BG image credit: TNG project

Mauro Rigo

Roberto Trotta, Matteo Viel

# JERALD: high resolution dark matter and baryonic maps from cheap N-body simulations

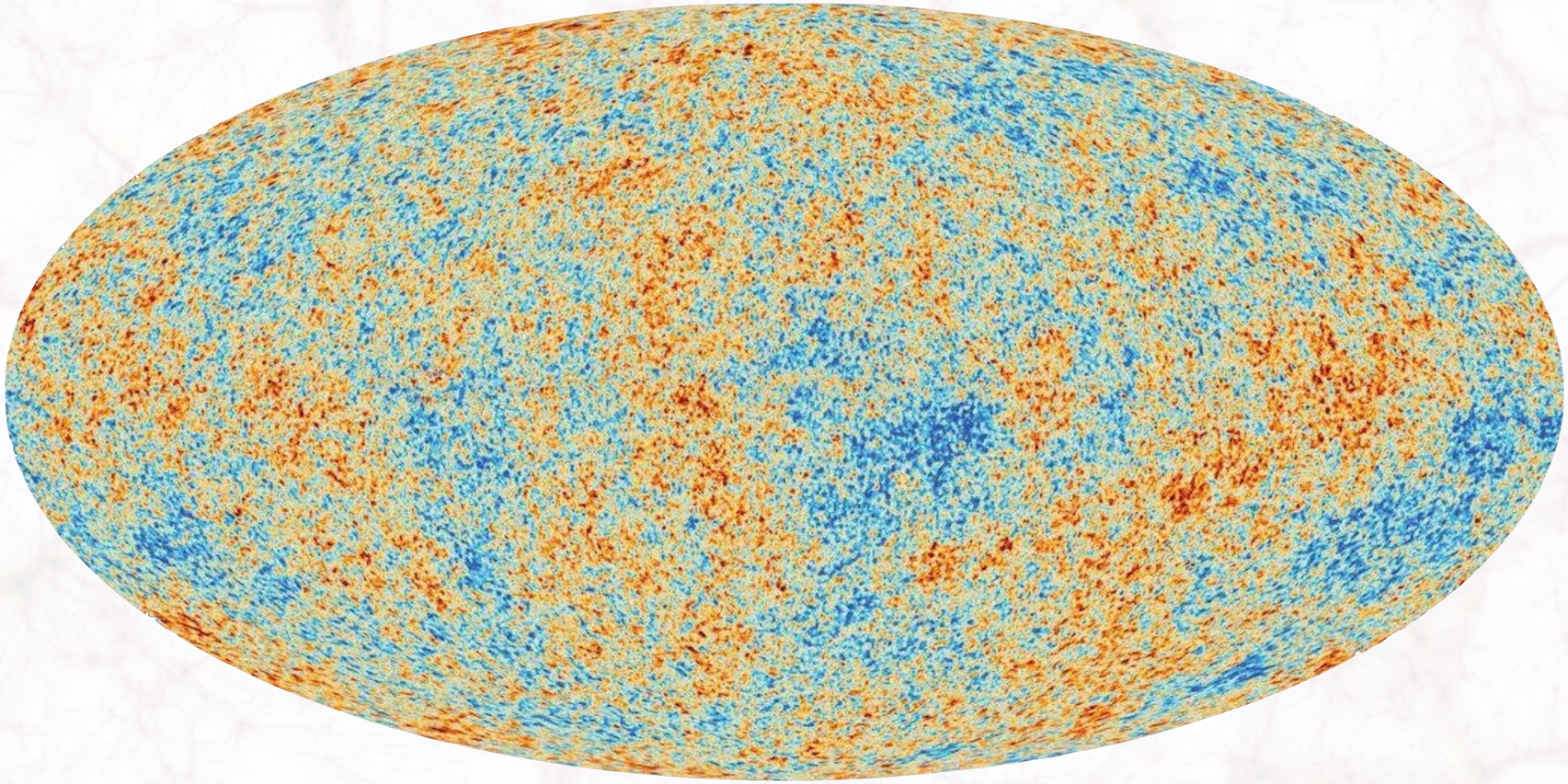
EUCAIFCON AMSTERDAM 01/05/2024

# Contents

- Studying the universe:
  - The problem of full hydrodynamical simulations;
- The Lagrangian Deep Learning (LDL) model:
  - What it is and how it works;
  - Dark matter and stellar mass maps;
- Neutral hydrogen and JERALD:
  - Challenges of neutral hydrogen;
  - JERALD modifications and improvements;
  - Results;
- Remarks and current/upcoming work;

