CAMELS Multifield Dataset

Release 1.0

Francisco Villaescusa-Navarro

Jul 14, 2023

CMD

CMD is a publicly available collection of hundreds of thousands 2D maps and 3D grids containing different properties of the gas, dark matter, and stars from more than 3,000 different universes. The data has been generated from thousands of state-of-the-art (magneto-)hydrodynamic and gravity-only N-body simulations from the CAMELS project.

Fig. 1: Examples of 2D maps from CMD. From top-left to bottom right: gas temperature, gas pression, neutral hydrogen mass, electron number, metallicity, gas mass, dark matter mass, total mass, stellar mass, magnetic fields, magnesium over iron, gas velocity, and dark matter velocity. Each map is characterized by the value of the cosmological parameters ($\Omega_{\rm m}$ and σ_8) and the astrophysical parameters ($A_{\rm SN1}$, $A_{\rm SN2}$, $A_{\rm AGN1}$, and $A_{\rm AGN2}$).

Each 2D map and 3D grid has a set of labels, or parameters, associated to it. Understanding the relation between the data and the labels will help cosmologists to decipher the mysteries of our own Universe.

CHAPTER

ONE

NEWS

July 2023: We have created a game to test the abilities of users to perform visual classification and regression using CMD images. Check *Game* for details.

July 2023: CMD now contains 2D maps and 3D grids from the CV, 1P, and EX simulation sets of CAMELS. This data can be useful for testing a variety of models.

June 2023: Hundreds of thousands of 2D maps and 3D grids created from the CAMELS-Astrid simulations have been added to CMD. Now CMD contains maps and grids from three different hydrodynamic codes. Check *Description* for details.

CHAPTER

TWO

SCIENCE

Our Universe is a strange place. All the things we know, from atoms to stars, only constitute about 5% of the Universe's content. The rest is split among dark matter (~25%) and dark energy (~70%). We believe dark matter should be some kind of elementary particle responsible to shaping up the large-scale structure of the Universe. On the other hand, we know very little about dark energy, besides being responsible of the current accelerated expansion of the Universe.

In the most accepted theoretical model, the Universe started with an explosion called the Big Bang. The fluctuations in the primordial plasma, originating from quantum fluctuations during cosmic inflation, were amplified by gravity; dark matter started clustering in those primordial gravitational potential wells, making them deeper and more massive, which in turn attracted more dark matter and gas to them. The gas fell into these potential wells and became cooler and denser, until stars were formed. This process gave rise to the abundance and spatial distribution of galaxies we observe in the Universe.

This model is able to precisely explain a very large and diverse set of cosmological observations, from the temperature anisotropies in the early Universe to the spatial distribution of galaxies in the present Universe. The model has some free parameters describing fundamental properties of the Universe, such as its age, geometry, composition…etc. Determining the values of these parameters with the highest accuracy is one of the main goals of modern cosmology. Knowing the values of these parameters as precisely as possible will allow us to shed light on some of the most fundamental questions in physics, such as: What is the nature of dark energy? What is the sum of the neutrino masses? How fast is the Universe expanding?

There is a huge amount of information in cosmological observations about the values of the cosmological parameters that is currently not being extracted since the non-linear evolution of cosmic structures gives rise to very complicated, and poorly understood, relationships between observables and these fundamental parameters. Cosmologists usually rely on summary statistics of observed data in order to describe those relationships, but a lot of information is lost in the process.

CHAPTER

THREE

DESCRIPTION

CMD contains hundreds of thousands of 2D maps and 3D grids created from cosmological simulations run with different codes. The CMD data is arranged into different files whose name indicate the properties of the simulations used to generate it. This is because the CMD data, as CAMELS, can be classified into into suites and sets (see this page for what concerns the CAMELS simulations):

3.1 Suites

CMD has been generated from thousands of state-of-the-art (magneto-)hydrodynamic and gravity-only N-body simulations from the CAMELS project. CMD data can be classified into different *suites*, that indicate the type of simulation used to create the data:

- **IllustrisTNG**. These magneto-hydrodynamic simulations follow the evolution of gas, dark matter, stars, and black-holes. They also simulate magnetic fields. CMD uses 1,088 of these simulations.
- **SIMBA**. These hydrodynamic simulations follow the evolution of gas, dark matter, stars, and black-holes. CMD uses 1,088 of these simulations.
- **Astrid**. These hydrodynamic simulations follow the evolution of gas, dark matter, stars, and black-holes. CMD uses 1,088 of these simulations.
- **N-body**. These gravity-only N-body simulation only follow the evolution of dark matter. Thus, they do not model astrophysical processes such as the formation of stars and the feedback from black-holes. There is an N-body simulation for each (magneto-)hydrodynamic simulation. CMD uses 2,000 of these simulations.

3.2 Sets

Each suite contains different *sets*, that indicate how the value of the labels of the underlying simulations are organized:

- **CV**. The value of the labels is always the same and correspond to the fiducial model. The 2D maps and 3D grids only differ on the initial conditions of the simulations run. This set contains 27 simulations.
- **1P**. The value of the labels is varied one-at-a-time. I.e. the 2D maps and 3D grids have labels whose value only differ in one element from the value of the fiducial maps (CV set). In this case, the initial conditions are always the same. This set contains 61 simulations.
- **LH**. The value of all labels is different in each simulation and the values are organized in a latin-hypercube. The value of the initial conditions is different in each simulation. This set contains 1,000 simulations.
- **EX**. The value of the labels is chosen to be *extreme* and the initial conditions of the simulations are the same. This set contains 4 simulations.

- **BE**. The underlying simulations have the same initial conditions and the same value of the labels (the fiducial ones). The only difference between the simulations is due to random noise from numerical approximations. This set contains 27 simulations. So far, this set is only present for the IllutrisTNG suite.

Attention: When working with CMD data, you will use files whose name will indicate the suite and the set. For instance, the file `Maps_Mcdm_Astrid_1P_z=0.00.npy` contains 2D maps of the cold dark matter field created from Astrid 1P simulations. In other workds, the simulations have been run with the Astrid model and their parameters follow the 1P configuration: all simulations have the same initial conditions but their parameters only vary from those of the fiducial ones in a single parameter.

3.3 Structure

CMD provides the following data generated from the above simulations:

IllustrisTNG

- 16,785 2D maps per field for 13 different fields. 218,205 2D maps in total.
- 16,380 3D grids per field for 13 different fields. 212,940 3D grids in total.

SIMBA

- 16,380 2D maps per field for 12 different fields. 196,560 2D maps in total.
- 16,380 3D grids per field for 12 different fields. 196,560 3D grids in total.

Astrid

- 16,380 2D maps per field for 12 different fields. 196,560 2D maps in total.
- 16,380 3D grids per field for 12 different fields. 196,560 3D grids in total.

Nbody

- 49,140 2D maps of one single field. 49,140 2D maps in total.
- 49,140 3D grids of one single field. 49,140 3D grids in total.

The table summarizes the properties of the 2D maps:

Field	Prefix	IllustrisTNG	SIMBA	Astrid	Nbody	Units
		Number of 2D maps				
Gas density	Mgas	16,785	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^2$
Gas velocity	Vgas	16,785	16,380	16,380	–	km/s
Gas temperature	T	16,785	16,380	16,380	–	Kelvin
Gas pressure	P	16,785	16,380	16,380	–	$h^2 M_\odot (\mathrm{km/s})^2/\mathrm{kpc}^3$
Gas metallicity	Z	16,785	16,380	16,380	–	dimensionless
Neutral hydrogen density	HI	16,785	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^2$
Electron number density	ne	16,785	16,380	16,380	–	$h^2/\mathrm{cm}^3(\mathrm{Mpc}/h)$
Magnetic fields	B	16,785	–	–	–	Gauss
Magnesium over Iron	MgFe	16,785	16,380	16,380	–	dimensionless
Dark matter density	Mcdm	16,785	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^2$
Dark matter velocity	Vcdm	16,785	16,380	16,380	–	km/s
Stellar mass density	Mstar	16,785	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^2$
Total matter density	Mtot	16,785	16,380	16,380	49,140	$(M_\odot/h)/(\mathrm{Mpc}/h)^2$
Total		218,205	196,560	196,560	49,140	

The table summarizes the properties of the 3D grids:

