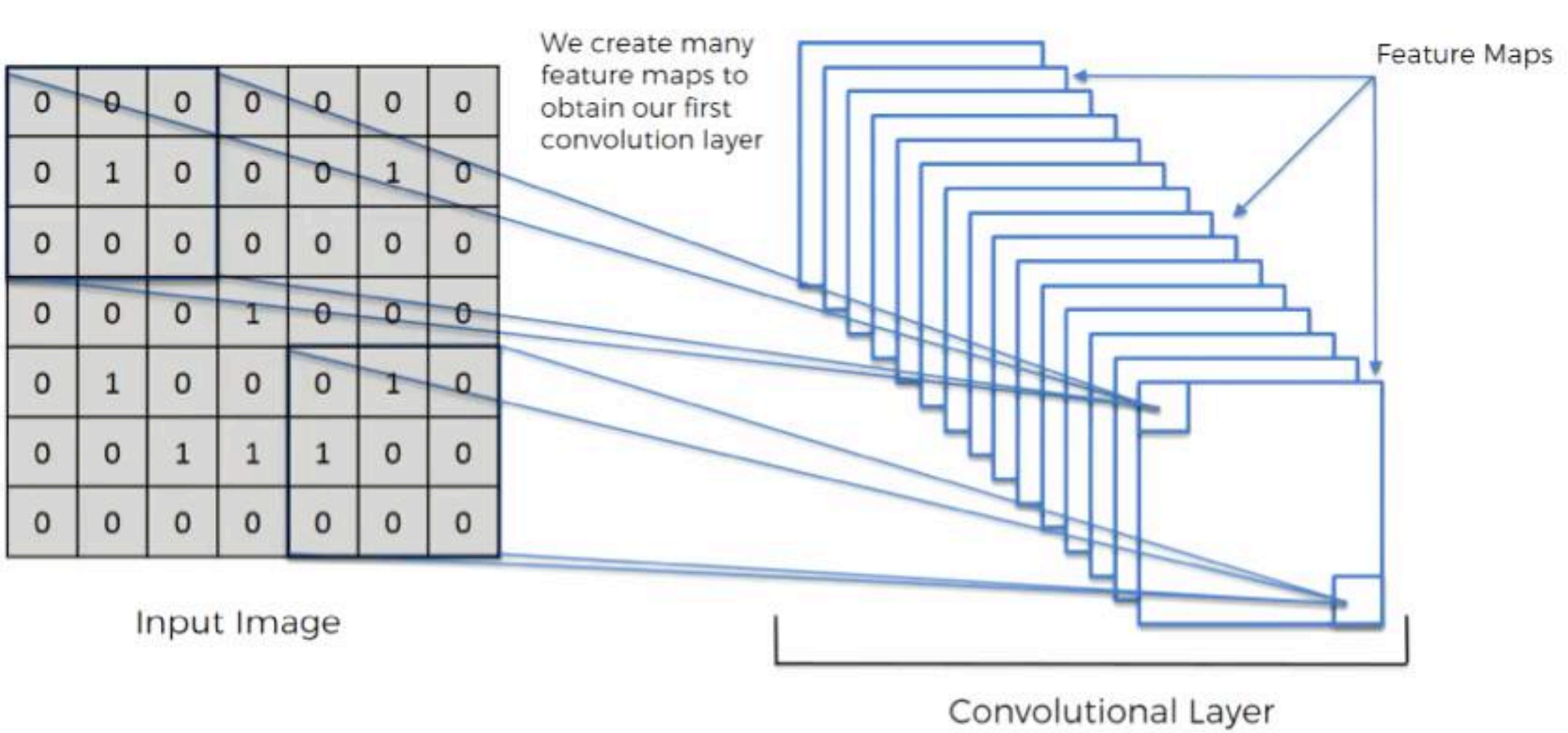


Convolutional Neural Networks

Convolution Layer

In reality, convolutional neural networks develop multiple feature detectors and use them to develop several feature maps which are referred to as convolutional layers

Through training, the network determines what features it finds important in order for it to be able to scan images and categorize them more accurately.



Convolutional Neural Networks

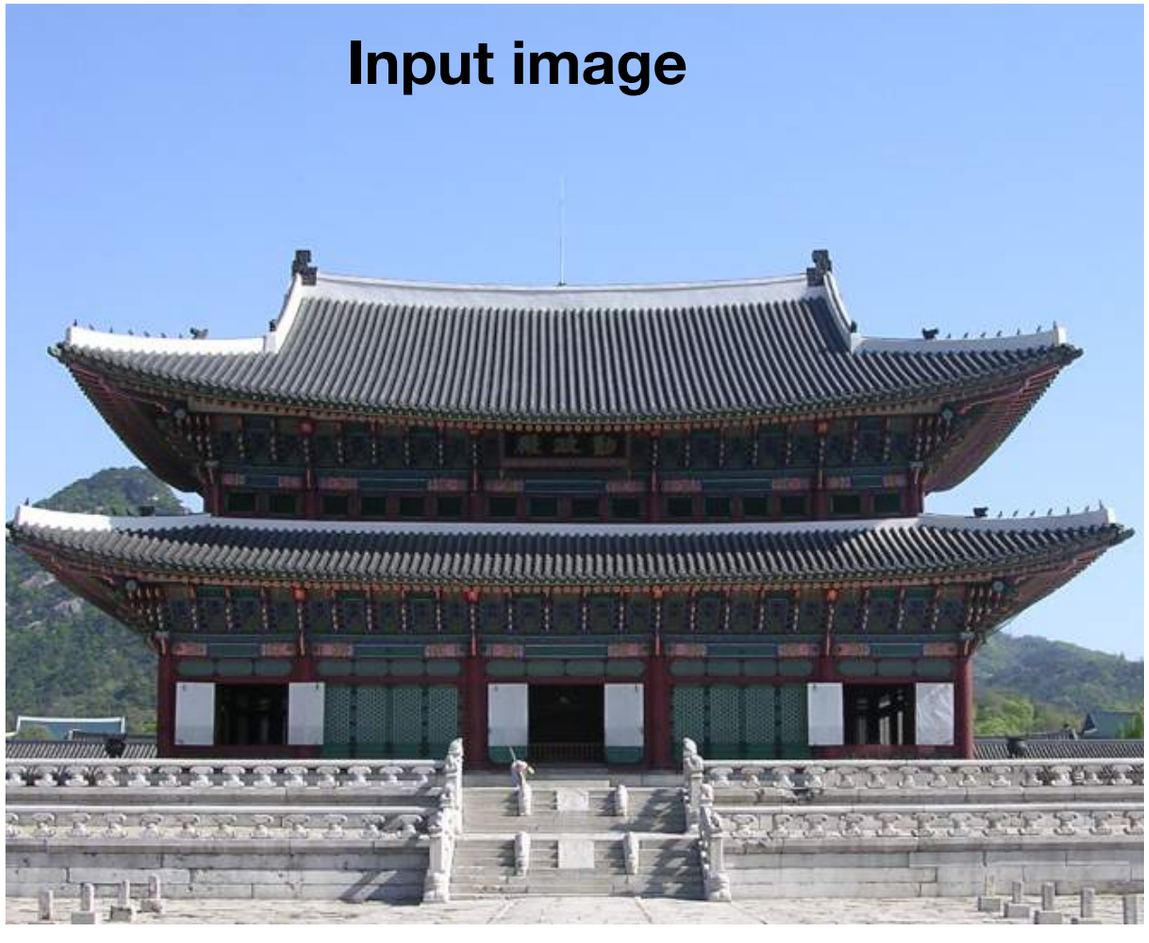
1. Convolution Layer

These filters have a use in image processing and that will help give us a more intuitive feeling about them

Sharpen

0	-1	0
-1	5	-1
0	-1	0

Input image



Feature Map



Convolutional Neural Networks

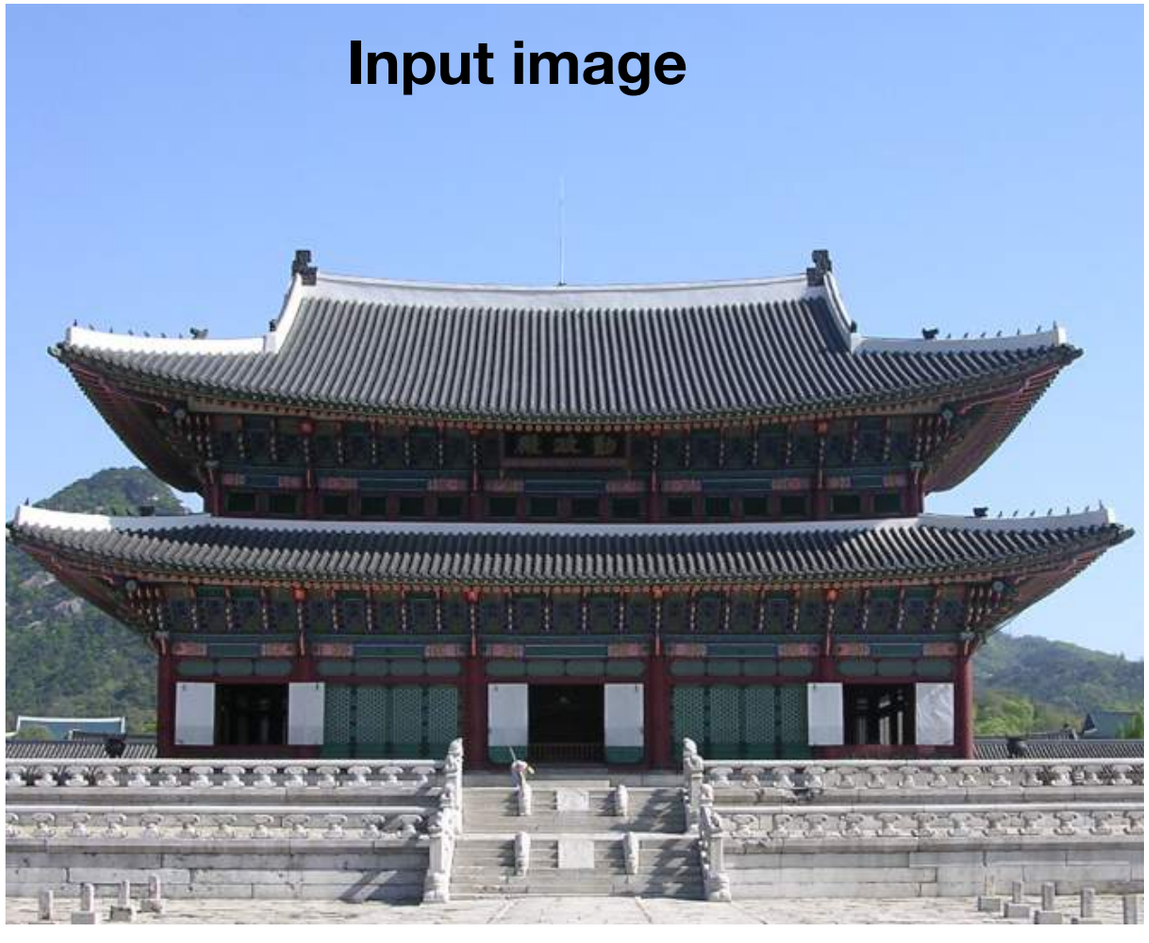
1. Convolution Layer

These filters have a use in image processing and that will help give us a more intuitive feeling about them

