Fast and Accurate Non-Linear Predictions of Universes with Deep Learning

Renan Alves de Oliveira
PPGCosmo/Flatiron Institute
fisica.renan@gmail.com

Yin Li
Flatiron Institute
yinli@flatironinstitute.org

Francisco Villaescusa-Navarro
Princeton University
villaescusa.francisco@gmail.com

Shirley Ho
Flatiron Institute
shirleysurelyho@gmail.com

David N. Spergel
Flatiron Institute
dspergel@flatironinstitute.org

Abstract

Cosmologists aim to model the evolution of initially low amplitude Gaussian density fluctuations into the highly non-linear "cosmic web" of galaxies and clusters. They aim to compare simulations of this structure formation process with observations of large-scale structure traced by galaxies and infer the properties of the dark energy and dark matter that make up 95% of the universe. These ensembles of simulations of billions of galaxies are computationally demanding, so that more efficient approaches to tracing the non-linear growth of structure are needed. We build a V-Net based model that transforms fast linear predictions into fully non-linear predictions from numerical simulations. Our NN model learns to emulate the simulations down to small scales and is both faster and more accurate than the current state-of-the-art approximate methods. It also achieves comparable accuracy when tested on universes of significantly different cosmological parameters from the one used in training. This suggests that our model generalizes well beyond our training set.

1 Introduction

Cosmology is at critical stage: the 2019 Nobel Prize recognized that a simple model with only five basic parameters fits a host of astronomical observations on large-scales where the fluctuations are in the linear regime. This model implies that atoms make up only 5% of the universe with the remaining 95% in the form of dark matter and dark energy. Astronomers have embarked on several large-scale surveys and are launching multiple satellites that aim to collect data to understand these mysteries. Analysis of this new data is limited by our computational abilities as numerical simulations are essential for comparing theoretical predictions to observations on the small non-linear scales that contain most of the cosmological information [1, 2, 3, 4, 5]. Unfortunately, the computational cost of these simulations is high; a single simulation may require from thousands to millions of CPU hours.

In this work, we aim to build a neural network (NN) model that can *learn* to emulate these numerical simulations both accurately and efficiently. This paper builds on the work of He et al. [6], which showed that neural networks can indeed learn to run *fast approximate* simulations. Our main

contributions is improvement of the model by training on *full N-body* simulations and extending to much smaller scales, where the learning task becomes more difficult due to nonlinearity.

2   Methods

N-body Simulation.   An N-body simulation evolves a large number of massive dark matter particles interacting with each other only through Newtonian gravity. Since our Universe began with matter distributed almost uniformly, the simulation starts with particles only slightly perturbed from a uniform grid and nearly at rest. During the simulation, a particle moves from its initial location $\mathbf{q}$ to its final position $\mathbf{x} = \mathbf{q} + \mathbf{\Psi}(\mathbf{q})$, where $\mathbf{\Psi}$ is the displacement vector. N-body simulations usually end at our current time, i.e. "today"; this is the time at which we compare them to our ML predictions. The left panel of Fig. 1 shows an example of a simulation output.

We use 210 N-body simulations from the Quijote suite [7] for training (180), validation (20), and testing (10). Each simulation contains 512^3 particles in a box of 1 $(\mathrm{Gpc}/h)^3$ volume (nearly 5 billion light years in size). In contrast to [6], the training set data was fast approximation simulations using FastPM [8] (see next section), with 32^3 particles in a volume of 128 $(\mathrm{Mpc}/h)^3$. The simulated universes resemble our own, and can be characterized by the value of five cosmological parameters: matter density parameter $\Omega_{\mathrm{m}} = 0.3175$ (this means 31.7% of the Universe is made of dark matter), baryon density parameter $\Omega_{\mathrm{b}} = 0.049$, dark energy density parameter $\Omega_{\Lambda} = 0.6825$, Hubble parameter $h = 0.6711$, matter fluctuation within a 8 Mpc/h sphere $\sigma_8 = 0.834$, primordial spectral index $n_{\mathrm{s}} = 0.9624$. In addition, we also test our model, trained on the above simulations, on three Quijote simulations with different values of the cosmological parameters, but the same configuration otherwise.

