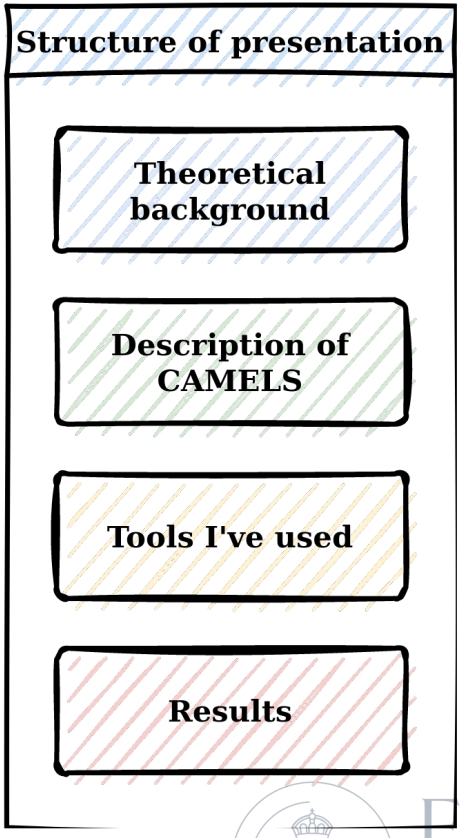
CAMELS Multifield Dataset

Balázs Pál
2022, Eötvös Loránd University

ELTE
EÖTVÖS LORÁND
UNIVERSITY

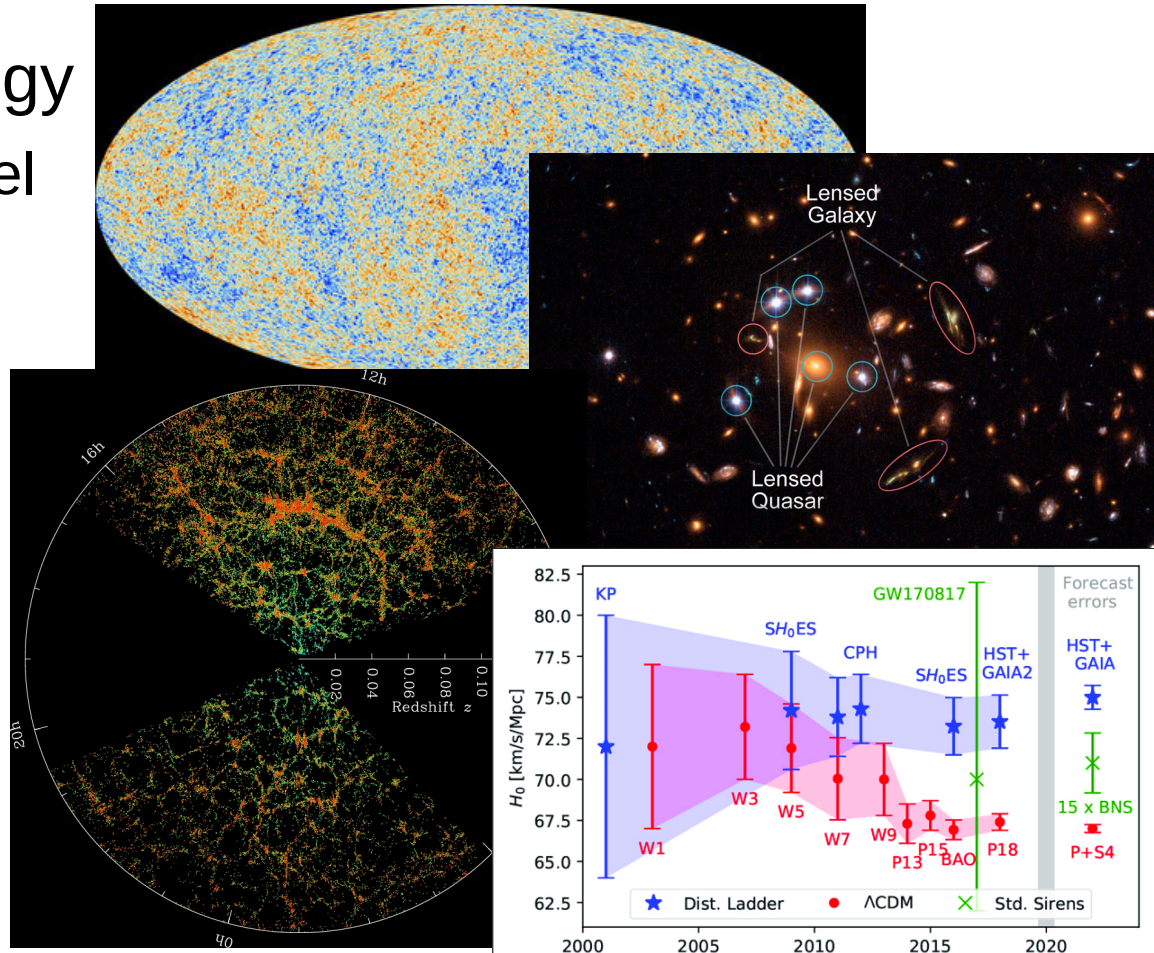
Overview

- Astrophysical background
  - What's the physical meaning of what we study here?
- Description of the CAMELS dataset
- Technical details
  - How do we approach the dataset? (CNN model)
  - What tools am I using and why I'm using those exactly?
- My work and results