Blur

1	1	1
1	1	1
1	1	1

Input image



Feature Map



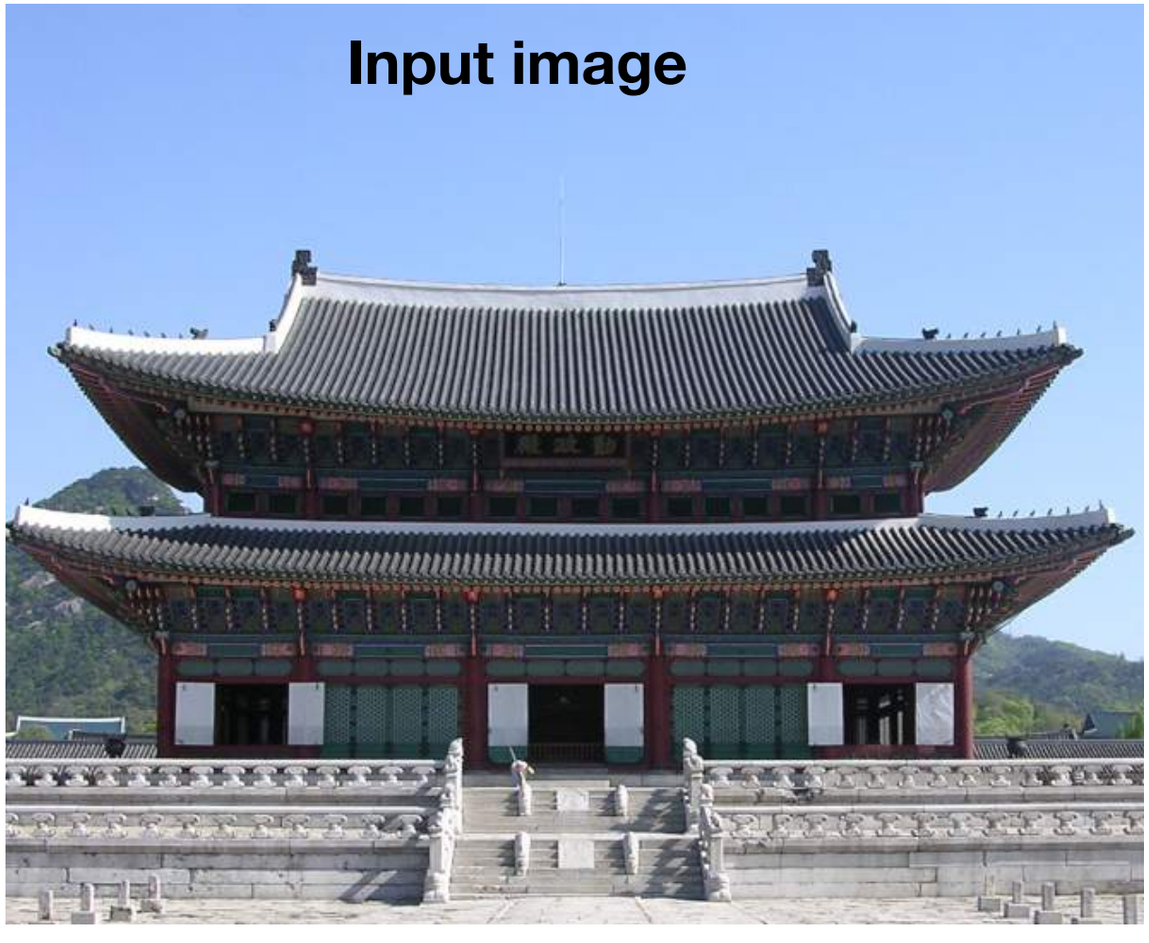
Convolutional Neural Networks

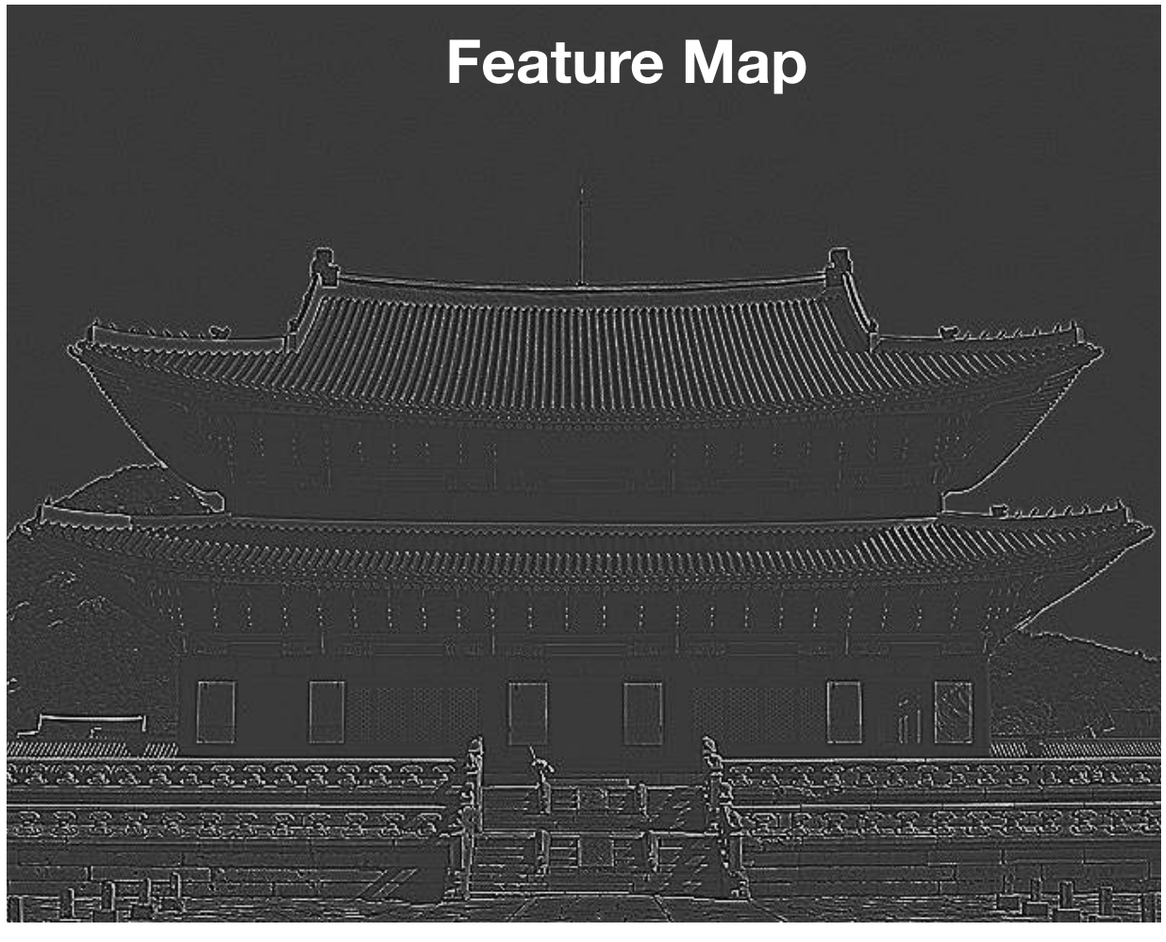
1. Convolution Layer

These filters have a use in image processing and that will help give us a more intuitive feeling about them

Edge Detect

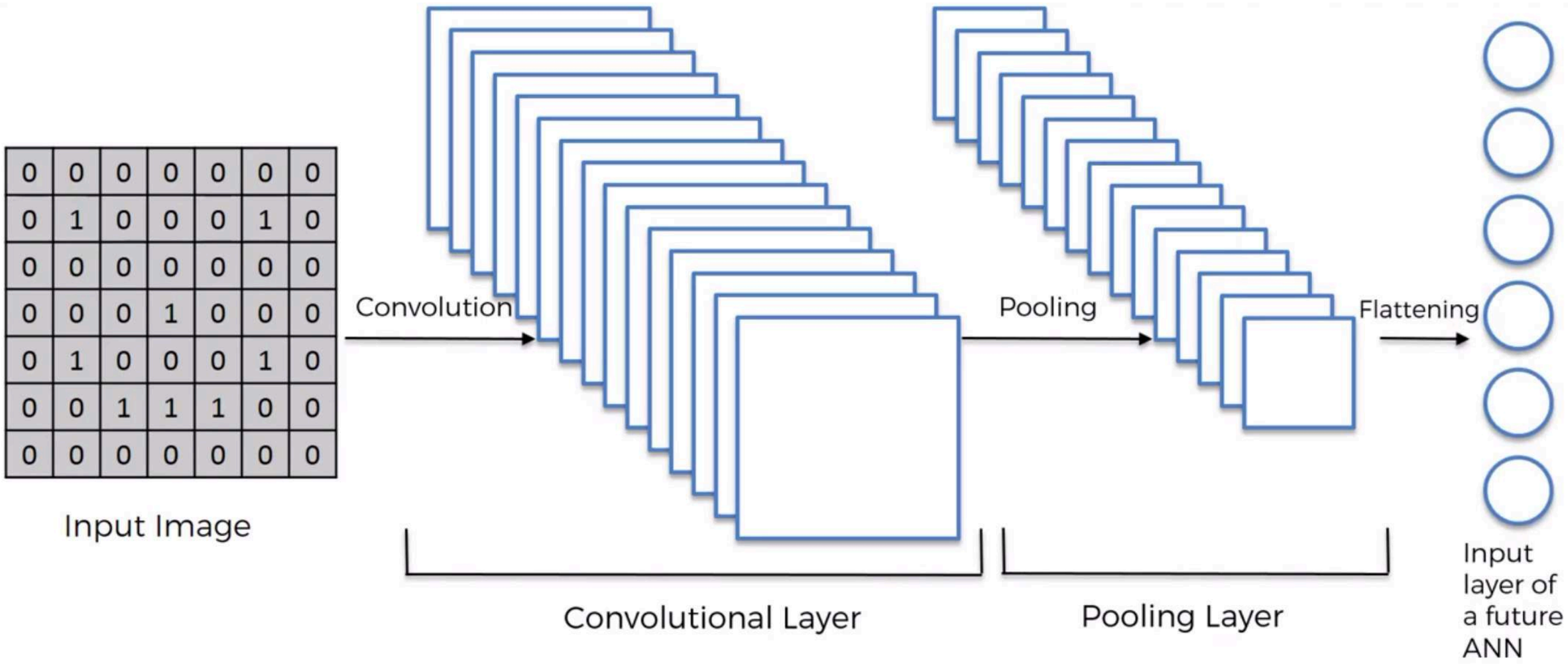
0	1	0
1	-4	1
0	1	0





Convolutional Neural Networks

Flattening

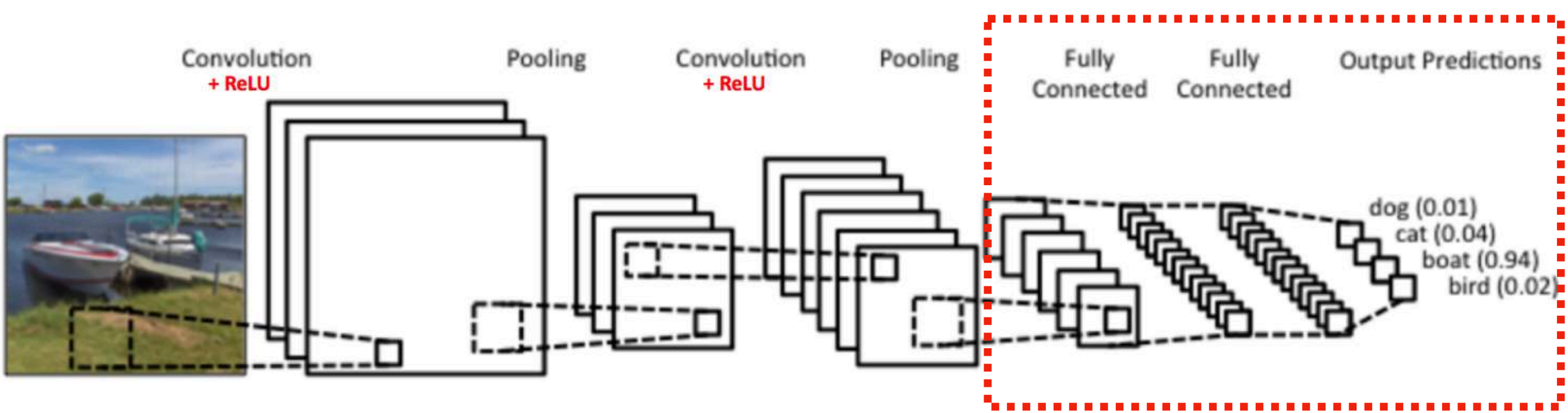


Convolutional Neural Networks

Fully connected layer(s)

The Fully Connected layer is a traditional Multi Layer Network that uses a softmax activation function in the output layer. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer.

The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

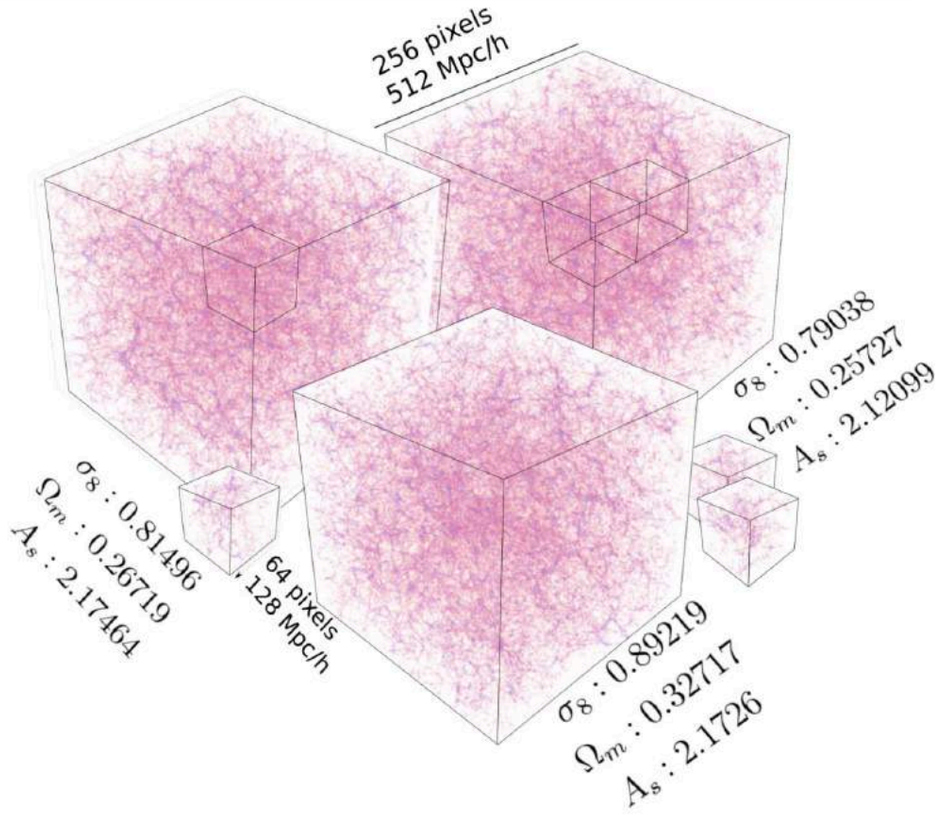


Deep Learning the Large Scale Structure

Ok lets apply this to cosmology!

Deep Learning the Large Scale Structure

Ravanbakhsh et al. (2017), Mathuriya et al. (2018) showed that convolutional neural networks can be trained to predict cosmological parameters from the visual shape of the large scale structure, i.e. the filaments, clusters and voids of the cosmic density field.