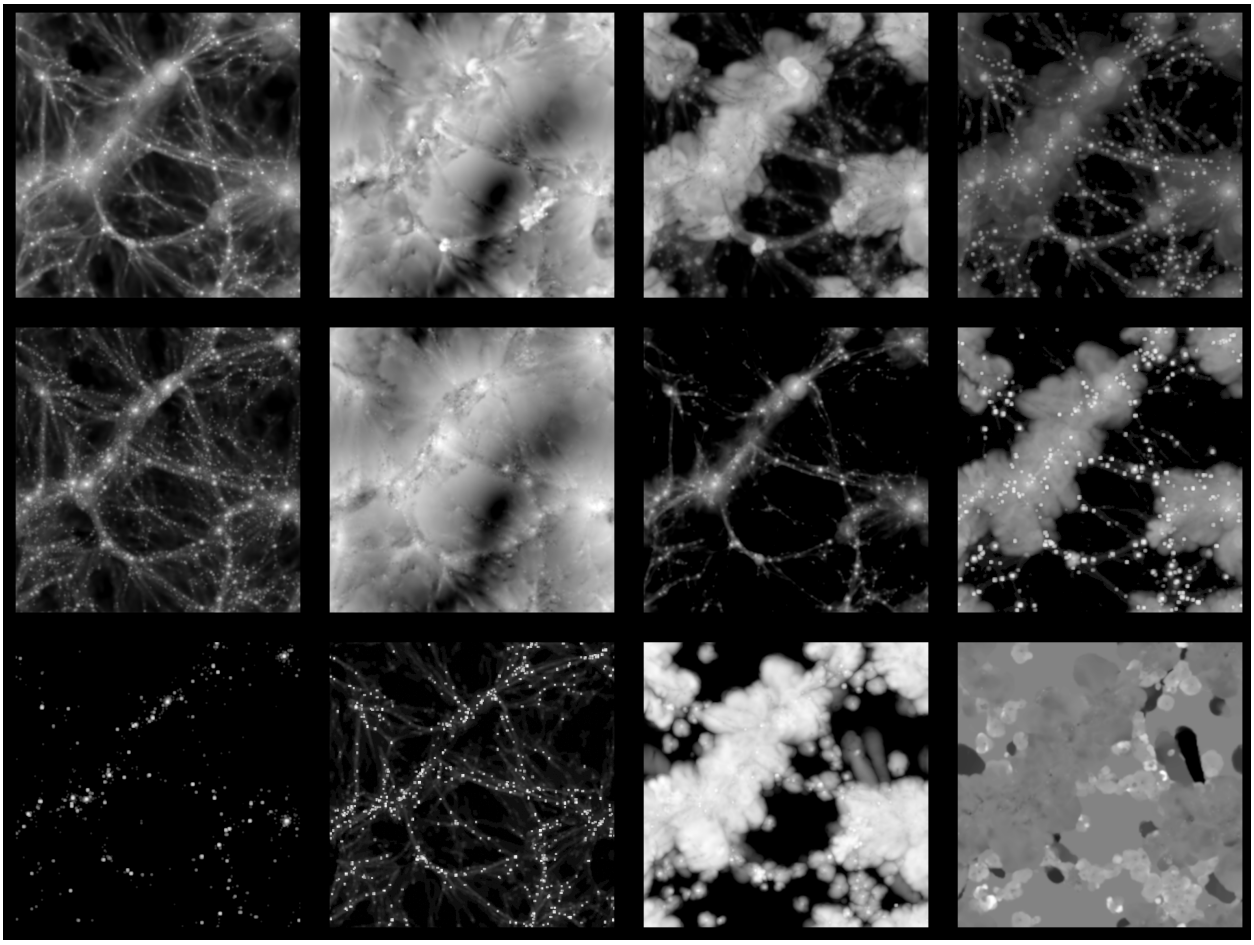
Field	Prefix	IllustrisTNG	SIMBA	Astrid	Nbody	Units
		Number of 3D grids				
Gas density	Mgas	16,380	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^3$
Gas velocity	Vgas	16,380	16,380	16,380	–	km/s
Gas temperature	T	16,380	16,380	16,380	–	Kelvin
Gas pressure	P	16,380	16,380	16,380	–	$h^2 M_\odot (\mathrm{km/s})^2/\mathrm{kpc}^3$
Gas metallicity	Z	16,380	16,380	16,380	–	dimensionless
Neutral hydrogen density	HI	16,380	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^3$
Electron number density	ne	16,380	16,380	16,380	–	h^2/cm^3
Magnetic fields	B	16,380	–	–	–	Gauss
Magnesium over Iron	MgFe	16,380	16,380	16,380	–	dimensionless
Dark matter density	Mcdm	16,380	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^3$
Dark matter velocity	Vcdm	16,380	16,380	16,380	–	km/s
Stellar mass density	Mstar	16,380	16,380	16,380	–	$(M_\odot/h)/(\mathrm{Mpc}/h)^3$
Total matter density	Mtot	16,380	16,380	16,380	49,140	$(M_\odot/h)/(\mathrm{Mpc}/h)^3$
Total		212,940	196,560	196,560	49,140	

where $M_\odot$ represents the mass of the Sun, km/s stands for kilometers per second, h is the reduced Hubble constant, that in all CMD is fixed to 0.67, and kpc stands for kiloparsec (3,260 light years). The coefficient A is 2 for 2D maps and 3 for 3D grids.

Warning: We note that some of the units reported in the CMD paper (see Table 1) are not correct. The units for the electron density are missing several factors and the pressure units lacks a h^2 factor. The above table shows the correct units of the 2D maps and 3D grids.

Note: All 2D maps have 256^2 pixels and cover a periodic area of $(25\ h^{-1}\mathrm{Mpc})^2$ at redshift 0. The 3D grids contain 128^3, 256^3 or 512^3 voxels over a volume of $(25\ h^{-1}\mathrm{Mpc})^3$ and are at redshifts 0, 0.5, 1, 1.5, and 2.

We show an example of how the IllustrisTNG images look like for the different fields:



where from top-left to bottom-right: gas density, gas velocity, gas temperature, gas pressure, dark matter density, dark matter velocity, electron number density, magnetic fields, stellar mass density, neutral hydrogen mass density, gas metallicity, and magnesium over iron ratio.

These images show different properties of the gas, dark matter, and stars in a given Universe. Determining the value of the cosmological parameters from these images will help us to decode the true value of our own Universe, allowing us to unveil some of the biggest mysteries in fundamental physics.

3.4 Labels

Each 2D map and 3D grid has a set of labels attached to it:

- $\Omega_{\rm m}$. This is a cosmological parameter that represents the fraction of matter in the Universe.
- σ_8. This is a cosmological parameter that controls the smoothness of the distribution of matter in the Universe.
- $A_{\rm SN1}$ and $A_{\rm SN2}$. These are two astrophysical parameters that controls two properties of supernova feedback.
- $A_{\rm AGN1}$ and $A_{\rm AGN2}$. These are two astrophysical parameters that control two properties of black-hole feedback.

The data from the IllustrisTNG, SIMBA, and Astrid simulations are described by all the above parameters, while the 2D maps and 3D grids generated from the N-body simulations are only characterized by the cosmological parameters $\Omega_{\rm m}$ and σ_8.

3.5 2D maps

The generic name of the files containing the maps is `Maps_prefix_suite_set_z=0.00.npy`, where `prefix` is the word identifying each field (see table above), `suite` is the suite (`IllustrisTNG`, `SIMBA`, `Astrid`, `Nbody_IllustrisTNG`, `Nbody_SIMBA`, or `Nbody_Astrid`) and `set` is the set (`1P`, `CV`, `LH`).

Note: In the case of the Nbody data we add an extra word, `IllustrisTNG`, `SIMBA`, or `Astrid`, to characterize the matching data from the (magneto-)hydrodynamics simulations. See *Matching data* for further details.

For instance, the file containing the gas density maps of the IllustrisTNG simulations is `Maps_Mgas_IllustrisTNG_LH_z=0.00.npy`. The 2D maps are stored as `.npy` files, and can be read with the numpy `load` routine. For instance, to read the SIMBA gas temperature maps do:

```
import numpy as np

# name of the file
fmaps = 'Maps_T_SIMBA_LH_z=0.00.npy'

# read the data
maps = np.load(fmaps)
```

The file contains 15,000 maps with 256^2 pixels each.

