The Astrophysical Journal, 889:151 (15pp), 2020 February 1
https://doi.org/10.3847/1538-4357/ab5f5e

A Hybrid Deep Learning Approach to Cosmological Constraints from Galaxy Redshift Surveys

Michelle Ntampaka[1,2], Daniel J. Eisenstein[2], Sihan Yuan[2], and Lehman H. Garrison[2,3]
[1] Harvard Data Science Initiative, Harvard University, Cambridge, MA 02138, USA; michelle.ntampaka@cfa.harvard.edu
[2] Center for Astrophysics | Harvard & Smithsonian, Cambridge, MA 02138, USA
[3] Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA

Received 2019 September 25; revised 2019 December 2; accepted 2019 December 4; published 2020 February 3

Abstract

We present a deep machine learning (ML)–based technique for accurately determining σ_8 and Ω_m from mock 3D galaxy surveys. The mock surveys are built from the `AbacusCosmos` suite of *N*-body simulations, which comprises 40 cosmological volume simulations spanning a range of cosmological parameter values, and we account for uncertainties in galaxy formation scenarios through the use of generalized halo occupation distributions (HODs). We explore a trio of ML models: a 3D convolutional neural network (CNN), a power spectrum–based fully connected network, and a hybrid approach that merges the two to combine physically motivated summary statistics with flexible CNNs. We describe best practices for training a deep model on a suite of matched-phase simulations, and we test our model on a completely independent sample that uses previously unseen initial conditions, cosmological parameters, and HOD parameters. Despite the fact that the mock observations are quite small ($\sim$0.07 h^{-3} Gpc^3) and the training data span a large parameter space (six cosmological and six HOD parameters), the CNN and hybrid CNN can constrain estimates of σ_8 and Ω_m to $\sim$3% and $\sim$4%, respectively.

Unified Astronomy Thesaurus concepts: Cosmological parameters from large-scale structure (340); Large-scale structure of the universe (902); Astrostatistics (1882); Cosmology (343)

1. Introduction

In the ΛCDM cosmological model, tiny density fluctuations in the early universe evolved into today's cosmic web of overdense dark matter halos, filaments, and sheets. Imprinted on this large-scale structure is information about the underlying cosmological model, provided one knows how and where to look. Measurements that describe the large-scale distribution of matter in the universe carry information about the cosmological model that drove its formation. These measurements include descriptions of the spatial distribution and clustering of galaxies (e.g., Huchra et al. 1990; Shectman et al. 1996; Percival et al. 2001; Tegmark et al. 2004), the abundance of massive galaxy clusters (e.g., Vikhlinin et al. 2009; Mantz et al. 2015; de Haan et al. 2016), the weak gravitational lensing of galaxies by intervening large-scale structure (e.g., Bacon et al. 2000; Kaiser et al. 2000; Van Waerbeke et al. 2000; Wittman et al. 2000; DES Collaboration et al. 2018; Hildebrandt et al. 2020; Hikage et al. 2019), and the length scale of baryon acoustic oscillations (e.g., Cole et al. 2005; Eisenstein et al. 2005; Alam et al. 2017). A hallmark difference between these and probes of the earlier universe is non-Gaussianity; though the early universe is well described by a Gaussian random field (e.g., Planck Collaboration et al. 2014a, 2014b), gravitational collapse drives the formation of non-Gaussian correlations in the late-time matter distribution. See Weinberg et al. (2013) for a review of these and other observational cosmological probes.

Galaxies are hosted in dark matter halos and are tracers, albeit biased ones, of large-scale structure. Large spectroscopic surveys such as the Sloan Digital Sky Survey (SDSS; York 2000) have produced maps of the 3D distribution of galaxies in the universe, and upcoming spectroscopic surveys such as the Dark Energy Spectroscopic Instrument (DESI; DESI Collaboration et al. 2016), Subaru Prime Focus Spectrograph (PFS; Takada et al. 2014), 4 m Multi-Object Spectroscopic Telescope (4MOST; de Jong et al. 2014), and *Euclid* (Amendola et al. 2013) will produce exquisitely detailed maps of the sky. The galaxy power spectrum provides one handle on summarizing and interpreting these 3D galaxy maps and can be used to put constraints on the parameters that describe a ΛCDM cosmology (e.g., Tegmark et al. 2004), but care must be taken when disentangling the effects of cosmology and galaxy bias (e.g., Cacciato et al. 2013; More et al. 2013; van den Bosch et al. 2013).

