



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Michigan Tech Publications

6-12-2023

Learning Neutrino Effects in Cosmology with Convolutional Neural Network

Elena Giusarma
Simons Foundation

Mauricio Reyes
Michigan Technological University, mrhurtad@mtu.edu

Francisco Villaescusa-Navarro
Simons Foundation

Siyu He
Simons Foundation

Shirley Ho
Simons Foundation

See next page for additional authors

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Physics Commons](#)

Recommended Citation

Giusarma, E., Reyes, M., Villaescusa-Navarro, F., He, S., Ho, S., & Hahn, C. (2023). Learning Neutrino Effects in Cosmology with Convolutional Neural Network. *Astrophysical Journal*, 950(1). <http://doi.org/10.3847/1538-4357/accd61>

Retrieved from: <https://digitalcommons.mtu.edu/michigantech-p/17256>

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>






Part of the [Physics Commons](#)

Authors

Elena Giusarma, Mauricio Reyes, Francisco Villaescusa-Navarro, Siyu He, Shirley Ho, and Chang Hoon Hahn



Learning Neutrino Effects in Cosmology with Convolutional Neural Network

Elena Giusarma^{1,2} , Mauricio Reyes², Francisco Villaescusa-Navarro^{1,3} , Siyu He¹, Shirley Ho^{1,3}, and ChangHoon Hahn^{4,5} ¹Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, 10010, New York, NY, USA; egiusarm@mtu.edu²Department of Physics, Michigan Technological University, Houghton, MI 49931, USA³Department of Astrophysical Sciences, Princeton University, Peyton Hall, Princeton, NJ 08544, USA⁴Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley CA 94720, USA⁵Berkeley Center for Cosmological Physics, University of California, Berkeley CA 94720, USA

Received 2019 November 8; revised 2023 March 12; accepted 2023 April 13; published 2023 June 12

Abstract

Measuring the sum of the three active neutrino masses, M_ν , is one of the most important challenges in modern cosmology. Massive neutrinos imprint characteristic signatures on several cosmological observables, in particular, on the large-scale structure of the universe. In order to maximize the information that can be retrieved from galaxy surveys, accurate theoretical predictions in the nonlinear regime are needed. Currently, one way to achieve those predictions is by running cosmological numerical simulations. Unfortunately, producing those simulations requires high computational resources—several hundred to thousand core hours for each neutrino mass case. In this work, we propose a new method, based on a deep-learning network (D^3M), to quickly generate simulations with massive neutrinos from standard Λ CDM simulations without neutrinos. We computed multiple relevant statistical measures of deep-learning generated simulations and conclude that our approach is an accurate alternative to the traditional N -body techniques. In particular the power spectrum is within $\simeq 6\%$ down to nonlinear scales $k = 0.7 \text{ h Mpc}^{-1}$. Finally, our method allows us to generate massive neutrino simulations 10,000 times faster than the traditional methods.

Unified Astronomy Thesaurus concepts: [Cosmology \(343\)](#); [Cosmological parameters from large-scale structure \(340\)](#); [Cosmological neutrinos \(338\)](#); [Large-scale structure of the universe \(902\)](#); [N-body simulations \(1083\)](#)

1. Introduction

The measurement of the mass and the ordering of neutrinos constitute the necessary steps to understand the physics of those particles and the beyond Standard Model mechanism. The discovery of neutrino oscillations (Fukuda et al. 1998; Ahmad et al. 2002; Araki et al. 2005; Adamson et al. 2008; Abe et al. 2012; Ahn et al. 2012; Abe et al. 2014; Forero et al. 2014) have robustly established that at least two out of the three active neutrinos (electron, muon, and tau) are massive. However, oscillation experiments only provide bounds on the neutrino mass squared differences. In the minimal neutrino scenario, the best-fit values for the solar and atmospheric mass splitting are $\Delta m_{21}^2 \equiv m_2^2 - m_1^2 \simeq 7.6 \times 10^{-5} \text{ eV}^2$ and $|\Delta m_{31}^2| \equiv |m_3^2 - m_1^2| \simeq 2.5 \times 10^{-3} \text{ eV}^2$ (Forero et al. 2014; Gonzalez-Garcia et al. 2014; Esteban et al. 2017). Since the sign of the atmospheric mass splitting is still unknown, we have two possible orderings of neutrino masses: normal ($\Delta m_{31}^2 > 0$) and inverted ($\Delta m_{31}^2 < 0$). The lower limits on the sum of neutrino masses set by oscillation experiments are $M_\nu \gtrsim 0.06 \text{ eV}$ in the normal hierarchy and $M_\nu \gtrsim 0.1 \text{ eV}$ in the inverted hierarchy. The study of the end point energy of electrons produced in β -decay places an upper limit on the total neutrino mass at the level of $M_\nu \lesssim 1.1 \text{ eV}$ (Kraus et al. 2005; Aker et al. 2019).

Cosmology provides an independent tool to constrain the neutrino masses and ordering. Neutrinos are the second-most abundant particles in the universe, after photons, and leave distinctive signatures on several cosmological observables.

They strongly affect the background evolution of the universe, as well as the evolution of cosmological perturbations. The light massive neutrinos are relativistic in the early universe and contribute to the radiation energy density. At late time, they became nonrelativistic,⁶ contributing to the total matter density of the universe. The nonrelativistic neutrinos behave as hot dark matter (HDM), possessing relatively large thermal velocities compared to any other massive particles. They thus cluster only on scales larger than their free-streaming length,⁷ suppressing the growth of perturbations on smaller scales. The presence of massive neutrinos also affects the anisotropies of the cosmic microwave background (CMB) as these particles may turn nonrelativistic around the decoupling period and gravitational lensing of the CMB. The current upper bound on M_ν is obtained by combining different cosmological data. In particular, the tightest constraints on M_ν , which arise by combining CMB temperature and polarization anisotropies measurements with different observations of the large-scale structure of the universe, range from 0.22 to 0.12 eV at 95% confidence level (Palanque-Desabrouille et al. 2015; Giusarma et al. 2016; Vagnozzi et al. 2017; Giusarma et al. 2018; Planck Collaboration et al. 2020b; Zennaro et al. 2018; Planck Collaboration et al. 2020a).

Future cosmological surveys, such as Euclid (Laureijs et al. 2011), DESI (Aghamousa et al. 2016), WFIRST (Spergel et al. 2015), Vera Rubin Observatory (VRO; Ivezić et al. 2019), and CMB-S4 (Abazajian et al. 2016), are expected to improve the constraints on the cosmological parameters. In particular,

⁶ The neutrino nonrelativistic transition takes place at $1 + z_{\text{nr}} \simeq 1890(m_\nu/1) \text{ eV}$.