We note that the name of the files for the Nbody 2D maps is slighty different to reflect the (magneto-)hydrodynamic simulation they should be matched on:

The values of the cosmological and astrophysical parameters characterizing the maps of a given field are given in `params_sim.txt` where `suite` can be `IllustrisTNG`, `SIMBA`, `Astrid`, or `Nbody`. These files can be read as follows:

```
import numpy as np

# name of the file
fparams = 'params_SIMBA.txt'

# read the data
params = np.loadtxt(fparams)
```

The file contains 1,000 entries with 6 values per entry. The first and second entries are the values of $\Omega_{\rm m}$ and σ_8, while the rest represent the values of the astrophysical parameters: $A_{\rm SN1}$, $A_{\rm AGN1}$, $A_{\rm SN2}$, $A_{\rm AGN2}$.

Note: In the case of the `Nbody` maps, only the first and second columns (the ones containing the values of $\Omega_{\rm m}$ and σ_8) are relevant. The other 4 columns can be disregarded (because the Nbody simulations do not model supernovae and black holes). They are only kept to standardize the training of the networks.

The values of the cosmological and astrophysical parameters of a given map can be found as

```
map_number = 765
params_map = params[map_number//15]
```

See this colab for further details on how to manipulate the images and the values of the parameters.

Note: 2D maps can be generated from 3D grids by taking slides and projecting along a given axis. See this colab for

an example.

3.6 3D grids

The generic name of the files containing the 3D grids is `Grids_prefix_suite_set_grid_z=redshift.npy`, where `prefix` is the word identifying each field (see table above), `suite` can be `IllustrisTNG`, `SIMBA`, `Astrid`, `Nbody_IllustrisTNG`, `Nbody_SIMBA` or `Nbody_Astrid`, `set` can be `1P`, `CV`, `LH`, `grid` can be `128`, `256`, or `512` and `redshift` can be 0, 0.5, 1, 1.5 or 2.

Note: In the case of the Nbody data we add an extra word, `IllustrisTNG`, `SIMBA` or `Astrid`, to characterize the matching data from the (magneto-)hydrodynamics simulations. See *Matching data* for further details.

For instance, the file containing the 3D gas metallicity of the IllustrisTNG simulations on a grid with `256^3` voxels at redshift 0 is `Grids_Z_IllustrisTNG_LH_256_z=0.00.npy`. The 3D grids are stored as `.npy` files, and can be read with the numpy `load` routine. For instance, to read the SIMBA neutral hydrogen mass density at redshift 1.0 with a grid of `128^3` voxels do:

```
import numpy as np

# name of the file
fgrids = 'Grids_HI_SIMBA_LH_128_z=0.00.npy'

# read the data
grids = np.load(fgrids)
```

The file contains 1,000 grids with 128^3 voxels each. For large files (e.g. those containing the grids with 512^3 voxels) it is better to read the files in a slightly different way, to avoid running out of RAM memory:

```
import numpy as np

# name of the file
fgrids = 'Grids_Mcdm_Nbody_LH_512_z=0.00.npy'

# read the data
grids = np.load(fgrids, mmap_mode='r')

# take the first 3D grid
grids[0]

# multiply all the grids from numbers 672 to 700 by 3
grids[672:700]*3
```

The values of the cosmological and astrophysical parameters characterizing the maps of a given field can be found in `params_set_suite.txt` where `suite` can be `IllustrisTNG`, `SIMBA`, `Astrid`, or `Nbody`, and `set` can be `1P`, `CV`, or `LH`. These files can be read as follows:

```
import numpy as np

# name of the file
fparams = 'params_LH_SIMBA.txt'
```

(continues on next page)

(continued from previous page)

```
# read the data
params = np.loadtxt(fparams)
```

The file contains 1,000 entries with 6 values per entry. The first and second entries are the values of $\Omega_{\rm m}$ and σ_8, while the rest represent the values of the astrophysical parameters: $A_{\rm SN1}$, $A_{\rm AGN1}$, $A_{\rm SN2}$, $A_{\rm AGN2}$.

Note: In the case of the `Nbody` maps, only the first and second columns (the ones containing the values of $\Omega_{\rm m}$ and σ_8) are relevant. The other 4 columns can be disregarded (because the Nbody simulations do not model supernovae and black holes). They are only kept to standardize the training of the networks.

The value of the cosmological and astrophysical parameters of a given grid can be found as

```
grid_number = 821
params_map  = params[map_number]
```

3.7 Symmetries

Each 2D map and 3D grid from CMD has a set of labels associated to it: two cosmological parameters and four astrophysical parameters (only in the case of data from IllustrisTNG, SIMBA, and Astrid simulations). These labels will remain the same if

- rotations
- translations
- parity

transformations are applied to the data. Another important thing to take into account is that the data is periodic in all dimensions. For instance, in the case of 2D maps

```
import numpy as np

# name of the file
fmaps = 'Maps_HI_IllustrisTNG_LH_z=0.00.npy'

# read the data
maps_HI = np.load(fmaps)

# take the map number 36
map_HI = maps_HI[36]

# the pixel map_HI[45,89] is adjacent to the pixel map_HI[46,89]
# the pixel map_HI[145,99] is adjacent to the pixel map_HI[145,98]
# the pixel map_HI[76,0] is adjancent to the pixel map_HI[76,255]
# the pixel map_HI[255,12] is adjancent to the pixel map_HI[0,12]
```

Note: When using convolutional neural networks, one can take advantage of this property by using periodic padding.

3.8 Matching data

There are several ways to match CMD.

1. The 2D maps and 3D grids can be matched across fields within the same simulation type. For instance, the maps number 2786 of the files `Maps_ne_IllustrisTNG_LH_z=0.0.npy` and `Maps_B_IllustrisTNG_LH_z=0.0.npy` represent the same region of the same simulation. The only difference is that the first map will show the electron abundance while the second shows the magnetic fields. The same thing applies to the 3D grids. For instance, the grids number 621 of the files `Grids_HI_SIMBA_LH_128_z=0.0.npy` and `Grids_Mgas_SIMBA_LH_128_z=0.0.npy` represent the same volume of the same simulation with the only difference that the first grid shows the neutral hydrogen mass density while the second contains the gas density.

Warning: This matching only applies to data within the same simulation. E.g. the files `Maps_Mcdm_IllustrisTNG_LH_z=0.0.npy` do not have any correspondence with the maps in the file `Maps_Mtot_SIMBA_LH_z=0.0.npy`.

2. The 3D grids can be matched across resolution within the same field and redshift. For instance, the grids number 167 of the files `Grids_Vcdm_SIMBA_LH_128_z=1.0.npy` and `Grids_Vcdm_SIMBA_LH_256_z=1.0.npy` represent exactly the same field over the same volume with the only difference that the first contains 128^3 voxels while the second has 256^3 voxels. Knowing this mapping is important for the *Superresolution* application.

3. The 2D maps and 3D grids can be matched between (magneto-)hydrodynamic and N-body simulations. For instance, the maps number 7413 of the files `Maps_Mtot_IllustrisTNG_LH_z=0.0.npy` and `Maps_Mtot_Nbody_IllustrisTNG_LH_z=0.0.npy` represent the same region of the same field (total matter), with the only difference that the first map was generated from an IllustrisTNG magneto-hydrodynamic simulation while the second one is from a gravity-only N-body simulation. Knowing this mapping is important to be able to quantify that impact of astrophysical processes on a given task.

Warning: This mapping only applies to the total matter field.

4. The 3D grids can be matched across cosmic time in both the (magneto-)hydrodynamic and the N-body simulations. For instance, the grids number 923 `Grids_Vgas_SIMBA_LH_512_z=0.0.npy` and `Grids_Vgas_SIMBA_LH_512_z=2.0.npy` represent the gas velocity of the same universe just at two different times: $z = 0$ in the first grid and $z = 2$ in the second grid.

Note: We do not recommend using the above time matching for the 2D maps. The reason is that in a simulation, particles will move with time, so particles that are in a given map at a given time may move to another map at a different time. While this is not a problem for the 3D grids, it may be a challenge for the 2D maps.