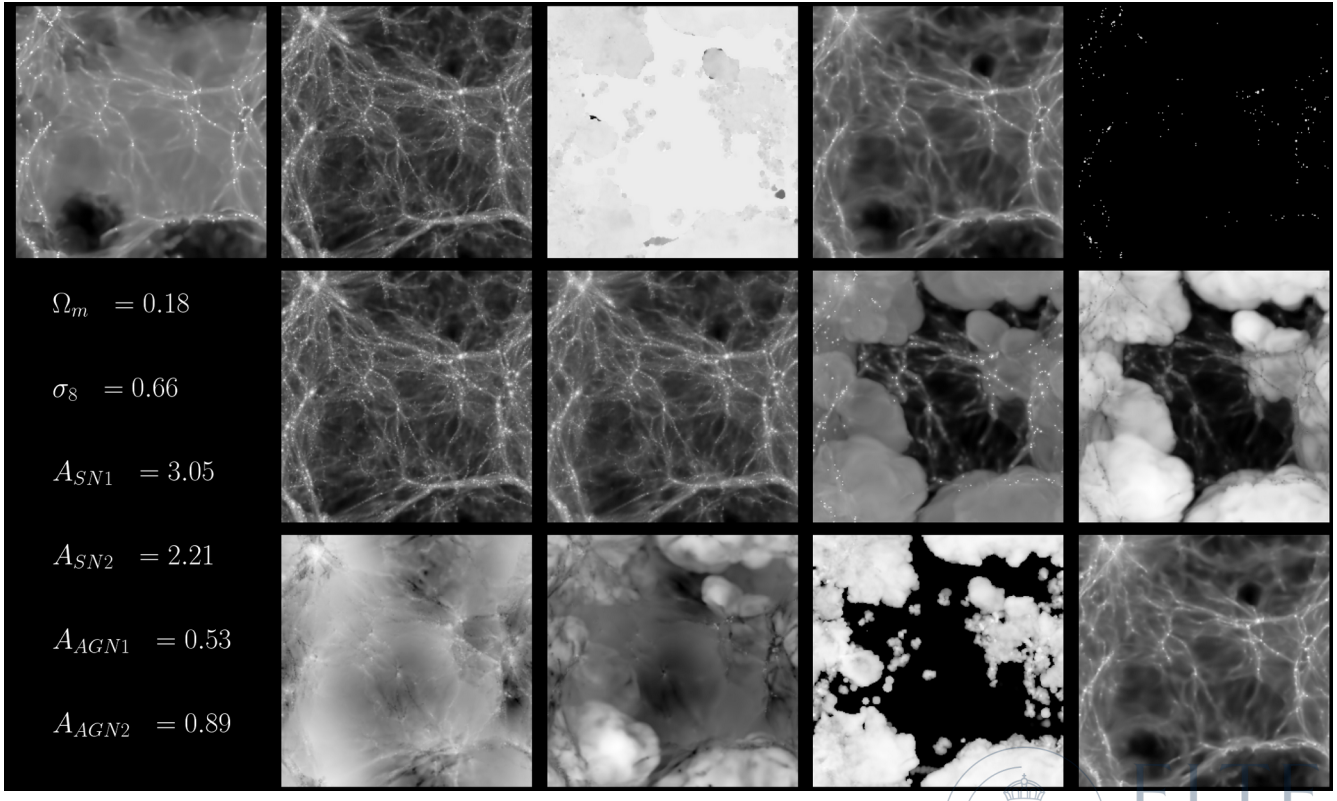
Astrophysical background

- Goals of modern cosmology
  - Improve the standard model of cosmology
  - Yes, that's it basically
- Solve challanges of the ΛCDM model
  - $\Omega_m - \sigma_8$ tension
  - H_0 tension (CMB vs. SN)
  - …



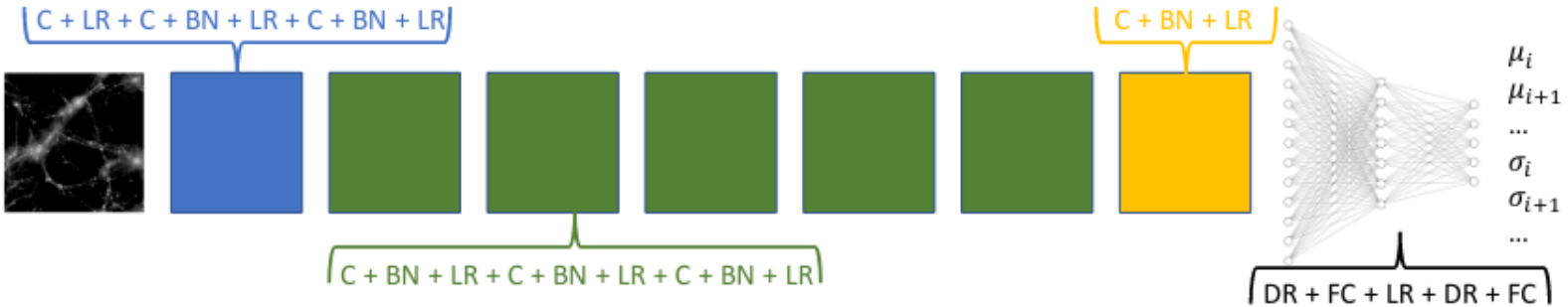
Description of the CAMELS dataset

- Comprehensive collection of simulations
- Purposefully designed for machine learning (**C**osmology and **A**strophysics with **M**achin**E** **L**earning **S**imulation)
- The CAMELS Multifield Dataset contains 1,000 HD and 1,000 MHD simulations
- 15 slices (5 Mpc/h) each for every different channel for every simulation



Technical details

- A basic CNN architecture was used
- 256^2 pixel images were reduced to 12 output values (6-6 values for the mean and variance of the posterior)
- The model was trained on all target parameters at the same time



C = Convolutional layer
BN = Batchnorm layer
LR = LeakyReLU activation layer
DR = Dropout
FC = Fully connected layer

Source: Villaescusa-Navarro, Francisco, et al. "Multifield Cosmology with Artificial Intelligence." arXiv preprint arXiv:2109.09747 (2021).

Technical details

- Two loss functions were used to get both mean and variance values during training

$$loss_1 = \frac{\sum_{axis=0} \left(y_{pred,mean} - y_{true} \right)^2}{n_{batch}}$$