Though it is an abundantly useful compression of the information contained in the distribution of galaxies, the power spectrum is not a complete accounting of this information because the late-time galaxy distribution is not a Gaussian random field. The deviations from Gaussian correlations are enormous at small length scales ($\lesssim$a few Mpc), where dark matter halos have collapsed and virialized and remain substantial at intermediate scales due to the cosmic web of filaments, walls, and voids. Additional statistics such as the squeezed three-point correlation function (Yuan et al. 2018a), redshift space power spectrum (Kobayashi et al. 2020), counts in cylinders (Wang et al. 2019), and minimum spanning tree (Naidoo et al. 2020) have been shown to be rich in complementary cosmological information by capturing non-Gaussian details of the galaxy distribution that are not described by the power spectrum alone.

These higher-order statistical descriptions of how galaxies populate 3D space typically need to be calibrated on cosmological simulations. Cosmological hydrodynamical simulations that trace the formation of galaxies are computationally expensive, so a more tractable approach is to use less expensive *N*-body simulations that have been populated with galaxies. This can be accomplished through a technique that matches galaxies to the simulated structure of dark matter, for example, through a halo occupation distribution (HOD; e.g., Peacock & Smith 2000; Scoccimarro et al. 2001; Berlind & Weinberg 2002; Zheng et al. 2005).

Under its simplest assumptions, an HOD uses halo mass as the sole property that determines whether a halo will host a particular type of galaxy. The breakdown of this assumption is known as galaxy assembly bias, which asserts that mass alone is insufficient and that additional environmental and assembly factors come into play. These factors include formation time (Gao et al. 2005) and halo concentration (Wechsler et al. 2006). Modern HOD implementations often provide flexibility to account for assembly bias (e.g., Hearin et al. 2016; Yuan et al. 2018b; Beltz-Mohrmann et al. 2019).

Machine learning (ML) offers a number of methods that can find and extract information from complex spatial patterns imprinted on the 3D distribution of galaxies. Therefore, ML is an enticing approach for inferring cosmological models in spite of myriad complicating effects. One promising class of tools for this task are convolutional neural networks (CNNs; e.g., Fukushima & Miyake 1982; LeCun et al. 1999; Krizhevsky et al. 2012; Simonyan & Zisserman 2014), which are often used in image recognition tasks. The CNNs employ many hidden layers to extract image features such as edges, shapes, and textures. Typically, CNNs pair layers of convolution and pooling to extract meaningful features from the input images, followed by deep, fully connected layers to output an image class or numerical label. Because these deep networks learn the filters necessary to extract meaningful information from the input images, they require very little image preprocessing. See Schmidhuber (2014) for a review of deep neural networks (NNs).

Traditionally, CNNs are applied to 2D images, which may be monochromatic or represented in several color bands. The 2D CNNs can extract information from non-Gaussianities in simulated convergence maps, remarkably improving cosmological constraints over a more standard statistical approach (e.g., Schmelzle et al. 2017; Gupta et al. 2018; Ribli et al. 2019a, 2019b), and recent work has extended this to put cosmological constraints on observations using CNNs (Fluri et al. 2019).

However, the application of CNNs is not limited to flat Euclidean images (e.g., Perraudin et al. 2019), nor is it limited to two dimensions. The algorithm can be extended to three dimensions, where the third dimension may be, for example, temporal (e.g., video input, as in Ji et al. 2013) or spatial (e.g., a data cube, as in Kamnitsas et al. 2016). Ravanbakhsh et al. (2017) employed the first cosmological application of a 3D CNN, showing that the tool can infer the underlying cosmological parameters from a simulated 3D dark matter distribution.

We present an application of 3D CNNs to learn estimates of cosmological parameters from simulated galaxy maps. Our hybrid deep learning architecture learns directly from the calculated 2D power spectrum and simultaneously harnesses non-Gaussianities by also learning directly from the raw 3D distribution of galaxies. In Section 2, we describe our mock observations: the suite of cosmological simulations in Section 2.1, the range of HODs applied to these simulations in Section 2.2, the training and validation mock observations in Section 2.3, and the carefully constructed and independent test mock observations at the *Planck* cosmology in Section 2.4. We describe our trio of deep learning architectures, including the hybrid method, in Section 3. We present our results in Section 4 and a discussion and conclusions in Section 5. The Appendix is more pedagogical in nature; it describes how the range of model predictions evolves with training and suggests new tests for assessing a model's fit.

