



NECOLA: Toward a Universal Field-level Cosmological Emulator

Neerav Kaushal¹, Francisco Villaescusa-Navarro^{2,3}, Elena Giusarma¹, Yin Li^{2,4}, Conner Hawry¹, and Mauricio Reyes¹¹Department of Physics, Michigan Technological University, Houghton, MI 49931, USA; kaushal@mtu.edu²Center for Computational Astrophysics, 162 5th Avenue, New York, NY 10010, USA³Department of Astrophysical Sciences, Princeton University, Peyton Hall, Princeton NJ 08544, USA⁴Center for Computational Mathematics, 162 5th Avenue, New York, NY 10010, USA

Received 2021 November 9; revised 2022 February 21; accepted 2022 March 8; published 2022 May 10

Abstract

We train convolutional neural networks to correct the output of fast and approximate N -body simulations at the field level. Our model, Neural Enhanced COLA (NECOLA), takes as input a snapshot generated by the computationally efficient COLA code and corrects the positions of the cold dark matter particles to match the results of full N -body Quijote simulations. We quantify the accuracy of the network using several summary statistics, and find that NECOLA can reproduce the results of the full N -body simulations with subpercent accuracy down to $k \simeq 1 \text{ hMpc}^{-1}$. Furthermore, the model that was trained on simulations with a fixed value of the cosmological parameters is also able to correct the output of COLA simulations with different values of Ω_m , Ω_b , h , n_s , σ_8 , w , and M_ν with very high accuracy: the power spectrum and the cross-correlation coefficients are within $\simeq 1\%$ down to $k = 1 \text{ hMpc}^{-1}$. Our results indicate that the correction to the power spectrum from fast/approximate simulations or field-level perturbation theory is rather universal. Our model represents a first step toward the development of a fast field-level emulator to sample not only primordial mode amplitudes and phases, but also the parameter space defined by the values of the cosmological parameters.

Unified Astronomy Thesaurus concepts: Cosmological parameters (339); N -body simulations (1083); Neural networks (1933); Large-scale structure of the universe (902)

1. Introduction

In order to extract valuable information about fundamental physics from cosmic surveys, we need theoretical predictions to confront the collected data. On semilinear scales, analytic tools like perturbation theory (Bernardeau et al. 2002) can be used to provide such theoretical predictions. However, on nonlinear scales, where a large amount of cosmological information resides (e.g., Allys et al. 2020; Banerjee et al. 2020; Dai et al. 2020; de la Bella et al. 2020; Friedrich et al. 2020; Giri & Smith 2020; Hahn et al. 2020; Villaescusa-Navarro et al. 2020; Uhlemann et al. 2020; Banerjee & Abel 2021a, 2021b; Gualdi et al. 2021a, 2021b; Bayer et al. 2021; Hahn & Villaescusa-Navarro 2021; Kuruvilla 2021; Kuruvilla & Aghanim 2021; Massara et al. 2021; Samushia et al. 2021; Valogiannis & Dvorkin 2021), numerical simulations become necessary.

Cosmological simulations can be classified into two broad categories: (1) N -body simulations that model the matter field accounting only for the force of gravity, and (2) hydrodynamic simulations that model not only gravity but also fluid hydrodynamics and astrophysical effects such as the formation of stars and feedback from black holes. While computationally more efficient than hydrodynamic simulations, N -body simulations are still expensive, and running large sets or high-resolution simulations requires a significant computational cost (e.g., Garrison et al. 2018; DeRose et al. 2019; McClintock et al. 2019a, 2019b; Nishimichi et al. 2019; Zhai et al. 2019; Villaescusa-Navarro et al. 2020; Angulo et al. 2021; Ishiyama et al. 2021; Maksimova et al. 2021). To overcome this, several methods have been developed that are much less

computationally demanding but come at the expense of being as accurate (e.g., ALPT, Kitaura & Heß 2013; PThalos, Scoccimarro & Sheth 2002; PINOCCHIO, Monaco et al. 2002; FastPM, Feng et al. 2016; COLA, Tassev et al. 2013, 2015; Howlett et al. 2015; EZMOCKS, Chuang et al. 2015; FlowPM, Modi et al. 2020; PATCHY, Kitaura et al. 2014; log-normal models, Coles & Jones 1991; Agrawal et al. 2017; HALOGEN, Avila et al. 2015; MUSCLE-UPS, Tosone et al. 2021; QPM, White et al. 2013; HaloNet, Berger & Stein 2018; mass-Peak Patch, Bond & Myers 1996; Stein et al. 2019).

Being able to run fast and accurate simulations is of main importance in cosmology in order to provide the theoretical predictions needed to retrieve the maximum information from cosmological surveys. In this work, we try to build a bridge between the fast and approximate COLA simulations, and the expensive and accurate full N -body simulations using deep learning. Deep-learning techniques have been used more recently to generate superresolution realizations of the full phase-space matter distribution of the universe from the low-resolution N -body simulations (Li et al. 2021; Ni et al. 2021). We build on the work of He et al. (2019) and Alves de Oliveira et al. (2020) who used neural networks to find the mapping between the displacement field generated by the Zel'dovich approximation to the one from fast and full N -body simulations, respectively. In this work, we train convolutional neural networks to correct the particle positions from COLA simulation snapshots to match those of full N -body Quijote simulations. The most important conclusion of our work is that our model seems to be universal, i.e., once trained on simulations with a fixed value of cosmological parameters, our network is able to correct the particle positions of COLA simulations with any other cosmology with surprising accuracy: the power spectrum is accurate at the 1% level down to $k = 1 \text{ hMpc}^{-1}$.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

This paper is organized as follows. In Section 2, we describe the simulations we use and the architecture of our neural network model. We present the results of the trained network in Section 3. Finally, we draw our conclusions in Section 4.

2. Methods

In this section, we describe the two types of simulations we used, together with the model architecture and the training procedure.

2.1. Simulations

2.1.1. Full N-body Simulations

We made use of the Quijote full N -body simulations (Villaescusa-Navarro et al. 2020) to both train and test the model. The simulations used in this work follow the evolution of 512^3 cold dark matter (CDM) particles (plus 512^3 neutrino particles in the case of massive neutrino cosmologies) from $z = 127$ down to $z = 0$ in a periodic volume of $(1000 h^{-1} \text{Mpc})^3$. We train the network using a set of 100 simulations from the fiducial cosmology, where the values of the cosmological parameters are fixed to $\Omega_m = 0.3175$, $\Omega_b = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $\sigma_8 = 0.834$, $w = -1$, and $M_\nu = 0.0 \text{ eV}$. These simulations are only different in the value of the initial random seed.

