

Neural Networks for Superresolving Cosmological Simulations

Samuel Grayson
grayson5

May 13, 2022

Introduction

Simulating cosmological structure has been of increased interest in the past decade. Simulating these structures can tell us about cosmological expansion [21], galaxy formation [33], and dark matter halos [15]. This leads us to a better understanding of cosmological parameters. However, the simulations need to be increasingly complex and detailed in order to make novel conclusions.

Simulating the behavior of an astrophysical system with a fine-granularity is computationally expensive. However, a technique known as **superresolution** may be able to reduce the cost by simulating the system at a coarse-granularity ‘big picture’ and using some other method to reconstruct the fine-grain ‘details.’ One could use superresolution to refine the resolution in space, time, or both, but I will use it for space.

Spatial superresolution was originally studied for videos. The hidden variable is the function from 2D-position to colors in the camera’s perspective. Although it exists continuously, it is only sampled/observed discretely at pixel points. Any movement between the subject and the observer in subsequent frames shifts the pixel grid slightly to yield a new sampling at *different* gridpoints, as in fig. 1. Features which may have fallen in between pixels in one frame may fall right on a pixel in another frame (because the camera or the subject moved slightly), so the feature’s existence in the first frame can be inferred. This is the ‘information theoretical’ basis for video superresolution given by Katsaggelos et al. [14].

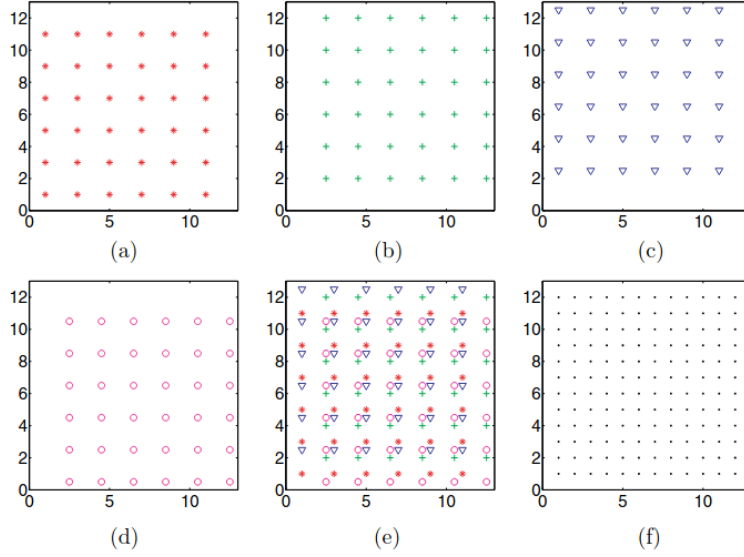


Figure 1: Low-resolution images (a), (b), (c), and (d), when overlaid, sample the underlying phenomenon at the points in (e). This set of low-resolution images can synthesize a higher-resolution image, (f), with classical superresolution.

However, in superresolution for simulations, the hidden variable is the high-granularity simulation, and it only exists discretely. There are no features in the high-granularity simulation to recover finer than the fine-grain gridpoints. This same theoretical justification does not apply, but perhaps another does: given the ergodic principle, every patch of space should have some patch in the training set that looks similar. In theory, superresolution can work by saving computational resources by learning what happens to small patches of space in the **training phase** and interpolating a novel patch based in the **prediction phase** (see fig. 2a.). While neural networks are universal function approximators, a neural network can be more computationally expensive than the function it was trained to approximate (in this case, the fine-grained fluid dynamic simulation); in order to be worthwhile, high-resolution inference (fig. 2b.) has to be cheaper than a bona fide high-resolution simulation (fig. 2c.).

Some [26, 19] have suggested using a generative adversarial neural network (GAN) for superresolution. GANs were first proposed by Ian Goodfellow in 2014 [10]. Suppose one wants to generate data in B based on a set of parameters A . A GAN consists of two networks: a generator $G : A \rightarrow B$ and a discriminator $D : A \times B \rightarrow [0, 1]$. To train a GAN, one needs genuine examples, $R \subseteq A \times B$. Intuitively, discriminator is trained to tell apart the genuine examples (elements