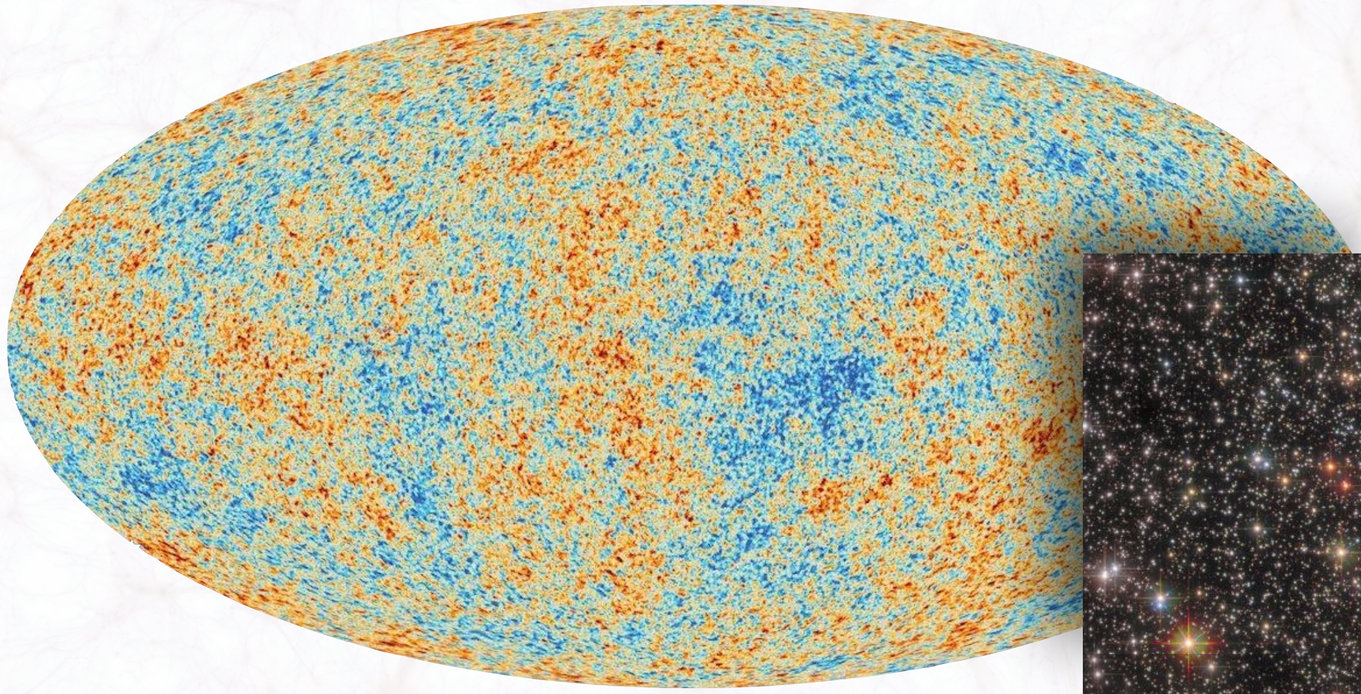
# Studying the universe

# Studying the universe



Credit: ESA

# Studying the universe

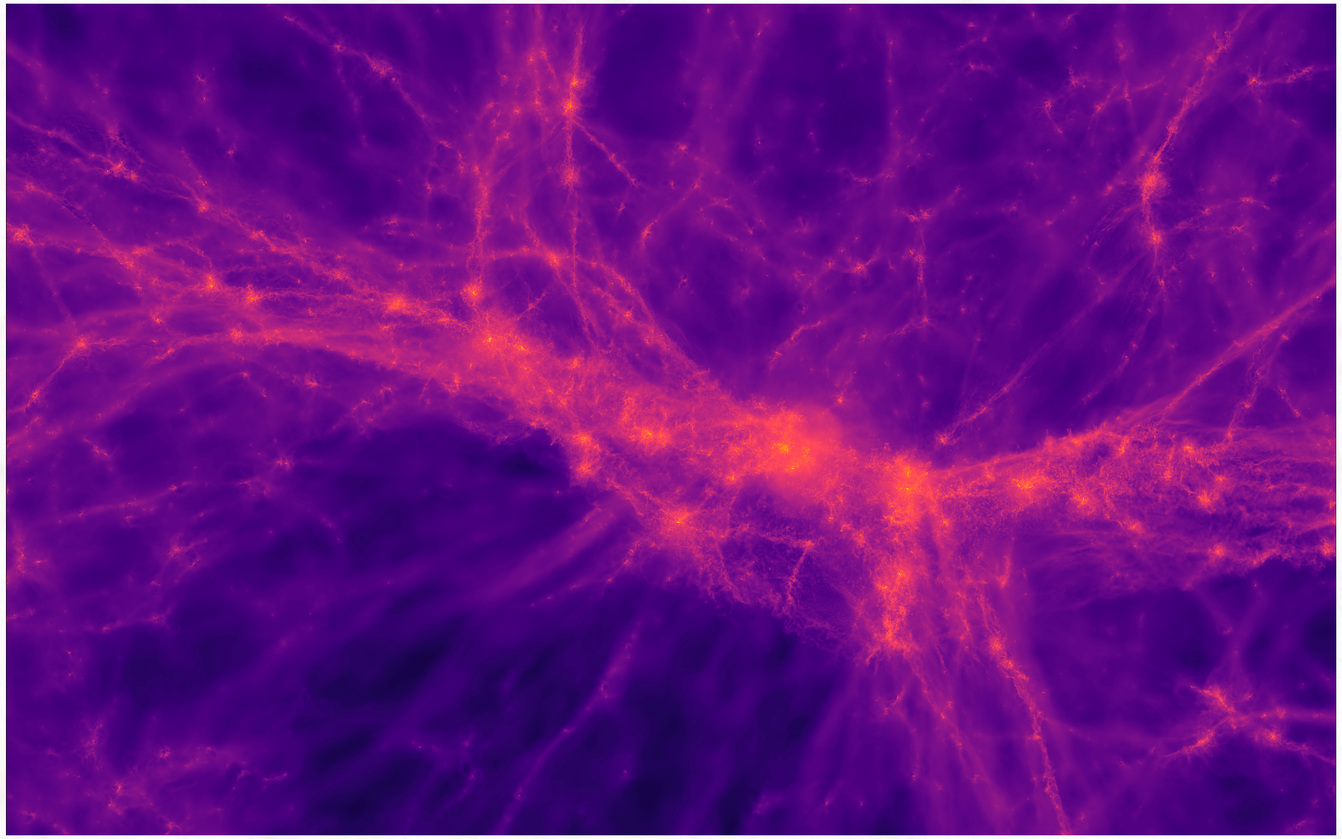


Credit: ESA

Credit: JWST

# Studying the universe

Full hydrodynamical simulations use millions of CPU core hours and take up to months to run.

