Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **541,** 166–178 (2025)
Advance Access publication 2025 June 12

https://doi.org/10.1093/mnras/staf948

JERALD: high-fidelity dark matter, stellar mass, and neutral hydrogen maps from fast *N*-body simulations

Mauro Rigo,[1] Roberto Trotta[1,2,3,4★] and Matteo Viel[1,2,3,5,6]

[1]*SISSA, Theoretical and Scientific Data Science, Via Bonomea 265, I-34136 Trieste, Italy*
[2]*INFN – National Institute for Nuclear Physics, Via Valerio 2, I-34127 Trieste, Italy*
[3]*ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data e Quantum Computing, Via Magnanelli 2, 40033 Bologna, Italy*
[4]*Astrophysics Group, Physics Department, Blackett Lab, Imperial College London, Prince Consort Road, London SW7 2AZ, UK*
[5]*INAF – Osservatorio Astronomico di Trieste, Via G. B. Tiepolo 11, I-34143 Trieste, Italy*
[6]*IFPU – Institute for Fundamental Physics of the Universe, Via Beirut 2, I-34151 Trieste, Italy*

Accepted 2025 May 29. Received 2025 May 8; in original form 2025 January 17

ABSTRACT

We present a new code and approach, JERALD – JAX Enhanced Resolution Approximate Lagrangian Dynamics –, that improves on and extends the Lagrangian Deep Learning method of Dai & Seljak (2021), producing high-resolution dark matter, stellar mass and neutral hydrogen maps from lower-resolution approximate N-body simulations. The model is trained using the Sherwood-Relics simulation suite (for a fixed cosmology), specifically designed for the intergalactic medium and the neutral hydrogen distribution in the cosmic web. The output is tested in the redshift range from $z = 5$ to $z = 0$ and the generalization properties of the learned mapping is demonstrated. JERALD produces maps with dark matter, stellar and neutral hydrogen power spectra in excellent agreement with full-hydrodynamical simulations with 8× higher resolution, at large and intermediate scales; in particular, JERALD's neutral hydrogen power spectra agree with their higher-resolution full-hydrodynamical counterparts within 90 per cent up to $k \simeq 1\,h\,\mathrm{Mpc}^{-1}$ and within 70 per cent up to $k \simeq 10\,h\,\mathrm{Mpc}^{-1}$. JERALD provides a fast, accurate, and physically motivated approach that we plan to embed in a statistical inference pipeline, such as Simulation-Based Inference, to constrain dark matter properties from large- to intermediate-scale structure observables.

Key words: methods: numerical – software: machine learning – galaxies: formation – (galaxies:) intergalactic medium – (cosmology:) large-scale structure of Universe.

1 INTRODUCTION

Cosmological simulations are essential tools for studying the universe, as modelling physical observables is necessary to extract the maximum amount of information from current and future sky surveys such as DESI (Dark Energy Spectroscopic Instrument, DESI Collaboration 2016), *Euclid* (Amendola 2018), SKA (Square Kilometre Array, SWG 2020), and LSST (Large Synoptic Survey Telescope, Ivezić et al. 2019). These surveys will probe vast volumes of the sky over a broad frequency spectrum, gathering data for a great variety of phenomena and observables such as galaxy clustering, weak lensing, supernovae type Ia, 21-cm radiation, and the Ly α forest with unprecedented resolution. As an example, *Euclid* will cover more than a third of the sky in its 6-yr mission, with angular resolutions up to few tenths of arcseconds, while LSST, in 10 yr, will study over 40 per cent of the sky with similar resolution. Such impressive numbers will allow to investigate the universe over very small scales, which offer a multitude of avenues for scientific research. For example, small-scale (putative) problems yet to be solved such as the core/cusp problem (Flores & Primack 1994; Moore 1994) and the missing satellite problem (Klypin et al. 1999; Moore et al. 1999) challenge our current understanding of the process of structure formation, and while solutions may exist within the Lambda cold dark matter (ΛCDM) paradigm, these inconsistencies constitute strong drivers for research beyond the standard cosmological model (Bullock & Boylan-Kolchin 2017).

Yet, the exciting opportunities that these surveys offer pose a major problem: on scales $\lesssim 1\,h^{-1}\,\mathrm{Mpc}$ and particularly at late times, perturbations become highly non-inear, thus requiring high-resolution simulations to be modelled accurately. The most advanced simulations, called full-hydrodynamical simulations owing to their treatment of baryons as fluids, are capable of tracking the evolution of several billions of dark matter (DM) and baryonic particles for several billions of years, modelling many of the physical phenomena that govern their dynamics on a wide range of scales (Bolton et al. 2016; Pillepich et al. 2017; Villaescusa-Navarro et al. 2021; Puchwein et al. 2022); none the less, they are computationally extremely expensive, often requiring millions of CPU hours (Bolton et al. 2016; Nelson et al. 2017). One of the major culprits for the enormous computational cost of numerical simulations is baryonic matter itself, since baryons are characterized by a vast range of physical processes that need to be implemented in order to accurately describe their interactions. On the contrary, DM is relatively inexpensive to simulate as it only interacts

---

★ E-mail: r.trotta@imperial.ac.uk

through gravity, which can either be solved approximately with a computational complexity of $\mathcal{O}(N \log N)$ for N particles (Tassev, Zaldarriaga & Eisenstein 2013; Feng et al. 2016) or even fully. Therefore, since gravity is the dominant component over large scales, where baryons trace the evolution of DM, one possible approach for mitigating the great computational cost of full-hydrodynamical simulations consists in running either full or approximate DM-only (N-body) simulations, and then populating them with baryons or extracting baryonic properties in a post-processing step.

Many different approaches have been proposed in this direction: semi-analytical methods (Davis et al. 1985; Berlind & Weinberg 2002; Springel et al. 2005; Baugh 2006; Croton et al. 2016; Behroozi et al. 2019), including abundance matching techniques (Behroozi, Conroy & Wechsler 2010; Moster et al. 2010; Moster, Naab & White 2012), are used to reproduce galaxy and halo properties, while in recent years machine learning (ML)-based methods have flourished, including e.g. support vector machines, $k-$nearest neighbour algorithms or tree-based methods (Xu et al. 2013; Kamdar, Turk & Brunner 2015, 2016; Agarwal, Davé & Bassett 2018). Other authors treat the problem of painting baryons on DM-only simulations as an image-to-image or an image generation task, developing pixel-based methods to map DM distributions to different baryonic observables using convolutional neural networks (Zhang et al. 2019; Bernardini et al. 2021; Chadayammuri et al. 2023; Wadekar et al. 2023), sometimes in conjunction with variational diffusion models (VDMs; Tanner 2024), or using variational auto-encoders (VAEs) and generative adversarial networks (Tröster et al. 2019) as well as score-based models (Bourdin et al. 2024).

In this work, we follow another approach, which is well-suited for problems with relatively simple large-scale physics but complex small-scale effects that can be approximately modelled using an effective theory framework, and it consists in rewriting the Lagrangian describing the system in a way that explicitly satisfies the symmetries of the problem, but parametrized by a set of free parameters meant to capture non-perturbative small-scale effects. Contrarily to pixel-based approaches – which treat DM and baryons distributions as images and therefore do not exploit intrinsic physical symmetries – this 'Lagrangian approach' works at the level of individual particles or fluid elements of simulations by modelling their displacement field through an effective potential.

In the context of cosmological simulations, this scheme has a straightforward application in improving approximate N-body simulations, where the Lagrangian framework is used to compensate for the loss of power of the DM distribution at small scales caused by the coarse-graining in time and space in these types of simulations (Dai et al. 2020). By writing the effective potential used to displace the particles in a suitable form, this technique preserves the translational and rotational symmetries, while simply adjusting the location of the particles explicitly satisfies mass conservation. In an approach they called *Lagrangian Deep Learning*[1] (LDL), Dai & Seljak (2021) propose to extend this idea to baryons and their properties. Since baryons trace the evolution of DM on large scales and the two distributions differ mostly on small scales, in the Lagrangian framework the distribution of DM can be thought of as a large-scale distribution of baryons, and the small-scale baryonic physics can instead be directly modelled at the desired redshift by displacing the DM particles, reinterpreted as baryons, such that their final density matches the one of a specific baryon type from some reference full-hydrodynamical simulation. Additionally, the authors extended the model to predict other baryonic observables such as thermal and kinetic Sunyaev–Zeldovich signals and X-ray emission.

Building on this idea, in this paper we present a new code and an improved approach, under the name of 'JAX Enhanced-Resolution Approximate Lagrangian Dynamics' or JERALD, which increases the computational efficiency of LDL and demonstrates the generalization properties of the method to other simulation suites, to higher redshift and to the modelling of neutral hydrogen (H I).

This paper is organized as follows. In Section 2, we introduce JERALD and the improvements it brings over LDL; we also explain how neutral hydrogen is modelled within our framework. In Section 3, we give details about the implementation and the performance of the code. In Section 4, we present our validation results, as well as generalization to different initial conditions. Concluding remarks and future perspectives are given in Section 5.

2 THE MODEL

In this section, we describe the JERALD model by detailing our new implementation of the original idea by Dai & Seljak (2021), and describing the pipeline of the approach. In Section 2.1, we first explain how the DM distributions from approximate N-body simulations are improved, also to account for baryonic feedback, and how stellar mass distributions are learned, while the case of neutral hydrogen is treated separately in Section 2.2, as it poses specific challenges that require tailored solutions.