We note that the above three matchings can be combined. For instance, in the *N-body to hydro* application we want to find the mapping between the total matter from an N-body simulation and a given field from a (magneto-)hydrodynamic simulation. In this case, the grids number 714 of the files `Grids_T_SIMBA_LH_256_z=0.0.npy` and `Grids_Mtot_Nbody_SIMBA_LH_256_z=0.0.npy` represent the same region at redshift 0, the first grid will contain the gas temperature from the hydrodynamic simulation while the second is the total matter field from the equivalent N-body simulation.

3.9 Storage

Each pixel of a 2D map and each voxel of a 3D grid is stored as a float, i.e. it occupies 4 bytes.

A single 2D map that has 256^2 pixels will take $256^2 \times 4 = 0.25$ Mb. CMD is organized into files that contain different number of maps. For instance, the files of the LH set contain 15,000 maps per field. Each of those files would thus require 3.75 Gb. If you want to download all the maps of the IllustrisTNG LH set (13 different fields) you would need ~50 Gb.

A single 3D grid with N^3 voxels will take $N^3 \times 4$ bytes, i.e. 8 Mb for $N = 128$, 64 Mb for $N = 256$, or 512 Mb for $N = 512$. CMD is organized into files that contain different numbers of 3D grids. For instance, the files of the LH sets contain 1,000 grids. Each of those LH files will occupy 7.8 Gb ($N = 128$), 62.5 Gb ($N = 256$), and 500 Gb ($N = 512$). If you want to download all 12 grids of the LH set for SIMBA at $N = 512$ it will require ~6 Tb.

CHAPTER

FOUR

DATA ACCESS AND STRUCTURE

To work with CMD the user can either download the data or work with it online.

4.1 Download

CMD data can be downloaded in two different ways:

1. Via globus.
2. Via url.

Note: Transfering the data with the url is simple and convenient, but it can also be slow and unstable, in particular for the large files. We thus recommend using globus as it is the fastest and more reliable way to transfer the data.

4.2 Binder

We provide access to the data through binder. Binder is an online system where the data can be accessed and manipulated without the need to transfer it. The user can either use a terminal-like or a python notebook-like to view and manipulate the data. Note that this system is not meant to perform heavy calculations, but to explore the data. The user can find some documentation on binder here.

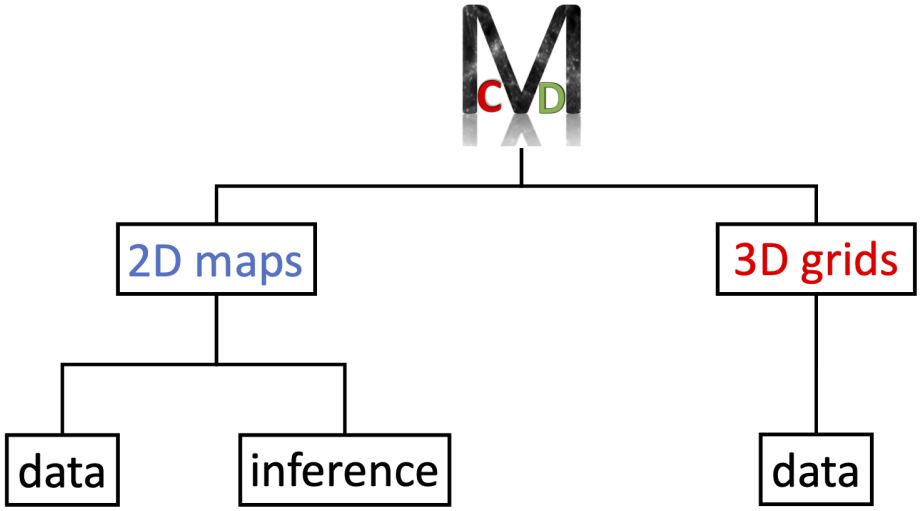
4.3 Structure

CMD data is organized as follows:

- **2D_maps**. This folder contains 2 subfolders:
  - **data**. It hosts all files with the 2D maps and the values of the cosmological and astrophysical parameters.
  - **inference**. It hosts the codes and network weights used to perform parameter inference with CMD.
- **3D_grids**. This folder only contains 1 subfolder:
  - **data**. All files with the 3D grids and the values of the cosmological and astrophysical parameters are here.

The total data volume of the 2D maps is approximate 100GB, and of the 3D grids is of 75TB.

The image below shows a scheme with the way the data is organized:


CMD
2D maps
3D grids
data
inference
data

CHAPTER

FIVE

GAME

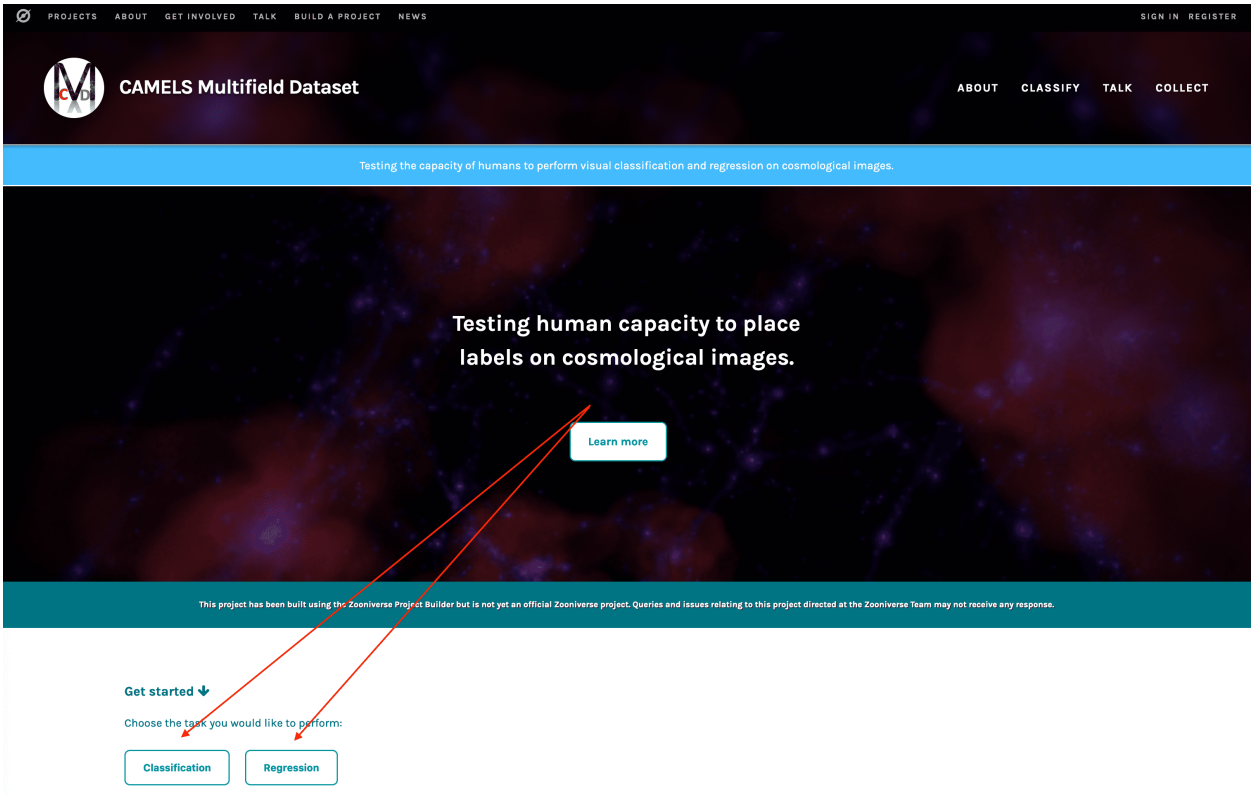
Do you want to test your classification and regression capabilities? We have created an app using CMD data where the user can guess

1) the field an image represent (classification)
2) the value of the cosmological parameters of an image (regression)

You can find the game here that is hosted by the Zooniverse platform.