We test the accuracy of our network on simulations with very different cosmologies to the one used in the training. For this, we made use of 100 out of 2000 simulations of the latin hypercube contained in the Quijote simulations, where the values of the cosmological parameters span the range $\Omega_m \in [0.1, 0.5]$, $\Omega_b \in [0.03, 0.07]$, $h \in [0.5, 0.9]$, $n_s \in [0.8, 1.2]$, and $\sigma_8 \in [0.6, 1.0]$. In these simulations, not only are the set of values of the cosmological parameters different, but the initial random seed varies as well. Furthermore, we test the accuracy of our network on models with massive neutrinos and on models where the dark energy equation of state is $w \neq -1$, making use of Quijote simulations labeled M_ν^+ , M_ν^{++} , M_ν^{+++} , w^+ , and w^- . On average, each of the Quijote simulations used in this work required ~ 500 Central Processing Unit (CPU) hours to run. We refer the reader to Villaescusa-Navarro et al. (2020) for further details on the Quijote simulations.

2.1.2. Approximate N-body Simulations

The fast and approximate simulations we use in this work are run with the COmoving Lagrangian Acceleration (COLA; Tassev et al. 2013) method, which combines second-order Lagrangian perturbation theory (2LPT; Bernardeau et al. 2002) on large scales with N -body methods on small scales. In particular, we use the MG-PICOLA (Wright et al. 2017) package. For each Quijote full N -body simulation, we run a COLA simulation by matching (1) the number of particles, (2) the set of values of the cosmological parameters, and (3) the value of the initial random seed, which gives rise to an identical initial Gaussian field for both Quijote and COLA. These simulations require fewer time steps than the full N -body simulations and are therefore much more computationally efficient. Each COLA simulation is run with 30 time steps equally spaced in log from $z = 9$ down to $z = 0$. On average, these simulations only take 3 CPU hours to run.

2.2. Model

2.2.1. Input and Target

Let us write the displacement vector of a particle as $\mathbf{d} = \mathbf{x}_f - \mathbf{x}_i$, where $\mathbf{x}_f$ and $\mathbf{x}_i$ are the final ($z = 0$) and initial (Lagrangian) position of the particle. Our goal is to train a neural network to correct the positions of the particles generated by COLA, to match them with those from a full N -body simulation, i.e.,

$$\mathbf{x}_{f,\text{Nbody}} = g(\mathbf{x}_{f,\text{COLA}}), \quad (1)$$

where g is an unknown function. Note that the right-hand side of Equation (1) should not be taken as the position of the particular particle considered, but also of all its neighboring particles. To preserve translational equivariance, we use displacement vectors instead of absolute particle positions. Thus, the input to the network is $\mathbf{d}_{\text{COLA}}$, rather than $\mathbf{x}_{f,\text{COLA}}$. The network is trained to learn $\mathbf{d}_{\text{Nbody}} - \mathbf{d}_{\text{COLA}} = \mathbf{x}_{f,\text{Nbody}} - \mathbf{x}_{f,\text{COLA}}$.

2.2.2. Model Architecture

We follow Alves de Oliveira et al. (2020) and use a V-Net-based model (Milletari et al. 2016) that consists of two downsampling and three 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 after the first one and the last two, and leaky ReLU activation with a negative slope of 0.01 after every batch normalization, as well as after the first and the second-to-last convolutions. The last activation in each residual block acts after the summation, following Milletari et al. (2016). As in U-Net/V-Net, at all resolution levels (with the exception of the bottleneck 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 three channels (the displacement vector along each Cartesian coordinate), as well as those after the concatenations (128-channeled). Finally, the input ($\mathbf{d}_{\text{COLA}}$) is directly added to the output, so that the network can learn the corrections to match the target ($\mathbf{d}_{\text{Nbody}} - \mathbf{d}_{\text{COLA}}$). Stride-2 2^3 convolutions and stride-1/2 2^3 transposed convolutions are used in downsampling and upsampling layers, respectively. A diagram of the network architecture is shown in Figure 1.

Following Alves de Oliveira et al. (2020), we minimize a loss function given by $L = \log_e(L_\delta L_\Delta^\lambda)$, where L_δ is the mean squared error (MSE) loss on $n(\mathbf{x})$ (the particle number in voxel $\mathbf{x}$) and L_Δ is the MSE on the displacement vector $\mathbf{d}$. With this loss function, we are able to train the model to make accurate predictions in both Lagrangian and Eulerian spaces. By combining the two losses with logarithm rather than summation, we can account for their absolute magnitudes and trade between their relative values; λ here serves as a weight on this trade-off of relative losses and $\lambda = 1$ works pretty well in our case.

The input cannot be fed into the network all at once due to the big size of the data (3×512^3), and we thus divide it into smaller chunks first. We crop the data into subcubes of size 3×128^3 , corresponding to a simulation box of length $250 h^{-1} \text{Mpc}$. In order to preserve the physical translational equivariance, no padding has been used in the 3^3 convolutions,

