A Quasi-Universal Neural Network to Model Structure Formation in the Universe

Neerav Kaushal
Department of Physics
Michigan Technological University
Houghton, MI 49931
kaushal@mtu.edu

Francisco Villaescusa-Navarro
Department of Astrophysical Sciences
Princeton University
Princeton, NJ 08544
fvillaescusa@princeton.edu

Elena Giusarma
Department of Physics
Michigan Technological University
Houghton, MI 49931
egiusarm@mtu.edu

Yin Li
Center for Computational Astrophysics
& Center for Computational Mathematics
Flatiron Institute, New York, NY 10010
yinli@flatironinstitute.org

Mauricio Reyes
Department of Physics
Michigan Technological University
Houghton, MI 49931
mrhurtad@mtu.edu

Abstract

The large-scale structure of the Universe is the direct consequence of its evolution over billions of years. The observations of this large-scale structure in terms of galaxy redshift surveys contain valuable cosmological information and in order to extract that information, we need to compare these observations to corresponding theory predictions from cosmological simulations, whose generation in itself is a very computationally intensive feat. This work uses deep convolutional neural networks to simulate the large-scale structure of the Universe and generate a typical cosmological simulation orders of magnitude faster than the standard N-body simulations within an accuracy of $\sim 1\%$ on the most common cosmological summary statistics. The most important feature of our model is that it extrapolates extremely well on universes with entirely different cosmologies than the one it has been trained on. The use of such an approach will be particularly useful in the near future to compare theory with predictions, to generate mock galaxy catalogs, to compute covariance matrices, and to optimize observational strategies.

1 Introduction

Cosmology and Astrophysics are in constant need of accurate theoretical predictions to compare with the state-of-the-art observations of numerous current and upcoming galaxy surveys. In the absence of analytical methods for computing quantities of interest, cosmological simulations are the only tool that provides the most rigorous theoretical predictions of the evolution and structure formation in the Universe. The traditional N-body simulations of the Universe are very accurate but are computationally expensive to generate. On the other hand, the fast approximations to the N-body simulations are computationally inexpensive but compromise accuracy on nonlinear scales. So, we

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

need tools that can overcome these limitations and generate fast as well as accurate simulations for cosmological analyses.

In this work, we have developed a deep learning-based Convolutional Neural Network (CNN) model capable of mapping from Fast Approximations to N-body simulations of the Universe. Specifically, we have shown that our model

- is able to generate simulated universes within an accuracy of $\sim 1\%$ on commonly employed cosmological summary statistics such as power spectrum and bispectrum down to scales as small as $k = 1\ h\mathrm{Mpc}^{-1}$.
- is at least 4 orders of magnitude faster than the N-body simulations.
- extrapolates extremely well on very different cosmologies, outperforming the previous state-of-the-art model [1].

2 Methods

N-body simulations are a suite of cosmological simulations in which the Cold Dark Matter (CDM) particles are evolved under the effect of gravity alone. The simulation starts with particles only slightly perturbed from a uniform grid using the Lagrangian Perturbation Theory (LPT). During the simulation, a particle moves from its initial (Lagrangian) position x_i to its final position $x_f = x_i + \Delta(x_i)$, where the displacement vector Δ is a function of the initial positions. A typical N-body simulation takes thousands of timesteps to solve the dynamics of billions of particles, rendering them computationally expensive. Unlike the N-body methods, fast approximation methods integrate only tens of timesteps to generate relatively less accurate simulations. In this work, we use the Lagrangian Perturbation Theory to generate the fast approximations of the Universe using the COmoving Lagrangian Acceleration (COLA) [7] method. COLA decouples the large and small scales of the Universe and evolves them separately using second-order Lagrangian Perturbation Theory (2LPT) [2] and N-body methods respectively. It utilizes the fact that the large-scales are well-described using LPT [8] and that the time integration for the large scales in N-body codes simply solves for the linear growth factor whose exact value is easily available in any standard textbook [3]. This allows us to take large N-body timesteps and save a lot of computations, and at the same time keep the accuracy on the largest scales. In addition, the Zel'dovich simulation is a significantly faster approximation to the N-body simulation produced by first-order perturbation theory.