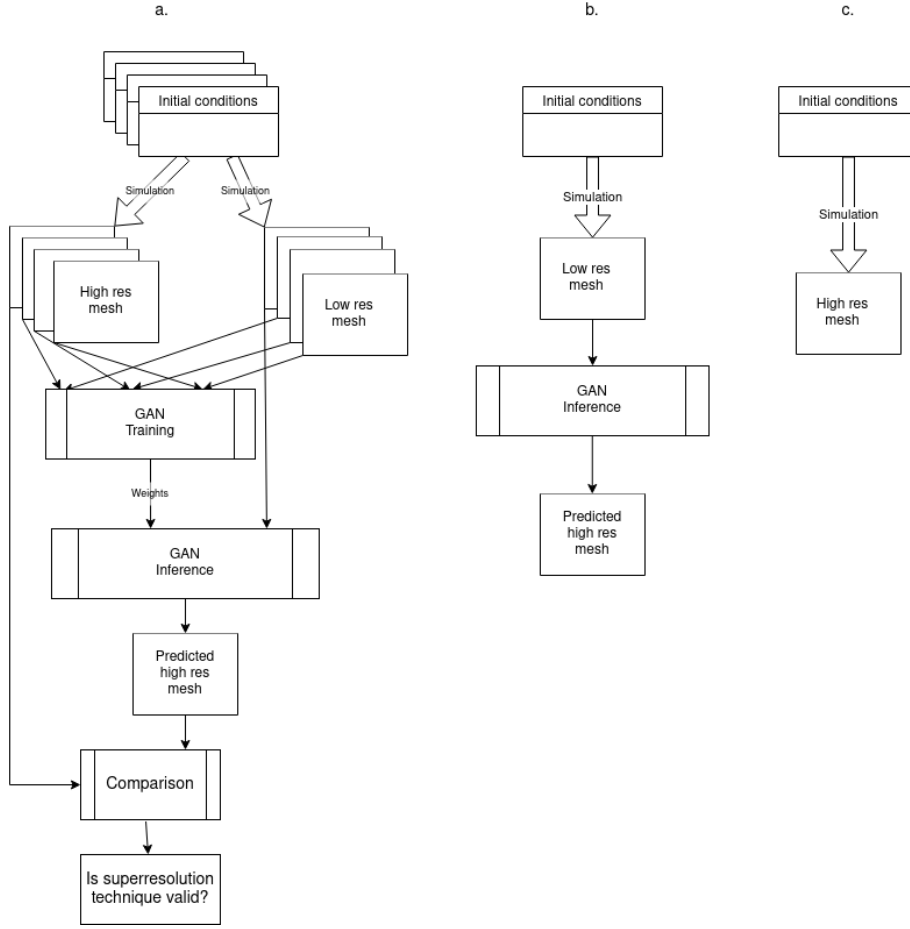


Figure 2: a. shows training and validation of the GAN; b. shows usage of the GAN; c. shows the process the GAN can replace.

of R), while the generator is trained to trick the discriminator with $(a, G(a))$ for $a \in A$. Let

$$L(G, D) = \mathbb{E}[\log D(a, b) | (a, b) \in R] + \mathbb{E}[\log(1 - D(G(a))) | a \in A]$$

where $\mathbb{E}(f(x) | x \in X)$ is the expected value of $f(x)$. Training runs in epochs where every even epoch updates the discriminator to maximize L , while every odd epoch updates the generator to minimize L , finding the $\min_G \max_D L(G, D)$.

A Nash Equilibrium for this system is guaranteed to exist, although training might not reach it in a reasonable time. See [35] for more background. Once training is done, one can throw away D and use G to map from A to B , hopefully

in a realistic way. In our case, A will be the low-resolution images, and B will be the high-resolution images.

Schaurecker et al. [26] apply a generative adversarial network to superresolution for cosmological structure in a dark-matter simulation called Illustris [22]. Rather than an *Eulerian* mesh, Illustris is based on a *Lagrangian* unstructured mesh (aka moving mesh) code called Arepo [31]. It is an open question whether Eulerian mesh simulators would also benefit from the same superresolution techniques. Neural networks have hyperparameters, such as number of layers, number of neurons, and network architecture, which can make or break the application in practice [6]. It is unclear if the hyperparameter choices in Schaurecker et al. for Illustris will transfer to a different simulator.

Schaurecker et al. [26] use a discriminator that progressively downsamples the input with convolutional layers. Convolutional layers are translationally invariant. For the generator, they use a U-net architecture introduced by [25] and adapted by [3, 9, 12]. The basic idea behind the U-net is to generate an image by first generating high-level structure, then adding one level of detail based on that structure, and then one more (see fig. 3). However, the authors replace the up-sampling convolution by a tri-linear interpolation.

