Final Results

Samuel Grayson
grayson5

May 4, 2022

1 Status of Experiments

I am trying to use the same neural network architecture as Schaurecker et al.[7]. Schaurecker et al. have open sourced their code. [1]

Recall my experimental setup summarized in fig. 1. 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.

2 Problems

More work needs to be done to put the output of Enzo into a format that can be processed by the map2map[2] neural network library. Map2map requires that 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.

[1] Source is available at https://github.com/dschaurecker/dl_halo/
[2] Source is available at https://github.com/eelregit/map2map

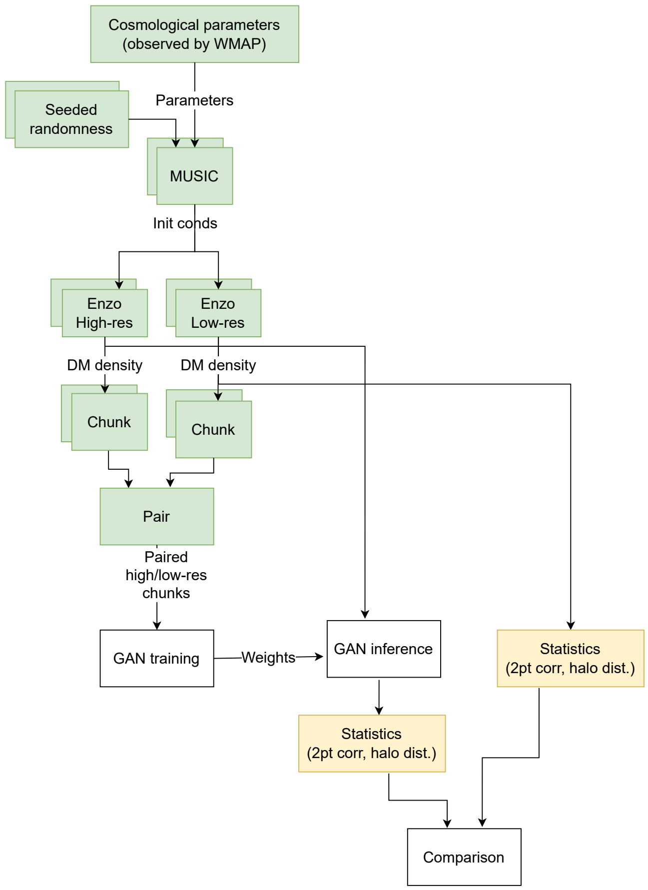

Figure 1: This figure shows the computational tasks in the experiment.

3 Simulation Output

I believe that generating initial conditions with MUSIC [4] and simulating them with Enzo [2, 1] is working correctly based on the images in section 3. They 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.

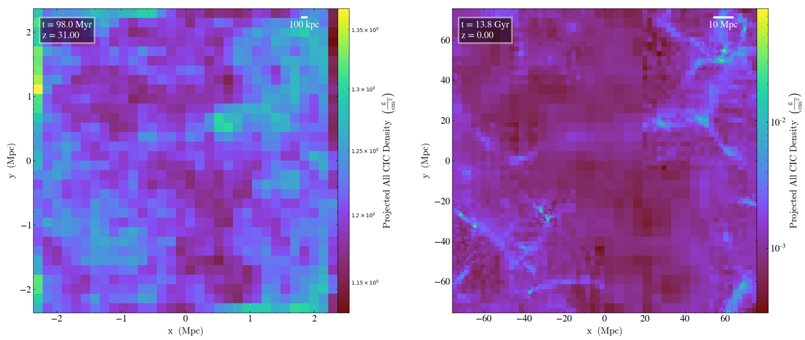

Figure 2: This plot shows the dark matter density integrated along the Z-axis at the beginning of the simulation (left) and at the end (right) in comoving coordinates.

4 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 [7, 6] 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.

5 Simulation Applicability

This approach to simulating cosmology relies on standard assumptions in ΛCDM. Following Schaurecker et al. [7], 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 [5]. Enzo assumes Newtonian gravity because the relativistic correction is small so long as the simulation size (100 Mpc 3×10^{15} meters) is small compared to $c/H = 1.4 \times 10^{26}$ meters) [1].

6 Reproducibility

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

1. All of my source code is available here[3], with a full revision history stored in Git.
2. I use the Spack package manager [3] 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 of 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 is 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 resued in other tasks as well.

[3] https://github.com/charmoniumQ/astrophysics-project

References

[1] 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. URL: https://doi.org/10.1088/0067-0049/211/2/19 (visited on 04/11/2022).

[2] 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. URL: https://ui.adsabs.harvard.edu/abs/2010ApJS..186..308C (visited on 04/11/2022).

[3] 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. URL: https://doi.org/10.1145/2807591.2807623 (visited on 04/10/2022).

[4] 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. arXiv: 1103.6031. URL: https://doi.org/10.1111/j.1365-2966.2011.18820.x (visited on 04/18/2022).

[5] 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. URL: https://ui.adsabs.harvard.edu/abs/2013ApJS..208...19H (visited on 04/11/2022).

[6] 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. URL: https://www.pnas.org/doi/10.1073/pnas.2022038118 (visited on 05/04/2022).

[7] 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. URL: http://arxiv.org/abs/2111.06393 (visited on 04/11/2022).