2.1 Input, Target and Benchmark

The displacement field of a set of particles is given by $\Delta = \vec{x_f} - \vec{x_i}$, where $\vec{x_f}$ are the final positions of the particles at redshift 0, which corresponds to the current epoch of the Universe, and $\vec{x_i}$ are the initial (Lagrangian) positions of the same particles on a uniform grid. We build a V-Net [5] based CNN that maps from the COLA displacement field (input) to the N-body displacement field (target) by training on their residual ($\Delta_{\mathrm{Nbody}} - \Delta_{\mathrm{COLA}}$).

We use 100 N-body simulations with a fiducial cosmology from the publicly available Quijote [9] suite that are run in a periodic box of length 1000 Mpc h^{-1} and follow the evolution of 512^3 CDM particles from $z = 127$ to $z = 0$. The COLA simulations are run with the publicly available MG-PICOLA [10] package for 30 timesteps with the same number of particles, parameter configuration, and random seeds as the N-body simulations to ensure the same initial conditions for both.

In order to compare the predictions of our model, NN(COLA), we have used three kinds of benchmarks: **(1) COLA**, which represents the results of running the COLA simulation itself, **(2) ZA**, where the positions of the particles at $z = 0$ are computed using the Zel'dovich approximation, and **(3) NN(ZA)**, our model trained on ZA simulations.

2.2 Model

We use a V-Net [5] based model, inspired by Alves de Oliveira et al. [1] that consists of 2 downsampling and 2 upsampling layers connected in a "V" shape. Blocks of two 3^3 convolutions connect the input, the resampling, and the output layers. 1^3 convolutions are added over each of these convolution blocks to realize a residual connection. We add batch normalization after every convolution except the first one and the last two, and leaky ReLU activation with negative slope 0.01 after every batch normalization, as well as the first and the second to last convolutions. The last activation in each residual block acts after the summation, following Milletari et al. [5]. As in U-Net/V-Net, at all except the bottom resolution levels, the inputs to the downsampling layers are concatenated to the outputs of the upsampling layers. All layers have a channel size of 64, except for the input and the output, that have 3 channels, as well as those after concatenations (128-channeled). Finally, the input $(\Delta_{\rm COLA})$ is directly added to the output, so that the network could learn the corrections to match the target $(\Delta_{\rm Nbody})$. Stride-2 2^3 convolutions and stride-$1/2$ 2^3 transposed convolutions are used in downsampling and upsampling layers, respectively.

Following Alves de Oliveira et al. [1], we composed a loss function given by $L = \log(\mathrm{L}_\delta \mathrm{L}_\Delta^\lambda)$, where L_δ is the Mean Squared Error (MSE) loss on $n(\mathbf{x})$, the particle number in voxel **x** and L_Δ is the MSE on Δ, the displacement field. By combining the two losses with logarithm rather than summation, we can ignore their absolute magnitudes and trade between their relative values. λ here serves as a weight on this trade-off of relative losses and we have used $\lambda = 1$ in this work.

The input, owing to the big size of the data (3×512^3) is cropped into smaller subcubes of size 3×128^3, corresponding to a length of 250 Mpc h^{-1}. In order to preserve the physical translational equivariance, no padding has been used in the 3^3 convolutions, which results in an output that is smaller than the input in spatial size. This limitation is compensated by padding the input cubes periodically with 20 voxels on each side, so that the effective spatial size of the input becomes 3×164^3. Furthermore, data augmentation is implemented to enforce the equivariance of displacement fields under rotational and parity transformations. We use the Adam optimizer [4] with a learning rate of 0.0001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and reduce the learning rate by half when the loss does not improve for 3 epochs. The model is trained on 70 realizations for 100 epochs and the remaining realizations are used for validation (20) and final testing (10).

3 Results

Figure 1 shows the predictions of our model, NN(COLA), on the test set of 10 realizations using the most commonly employed statistics in cosmology, the power spectrum, $P(k)$ (top left) and the bispectrum, $B(k)$ (top right). The power spectrum quantifies the correlation of density fluctuations as a function of scale (the wavenumber k denotes the scale, with low/high k representing large/small scales), and the bispectrum quantifies correlations in closed triangles in fourier space. For the bispectrum, we show the triangle configuration with $k_1 = 0.15\ h\mathrm{Mpc}^{-1}$ and $k_2 = 0.25\ h\mathrm{Mpc}^{-1}$ as a function of the angle θ between k_1 and k_2.