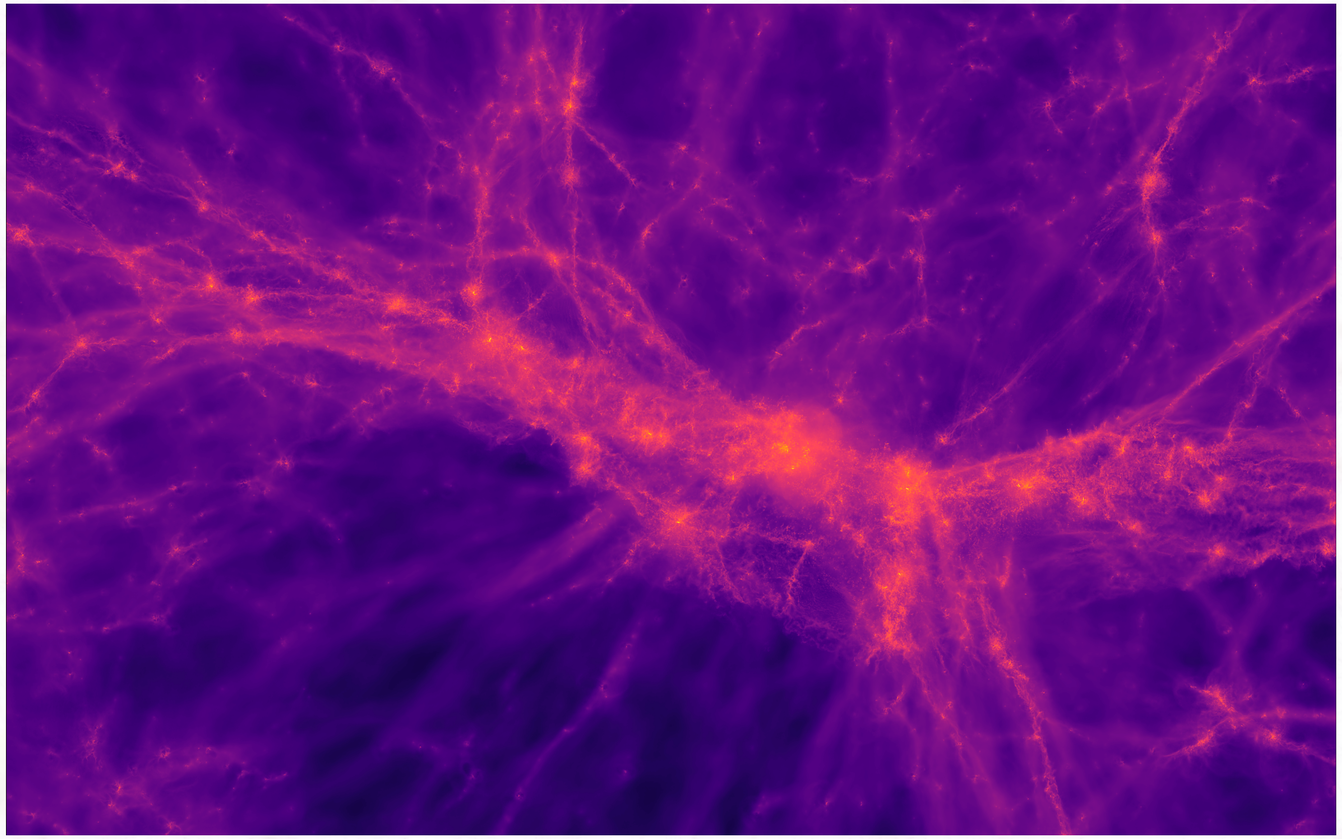


Credit: TNG project

# Studying the universe

Full hydrodynamical simulations use millions of CPU core hours and take up to months to run.

To alleviate this cost, one could run approximate N-body (DM only) simulations and populate them with baryons with ML techniques.



Credit: TNG project

# The Lagrangian Deep Learning model

# The Lagrangian Deep Learning model

Dark matter particles from approximate N-body simulations are moved via a displacement field $\boldsymbol{S}_\theta$:

$$\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{S}_\theta \quad , \qquad \boldsymbol{S}_\theta = \nabla U_\theta(\delta(\boldsymbol{X}))$$

Dai and Seljak, 2021

# The Lagrangian Deep Learning model

Dark matter particles from approximate N-body simulations are moved via a displacement field $\boldsymbol{S}_\theta$:

$$\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{S}_\theta \quad , \qquad \boldsymbol{S}_\theta = \nabla U_\theta(\delta(\boldsymbol{X}))$$

The parameters $\theta$ are trained by matching the output density $\delta_{\text{model}}(\boldsymbol{X}')$ with a target map $\delta_{\text{target}}$ through a L1 or L2 loss:

$$\mathscr{L} = \frac{1}{N_{\text{pixel}}} \| \delta_{\text{model}} - \delta_{\text{target}} \|^p$$

Dai and Seljak, 2021

# The Lagrangian Deep Learning model

Dark matter particles from approximate N-body simulations are moved via a displacement field $\boldsymbol{S}_\theta$:

$$\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{S}_\theta \quad , \qquad \boldsymbol{S}_\theta = \nabla U_\theta(\delta(\boldsymbol{X}))$$

The parameters $\theta$ are trained by matching the output density $\delta_{\text{model}}(\boldsymbol{X}')$ with a target map $\delta_{\text{target}}$ through a L1 or L2 loss:

$$\mathscr{L} = \frac{1}{N_{\text{pixel}}} \| \delta_{\text{model}} - \delta_{\text{target}} \|^p$$

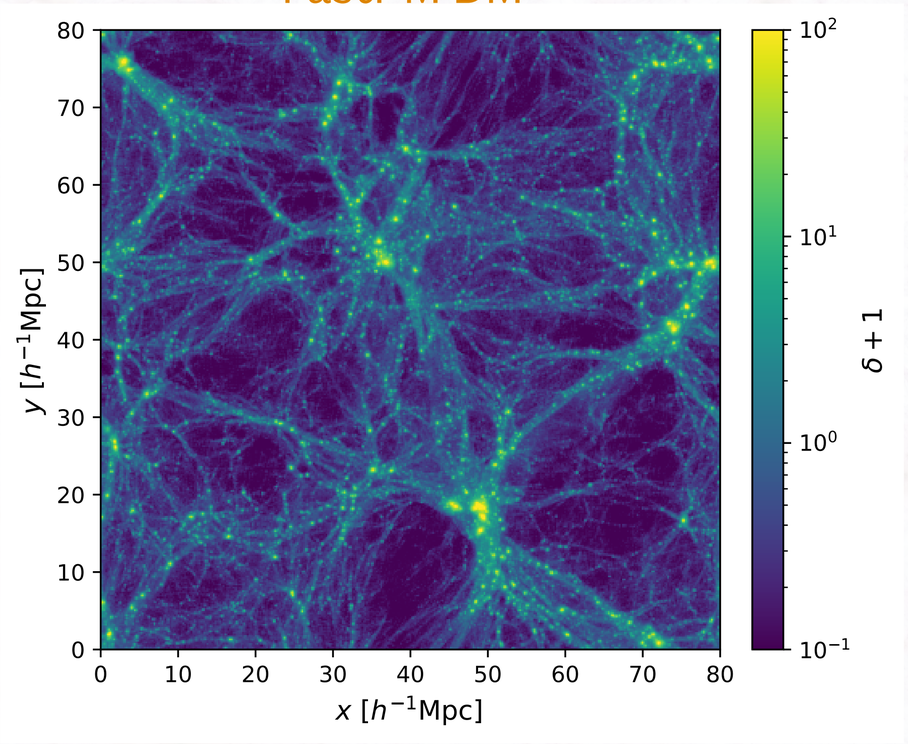
Optionally, a non-linearity can be added after the displacements to model phenomena that cannot be modelled via matter transport:

$$\delta_{\text{model}} = \text{ReLU}(b_1 \, \delta(\boldsymbol{X}')^\mu + b_0)$$