⁷ The free-streaming scale is defined as the scale below which the growth of neutrino perturbations is strongly suppressed. It can be written as $k_s \simeq 0.018 \sqrt{\Omega_m} (M_\nu/1 \text{ eV})^{1/2} \text{ h Mpc}^{-1}$.

upcoming galaxy surveys will provide one of the most important sources of information to determine the neutrino masses and their mass ordering. To achieve those goals, accurate theoretical predictions for the spatial distribution of matter and luminous tracers in the presence of massive neutrino will be crucial. A powerful tool to obtain rigorous predictions in cosmology is through cosmological simulations. In particular, N -body simulations are numerical simulations where cosmological matter fluctuations are evolved under only gravity. They allow us to make comparisons between observations and theory. They are used to generate mock galaxy catalogs and to compute error covariance matrices. Lastly, they are also used to optimize observational strategies. Over the past years, a large number of N -body simulations, including massive neutrinos, have been proposed and developed in the literature. Those simulations have made it possible to study the impact of neutrino masses on clustering in fully nonlinear scale in real space (Bird et al. 2012; Lesgourgues & Pastor 2012; Villaescusa-Navarro et al. 2013, 2014; Peloso et al. 2015), on clustering and abundance of halo and cosmic voids (Castorina et al. 2014, 2015; Massara et al. 2015), and finally on clustering of matter in redshift space (Villaescusa-Navarro et al. 2018; Bel et al. 2019). However, simulations are computationally expensive. The development of new computational methods is thus needed to accelerate this process.

In the last decade, deep-learning approaches have been applied to many different fields of research, achieving outstanding results, especially in computer vision and image recognition challenges (Krizhevsky et al. 2012). Deep learning is a branch of machine learning that makes use of deep neural networks. An increasing number of studies are also adopting these methods for a variety of problems in cosmology. For example, Ravanbakhsh et al. (2017), He et al. (2018), and Ntampaka et al. (2019) applied convolutional neural networks (CNNs) to estimate the cosmological parameters directly from simulated dark matter distributions and the distribution of photons in CMB; He et al. (2019) built a deep CNN to predict the nonlinear structure formation of the universe from simple linear perturbation theory; Rodriguez et al. (2018) and Zamudio-Fernandez et al. (2019) used generative adversarial networks (GANs) models to synthesize samples of the cosmic web and predict high-resolution 3D distributions of cosmic neutral hydrogen; Ramanah et al. (2019) introduced a novel halo-painting network to map the cosmic density fields of dark matter to realistic halo distributions; Hezaveh et al. (2017) exploited the CNNs to estimate the strong gravitational lensing parameters accurately; and finally, List et al. (2019) applied conditional GANs to predict how changing the underlying physics alters the simulation results.

In this paper, we apply a modified version of the deep neural network proposed by He et al. (2019) to establish the mapping between simulations with massless and massive neutrinos. The important advantage of our approach consists in the ability to produce complex numerical simulations from the standard ones in a fraction of few minutes using a modern graphics processing unit (GPU) and with the same statistical properties as the standard N -body techniques.

The paper is organized as follows. In Section 2 we briefly describe the cosmological simulations we have used in this work. We then present a description of the training process and the quantitative results in Sections 3 and 4. Finally, we draw our conclusions in Section 5.

2. N -body Simulation Data

In this section, we briefly describe the N -body simulations used in our work. We refer the reader to Villaescusa-Navarro et al. (2018; Villaescusa-Navarro et al. 2020) for further details on the simulations.

We use two subsets of HADES simulations (Villaescusa-Navarro et al. 2018), the precursor of the QUIJOTE simulations (Villaescusa-Navarro et al. 2020): the standard Λ CDM simulations (without neutrinos) and simulations with massive neutrinos. The value of the cosmological parameters are $\Omega_m = \Omega_{\text{cdm}} + \Omega_b + \Omega_\nu = 0.3175$, $\Omega_b = 0.049$, $h = 0.6711$, $\Omega_\Lambda = 0.6825$, $n_s = 0.9624$, and $A_s = 2.13$. The simulations with massive neutrinos have $M_\nu = 0.15$ eV, and since A_s is fixed in both sets, the value of σ_8 is different: $\sigma_8 = 0.833$, and $\sigma_8 = 0.798$ for massless and massive neutrinos, respectively. All the simulations are run in a periodic box size of $1 \text{ h}^{-1} \text{ Gpc}$ and follow the evolution of the cold dark matter (CDM) and neutrino particles (only in the case of simulations with massive neutrinos) from $z = 99$ to $z = 0$. We use 100 independent realizations for each model containing the same number of CDM and neutrino particles ($N_{\text{cdm}} = N_\nu = 512^3$). We focus our analysis at redshift $z = 0$, corresponding to the present time.

In order to speed up the process of training the neural network and to avoid GPU memory problems, we split each of the 100 simulations into subcubes of size 32^3 voxels in Lagrangian space,⁸ corresponding to regions of size $62.5 \text{ h}^{-1} \text{ Mpc}$. Each realization thus contains 4096 subcubes, making a total of 409,600 voxels for each simulation model. We then split the realizations of each model into three chunks: 70% for training (70 realizations), 20% for validation (20 realizations), and 10% (10 realizations) for testing. After the training and the testing, we concatenate the subcubes of each of the 10 realizations in the test data in a cube of size $1 \text{ h}^{-1} \text{ Gpc}$, and we compute the relevant summary statistics.

3. Model

3.1. Model and Training

We use a modified version of the deep neural network, Deep Density Displacement Model (D^3M), introduced by He et al. (2019) with 15 convolutions plus deconvolution layer. D^3M is a generalization of the standard U-Net, first proposed in Ronneberger et al. (2015) for biomedical image segmentation, to work with three-dimensional data. A brief description of the architecture is shown in Appendix A.

Following He et al. (2019), we train our model using the displacement field rather than density field (see He et al. 2019 for more details). The displacement field is defined as

$$\mathbf{d} = \mathbf{x}_f - \mathbf{x}_i, \quad (1)$$

where \mathbf{x}_f is the final position of the particles (at redshift $z = 0$) and \mathbf{x}_i is the Lagrangian position of the same particles on a uniform grid. The input and target⁹ of the neural network is the displacement field from simulations without and with massive neutrinos, respectively.

⁸ Notice that the initial conditions of our simulations are generated from a regular grid.

⁹ In machine learning the target is defined as the desired response we are trying to predict. In particular, the training is performed by evaluating the model using a data sample for which we already know the result.