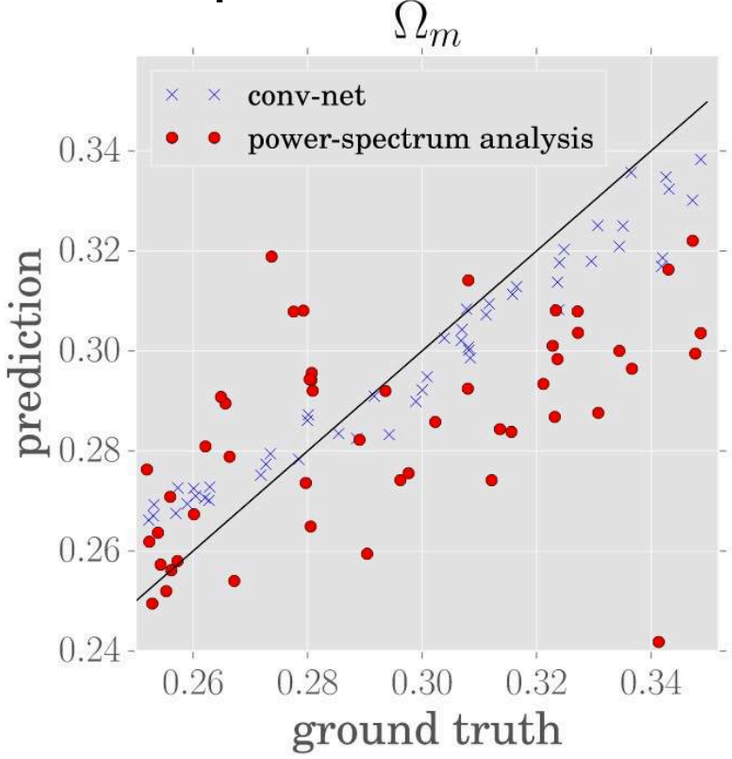


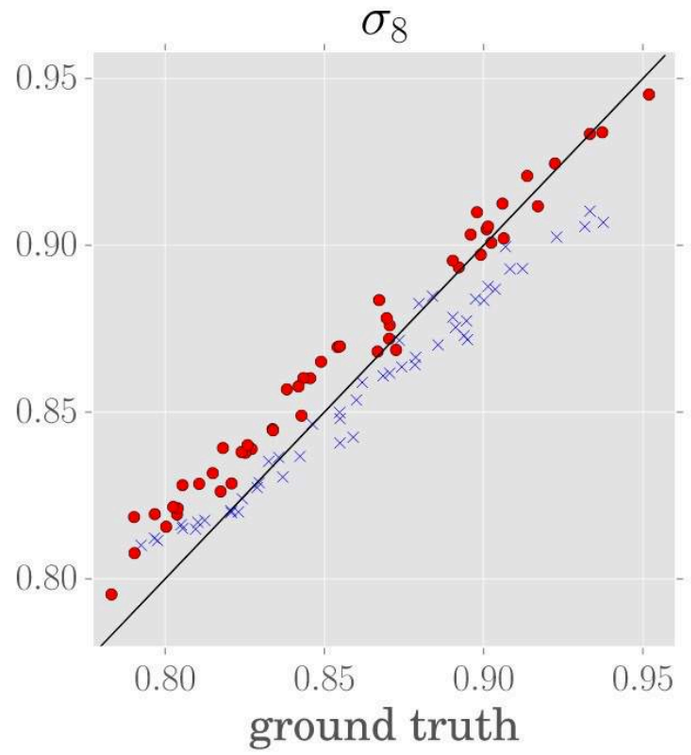
500 COmoving Lagrangian Acceleration(COLA) simulations

512Mpc box with 512^3 dark matter particles

Output at z=0

Limitations:
Cubic volume at z=0
Real space
Dark Matter particles
Fixed Architecture





Deep Learning the Large Scale Structure

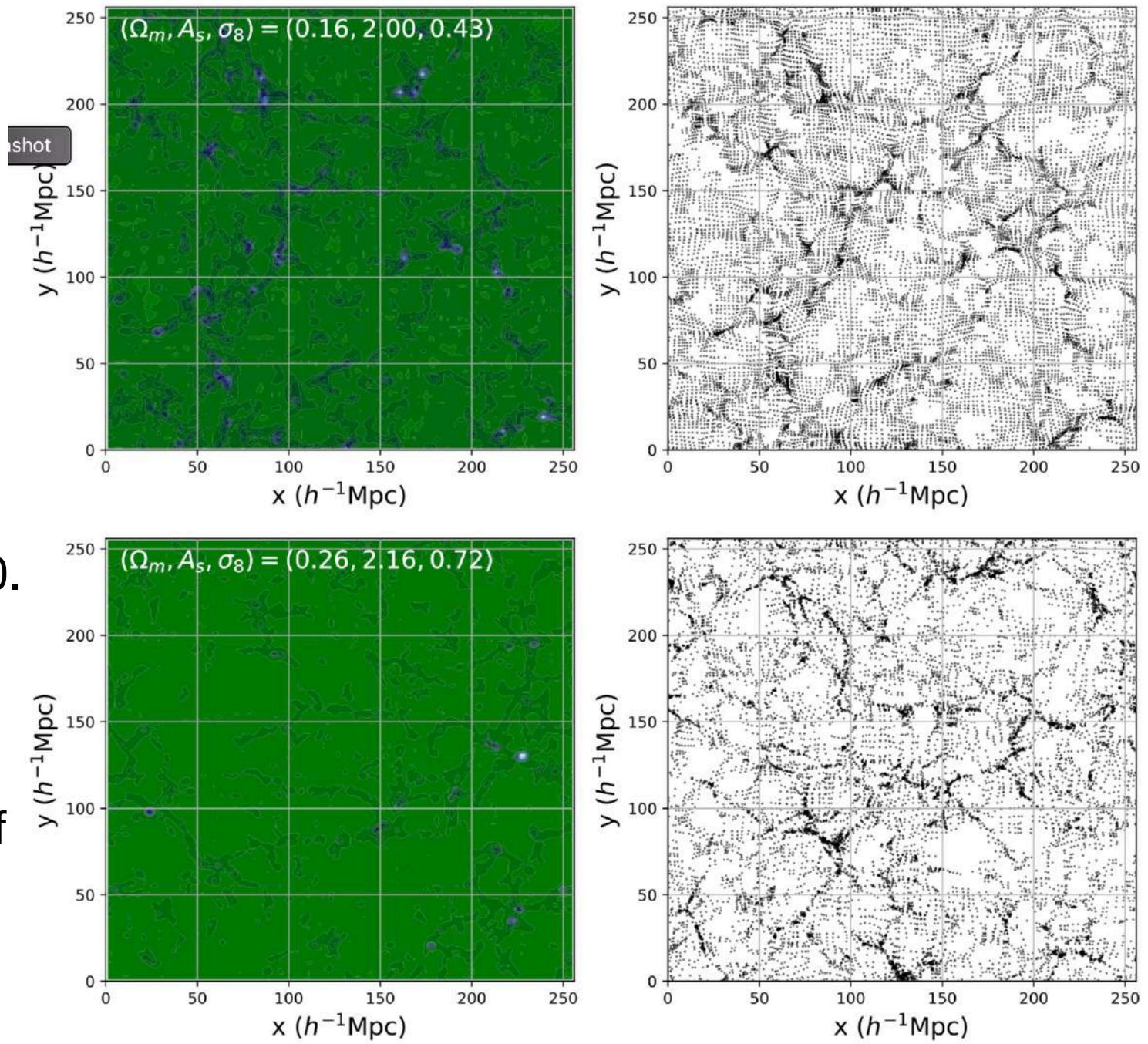
In a grid of 31x15 parameter combinations

$$0.16 < \Omega_M < 0.46$$
$$0.4 < \sigma_8 < 1.1$$

We run COLA DM simulations with with 128^3 particles, in a 256 Mpc box, using timesteps 40 output at z=0.

We grid the data onto 2Mpc voxels.

The input of the whole network is a 32^3-voxel (i.e. (64Mpc)3) subcube of the density field.