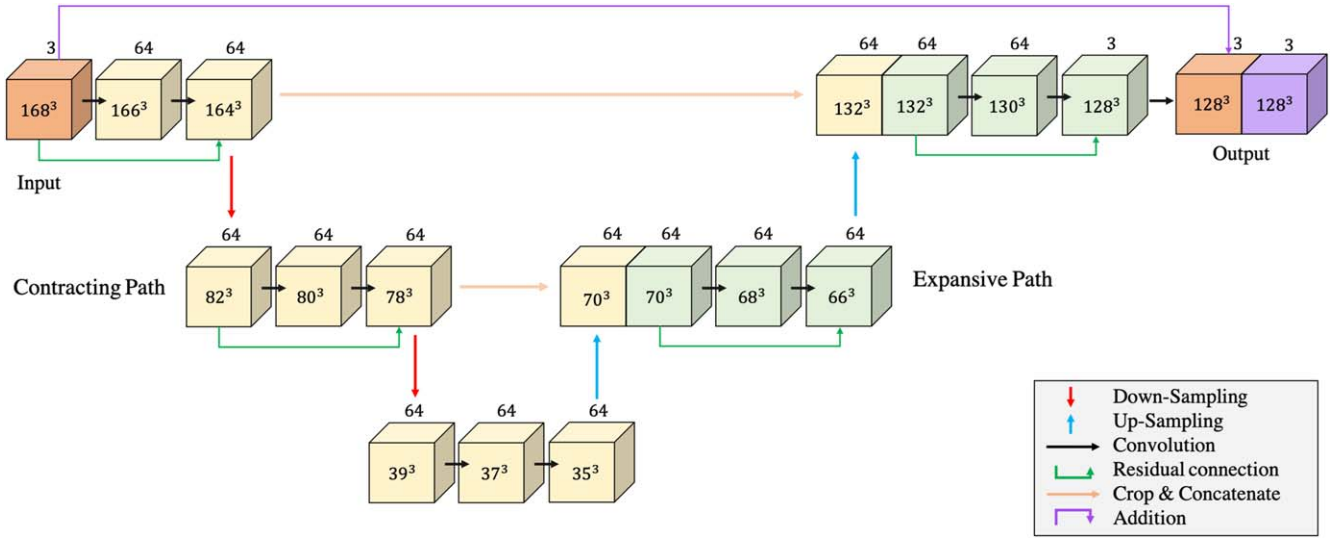


Figure 1. The diagram shows the architecture of our model, NECOLA. The top-leftmost cube (orange) represents the input and the top-rightmost cubes (orange and purple) represent the output. The cubes in yellow and green represent various multichannel feature maps. The number inside each cube represents the size of the feature map while the number on the top of each cube represents the number of channels in the map. See Section 2.2.2 for more details on the convolution operations.

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 3×168^3 . Furthermore, data augmentation is implemented to enforce the equivariance of displacement fields under rotational and parity transformations. We use the Adam optimizer (Kingma & Ba 2017) 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). From now on, we will refer to this model as NECOLA, from Neural Enhanced COLA, in order to avoid any confusion with the model by Alves de Oliveira et al. (2020), which uses Zel’dovich simulations as input and a different value of λ . Note that the model architecture of NECOLA is the same as that of Alves de Oliveira et al. (2020).

2.3. Benchmark Models

In order to compare the predictions of our model, we have used four different benchmarks:

1. *COLA*: This benchmark represents the results of running the COLA simulation itself.
2. *Zel’dovich Approximations (ZA)*: In this case, the positions of the particles at $z=0$ are computed using the Zel’dovich approximation.
3. *mod(ZA)*: This benchmark represents our model but trained on ZA simulations as input and correcting the output to match the target N -body simulations.
4. *Neural Network(Zel’dovich Approximations) ([NN(ZA)])*: This benchmark is the model developed by Alves de Oliveira et al. (2020), which takes ZA simulations as input and corrects the output to match full N -body simulations. We refer the reader to Alves de Oliveira et al. (2020) for further details on this model.

3. Results

In this section, we investigate the performance of our model. We first make use of several summary statistics to quantify the accuracy of our model for simulations with the same cosmology as the one used to train the network. Then, we investigate how well does our network extrapolate to other cosmological models.

3.1. Fiducial Cosmology

We first present the results of testing the network on simulations that have the same cosmology as the one used for its training.

3.1.1. Visual Comparison

Before quantifying the accuracy of the network using summary statistics, we perform a visual inspection of its output. In Figure 2, we show the distribution of matter at $z=0$ from the full N -body simulation (top row), the COLA simulation (middle row), and NECOLA (bottom row).

While looking at large scales, the agreement between the three methods is really good, but when we look at small scales, some differences are visible. In the case of COLA, the output is more diffuse and halos do not exhibit a high concentration in their centers, in contrast to the corresponding N -body simulation. On the other hand, NECOLA produces much sharper results, clearly defining the positions and boundaries of dark matter halos.

3.1.2. Power Spectrum

The power spectrum is defined as the Fourier transform of the two-point correlation function, which measures the excess probability of finding a pair of random galaxies (or points) at a given separation compared to the one from a random distribution. The power spectrum is one of the most important summary statistics used in cosmology because for Gaussian density fields (like the one our universe resembles on large,