Fast Approximation. N-body simulations can be very computationally expensive, as they typically solve the dynamics of billions of particles for thousands of time steps. Fast approximate simulations [9, 8] are usually adopted when a large number of simulations are needed. These methods save computation time by only integrating tens of time steps; they are thus less accurate than full N-body simulations. We aim to build an NN model that is more accurate and faster than these approximators. In this work we use the widely used method COLA (COmoving Lagrangian Acceleration) [9], as implemented in the publicly available package `L-PICOLA` [10], as a benchmark. COLA solves the particle motions relative to predictions by the second order Lagrangian perturbation theory [11]. We setup `L-PICOLA` using the same configurations as the full N-body simulations, running it for only 10 time steps.

Linear Theory.   Fast and accurate solutions exist when matter distribution is close to uniform, and density fluctuations are small. For instance, at linear order, particles move along straight lines, with the distance determined by the growth function D of time: $\mathbf{\Psi}_{\mathrm{lin}}(\mathbf{q}, t) = D(t)/D(t_0)\mathbf{\Psi}_{\mathrm{lin}}(\mathbf{q}, t_0)$. At early times $t_0 \to 0$, such as the starting time of the simulations, $\mathbf{\Psi}_{\mathrm{lin}}(t_0)$ agrees very well with the simulation prediction for $\mathbf{\Psi}(t_0)$ due to uniformity.

The linear theory prediction is a very good approximation on large scales, where density fluctuations are small. However, on small scales, the density contrast increases drastically and structure formation becomes non-linear, limiting the validity of linear theory predictions. Therefore, we use $\mathbf{\Psi}_{\mathrm{lin}}(\mathbf{q}, t)$ as the input to our NN model, which predicts the fully non-linear target $\mathbf{\Psi}(\mathbf{q}, t)$ given by the N-body simulations at the same t. By design, our NN will make accurate predictions on large scales. This is true even if we test the model on universes with cosmological parameters different from the one used for training, as we show below.

Neural Network Model.   Both the input (linear theory) $\mathbf{\Psi}_{\mathrm{lin}}$'s and target ($N$-body simulation) $\mathbf{\Psi}$'s of our NN model are functions of $\mathbf{q}$'s, that form a uniform grid. So each of them is a displacement field that can be viewed as a 3D image, with three channels being the three Cartesian components of the displacement vectors. This allows us to apply many computer vision models to our problem.

In this work we adopt a simple U-Net / V-Net [12, 13] type architecture similar to that in [6]. The model works on 3 levels of resolution connected in a "V" shape by 2 downsampling layers and 2 upsampling layers, achieved by stride-2 2^3 convolutions and stride-½ 2^3 transposed convolutions, respectively. Blocks of 2 3^3 convolutions connect the input, the resampling, and the output layers. As in V-Net, a residual connection, in this case, 1^3 convolutions instead of identity, are added over

each of these convolution blocks. 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. As in the original ResNet [14], the last activation in each residual block acts after the summation. And 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 (3), as well as those after concatenations (128). Finally, an important difference from the original U-Net / V-Net is that we also add the input $\boldsymbol{\Psi}_{\mathrm{lin}}$ directly to the output, so effectively the network is learning the corrections to match the target $\boldsymbol{\Psi}$.

Given a displacement field, we can compute the particle positions $\mathbf{x}$ and calculate their density distribution, characterized by the overdensity field $\delta(\mathbf{x}) \equiv n(\mathbf{x})/\bar{n} - 1$, where $n(\mathbf{x})$ is the particle number in voxel $\mathbf{x}$ and $\bar{n}$ is its mean value. See Fig. 1 for an example of $1 + \delta$ at "today". Because in cosmology the density field is closely related to observables, e.g. galaxies, we compose a loss function that involves both $\boldsymbol{\Psi}$ and n, and combine them as $L = \ln(L_\delta L_{\boldsymbol{\Psi}}^\lambda)$, with L_δ and $L_{\boldsymbol{\Psi}}$ being the MSE losses on $n(\mathbf{x})$ and $\boldsymbol{\Psi}(\mathbf{q})$ respectively. We compute n from $\boldsymbol{\Psi}$ using the second order B-spline kernel (known as cloud-in-cell), so that n is differentiable. By combining the two losses with logarithm rather than summation, we can ignore their absolute magnitudes and trade between their relative values. λ serves as a weight on this trade-off of relative losses. Through experiments, we find that a value of λ from 2 to 5 yields the lowest $\ln(L_\delta L_{\boldsymbol{\Psi}})$, and in this work, we use $\lambda = 3$.