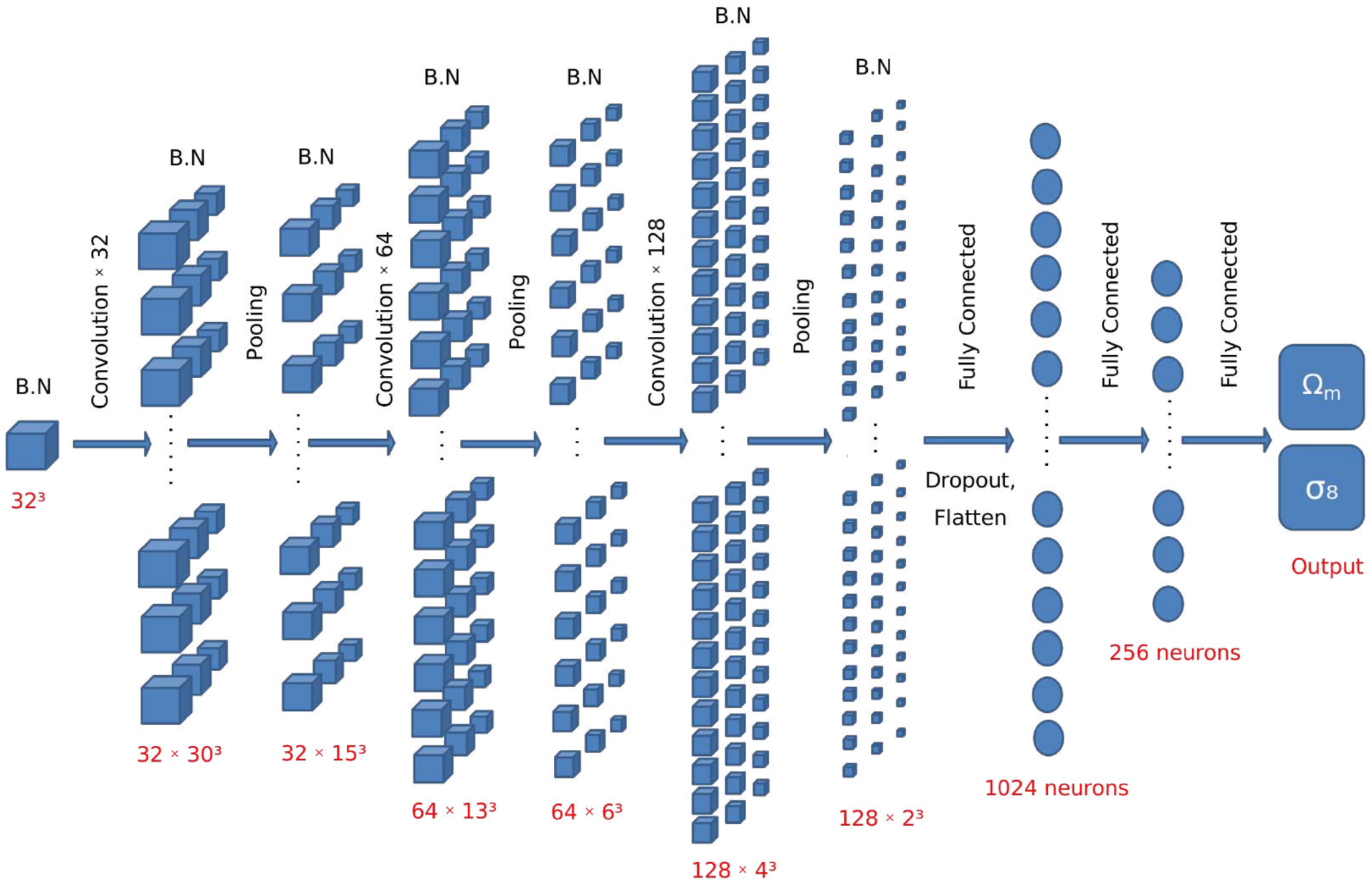


arXiv:1908.10590

Pan, Liu, Forero-Romero, Sabiu, Li, Miao, (led by) Xiao-Dong Li

Deep Learning the Large Scale Structure

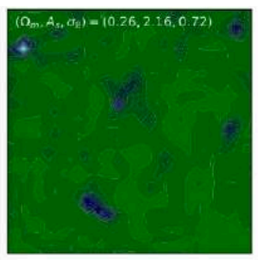
Default Architecture

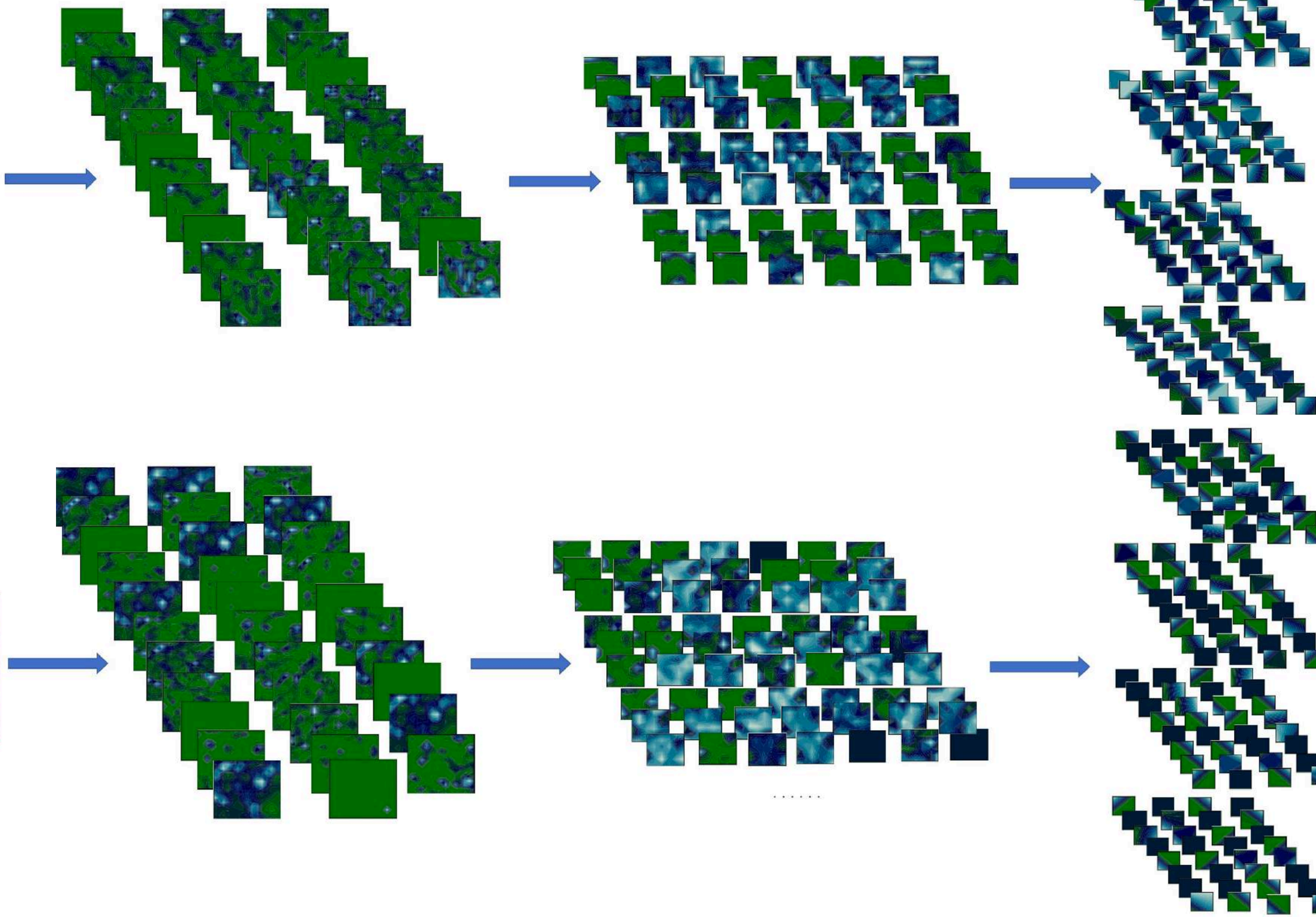


Deep Learning the Large Scale Structure

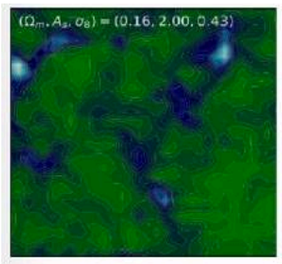
Inspecting the CNN

Om = 0.26





Om = 0.16



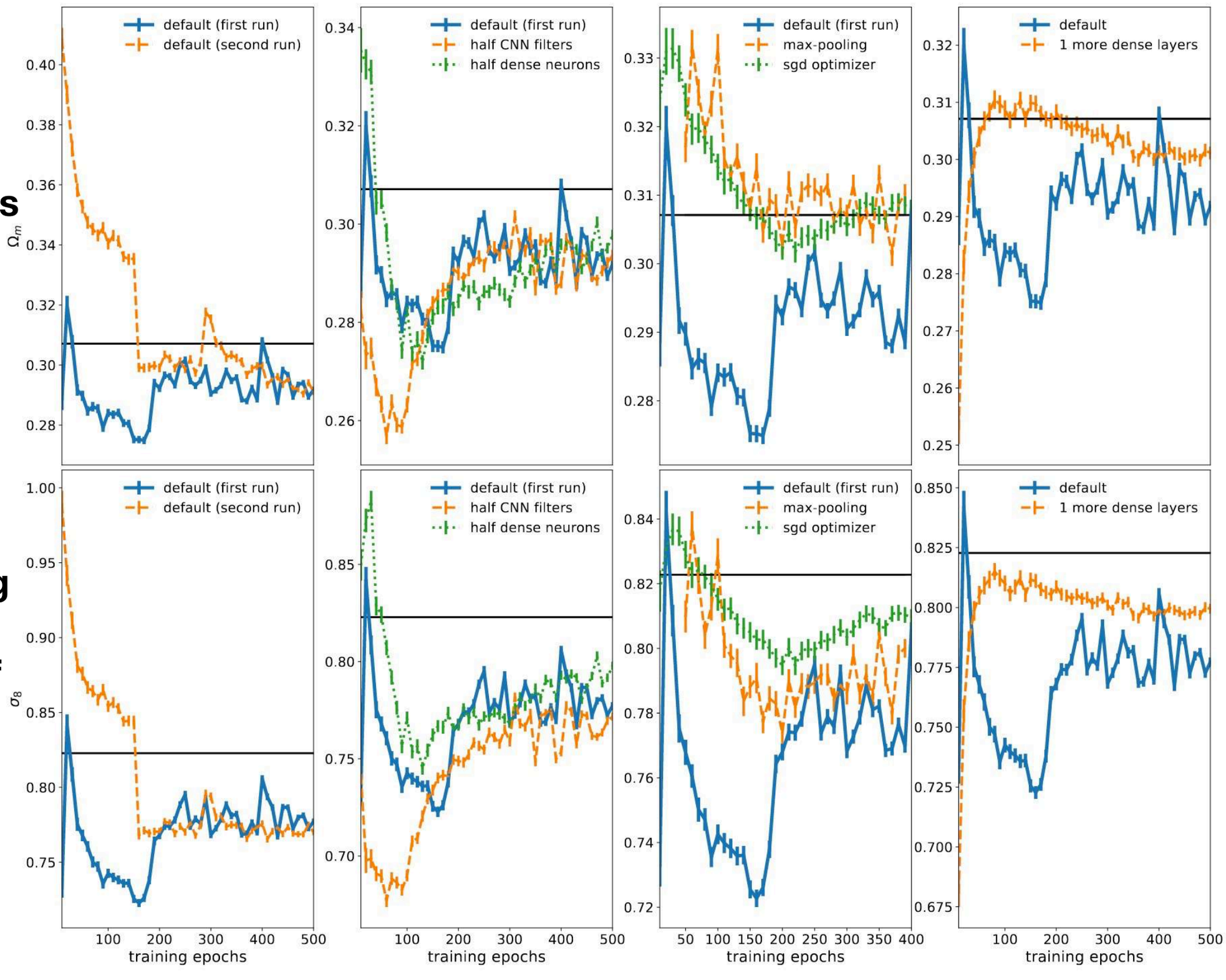
Deep Learning the Large Scale Structure

Learning Curves

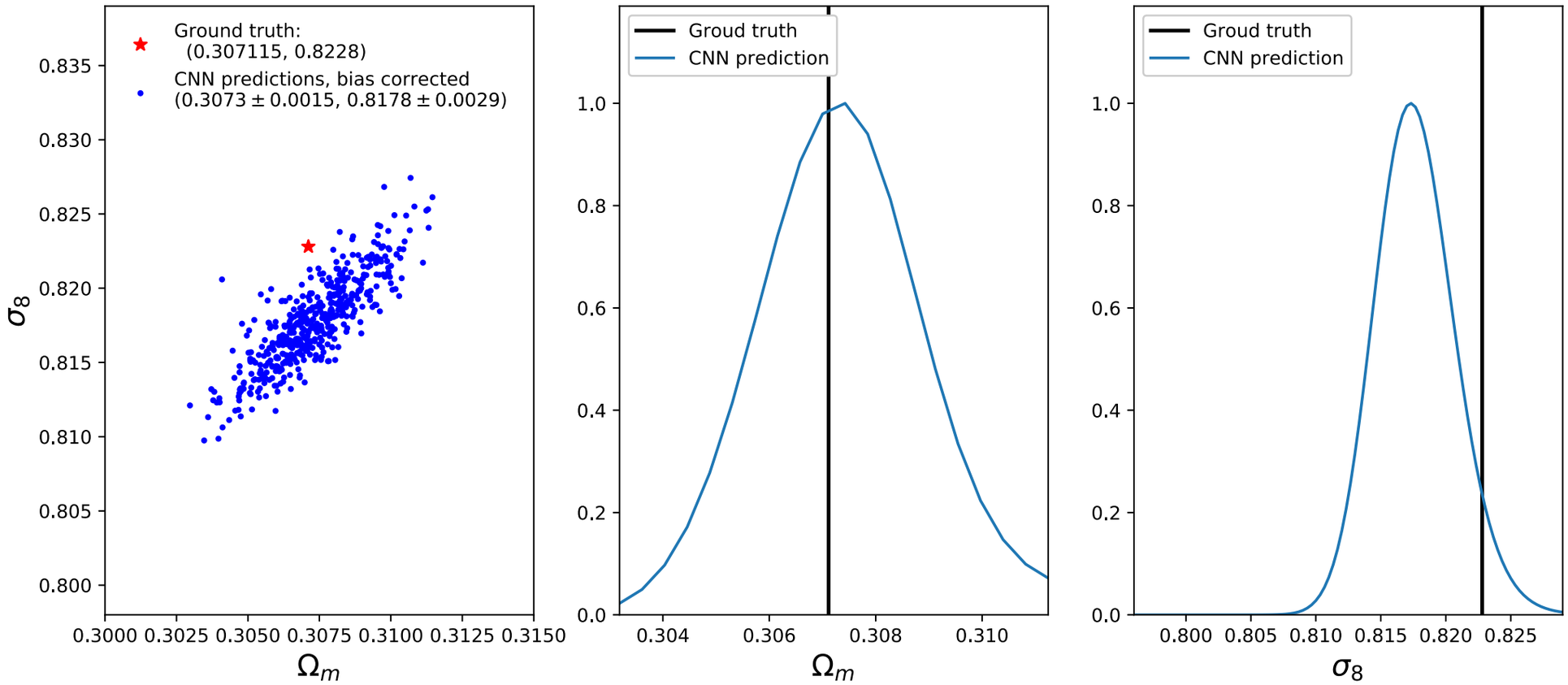
Varying:
- **# of CNN filters**
- **# of dense neurons**
- **# of neuron layers**
- **Optimiser**
- **Pooling type**

Clear advantage in max-pooling over the average pooling

Clear advantage of the sgd optimiser over the default ‘Adam’ optimiser.

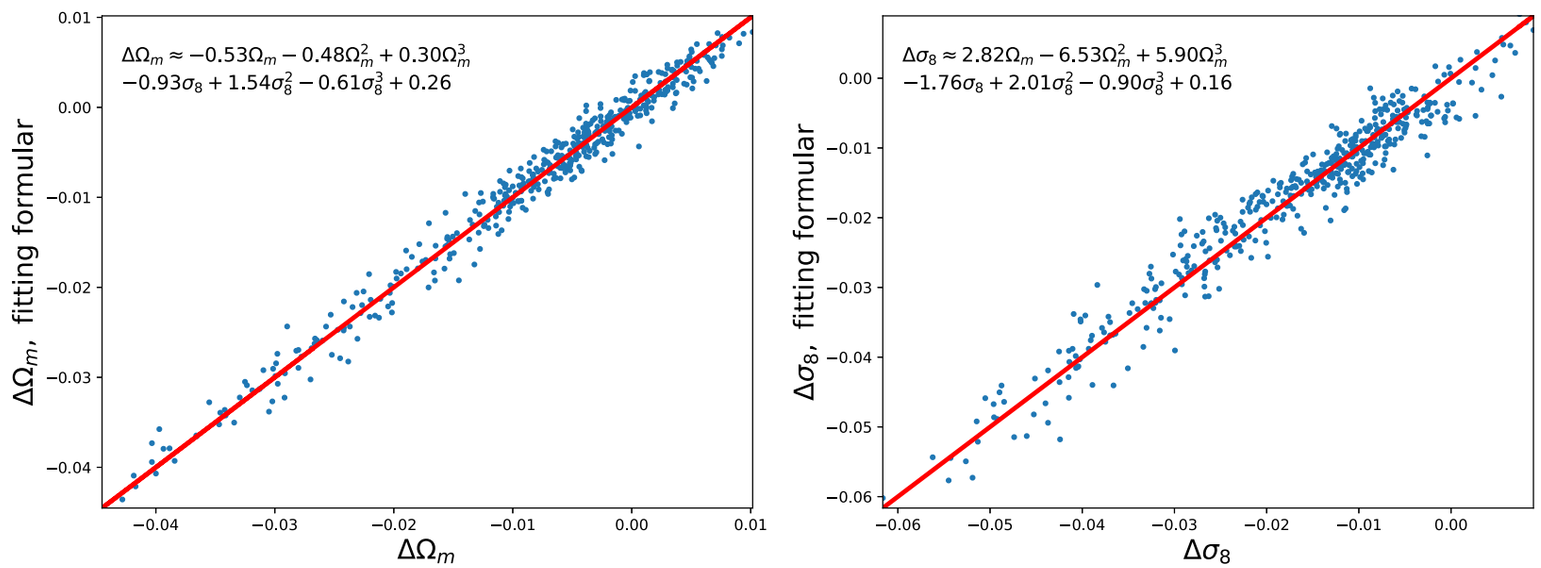


Deep Learning the Large Scale Structure



We found a significant bias in sigma8, this was found in previous works although not really highlighted