The transfer function, $T(k)$ (middle left), is the square root of the ratio of the predicted power spectra and the target power spectra. r_δ is the cross-correlation coefficient that quantifies the correlation between the phases of different fourier modes and $1 - r_\delta^2$ (bottom left) gives the amount of unexplained variance between the predicted and the target fields. A $T(k) = 1$ and $1 - r_\delta^2 = 0$ signify a perfect emulation of the N-body field by the model. The target (N-body simulations) and the primary benchmark, COLA, are shown with solid black and blue dotted curves respectively. NN(COLA) (red dashed line) shows the predictions of our model with COLA as input. In order to see how Zeldovich (ZA) approximations compare to the COLA simulations as the model input, we have also performed a standard ZA to N-body mapping with our model and NN(ZA) (yellow solid line) refers to the predictions of our model with ZA as input. Our model when trained on COLA simulations (NN(COLA)), outperforms the benchmark simulations (COLA) as well as its predictions when it is trained on ZA approximations (NN(ZA)), producing percent-level accurate results down to scales as small as $k \sim 1\ h\mathrm{Mpc}^{-1}$ and establishing COLA as a much better choice for model training than the ZA approximations.

A typical N-body simulation takes roughly 500 CPU hours to run, or $\sim 10^6$ CPU seconds, while a single COLA simulation takes around 3 CPU hours or $\sim 10^4$ CPU seconds (on an 408 Intel Skylake). Our model, on the other hand, takes $\sim$ 125 GPU-seconds to run on a single GPU (320 NVIDIA P100-16GB) using the PyTorch [6] framework. A runtime comparison of the target, benchmark, and our model is shown in Table 1. Thus, in practice, the main limitation of our model comes from the computational cost associated with running COLA simulations itself. Despite this, our model allows us to speed up the computational cost by a factor of 100.

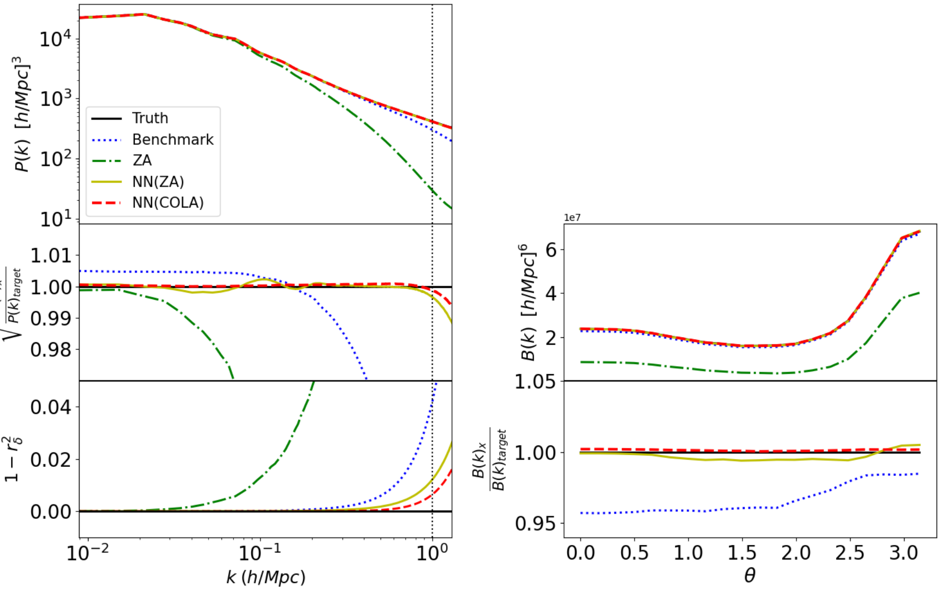

Figure 1: The left plot shows the 3D matter power spectrum (top), the transfer function (middle) and the cross-correlation coefficient (bottom), while the right plot shows the bispectrum (top) and the bispectrum ratio (bottom) for the target N-body simulations (black solid), the primary benchmark cola simulations (blue dotted), the ZA approximations (green dashed-dotted), and the model predictions with ZA as input (yellow solid) and COLA as input (red dashed). Our model (red dashed line) outperforms the benchmark (blue dotted) in all cases.