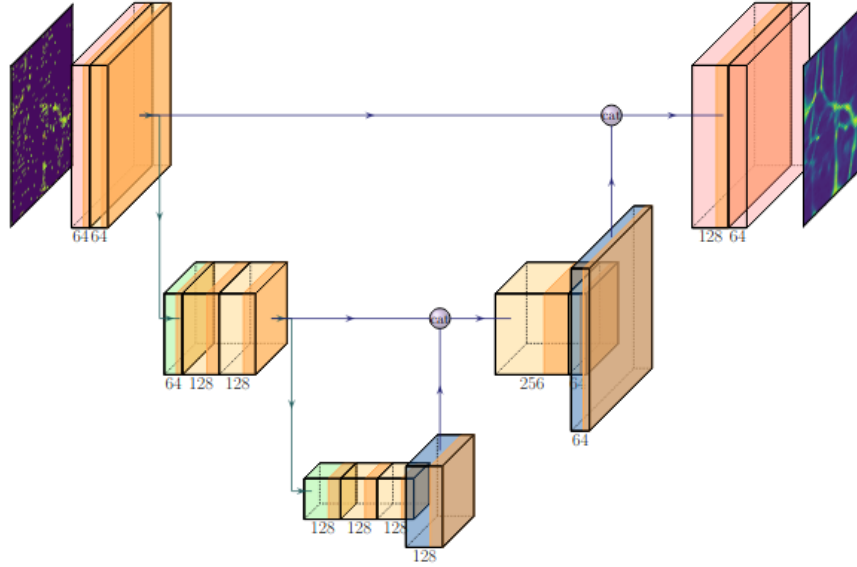


Figure 3: U-net architecture used for the generator in [26]. Note the three levels of hierarchy in generation.

Another hyperparameter in Schaurecker et al. is the objective function. Schau-

recker et al. uses a regularization term suggested by [20]

$$L(G, D) = \mathbb{E}[\log D(a, b) | (a, b) \in R] + \mathbb{E}[\log(1 - D(G(a))) | a \in A] + \gamma \mathbb{E}[\nabla D(a, b) | (a, b) \in R]$$

with $\gamma = 5$. It is unclear if this hyperparameter will transfer to a different space.

Illustris and AREPO

Illustris dataset[23, 34] uses the AREPO solver, which evolves the fluid quantities according to the “hyperbolic conservation laws of ideal hydrodynamics:”[30]

State vector	$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix}$
Flux	$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \mathbf{v}^T + P \\ (\rho E + P) \mathbf{v} \end{pmatrix}$
RHS	$\mathbf{W} = \begin{pmatrix} 0 \\ -\frac{\dot{a}}{a} \rho \mathbf{v} - \frac{\rho}{a^2} \nabla \phi \\ -2 \frac{\dot{a}}{a} \rho E - \frac{\rho \mathbf{v}}{a^2} \nabla \phi \end{pmatrix}$
Evolution with time	$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{a} \nabla \cdot \mathbf{F} = \mathbf{W}$
Equation of state	$P = (\gamma - 1)u$
Energy	$E = u + \frac{1}{2} \rho v^2$
Gravitational potential	$\nabla^2 \phi = \frac{4\pi G}{a} \rho_{\text{total}} - \rho_0$

where ρ is the density field, ρ_0 is the mean density field, \mathbf{v} is the velocity vector field, P is the pressure field, u is the internal energy field, a is the cosmological expansion constant, G is the gravitational constant, and γ is the ratio of specific heats, on an unstructured, moving, Voronoi tessellation mesh [30]. AREPO is second order in space and second order in time, due to hierarchical adaptive time-stepping [23].

AREPO uses Godunov’s method in a MUSCL-Hancock scheme to solve these equations [30]. The MUSCL-Hancock scheme is “a slope-limited piece-wise linear reconstruction step within each cell, a first-order prediction step for the evolution over half a time-step, and finally a Riemann solver to estimate the time-averaged inter-cell fluxes for the time-step”[16, 17]. Note that since the mesh is not cartesian, AREPO cannot employ operator splitting [32].

AREPO uses Tree-PM (particle mesh) approach for computing self-gravity. Long-range forces are calculated using the Fourier particle-mesh method while short-range forces are computed with a hierarchical tree algorithm [23].

The Illustris simulation further use Monte Carlo tracer particle scheme described in [8].

The Illustris initial conditions are set by running CAMB [29] using parameters found in the Wilkinson Anisotropy probe [13] or RECFast [28] (see Table 1).

Ω_m	0.2726
Ω_Λ	0.7274
Ω_b	0.0456
σ_8	0.809
n_s	0.963
H_0	100 h km / sec Mpc
h	0.704
$T(z = 157)$	245 K

Table 1: Initial conditions for CAMB.