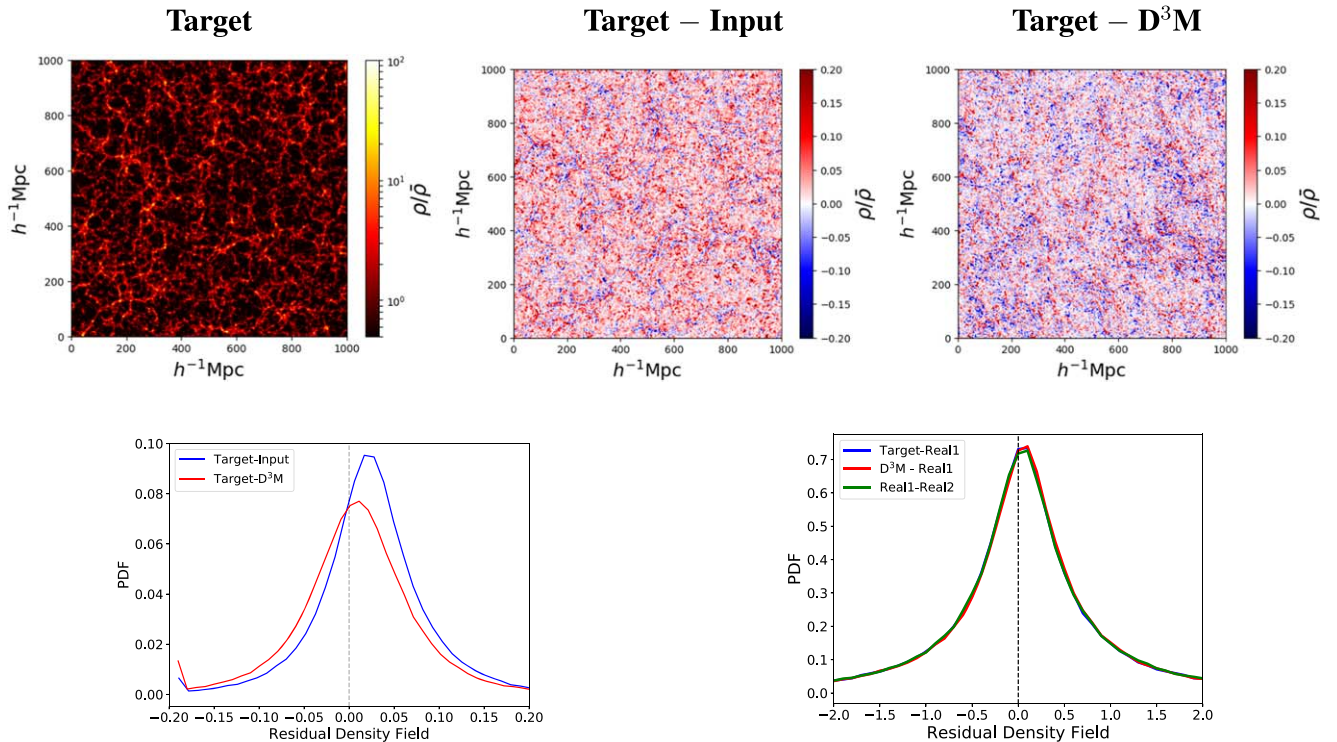


Figure 1. The top left panel shows the CDM density field at $z = 0$ from a region of $1000 \times 1000 \times 15$ ($h \text{ Mpc}$)⁻³ for a massive neutrino model ($M_\nu = 0.15 \text{ eV}$). The top center panel shows the residuals between the target (massive neutrino simulations) and the simulations with massless neutrinos (the input); the top right panel depicts the residual between the target and D³M. The two bottom panels illustrate the probability density function (PDF) of the residual density field as described in the text.

Our benchmark model consists of the Zel’dovich approximation (hereafter ZA) applied to massive neutrino models. The ZA is a significantly faster approximation to full N -body simulations produced by first-order perturbation theory with neutrinos. The reason why we use the ZA and not the second-order Lagrangian perturbation theory (2LPT) is because it is currently unknown how to predict 2LPT in massive neutrino models. In particular, there is no estimate for the two significant quantities, the second-order scale-dependent growth factor, and growth rate, necessary to calculate the 2LPT in presence of massive neutrinos.

We train our neural network by using the mean absolute error loss function that is given by the sum of all the absolute differences between the target displacement field and the predicted values:

$$\mathbb{L} = \sum_{i=1}^N |\mathbf{d}_i^t - \mathbf{d}_i^p|, \quad (2)$$

where \mathbf{d}_i^t is the true displacement field (from massive neutrino simulations), \mathbf{d}_i^p is the prediction from the D³M, and N is the total number of particles.

The total number of trainable parameters is 8.4×10^6 .

Figure 1 shows the projected density field of N -body simulations with massive neutrinos at $z = 0$ (top left panel), together with the residuals between the target and the input (top center panel) and the target and the output of our neural network (top right panel). From the top left panel, we can clearly see the presence of the filaments, surrounded by large underdense regions (dark regions), called voids, in between. The presence of massive neutrinos in the simulations induces a

suppression in the growth of density perturbations on small scales, visible in the matter power spectrum.

The bottom left panel of Figure 1 shows the PDF of the residuals from “Target–Input” (blue line) and “Target–D³M” (red line). By computing a Q–Q plot test, we established that both curves approximately follow a Gaussian distribution. However, the blue curve exhibits a shift induced by the presence and the absence of neutrinos in the simulations. In order to verify if the statistical variations between the simulations is reflected in the D³M prediction, we also consider a different simulation instance in the training data with $M_\nu = 0.15 \text{ eV}$; see bottom right panel of Figure 1. In particular, by comparing the residuals between “D³M–realization 1” (red line), “Target–realization 1” (blue line), and “realization 1–realization 2” (green line), we can note that the three residual distributions present the same shape and width. This is an indication that the density residuals coming from our D³M model have similar statistical properties as the simulations with the same neutrino mass.

3.2. Training Time

One standard cosmological simulation without neutrinos currently requires ~ 500 CPU hours to complete. The addition of massive neutrinos increases the simulation time to ~ 700 CPU hours. The key advantage of the approach we describe in this work is its ability to reduce the additional computational cost of producing massive neutrino simulations. In particular, it requires ~ 10 GPU hours to train and only few minutes to generate one simulation with massive neutrinos: a substantial gain in computing time compared to standard N -body techniques. We run our CNN model on two GPUs, 320

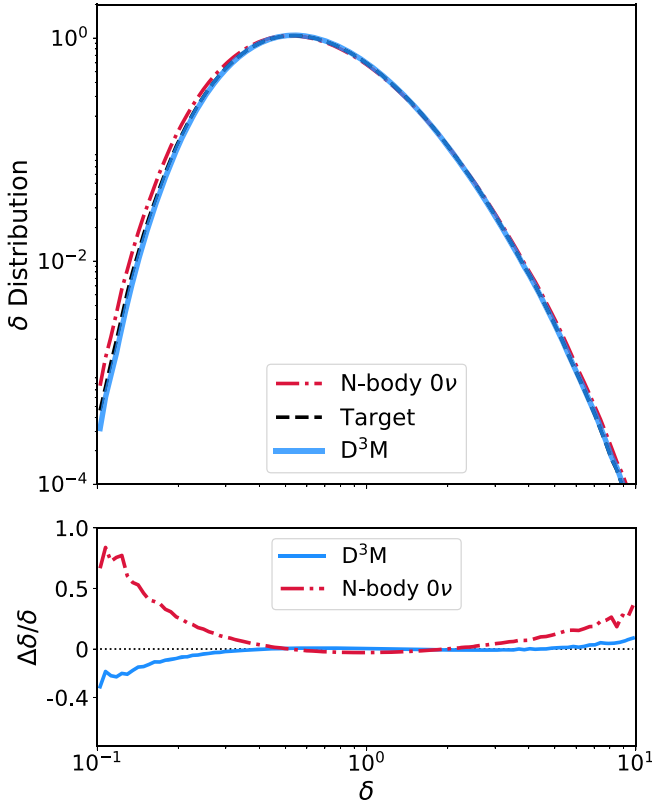


Figure 2. The top panel shows the normalized probability distribution function of CDM overdensities. The red and black lines show the overdensity distribution for N -body simulations with massless (the input of our D^3M) and massive neutrinos (target), respectively. The blue line depicts the D^3M results. The bottom panel illustrates the residual for the D^3M with respect to the simulations with massive neutrinos.