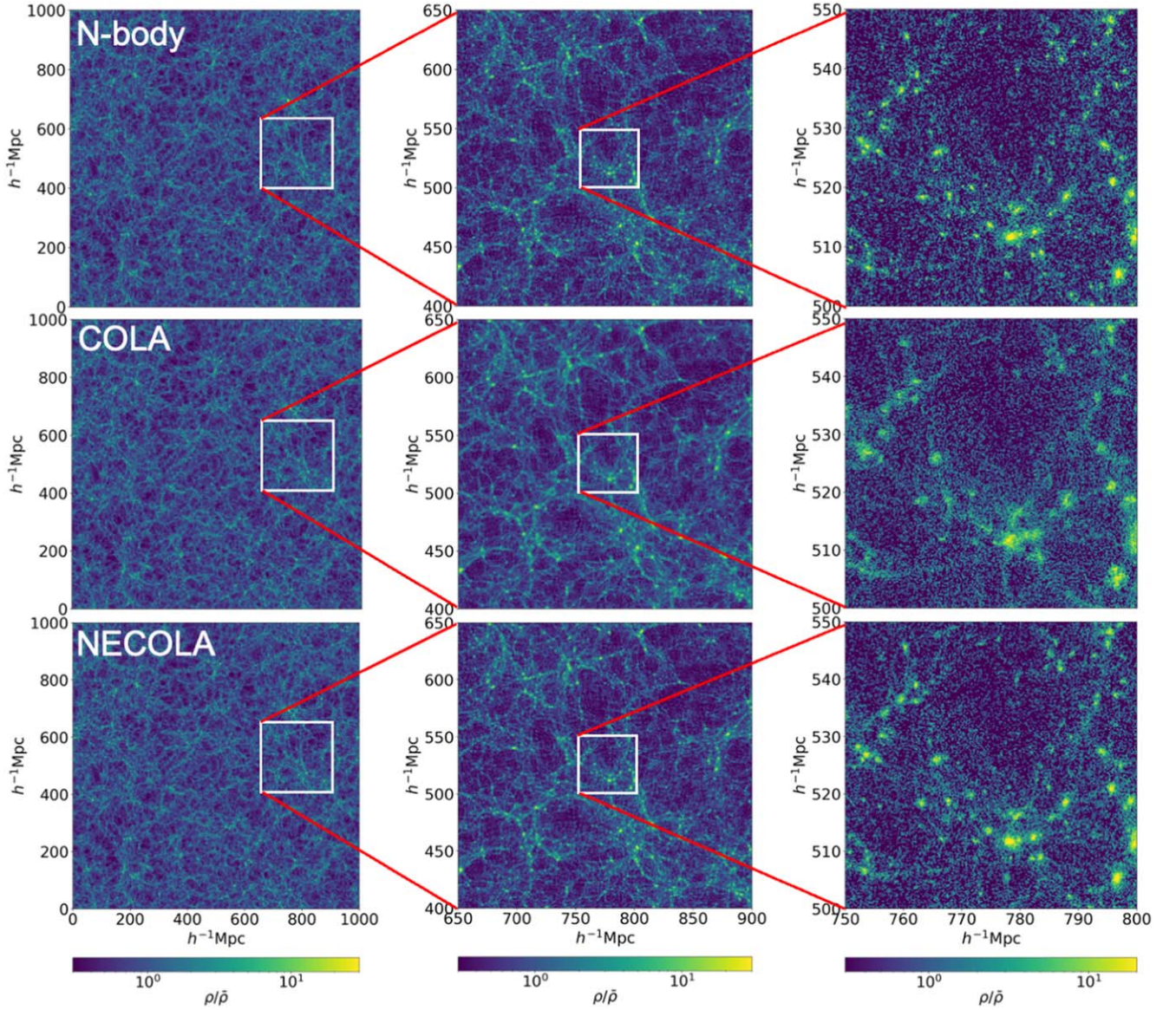


Figure 2. The figure shows the cold dark matter density fields for the target N -body simulations (top), the input/benchmark COLA simulations (middle), and the predictions of our model (bottom), at a scale of 1000 Mpc h^{-1} (left column), 250 Mpc h^{-1} (middle column), and 50 Mpc h^{-1} (right column). Each figure is a zoomed-in image of the white box in the figure on its left.

linear scales), it fully characterizes the statistical properties of the field.

In the top-left panel of Figure 3, we show with a solid black line the average power spectra from 10 Quijote simulations of the test set. The dotted blue line shows the average power spectrum from the corresponding COLA simulations, while the green dotted-dashed line outputs the average power spectrum of Zel'dovich-evolved simulations. The solid yellow and dashed red lines show the average power spectrum from mod (ZA) and NECOLA, respectively. As can be seen, the worst model is the one that only employs the Zel'dovich approximation, followed by the COLA simulation.

In order to better visualize the differences between the output of the N -body simulation and the networks, we plot in the middle-left panel of Figure 3, the transfer function, defined as

$$T(k) = \sqrt{\frac{P_{\text{pred}}(k)}{P_{\text{target}}(k)}}, \quad (2)$$

where $P_{\text{pred}}(k)$ and $P_{\text{target}}(k)$ are the average matter power spectra of the predictions and the target density fields, respectively. Values close to one indicate a better agreement between the prediction and the target. As can be seen, both networks achieve a subpercent accuracy on the power spectrum down to $k = 1 \text{ h Mpc}^{-1}$, though the results obtained from NECOLA are slightly more accurate. We note that in the case of the Quijote simulations, it does not make sense to look into much smaller scales than $k \sim 1 \text{ h Mpc}^{-1}$, as those are not numerically converged in the simulations due to mass resolution (Villaescusa-Navarro et al. 2020).

We note that there exists state-of-the-art power spectrum emulators such as COSMIC EMU (Heitmann et al. 2009, 2010; Lawrence et al. 2010), FRANKEN EMU (Heitmann et al. 2014), and MIRA TITAN (Heitmann et al. 2016; Lawrence et al. 2017) that are computationally faster and more accurate in estimating the power spectrum but are not used in our

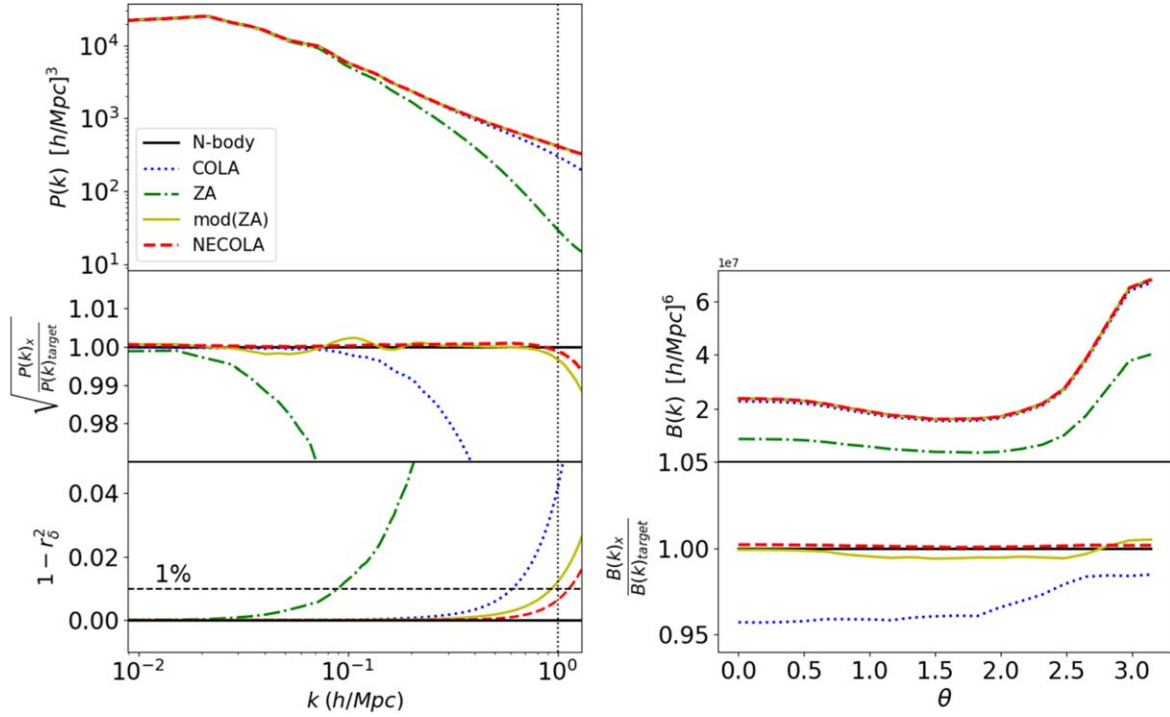


Figure 3. The left plot shows the 3D-matter power spectrum (top), the transfer function (middle), and the cross-correlation coefficient (bottom), while the right plot shows the bispectrum for $k_1 = 0.15 \text{ hMpc}^{-1}$ and $k_2 = 0.25 \text{ hMpc}^{-1}$ (top) and the bispectrum ratio (bottom) for the target N -body simulations (solid black), the COLA simulations (dotted blue), the ZA approximations (dashed-dotted green), mod(ZA) (solid yellow), and NECOLA (dashed red). As can be seen, NECOLA outperforms all benchmarks in all cases.

comparisons as the primary objective of our work is to provide a field-level emulator itself and not a power spectrum emulator.