2.1 Lagrangian approach and JERALD

This section closely follows Dai & Seljak (2021) except for small changes. Like LDL, JERALD takes as input DM particles positions, moves them via one or more layers of Lagrangian displacement (in this paper, we mostly use 2), and finally produces as output a density field via a mass-assignment scheme (Cloud-In-Cell or CIC), optionally applying a non-linear transformation to the output. Given a particle i with position $\boldsymbol{r}_i = (r_i^x, r_i^y, r_i^z)$ in comoving coordinates, a Lagrangian displacement layer acts as

$$\boldsymbol{r}_i \to \boldsymbol{r}_i + \boldsymbol{S}(\boldsymbol{r}_i)\,, \tag{1}$$

where the displacement field $\boldsymbol{S}(\boldsymbol{r})$ is modelled as the gradient of an effective potential, the source of which, similarly to gravity, is a simple power law of the normalized density:

$$\boldsymbol{S} = \alpha \nabla \left(\hat{G} f(\delta) \right) \quad , \qquad f(\delta) = (1+\delta)^{\gamma}\,, \tag{2}$$

where we omitted the dependence on $\boldsymbol{r}$ for ease of notation and

$$1 + \delta(\boldsymbol{r}) = \rho(\boldsymbol{r}) / \sum_i \rho(\boldsymbol{r}_i) \tag{3}$$

is the normalized density, computed assuming the same mass for each particle (which cancels out when normalizing by the average density). In equation (2), α and γ are free (learnable) parameters and $\hat{G}$ is the Green's operator, the action of which can be explicitly written as

$$\hat{G} f(\delta)(\boldsymbol{r}) = \int G(\boldsymbol{r}, \boldsymbol{r}') f(\delta(\boldsymbol{r}'))\mathrm{d}\boldsymbol{r}'\,, \tag{4}$$

with $G(\boldsymbol{r}, \boldsymbol{r}') = G(\boldsymbol{r} - \boldsymbol{r}')$ because of translational symmetry. Being a convolution, equation (4) has a straightforward expression in

---

[1] This name was motivated by the presence of multiple displacement layer in the original implementation. However, 'deep learning' as usually understood in a ML sense involves a large number of free learnable parameters, which is not the case here; this is why we prefer to refer to our implementation in terms of 'approximate Lagrangian dynamics'.

Fourier space as

$$\mathcal{F}\left[\hat{G}\,f(\delta)\right](\boldsymbol{k}) = \tilde{G}(\boldsymbol{k})\,\tilde{f}(\delta)(\boldsymbol{k})\,, \tag{5}$$

with $\tilde{G}(\boldsymbol{k}) = \tilde{G}(k)$ due to rotational symmetry.

The explicit form of $\tilde{G}(k)$ depends on the target quantity: when modelling stellar mass, the Green's operator is parametrized, as in LDL, in the following way:

$$\tilde{G}_{\rm sm}(k) = \exp\left[-\left(\frac{k^h_{\rm sm}}{k}\right)^2\right]\exp\left[-\left(\frac{k}{k^l_{\rm sm}}\right)^2\right]k^{\nu_{\rm sm}}\,, \tag{6}$$

where the first factor suppresses large-scale growth for $k < k^h_{\rm sm}$, since the physics to model through an effective description happens at small scales, while the second term mimics force softening. The final power law term is motivated by the fact that fixing $\alpha = 4\pi G\bar{\rho}$ and $\gamma = 1$ in equation (2), setting $k^h_{\rm sm} = 0$, $k^l_{\rm sm} = \infty$ and $\nu_{\rm sm} = -2$ in equation (6) retrieves the gravitational potential. Here, $k^h_{\rm sm}$, $k^l_{\rm sm}$ and $\nu_{\rm sm}$ are again free (learnable) parameters.

In the case of DM, we employ an expression for the Green's function which is slightly different with respect to LDL, as we find it works best:

$$\tilde{G}_{\rm dm}(k) = \exp\left[-\frac{k^h_{\rm dm}}{k}\right]\exp\left[-\frac{k}{k^l_{\rm dm}}\right]k^{\nu_{\rm dm}}\,. \tag{7}$$

In practice, each displacement layer is implemented as follows. Given particle positions $\boldsymbol{r}$, a density field is produced via CIC and raised to some power (learned and different between displacement layers), as indicated in equation (2). The potential in Fourier space is then computed as the product of the Discrete Fourier Transform (DFT) of this field and the Green's function parametrized based on the quantity to model – i.e. as in equation (6) for stellar mass, or as in equation (7) for DM. The gradient is calculated using the finite difference method; since, in real space, a finite difference approximation of a derivative can be written as a convolution, this simply translates into a multiplication by a filter in Fourier space. Finally, the gradient of the potential is transformed back into real space and interpolated from the grid points of the CIC mesh to the particle positions $\boldsymbol{r}$, using an interpolation scheme analogous to CIC.

After the displacement layers, a density map $\rho_{\rm out}$ is produced again using CIC. The final transformation applied to $\rho_{\rm out}$ depends on the target map type and is designed to capture the physical processes that cannot be modelled simply via matter transport, such as star formation for a stellar mass target and ionization for neutral hydrogen.

For stellar mass, the transformation has the form:

$$F_{\rm sm}(\boldsymbol{r}) = {\rm ReLU}(w_{\rm sm}(1+\delta_{\rm out}(\boldsymbol{r}))^{\mu_{\rm sm}} - b_{\rm sm})\,, \tag{8}$$

where $F_{\rm sm}(\boldsymbol{r})$ is the final value of stellar mass at grid point (or voxel) position $\boldsymbol{r}$, $w_{\rm sm}$, $\mu_{\rm sm}$ and $b_{\rm sm}$ are free (learnable) parameters, $\delta_{\rm out}$ is the overdensity associated to the density $\rho_{\rm out}$ and ReLU is the Rectified Linear Unit, defined as ${\rm ReLU}(x) = \max(x, 0)$. This transformation is meant to select regions of high density where star formation occurs.

In the case of DM, no final transformation is applied, implying

$$F_{\rm dm}(\boldsymbol{r}) = 1 + \delta_{\rm out}(\boldsymbol{r})\,. \tag{9}$$

The free parameters of the model are optimized by matching the output field $F(\boldsymbol{r})$ to a distribution of the physical quantity to be modelled $F_{\rm ref}(\boldsymbol{r})$, obtained from a high-resolution full-hydrodynamical reference simulation. This is achieved by minimizing an L1 or L2 loss:

$$\mathcal{L} = \frac{1}{N_{\rm grid}}\left\|\hat{O}_s F(\boldsymbol{r}) - \hat{O}_s F_{\rm ref}(\boldsymbol{r})\right\|^p\,, \tag{10}$$

where $\|\cdot\|^p$ is the pth norm ($p = 1, 2$), $N_{\rm grid}$ is the number of voxels in the maps F, $F_{\rm ref}$ and $\hat{O}_s$ is an optional smoothing operator (used for the stellar mass maps and H I at $z = 5$ only), defined in Fourier space as

$$\tilde{O}_s(k) = 1 + \left(\frac{k}{1h{\rm Mpc}^{-1}}\right)^{-n}\,, \tag{11}$$

where n is a hyperparameter that determines the relative weight between the large-scale and small-scale modes, introduced to prevent the model from focusing predominantly on the small scales. Note that, importantly, for a sensible voxel-based comparison, the target full-hydrodynamical and the approximate N-body simulations need to share the same initial white noise map. The way this is achieved in this work is detailed below. Similarly to the original implementation, in the case of DM, we use an L2 loss and no smoothing, while for stellar mass we use an L1 loss and a different n for different redshifts, as we shall explain below.

2.2 Predicting neutral hydrogen

Going beyond Dai & Seljak (2021), we now extend the model to predict H I maps. In this case, the key challenge is represented by galactic feedback, a complex process whose impact on the H I distribution propagates from small scales, where it originates, to large scales. The role of feedback on the H I distribution is similar to that on the overall gas distribution: at small/intermediate scales, galactic winds are likely to push gas outside galaxies in the low-density intergalactic medium (IGM), either with thermal or mechanical feedback, while at small scales baryons can condense and determine an increase of power compared to a simulation in which baryons are absent (see e.g. van Daalen, McCarthy & Schaye 2020). H I is expected to follow, at least to some extent, the gas distribution, providing an overall suppression of power at small/intermediate scales and increase of power at very small scales (van Daalen et al. 2011; Parimbelli, Viel & Sefusatti 2019; Schneider et al. 2019); however, differently from other baryonic components, the H I distribution is also sensitive to the local gas temperature, UV background and radiative transfer effects from astrophysical sources that ionize the surrounding medium, including patchy reionization (Puchwein et al. 2022).

Furthermore, the dynamical range of H I is much greater compared to that of, e.g. stellar mass, and it can stretch over several orders of magnitude, from neutral fractions of 10^{-7} in filaments up to fully neutral gas inside galaxies in self-shielded regions at low redshift (Villaescusa-Navarro et al. 2018): for this reason, devising an approximate model that can reproduce the H I distribution in its entire dynamical range is a very challenging task. None the less, different parts of this range can be probed through different observables: in particular, the high-density regions, found within haloes (i.e. at $z = 0$, with number densities above $\sim 10^{-10}\,{\rm cm}^{-3}$ and up to $\sim 10^{-5}\,{\rm cm}^{-3}$), are observable via 21 cm intensity mapping, while the low-to-intermediate densities, found in the IGM (at $z = 0$, with H I number density reaching down to $\sim 10^{-17}\,{\rm cm}^{-3}$ at $z = 0$), are relevant for the study of the Ly α forest.

In this work, we focus on the IGM; clearly, using the loss in equation (10) to train JERALD on a reference simulation that models the full dynamical range of H I without any pre-processing would result in the model focusing solely on the high-density range, which would dominate the loss. Since, for the purpose of modeling the Ly α forest, we are not interested in the high-density regions, this problem can be circumvented, for example, by clipping the H I density above a given threshold in the reference simulation, thus allowing the cosmic web to stand out and be learned. Here, this step is not necessary, as the reference simulations utilized do not implement self-shielding (Bolton et al. 2016). This results in the H I inside haloes being greatly underestimated: this is an unrealistic scenario, but if one focuses on the IGM rather than the halos, implementing self-shielding is not strictly necessary, and this is actually beneficial for our purpose as it reduces the dynamical range of H I. More generally, if the main focus is the IGM, the features of the distribution one has to model depend on the prescription used to mask out the high-density regions: the approach described below is therefore well-suited for reference simulations without self-shielding, while, for example, clipped maps would require a different treatment – we plan to explore this in the future.