Table 1: Runtime benchmark

Simulation	N-Body (QUIJOTE)	Fast (COLA)	Model (GPU)
CPU-/GPU-sec	10^6	10^4	125

3.1 Model Extrapolation

For all the model training and testing so far, we have used simulations with a fixed value of cosmological parameters ($n_s = 0.9624, \sigma_8 = 0.834, h = 0.6711, \Omega_m = 0.3175, \Omega_b = 0.049$). Different choices of these parameters change the large-scale structure of the Universe. In order to further test the performance and robustness of our model, we test it on a set of 100 simulations with different cosmologies spanning the range $\Omega_\mathrm{m} \in [0.1, 0.5]$, $\Omega_\mathrm{b} \in [0.03, 0.07]$, $h \in [0.5, 0.9]$, $n_s \in [0.8, 1.2]$ and $\sigma_8 \in [0.6, 1.0]$. All the five cosmological parameters in these simulations (as well as the random seed) are varied together so that no pair of universes has a single identical parameter. This means that the structure formation in these universes proceeds in entirely different ways than the ones used in the training, thus providing very diverse displacement fields to test the model. In Figure 2, we compare our model predictions (right) to the model by Alves de Oliveira et al. [1] (left) which maps from the ZA approximations to the N-body simulations, denoted by NN(ZA)$_{dO}$. We find that our model captures the variations between different universes with surprising accuracy: below $\simeq 1\%$ down to $k = 1\ h\mathrm{Mpc}^{-1}$ and outperforms the extrapolations predicted by the state-of-the-art model [1]. From a computational viewpoint, this also suggests that our model is capable of generating simulations for a diverse range of cosmological parameters, with minimal training data and can be deployed to generate simulations from a wide range of parameters beyond the parameter space covered by the training data.

4 Conclusions

In this work, we have shown that neural networks can efficiently and accurately emulate the computationally expensive N-body simulations. By computing a variety of summary statistics, we show

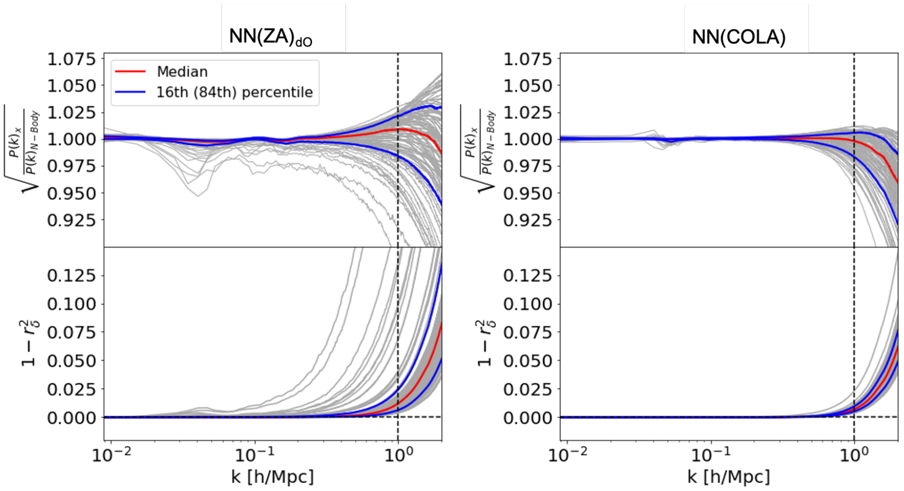

Figure 2: The figure shows the transfer function (top) and the cross-correlation coefficient (bottom) for 100 simulations with different cosmologies using the model by Alves de Oliveira et al. [1] (left) and our model (right). The red lines represent the median while the blue lines represent the 16th (and 84th) percentile of the predictions. Our model outperforms the model by Alves de Oliveira et al. [1] and also does a relatively better job of smoothing out the baryon acoustic oscillations (the bump around $k = 0.04\ h\mathrm{Mpc}^{-1}$) over the 100 realizations used.

that *our model perfectly reproduces the N-body simulations down to highly nonlinear scales with* $k = 1\ h\mathrm{Mpc}^{-1}$ *with an accuracy of* $\sim 1\%$, and outperforms the benchmark COLA simulations. Furthermore, *our model generalizes very well to the universes with different cosmologies on which it is never trained, with an accuracy of* $\sim 1\%$ *on power spectrum and cross-correlation coefficient*, all the while outperforming the state-of-the-art emulators. Our approach also renders the time needed to generate a typical cosmological simulation *four orders of magnitude less* than a traditional N-body simulation.