3.1.3. Cross-correlation Coefficient

In Fourier space, every mode can be written as $\delta(\mathbf{k}) = Ae^{i\theta}$, where A and θ are the mode amplitude and phase, respectively. When using the power spectrum, we are effectively comparing how well the amplitude of the modes from the network and the simulation agree. However, that statistic neglects the correlations in mode phases, which are very important in the nonlinear regime. To quantify the correlations between the mode phases, we use the cross-correlation coefficient, r , defined as

$$r(k) = \frac{P_{\text{pred} \times \text{target}}(k)}{\sqrt{P_{\text{pred}}(k)P_{\text{target}}(k)}}, \quad (3)$$

where the numerator is the cross-power spectrum between the predictions and the target, and the denominator contains the autopower spectrum of the prediction and the target. Values of r close to one indicate a very good correlation in mode phases. In the bottom-left panel of Figure 3, we show the cross-correlation coefficient averaged over the testing set for the different cases considered. We find that NECOLA achieves the highest accuracy, being within 1% down to $k = 1 \text{ hMpc}^{-1}$.

3.1.4. Bispectrum

The third statistic that we consider to quantify the agreement between the full simulations and the network predictions is the bispectrum, defined as

$$\langle \delta_{k_1} \delta_{k_2} \delta_{k_3} \rangle \equiv \delta_D(\mathbf{k}_{123}) B(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3), \quad (4)$$

where $\delta(\mathbf{k})$ is the overdensity in the Fourier space and $\mathbf{k}_{123} \equiv \mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3$.

Differently to the power spectrum, the bispectrum quantifies the correlation between triplets of modes in closed triangles. For Gaussian density fields, this quantity is zero, and therefore, its amplitude and shape capture information about the non-Gaussianities in a given field. In the top-right panel of Figure 3, we show the bispectrum for $k_1 = 0.15 \text{ hMpc}^{-1}$ and $k_2 = 0.25 \text{ hMpc}^{-1}$ as a function of the angle between k_1 and k_2 , θ . On this scale, we cannot see large differences, besides the fact that the Zel'dovich approximation underestimates the amplitude of the bispectrum, as expected. In the bottom-right panel of Figure 3, we show the ratio between the different bispectra to the bispectrum of the N -body simulation. We find that both neural networks give very accurate results, although NECOLA is slightly more accurate. The Appendix also shows a comparison of the results from various models on a single realization in the testing set.

The above values of k_1 and k_2 are chosen in order to probe the nonlinear scales of the universe at which the non-Gaussian signatures in the mass distribution (induced by nonlinear gravitational instability) are imprinted. The model has been evaluated at other values of k_1 and k_2 as well, and performs equally well.

3.2. Model Extrapolation

We now explore how our model extrapolates to cosmologies different from the one used to train the model.

We first test the extrapolation properties of the model on the parameters Ω_m , Ω_b , h , n_s , and σ_8 by using 100 simulations of the Quijote latin-hypercube set. We emphasize that for the