Enzo

Enzo [24, 2] solves the magneto-hydrodynamics problem, in contrast to Illustris/AREPO which ignore magnetism. Enzo evolves the “equations of ideal magnetohydrodynamics (MHD) including gravity, in a coordinate systems co-moving with the cosmological expansion:”

State vector	$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix}$
Flux	$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \mathbf{v}^T + P + \frac{B^2}{2a} - \frac{\mathbf{B}\mathbf{B}}{a} \\ (\rho E + P + \frac{B^2}{2a}) \mathbf{v} - \frac{1}{a} \mathbf{B}(\mathbf{B} \cdot \mathbf{v}) \end{pmatrix}$
RHS	$\mathbf{W} = \begin{pmatrix} 0 \\ -\frac{\dot{a}}{a} \rho \mathbf{v} - \frac{1}{a} \rho \nabla \phi \\ -\frac{\dot{a}}{a} \left(2u - E - \frac{B^2}{2a} \right) - \frac{\rho}{a} \mathbf{v} \cdot \nabla \phi - \Lambda + \Gamma + \frac{1}{a^2} \nabla \cdot \mathbf{F}_{\text{cond}} \end{pmatrix}$
Evolution with time	$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{a} \nabla \cdot \mathbf{F} = \mathbf{W}$
Equation of state	$P = (\gamma - 1)u$
Energy	$E = u + \frac{1}{2} \rho \mathbf{v}^2 + \frac{B^2}{2a}$
Maxwell's Equations	$\frac{\partial \mathbf{B}}{\partial t} - \frac{1}{a} \nabla \times (\mathbf{v} \times \mathbf{B}) = 0$
Gravitational potential	$\nabla^2 \phi = \frac{4\pi G}{a} \rho_{\text{total}} - \rho_0$

where the variables are the same as in Illustris, but with magnetic vector field \mathbf{B} , radiative cooling Λ , radiative heating Γ , and thermal heat conduction \mathbf{F}_{cond} .

However, the situation I am simulating lacks a magnetic field anyway, so the extra magnetic capabilities won't matter. These equations are simulated on a structured grid with adaptive mesh refinement [2]. Enzo takes adaptive timesteps. The order of accuracy depends on the spatial solver used in Enzo.

For a spatial solver, Enzo can use:

1. the hydrodynamic-only piecewise parabolic method (PPM) [4, 1],
2. the MUSCL-like Godunov scheme [16, 17],
3. a constrained transport (CT) staggered MHD scheme [5],
4. the second-order finite difference hydro-dynamics method described in ZEUS [stone_zeus-2d_1992-1]

Enzo uses cloud-in-cell (CIC) at half-time steps and the Fourier method to compute the gravitational field.

Applicability

I am mostly not deciding which class of solvers to use, because I am wanting to reuse the neural network in [27] trained on Illustris data. Thus I am bound by the choices that Schaurecker et al. made in [27] and Nelson et al. in [23, 34]. One in particular, Schaurecker chose to simulate dark-matter only. This is a coarser understanding of the universe, but it will be easier to process in a neural network.

Intermediate Results

I am trying to use the same neural network architecture as Schaurecker et al. [27]. Schaurecker et al. have open sourced their code.¹

Recall my experimental setup summarized in fig. 4. I have written a script that can preform the green tasks in an automated fashion. I have partially implemented the yellow tasks. I have yet to implement the white tasks.

I believe that generating initial conditions with MUSIC [11] and simulating them with Enzo [5, 2] is working correctly based on the images in ?? . They