Limited by the GPU memory, the entire input, $\boldsymbol{\Psi}_{\mathrm{lin}}$ (3×512^3), cannot be feed all at once to the model and needs to be cropped into smaller cubes of size 3×128^3. To preserve the physical translational equivariance, we do not use any padding in the 3^3 convolutions, resulting in a smaller output than the input in spatial size. We compensate this by periodically padding 20 voxels per side to the input cubes. To preserve the rotational equivariance of the simulation box, we implement data augmentation that rotates and reflects the displacement fields as in Ref. [6]. We use the Adam optimizer [15] with learning rate 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and reduce the learning rate by half when loss does not improve for 3 epochs.

3 Results

Accuracy We quantify model accuracy by compare the simulation power spectra with the power spectra estimated by the NN and by the benchmark method. The density power spectrum quantifies the correlation of density fluctuations as a function of scale: $P_\delta(k_i) = (N_i V)^{-1} \sum_{k_i < k \le k_{i+1}} \delta(\mathbf{k})\delta(-\mathbf{k})$, where V is the volume, $\delta(\mathbf{k})$ is the Fourier transform of $\delta(\mathbf{x})$, k_i's are bin edges for the wavevector $\mathbf{k}$, and N_i is the number of $\mathbf{k}$'s falling in $(k_i, k_{i+1}]$. The wavenumber $k \equiv |\mathbf{k}|$ describes the scale, with low/high k representing large/small scales. The cross-power spectrum, $\delta_{\mathrm{pred}}(\mathbf{k})\delta_{\mathrm{true}}(-\mathbf{k})$ measures the covariance between the predicted and target density fields. We define a transfer function, $T(k) = \sqrt{P_{\mathrm{pred}}(k)/P_{\mathrm{true}}(k)}$, where $P_{\mathrm{pred}}(k)$ and $P_{\mathrm{true}}(k)$ are the power spectra of the prediction and the simulation, respectively. If $T(k) = 1$, the predictions accurate capture the amplitude of the density field. The cross-correlation coefficient, r, measures the phase correlations: $r(k) = P_{\mathrm{pred} \times \mathrm{true}}(k)/\sqrt{P_{\mathrm{pred}}(k)P_{\mathrm{true}}(k)}$, with the numerator being the cross-power spectrum between the prediction and the simulation. T and r are good estimators to quantify the accuracy of the model, because when they are both 1 the prediction and target are identical as proved in Ref. [6]. Instead of r, we use $1 - r^2$, which gives the amount of unexplained variance between the two fields. Similar to δ, we can also compute $P(k)$, $T(k)$, and $r(k)$ for $\boldsymbol{\Psi}$ by simply replacing the scalar product with the dot vector product.

We consider ten Quijote simulations from the test set (see the left panel of Fig. 1 for an example), together with their fast approximate and NN counterparts. From each of them, we compute their displacement and density power spectra, as well as the corresponding T and $1 - r^2$. We show the results in the central and right panels of Fig. 1. We find that the NN outperforms the benchmark model in accuracy on all scales and for all considered quantities. We emphasize that our NN model produces percent-level accurate results down to scales as small as $k \simeq 1 \ h/\mathrm{Mpc}$.