NVIDIA P100-16 GB. This allows us to speed up the training time by a factor of 2.5.

4. Results

In this section, we present the main results obtained by using the D^3M architecture and compare them with the standard N -body simulations and the benchmark model. We quantify the agreement between our model and numerical simulations by considering four different summary statistics: the overdensity distribution function, the power spectrum, the bispectrum, and the void size function. By doing this, we can compare the properties of the fully three-dimensional density distributions. We compute the considered statistics by concatenating the subcubes of each of the ten realizations in the test data and calculating the mean summary statistics over ten simulations.

4.1. 1D Probability Distribution Function

We compute the overdensity distribution of the CDM for the N -body simulations and D^3M prediction. We first place the particles on a grid with 512^3 cells, and we smooth the density field with a top-hat filter on a scale of $10 \text{ Mpc } h^{-1}$. We then calculate the distribution as the fraction of cells with overdensity lying within a given interval. The probability distribution function provides information on the distribution of CDM overdensities in the cells.

The results are depicted in Figure 2, where we show (upper panel) the comparison of the overdensity distribution among the standard simulations (the input; red dashed-dotted line),

massive neutrino simulations (the target; black dashed line), and our D^3M model (blue line).¹⁰ The bottom panel of Figure 2 shows the residual of the neural network with respect to the target. The residual is approximately 0 for $0.3 < \delta < 6$ and differs from zero by 15% and 6% for $\delta \approx 0.15$ and 9, respectively. Those results indicate a good agreement between the D^3M and the target.

4.2. Power Spectrum

The most widely used statistic in cosmology to extract information from cosmological observations is the two-point correlation function (2PCF) $\xi(r)$, which is defined as the excess probability, compared with a random distribution of galaxies, of finding a pair of galaxies at a given separation. The Fourier transform of the 2PCF is the power spectrum, $P(k)$:

$$\begin{aligned} \xi(r) &= \langle \delta(\mathbf{r}') \delta(\mathbf{r}' + \mathbf{r}) \rangle \\ P(k) &= \int d^3r \xi(r) e^{i\mathbf{k}\cdot\mathbf{r}}, \end{aligned} \quad (3)$$

where k is wavenumber of a fluctuation, which is related to the wavelength λ by $k = 2\pi/\lambda$, and $\delta(\mathbf{r})$ is the overdensity at position \mathbf{r} defined as $\delta = \rho/\bar{\rho} - 1$ with $\bar{\rho}$ denoting the mean density.¹¹ The power spectrum is the quantity predicted directly by theories for the formation of large-scale structure, and in the case of a Gaussian density field, it provides a complete statistical description of its fluctuations.

In order to quantify the performance of our model against the target, we compute the transfer function $T(k)$, defined as the square root of the ratio of the two power spectra:

$$T(k) = \sqrt{\frac{P_{\text{pred}}(k)}{P_{\text{target}}(k)}}, \quad (4)$$

where $P_{\text{pred}}(k)$ and $P_{\text{target}}(k)$ are the power spectra of the density field predicted by the D^3M and the target, respectively.

Figure 3 shows the average power spectrum and the transfer function of the density field over ten simulations. The red and black lines in the top panel correspond to the power spectrum of N -body simulations without and with massive neutrinos. We note that massive neutrinos induce a suppression of the power spectrum at small scales (larger k). This is the most significant effect of neutrino masses on the several cosmological observables. Neutrinos, being hot thermal relics, possess large velocity dispersion. Consequently, the nonrelativistic neutrino overdensities will only cluster at wavelengths larger than their free-streaming wavenumber, reducing the growth of CDM fluctuations on small scales.

The blue and green lines depict the power spectrum from the D^3M and benchmark model (ZA). Notice that the latter reproduces the power spectrum accurately on very large scales ($k < 0.05 h \text{ Mpc}^{-1}$). This is what we expect since we are using the ZA and not 2LPT (see Section 3.1).

We find that the density distribution from the D^3M sample has, on large scales, an average power spectrum that is very close in amplitude and in shape to that of N -body simulations with

¹⁰ Here the overdensity is defined as the ratio of the density field smoothed over a scale of 10 Mpc and the global average density for the simulation.

¹¹ In Equation (3), we make the assumption that the density field is statistically homogeneous and isotropic. This implies that the correlation function, $\xi(\mathbf{r}) = \xi(r)$, is a function only of the scalar separation $r = |\mathbf{r} - \mathbf{r}'|$ of the points \mathbf{r} and \mathbf{r}' , not of their overall location or orientation.

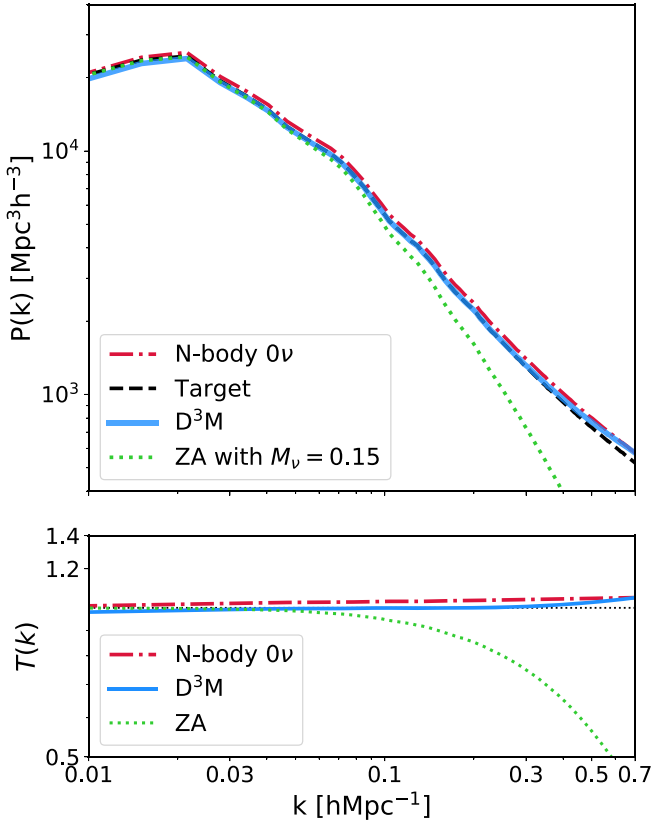


Figure 3. The top panel shows the matter power spectrum comparison among the standard N -body simulations (the input of the D^3M ; red dashed–dotted line), massive neutrino simulations (the target; black dashed line), CNN model dubbed D^3M (blue line), and ZA (the benchmark model; green dotted line). The bottom panel depicts the transfer function for the input, the D^3M , and the benchmark model.

neutrinos (the target). On smaller scales, the power spectrum of our model departs from that of the target. To quantify this deviation, we focus on the bottom panel of Figure 3, which shows the transfer function as a function of wavenumber for the benchmark model, the D^3M , and the input. For an entirely accurate prediction, the transfer function is expected to be 1. The transfer function of the ZA approximation is close to 1 up to $k < 0.05 \text{ h Mpc}^{-1}$ and then decreases as expected. On the other hand, the transfer function of the D^3M prediction is approximately 1 for $0.03 < k < 0.5 \text{ h Mpc}^{-1}$ and differs from the unity by 3.5% on scales $k \approx 0.55 \text{ h Mpc}^{-1}$. This discrepancy increases to 6.3% as k increases around 0.7 h Mpc^{-1} . Those results suggest that our model manages to predict the power spectrum accurately with massive neutrinos from large to intermediate scales. On smaller scales ($k > 0.7 \text{ h Mpc}^{-1}$) the prediction starts to deviate from the target. This is not a surprise since those are the scales where the effects of nonlinear evolution are important, and the mapping becomes highly nontrivial.