5.1 Classification

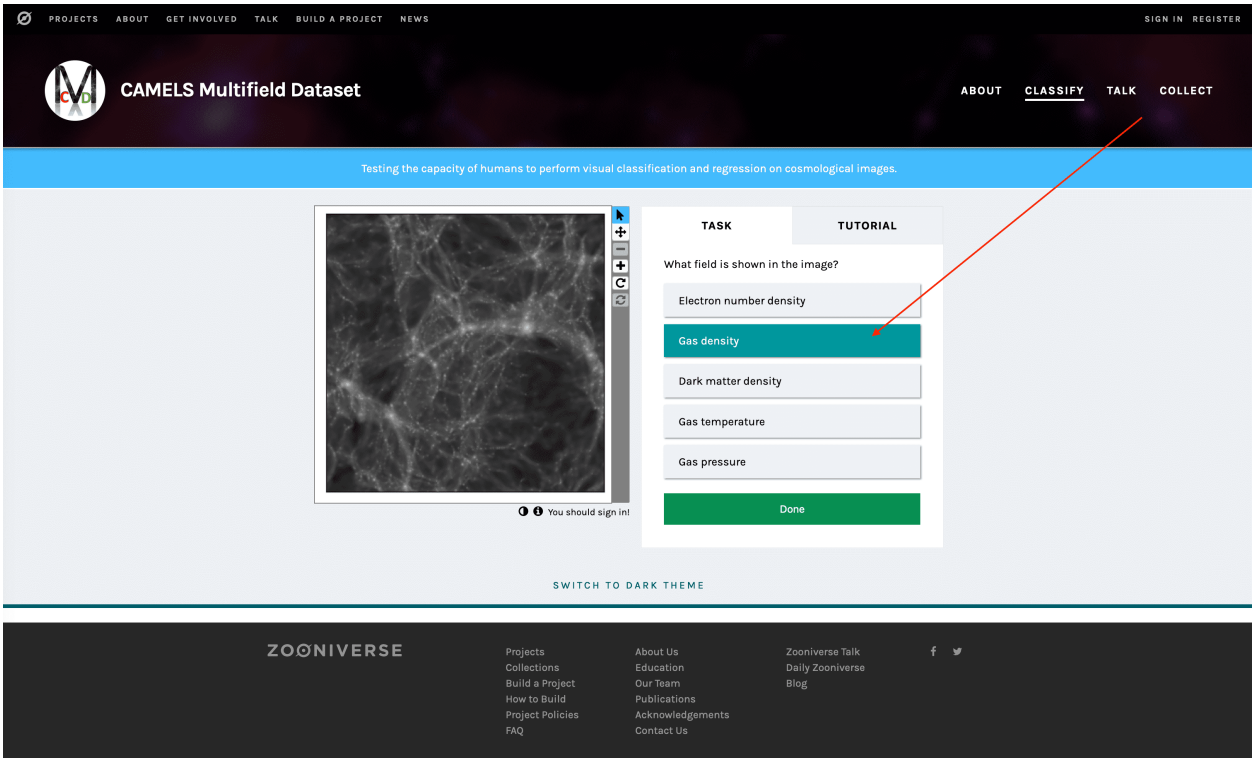
Go to the game website and click on the classification buttom:



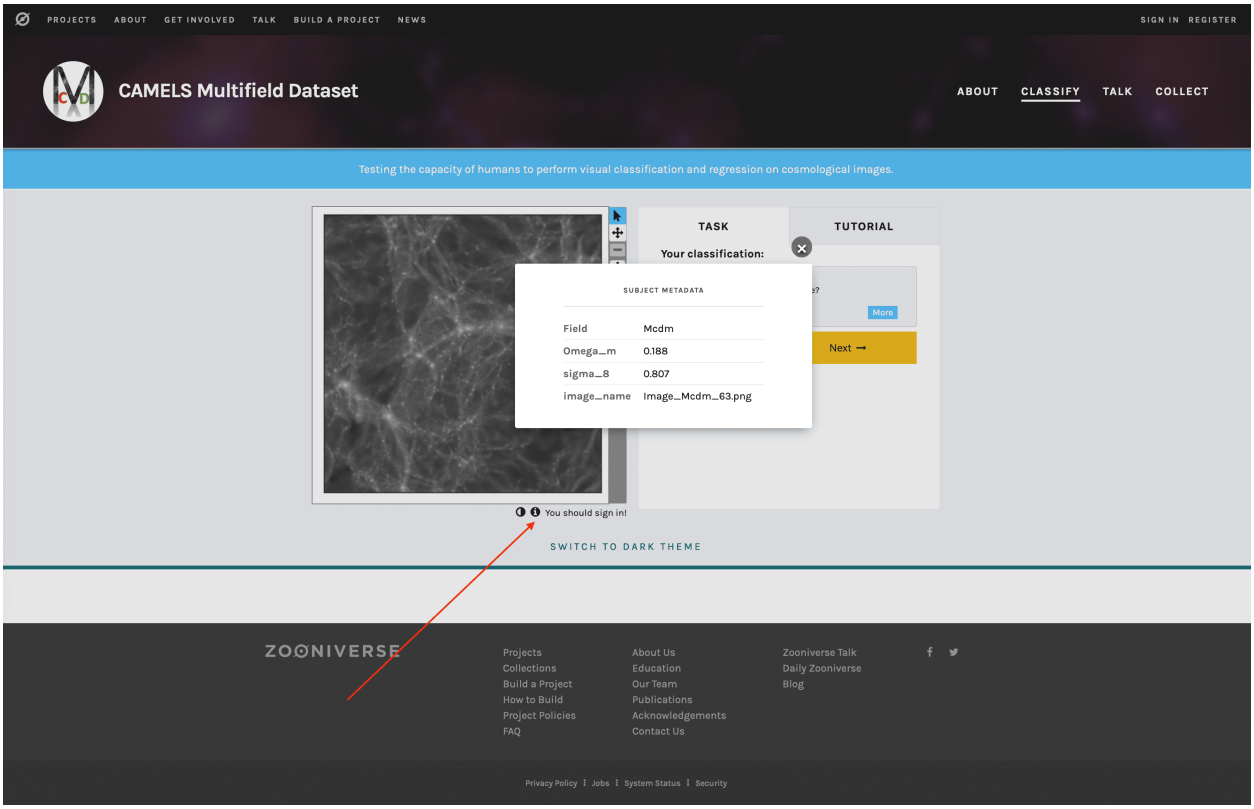
Take a look at the image shown and click on the field you think it represents:

- Dark matter density

- Gas density
- Gas temperature
- Gas pressure
- Electron number density



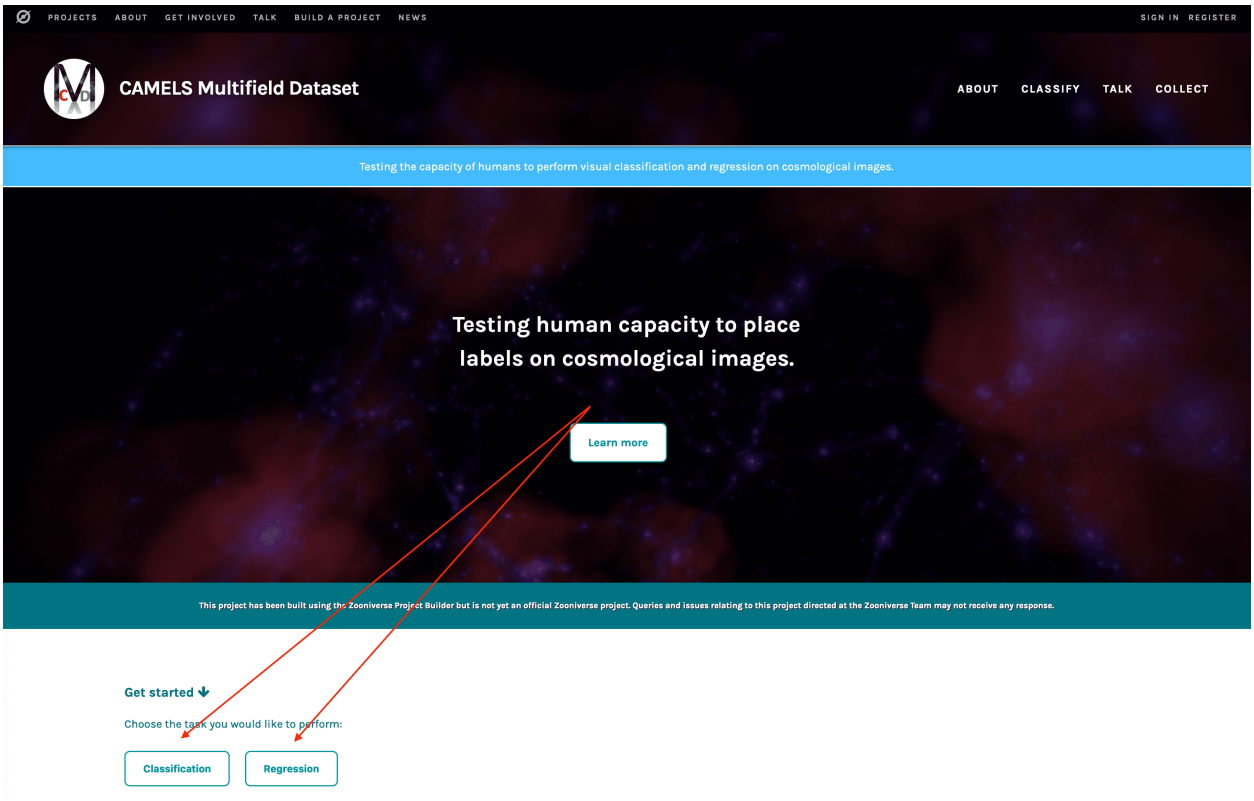
Click on the Done buttom. Next, click on the information buttom to see the true label