It can be easily corrected…



Pan et al. arXiv:1908.10590

Deep Learning the Large Scale Structure

In collaboration with the Korean Astronomy Machine Learning Group:
David Parkinson, Sungwook E. Hong, Srivatsan Sridhar, Shi Feng, Sangnam Park, Inku Park, Dongsu Bak, Jacobo Asorey Barreiro, Benjamin L'Hullier, James Jee, Arman Shafieloo, Hanwool Koo, Ryan Keeley +++

- ★ **Initiated a project in Korea to further develop this approach**
- ★ **By considering redshift evolution of the density field we aim to constrain expansion dynamics**
- ★ **We will consider more realistic tracers such as mock galaxy or weak lensing convergence fields**
- ★ **We will add systematics such as redshift error, photo-z etc, angular completeness, etc**
- ★ **How to mitigate nonlinear effects and other small scale physics that are difficult to model - clipping, field transforms, etc**

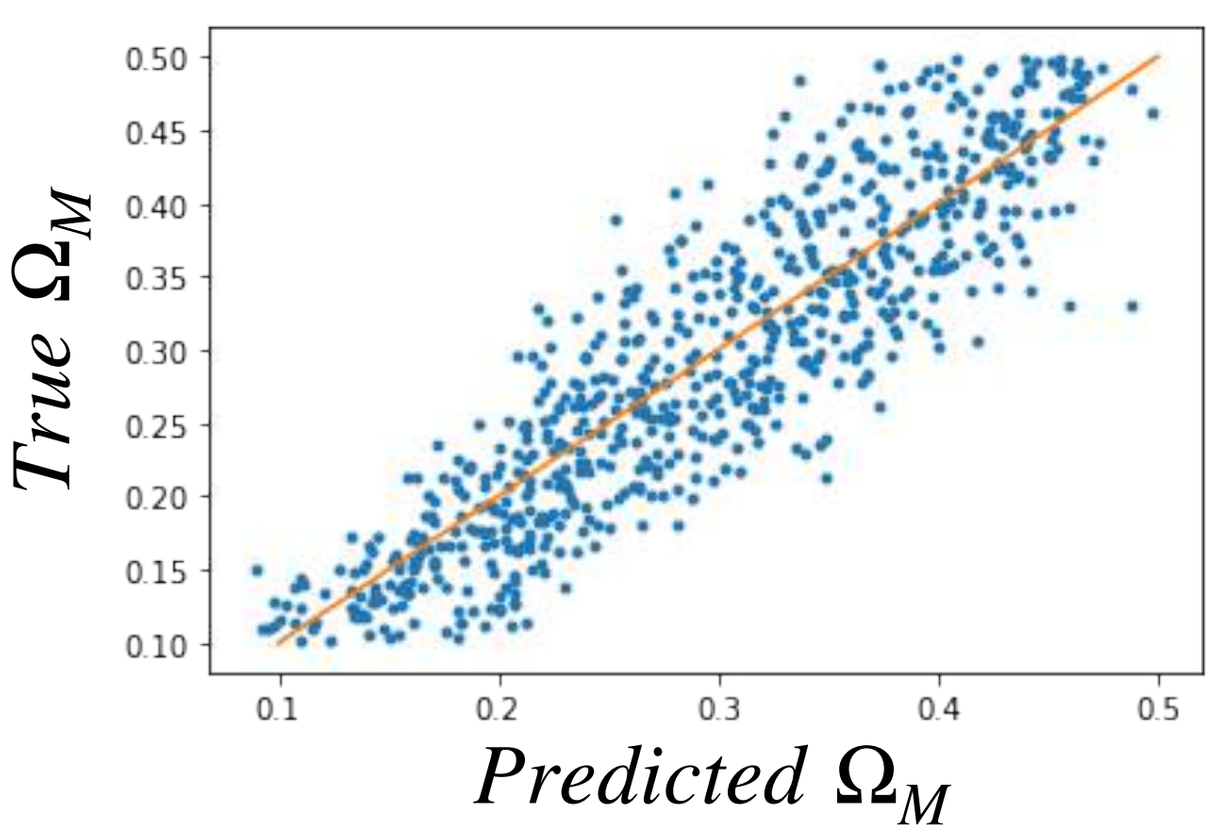
Information Content from CNN

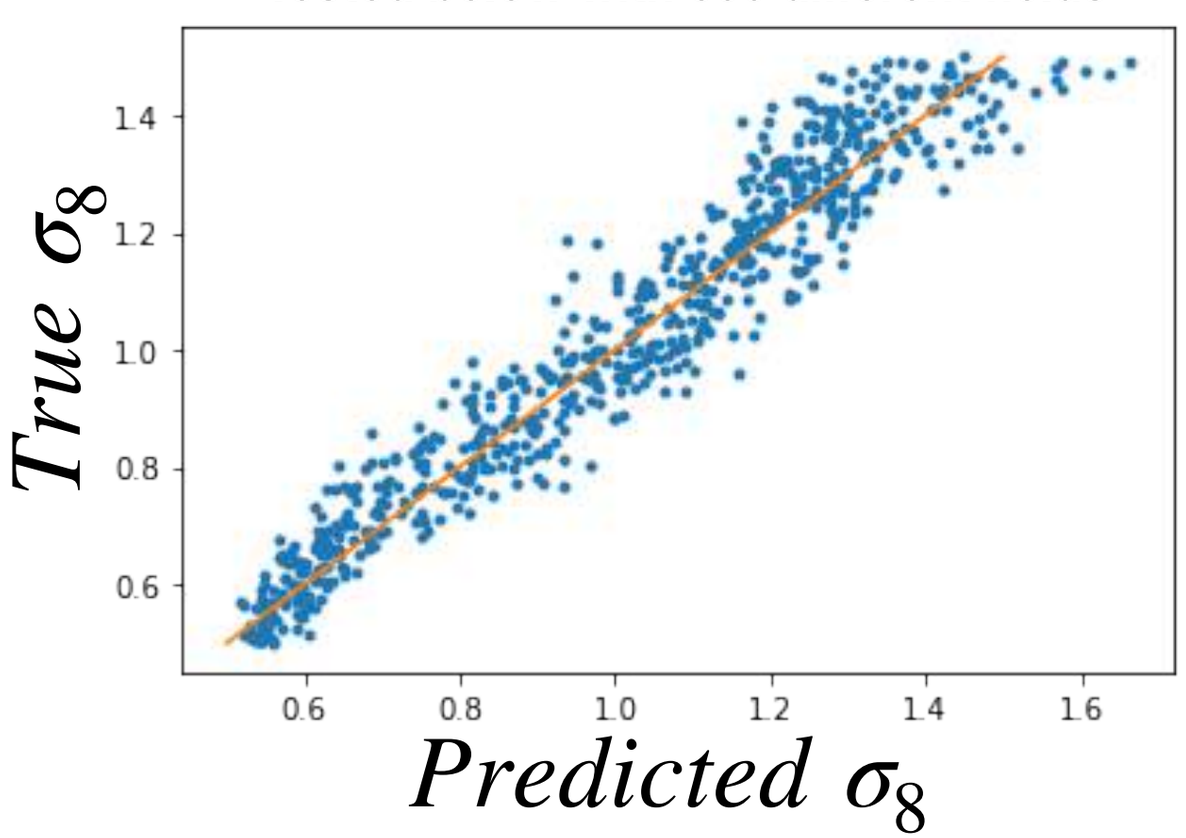
With Stephen Appleby

But first lets step back and look at Gaussian fields…
We know that their information is confined to ≤2nd order statistics
We can compute the errors on parameters from
1) monte carlo realisations
2) Fisher matrix

CNN Trained with 6,000 gaussian fields with linear cosmological power spectra

Tested below with 600 different fields





Information Content from CNN

Stepping back lets looks at Gaussian fields.
We know that their information is confined to ≤2nd order statistics
We can compute the errors on parameters from
1) monte carlo realisations
2) Fisher matrix
3) Via the CNN directly?

We can write a custom loss function,

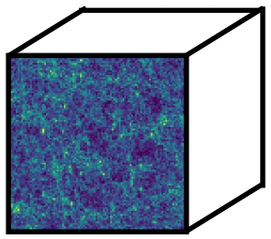
$$L = \frac{1}{2}\left(\ln\left(|\Sigma|\right) + \left(\theta_p - \theta_t\right)^T \Sigma^{-1} \left(\theta_p - \theta_t\right)\right),$$

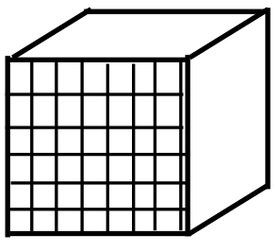
$$L = \begin{bmatrix} a & 0 \\ b & c \end{bmatrix} \qquad \Sigma^{-1} = LL^T,$$

Where the parameters a,b,c are learned by the network during training.
The network can learn the covariance matrix directly without an input ‘label’