In order to test the robustness of our model, in Appendix B we also show the results for the case of neutrinos with mass $M_\nu = 0.6 \text{ eV}$.

4.3. Bispectrum

The inflationary scenario predicts Gaussian initial conditions. The statistical properties of the universe’s density field should be fully characterized by the 2PCF or the power spectrum. However, on small scales, nonlinear gravitational instability induces non-Gaussian signatures in the mass

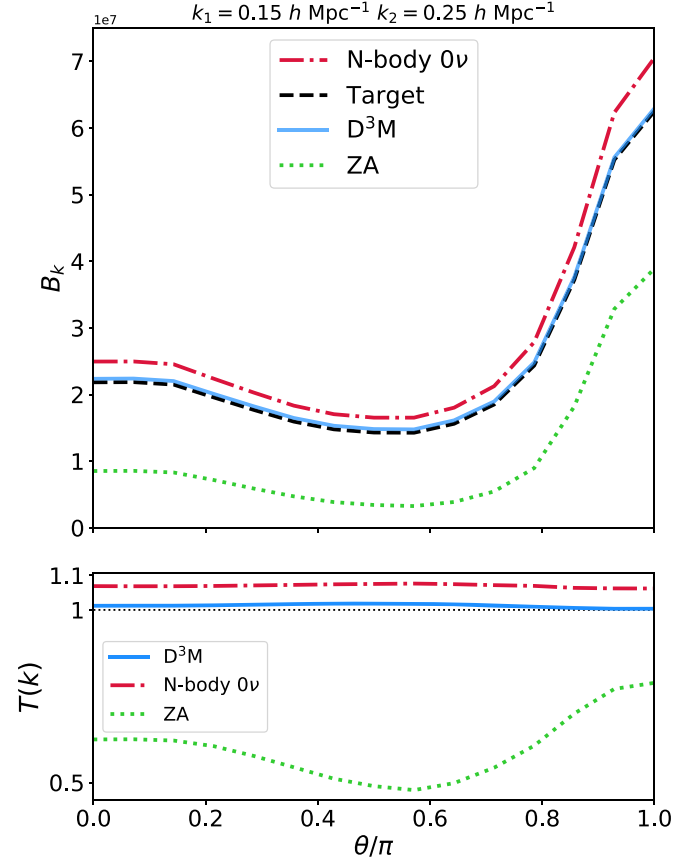


Figure 4. The plot shows the bispectrum predicted by standard Λ CDM simulations (the input of our architecture; red dashed–dotted line), massive neutrino simulations (the target; black dashed line), CNN model (blue line), and ZA (the benchmark model; green dotted line). We set $k_1 = 0.15 \text{ h Mpc}^{-1}$ and $k_2 = 0.25 \text{ h Mpc}^{-1}$. Notice that the bispectrum of the D^3M model is very close to that of the target.

distribution, which contain information on the nature of gravity and the dark matter. The lowest order statistical tool to describe a non-Gaussian field is the three-point correlation function, or equivalently, its Fourier transform, the bispectrum, defined as

$$B(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3) = \langle \delta_{\mathbf{k}_1} \delta_{\mathbf{k}_2} \delta_{\mathbf{k}_3} \rangle. \quad (5)$$

The bispectrum plays an important role in cosmology because it carries information about the spatial coherence of large-scale structures and can place strong constraints on models of structure formation. The presence of massive neutrinos induces a suppression of the amplitude of the bispectrum (Ruggeri et al. 2018). Furthermore, Hahn et al. (2019) have shown that the bispectrum is a powerful probe in constraining the total neutrino mass. In particular, the authors have confirmed that such summary statistics can break the degeneracies between M_ν and the other cosmological parameters.

Figure 4 shows the average bispectrum over ten simulations for our D^3M (blue line), the benchmark model (green line), the input (red line), and the target (black line). We have selected a value of $k_1 = 0.15 \text{ h Mpc}^{-1}$ and of $k_2 = 0.25 \text{ h Mpc}^{-1}$ and varied the angle among those two vectors. Those scales correspond to the maximum wavenumbers adopted in cosmology to infer the cosmological parameters from the galaxy power spectrum measurements.

We find that the mean relative bispectrum residual of our model compared to the target is 0.4%. This suggests that the

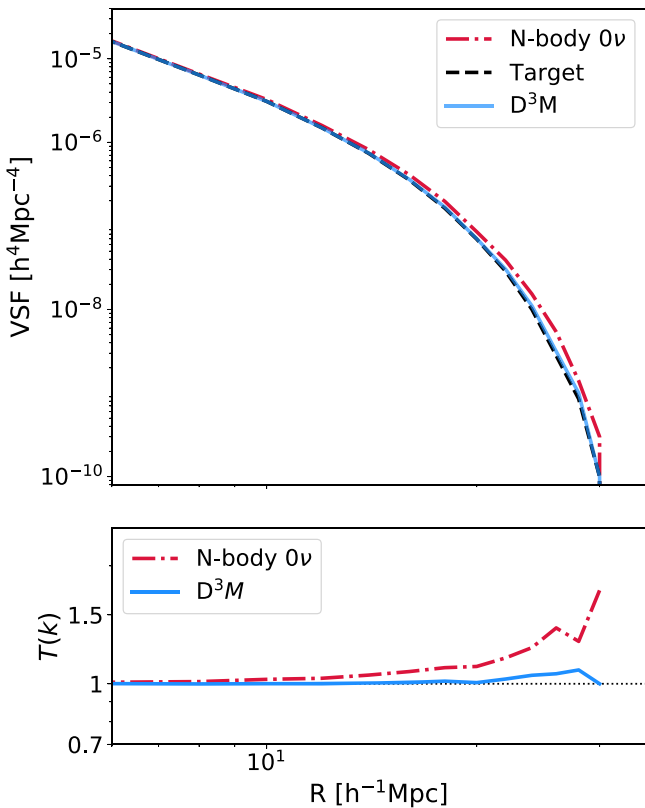


Figure 5. The upper panel shows the void size function for the input (standard N -body simulations red dashed–dotted line), target (massive neutrino simulations; black dashed line), and D^3M (blue line). The bottom panel depicts the transfer function for the input and the neural network model.

D^3M model can generate neutrino simulations with correct higher-order statistics in the mildly nonlinear regime. Notice that, on the other hand, the benchmark model is not able to reproduce the bispectrum from the target. This is not surprising as the validity of the ZA will be limited to large scales.

4.4. Void Abundance

Voids are the most underdense regions of our universe and together with halos, filaments, and walls, constitute the large-scale structure of the universe, known as the cosmic web. Voids enclose a large amount of cosmological information since they fill most of the volume of the universe. Massive neutrinos modify the properties of cosmic voids, such as their number, size, and shape. This occurs because the large thermal velocities of massive neutrinos prevent their evacuation from cosmic voids. Consequently, such additional mass within voids affects their evolution.