¹Source is available at https://github.com/dschaurecker/dl_halo/

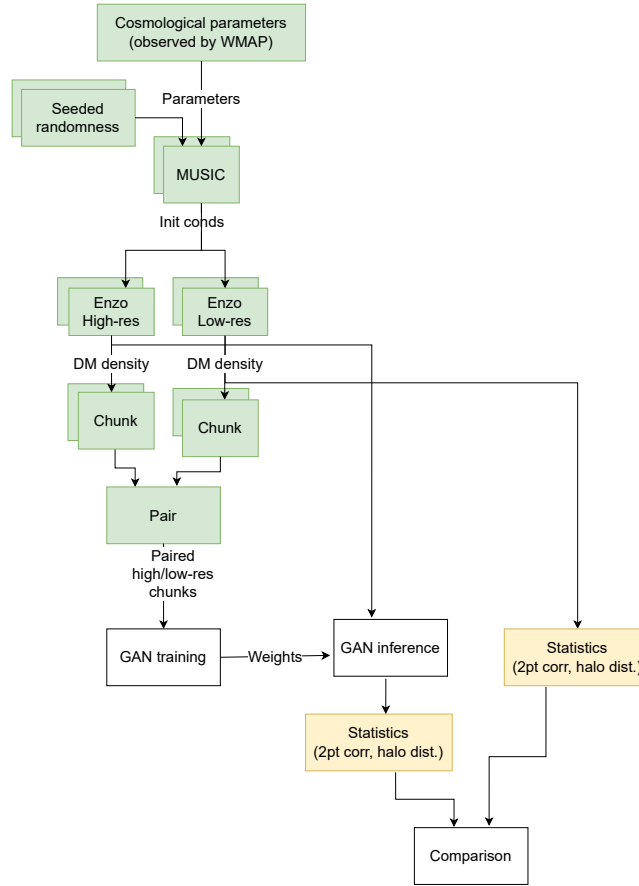


Figure 4: This figure shows the computational tasks in the experiment.

show dark matter initially smeared out evenly across the universe (left figure) at $z = 31$, 98 Myr after the big bang. Then dark matter condenses into filaments and halos at $z = 0$, present day. One can see the adaptive mesh refine in “interesting” areas. Also note that the simulation region has physically expanded due to Hubble expansion.

Problems

More work needs to be done to put the output of Enzo into a format that can be processed by the map2map² neural network library. Map2map requires that

²Source is available at <https://github.com/eelregit/map2map>

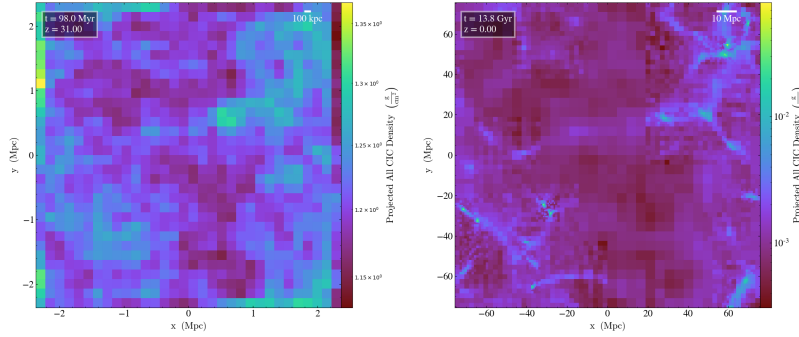


Figure 5: This plot shows the dark matter density integrated along the Z-axis at the beginning of the simulation (left) and at the end (right) in comoving coordinates.

the AMR data be sampled desnely and split into equal-sized rectangular chunks with padding.

I have been unable to install PyTorch on the campus cluster with CUDA. I have tried using the conda package manager, but it seems there is a diamond dependency conflict between the CUDA version of PyTorch and yt, another package I need to do this analysis.

Additionally, the FoaF halo finder requires the generated density field be resampled into individual particles. This is the opposite of CIC interpolation.

Discussion

Limitations of this approach

Most well studied neural net operates on a field sampled at a regular grid, not an AMR grid. As such, the AMR output of Enzo has to be interpolated into a fixed grid. This ends up losing the advantages that the AMR had in the first place. If one uses the finest granularity everywhere, the sampling uses an extravagant amount of memory; if one uses a coarser resolution, the sampling fails to capture all of the details. This is particularly bad for cosmological simulations where there are extremely intricate halos and filaments surrounded by massive voids. It would be ideal if this neural networks could work natively on AMR grids, but this is a less well-studied case. With regular grids, we can reuse

established building-blocks like the U-net and the concept of convolutional layers.

The prior literature [27, 18] do not define explicit criteria for accepting or rejecting their superresolution approximation. They compare the power spectra, halo mass function, and halo two-point correlation functions graphically. It is difficult to know if an approximation is going to be acceptable without knowing what downstream analysis it is used in.

Simulation Applicability

This approach to simulating cosmology relies on standard assumptions in Λ CDM. Following Schaurecker et al. [27], this simulation neglects baryonic contributions to gravity formation. According to the Wilkinson Anisotropy Probe, baryonic matter only accounts for 4 – 5% of the mass-energy in the universe, while dark matter and dark energy accounts for the rest [13]. Enzo assumes Newtonian gravity because the relativistic correction is small so long as the simulation size ($100 \text{ Mpc} \approx 3 \times 10^{15} \text{ meters}$) is small compared to $c/H = 1.4 \times 10^{26} \text{ meters}$ [2].

Conclusion

I successfully got Enzo working, and I learned a lot about the tools-of-the-trade for astrophysical simulation. More work can be done to integrate astrophysical toolkits like yt with neural network libraries like Keras and PyTorch. This would enable rapid development of neural network models for astrophysical simulation, which remains a promising direction.