Information Content from CNN

Cosmic Gaussian field

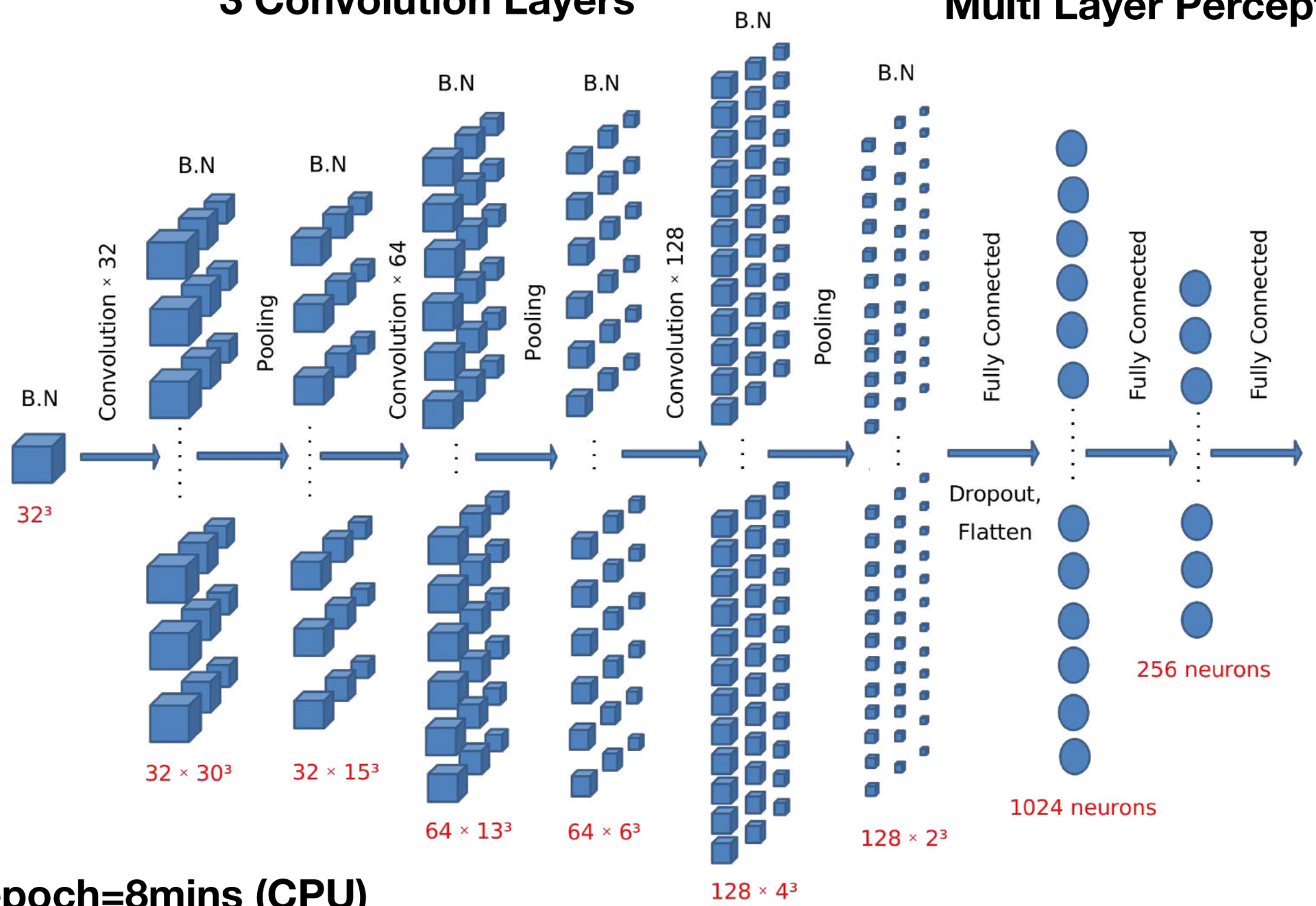




Neural Network Architecture

3 Convolution Layers

Multi Layer Perceptron



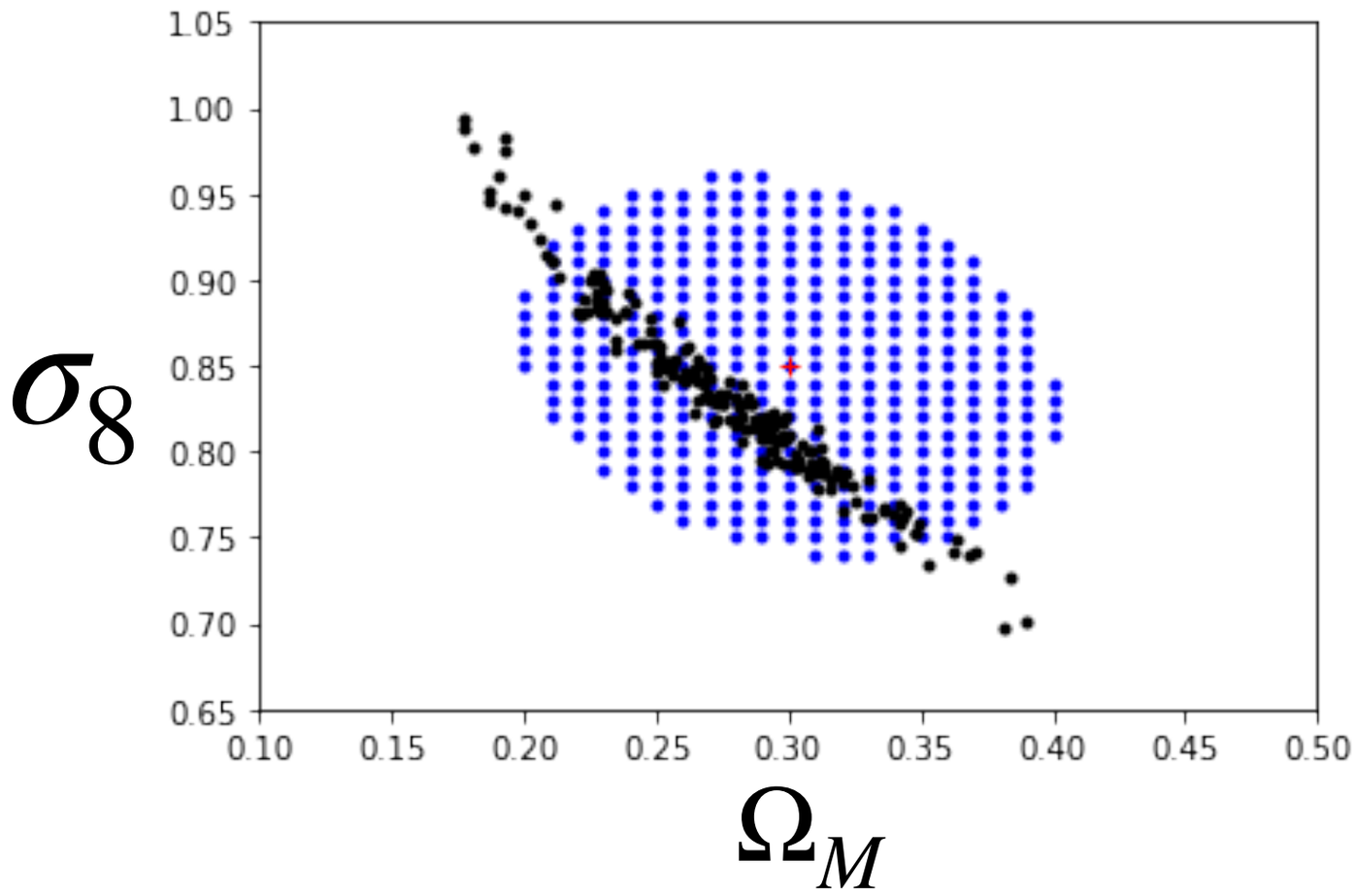
1 training epoch=8mins (CPU)

1 training epoch=7s (GPU) !!!! Thanks google!

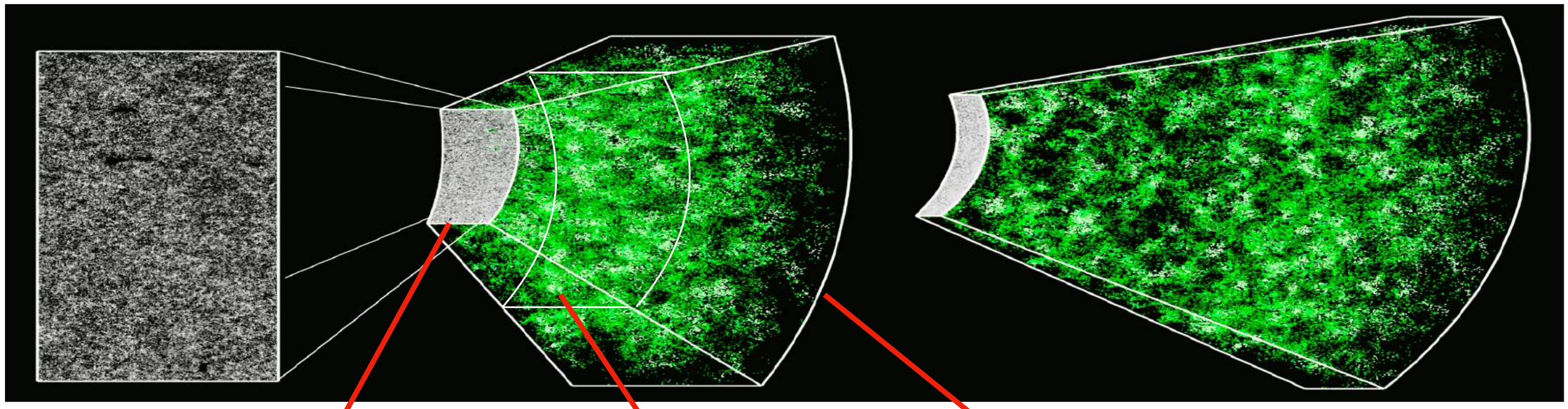
Information Content from CNN

Stepping back lets looks at Gaussian fields.
We know that their information is confined to ≤2nd order statistics
We can compute the errors on parameters from

1) **monte carlo realisations**
2) **Fisher matrix**
3) **Via the CNN directly?**

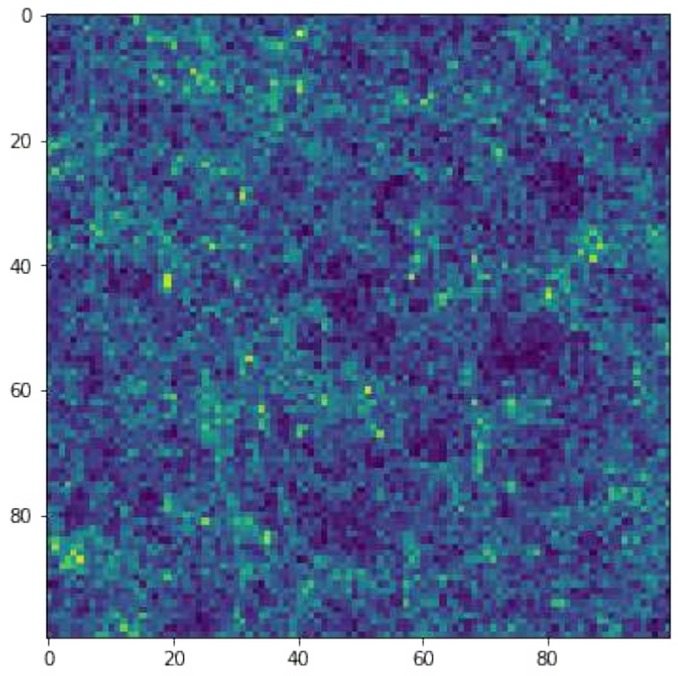


Deep Learning the Expansion History

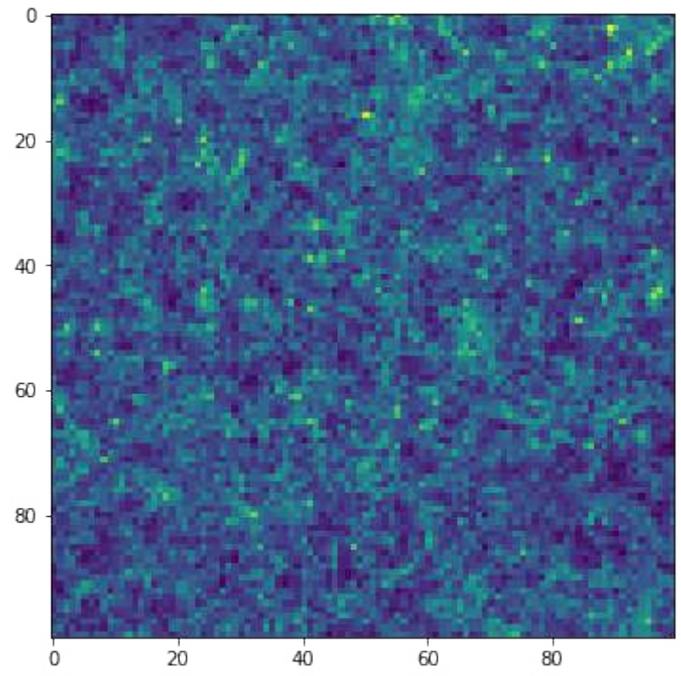


100x100sq deg

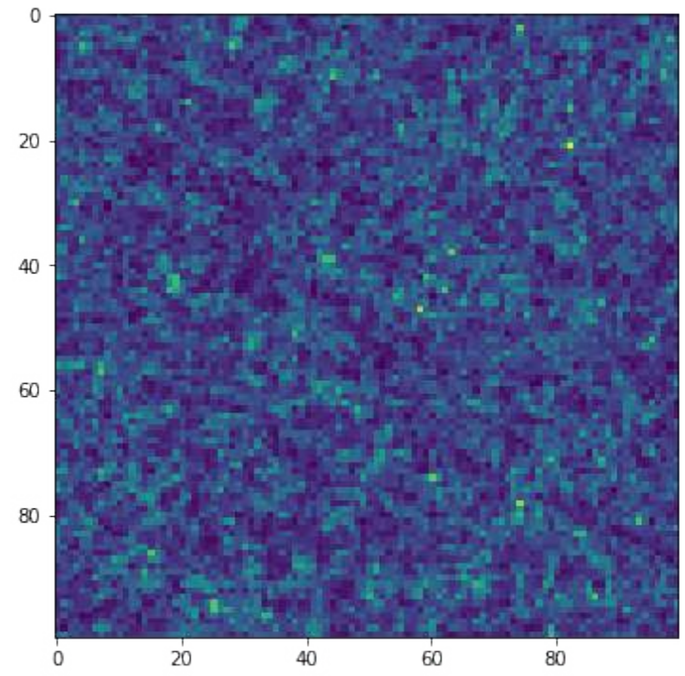
z=0



z=1



z=2



Training Data

In Machine learning training data will be our ‘model’
We use fast approximate N-body simulations as training data

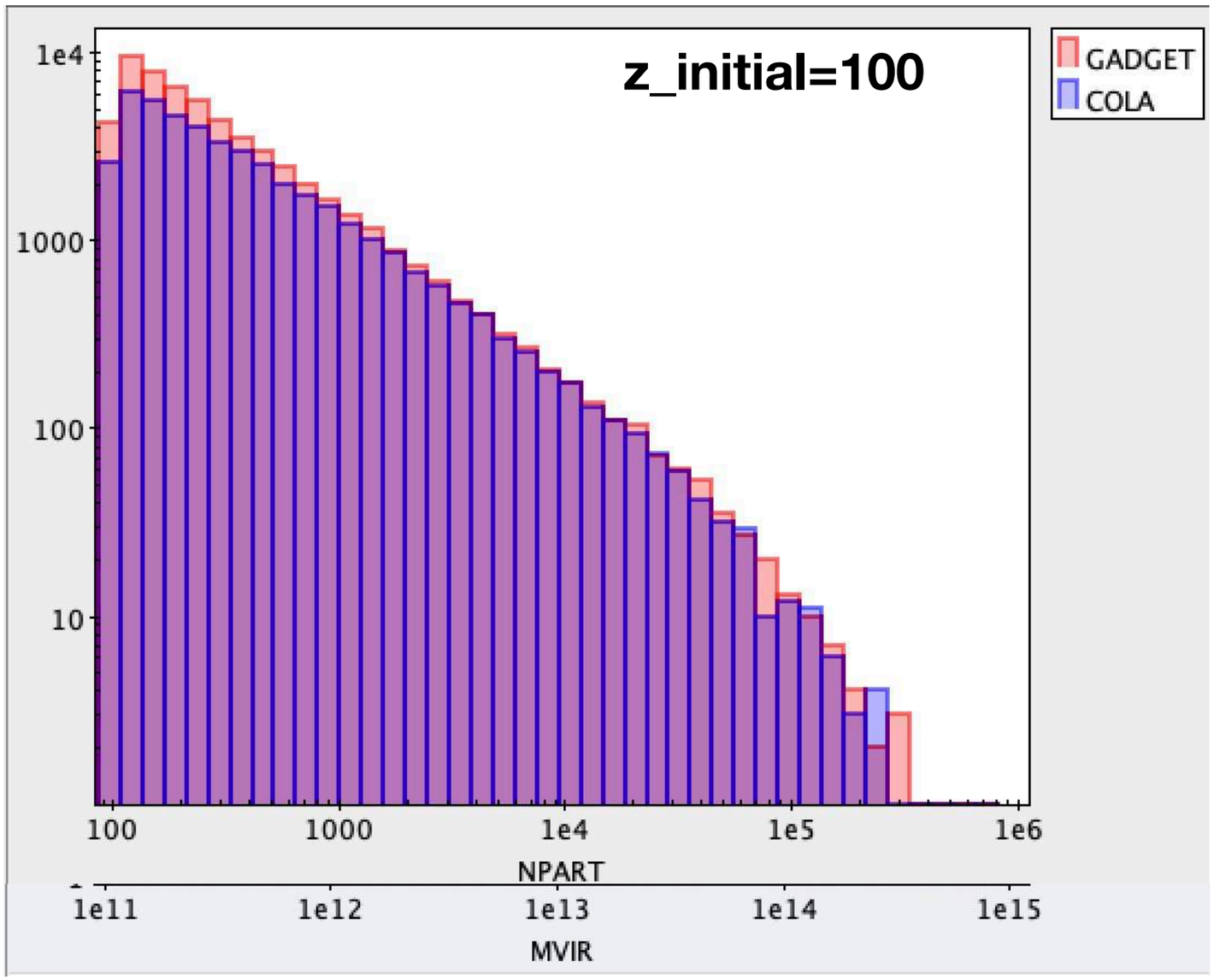
Aim: ‘Patch’ COLA to reproduce small scales
Satellites their positions and abundances etc