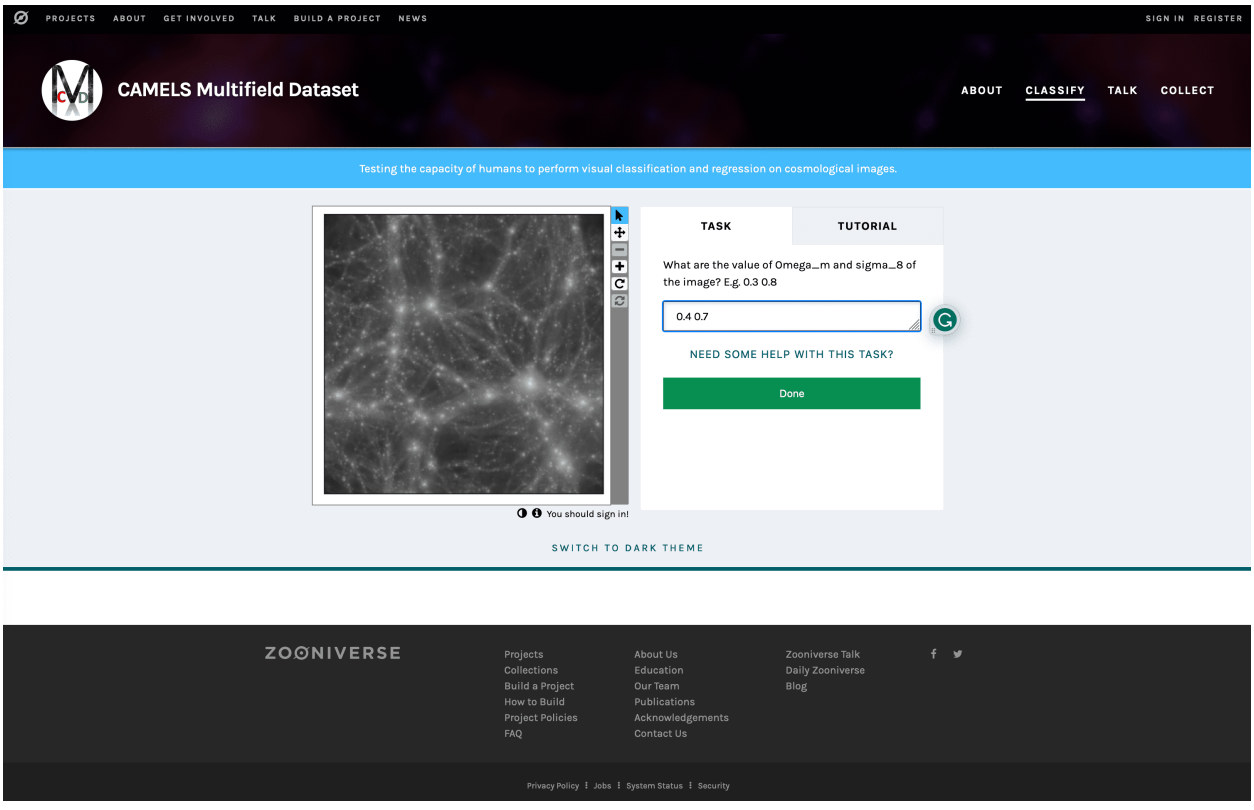
Note: It is possible to see the true label before selecting the field. Keep in mind that this is designed as a game whose main purpose is to train a user to perform these tasks.

5.2 Regression

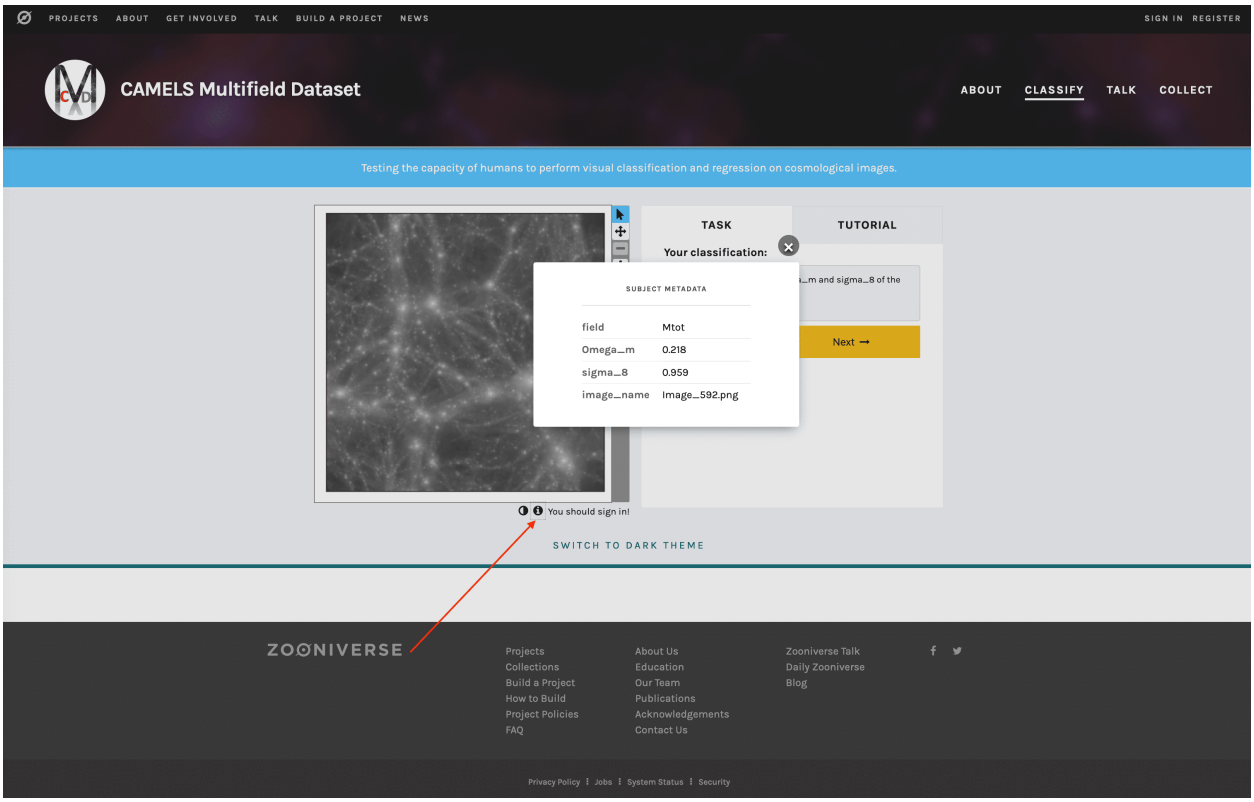
Go to the game website and click on the regression buttom:



Take a look at the image shown and write the value of $Omega_{\rm m}$ and σ_8 you think the image has.



Click on the Done buttom. Next, click on the information buttom to see the true values of the parameters



Note: It is possible to see the true labels before selecting the field. Keep in mind that this is designed as a game whose main purpose is to train a user to perform these tasks.

CHAPTER

SIX

PUBLICATIONS

CMD belongs to the CAMELS project and in order to avoid duplicates or missing entries we refer the reader to the CAMELS publication website here.