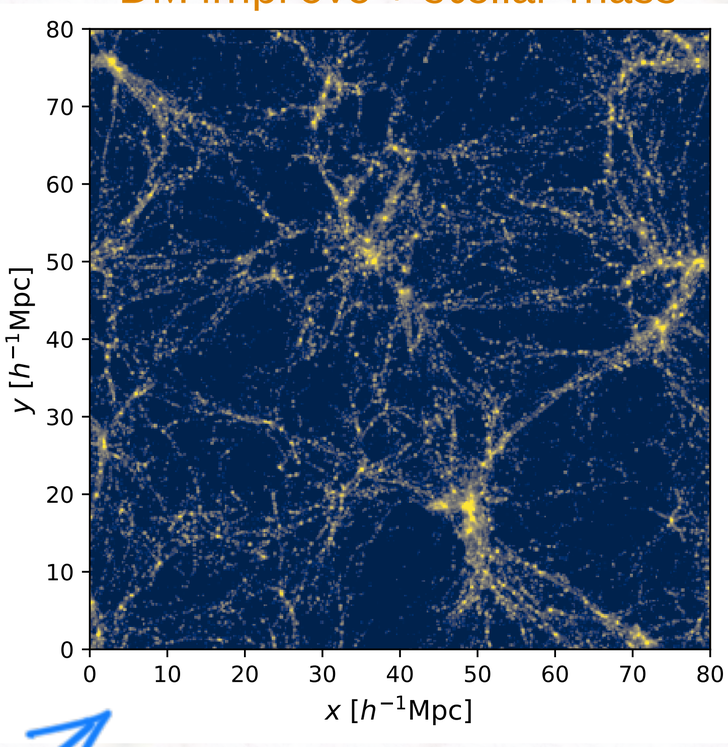
Dai and Seljak, 2021

# DM and stellar mass maps

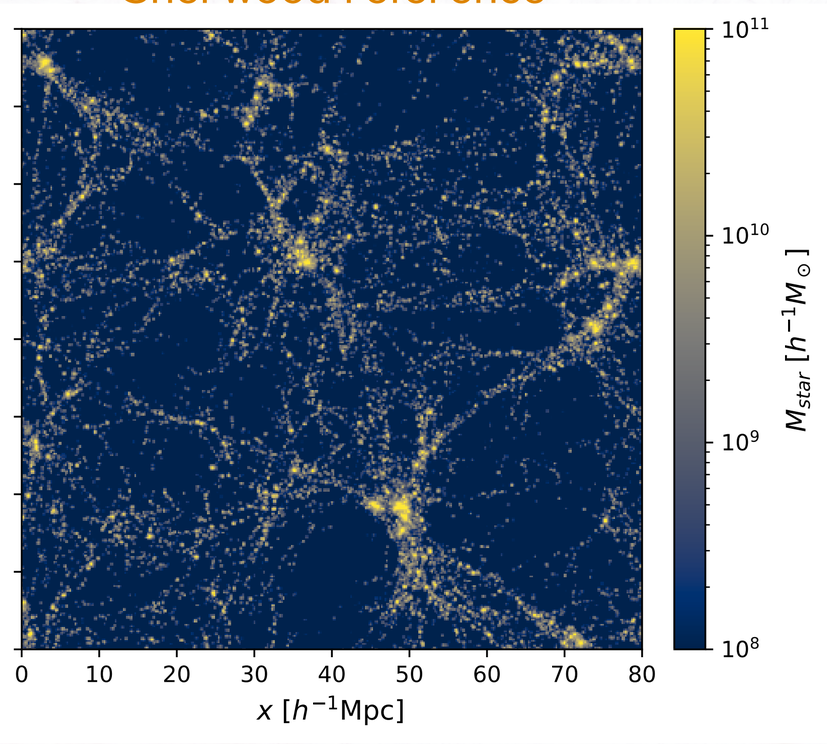
FastPM DM




DM improve + stellar mass

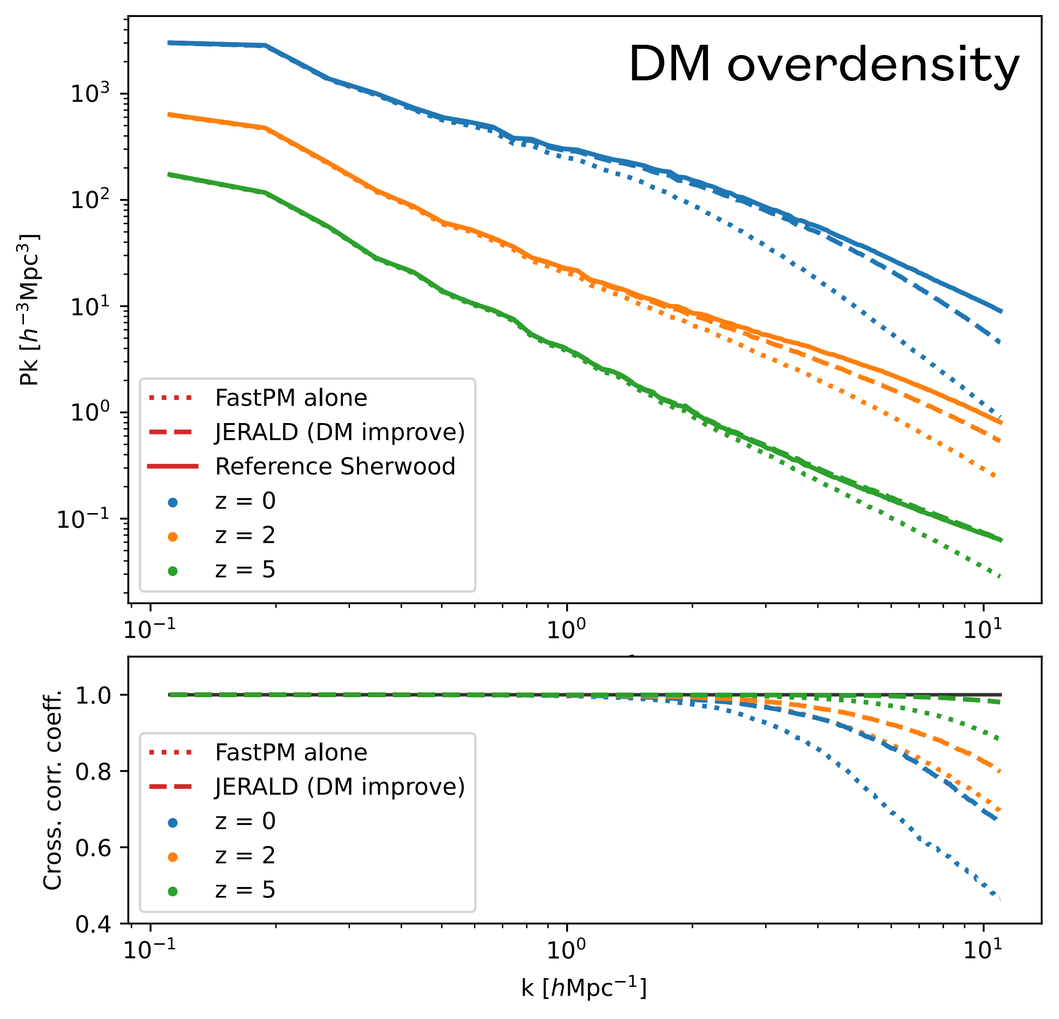


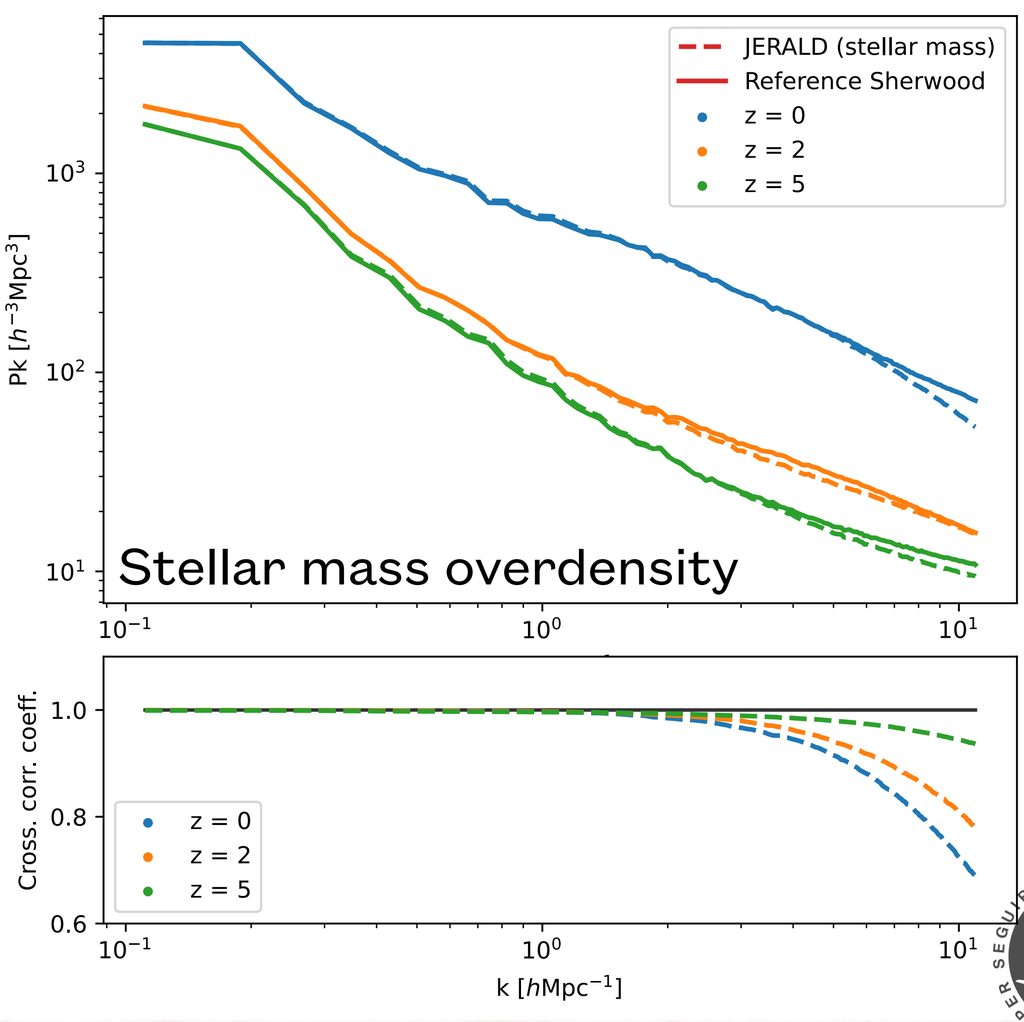

Sherwood reference



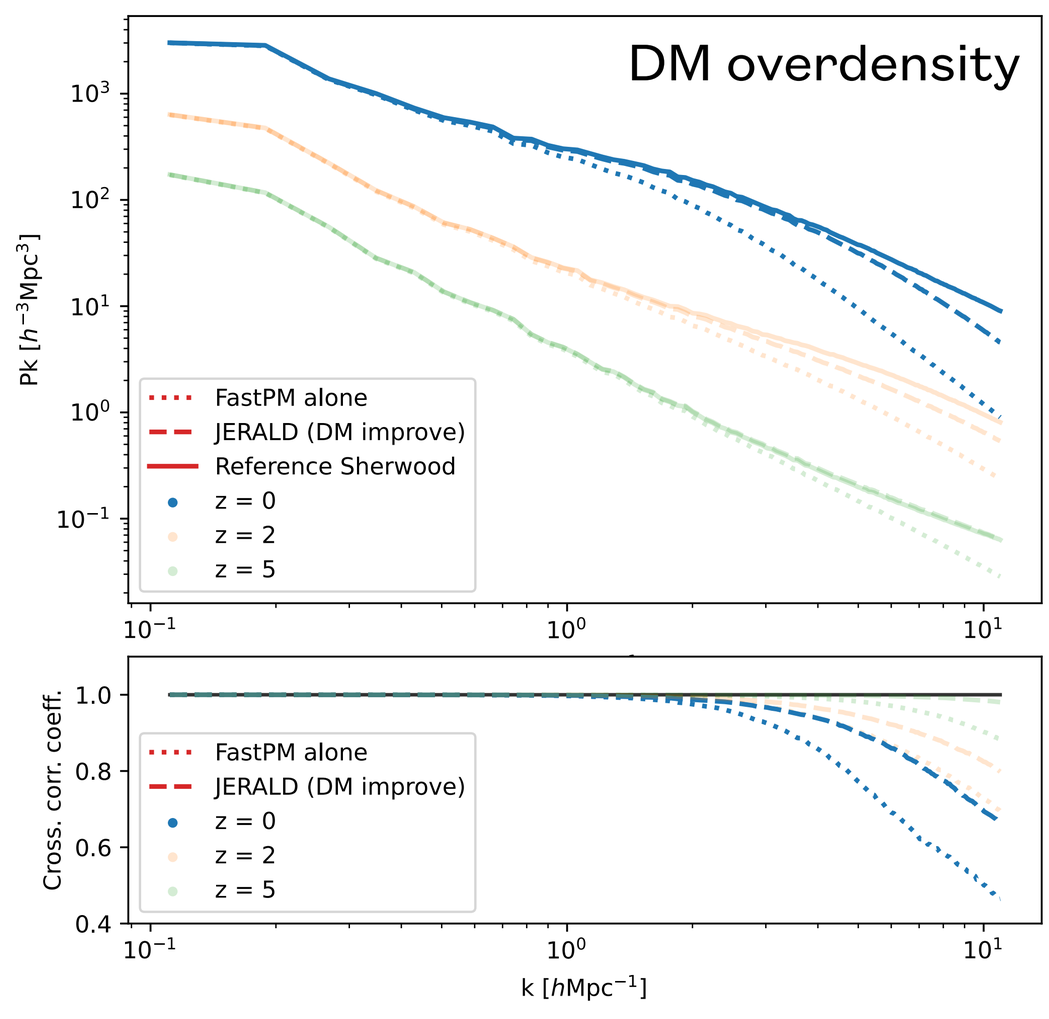

JERALD

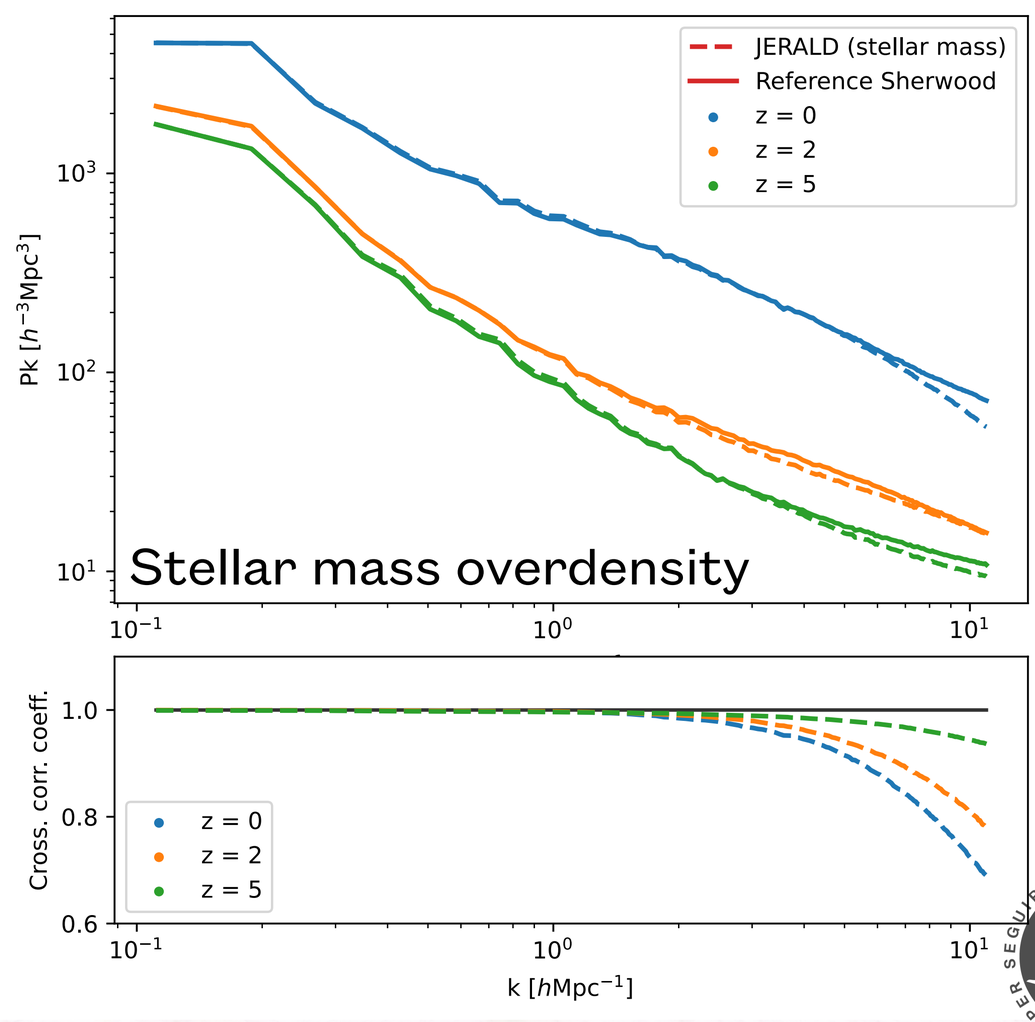
# DM and stellar mass maps



# DM and stellar mass maps



# Neutral hydrogen

# JERALD and low/mid density nHI

JERALD is a new