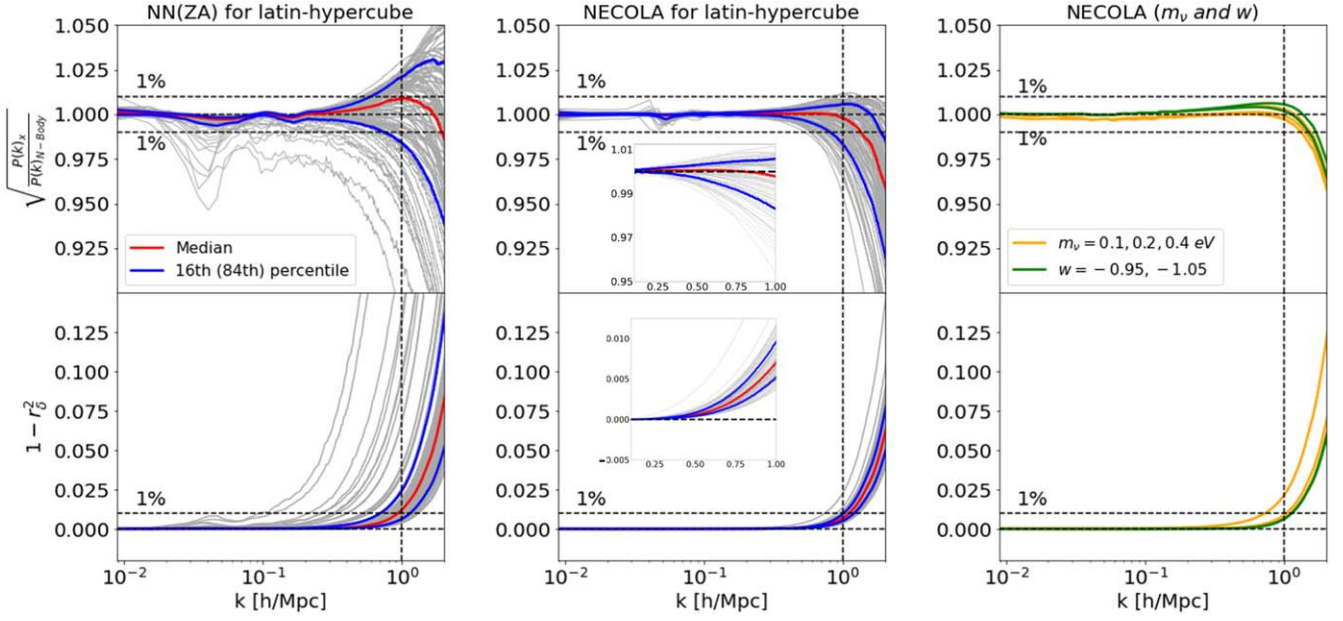


Figure 4. We test the NN(ZA) and NECOLA models, which are trained on simulations with a fixed cosmology, on models with very different values of the cosmological parameters. The left and middle panels show the results when using 100 simulations of the Quijote latin hypercube (that varies Ω_m , Ω_b , h , n_s , and σ_8), while the right panel displays the results for cosmologies with massive neutrinos and a dark energy equation of state different to -1 . The red lines represent the median while the blue lines represent the 16th (and 84th) percentile of the predictions. As can be seen, NECOLA not only performs better than NN(ZA), but it is surprisingly accurate all the way down to $k \sim 1 \text{ hMpc}^{-1}$. Besides, it also works for models with massive neutrinos and $w \neq -1$. The curve with the largest difference in the neutrino cross-correlation coefficient corresponds to a model with $M_\nu = 0.4 \text{ eV}$.

Table 1
Computational Cost Associated with Running a Full N -body Simulation, a COLA Simulation, NN(ZA), and NECOLA

Simulation	N -body (QUIJOTE)	Fast (COLA)	NECOLA (PyTorch-GPU)	NN(ZA)
CPU-/GPU-sec	10^6	10^4	125	59

Note. Note that in the cases of NECOLA and NN(ZA), we report the GPU wall time.

simulations in this set, the values of these five cosmological parameters are varied at the same time, together with the value of the initial random seed. For each of these simulations, we run its COLA counterpart and input it to the network, which corrects the positions of the particles.

For each cosmology, we compute the power spectrum of the output of NECOLA and of the full N -body Quijote simulation. In Figure 4, we show in the middle panel, the transfer function together with the cross-correlation coefficient. As can be seen, NECOLA is able to correct the output of the COLA simulations in all cases with surprising accuracy: below $\simeq 1\%$ down to $k = 1 \text{ hMpc}^{-1}$.

Next, we repeat the same exercise but using NN(ZA) and show the results in the left panel of Figure 4. As can be seen, the network trained on COLA snapshots exhibits much stronger extrapolation features than the one trained on Zel'dovich displacements.

We now investigate if NECOLA is also able to correct COLA outputs for simulations with massive neutrinos. We emphasize that no simulations used for training the model contain massive neutrinos. For this, we made use of simulations from the M_ν^+ , M_ν^{++} , and M_ν^{+++} Quijote sets, corresponding to cosmologies with sums of the neutrino masses equal to 0.1, 0.2, and 0.4 eV. In these simulations, we have both dark matter and neutrino particles. From each set, we take 10 simulations and run their COLA counterpart. Next, we input to NECOLA the displacement vectors of the dark matter

particles of the COLA simulation, and NECOLA outputs the corrected positions of the dark matter particles for these massive neutrino models.

In the right panel of Figure 4, we show the results of this calculation with yellow lines. As can be seen, NECOLA is able to correct the positions of the dark matter particles such that their power spectrum and cross-correlation coefficient agree with the full N -body calculation below 1% down to $k = 1 \text{ hMpc}^{-1}$. We note that although our network only works with the cold dark matter field, assuming a linear neutrino field correlated with the initial Gaussian field, it will for most of the cases, give very accurate predictions for the total matter field (Massara et al. 2014). On the other hand, the cold dark matter field is the one responsible for the abundance and clustering of dark matter halos and galaxies (Castorina et al. 2014; Villaescusa-Navarro et al. 2014). In Giusarma et al. (2019), the authors proposed a deep-learning-based Convolutional Neural Network (U-Net) model to generate simulations with massive neutrinos from standard Λ CDM simulations without neutrinos. Their model was able to reproduce the three-dimensional spatial distribution of matter up to scales of 0.7 hMpc^{-1} (see Figure 4 of Giusarma et al. 2019), thus emulating the effect of massive neutrinos on the large-scale structure. It is interesting to note that NECOLA gives more accurate results than the model by Giusarma et al. (2019) at all scales, capturing the effects of nonlinear evolution.

Lastly, we study the performance of our model for cosmologies with values of the dark energy equation of state, w , different from -1 . For this, we made use of the 10 simulations of the w^+ and w^- Quijote sets, which have a value of w equal to -0.95 and -1.05 , respectively. For each of these simulations, we run their COLA counterpart and compute the displacement vectors. We then input those into the network that returns the corrected positions of the dark matter particles. In the right panel of Figure 4, we show with green lines the results of computing the transfer function and cross-correlation coefficient between the output of the network and the full N -body simulations. As can be seen, in this case as well, NECOLA is able to correct the output of the cosmologies that it has never seen before.

3.3. Computational Cost

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. We run our Convolutional Neural Net (CNN) model on 1 Graphics Processing Unit (GPU) (320 NVIDIA P100-16GB) using PyTorch (Paszke et al. 2019) and it takes ~ 125 GPU seconds to run. 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.

4. Summary

Providing accurate theoretical predictions is necessary in order to extract the maximum amount of information from upcoming cosmological surveys. The computational cost of running full N -body simulations is currently too expensive to carry out standard analysis such as Markov Chain Monte Carlo. On the other hand, fast simulations can reduce the computational cost by orders of magnitude at the expense of sacrificing accuracy.

In this work, we have shown that we can use neural networks to correct the output of approximate simulations to match full N -body simulations from the Quijote suite. Our model, coined NECOLA, from Neural Enhanced COLA, has been trained on simulations with a fixed value of the cosmological parameters. We have shown that our model is not only able to correct the output of COLA simulations run with the same cosmology as the one used to train the network, but is also able to correct COLA simulations that have very different values of the parameters Ω_m , Ω_b , h , n_s , σ_8 , M_ν , and w . This surprising feature of our network indicates that the correction from the output of COLA to a full N -body might be universal, i.e., independent of cosmology.

This may have important consequences for perturbation theory studies that are able to accurately model the linear and

perturbative regime but fail on nonlinear scales. Our work indicates that a generic, cosmology-independent correction may be feasible, at least in the case of the power spectrum.