In practice, to model neutral hydrogen, we modify the original LDL approach in two ways. First, we introduce a new term in the effective potential:

$$\boldsymbol{S} = \alpha^{(1)}_{\mathrm{H\,I}} \nabla \left(\hat{G}^{(1)}_{\mathrm{H\,I}} f^{(1)}_{\mathrm{H\,I}}(\delta) \right) + \alpha^{(2)}_{\mathrm{H\,I}} \nabla \left(\hat{G}^{(2)}_{\mathrm{H\,I}} f^{(2)}_{\mathrm{H\,I}}(\delta_{\mathrm{star}}) \right) , \tag{12}$$

where $\alpha^{(1)}_{\mathrm{H\,I}}$ and $\alpha^{(2)}_{\mathrm{H\,I}}$ are free parameters, both Green's operators $\hat{G}^{(1)}_{\mathrm{H\,I}}$ and $\hat{G}^{(2)}_{\mathrm{H\,I}}$ are defined as in equation (6) and have separate free parameters, $f^{(1)}_{\mathrm{H\,I}}$ and $f^{(2)}_{\mathrm{H\,I}}$ are power laws with separate free exponents as in equation (2), δ is once again the input particles overdensity while δ_{star} is the stellar overdensity associated to the distribution δ. The purpose of the second term in equation (12) is to approximately model the effect of galactic winds. In order to predict H I using solely the input DM particle positions, for δ_{star} we use the prediction obtained from JERALD as described in the previous section and pre-trained separately, i.e. $1 + \delta_{\mathrm{star}} = F_{\mathrm{sm}}$ from equation (8).

Secondly, we replace the final transformation of equation (8) with the expression:

$$F_{\mathrm{H\,I}}(\boldsymbol{r}) = \mathrm{ReLU}(w_{\mathrm{H\,I}}(1 + \delta_{\mathrm{out}}(\boldsymbol{r}))^{\mu_{\mathrm{H\,I}}} - D(\boldsymbol{r}) - b_{\mathrm{H\,I}}) , \tag{13}$$

where $D(\boldsymbol{r})$ is a depletion term given by

$$D = \mathrm{ReLU}(\beta^{(1)} \hat{O}_{\mathrm{I}} (1 + \delta_{\mathrm{star}}) + \eta) + \beta^{(2)} f^{(3)}(\delta_{\mathrm{star}}) , \tag{14}$$

where $\beta^{(1)}$, $\beta^{(2)}$, and η are free parameters, $f^{(3)}$ is once again a power law as in equation (2) with a different free exponent and $\hat{O}_{\mathrm{I}}$ is a filter defined in Fourier space as

$$\tilde{O}_{\mathrm{I}}(k) = \exp\left[-\frac{k^h_{\mathrm{I}}}{k} \right] \exp\left[-\frac{k}{k^l_{\mathrm{I}}} \right] k^{\nu_{\mathrm{I}}} + \xi , \tag{15}$$

with k^h_{I}, k^l_{I}, ν_{I}, and ξ learnable parameters. Heuristically, the first term in equation (14), which affects the H I distribution both where stars are present and in their surroundings, is meant to capture the depletion of H I due to processes such as ionization, while the second is more local and it accounts for the conversion between gas and stars.

We train all parameters as discussed in the previous section, using an L1 loss (see equation 10) as we find it produces the best results.

A full pipeline of the JERALD approach to produce H I maps, including the approximate N-body simulation and the steps to improve the DM distribution and to produce the stellar mass map, is illustrated in Fig. 1. All free parameters in the model for DM, stellar mass, and H I are summarized in Table 1.

3 CODE SPECIFICATIONS AND AVAILABILITY

The JERALD code is written partly in PYTHON using JAX (Bradbury et al. 2018) and partly in C++. The CIC paint and readout – i.e. the algorithms to compute density fields from properties defined at particle positions and vice-versa – and the DFTs are implemented in C++, the latter via MPI-enabled FFTW (Frigo 2005), and bound to PYTHON via PYBIND11 (Jakob, Rhinelander & Moldovan 2017). In PYTHON, CIC paint and readout as well as forward and backward DFTs are defined as JAX primitives and equipped with definitions of vector-Jacobian products,[2] making them jittable and available for automatic differentiation. All methods are parallelized using MPI – in PYTHON, via MPI4JAX (Häfner & Vicentini 2021). Each MPI rank deals with approximately $1/W$ of the particles and $1/W$ of the density fields, where W is the size of the MPI communicator, making the code scale approximately inversely proportional in time and constant in memory as W increases (see Fig. 2). For the 512^3 particles and 512^3 mesh points maps discussed in this work, each step in the optimization (evaluation of training and validation loss and training loss gradient) takes ∼15–20 s per displacement layer in one node of the Leonardo supercomputer in CINECA (32 CPUs). The number of optimization steps depends on the kind of reference map and on the optimizer parameters; generally, at most 500 steps are sufficient for the validation loss to reach convergence, and one training typically takes ∼2–5 hr, depending on the number of displacement layers (in this work, we used at most 3). Evaluation of the maps with a pre-trained model takes ∼10 s with the same resources.

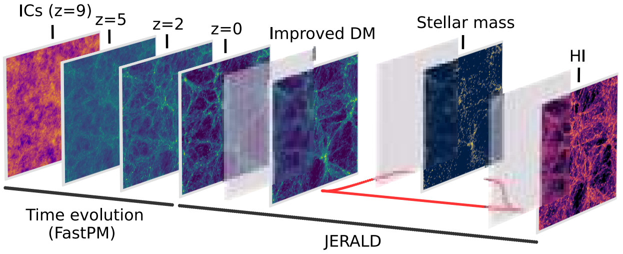

Figure 1. Schematic of the JERALD pipeline: the FASTPM approximate N-body solver starts from initial conditions at redshift $z = 9$ generated using 2LPT and then, though a small number of leapfrog integration steps, produces a final array of DM particle positions at $z = 5$, $z = 2$ and $z = 0$, the cases analysed in this work. Starting from these positions – here, we illustrate the case of $z = 0$ – JERALD improves the DM distribution by optimizing the free parameters of the displacement layers on the DM target from a full-hydrodynamical simulation. Then, JERALD applies additional displacement layers to the improved DM positions, and a final transformation to produce a stellar mass distribution, with parameters optimized on the stellar mass target. Finally, starting again from the improved DM particles positions and using the pre-trained stellar mass map, JERALD produces a H I map by displacing the particles via equation (12) and applying the transformation in equation (13). Grey layers represent additional intermediate displacement layers.

For a comparison of speed and memory usage with the original LDL code from Dai & Seljak (2021), in Fig. 2 we show time per optimization step (left) and resident set size (right) for 20 training steps in the case of a 512^3 voxels DM target with 512^3 particles and one displacement layer (the number of CPUs is also the size of the MPI communicator W). We observe that, while the scaling law for the time per optimization step is approximately the same, the proportionality coefficient for JERALD is significantly smaller with respect to the one of LDL; none the less, the curves for the RSS indicate that JERALD is also more memory efficient than LDL.

The JERALD code, together with installation requirements and instructions, is available at https://github.com/maurorigo/JERALD/tree/main.

---

[2] For a definition and an application to JAX, see https://docs.jax.dev/en/latest/notebooks/autodiff_cookbook.html.

Table 1. JERALD parameters for DM, stellar mass and H I targets, at a given redshift. N_p is the number of parameters in each part of the model and N_p^{tot} the total number of parameters for each predicted quantity, with S_{dm}, S_{sm} and $S_{\mathrm{H\,I}}$ being the number of displacement layers used for DM, stellar mass and H I respectively (all set to 2 except for $S_{\mathrm{sm}} = 3$ at $z = 5$). N_p^{used} is the total the number of parameters used in this work (except for stellar mass at $z = 5$, where $N_p^{\mathrm{used}} = 18\ (+10)$). For stellar mass and H I, the total number of parameters also includes, in brackets, those used to improve the DM distribution (for both) and to produce the stellar mass map (for H I).

Quantity		Equation	Parameters	N_p	N_p^{tot}	N_p^{used}
Dark matter	Displacement	(2)	$\alpha_{\mathrm{dm}}, \gamma_{\mathrm{dm}}$	$2S_{\mathrm{dm}}$	$5S_{\mathrm{dm}}$	10
	Green's function	(7)	$k^h_{\mathrm{dm}}, k^l_{\mathrm{dm}}, \nu_{\mathrm{dm}}$	$3S_{\mathrm{dm}}$		
	Transformation	–	–	–		
Stellar mass	Displacement	(2)	$\alpha_{\mathrm{sm}}, \gamma_{\mathrm{sm}}$	$2S_{\mathrm{sm}}$	$5S_{\mathrm{sm}} + 3\ (+5S_{\mathrm{dm}})$	13 (+10)
	Green's function	(6)	$k^h_{\mathrm{sm}}, k^l_{\mathrm{sm}}, \nu_{\mathrm{sm}}$	$3S_{\mathrm{sm}}$		
	Transformation	(8)	$w_{\mathrm{sm}}, \mu_{\mathrm{sm}}, b_{\mathrm{sm}}$	3		
Neutral hydrogen	Displacement	(12)	$\alpha^{(1)}_{\mathrm{H\,I}}, \gamma^{(1)}_{\mathrm{H\,I}}$	$2S_{\mathrm{H\,I}}$	$10S_{\mathrm{H\,I}} + 11$ $(+5S_{\mathrm{dm}} + 5S_{\mathrm{sm}} + 3)$	31 (+10 + 13)
			$\alpha^{(2)}_{\mathrm{H\,I}}, \gamma^{(2)}_{\mathrm{H\,I}}$	$2S_{\mathrm{H\,I}}$		
	Green's function(s)	(6)	$k^{h\,(1)}_{\mathrm{H\,I}}, k^{l\,(1)}_{\mathrm{H\,I}}, \nu^{(1)}_{\mathrm{H\,I}}$	$3S_{\mathrm{H\,I}}$		
			$k^{h\,(2)}_{\mathrm{H\,I}}, k^{l\,(2)}_{\mathrm{H\,I}}, \nu^{(2)}_{\mathrm{H\,I}}$	$3S_{\mathrm{H\,I}}$		
	Transformation	(13)	$w_{\mathrm{H\,I}}, \mu_{\mathrm{H\,I}}, b_{\mathrm{H\,I}}$	3		
		(14)	$\beta^{(1)}, \beta^{(2)}, \eta, \gamma^{(3)}_{\mathrm{H\,I}}$	4		
		(15)	$k^h_{\mathrm{I}}, k^l_{\mathrm{I}}, \nu_{\mathrm{I}}, \xi$	4		

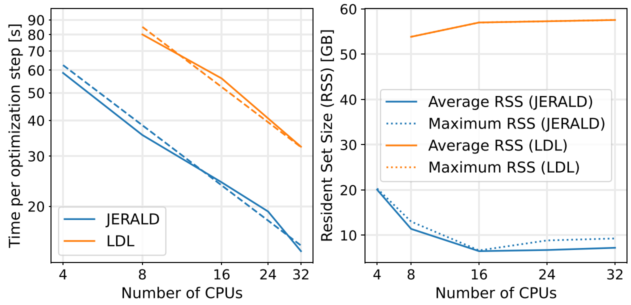

Figure 2. Time per optimization step (left) and resident set size (RSS; right, a measure of memory used) for JERALD and the original LDL code from Dai & Seljak (2021) on one node of the Leonardo supercomputer. The RSS is given by the slurm tool `sacct`, which may not capture rapid memory usage spikes. Times are given as averages over 20 optimization steps in the case of 512^3 particles and a 512^3 voxels DM target map and 1 displacement layer, and the RSS values are those associated with the 20 steps. The dashed lines in the plot on the left indicate the power law $\propto (N_{\mathrm{CPUs}})^{-0.7}$ for reference. The maximum RSS for LDL is not visible as it coincides with the average RSS.