2. Methods: Mock Observations

We use the `AbacusCosmos` suite of simulations[4] (Garrison et al. 2018, 2019) to create three data sets: a training set, a validation set, and a testing set. The training set is used to fit the ML model; it spans a range of CDM cosmological parameter combinations (or, simply, "cosmologies") and is populated with galaxies in a way that mimics a variety of galaxy formation models. The validation set is used to assess how well the ML model has fit; it also spans a range of cosmological parameters and galaxy formation models. The testing set is independent of both the training and validation sets; it is at the *Planck* fiducial cosmology (Planck Collaboration et al. 2016), built from simulations with initial conditions not used in the training or validation data sets, and populated with galaxies using HODs not used in the training or testing data sets. The creation of the three data sets is described in the following subsections.

2.1. `AbacusCosmos` Simulations

The `AbacusCosmos` simulations are a suite of publicly available *N*-body simulations. The suite includes the `AbacusCosmos` `1100box` simulations, a sample of large-volume *N*-body simulations at a variety of cosmologies, as well as the `1100box` *Planck* simulations, a sample of simulations with cosmological parameters consistent with the *Planck* fiducial cosmology.

The `AbacusCosmos` `1100box` simulations are used to create the training and validation sets. This suite of simulations comprises 40 simulations at a variety of cosmologies that differ for six cosmological parameters: $\Omega_{\mathrm{CDM}} h^2$, $\Omega_b h^2$, σ_8, H_0, w_0, and n_s. The cosmologies for this suite of simulations were selected by a Latin hypercube algorithm and are centered on the *Planck* fiducial cosmology (Planck Collaboration et al. 2015). Each simulation has side length $1100 h^{-1}$ Mpc and particle mass $4 \times 10^{10} h^{-1} M_\odot$. The suite of 40 simulations is phase-matched.

While the `AbacusCosmos` `1100box` simulations are used to create the training and validation sets, the `AbacusCosmos` *Planck* simulations are used to create the testing set. These 20 simulations have cosmological parameters consistent with those of Planck Collaboration et al. (2015): $\Omega_b h^2 = 0.02222$, $\Omega_m h^2 = 0.14212$, $w_0 = -1$, $n_s = 0.9652$, $\sigma_8 = 0.830$, $H_0 = 67.26$, and $N_{\mathrm{eff}} = 3.04$. They have identical side length ($1100\ h^{-1}$ Mpc) and particle mass ($4 \times 10^{10} h^{-1} M_\odot$) to the `1100box` suite of simulations, but each uses unique initial conditions and none are phase-matched to the `1100box` simulations. See Garrison et al. (2018) for more details about the `AbacusCosmos` suite of simulations.

2.2. HOD

A halo HOD is a way to populate dark matter halos with galaxies. In their most basic form, HODs are probabilistic models that assume that halo mass is the sole halo property governing the halo–galaxy connection (Berlind & Weinberg 2002). A standard HOD models the probability of a halo hosting a central galaxy, $\bar{n}_{\mathrm{central}}$, and the mean number of

---

[4] https://lgarrison.github.io/AbacusCosmos/

satellites, $\bar{n}_{\rm satellite}$, as a function of a single halo property, the mass M. The standard HOD by Zheng & Weinberg (2007) gives the mean number of central and satellite galaxies as

$$\bar{n}_{\rm central} = \frac{1}{2}\,{\rm erfc}\left[\frac{\ln(M_{\rm cut}/M)}{\sqrt{2}\,\sigma}\right]$$
$$\bar{n}_{\rm satellite} = \left[\frac{M - \kappa M_{\rm cut}}{M_1}\right]^{\alpha}\bar{n}_{\rm central}, \qquad (1)$$

where $M_{\rm cut}$ sets the halo mass scale for central galaxies, σ sets the width of the error function of $\bar{n}_{\rm central}$, M_1 sets the mass scale for satellite galaxies, α sets the slope of the power law, and $\kappa M_{\rm cut}$ sets the limit below which a halo cannot host a satellite galaxy. Here M denotes the halo mass, and we use the virial mass definition $M_{\rm vir}$. The actual number of central galaxies in a halo follows the Bernoulli distribution with the mean set to $\bar{n}_{\rm central}$, whereas the number of satellite galaxies follows the Poisson distributions with the mean set to $\bar{n}_{\rm satellite}$.