Data Reproducibility

I have attempted to make my code as reproducible as possible.

1. All of my source code is available here³, with a full revision history stored in Git.

³<https://github.com/charmoniumQ/astrophysics-project>

2. I use the Spack package manager [7] to manage binary dependencies. Spack creates a ‘lockfile’ which contains an exact specification of the source code and instructions to compile for every package. One can replicate my environment with `spack env create myenv spack.lock`.
3. I use the Conda package manager to manage Python dependencies. Like Spack, this makes the software environment reproducible by others with `conda env create --name myenv --file environment.yaml`.
4. I wrote detailed instructions in the `README.md` to set up my software.
5. I combined the complete workflow (simulation parameters all the way to `grpahs`) into a single script. However, the script knows to skip a task if the data already exists. This way, there are fewer steps, and the user doesn’t need to manually run tasks or send the output of one task as the input to the next. Furthermore, the script can use ephemeral storage, such as `/scratch` on the Campus Cluster; If data gets deleted, it will just regenerate it.
6. I have each task as a function that gets called by the main script. This makes my code reusable in other tasks.
7. I wrote a library for running Slurm tasks on a remote machine and manipulating files on a remote machine. These can be reused in other tasks as well.

References

- [1] Greg L. Bryan et al. “A piecewise parabolic method for cosmological hydrodynamics”. In: *Computer Physics Communications* 89 (Aug. 1, 1995). ADS Bibcode: 1995CoPhC..89..149B, pp. 149–168. ISSN: 0010-4655. DOI: [10.1016/0010-4655\(94\)00191-4](https://doi.org/10.1016/0010-4655(94)00191-4). URL: <https://ui.adsabs.harvard.edu/abs/1995CoPhC..89..149B> (visited on 04/11/2022).
- [2] Greg L. Bryan et al. “Enzo: An Adaptive Mesh Refinement Code for Astrophysics”. In: *The Astrophysical Journal Supplement Series* 211.2 (Mar. 2014). Publisher: American Astronomical Society, p. 19. ISSN: 0067-0049. DOI: [10.1088/0067-0049/211/2/19](https://doi.org/10.1088/0067-0049/211/2/19). URL: <https://doi.org/10.1088/0067-0049/211/2/19> (visited on 04/11/2022).
- [3] Chang Chen et al. *Learning the Evolution of the Universe in N-body Simulations*. 2020. DOI: [10.48550/ARXIV.2012.05472](https://doi.org/10.48550/ARXIV.2012.05472). URL: <https://arxiv.org/abs/2012.05472>.

- [4] P. Colella and Paul R. Woodward. “The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations”. In: *Journal of Computational Physics* 54 (Sept. 1, 1984). ADS Bibcode: 1984JCoPh..54..174C, pp. 174–201. ISSN: 0021-9991. DOI: [10.1016/0021-9991\(84\)90143-8](https://ui.adsabs.harvard.edu/abs/1984JCoPh..54..174C). URL: <https://ui.adsabs.harvard.edu/abs/1984JCoPh..54..174C> (visited on 04/11/2022).
- [5] David C. Collins et al. “Cosmological Adaptive Mesh Refinement Magnetohydrodynamics with Enzo”. In: *The Astrophysical Journal Supplement Series* 186 (Feb. 1, 2010). ADS Bibcode: 2010ApJS..186..308C, pp. 308–333. ISSN: 0067-0049. DOI: [10.1088/0067-0049/186/2/308](https://ui.adsabs.harvard.edu/abs/2010ApJS..186..308C). URL: <https://ui.adsabs.harvard.edu/abs/2010ApJS..186..308C> (visited on 04/11/2022).
- [6] Matthias Feurer and Frank Hutter. “Hyperparameter optimization”. In: *Automated machine learning*. Springer, Cham, 2019, pp. 3–33.
- [7] Todd Gamblin et al. “The Spack package manager: bringing order to HPC software chaos”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC ’15. interest: 50. New York, NY, USA: Association for Computing Machinery, Nov. 15, 2015, pp. 1–12. ISBN: 978-1-4503-3723-6. DOI: [10.1145/2807591.2807623](https://doi.org/10.1145/2807591.2807623). URL: <https://doi.org/10.1145/2807591.2807623> (visited on 04/10/2022).
- [8] Shy Genel et al. “Following the flow: tracer particles in astrophysical fluid simulations”. In: *Monthly Notices of the Royal Astronomical Society* 435.2 (Oct. 21, 2013), pp. 1426–1442. ISSN: 0035-8711. DOI: [10.1093/mnras/stt1383](https://doi.org/10.1093/mnras/stt1383). URL: <https://doi.org/10.1093/mnras/stt1383> (visited on 04/11/2022).
- [9] Elena Giusarma et al. *Learning neutrino effects in Cosmology with Convolutional Neural Networks*. 2019. DOI: [10.48550/ARXIV.1910.04255](https://arxiv.org/abs/1910.04255). URL: <https://arxiv.org/abs/1910.04255>.
- [10] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- [11] Oliver Hahn and Tom Abel. “Multi-scale initial conditions for cosmological simulations”. In: *Monthly Notices of the Royal Astronomical Society* 415.3 (Aug. 11, 2011), pp. 2101–2121. ISSN: 00358711. DOI: [10.1111/j.1365-2966.2011.18820.x](https://doi.org/10.1111/j.1365-2966.2011.18820.x). arXiv: [1103.6031](https://arxiv.org/abs/1103.6031). URL: <https://doi.org/10.1111/j.1365-2966.2011.18820.x> (visited on 04/18/2022).