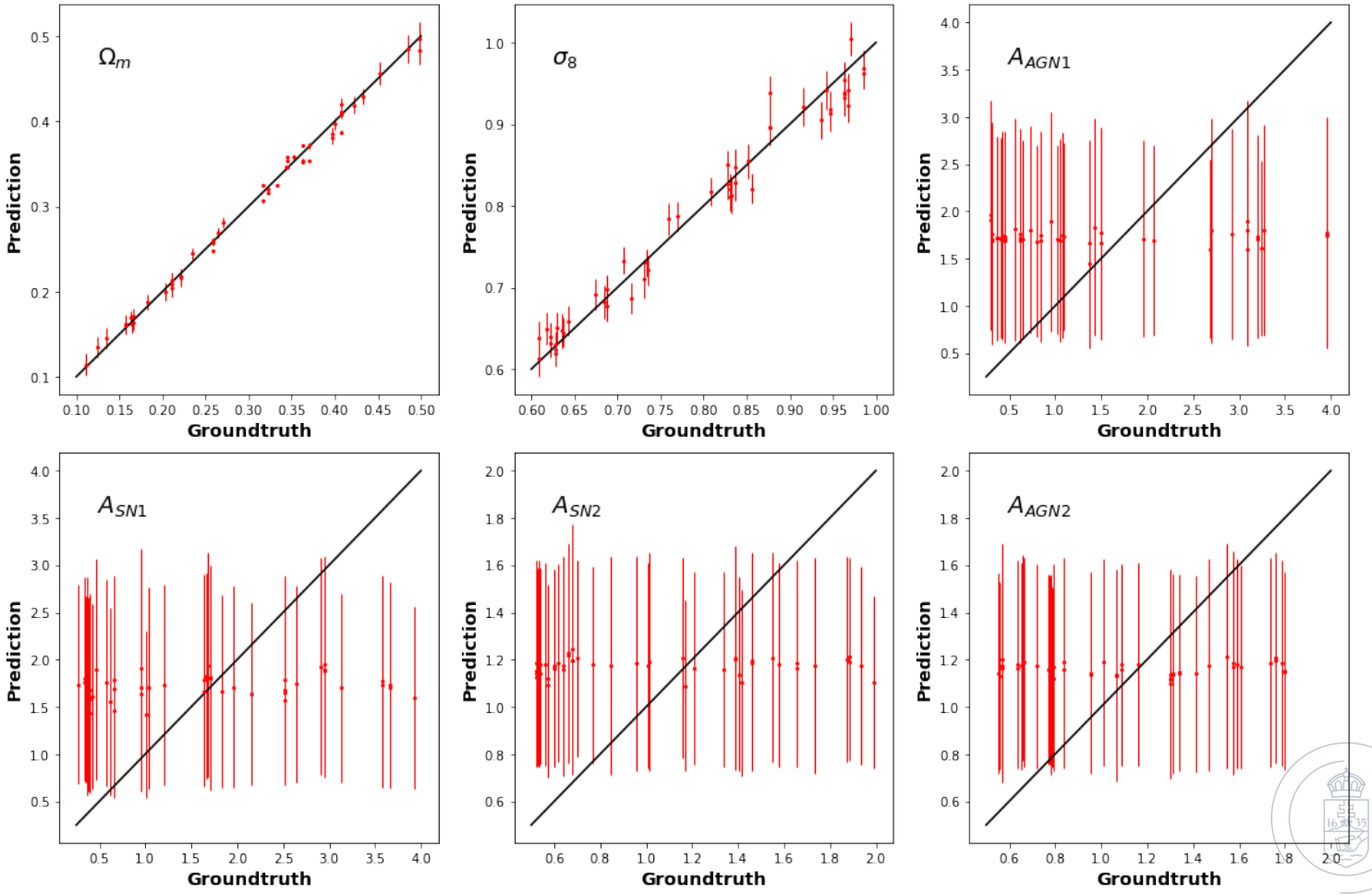
$$loss_2 = \frac{\sum_{axis=0} \left[\left(y_{pred,var} - y_{true} \right)^2 - y_{pred,var}^2 \right]^2}{n_{batch}}$$

$$loss_{total} = \frac{\sum \left[\log\left(loss_1\right) + \log\left(loss_2\right) \right]}{N_{parameters}}$$