code is partly written in Python with JAX and partly in C++ and parallelized using MPI. At present it is limited to CPUs.

It models the effect of stars in two different ways, by using a pretrained stellar mass map:

$$\boldsymbol{X}' = \boldsymbol{X} + \nabla U_\theta(\delta(\boldsymbol{X}))$$

$$\delta_{\text{model}} = \text{ReLU}(b_1\, \delta(\boldsymbol{X}')^\mu + b_0)$$

# JERALD and low/mid density nHI

JERALD is a new code is partly written in Python with JAX and partly in C++ and parallelized using MPI. At present it is limited to CPUs.

It models the effect of stars in two different ways, by using a pretrained stellar mass map:

$$X' = X + \nabla U_\theta(\delta(X)) + \nabla U'_{\theta'}(\delta_{star})$$

$$\delta_{\text{model}} = \text{ReLU}(b_1\, \delta(X')^\mu + b_0)$$

# JERALD and low/mid density nHI

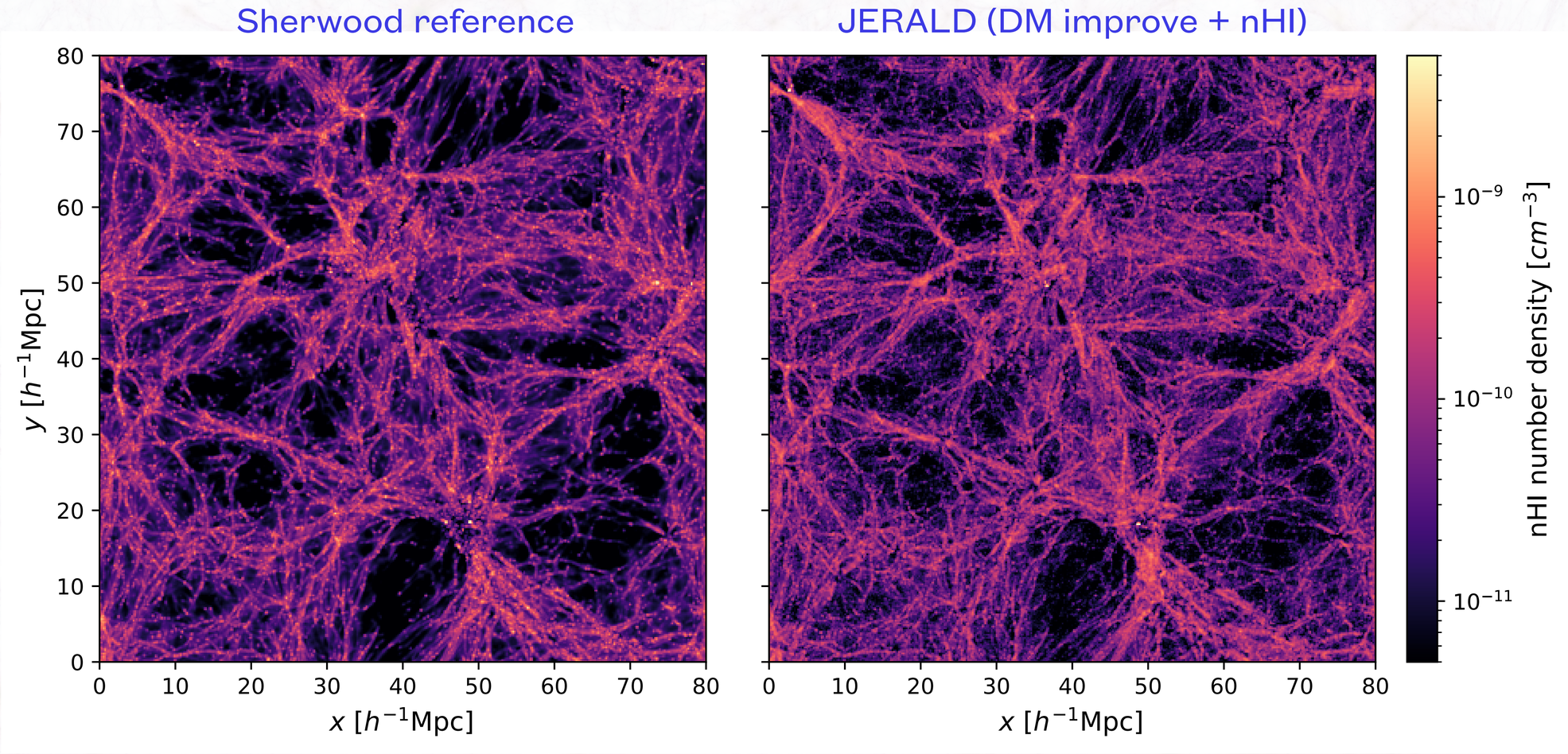
JERALD is a new code is partly written in Python with JAX and partly in C++ and parallelized using MPI. At present it is limited to CPUs.