While this standard HOD populates halos probabilistically according to halo mass, recent variations of the HOD incorporate more flexibility in modeling. These flexible HODs allow additional halo properties—beyond the halo mass—to inform galaxy occupation (e.g., Hearin et al. 2016; Yuan et al. 2018b). The HOD implemented here is one such flexible model; it uses the publicly available `GRAND-HOD` package.[5] This HOD implementation introduces a series of extensions to the standard HOD, including flexibility in the distribution of satellite galaxies within the halo, velocity distribution of the galaxies, and galaxy assembly bias. To add this flexibility, we invoke two extensions: the satellite distribution parameter, s, and the galaxy assembly bias parameter, A. The satellite distribution parameter allows for a flexible radial distribution of satellite galaxies within a dark matter halo, and the galaxy assembly bias parameter allows for a secondary HOD dependence on halo concentration. For complete information about `GRAND-HOD` and its HOD extensions, see Yuan et al. (2018a).

Fifteen sets of HOD model parameters are generated for each `AbacusCosmos` simulation box, and 31 are generated for each *Planck* box. For each simulation box, a baseline HOD model is selected as a function of cosmology; these baseline models vary only in $M_{\rm cut}$ and M_1, and baseline values of all other HOD parameters remain the same. This ensures that the combined effect of perturbing the cosmology and HOD is mild. This is done because, despite the fact that the cosmological parameters of each simulation are only perturbed by a few percent, coupling these cosmological changes with perturbations to the HOD can lead to drastic changes to the mock catalogs and clustering statistics. To minimize these effects, instead of populating galaxies according to HOD parameters in an ellipse aligned with the default parameter basis, we populate according to parameters in an ellipse defined over a custom parameter basis.

For the *Planck* cosmology, the HOD parameters are chosen in reference to the parameter ranges in Kwan et al. (2015): $\log_{10}(M_{\rm cut}/h^{-1}\,M_\odot) = 13.35$, $\log_{10}(M_1/h^{-1}\,M_\odot) = 13.8$, $\sigma = 0.85$, $\alpha = 1$, $\kappa = 1$, $s = 0$, and $A = 0$. However, we modify two baseline HOD parameter values—$M_{\rm cut}$ and M_1—for the non-*Planck* simulations. We set the baseline value of $M_{\rm cut}$ in each cosmology box such that the projected two-point correlation function w_p(5–10 Mpc) of all the halos with $M > M_{\rm cut}$ is equal to the w_p(5–10 Mpc) of the centrals in the baseline HOD at *Planck* cosmology, where w_p(5–10 Mpc) is defined as

$$w_p(5\,-\,10\,{\rm Mpc}) = \int_{5\,{\rm Mpc}}^{10\,{\rm Mpc}} w_p\, d(r_\perp). \qquad (2)$$

This effectively holds the baseline w_p of the centrals approximately constant across all of the different simulations. Then M_1 is selected such that the baseline satellite-central fraction in each cosmology box is the same as that of the baseline HOD in *Planck* cosmology.

For each `1100box`, seven additional pairs of model parameters uniformly sample the parameter space within 5% of the baseline HOD (15 additional pairs for each *Planck* box). For HOD parameters s and A, whose baseline parameters are zero, we draw uniform samples between −0.05 and 0.05. The two HODs of each pair are symmetrically offset across the baseline HOD. Excluding the baseline HOD, 14 unique HODs are generated for each `AbacusCosmos` `1100box` simulation, and 30 unique HODs are generated for each *Planck* simulation. Four random seeds are used to populate the simulations with realizations of galaxies according to the HOD; this results in four unique galaxy catalogs for each HOD. The details of how these are used are described in the next section. For complete information about the HOD implementation, see Yuan et al. (2019).