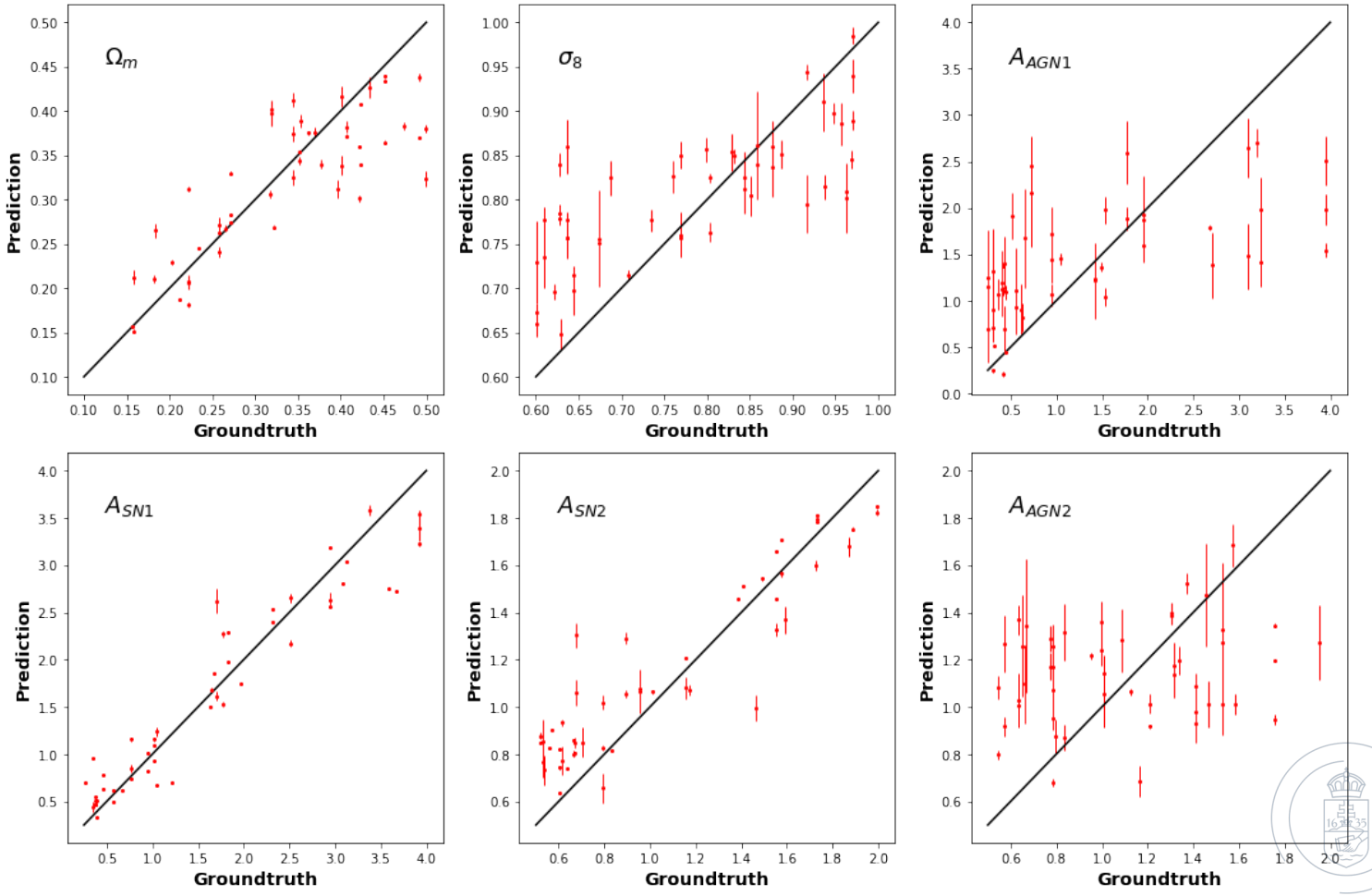
Technical details

- The architecture was constructed in Pytorch
- Pros:
  - Behaviour of TensorFlow is „strange” in this first half of 2022
  - There is already an existing Pytorch implementation of the network used in the article
- Cons:
  - Pytorch requires the input data to be in a very specific form, which means we need much more data preprocessing
  - Since we’re using **multifield** data, it needs even more preprocessing

My work and results



My work and results



My work and results