4 RESULTS

We now turn to evaluating JERALD's performance. We first give details about the reference simulations and the training procedure, then we investigate the quality of JERALD's outputs against the same maps used for training, and finally we provide a first assessment of its generalization capabilities by applying the trained model to an approximate N-body simulation with new initial conditions (ICs) and compare the output maps with targets that were not seen during training.

4.1 Simulations and training

The reference maps used in the subsequent section were extracted from the Sherwood–Relics simulation suite (Bolton et al. 2016; Puchwein et al. 2022). This set of simulations follows the evolution of DM, gas and stars in a periodic cosmological volume, and the key physical ingredients are: (i) an evolving UV background that returns a thermal history for the IGM which is in agreement with observations; (ii) a simplified star formation criterion which converts into star particles all the gas particles whose overdensities are larger than 1000 times the mean baryon density and whose temperatures are below 10^5 K. These simulations are supposed to offer an accurate description of the filamentary cosmic web, and in particular Ly α forest statistics in agreement with observations (Iršič et al. 2024). The specific simulations used here feature a box size of $80h^{-1}$ Mpc and 1024^3 DM and gas particles, with fiducial parameters $\Omega_m = 0.308$, $\Omega_\Lambda = 0.692$, $\Omega_b = 0.0482$, $h = 0.678$, $\sigma_8 = 0.829$, and $n_s = 0.961$.

Input DM particles positions for JERALD are computed using the approximate N-body solver FASTPM (Feng et al. 2016); we generate the ICs for 512^3 particles at $z = 9$ using 2LPT with a $(2 \times 512)^3$ linear density map, and run 10 FASTPM steps up to the desired redshift values ($z = 0, 2, 5$). Crucially, the initial white noise map is generated using a random number generator compatible with the one used for the target Sherwood-Relics simulation (part of the N-GENIC code by Springel (2005)) and the same seed, as well as same initial power spectrum and σ_8 (through the normalization constant of the power spectrum). Starting from the FASTPM input, JERALD's outputs were computed using the same 512^3 particles and a 512^3 mesh for all fields, making the resolution of the output maps 8× smaller than the reference simulation.

We use the Adam optimizer (Kingma & Ba 2017) with a scheduled learning rate ranging from 0.002 to 0.04 after extensive experimentation with other optimizers and hyperparameters. Model parameters are initialized to arbitrary values, and, in the code, they are multiplied by constants to loosely guarantee them to vary in the same range for a more stable training procedure. Specifically, all parameters are expected to be $\mathcal{O}(1)$ except for the coefficients in front of the potential terms (e.g. α in equation 2), which are generally of order 10^{-2}. Yet, parameters that are initially considerably far from the optimal ones may be associated with very large gradients; for this reason, we adopt a time-varying schedule for the learning rate $\eta(i)$, increasing it during training following a softmax schedule:

$$\eta(t) = (\eta(N_{\mathrm{step}}) - \eta(0)) \left(\frac{2}{e^{-t/\tau_\eta} + 1} - 1 \right) + \eta(0), \tag{16}$$

where generally, for N_{step} training steps, $\eta(0) \sim \mathcal{O}(10^{-3})$, $\eta(N_{\mathrm{step}}) \sim \mathcal{O}(10^{-2})$, and $\tau_\eta \sim 100$.

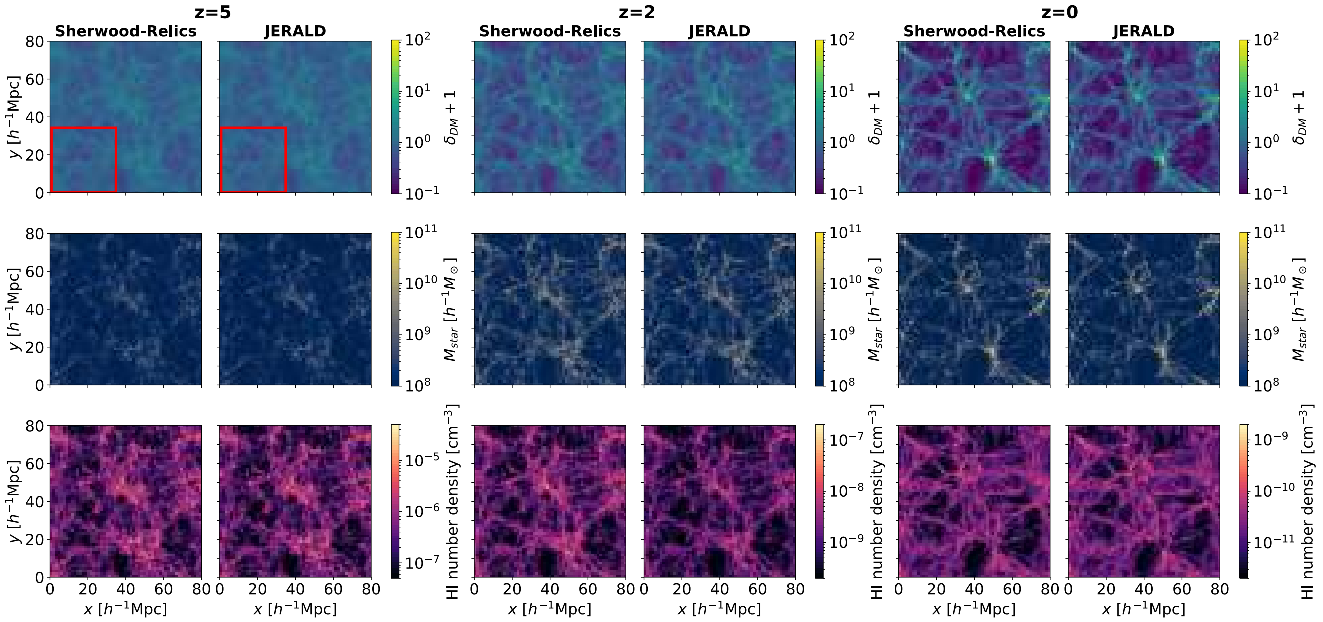

Figure 3. DM overdensity, stellar mass and neutral hydrogen number density maps, from top to bottom. Each pair of maps compares the reference map from Sherwood-Relics (left panel) with JERALD's output (right panel), for $z = 5$, $z = 2$, and $z = 0$ (left to right). Notice the different range in the colourmaps for neutral hydrogen. All images were obtained as $10h^{-1}$ Mpc z-slices of the full $(80h^{-1}\mathrm{Mpc})^3$ simulation box. All voxels used to produce the images were part of either the validation or the test set; the red rectangles in the top left pair of maps–omitted in the others–represent the test sub-box, and it extends for the full length of the z-slice used to produce these images. Note that the H I map for $z = 0$ was normalized with the average of the reference map.

The training/validation/test sets splitting is chosen as follows: for a $M \times M \times M$ map, where M is the number of voxels in each direction, we use a $(0.5M) \times M \times M$ box for training, a $(0.43M) \times (0.43M) \times (0.43M)$ box for testing, and the rest for validation, which correspond to approximately 50 per cent of the pixels in the original map for training, 42 per cent for validation and 8 per cent for testing.

As shown in Table 1, all results included in the next sections were obtained using two displacement layers for each target at each redshift, except in the case of stellar mass for $z = 5$, where three displacement layers were utilized. We adopted a unit n hyperparameter for the smoothing operator in equation (11) in the case of stellar mass at $z = 0$ and $z = 2$ and H I at $z = 5$, while for stellar mass at $z = 5$ $n = 0.7$ was found to give the best result. In order to bring all training values to the same scale, of order unity, all calculations involving stellar mass were performed in units of $10^{10}h^{-1}M_{\odot}$, while target H I number densities were divided by a redshift-dependent constant. For completeness, we report all the optimized parameters used to obtain the results in the next sections in Appendix A (Tables A1–A3). Note that optimizations starting from different initial guesses for the model parameters can lead to different optimized values, yet with very similar performances. This is discussed in further detail in Appendix B. As this indicates the presence of similar local minima, or non-trivial degeneracies in parameter space, assigning uncertainties to the optimized parameters is not particularly informative, so we refrain from doing so.