The last statistical tool we consider in this section is the void size function, defined as the abundance of voids as a function of the void’s radius. In the presence of massive neutrinos, the abundance of large voids is suppressed. This occurs because the additional neutrino masses inside the voids slow down their evolution and make voids smaller and denser (Massara et al. 2015; Kreisch et al. 2019; Pisani et al. 2019). Here we used the algorithm described in Banerjee & Dalal (2016) in order to identify voids in the galaxy distribution of our D^3M , the target, and the input. In Figure 5, upper panel, we compare the average void size function over ten realizations from the D^3M (blue line), the input (red line), and the target (black line). We can see the suppression of the void’s abundance induced by massive

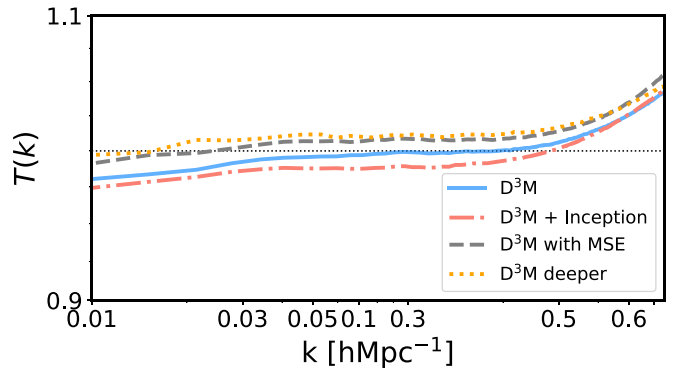


Figure 6. Transfer function comparison among the D^3M (blue line) used in the analysis and the modified version of the architecture obtained by including the inception module (pink dashed–dotted line), varying the loss function (gray dashed line), and increasing the number of layers of about the double of D^3M (orange dotted line).

neutrinos (blue and black lines) at larger radii. The bottom panel shows the transfer function as a function of radius for the input and the neural network model. We can note the good agreement in the void size function among the D^3M and the target up to $R \approx 20$ [h^{-1} Mpc]. The discrepancy from the unity increases by 8% as the radius increases around 30 [h^{-1} Mpc]. This indicates that our approach is also able to capture voids’ information at the relevant scales.

4.5. Additional Tests

Although our model reproduces the target’s power spectrum within 6.3% up to scales $k \approx 0.7$ h Mpc $^{-1}$, the accuracy degrades on smaller nonlinear scales. In order to improve the model efficiency at those scales, we perform some additional tests.

We first tune the hyperparameters in the neural network. In particular, we increase the number of the layers, change the learning rate and the batch size, and apply the mean square error (MSE) loss function.

We then test different CNN architectures. Specifically, we implement the D^3M by including an inception module after the first block in the downsampling and before the last block in the expansive path. The inception network (Szegedy et al. 2014) uses convolutions of different sizes to capture details at varied scales. In our case, we are interested in extracting detailed features on smaller scales. This motivates the use of the inception module at the beginning and at the end of the D^3M . In particular, we use the first original version of the inception architecture with three different sizes of filters (1×1 , 3×3 , and 5×5) and max pooling layers. The outputs are then concatenated and passed as input to the second/last block of the D^3M .

The results are depicted in Figure 6. We compare the transfer function of the additional tests we performed with the D^3M architecture used in our analysis. We can see that these tests do not significantly improve the performance of our model at smaller scales. We only find a slight improvement on larger scales when we consider the MSE loss function or a deeper D^3M architecture. Moreover the agreement for small scales is compromised in favor of the agreement for large scales when we use a deeper D^3M model. This demonstrates the stability of our model and shows that more accurate trials are needed to improve the predictions on nonlinear scales. We leave this exploration to future work. It is lastly worth mentioning that in

order to speed up the learning process of our D^3M architecture, we parallelized the training across the data dimension using two GPUs. In particular, to train on multiple GPUs we used the PyTorch multi-GPU techniques data parallelism. By doing that, the two GPUs synchronize the model parameters (the weights) to ensure that they are training a consistent model. Unfortunately, such data parallelism suffers from excessive inter-GPU communication overheads due to frequent weight synchronization among GPUs. This process introduces some systematics during the training that affects the final prediction (see, for example, the orange dotted line in Figure 6). The development of new parallelization approaches to speed up the training process in deep neural networks and improve the predictions is one of major research area of study in machine learning (Krizhevsky 2014; Chen et al. 2018; Lin et al. 2018), and it is well beyond the scope of our paper.

5. Conclusions

Upcoming galaxy surveys, such as Euclid, DESI, DES, and LSST, are expected to reach the sensitivity necessary to detect the absolute scale of neutrino masses. To achieve this goal, precise theory predictions are required to analyze the data from those surveys and to understand and model the impact of nonlinearities in the matter power spectrum, galaxy bias, and redshift space distortion in the presence of massive neutrinos. A powerful way to obtain accurate theory predictions is by running numerical cosmological simulations. However, producing those simulations requires high computational resources.

In this work, we have presented a deep-learning approach to predict fast, nonstandard cosmological simulations with massive neutrinos from standard N -body simulations. We have quantified the performance of our D^3M model by considering four summary statistics often used in cosmology: the 1D PDF, the power spectrum, the bispectrum, and the void abundance. We have shown that our model is able to learn the mapping between the displacement field in simulations with massless and massive neutrinos, down to scales as small as $0.7 h/\text{Mpc}$ at $z=0$, and reproduce the effect of massive neutrinos in large-scale structures. Moreover, with our method, we can generate massive neutrino simulations 5 orders of magnitude faster than the traditional methods and dramatically reduce the computational costs necessary to implement simulations with massive neutrinos.

This work demonstrates that the deep-learning approach is an accurate alternative to the traditional N -body techniques by mapping simulations with massive neutrinos directly from simulations with massless neutrinos. The use of such approach will be particularly useful in the near future. The need for fast N -body simulations will increase with the upcoming large cosmological surveys such as Euclid and LSST. Our methods will thus be crucial for producing fast and large simulation data sets for cosmological analyses.

In future work, we will improve our model to capture the effect of massive neutrinos on smaller nonlinear scales by using the Quijote high-resolution simulations (Villaescusa-Navarro et al. 2020). Another direction will be to explore the possibility of generating fast neutrino simulations with an arbitrary mass from the standard ΛCDM simulations or more complex nonstandard cosmological simulations that include, for example, modified gravity or primordial non-Gaussianity effects (Novaes et al. 2015; Peel et al. 2019). We leave for future work the investigation of these different effects.

We note that recently, Zennaro et al. (2019) have presented a new method to rescale massless neutrino simulations to massive neutrino simulations, following a very different approach than the one used in this work. Using the output of that method as the input of our neural net has the potential to improve the accuracy of both our method and Zennaro et al.'s (2019) results, as neural nets train faster on residuals. We plan to study this in future work.

The HADES and Quijote simulations are publicly available at <https://franciscovillaescusa.github.io/hades.html> and <https://github.com/franciscovillaescusa/Quijote-simulations>, respectively. The trained models, predictions, and statistics extracted from the testing and extrapolation sets are hosted under the public github repository <https://github.com/egiusarma/nuD3M>, and the model has been trained using the code <https://github.com/siyucosmo/ML-Recon>.