- [12] Siyu He et al. “Learning to predict the cosmological structure formation”. In: *Proceedings of the National Academy of Sciences* 116.28 (2019), pp. 13825–13832. DOI: [10.1073/pnas.1821458116](https://doi.org/10.1073/pnas.1821458116). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1821458116>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1821458116>.
- [13] G. Hinshaw et al. “Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Parameter Results”. In: *The Astrophysical Journal Supplement Series* 208 (Oct. 1, 2013). ADS Bibcode: 2013ApJS..208...19H, p. 19. ISSN: 0067-0049. DOI: [10.1088/0067-0049/208/2/19](https://doi.org/10.1088/0067-0049/208/2/19). URL: <https://ui.adsabs.harvard.edu/abs/2013ApJS..208...19H> (visited on 04/11/2022).
- [14] Aggelos K Katsaggelos, Rafael Molina, and Javier Mateos. “Super resolution of images and video”. In: *Synthesis Lectures on Image, Video, and Multimedia Processing* 1.1 (2007), pp. 1–134. DOI: <https://doi.org/10.2200/S00036ED1V01Y200606IVM007>. URL: <https://www.morganc laypool.com/doi/abs/10.2200/S00036ED1V01Y200606IVM007>.
- [15] Anatoly A Klypin, Sebastian Trujillo-Gomez, and Joel Primack. “Dark matter halos in the standard cosmological model: Results from the bolshoi simulation”. In: *The Astrophysical Journal* 740.2 (2011), p. 102.
- [16] Bram van Leer. “On the Relation Between the Upwind-Differencing Schemes of Godunov, Engquist–Osher and Roe”. In: *SIAM Journal on Scientific and Statistical Computing* 5.1 (Mar. 1984). Publisher: Society for Industrial and Applied Mathematics, pp. 1–20. ISSN: 0196-5204. DOI: [10.1137/0905001](https://doi.org/10.1137/0905001). URL: <https://epubs.siam.org/doi/10.1137/0905001> (visited on 04/11/2022).
- [17] Bram van Leer. “Upwind and High-Resolution Methods for Compressible Flow: From Donor Cell to Residual-Distribution Schemes”. In: *16th AIAA Computational Fluid Dynamics Conference*. Session: CFD-8: Thirty Years of CFD II. _eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2003-3559>. American Institute of Aeronautics and Astronautics, June 25, 2012. DOI: [10.2514/6.2003-3559](https://doi.org/10.2514/6.2003-3559). URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2003-3559> (visited on 04/11/2022).
- [18] Yin Li et al. “AI-assisted superresolution cosmological simulations”. In: *Proceedings of the National Academy of Sciences* 118.19 (May 11, 2021). Publisher: Proceedings of the National Academy of Sciences, e2022038118. DOI: [10.1073/pnas.2022038118](https://doi.org/10.1073/pnas.2022038118). URL: <https://www.pnas.org/doi/10.1073/pnas.2022038118> (visited on 05/04/2022).
- [19] Hongying Liu et al. “Video super resolution based on deep learning: A comprehensive survey”. In: *arXiv* (2020). URL: <https://arxiv.org/abs/2007.12928>.