512^3 particle simulation
120Mpc size box

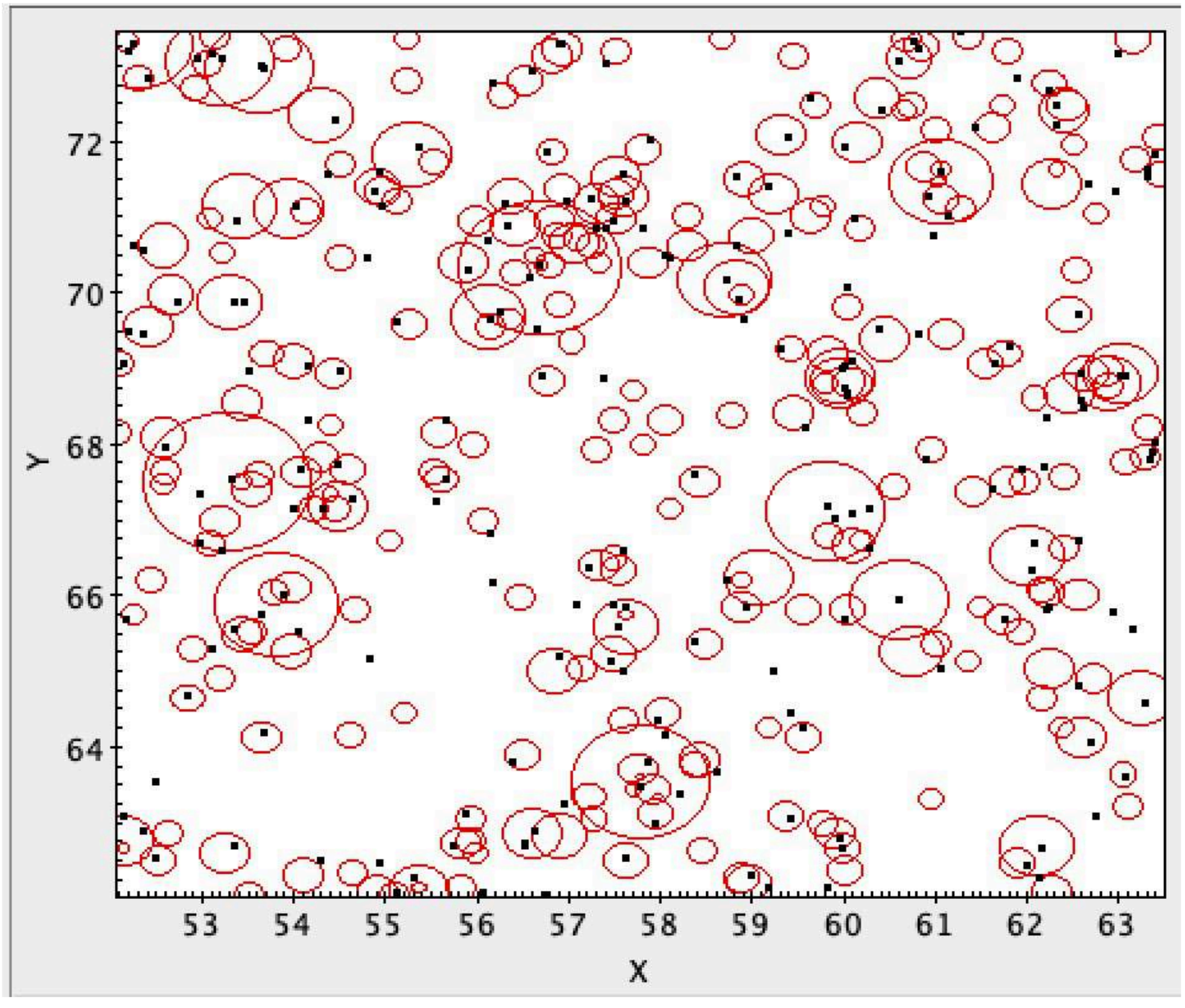
Gadget vs COLA

1+ days vs seconds/minutes

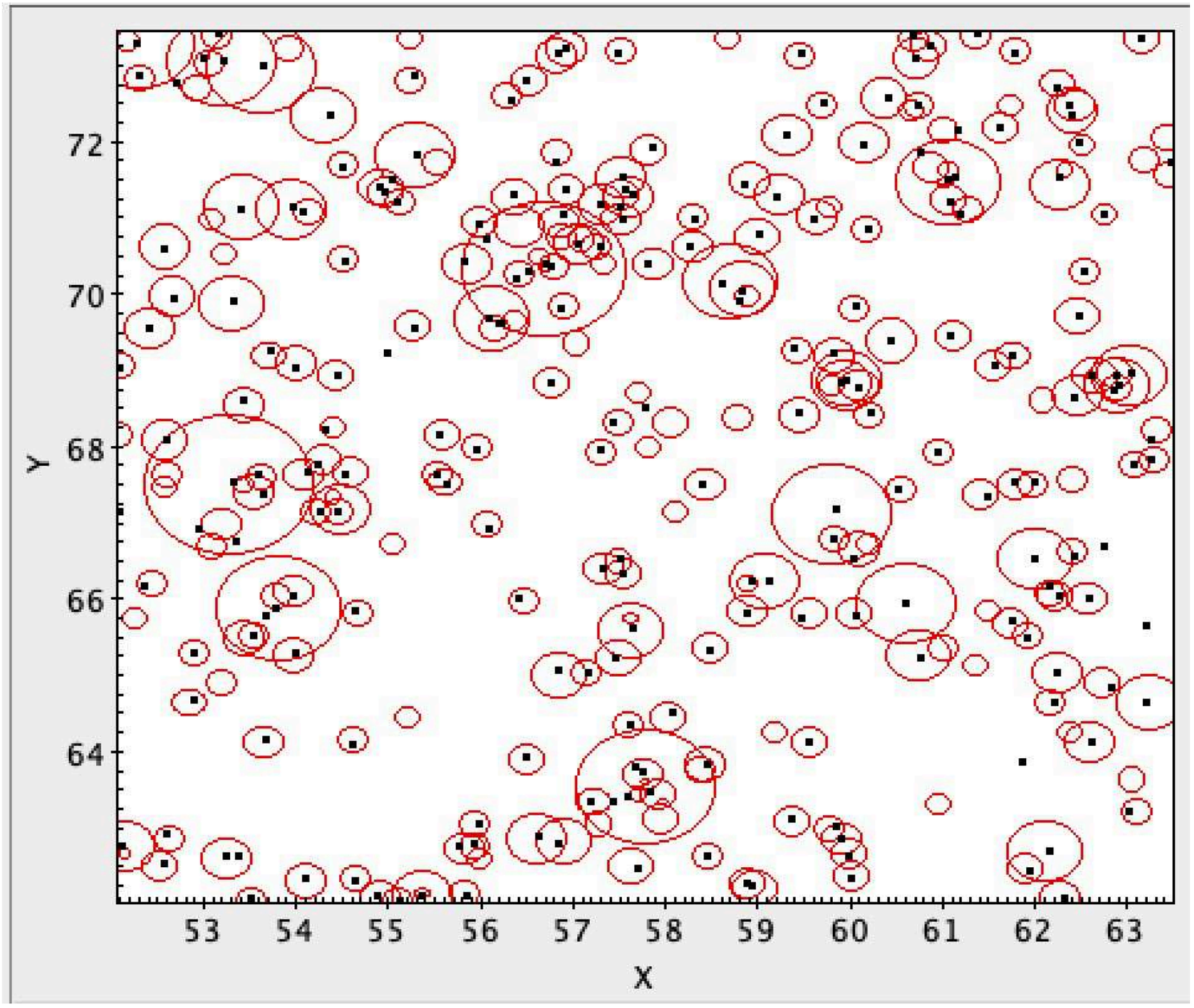
Thanks to Jihye Shin for the Gadget comparison!