Our network can be used as a field-level emulator for covariance estimation, likelihood-free analysis, detecting features in the cosmic web, and to explore not only the initial mode amplitudes and phases (Jasche & Wandelt 2013a), but also the cosmological parameter space (Jasche & Wandelt 2013b).

We note however that further work is needed to claim that our model is precise for statistics other than the power spectrum and cross-correlation function when using it in extrapolation. In future work, we will quantify the accuracy of our network on other summary statistics like bispectrum, halo mass function, etc. Additional work is also needed to incorporate velocities into this framework, which will allow performing studies in redshift space. Besides, further work is needed to quantify the accuracy of NECOLA at redshifts other than the one used for training, together with the universality of the network under changes of simulation resolution.

Overall, this work opens an interesting direction in the development of fast and generalized field-level emulators needed to maximize the scientific return of upcoming cosmological missions.

The trained models, predictions, and statistics extracted from the testing and extrapolation sets are hosted under the public GitHub repository <https://github.com/neeravkaushal/cola-to-nbody.git> and the model has been trained using the map2map code <https://github.com/eelregit/map2map.git>.

The Quijote simulations used in this work are publicly available at <https://github.com/franciscovillaescusa/Quijote-simulations.git>. The COLA simulations have been run using the MG-PICOLA code, publicly available at <https://github.com/HAWinther/MG-PICOLA-PUBLIC.git>. 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). E.G. thanks Michigan Space Grant Consortium for their support. The work of F.V.-N has been supported by the WFIRST program through NNG26PJ30C and NNN12AA01C. F.V.-N and Y.L. are supported by the Simons Foundation. Research reported in this publication was supported in part by funding provided by the National Aeronautics and Space Administration (NASA), under award No. NNX15AJ20H, Michigan Space Grant Consortium (MSGC).

Appendix Predictions on a Single Realization

To see how well our model performs on a single realization, the performance of various models on a single realization of the testing set is compared in Figure 5. The results averaged over the entire testing set are shown in Figure 3.

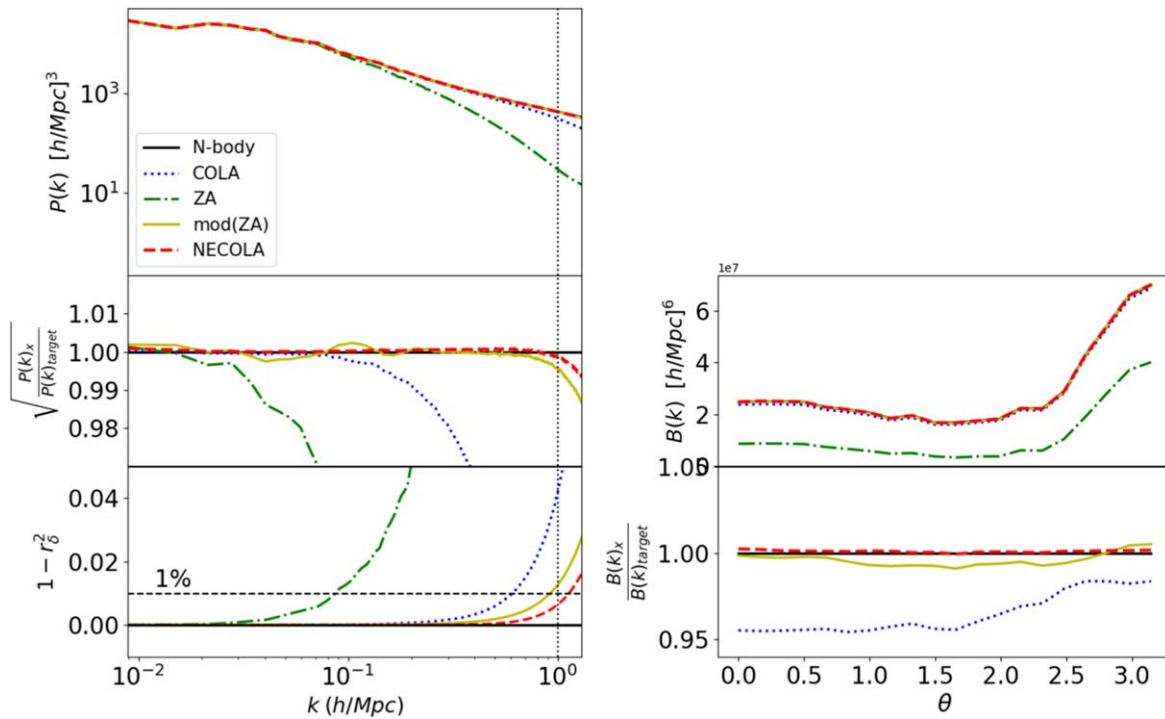


Figure 5. Model predictions for a single realization from the testing set.

ORCID iDs

Neerav Kaushal <https://orcid.org/0000-0003-4786-2348>
 Francisco Villaescusa-Navarro <https://orcid.org/0000-0002-4816-0455>
 Elena Giusarma <https://orcid.org/0000-0003-3052-3059>
 Yin Li <https://orcid.org/0000-0002-0701-1410>