CHAPTER

SEVEN

CITATION

If you made use of CMD data please cite the CAMELS Multifield Dataset paper and the CAMELS project paper:

```
@ARTICLE{CMD,
    author = {{Villaescusa-Navarro}, Francisco and {Genel}, Shy and {Angles-Alcazar},␣
↪Daniel and {Thiele}, Leander and {Dave}, Romeel and {Narayanan}, Desika and {Nicola},␣
↪Andrina and {Li}, Yin and {Villanueva-Domingo}, Pablo and {Wandelt}, Benjamin and
↪{Spergel}, David N. and {Somerville}, Rachel S. and {Zorrilla Matilla}, Jose Manuel␣
↪and {Mohammad}, Faizan G. and {Hassan}, Sultan and {Shao}, Helen and {Wadekar},␣
↪Digvijay and {Eickenberg}, Michael and {Wong}, Kaze W.~K. and {Contardo}, Gabriella␣
↪and {Jo}, Yongseok and {Moser}, Emily and {Lau}, Erwin T. and {Machado Poletti Valle},␣
↪Luis Fernando and {Perez}, Lucia A. and {Nagai}, Daisuke and {Battaglia}, Nicholas and
↪{Vogelsberger}, Mark},
     title = "{The CAMELS Multifield Dataset: Learning the Universe's Fundamental␣
↪Parameters with Artificial Intelligence}",
   journal = {arXiv e-prints},
  keywords = {Computer Science - Machine Learning, Astrophysics - Cosmology and␣
↪Nongalactic Astrophysics, Astrophysics - Astrophysics of Galaxies, Astrophysics -␣
↪Instrumentation and Methods for Astrophysics, Computer Science - Computer Vision and␣
↪Pattern Recognition},
      year = 2021,
     month = sep,
     eid = {arXiv:2109.10915},
     pages = {arXiv:2109.10915},
     archivePrefix = {arXiv},
     eprint = {2109.10915},
     primaryClass = {cs.LG},
     adsurl = {https://ui.adsabs.harvard.edu/abs/2021arXiv210910915V},
     adsnote = {Provided by the SAO/NASA Astrophysics Data System}
     }

@ARTICLE{CAMELS,
    author = {{Villaescusa-Navarro}, Francisco and {Angl{\'e}s-Alc{\'a}zar}, Daniel and
↪{Genel}, Shy and {Spergel}, David N. and {Somerville}, Rachel S. and {Dave}, Romeel␣
↪and {Pillepich}, Annalisa and {Hernquist}, Lars and {Nelson}, Dylan and {Torrey}, Paul␣
↪and {Narayanan}, Desika and {Li}, Yin and {Philcox}, Oliver and {La Torre}, Valentina␣
↪and {Maria Delgado}, Ana and {Ho}, Shirley and {Hassan}, Sultan and {Burkhart},␣
↪Blakesley and {Wadekar}, Digvijay and {Battaglia}, Nicholas and {Contardo}, Gabriella␣
↪and {Bryan}, Greg L.},
     title = "{The CAMELS Project: Cosmology and Astrophysics with Machine-learning␣
↪Simulations}",
```

(continues on next page)

(continued from previous page)

```
   journal = {\apj},
  keywords = {Cosmology, Cosmological parameters from large-scale structure, Galaxy␣
↪formation, Astrostatistics, 343, 340, 595, 1882, Astrophysics - Cosmology and␣
↪Nongalactic Astrophysics, Astrophysics - Astrophysics of Galaxies, Astrophysics -␣
↪Instrumentation and Methods for Astrophysics},
      year = 2021,
     month = jul,
    volume = {915},
    number = {1},
       eid = {71},
     pages = {71},
       doi = {10.3847/1538-4357/abf7ba},
    archivePrefix = {arXiv},
  eprint = {2010.00619},
  primaryClass = {astro-ph.CO},
    adsurl = {https://ui.adsabs.harvard.edu/abs/2021ApJ...915...71V},
   adsnote = {Provided by the SAO/NASA Astrophysics Data System}
   }
```

CHAPTER

EIGHT

PARAMETER INFERENCE

Below we describe this task and the benchmark model. We also discuss the main problems and challenges involved in this task.

8.1 Description

One of the most obvious applications of CMD is parameter inference. Each 2D map and 3D grid is characterized by 6 numbers: two cosmological parameters describing fundamental properties of the Universe and four astrophysical parameters that quantify the efficiency of astrophysical processes supernova explosions or feedback from black holes. The goal of this application is to develop robust models that can infer the value of the two cosmological parameters from the data with the highest accuracy.

8.2 Benchmark

Below we describe the benchmark model for this task. We note that this only applies to 2D maps.

8.2.1 Requisites

- python3
- numpy
- Pytorch
- optuna

8.2.2 Organization

The folder containing the codes and weights is called `benchmark`, and is located within the `2D_maps` folder. Inside it, there are three folders called `scripts`, containing the codes used to train and test the networks, `databases` with the optuna databases and `weights`, which contains the weights of the networks.

8.2.3 scripts

This folder contains the following codes:

- `architecture.py`. This script contains different architecture models.
- `data.py`. This script processes the data to train the networks.
- `train.py`. This is the code used to train the network.
- `test.py`. This is the code used to test the network.

To train a new model on a given field, or a multifield, follow these steps:

1) Open the code `train.py` and set the value of the different parameters in the INPUT section.
2) Run the code: `python train.py`. Note that when using optuna, several trials can be run in parallel (see the optuna documentation for details).
3) The code will output an optuna database, and will save the losses and weights of each of the considered trials.

To test a model on a given field(s), or a multifield, follow these steps:

1) Open the code `test.py` and set the value of the different parameters in the INPUT section.
2) Run the code: `python test.py`.
3) The code will generate a file with true value of the parameters, together with the mean and standard deviation of the posterior for each parameter.

This colab shows an example on how to train and test a model using these scripts.

8.2.4 databases

We use the optuna software to find the best value of the hyperparameters (learning rate, weight decay...etc) of one particular model. Typically, for each field, we perform 50 trials, where a trial corresponds to the training carried out with one particular choice of the value of the hyperparameters. This folder contains the databases created by optuna with the information about the different trials for the different simulation types.

The generic name of one of these files is `sim_o3_field_all_steps_500_500_o3.db` where `sim` can be IllustrisTNG, SIMBA, or Nbody, `field` is the considered field (can be several fields together). In some cases we smooth out the fields with a Gaussian kernel with width `width` (in pixel units). The generic name of these files is `sim_o3_field_all_steps_500_500_o3_smoothing_width.db`.

These files can be read with optuna package, and two arguments are needed:

- `study_name`. For all our databases, this variable is set to `wd_dr_hidden_lr_o3`.
- `storage`. This should be set to the a sqlite database with the name of the file, e.g. `sqlite:////home/fvillaescusa/CMC/2D_maps/benchmark/databases/SIMBA_o3_HI_all_steps_500_500_o3.db`

We provide an example on how to read the information of these files in this colab.

Note: The databases files are very light (typically less than 1 Mb each), so they all can be downloaded easily. The files containing the network weights are however bigger (~100 Mb), so it may be a good idea to only download the weights for the best model or the top 5 models.

8.2.5 weights

This folder contains the weights of all the models trained. Thus, for each field, there will be at least 50 different files containing the weights of the 50 different trials considered. The generic name of these files is `weights_sim_field_trialnumber_all_steps_500_500_o3.pt`, where `sim` can be IllustrisTNG, SIMBA, or Nbody, `field` is the name of the considered field (can be several of them), and `trialnumber` is the trial number. In the cases where the neural network is trained on fields that are smoothed with a Gaussian kernel, the generic name of the files is `weights_sim_field_trialnumber_all_steps_500_500_o3_smoothing_width.pt`, where width is the width of the Gaussian kernel in pixel units.

These files can be read by Pytorch routines once the architecture is specified. In all the cases, the architecture employed is `o3_err` (see `architecture.py` from the released codes). We provide an example on how to read these files in this colab.

8.3 Challenges

The results obtained with the above model were very promising. The networks were able to determine the value of the cosmological parameters with a few percent accuracy. However, some problems raised up. The most important was that for some fields, the model was not robust. For instance, if the model was trained on 2D maps from the IllustrisTNG simulations, it worked ery well when tested on maps from those simulations, but it failed when tested on 2D maps from the SIMBA simulations.