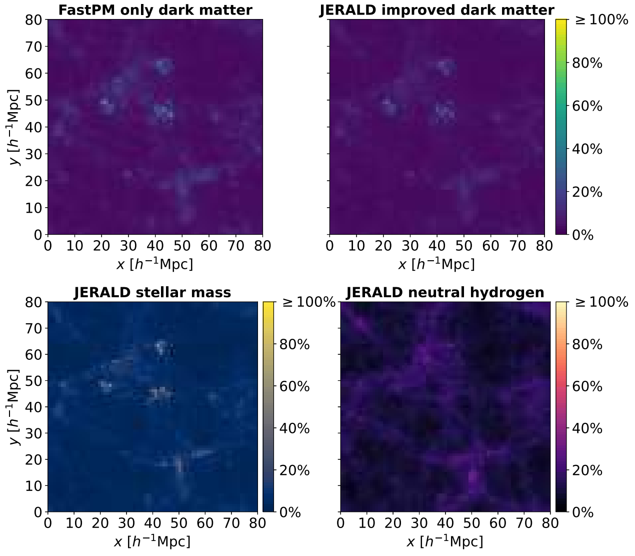

Figure 4. Relative difference between $z = 0$ FASTPM only and JERALD's output for DM, stellar and neutral hydrogen maps. Each map used to compute the relative differences was clipped below 0.5 times the average mass of the reference map and smoothed with a Gaussian filter with a radius of $200h^{-1}$kpc. The plots show the relative difference averaged over the entire length of the $80h^{-1}$ Mpc box along the z-direction.

4.2 Evaluation of JERALD's output maps

Fig. 3 shows slices of DM overdensity, stellar mass and neutral hydrogen number density maps obtained using JERALD from an input approximate DM distribution from FASTPM at redshift $z = 0$, $z = 2$ and $z = 5$, compared with the respective Sherwood-Relics references. The red squares show the test region – i.e. the part of the cube that has not been used for training nor validation. All the voxels used to produce the images in the figure were either part of the validation or the test sets. The visual agreement is remarkable.

To evaluate more quantitatively the performances of JERALD, in Figs 4 and 5 we also show the relative difference between the refer-

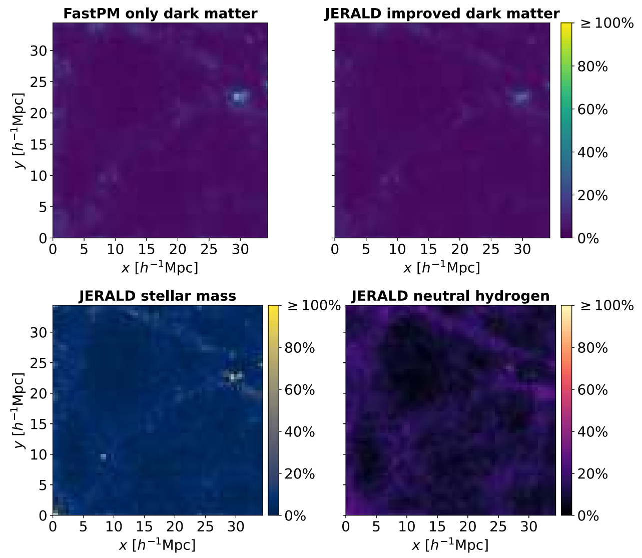

Figure 5. As in Fig. 4, but showing the relative difference only for the test set, averaged over the 34.4h^{-1} Mpc of the test sub-box along the z direction.

ence maps and JERALD's outputs, both post-processed in two steps.[3] First, each map was clipped below 0.5 $\bar{\phi}_i$, where $\bar{\phi}_i$, i = dm/sm/H I, are the averages of the reference Sherwood-Relics DM, stellar mass or H I maps. This is because we expect mesh points with small density/mass to only bring minor contributions to the quantities we plan to compute using maps produced by JERALD. By construction, the loss in equation (10) prioritizes matching mesh points with large values; for this reason, mesh points with small values, being very abundant, may produce large relative differences which would dominate over the ones associated to the actual range of interest. Indeed, in each case (DM, stellar mass and H I), the clipped values are above 80 per cent of the entire map, but the mass contribution of the rest of the map is at least 10 times greater. The clipping threshold is chosen arbitrarily and it relies on the knowledge of the averages of the reference maps $\bar{\phi}_i$, i = dm/sm/HI, only to define a value that is small *relative to each map type*; of course, when using JERALD on unseen maps, these averages would not be available nor necessary. Secondly, each map was smoothed using a Gaussian filter with a radius of 200h^{-1}kpc, since, in view of using this approach to model real data, we expect very small scale differences not to lead to significant mismatches. We only illustrate the relative differences for $z = 0$, as distributions become progressively more non-linear for lower redshifts and therefore more complicated to model – the relative difference maps for $z = 2$ and $z = 5$ display considerably less features than the ones illustrated here. Despite the high resolution and small smoothing radius, the relative differences are characterized by only several notable features, with most of the maps lying below 20 per cent. Note that the locations of the prominent features present in the maps are approximately the same.

To analyse the statistics of JERALD's outputs, we also compare the power spectra of the maps with the reference ones in Figs 6–8 for DM, stellar and neutral hydrogen overdensities, respectively. Similarly to Dai & Seljak (2021), the power spectra were computed

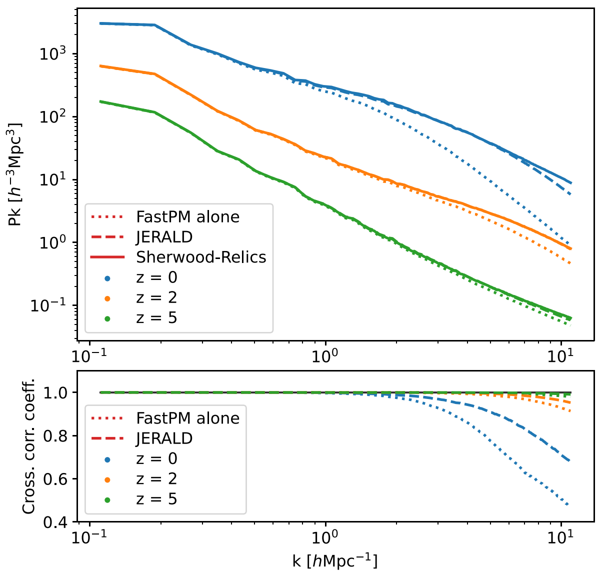

Figure 6. DM power spectra (top) and cross-correlation coefficients (bottom) for the reference Sherwood-Relics simulation (solid), the approximate N-body FASTPM simulation (dotted), and the map improved by JERALD (dashed) for three different redshifts. For $z = 2$ and $z = 5$, the improved power spectra are almost not visible, as they overlap with the reference ones.

on the entire (80h^{-1} Mpc)^3 simulation box, rather than on the test subset only, in order to capture the large scales as well and determine whether the model affects the distributions at those scales. Including the training data in this analysis may bias the results, therefore in the next section we will also analyse the performance of the approach on unseen maps. The bottom panel of Figs 6–8 shows the cross-correlation coefficient, calculated as

$$c(k) = \frac{P_{\rm cross}(k)}{\sqrt{P_{\rm ref}(k)P_{\rm JERALD}(k)}}, \tag{17}$$

where $P_{\rm cross}(k)$ is the cross power spectrum and $P_{\rm ref}(k)$ and $P_{\rm JERALD}(k)$ are the power spectra of the Sherwood-Relics reference simulation and JERALD, respectively. All power spectra and cross-correlation coefficients were calculated using the PYLIANS libraries. In each case and at each redshift, the power spectra of the maps produced by JERALD are in excellent agreement with the reference ones even up to highly non-linear scales. As expected, the cross-correlation coefficients show that the agreement between the maps increases as redshift increases. Note that, in computing the stellar overdensities and H I overdensity at $z = 5$, we used the average of the reference maps rather than that of the maps produced by JERALD. This was done because the introduction of the smoothing filter of equation (11) in the loss leads to sharp mass peaks being smeared out and, as a consequence, the average of the map produced by JERALD is smaller than the one of the reference (though only by a small factor of ~ 5 per cent). None the less, these mass peaks do not bring significant contributions to the overall shape of the power spectra, as Figs 7 and 8 highlight, and we expect them not to pose problems in applications of this method.

4.3 Generalization capabilities

Until now, we presented results evaluated on the same reference maps used in the training procedure. Despite the fact that only 50 per cent of the voxels in each map are used for training and the model

---

[3] We emphasize that these two steps were only implemented when computing relative differences; for all plots of each individual map, all power spectra and all cross-correlation coefficients the maps used are the ones produced by JERALD, unprocessed.

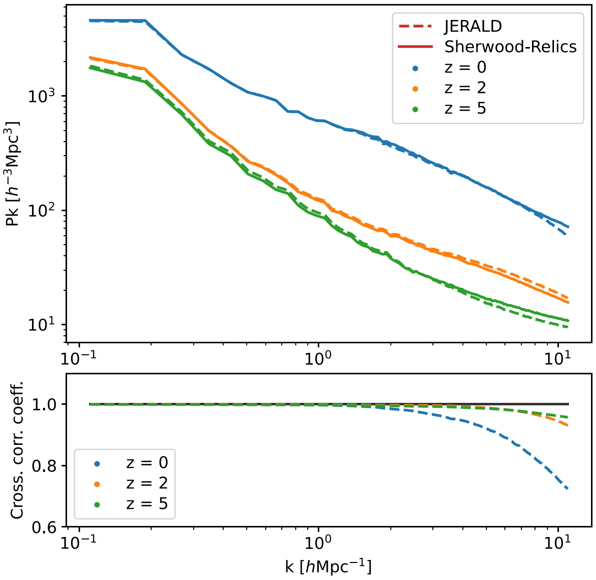

Figure 7. Stellar power spectra (top) and cross-correlation coefficients (bottom) for the reference Sherwood-Relics simulation (solid) and the map produced using JERALD (dashed) at three different redshifts.