Generalization to Different Universes. We test the NN model on 2000 universes with drastically different cosmological parameters, where structure formation proceeds in a very different way from that in the training simulations. In particular, we varied all five cosmological parameters:
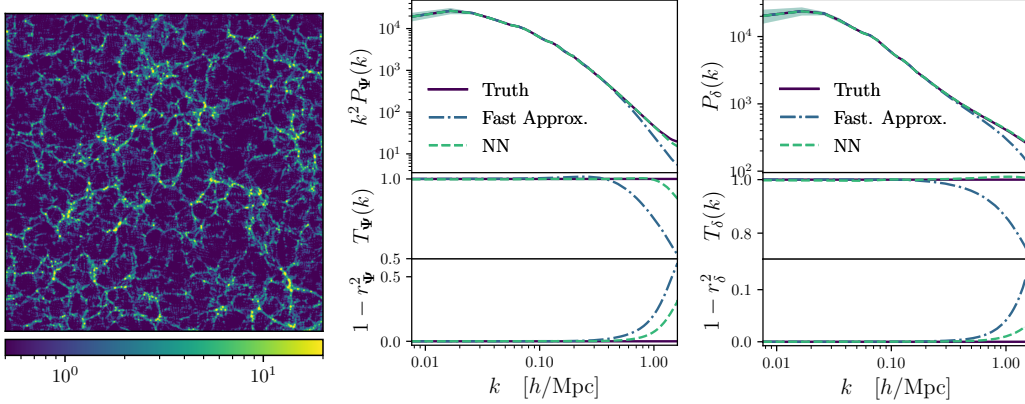
Figure 1: The left panel shows an example of the projected density field in a Universe as predicted by an N-body simulation. We quantify the accuracy of the NN against the fast approximator and the N-body simulation using the power spectrum of the displacement field (middle panel) and the density constrast (right panel). The NN model outperforms the fast simulator in all cases.
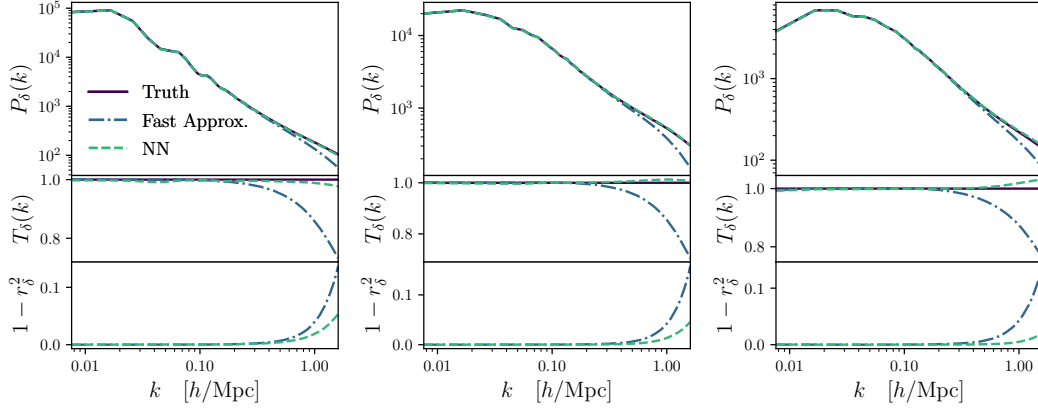

Figure 2: Accuracy comparison between predictions by the fast approximator (blue dot-dashed) and our NN (green dashed) for universes with cosmological parameters very different to those used in training: $\{\Omega_\mathrm{m}, \Omega_\mathrm{b}, h, n_s, \sigma_8\}$ equal to $\{0.1755, 0.0668, 0.7737, 0.8849, 0.6641\}$ (left), $\{0.3889, 0.0363, 0.6275, 0.9513, 0.9137\}$ (middle), $\{0.4291, 0.0337, 0.5511, 1.1687, 0.6619\}$ (right). This demonstrates that our NN generalizes very well.

$\{\Omega_\mathrm{m}, \Omega_\mathrm{b}, h, n_s, \sigma_8\}$ that are relevant to the simulations. We show the prediction comparisons for the density field in Fig. 2, and find that the NN generalizes very well for cosmologies with low $\Omega_\mathrm{b}/\Omega_\mathrm{m}$ and achieves a similar accuracy to that in Fig. 1 (same conclusion holds for the displacement field). It also outperforms the fast approximator COLA, even though COLA depends explicitly on the varied cosmological parameters. This result is in concordance with that in Ref. [6], where Ω_m and σ_8 were varied individually.

Efficiency. We now compare the computational cost of the NN versus the fast approximation benchmark, as well as that of the N-body simulations. We show in Table 1 the total inference time to obtain the whole 512^3 displacement field using 64 128^3 cubes.