2.3. Training and Validation Sets

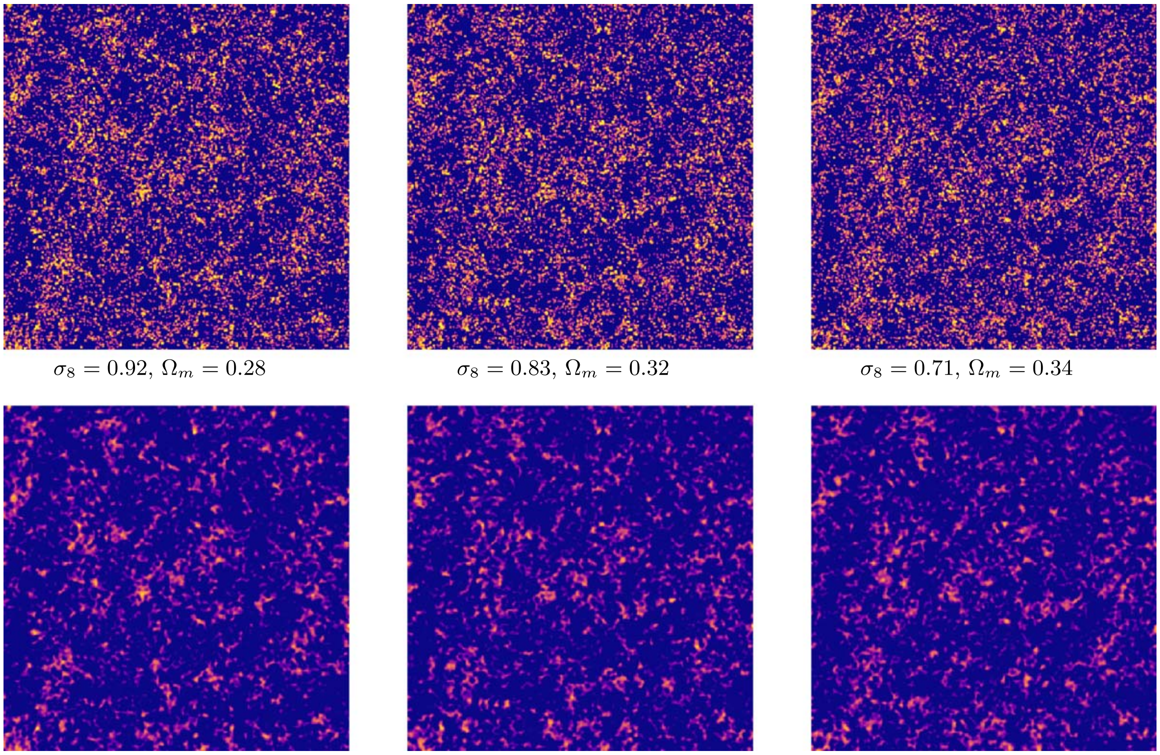
The training sample of mock observations (for training the deep learning models) and validation sample of mock observations (for assessing when the models have sufficiently fit) are created from the `AbacusCosmos` suite of `1100box` simulations.

`AbacusCosmos` includes 40 simulated cosmologies, and for each of these, we select a random distance along the x- and y-axes to become the new zero-point of the box, taking advantage of the periodic boundary conditions of the simulation to recenter the structure along these axes. The line-of-sight direction, z, includes redshift space distortions stemming from the peculiar velocities of halos, and we do not recenter the box along this direction. Because the `1100box` simulations all have the same initial conditions, this random reshuffling minimizes the chances of our model learning about correlated structure across simulations.[6] The mock observations of the training set are built from the portion of the box with $220\ h^{-1}$ Mpc $\leqslant z < 1100\ h^{-1}$ Mpc, while the validation set is built from the structure in the range $0h^{-1}$ Mpc $\leqslant z < 220h^{-1}$ Mpc. By completely excluding this portion of the simulation from the training set, we can test and ensure that the ML model does not rely on its ability to identify or memorize

---

[5] https://github.com/SandyYuan/GRAND-HOD

[6] Simulations with matched initial conditions will produce portions of the cosmic web with, for example, a unique or unusual fingerprint of filamentary structure. The evolutionary stage of a particular structure is highly dependent on the simulation's σ_8 and other cosmological parameters. Because CNNs are particularly adept at pattern finding, care must be taken to prevent a CNN from learning to identify some unique structure—especially one that is particular to a suite of simulations and the initial conditions of those simulations—and infer cosmological parameters from its details. This is not an approach that will generalize to real observations and can give overly optimistic results.