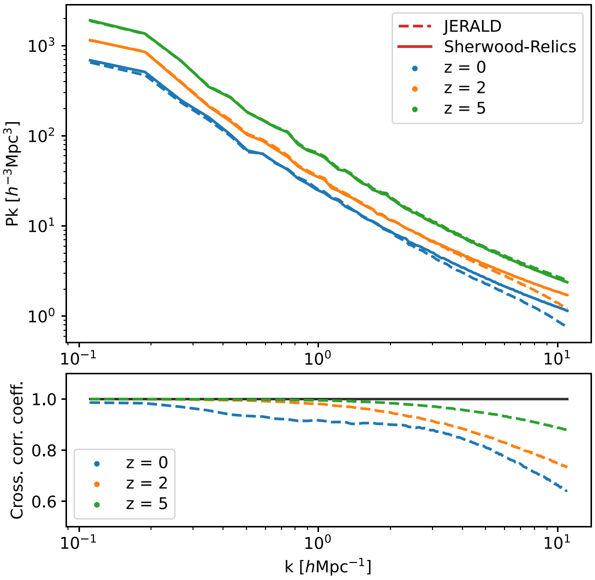

Figure 8. Low-intermediate density neutral hydrogen power spectra (top) and cross-correlation coefficients (bottom) for the reference Sherwood-Relics simulation (solid) and the map produced using JERALD (dashed), at three different redshifts. For visualization purposes, the green curves ($z = 5$) in the top panel were offset by multiplying them by an arbitrary constant (1.7), otherwise they would overlap with the orange ones ($z = 2$). The correct scale of the power spectra at $z = 5$ is the same as for $z = 2$.

has only few parameters, the power spectra and relative difference plots reported above may still be affected by overfitting; for this reason, in this section we investigate the performance of JERALD on unseen simulations. Specifically, we run a new full-hydrodynamical simulation using the same code as the one used for the reference simulation of the previous sections, same box and mesh sizes and same cosmological and astrophysical parameters, but a different seed for the generation of the ICs. We then run FASTPM as in Section 4.1 with the new ICs and apply JERALD with fixed parameters, trained on the simulation used in the previous sections.

Fig. 9 shows DM overdensity, stellar mass and neutral hydrogen number density maps as in Fig. 3 with the new unseen simulation. We only report the $z = 0$ maps, as they are the most difficult to model. The visual agreement is good also in this case, with mild discrepancies for H I in the central band. In Fig. 10, we show the associated power spectra and cross- correlation coefficients, including the $z = 2$ case for completeness. We compare the new reference (solid) with JERALD's output (dashed), and we observe similar performance as what was obtained on the reference simulation on which JERALD was trained. This is a strong indication of the method's generalization capabilities, which we plan to investigate in more detail in future works, when we will also evaluate the performance for previously unseen simulations with different astrophysical and cosmological parameters as well as transferability of results to different simulation codes.

5 CONCLUSIONS

We presented a new, improved implementation of a physically motivated approach to reducing the enormous computational cost of cosmological simulations in order to obtain accurate DM and baryonic maps. Our code, JERALD (JAX Enhanced Resolution Approximate Lagrangian Dynamics), builds on the Lagrangian Deep Learning method originally proposed by Dai & Seljak (2021), and focuses in particular on producing, on top of more accurate DM and stellar maps, high-fidelity neutral hydrogen maps.

We showed that our improved implementation is capable of producing maps in excellent agreement, in terms of power spectra for DM, stars and H I, with full-hydrodynamical simulations, up to several h Mpc^{-1}. The performances of JERALD are more impressive at higher redshift ($z \geq 2$), where non-linearities are less prominent: in this regime, the agreement, in terms of power spectra, is above the 70 per cent level all the way to very small scales ($k \gtrsim 10\,h$ Mpc^{-1}). This opens the door to investigating, in quantitative detail, the impact on very small scales of different DM models.

JERALD was developed from scratch, with the aim of being user-friendly, fully parallel and highly computationally efficient. Its ultimate goal is not limited to producing baryonic maps: we aim at exploiting this approach to study observable H I-related quantities such as the Ly α forest and the 21cm intensity (Villaescusa-Navarro et al. 2018). These two observables are particularly important, as neutral hydrogen is an excellent probe of DM models beyond cold DM both in emission and absorption (Carucci et al. 2015; Iršič et al. 2024) – an alternative that could in principle alleviate some of the small-scale inconsistencies of the ΛCDM model: for example, a lighter ('warm') DM particle could offer the solution to the core/cusp problem, as the main effect of its larger velocity is to 'smear out' structures at small scales (Bode, Ostriker & Turok 2001; Weinberg et al. 2015). The quantitative study of such models from large-scale structure and Ly α forest, as well as 21cm mapping, can benefit from being embedded in a neural network-enabled simulation-based inference (SBI) framework (e.g. Charnock, Lavaux & Wandelt 2018; Alsing et al. 2019; Makinen et al. 2022, 2025), full high-dimensional likelihood evaluation (Seljak et al. 2017; Dai & Seljak 2022) or field-level inference (e.g. Porqueres et al. 2021) approaches, which have been shown to extract a great deal more information from the data than traditional hand-crafted summaries such as the power spectrum. Often, the bottleneck to the full exploitation of such techniques lies in the limited amount of high-resolution simulations on which to train

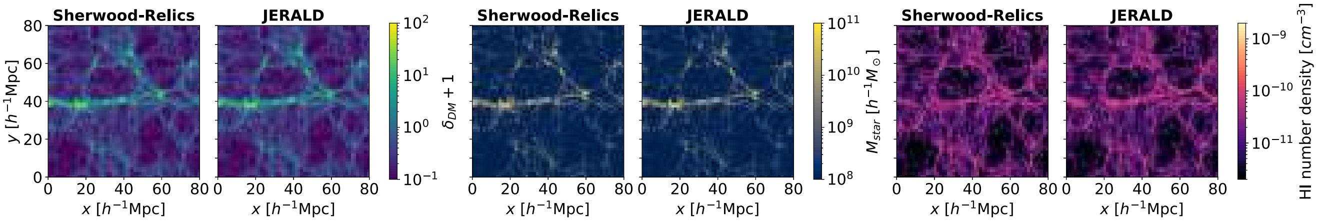

Figure 9. Similar to Fig. 3, but for a new Sherwood-Relics simulation unseen by JERALD. In this case, all plots correspond to $z = 0$ and the different quantities – i.e. DM overdensity, stellar mass and H I number density – are arranged horizontally. Here, no test sub-box is highlighted, because JERALD was pre-trained on a different simulation.

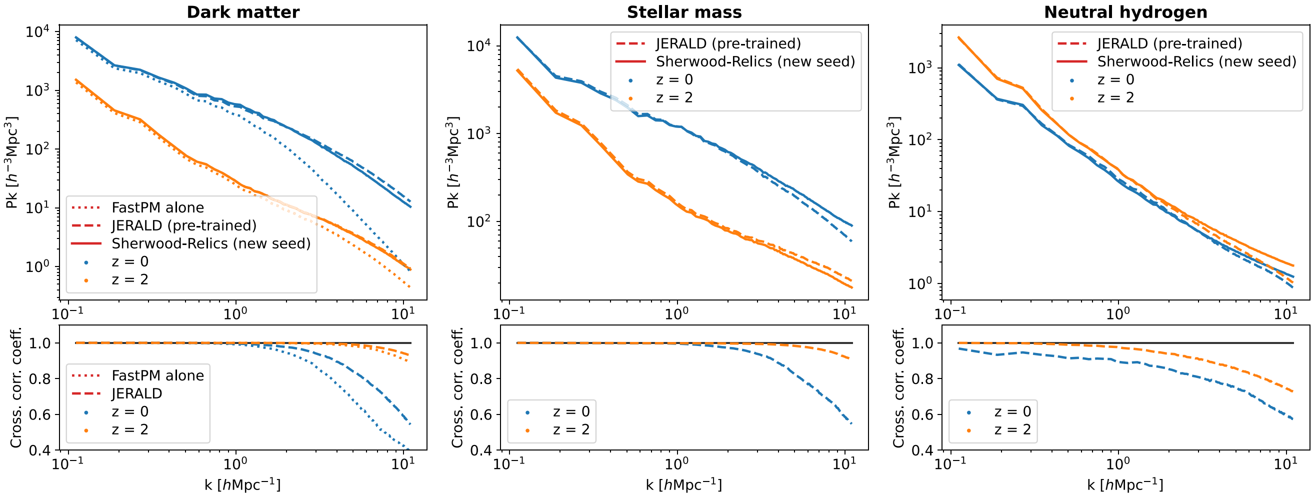

Figure 10. DM, stellar and neutral hydrogen power spectra at $z = 0$ and $z = 2$ for a target simulation with box size $(80h^{-1}\ \mathrm{Mpc})^3$ and mesh size 1024^3 with different initial conditions with respect to the one on which JERALD's parameters were trained, along with the results from JERALD.

the neural models. With JERALD, we have introduced a tool that will enable us to take a major step in future work to enable such studies.

ACKNOWLEDGEMENTS

We thank Francisco Villaescusa-Navarro as the main developer of ⟨0:sc ⟩pylians⟨/0:sc⟩, and an anonymous referee for helpful suggestions. RT acknowledges co-funding from Next Generation EU, in the context of the National Recovery and Resilience Plan, Investment PE1–Project FAIR ‘Future Artificial Intelligence Research’. This resource was co-financed by the Next Generation EU [DM 1555 del 11.10.22]. RT and MV are partially supported by the Fondazione ICSC, Spoke 3 ‘Astrophysics and Cosmos Observations’, Piano Nazionale di Ripresa e Resilienza Project ID CN00000013 ‘Italian Research Center on High-Performance Computing, Big Data and Quantum Computing’ funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di ‘campioni nazionali di R&S (M4C2-19)’ – Next Generation EU (NGEU). The Sherwood and Sherwood-Relics simulations used in this work were possible thanks to the Partnership for Advanced Computing in Europe (PRACE), during the 16th Call. The Sherwood-Relics simulations were also performed using time allocated during the Science and Technology Facilities Council (STFC) DiRAC 12th Call. We used the Cambridge Service for Data Driven Discovery (CSD3), part of which is operated by the University of Cambridge Research Computing on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The DiRAC component of CSD3 was funded by BEIS capital funding via STFC capital grants ST/P002307/1 and ST/R002452/1 and STFC operations grant ST/R00689X/1. Part of the simulations were postprocessed on Ulysses supercomputer at SISSA. MV and RT are also partially supported by INFN INDARK grant.

DATA AVAILABILITY

The JERALD code, together with installation requirements and instructions, is available at https://github.com/maurorigo/JERALD/tree/main. All data used to generate the results in this paper are available upon request.