- [20] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. “Which Training Methods for GANs do actually Converge?” In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3481–3490. URL: <https://proceedings.mlr.press/v80/mescheder18a.html>.
- [21] Ariadna Montiel et al. “Nonparametric reconstruction of the cosmic expansion with local regression smoothing and simulation extrapolation”. In: *Physical Review D* 89.4 (2014), p. 043007.
- [22] D. Nelson et al. “The Illustris simulation: Public data release”. In: *Astronomy and Computing* 13 (Sept. 2015), pp. 12–37. ISSN: 2213-1337. DOI: <https://doi.org/10.1016/j.ascom.2015.09.003>. URL: <https://www.sciencedirect.com/science/article/pii/S2213133715000864>.
- [23] D. Nelson et al. “The illustris simulation: Public data release”. In: *Astronomy and Computing* 13 (Nov. 1, 2015). interest: 70, pp. 12–37. ISSN: 2213-1337. DOI: [10.1016/j.ascom.2015.09.003](https://doi.org/10.1016/j.ascom.2015.09.003). URL: <https://www.sciencedirect.com/science/article/pii/S2213133715000864> (visited on 04/11/2022).
- [24] Brian W. O’Shea et al. “Introducing Enzo, an AMR Cosmology Application”. In: *arXiv e-prints* (Mar. 2004), astro-ph/0403044. URL: <https://ui.adsabs.harvard.edu/abs/2004astro.ph..3044O/abstract> (visited on 04/11/2022).
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nasir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.
- [26] David Schaurecker et al. *Super-resolving Dark Matter Halos using Generative Deep Learning*. 2021. eprint: [arXiv:2111.06393](https://arxiv.org/abs/2111.06393). URL: <https://arxiv.org/abs/2111.06393>.
- [27] David Schaurecker et al. “Super-resolving Dark Matter Halos using Generative Deep Learning”. In: *arXiv:2111.06393 [astro-ph]* (Nov. 11, 2021). arXiv: [2111.06393](https://arxiv.org/abs/2111.06393). URL: <http://arxiv.org/abs/2111.06393> (visited on 04/11/2022).
- [28] S. Seager, D. D. Sasselov, and D. Scott. “A New Calculation of the Recombination Epoch”. In: *The Astrophysical Journal* 523 (Sept. 1, 1999). ADS Bibcode: 1999ApJ...523L...1S, pp. L1–L5. ISSN: 0004-637X. DOI: [10.1086/312250](https://doi.org/10.1086/312250). URL: <https://ui.adsabs.harvard.edu/abs/1999ApJ...523L...1S> (visited on 04/11/2022).

- [29] Uros Seljak and Matias Zaldarriaga. “A Line-of-Sight Integration Approach to Cosmic Microwave Background Anisotropies”. In: *The Astrophysical Journal* 469 (Oct. 1, 1996). ADS Bibcode: 1996ApJ...469..437S, p. 437. ISSN: 0004-637X. DOI: [10.1086/177793](https://doi.org/10.1086/177793). URL: <https://ui.adsabs.harvard.edu/abs/1996ApJ...469..437S> (visited on 04/11/2022).
- [30] Volker Springel. “E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh”. In: *Monthly Notices of the Royal Astronomical Society* 401.2 (Jan. 11, 2010), pp. 791–851. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2009.15715.x](https://doi.org/10.1111/j.1365-2966.2009.15715.x). URL: <https://doi.org/10.1111/j.1365-2966.2009.15715.x> (visited on 04/11/2022).
- [31] Volker Springel. “Moving-mesh hydrodynamics with the AREPO code”. In: *Proceedings of the International Astronomical Union* 6.S270 (2010), pp. 203–206. DOI: [10.1017/S1743921311000378](https://doi.org/10.1017/S1743921311000378).
- [32] Gilbert Strang. “On the Construction and Comparison of Difference Schemes”. In: *SIAM Journal on Numerical Analysis* 5.3 (Sept. 1968). Publisher: Society for Industrial and Applied Mathematics, pp. 506–517. ISSN: 0036-1429. DOI: [10.1137/0705041](https://doi.org/10.1137/0705041). URL: <https://epubs.siam.org/doi/10.1137/0705041> (visited on 04/11/2022).
- [33] Mark Vogelsberger et al. “Cosmological simulations of galaxy formation”. In: *Nature Reviews Physics* 2.1 (2020), pp. 42–66.
- [34] Mark Vogelsberger et al. “Introducing the Illustris Project: simulating the coevolution of dark and visible matter in the Universe”. In: *Monthly Notices of the Royal Astronomical Society* 444.2 (Oct. 21, 2014), pp. 1518–1547. ISSN: 0035-8711. DOI: [10.1093/mnras/stu1536](https://doi.org/10.1093/mnras/stu1536). URL: <https://doi.org/10.1093/mnras/stu1536> (visited on 04/11/2022).
- [35] Thomas Wood. “Generative Adversarial Network”. In: *DeepAI* (). URL: <https://deepai.org/machine-learning-glossary-and-terms/generative-adversarial-network>.