Broader Impact

In the era of multi-billion dollar cosmological surveys mapping the entire sky and generating heaps of observational data, it is indispensable to develop faster and accurate tools to generate corresponding theoretical predictions. Our work provides one such tool by using deep neural networks to emulate the Universe as accurately and as generally as possible. It also speeds up the generation of computationally expensive numerical simulations by at least two orders of magnitude. Our model captures most of the non-linear astrophysics involved in the process of structure formation at smaller scales of the Universe very well. This in turn, increases the scientific return of these billion dollar projects in time as well as in accuracy.

Data availability

The trained models, predictions and statistics extracted from the testing are hosted under the public github repository `cola-to-nbody`[1] and the model training has been performed with the `map2map`[2] code. The N-body data has been taken from the `Quijote-simulations`[3] while the COLA simulations have been generated using the `MG-PICOLA-PUBLIC` [4] code. `Quijote-simulations` is publicly available under an MIT license and the other three are available under a GNU General Public License.

[1] https://github.com/neeravkaushal/cola-to-nbody.git

[2] https://github.com/eelregit/map2map.git

[3] https://github.com/franciscovillaescusa/Quijote-simulations.git

[4] https://github.com/HAWinther/MG-PICOLA-PUBLIC.git

Acknowledgements

We acknowledge that our work has been performed using the Princeton Research Computing resources at Princeton University which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE).

References

[1] Renan Alves de Oliveira, Yin Li, Francisco Villaescusa-Navarro, Shirley Ho, and David N. Spergel. Fast and Accurate Non-Linear Predictions of Universes with Deep Learning. *arXiv e-prints*, art. arXiv:2012.00240, November 2020.

[2] F. Bernardeau, S. Colombi, E. Gaztañaga, and R. Scoccimarro. Large-scale structure of the universe and cosmological perturbation theory. *Physics Reports*, 367(1-3):1–248, Sep 2002. ISSN 0370-1573. doi: 10.1016/s0370-1573(02)00135-7. URL `http://dx.doi.org/10.1016/S0370-1573(02)00135-7`.

[3] E., Peebles P J. *The large-scale structure of the universe*. Princeton University Press, 2020.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[5] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation, 2016.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[7] Svetlin Tassev, Matias Zaldarriaga, and Daniel J Eisenstein. Solving large scale structure in ten easy steps with COLA. *Journal of Cosmology and Astroparticle Physics*, 2013(06):036–036, jun 2013. doi: 10.1088/1475-7516/2013/06/036. URL `https://doi.org/10.1088`.

[8] Svetlin V. Tassev and M. Zaldarriaga. The Mildly Non-Linear Regime of Structure Formation. In *American Astronomical Society Meeting Abstracts #220*, volume 220 of *American Astronomical Society Meeting Abstracts*, page 524.23, May 2012.

[9] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Ana Maria Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, Antoine Brochard, Cora Uhlemann, Chi-Ting Chiang, Siyu He, Alice Pisani, Andrej Obuljen, Yu Feng, Emanuele Castorina, Gabriella Contardo, Christina D. Kreisch, Andrina Nicola, Justin Alsing, Roman Scoccimarro, Licia Verde, Matteo Viel, Shirley Ho, Stephane Mallat, Benjamin Wandelt, and David N. Spergel. The quijote simulations. *The Astrophysical Journal Supplement Series*, 250(1):2, aug 2020. doi: 10.3847/1538-4365/ab9d82. URL `https://doi.org/10.3847/1538-4365/ab9d82`.

[10] Bill S. Wright, Hans A. Winther, and Kazuya Koyama. COLA with massive neutrinos. , 2017 (10):054, October 2017. doi: 10.1088/1475-7516/2017/10/054.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the bullet points in Section 1 and the Conclusions.
   (b) Did you describe the limitations of your work? [No]

(c) Did you discuss any potential negative societal impacts of your work? [No] Not relevant to our work

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A] The work deals with developing a model that could emulate the predictions of an existing theory and as such, does not need an explicit set of assumptions.

   (b) Did you include complete proofs of all theoretical results? [N/A] Following above, not applicable to this work.

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and data used for this work has been provided in the Data Availability section.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 2.2, last paragraph (lines 86-90)

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Percentiles have been included instead of error bars as they are more relevant to the work. See Figure 2.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3, Paragraph 2

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] References 1 and 7

   (b) Did you mention the license of the assets? [Yes] See the Data Availability section.

   (c) Did you include any new assets either in the supplemental material or as a URL? [No] The ones used have already been cited and provided in the Data Availability section.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The data and codes are publicly available (license specified).

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]