REFERENCES

Agarwal S., Davé R., Bassett B. A., 2018, MNRAS, 478, 3410
Alsing J., Charnock T., Feeney S., Wandelt B., 2019, MNRAS, 488, 4440
Amendola L. e., 2018, Living Rev. Relativ., 21, 2
Barrera M. et al., 2023, MNRAS, 525, 6312
Baugh C. M., 2006, Rep. Prog. Phys., 69, 3101
Behroozi P. S., Conroy C., Wechsler R. H., 2010, ApJ, 717, 379
Behroozi P., Wechsler R. H., Hearin A. P., Conroy C., 2019, MNRAS, 488, 3143
Berlind A. A., Weinberg D. H., 2002, ApJ, 575, 587
Bernardini M., Feldmann R., Anglés-Alcázar D., Boylan-Kolchin M., Bullock J., Mayer L., Stadel J., 2021, MNRAS, 509, 1323
Bode P., Ostriker J. P., Turok N., 2001, ApJ, 556, 93
Bolton J. S., Puchwein E., Sijacki D., Haehnelt M. G., Kim T.-S., Meiksin A., Regan J. A., Viel M., 2016, MNRAS, 464, 897

Bourdin A., Legin R., Ho M., Adam A., Hezaveh Y., Perreault-Levasseur L., 2024, preprint (arXiv:2408.00839)

Bradbury J. et al., 2018, JAX: composable transformations of Python+NumPy programs. http://github.com/jax-ml/jax

Bullock J. S., Boylan-Kolchin M., 2017, ARA&A, 55, 343

Carucci I. P., Villaescusa-Navarro F., Viel M., Lapi A., 2015, J. Cosmol. Astropart. Phys., 2015, 047

Chadayammuri U., Ntampaka M., ZuHone J., Bogdán A., Kraft R. P., 2023, MNRAS, 526, 2812

Charnock T., Lavaux G., Wandelt B. D., 2018, Phys. Rev. D, 97, 083004

Croton D. J. et al., 2016, ApJS, 222, 22

DESI Collaboration, 2016, preprint (arXiv:1611.00036)

Dai B., Seljak U., 2021, Proc. Natl. Acad. Sci., 118, e2020324118

Dai B., Seljak U., 2022, MNRAS, 516, 2363

Dai B., Feng Y., Seljak U., Singh S., 2020, J. Cosmol. Astropart. Phys., 2020, 002

Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, ApJ, 292, 371

Feng Y., Chu M.-Y., Seljak U., McDonald P., 2016, MNRAS, 463, 2273

Flores R. A., Primack J. R., 1994, ApJ, 427, L1

Frigo M., Johnson S. G., 2005, Proceedings of the IEEE, 93, 216

Häfner D., Vicentini F., 2021, J. Open Source Softw., 6, 3419

Iršič V. et al., 2024, Phys. Rev. D, 109, 043511

Ivezić Ž. et al., 2019, ApJ, 873, 111

Jakob W., Rhinelander J., Moldovan D., 2017, pybind11–Seamless operability between C++11 and Python.

Kamdar H. M., Turk M. J., Brunner R. J., 2015, MNRAS, 455, 642

Kamdar H. M., Turk M. J., Brunner R. J., 2016, MNRAS, 457, 1162

Kingma D. P., Ba J., 2017, preprint (arXiv:1412.6980), https://arxiv.org/abs/1412.6980

Klypin A., Kravtsov A. V., Valenzuela O., Prada F., 1999, ApJ, 522, 82

Makinen T. L., Charnock T., Lemos P., Porqueres N., Heavens A. F., Wandelt B. D., 2022, Open J. Astrophys., 5

Makinen T. L., Heavens A., Porqueres N., Charnock T., Lapel A., Wandelt B. D., 2025, J. Cosmol. Astropart. Phys., 2025, 095

Moore B., 1994, Nature, 370, 629

Moore B., Quinn T., Governato F., Stadel J., Lake G., 1999, MNRAS, 310, 1147

Moster B. P., Somerville R. S., Maulbetsch C., van den Bosch F. C., Macciò A. V., Naab T., Oser L., 2010, ApJ, 710, 903

Moster B. P., Naab T., White S. D. M., 2012, MNRAS, 428, 3121

Nelson D. et al., 2017, MNRAS, 475, 624

Parimbelli G., Viel M., Sefusatti E., 2019, J. Cosmol. Astropart. Phys., 2019, 010

Pillepich A. et al., 2017, MNRAS, 473, 4077

Porqueres N., Heavens A., Mortlock D., Lavaux G., 2021, MNRAS, 502, 3035

Puchwein E. et al., 2022, MNRAS, 519, 6162

SWG S. C., 2020, PASA, 37, e007

Schneider A., Teyssier R., Stadel J., Chisari N. E., Le Brun A. M. C., Amara A., Refregier A., 2019, J. Cosmol. Astropart. Phys., 2019, 020

Seljak U., Aslanyan G., Feng Y., Modi C., 2017, J. Cosmol. Astropart. Phys., 2017

Springel V., 2005, MNRAS, 364, 1105

Springel V. et al., 2005, Nature, 435, 629

Tanner S., 2024, International Conference on Machine Learning for Astrophysics. https://indico.ict.inaf.it/event/2690/contributions/18549/contribution.pdf

Tassev S., Zaldarriaga M., Eisenstein D. J., 2013, J. Cosmol. Astropart. Phys., 2013, 036

Tröster T., Ferguson C., Harnois-Déraps J., McCarthy I. G., 2019, MNRAS, 487, L24

van Daalen M. P., Schaye J., Booth C. M., Dalla Vecchia C., 2011, MNRAS, 415, 3649

van Daalen M. P., McCarthy I. G., Schaye J., 2020, MNRAS, 491, 2424

Villaescusa-Navarro F. et al., 2018, ApJ, 866, 135

Villaescusa-Navarro F. et al., 2021, ApJ, 915, 71

Wadekar D., Villaescusa-Navarro F., Ho S., Perreault-Levasseur L., 2023, ApJS, 265, 54

Weinberg D. H., Bullock J. S., Governato F., Kuzio de Naray R., Peter A. H. G., 2015, Proc. Natl. Acad. Sci., 112, 12249

Xu X., Ho S., Trac H., Schneider J., Poczos B., Ntampaka M., 2013, ApJ, 772, 147

Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, preprint (arXiv:1902.05965)

APPENDIX A: OPTIMIZED PARAMETERS

For completeness, in Tables A1–A3, we report the optimized JERALD parameters used for the results presented in this work.

Table A1. Optimized JERALD parameters for dark matter, stellar mass and H I, at $z = 0$.

Quantity	Parameter	Layer 1	Layer 2	Output
Dark matter	$\alpha_{\rm dm}$	−0.06984	−0.05698	–
	$\gamma_{\rm dm}$	0.6319	0.3526	–
	$k^h_{\rm dm}$	0.01734	0.08013	–
	$k^l_{\rm dm}$	0.6846	7.222	–
	$\nu_{\rm dm}$	1.046	−1.542	–
Stellar mass	$\alpha_{\rm sm}$	−0.003235	−0.008951	–
	$\gamma_{\rm sm}$	0.4412	0.4832	–
	$k^h_{\rm sm}$	1.24	1.247	–
	$k^l_{\rm sm}$	8.286	8.015	–
	$\nu_{\rm sm}$	0.08315	−0.05298	–
	$w_{\rm sm}$	–	–	0.00498
	$\mu_{\rm sm}$	–	–	0.9032
	$b_{\rm sm}$	–	–	0.04294
Neutral hydrogen	$\alpha^{(1)}_{\rm H\,I}$	−0.003331	−0.004195	–
	$\gamma^{(1)}_{\rm H\,I}$	0.3914	0.4102	–
	$k^{h\,(1)}_{\rm H\,I}$	0.00001	0.00001	–
	$k^{l\,(1)}_{\rm H\,I}$	7.529	6.701	–
	$\nu^{(1)}_{\rm H\,I}$	0.4313	0.8997	–
	$\alpha^{(2)}_{\rm H\,I}$	0.001248	0.001384	–
	$\gamma^{(2)}_{\rm H\,I}$	0.1498	0.2117	–
	$k^{h\,(2)}_{\rm H\,I}$	3.307	3.813	–
	$k^{l\,(2)}_{\rm H\,I}$	8.492	9.049	–
	$\nu^{(2)}_{\rm H\,I}$	0.4263	0.681	–
	$w_{\rm H\,I}$	–	–	2.015
	$\mu_{\rm H\,I}$	–	–	0.915
	$b_{\rm H\,I}$	–	–	1.153
	$\beta^{(1)}$	–	–	3.52
	η	–	–	−5.395
	$k^h_{\rm I}$	–	–	0.0
	$k^l_{\rm I}$	–	–	0.9679
	$\nu_{\rm I}$	–	–	0.09137
	ξ	–	–	0.9635
	$\beta^{(2)}$	–	–	−2.75
	$\gamma^{(3)}_{\rm H\,I}$	–	–	0.7099

Table A2. Optimized JERALD parameters for dark matter, stellar mass and H I, at $z = 2$.

Quantity	Parameter	Layer 1	Layer 2	Output
Dark matter	$\alpha_{\rm dm}$	−0.02884	−0.02406	–
	$\gamma_{\rm dm}$	0.9202	0.4641	–
	$k^h_{\rm dm}$	0.0	0.7675	–
	$k^l_{\rm dm}$	0.7628	7.741	–
	$\nu_{\rm dm}$	−0.616	−1.115	–
Stellar mass	$\alpha_{\rm sm}$	−0.006641	0.001087	–
	$\gamma_{\rm sm}$	0.5129	0.4849	–
	$k^h_{\rm sm}$	1.146	1.002	–
	$k^l_{\rm sm}$	8.174	7.989	–
	$\nu_{\rm sm}$	0.04874	−0.009964	–
	$w_{\rm sm}$	–	–	0.003047
	$\mu_{\rm sm}$	–	–	0.9941
	$b_{\rm sm}$	–	–	0.01744
Neutral hydrogen	$\alpha^{(1)}_{\rm H\,I}$	−0.002289	−0.001464	–
	$\gamma^{(1)}_{\rm H\,I}$	0.6629	0.8039	–
	$k^{h\,(1)}_{\rm H\,I}$	0.00001	0.00001	–
	$k^{l\,(1)}_{\rm H\,I}$	7.645	7.191	–
	$\nu^{(1)}_{\rm H\,I}$	0.3326	−0.355	–
	$\alpha^{(2)}_{\rm H\,I}$	−0.0000917	0.000499	–
	$\gamma^{(2)}_{\rm H\,I}$	0.8708	0.1705	–

Table A2 – *continued*

Quantity	Parameter	Layer 1	Layer 2	Output
	$k_{\mathrm{H\,I}}^{h\,(2)}$	2.059	1.281	–
	$k_{\mathrm{H\,I}}^{l\,(2)}$	7.312	7.957	–
	$\nu_{\mathrm{H\,I}}^{(2)}$	−0.497	0.3425	–
	$w_{\mathrm{H\,I}}$	–	–	1.011
	$\mu_{\mathrm{H\,I}}$	–	–	0.7689
	$b_{\mathrm{H\,I}}$	–	–	0.02597
	$\beta^{(1)}$	–	–	0.04662
	η	–	–	0.3443
	k_{I}^{h}	–	–	0.0
	k_{I}^{l}	–	–	8.394
	ν_{I}	–	–	−0.1218
	ξ	–	–	0.4362
	$\beta^{(2)}$	–	–	−3.117
	$\gamma_{\mathrm{H\,I}}^{(3)}$	–	–	0.7483

Table A3. Optimized JERALD parameters for dark matter, stellar mass and H I, at $z = 5$.