$\sigma_8 = 0.92,\ \Omega_m = 0.28$ $\sigma_8 = 0.83,\ \Omega_m = 0.32$ $\sigma_8 = 0.71,\ \Omega_m = 0.34$

$\sigma_8 = 0.92,\ \Omega_m = 0.28$ $\sigma_8 = 0.83,\ \Omega_m = 0.32$ $\sigma_8 = 0.71,\ \Omega_m = 0.34$

Figure 1. Top: sample of training input images. Shown is a 2D summed projection of the 3D image (or "slab"). The training, validation, and testing samples include a number of choices designed to reduce the likelihood of giving the ML model an unfair advantage: we employ a zero-point shift to minimize learning from images with correlated structure, we use random HODs and seeds to allow for uncertainties in galaxy formation physics, we use axial flips of the slabs to augment the data, and we use unique portions of the simulation and unique HODs in the validation set to provide a way to test that the model does not rely on the particulars of the structure or HOD. To highlight the differences in the images that are strictly due to cosmology and HOD, the zero-point shift has been omitted for these images. Bottom: same images as above, smoothed with a Gaussian filter ($\sigma = 1$ pixel) to emphasize the differences between images that are due to cosmological models.

large-scale structure correlations stemming from the matched initial conditions.

The box is divided into 20 nonoverlapping slabs, which are 550 h^{-1} Mpc in the *x*- and *y*-directions and 220h^{-1} Mpc along the line-of-sight *z*-direction. Halo catalogs generated by the `ROCKSTAR` halo finder (Behroozi et al. 2012) become the basis for four mock observations per slab.

For each slab, we select and apply one HOD from the 15 that are generated. Eleven of the HODs are reused as necessary in the 16 training slabs. The remaining four HODs are reserved exclusively for the four validation slabs. By setting aside four HODs for the validation set, the validation set is populated with galaxies in a way that is unique from the observations used for training, and we can ensure that the ML model results are not dependent on memorization or previous knowledge of the details of the HOD.

For each of the four random HOD seeds, the slabs are populated with galaxies. These training slabs vary in number of galaxies, ranging from ~17,000 to ~46,000 galaxies per slab, with the number of galaxies correlating weakly with the underlying cosmology. To show that the CNN can learn patterns in 3D galaxy distributions (beyond simply counting the number of galaxies in the mock observation), we randomly subselect the galaxy population so that all observations have 15,000 galaxies.

The selected galaxies are binned into a 275 × 275 × 55, 3D, single-color image. Galaxies are assigned to voxels using a triangular-shaped cloud (TSC) for voxels of size 2 × 2 × 5h^{-1} Mpc. Projected galaxy densities for three sample cosmologies are shown in Figure 1.

Because the ML model described in Section 3 is not invariant under mirroring of images, we augment our data by applying an axial flip along the *x*- and/or *y*-directions to three of the four slabs. For each of these three mirror images, we use a new random seed for the HOD and uniquely subselect to 15,000 galaxies.

The power spectrum of the galaxy density field is computed for each slab. To perform this calculation, we pad the galaxy density field with zeros to double the image size in each direction to account for the lost periodic boundary conditions, Fourier transform the resulting 550 × 550 × 110 image, and convert the result to a power spectrum in physical units. This 3D power spectrum is next deconvolved to account for the TSC-aliased window function (as in, e.g., Jeong 2010). The 3D power is summarized as a 1D power spectrum by averaging the power in binned spherical annuli, treating the *x*-, *y*-, and *z*-components of the power as being equivalent. (See, e.g., Jeong 2010 and Hand et al. 2018 for further details on calculating a 1D power spectrum from a 3D density field.) Due to the anisotropic nature of the slab and voxel dimensions, the most conservative choices for minimum and maximum *k* values are selected. These are set by the shortest box dimension (220h^{-1} Mpc) and the Nyquist frequency of the largest pixel dimension (5h^{-1} Mpc), respectively. Power spectra for a sample of galaxy catalogs are shown in Figure 2.

To recap, the method for building mock observations from each of the simulations is as follows.