Acknowledgments

We thank Gabriella Contardo, Elena Massara, Barnabas Pócsocz, Siamak Ravanbakhsh, and David Spergel for useful discussions. E.G. acknowledges Nick Carriero from the Flatiron Institute for help in solving technical problems with GPUs. The work of E.G., F.V.N., S.H., and S.H. is supported by the Simons Foundation. S.H. thanks NASA for their support with NASA grant 15-WFIRST15-0008 and NASA Research Opportunities in Space and Earth Sciences grant 12-EUCLID12-0004. The analysis of the simulations and training of the neural network has been carried out in the Rusty cluster of the Flatiron Institute.

Appendix A Method

In this section, we introduce a brief description of the CNNs and the D^3M architecture used in this work.

A.1. Convolutional Neural Networks

An artificial neural network is an interconnected group of nodes (neurons) that combine to learn complex patterns. Neural networks are modeled after biological neural network systems and attempt to allow computers to learn in a similar manner to humans. An artificial neuron receives one or more input x_i and multiplies them by a set of connection weights w_i plus a bias. The result of such operation is called net input:

$$z = \sum w_i x_i + b. \quad (\text{A1})$$

The output can be then calculated by applying the activation function over the net input. The activation function is a nonlinear function that evaluates if the neuron would fire or not. There exist different types of activation functions, but the rectified linear unit (ReLU) is the most widely used in neural networks nowadays. ReLU activation function is defined as

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } x \geq 0. \end{cases} \quad (\text{A2})$$

One of the greatest advantages of ReLU is that it does not activate all the neurons at the same time. It converts all negative inputs to zero, and the neuron does not get activated. This makes it more computationally efficient than other activation functions.

A particular class of neural networks that can be used to learn spatial features is called CNN. A CNN consists of an input and an output layer, as well as multiple hidden layers. The input consists of images that computers read as pixels and express them as matrices. The convolutional layer contains a set of learnable parameters (also known as filters or kernels). A filter is used to detect the presence of specific features or patterns present in the input. This filter is convolved across the input file, and a dot product is computed to give an activation map. The feature images produced by a convolutional layer are then passed to the next layer in the CNN.

One of the main problems we can encounter when training learning systems is a change in the distribution of the outputs of each hidden layer. Each input layer is affected by the model parameters of all preceding layers. As a consequence, a small change made within a layer can produce a large variation of the inputs of another layer and change feature maps' distribution. This means that later layers need to continuously adapt to the new distribution obtained from the previous one. This change can slow down the convergence of the learning algorithm. In order to improve the performance of the neural network, we can make use of the Batch normalization (BN) technique (Ioffe & Szegedy 2015). Batch normalization consists in normalizing the output of a previous activation layer during the training so that the input to the activation function across each training batch has a fixed mean (0) and a variance (1). Instead of using the entire data set to normalize activation functions, the normalization is restrained to each mini-batch in the training process.

A.2. The D^3M Architecture

We follow He et al. (2019), and we use a D^3M architecture that consists of three parts: the contracting path, a bottleneck, and the expansive path. During the contracting path, the spatial resolution is reduced, while the feature information is increased. The expansive path increases the spatial resolution and combines it with the feature information. The bottleneck is in between the contracting and expansive path and allows the model to learn a compression of the input data.

The contracting and expansive paths each consist of blocks of two 3^3 CNNs, while the bottom of the U-Net consists of two CNNs. Every convolution is followed by zero padding, BN, and ReLU. As in U-Net, at all resolution levels, with exception of the bottleneck, we concatenate the input of the down-sampling layers to the output of the up-sampling layers. This allows to transfer the small-scale spatial information from the contracting path to the expansive path, which would otherwise have poor spatial resolution. At the final layer, a 1^3 convolution is used to map 64 features to the final 3D data. We refer the reader to He et al. (2019) and Ronneberger et al. (2015) for a more detailed description of the architecture.

Appendix B Robustness of D^3M model

In order to test the robustness of the D^3M , we train our neural network by considering a different neutrino mass case. In particular, while the input still includes standard simulations without neutrinos, the new target consists of neutrino simulations with mass $M_\nu = 0.6$ eV. The results are depicted in Figure 7, where we show the power spectrum and the transfer function for the input, target, and the prediction of the neural

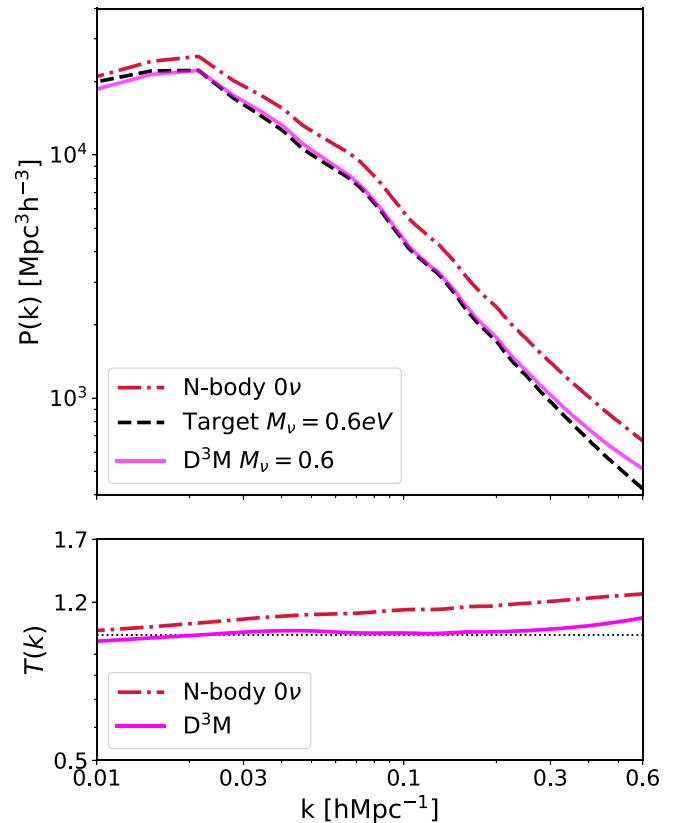





Figure 7. The top panel shows the matter power spectrum comparison among the standard N -body simulations (red dashed-dotted line), the new target (black dashed line), and the prediction (magenta line). The bottom panel depicts the transfer function for the input and the D^3M .

network. Also for this neutrino mass case, we can note a very good agreement in the power spectrum between the target and the D^3M on large scales and a deviation from the target on small nonlinear scales. In particular, the transfer function is approximately 1 for $0.03 < k < 0.45$ $h\text{Mpc}^{-1}$ and differs from the unity by 8% from scales $k \approx 0.5$ $h\text{Mpc}^{-1}$. Those results suggest that our model is applicable independently of neutrino mass, ensuring the robustness of our approach.