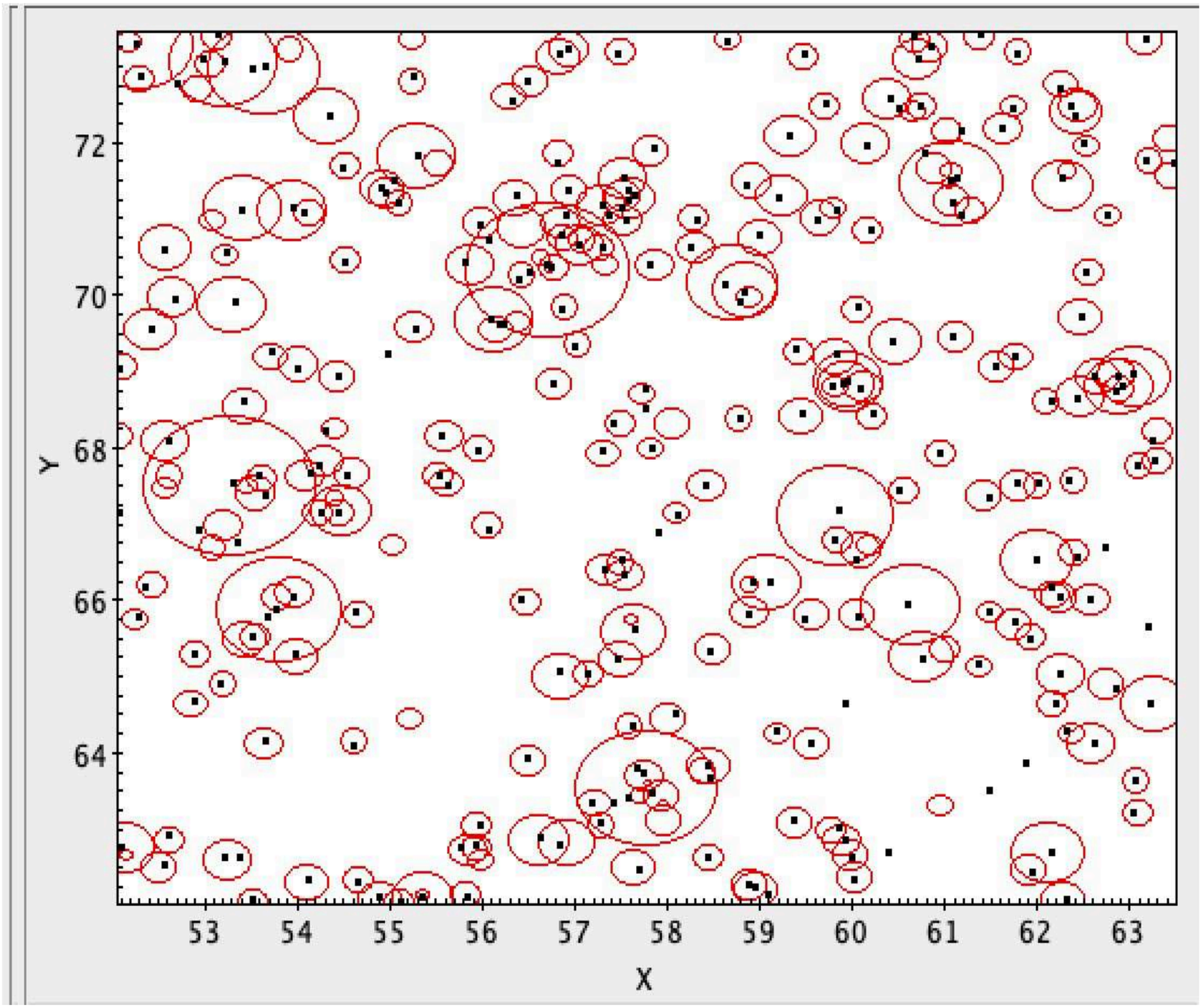
Gadget vs COLA mesh=1024 z_initial=100



Gadget vs COLA mesh=1536 z_initial=19



Gadget vs COLA mesh=1536 + abundance match z_initial=19



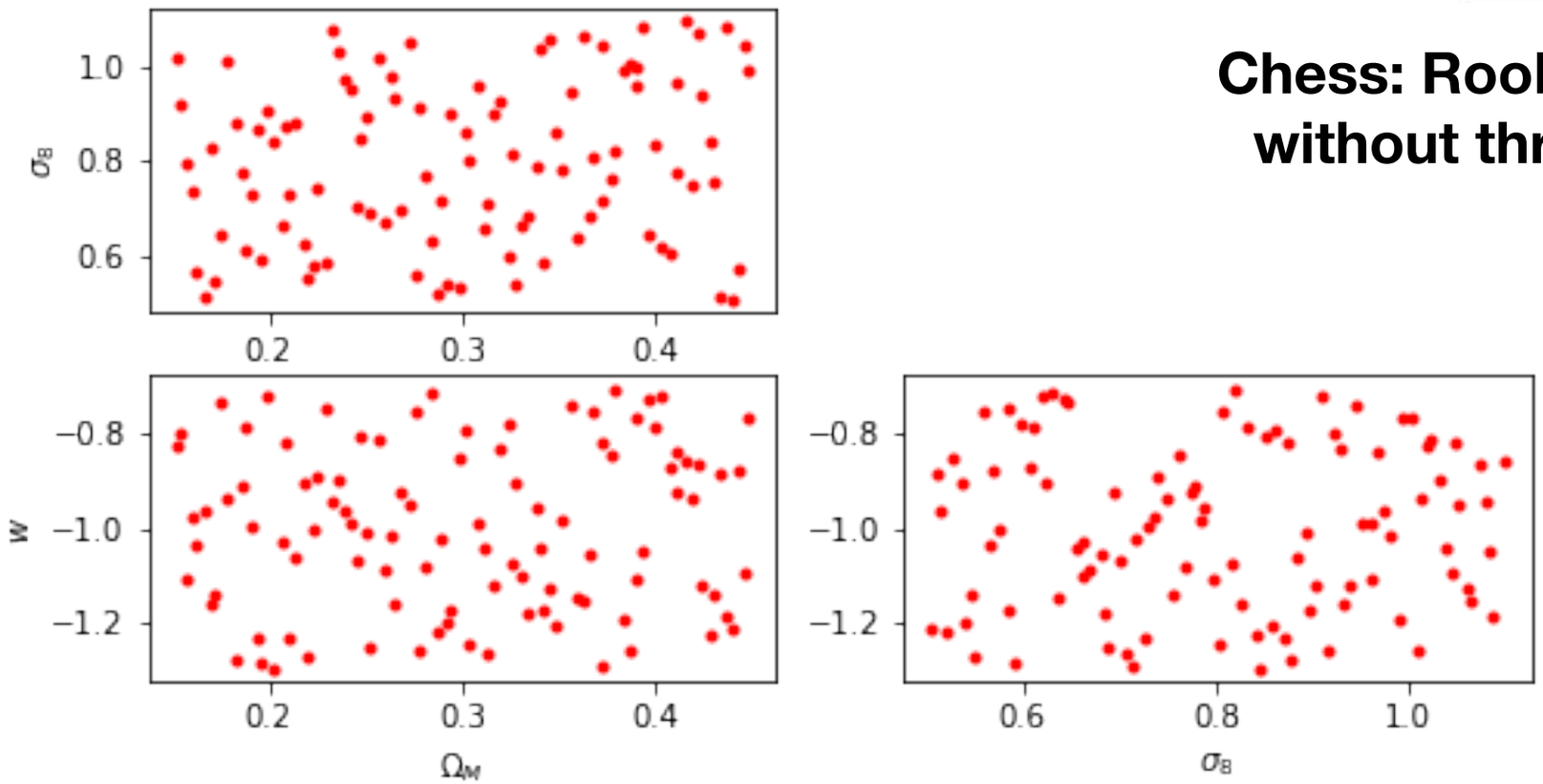
Can match almost all halos down to ~2E11 Msol with a fraction of computational time

Deep Learning the Expansion History

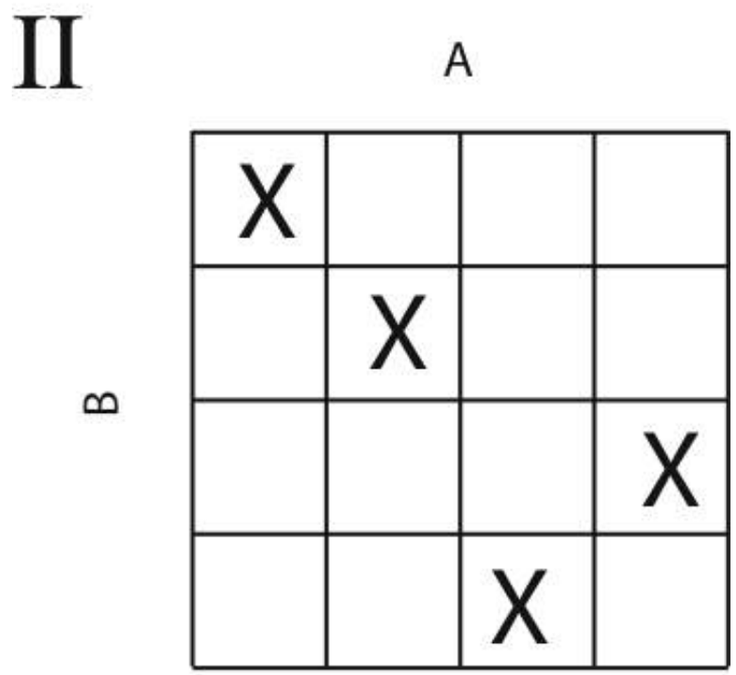
Fast approx N-body method

- 2 Gpc/h box, 1024^3 particles
- FoF halos
- Light-cone output 0<z<2
- 6 redshift slices

So far 100 simulations run
Exploring Om, s8 and w
Parameters sampled from a orthogonal Latin Hypercube



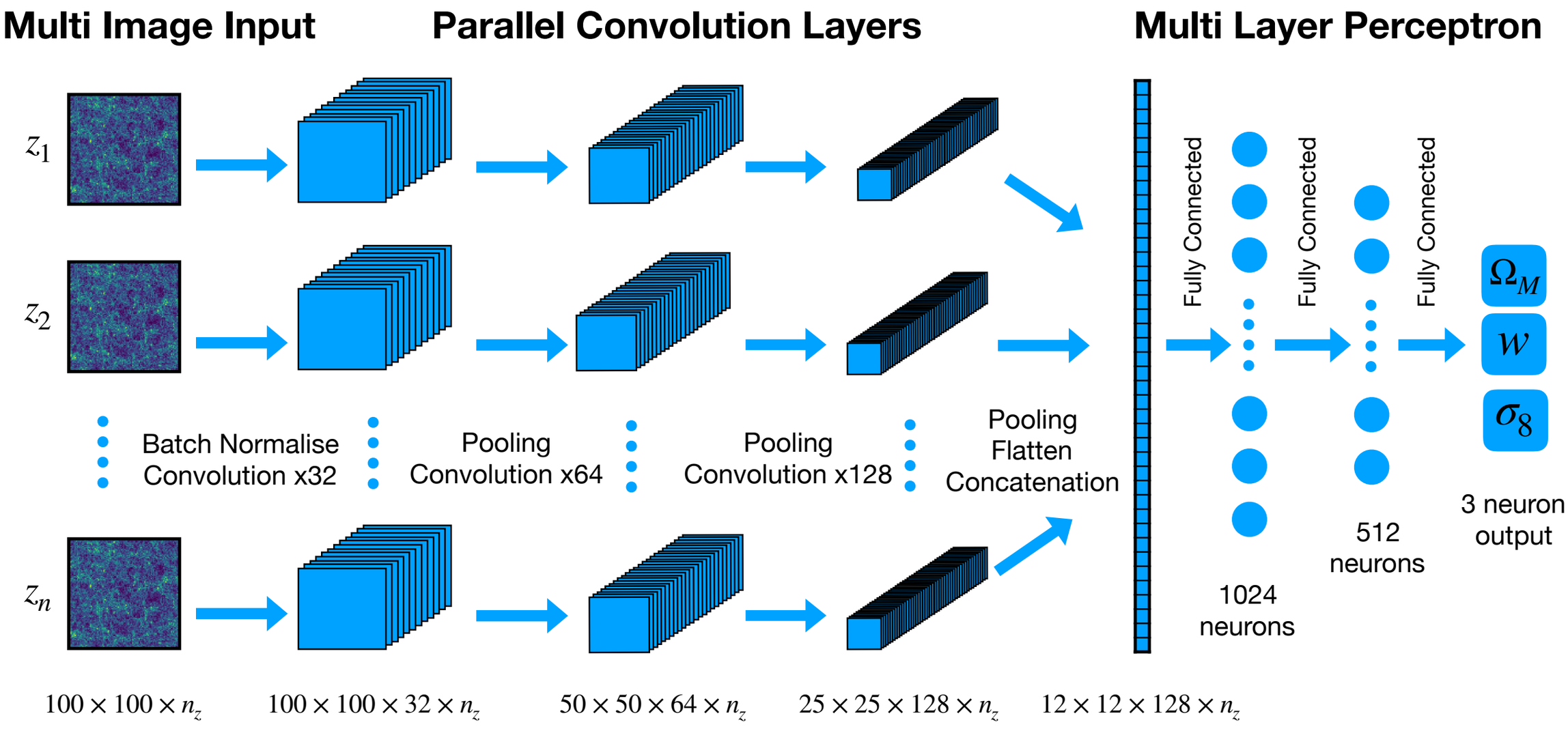
Training Data



Chess: Rook on each row and file without threatening each other

Deep Learning the Expansion History

Neural Network Architecture

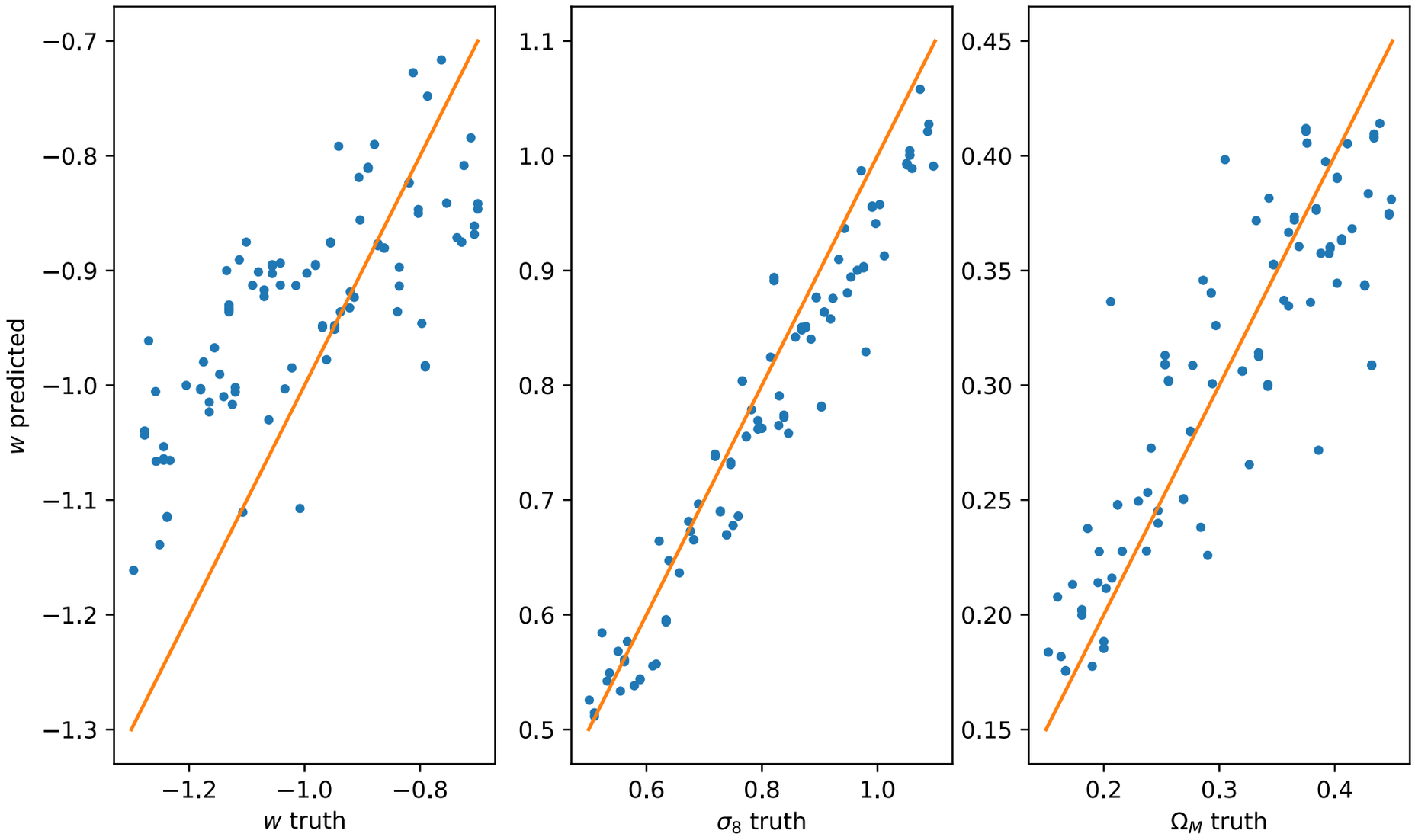


Deep Learning the Expansion History

Preliminary Results

Promising results

- **More training data**
- **Experiment different architecture and hyper parameters**
- **Test redshift space distortions / photo-z errors / etc**



Future Plan

The Korea Institute of Science and Technology Information (KISTI) Supercomputer Center recently had an open call for proposals specifically aimed at Machine Learning and Big Data.

We were successful and awarded 15 million CPU hours on ‘Nurion’ (14th fastest in the world)



It is a Cray CS500 system with 570,020 cores

Proposal:
- run ~1000 simulations
- Create realistic galaxy light-cones
- Find halos
- Apply a galaxy assignment model or several models (eg HODs)

- Add observational effects, redshift distortions, photo-z, incompleteness, etc

- Train models and constrain parameters within LCDM

- Apply to observational data, KiDS, SDSS, DLS, early DESI data