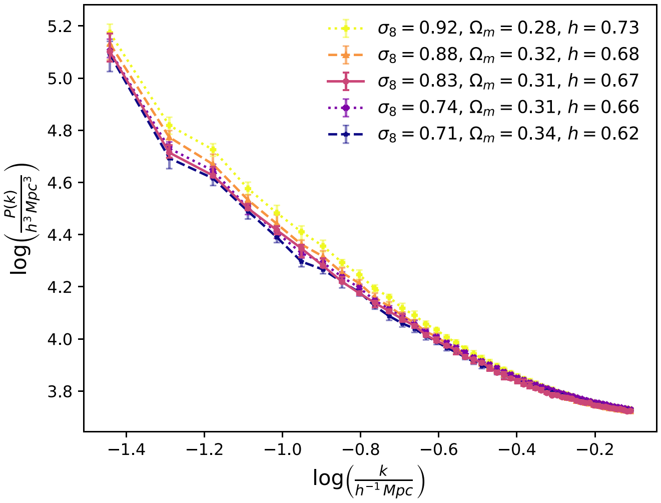


Figure 2. Mean galaxy power spectra, $P(k)$, for four of the 40 training cosmologies (yellow, orange, purple, and blue), as well as for the *Planck* test cosmology (pink). Points indicate the mean power, while error bars show the middle 68% of the mock observations. The "vector features" input, shown in Figure 3, is a single realization of this power spectrum; for each mock observation, the power spectrum is calculated directly from a single 3D mock galaxy observation.

1. A random x- and y-value is selected to be the new zero-point of the box, and $z = 0$, along the line-of-sight direction with redshift space distortion, remains unchanged.
2. The box is divided into 20 nonoverlapping slabs, each $550 \times 550 \times 220 h^{-1}$ Mpc.
3. The following applies for each slab.
    - (a) An HOD is selected. Eleven HODs, some of which are reused as necessary, are used to populate the 16 training slabs with galaxies. Four unique HODs are reserved exclusively for the four validation slabs.
    - (b) There are 15,000 galaxies randomly selected. These are binned in $2 \times 2 \times 5\ h^{-1}$ Mpc bins using a TSC.
    - (c) The previous step is repeated for each of four random seeds, incorporating mirror image(s) of the slab.
    - (d) The power spectrum of the slab is calculated.

This method results in 3200 mock observations built from 40 simulations, with 20 slabs per simulation and four seeds (with axial flips) per slab.

The 2560 slabs built from the portion of the simulation with $z \geqslant 220 h^{-1}$ Mpc comprise the training set and are used to train the ML model described in Section 3. The remaining 640 slabs are built from a nonoverlapping portion of the simulation ($z < 220 h^{-1}$ Mpc). These make up the validation set and are used to assess the models' fit.

Our creation of the test and validation sets includes a number of choices to reduce the likelihood of giving the ML model an unfair advantage: we employ a recentering of the box to minimize learning from images with correlated structure, we use random HODs and seeds to allow for uncertainties in galaxy formation physics, we use axial flips of the slabs to augment the data to account for rotational invariance, and we use unique portions of the simulation and unique HODs in the validation fold to provide a way to ensure that the model does not rely on the details of the structure or HOD.

2.4. *Planck* Testing Set

The testing sample is built from the `AbacusCosmos` *Planck* simulations. The 20 *Planck* simulations each have initial conditions that are unique from the simulation sample described in Section 2.3. Mock observations of the *Planck* testing set are built using a similar process as described in Section 2.3 with one exception: the 20 nonoverlapping slabs are each populated with galaxies according to 20 unique HODs selected randomly from the 31 HODs available. Accounting for the axial flips to augment the data, the resulting testing sample is 1600 slabs with associated power spectra. Our testing set is a truly independent sample from the training and validation sets. Though the cosmologies used in the training and validation sets are near the *Planck* fiducial cosmology, this exact cosmology is never explicitly used for training or testing.

3. Methods: ML Models

We assess three ML models: (1) a standard CNN that learns from the 3D galaxy images to regress estimates of cosmological parameters, (2) a fully connected NN that learns from the power spectrum of the galaxy images to regress estimates of cosmological parameters, and (3) a hybrid CNN (hCNN) that employs a standard CNN but also can take advantage of meaningful summary information—in this case, the galaxy power spectrum—to inject physically meaningful information into the fully connected layers. These three models are described in detail below.

3.1. CNN

The CNNs (Fukushima & Miyake 1982; LeCun et al. 1999; Krizhevsky et al. 2012) are a class of ML algorithms that are commonly used in image recognition tasks. Over many cycles, called "epochs," the network learns the convolutional filters, weights, and biases necessary to extract meaningful patterns from the input image. For cosmological applications, CNNs are traditionally applied to monochromatic (e.g., Lanusse et al. 2018; Ntampaka et al. 2019; Ho et al. 2019) or multiple-color 2D images (e.g., Dieleman et al. 2015; Huertas-Company et al. 2015; La Plante & Ntampaka 2018). However, CNNs are not confined to 2D training data; they can also be used on 3D data cubes. The 3D CNNs became popular for interpreting videos, using time as the third dimension (e.g., Ji et al. 2013), but recent cosmological applications of this algorithm have applied the technique to 3D data (e.g., Ravanbakhsh et al. 2017; He et al. 2019; Mathuriya et al. 2018; Peel et al. 2019; Aragon-Calvo 2019; Berger & Stein 2019; Zhang et al. 2019; Pan et al. 2019).