ORCID iDs

Elena Giusarma  <https://orcid.org/0000-0003-3052-3059>
Francisco Villaescusa-Navarro  <https://orcid.org/0000-0002-4816-0455>
ChangHoon Hahn  <https://orcid.org/0000-0003-1197-0902>

References

- Abazajian, K. N., Adshead, P., Ahmed, Z., et al. 2016, arXiv:1610.02743
Abe, Y., Aberle, C., dos Anjos, J. C., et al. 2012, *PhRvD*, **86**, 52008
Abe, K., Adam, J., Aihara, H., et al. 2014, *PhRvL*, **112**, 061802
Adamson, P., Andreopoulos, C., Arms, K. E., et al. 2008, *PhRvL*, **101**, 131802
Ahmad, Q. R., Allen, R. C., Andersen, T. C., et al. 2002, *PhRvL*, **89**, 011301
Ahn, J. K., Chebotaryov, S., Choi, J. H., et al. 2012, *PhRvL*, **108**, 191802
Aker, M., Altenmüller, K., Arenz, M., et al. 2019, *PhRvL*, **123**, 221802
Araki, T., Eguchi, K., Enomoto, S., et al. 2005, *PhRvL*, **94**, 081801
Banerjee, A., & Dalal, N. 2016, *JCAP*, **2016**, 015
Bel, J., Pezzotta, A., Carbone, C., Sefusatti, E., & Guzzo, L. 2019, *A&A*, **622**, A109
Bird, S., Viel, M., & Haehnelt, M. G. 2012, *MNRAS*, **420**, 2551
Castorina, E., Carbone, C., Bel, J., Sefusatti, E., & Dolag, K. 2015, *JCAP*, **2015**, 043

- Castorina, E., Sefusatti, E., Sheth, R. K., Villaescusa-Navarro, F., & Viel, M. 2014, *JCAP*, 2014, 049
- Chen, C.-C., Yang, C.-L., & Cheng, H.-Y. 2018, arXiv:1809.02839
- DESI Collaboration, Aghamousa, A., Aguilar, J., et al. 2016, arXiv:1611.00036
- Esteban, I., Gonzalez-Garcia, M. C., Maltoni, M., Martinez-Soler, I., & Schwetz, T. 2017, *JHEP*, 2017, 87
- Forero, D. V., Tórtola, M., & Valle, J. W. F. 2014, *PhRvD*, 90, 093006
- Fukuda, Y., Hayakawa, T., Ichihara, E., et al. 1998, *PhRvL*, 81, 1562
- Giusarma, E., Gerbino, M., Mena, O., et al. 2016, *PhRvD*, 94, 083522
- Giusarma, E., Vagnozzi, S., Ho, S., et al. 2018, *PhRvD*, 98, 123526
- Gonzalez-Garcia, M. C., Maltoni, M., & Schwetz, T. 2014, *JHEP*, 2014, 52
- Hahn, C., Villaescusa-Navarro, F., Castorina, E., & Scoccamarro, R. 2020, *JCAP*, 2020, 040
- He, S., Li, Y., Feng, Y., et al. 2019, *PNAS*, 116, 13825
- He, S., Ravanbakhsh, S., & Ho, S. 2018, in Analysis of Cosmic Microwave Background with Deep Learning (Vancouver, BC: ICLR) <http://openreview.net/forum?id=B15uoOyvyz>
- Hezaveh, Y. D., Perreault Levasseur, L., & Marshall, P. J. 2017, *Natur*, 548, 555
- Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, 873, 111
- Kodi Ramanah, D., Charnock, T., & Lavaux, G. 2019, *PhRvD*, 100, 043515
- Kraus, C., Bornschein, B., Bornschein, L., et al. 2005, *EPJC*, 40, 447
- Kreisch, C. D., Pisani, A., Carbone, C., et al. 2019, *MNRAS*, 488, 4413
- Krizhevsky, A. 2014, arXiv:1404.5997
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in NeurIPS Proc.: Advances in Neural Information Processing Systems 25 (NIPS 2012), ed. F. Pereira et al. (Red Hook, NY: Curran Associates, Inc.), 1097
- Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv:1110.3193
- Lesgourgues, J., & Pastor, S. 2012, arXiv:1212.6154
- Lin, Y., Han, S., Mao, H., Wang, Y., & Dally, B. 2018, in ICLR Conference Track: 6th International Conference on Learning Representation, <http://openreview.net/forum?id=SkhQHMW0W>
- List, F., Bhat, I., & Lewis, G. F. 2019, *MNRAS*, 490, 3134
- Massara, E., Villaescusa-Navarro, F., Viel, M., & Sutter, P. M. 2015, *JCAP*, 2015, 018
- Novaes, C., Bernui, A., Ferreira, I., & Wuensche, C. 2015, *JCAP*, 2015, 064
- Ntampaka, M., Eisenstein, D. J., Yuan, S., & Garrison, L. H. 2019, *ApJ*, 889, 151
- Palanque-Delabrouille, N., Yèche, C., Baur, J., et al. 2015, *JCAP*, 2015, 011
- Peel, A., Lalande, F., Starck, J.-L., et al. 2019, *PhRvD*, 100, 023508
- Peloso, M., Pietroni, M., Viel, M., & Villaescusa-Navarro, F. 2015, *JCAP*, 2015, 001
- Pisani, A., Massara, E., Spergel, D. N., et al. 2019, *BAAS*, 51, 40
- Planck Collaboration, Aghanim, N., Akrami, Y., et al. 2020a, *A&A*, 641, A5
- Planck Collaboration, Aghanim, N., & Akrami, Y. 2020b, *A&A*, 641, A1
- Ravanbakhsh, S., Oliva, J., Fromenteau, S., et al. 2017, arXiv:1711.02033
- Rodriguez, A. C., Kacprzak, T., Lucchi, A., et al. 2018, *ComAC*, 5, 4
- Ronneberger, O., Fischer, P., & Brox, T. 2015, in International Conference on Medical Image Computing and Computer-Assisted Intervention, 9351, ed. N. Navab et al. (Cham: Springer), 234, doi:10.1007/978-3-319-24574-4_28
- Ruggeri, R., Castorina, E., Carbone, C., & Sefusatti, E. 2018, *JCAP*, 2018, 003
- Spergel, D., Gehrels, N., Baltay, C., et al. 2015, arXiv:1503.03757
- Szegedy, C., Liu, W., Jia, Y., et al. 2014, arXiv:1409.4842
- Vagnozzi, S., Giusarma, E., Mena, O., et al. 2017, *PhRvD*, 96, 123503
- Villaescusa-Navarro, F., Banerjee, A., Dalal, N., et al. 2018, *ApJ*, 861, 53
- Villaescusa-Navarro, F., Marulli, F., Viel, M., et al. 2014, *JCAP*, 2014, 011
- Villaescusa-Navarro, F., Vogelsberger, M., Viel, M., & Loeb, A. 2013, *MNRAS*, 431, 3670
- Villaescusa-Navarro, F., Hahn, C., Massara, E., et al. 2020, *ApJS*, 250, 2
- Zamudio-Fernandez, J., Okan, A., Villaescusa-Navarro, F., et al. 2019, arXiv:1904.12846
- Zennaro, M., Angulo, R. E., Aricò, G., Contreras, S., & Pellejero-Ibáñez, M. 2019, *MNRAS*, 489, 5938
- Zennaro, M., Bel, J., Dossett, J., Carbone, C., & Guzzo, L. 2018, *MNRAS*, 477, 491