The fast approximation uses 20 CPU cores (Intel Xeon CPU E5-2640 v4), and the NN inference uses PyTorch [16] on 1 GPU (NVIDIA Tesla P100-PCIE-16GB). The N-body simulation requires 500 CPU hours on 48 cores, or $\sim 10^5$ seconds if we scale it to 20 cores to compare with the fast approximation. Our NN model is two times faster than the fast approximation benchmark, and more than 10^3 times faster than the N-body simulation.

Table 1: Runtime benchmark. The NN model is $2\times$ faster than the fast approximator, and more than $10^3\times$ faster than the N-body simulation.

	QUIJOTE	L-PICOLA	PyTorch-GPU
Wall Time (s)	10^5	112	59

4   Conclusions

In this paper, we have shown that neural networks are able to accurately emulate the output of expensive N-body simulations. We have shown that our model reproduces the results of full N-body simulations down to scales as small as $k = 1\ h/\mathrm{Mpc}$ at present time to within $\sim 1\%$ accuracy. When compared to the state-of-the-art N-body simulation approximator in cosmology, our V-Net based network achieves better accuracy in less time. Furthermore, we have demonstrated that our network generalizes extremely well, by showing a comparable level of accuracy for universes that it has not been trained on. Our method represents a big step forward in the direction of reducing the computational time needed to provide theory predictions in the non-linear regime. This paves the way to maximizing the scientific return of current and upcoming billion dollar astronomical missions.

Broader Impact

We think the ML methods used in this work may be useful to motivate and enlighten students from different fields, especially those who are interested in astronomy, and to use new tools to help us for a better understanding of our Universe. Moreover, because of the upcoming astronomical probes for the next ten years, we believe that the outcomes of this work might help in future scientists run fast simulations of universes with high precision using a regular computer.

References

[1] ChangHoon Hahn, Villaescusa-Navarro Francisco, Castorina Emanuele, and Scoccimarro Roman. Constraining M_ν with the Bispectrum I: Breaking Parameter Degeneracies. *arXiv e-prints*, page arXiv:1909.11107, Sep 2019.

[2] Elena Massara, Francisco Villaescusa-Navarro, Shirley Ho, Neal Dalal, and David N. Spergel. Using the Marked Power Spectrum to Detect the Signature of Neutrinos in Large-Scale Structure. *arXiv e-prints*, page arXiv:2001.11024, January 2020.

[3] Cora Uhlemann, Oliver Friedrich, Francisco Villaescusa-Navarro, Arka Banerjee, and Sand rine Codis. Fisher for complements: Extracting cosmology and neutrino mass from the counts-in-cells PDF. *arXiv e-prints*, page arXiv:1911.11158, Nov 2019.

[4] E. Allys, T. Marchand, J. F. Cardoso, F. Villaescusa-Navarro, S. Ho, and S. Mallat. New Interpretable Statistics for Large Scale Structure Analysis and Generation. *arXiv e-prints*, page arXiv:2006.06298, June 2020.

[5] Arka Banerjee and Tom Abel. Nearest Neighbor distributions: new statistical measures for cosmological clustering. *arXiv e-prints*, page arXiv:2007.13342, July 2020.

[6] Siyu He, Yin Li, Yu Feng, Shirley Ho, Siamak Ravanbakhsh, Wei Chen, and Barnabás Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Sciences*, 116(28):13825–13832, 2019.

[7] 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.

[8] Yu Feng, Man-Yat Chu, Uroš Seljak, and Patrick McDonald. Fastpm: a new scheme for fast simulations of dark matter and haloes. *Monthly Notices of the Royal Astronomical Society*, 463(3):2273–2286, 2016.

[9] Svetlin Tassev, Matias Zaldarriaga, and Daniel Eisenstein. Solving Large Scale Structure in Ten Easy Steps with COLA. *JCAP*, 06:036, 2013.

[10] Cullan Howlett, Marc Manera, and Will J. Percival. L-PICOLA: A parallel code for fast dark matter simulation. *Astron. Comput.*, 12:109–126, 2015.

[11] Francis Bernardeau, S Colombi, E Gaztanaga, and R Scoccimarro. Large-scale structure of the universe and cosmological perturbation theory. *Physics reports*, 367(1-3):1–248, 2002.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing.

[13] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, October 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. arXiv: 1412.6980.

[16] 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 32*, pages 8026–8037. Curran Associates, Inc., 2019.