References

- Agrawal, A., Makiya, R., Chiang, C.-T., et al. 2017, *JCAP*, 2017, 003
 Allys, E., Marchand, T., Cardoso, J. F., et al. 2020, *PhRvD*, 102, 103506
 Alves de Oliveira, R., Li, Y., Villaescusa-Navarro, F., Ho, S., & Spergel, D. N. 2020, arXiv:2012.00240
 Angulo, R. E., Zennaro, M., Contreras, S., et al. 2021, *MNRAS*, 507, 5869
 Avila, S., Murray, S. G., Knebe, A., et al. 2015, *MNRAS*, 450, 1856
 Banerjee, A., & Abel, T. 2021a, *MNRAS*, 500, 5479
 Banerjee, A., & Abel, T. 2021b, *MNRAS*, 504, 2911
 Banerjee, A., Castorina, E., Villaescusa-Navarro, F., Court, T., & Viel, M. 2020, *JCAP*, 2020, 032
 Bayer, A. E., Villaescusa-Navarro, F., Massara, E., et al. 2021, arXiv:2102.05049
 Berger, P., & Stein, G. 2018, *MNRAS*, 482, 2861
 Bernardeau, F., Colombi, S., Gaztañaga, E., & Scoccimarro, R. 2002, *PhR*, 367, 1
 Bond, J. R., & Myers, S. T. 1996, *ApJS*, 103, 1
 Castorina, E., Sefusatti, E., Sheth, R. K., Villaescusa-Navarro, F., & Viel, M. 2014, *JCAP*, 2, 049
 Chuang, C.-H., Kitaura, F.-S., Prada, F., Zhao, C., & Yepes, G. 2015, *MNRAS*, 446, 2621
 Coles, P., & Jones, B. 1991, *MNRAS*, 248, 1
 Dai, J.-P., Verde, L., & Xia, J.-Q. 2020, *JCAP*, 2020, 007
 de la Bella, L. F., Tessore, N., & Bridle, S. 2020, arXiv:2011.06185
 DeRose, J., Wechsler, R. H., Tinker, J. L., et al. 2019, *ApJ*, 875, 69
 Feng, Y., Chu, M.-Y., & Seljak, U. 2016, arXiv:1603.00476
 Friedrich, O., Uhlemann, C., Villaescusa-Navarro, F., et al. 2020, *MNRAS*, 498, 464
 Garrison, L. H., Eisenstein, D. J., Ferrer, D., et al. 2018, *ApJS*, 236, 43
 Giri, U., & Smith, K. M. 2020, arXiv:2010.07193
 Giusarma, E., Reyes Hurtado, M., Villaescusa-Navarro, F., et al. 2019, arXiv:1910.04255
 Gualdi, D., Gil-Marín, H., & Verde, L. 2021a, arXiv:2104.03976
 Gualdi, D., Novell, S., Gil-Marín, H., & Verde, L. 2021b, *JCAP*, 2021, 015
 Hahn, C., & Villaescusa-Navarro, F. 2021, *JCAP*, 2021, 029
 Hahn, C., Villaescusa-Navarro, F., Castorina, E., & Scoccimarro, R. 2020, *JCAP*, 2020, 040
 He, S., Li, Y., Feng, Y., et al. 2019, *PNAS*, 116, 13825
 Heitmann, K., Bingham, D., Lawrence, E., et al. 2016, *ApJ*, 820, 108
 Heitmann, K., Higdon, D., White, M., et al. 2009, *ApJ*, 705, 156
 Heitmann, K., Lawrence, E., Kwan, J., Habib, S., & Higdon, D. 2014, *ApJ*, 780, 111
 Heitmann, K., White, M., Wagner, C., Habib, S., & Higdon, D. 2010, *ApJ*, 715, 104
 Howlett, C., Manera, M., & Percival, W. J. 2015, *A&C*, 12, 109
 Ishiyama, T., Prada, F., Klypin, A. A., et al. 2021, *MNRAS*, 506, 4210
 Jasche, J., & Wandelt, B. D. 2013a, *ApJ*, 779, 15
 Jasche, J., & Wandelt, B. D. 2013b, *MNRAS*, 432, 894
 Kingma, D. P., & Ba, J. 2017, arXiv:1412.6980
 Kitaura, F.-S., & Heß, S. 2013, *MNRAS*, 435, L78
 Kitaura, F.-S., Yepes, G., & Prada, F. 2014, *MNRAS*, 439, L21
 Kuruvilla, J. 2021, arXiv:2109.13938
 Kuruvilla, J., & Aghanim, N. 2021, arXiv:2102.06709
 Lawrence, E., Heitmann, K., Kwan, J., et al. 2017, *ApJ*, 847, 50
 Lawrence, E., Heitmann, K., White, M., et al. 2010, *ApJ*, 713, 1322
 Li, Y., Ni, Y., Croft, R. A. C., et al. 2021, *PNAS*, 118, e2022038118
 Maksimova, N. A., Garrison, L. H., Eisenstein, D. J., et al. 2021, *MNRAS*, 508, 4017
 Massara, E., Villaescusa-Navarro, F., Ho, S., Dalal, N., & Spergel, D. N. 2021, *PhRvL*, 126, 011301
 Massara, E., Villaescusa-Navarro, F., & Viel, M. 2014, *JCAP*, 2014, 053
 McClintock, T., Rozo, E., Banerjee, A., et al. 2019a, arXiv:1907.13167
 McClintock, T., Rozo, E., Becker, M. R., et al. 2019b, *ApJ*, 872, 53
 Milletari, F., Navab, N., & Ahmadi, S.-A. 2016, arXiv:1606.04797
 Modi, C., Lanusse, F., & Seljak, U. 2020, arXiv:2010.11847
 Monaco, P., Theuns, T., & Taffoni, G. 2002, *MNRAS*, 331, 587
 Ni, Y., Li, Y., Lachance, P., et al. 2021, *MNRAS*, 507, 1021
 Nishimichi, T., Takada, M., Takahashi, R., et al. 2019, *ApJ*, 884, 29
 Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv:1912.01703
 Samushia, L., Slepian, Z., & Villaescusa-Navarro, F. 2021, arXiv:2102.01696
 Scoccimarro, R., & Sheth, R. K. 2002, *MNRAS*, 329, 629
 Stein, G., Alvarez, M. A., & Bond, J. R. 2019, *MNRAS*, 483, 2236
 Tassev, S., Eisenstein, D. J., Wandelt, B. D., & Zaldarriaga, M. 2015, arXiv:1502.07751
 Tassev, S., Zaldarriaga, M., & Eisenstein, D. J. 2013, *JCAP*, 2013, 036

- Tosone, F., Neyrinck, M. C., Granett, B. R., Guzzo, L., & Vittorio, N. 2021, [MNRAS](#), **505**, 2999
- Uhlemann, C., Friedrich, O., Villaescusa-Navarro, F., Banerjee, A., & Codis, S. 2020, [MNRAS](#), **495**, 4006
- Valogiannis, G., & Dvorkin, C. 2021, arXiv:[2108.07821](#)
- Villaescusa-Navarro, F., Hahn, C., Massara, E., et al. 2020, [ApJS](#), **250**, 2
- Villaescusa-Navarro, F., Marulli, F., Viel, M., et al. 2014, [JCAP](#), **3**, 011
- White, M., Tinker, J. L., & McBride, C. K. 2013, [MNRAS](#), **437**, 2594
- Wright, B. S., Winther, H. A., & Koyama, K. 2017, [JCAP](#), **2017**, 054
- Zhai, Z., Tinker, J. L., Becker, M. R., et al. 2019, [ApJ](#), **874**, 95