Quantity	Parameter	Layer 1	Layer 2	Layer 3	Output
Dark matter	α_{dm}	−0.01189	−0.04607	–	
	γ_{dm}	1.278	0.3793	–	–
	k_{dm}^{h}	0.0	0.0	–	–
	k_{dm}^{l}	0.6936	8.247	–	–
	ν_{dm}	0.7298	−0.9677	–	–
Stellar mass	α_{sm}	0.02964	−0.005002	−0.004058	–
	γ_{sm}	0.3977	0.8495	0.6279	–
	k_{sm}^{h}	0.5802	1.694	1.202	–
	k_{sm}^{l}	7.723	8.645	8.099	–
	ν_{sm}	−0.007005	0.1815	−0.01354	–
	w_{sm}	–	–	–	0.01492
	μ_{sm}	–	–	–	1.154
	b_{sm}	–	–	–	0.11
Neutral hydrogen	$\alpha_{\mathrm{H\,I}}^{(1)}$	−0.0006665	−0.0006563	–	–
	$\gamma_{\mathrm{H\,I}}^{(1)}$	0.04334	0.05415	–	–
	$k_{\mathrm{H\,I}}^{h\,(1)}$	0.00001	0.00001	–	–
	$k_{\mathrm{H\,I}}^{l\,(1)}$	7.847	7.845	–	–
	$\nu_{\mathrm{H\,I}}^{(1)}$	−0.1794	−0.1825	–	–
	$\alpha_{\mathrm{H\,I}}^{(2)}$	−0.0007138	−0.0007894	–	–
	$\gamma_{\mathrm{H\,I}}^{(2)}$	0.08366	0.08671	–	–
	$k_{\mathrm{H\,I}}^{h\,(2)}$	0.2034	0.12	–	–
	$k_{\mathrm{H\,I}}^{l\,(2)}$	6.914	6.693	–	–
	$\nu_{\mathrm{H\,I}}^{(2)}$	1.395	1.456	–	–
	$w_{\mathrm{H\,I}}$	–	–	–	1.17
	$\mu_{\mathrm{H\,I}}$	–	–	–	0.8938
	$b_{\mathrm{H\,I}}$	–	–	–	0.3218
	$\beta^{(1)}$	–	–	–	0.01011
	η	–	–	–	1.195
	k_{I}^{h}	–	–	–	0.0
	k_{I}^{l}	–	–	–	6.444
	ν_{I}	–	–	–	−0.1664
	ξ	–	–	–	−0.4002
	$\beta^{(2)}$	–	–	–	−1.2
	$\gamma_{\mathrm{H\,I}}^{(3)}$	–	–	–	0.6996

APPENDIX B: PARAMETER DEGENERACIES

The number of parameters used in this work, specified in Table 1, may seem excessive when compared with semi-analytical approaches (e.g. Barrera et al. 2023). Despite being a physics-based approach, JERALD also needs to correct for effects that do not necessarily have a direct physical interpretation, such as the coarse-graining in space due to the usage of a particle-mesh-based simulator and in time due to the integration scheme employed by FASTPM, which requires additional degrees of freedom. Additionally, while semi-analytical models derive expressions for different processes using a specific theoretical model, the goal of JERALD is to produce fields using a more general effective approach. None the less, it is possible that some parameters in the current model may be redundant: we observed

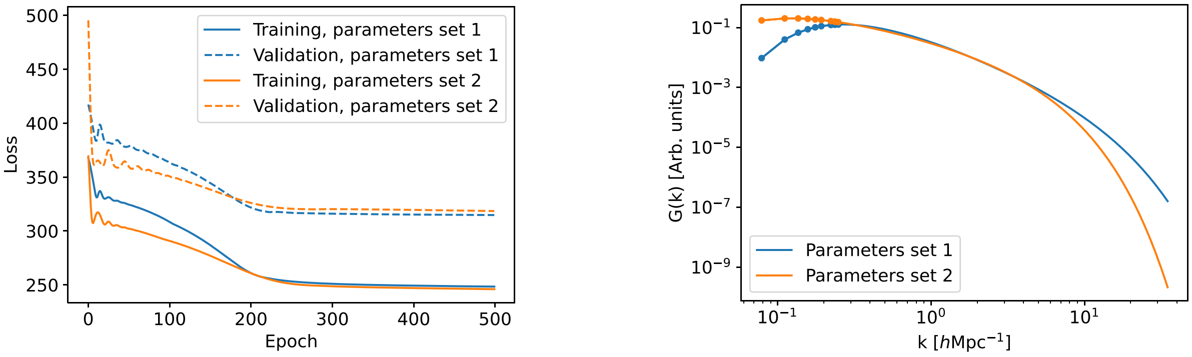

Figure B1. *Left:* training and validation losses for two DM optimizations as discussed in Section 4, but with a single displacement layer, for two different sets of initial parameters. *Right:* Green's function at convergence for the two set of initial parameters, illustrated in the range of frequencies of the associated maps. The dots on the top-left highlight the very few wavenumbers (10) for which the two curves do not overlap at large scales, whereas there are $\gtrsim 3\times10^5$ values of k above $\sim 5h\,\mathrm{Mpc}^{-1}$ for which the curves are significantly different.

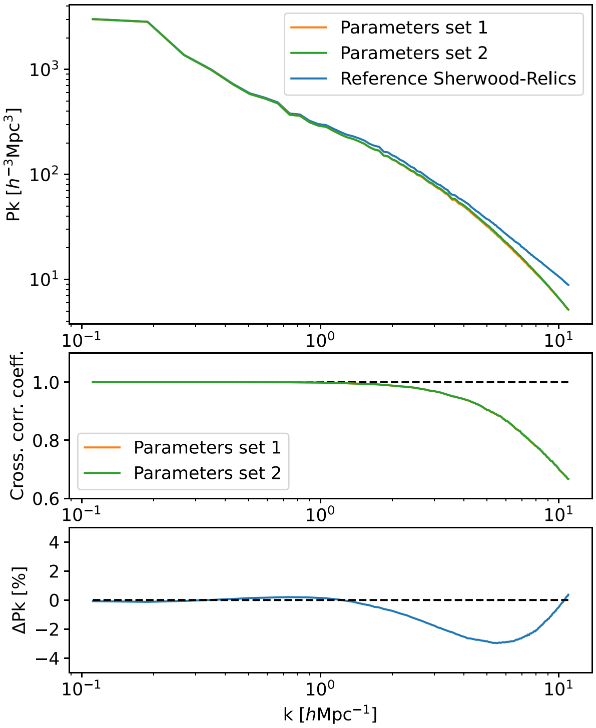

Figure B2. Final power spectra (top), cross-correlation coefficients (middle) and power spectra relative difference (bottom) for two DM optimizations as discussed in Section 4, but with a single displacement layer, for two different sets of initial parameters. The curves overlap within a few percent.

Table B1. Initial parameters and their values at convergence for two different optimizations in the case of DM discussed in Section 4, but with a single displacement layer.

	Initial values		At convergence	
	Set 1	Set 2	Set 1	Set 2
α_{dm}	0.001	0.002	0.06775	0.053
γ_{dm}	0.5	0.8	0.61536	0.62374
k^h_{dm}	1	0.5	0.58686	0.16695
k^l_{dm}	10	5	6.64337	2.37339
ν_{dm}	0	0	-2.17048	-1.31428

that different combinations of parameters can lead to very similar losses and performance in terms of summary statistics, suggesting the presence of multiple similar minima (or a global minimum which is rather flat along some degenerate directions), which we may be related to the chosen functional forms for the displacements or the final transformations, and it is conceivable that different expressions with fewer parameters could remove possible degeneracies while leading to comparable performance. For example, considering the displacement layers in the case of DM, equations (2) and (7), we observe that, when using only one layer, different initial parameters can lead to very similar optimized values for γ_{dm} but different ones for α_{dm}, k^h_{dm}, k^l_{dm} and ν_{dm}, although with very similar behaviours of $\tilde{G}_{\mathrm{dm}}(k)$ in the range of frequencies associated to the maps (generally, as one may expect, with greater differences at large frequencies $k \gtrsim 5h\,\mathrm{Mpc}^{-1}$). As an illustration of this observation, in Figs B1 and B2 we show the results of two optimizations starting from different initial parameters (reported in Table B1) and converging to different parameters (also reported in Table B1) for the DM case discussed in Section 4, but with a single displacement layer. Notice that, at convergence, both training and validation losses are within ~ 1 per cent between the two sets of parameters, and the γ_{dm} parameters of equation (2) are also within ~ 1 per cent.

Employing multiple displacement layers may complicate this even further. However, modifying these functional forms and reducing the number of parameters can heavily affect the optimization, e.g. by leading more easily to local minima. We did test many different functional forms – the ones reported in this work are those which led to the best results in terms of agreement in the power spectra and cross-correlation coefficients.

This paper has been typeset from a TEX/LATEX file prepared by the author.