We think the four main challenges for the parameter inference task are:

- Build models to infer the value of the cosmological parameters that are more accurate than the above benchmark.
- Build models that trained on one suite of simulations (e.g. IllustrisTNG) can infer the correct cosmology when tested on the other suite (e.g. SIMBA).
- Determine the minimum set of fields than contain 90% and 95% of the cosmological and astrophysical information.
- Understand what operation(s) is the network performing for each different field and in multifield in order to determine the value of the cosmological parameters.

Solving the above challenges will help cosmologists to extract the maximum robust information from cosmological surveys, unveiling the laws and constituents of our own Universe.

CHAPTER

NINE

EMULATORS

Since each 2D map and 3D grid is characterized by a set of 6 parameters, one natural application of CMD is to build emulators, i.e.

$$\mathbf{X} = f(\vec{\theta})$$

where $\mathbf{X}$ can be a summary statistics (e.g. the probability distribution function of the pixels in a 2D map), or can be the entire 2D map or 3D grid. $\vec{\theta}$ is a vector with the value of the parameters (e.g. $\vec{\theta} = \{\Omega_{\rm m}, \sigma_8, A_{\rm SN1}, A_{\rm SN2}, A_{\rm AGN1}, A_{\rm AGN2}\}$) and f is the function relating the parameters with the data.

The idea is to use neural networks, or other methods, to approximate f. CMD provides a significant amount of data to achieve this task.

CHAPTER

TEN

N-BODY TO HYDRO

For a fixed volume, mass, and spatial resolution, running (magneto-)hydrodynamic simulations is much more computationally expensive than running a gravity-only N-body simulations.

This is a significant limitation and the reason why even today, we cannot run hydrodynamic simulations over scales of billions of ligh years, the ones that would be needed to analyze data from cosmological surveys. On the other hand, N-body simulations can be run over such large volumes with enough resolution.

One possible way around this is to learn how to paint gas and stars into the dark matter field simulated by N-body simulations. This way, we could simulate the result of a full (magneto-)hydrodynamic simulation from the output of a computationally cheap N-body simulation.

Note: This mapping can be done for a fixed cosmological and astrophysical model or can be done as a function of cosmology and astrophysics using CMD labels.

CHAPTER

ELEVEN

SUPERRESOLUTION

Cosmological simulations can be run at different mass and spatial resolutions. For a fixed volume, the higher the mass and spatial resolution the more computationally expensive the simulation will be. This is a major bottleneck for current numerical simulations. One possible way around this is to run a low resolution simulation and increase its resolution a-posteriori using some sophisticated method.

CMD provides for each field and redshift three different 3D grids of the same spatial distribution for N-body simulations but also for (magneto-)hydrodynamic simulations. That data can be used to train and test superresolution methods that can tackle this important challenge.

Warning: We note that the three different sets of grids available in CMD have different spatial resolution but the simulations used to create them have the same mass resolution. Previous works have changed both mass and spatial resolution in data from N-body simulations: 2001.05519, 2010.06608, 2105.01016.

CHAPTER

TWELVE

TIME EVOLUTION

Cosmological simulations typically start at high redshift (early cosmic times) and finish at redshift z=0 (present time). Several snapshots at different cosmic times are stored before the simulation finishes, and are used to post-process the data and to analyze the result of the simulation. In general, the more snapshots that can be saved the better. On the other hand, each snapshot occupies disk space, so in practice a limited number of snapshots can be saved.

Similarly to the emulation task, it would be desirable to develop a method that takes as input a set of snapshots at some cosmic times and returns new snapshots at different times. This way, the main limitation associated to this problem will be fixed as snapshots can easily be produced a-posteriori.

CMD provides a rich dataset with plenty of data at different cosmic times for N-body and state-of-the-art (magneto-)hydrodynamic simulations.

Note: 2012.05472 carried out this task using N-body simulations.

CHAPTER

THIRTEEN

TERMINOLOGY

Here we describe briefly some concepts that non-cosmologists may not be familiar with.

Dark matter. All matter we know (atoms, molecules, quarks…etc) only represent about 15% of the matter present in the Universe. The other 85% is called dark matter, and it is believed to only be subject to, and interact with other types of matter through, the force of gravity but not other forces. The effects of dark matter has been observed in many different systems, but the nature and properties of it is still a mystery.

Total matter. We define total matter as the sum of mass in dark matter, gas, stars, and black holes.

Dark energy. The Universe is currently accelerating its expansion. The substance responsible for that behaviour is called dark energy, and its nature and properties are one of the biggest mysteries in modern physics. Furthermore, dark energy represents around 70% of the full mass-energy content of the Universe. Learning more about dark energy is one of the main goals of modern cosmology.

Cosmological parameters. Parameters describing fundamental properties of the Universe such as its age, geometry, composition…etc. In CMD, we only consider two of them: $\Omega_{\rm m}$ and σ_8.

$\Omega_{\rm m}$: The fraction of the mass-energy density in the Universe in the form of total matter. Higher values of this parameter indicate that the fraction of dark matter mass plus gas mass plus stars mass plus black holes mass is higher. Since we are considering the Universe to be flat, a higher value of $\Omega_{\rm m}$ will imply a lower fraction of the mass-energy content of the Universe in the form of dark energy.

σ_8: The variance of the total matter density field, or the amplitude of its fluctuations. This parameter quantifies how clustered the considered field is. For instance, a very homogeneous and isotropic field will have a low value of σ_8, while if the fields exhibit large variations (e.g. some regions with very high and others with very low concentration of matter), this value will be larger.

Astrophysical parameters. Parameters describing astrophysical effects such as supernova and active galactic nucleus (AGN) feedback.

Supernova feedback. This refers to the energy and momentum released by supernova explosions to their surroundings.

AGN feedback. This refers to the energy and momentum released by active galactic nuclei (powered by supermassive black holes) to their surroundings.

Astrophysical effects. Also called baryonic effects, they refer to astrophysical processes such as supernova explosions or feedback from black holes. The physics of these physical processes is poorly known, but it is known that they can affect the properties of gas, dark matter, stars, and black holes on small scales.

Megaparsec. 1 megaparsec correspond to 1 million parsec. 1 parsec is 3.26 light years, i.e. the distance traveled by light over 3.26 years: 30.9 trillion kilometers or 19.2 trillion miles.

Redshift. In cosmology, redshift is commonly used as a measure of time. Redshift 0 corresponds to current time, while higher redshifts correspond to more distant times in the past. For instance, redshift 1 corresponds to around 8 billion years ago, or the epoch when the Universe was approximately half as old as it is today.

Subgrid physics. Cosmological hydrodynamic simulations sample very large cosmological distances of tens or hundreds of millions of light years. Unfortunately, astrophysical processes such as the formation of stars and black holes, supernova explosions, feedback from black holes...etc happen on much smaller scales that cannot be resolved in these simulations due to the very large dynamic range. In this case, these astrophysical processes are modelled in a phenomenological manner aimed at mimic the physics of these processes; this is called subgrid physics.

IllustrisTNG. This denotes data from cosmological magneto-hydrodynamic simulations run with the AREPO code (a code used to solve the gravity plus magneto-hydrodynamics equations), employing the same subgrid physics as the IllustrisTNG subgrid model, up to variations of the astrophysical parameters.

SIMBA. This denotes data from cosmological hydrodynamic simulations run with the GIZMO code (a code used to solve the gravity plus hydrodynamics equations), employing the same subgrid physics as the SIMBA subgrid model, up to variations of the astrophysical parameters.

Astrid. This denotes data from cosmological hydrodynamic simulations run with the MG-Gadget code (a code used to solve the gravity plus hydrodynamics equations), employing the same subgrid physics as the Astrid subgrid model, up to variations of the astrophysical parameters.

Metallicity. In astronomy, all elements other than hydrogen and helium are called metals. The metallicity of a gas cloud, a galaxy or a star is the fraction of mass in metals in that system.

Magnesium over iron. Cosmic gas, stars and galaxies contain a fraction of their total mass in different elements, such as carbon, oxigen, magnesium, iron...etc. The ratio between the magnesium and iron, Mg/Fe, is an interesting quantity from the point of view of astrophysics. In CMD, we use it to see if we can extract cosmological information from it.

CHAPTER

FOURTEEN

LICENSE

CHAPTER

FIFTEEN

HELP

For problems, questions and general help you can reach us at fvillaescusa@flatironinstitute.org or camel.simulations@gmail.com.

We would also love to hear if you would like we add more data to CMD, e.g. more fields, more redshifts, more simulations…etc.