It models the effect of stars in two different ways, by using a pretrained stellar mass map:

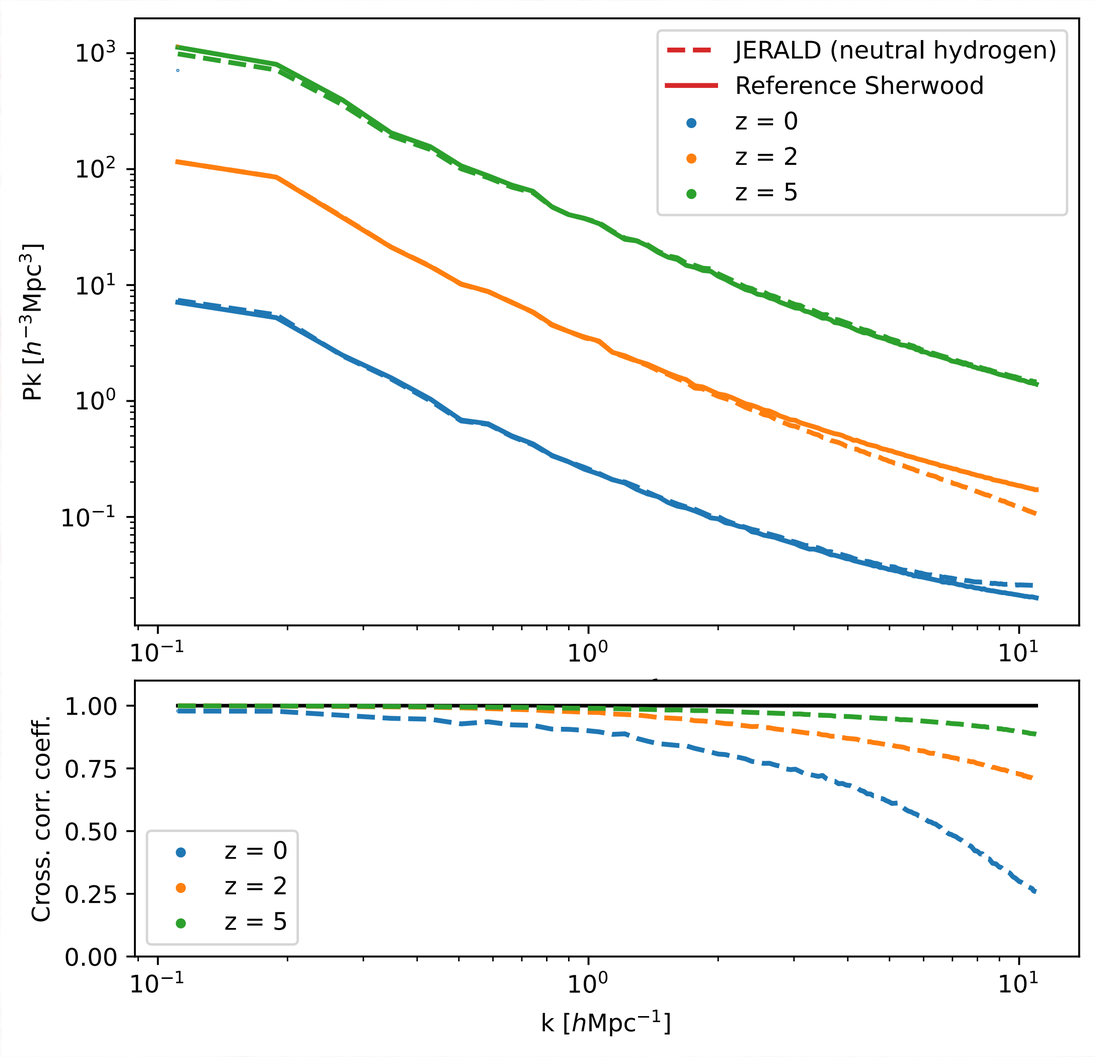
$$X' = X + \nabla U_\theta(\delta(X)) + \nabla U'_{\theta'}(\delta_{star})$$

$$\delta_{\text{model}} = \text{ReLU}(b_1 \text{ReLU}(\delta(X') + F_{\theta''}(\delta_{star}) + b_2)^\mu + b_0)$$

# JERALD and low/mid density nHI



# JERALD and low/mid density nHI



# Remarks and current/upcoming work

- The model can produce maps with power spectra in good agreement with the references, independently of the code that generated them and the redshift.
- Currently investigating its generalization under variation of cosmological parameters and DM models.
- Regarding the Ly-$\alpha$ forest, currently focusing on LOS data, possibly integrating it in the pipeline.
- Currently exploring the performance of the code also for high nHI density regions.
- Use the approach to build mock catalogs.

SISSA DATASCIENCE
Machine Learning for the Natural Sciences

# Thank you for your attention

Email me: mrigo@sissa.it

JERALD


Supervisors:

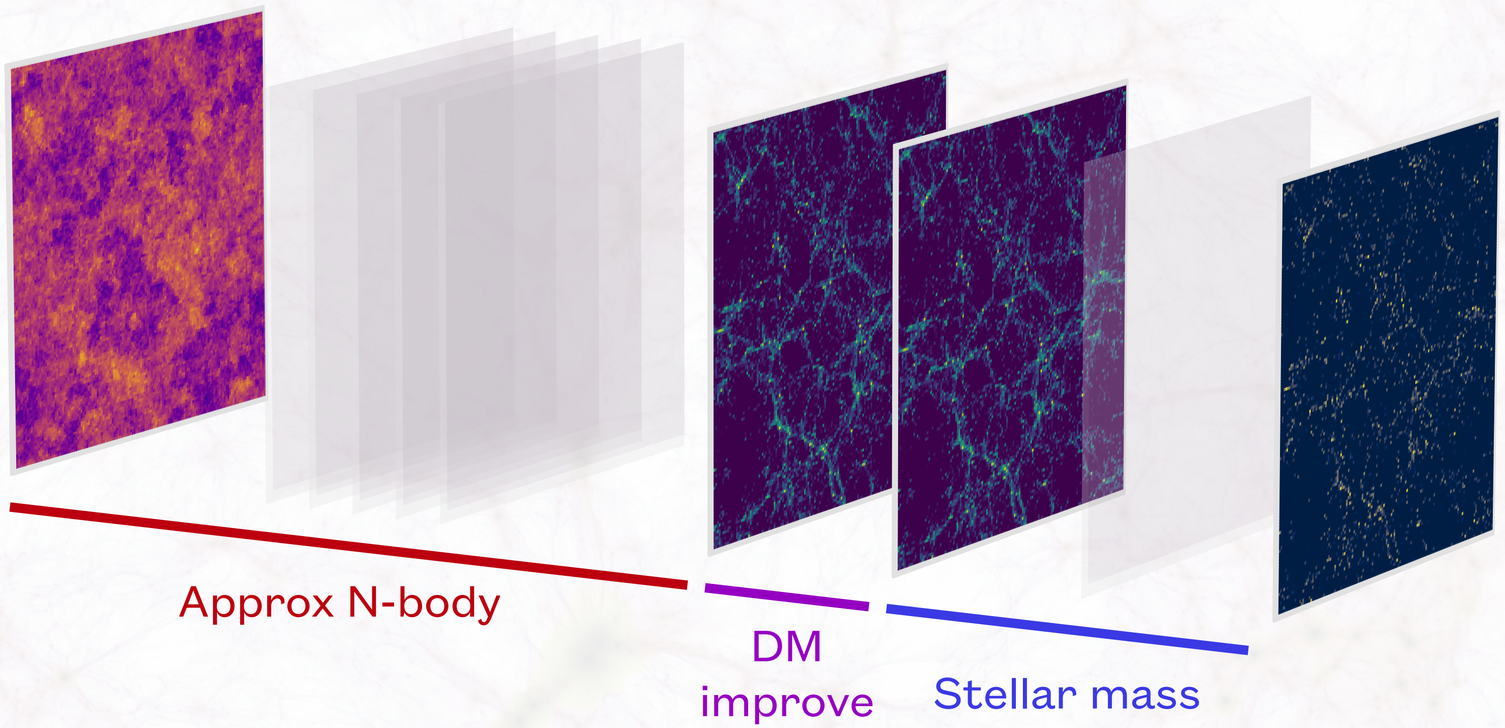
Roberto Trotta

Matteo Viel

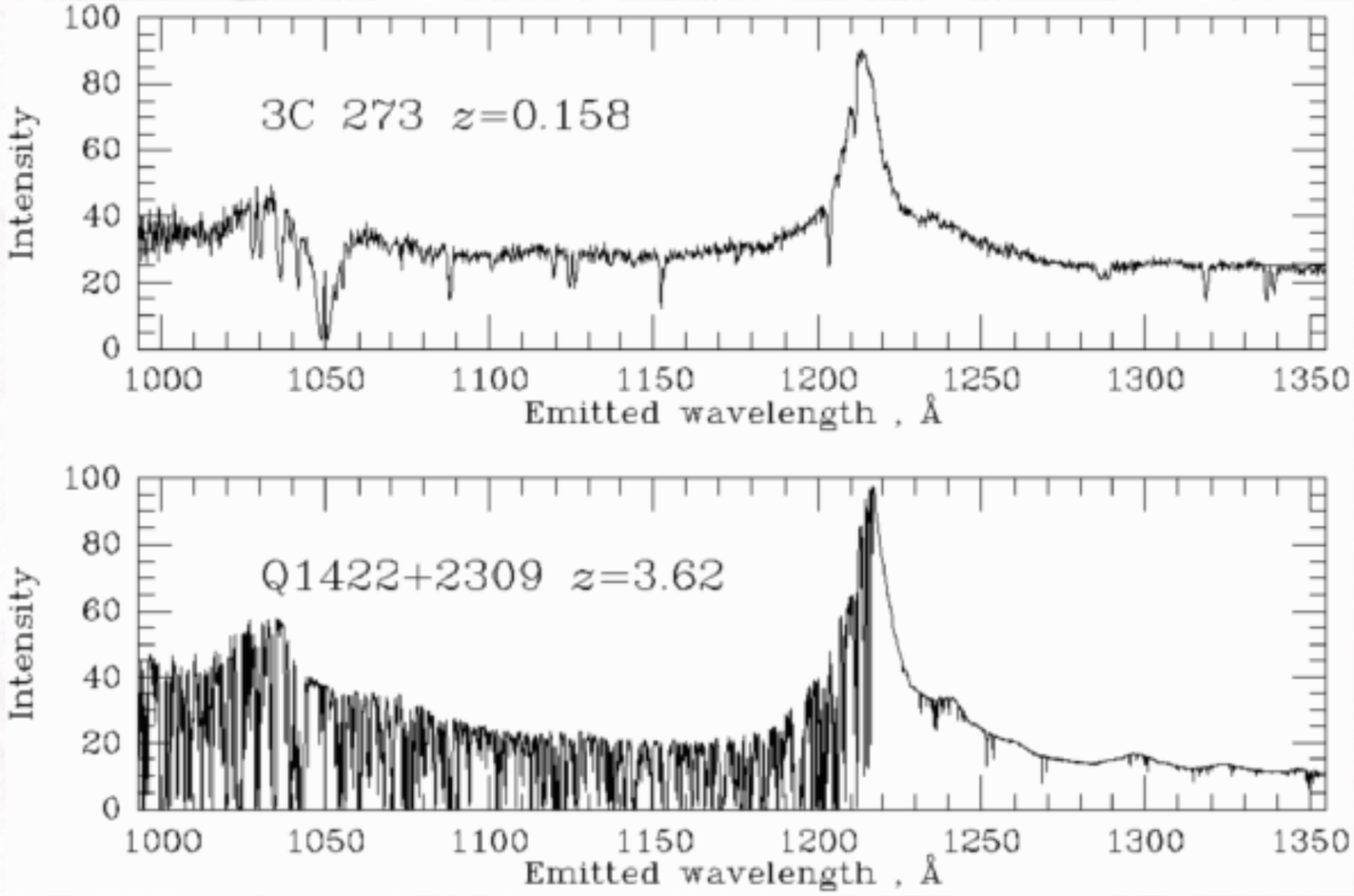
# The LDL potential

$$U_\theta(\delta) = \hat{O}_G\, f(\delta) \quad , \qquad f(\delta) = \alpha(1+\delta)^{\gamma} \quad , \qquad \hat{O}_G = \exp\left[-\left(\frac{k_h}{k}\right)^2\right] \exp\left[-\left(\frac{k}{k_l}\right)^2\right] k^n$$

# The LDL pipeline



# Neutral hydrogen: the Ly-$\alpha$ forest




Credit: https://astro.ucla.edu/~wright/Lyman-alpha-forest.html