Typically, CNNs use pairs of convolutional filters and pooling layers to extract meaningful patterns from the input image. These are followed by several fully connected layers. Our standard CNN architecture includes several consecutive fully convolutional layers at the onset and mean and max pooling branches in parallel. It is implemented in Keras[7] with a Tensorflow (Abadi et al. 2016) back end and shown in Figure 3. The full architecture is as follows.

1. $3 \times 3 \times 3$ convolution with four filters
   leaky ReLU activation
   batch normalization.

---

[7] https://github.com/fchollet/keras

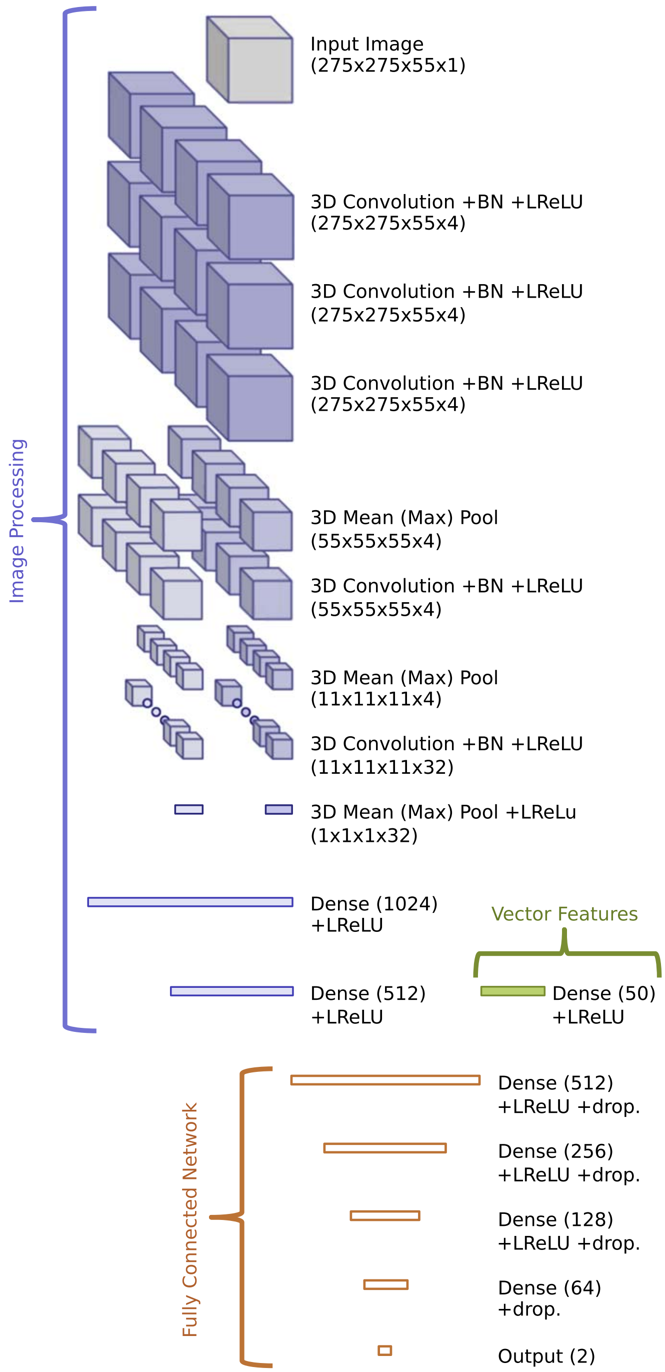


Image Processing
Input Image (275x275x55x1)
3D Convolution +BN +LReLU (275x275x55x4)
3D Convolution +BN +LReLU (275x275x55x4)
3D Convolution +BN +LReLU (275x275x55x4)
3D Mean (Max) Pool (55x55x55x4)
3D Convolution +BN +LReLU (55x55x55x4)
3D Mean (Max) Pool (11x11x11x4)
3D Convolution +BN +LReLU (11x11x11x32)
3D Mean (Max) Pool +LReLu (1x1x1x32)
Dense (1024) +LReLU
Dense (512) +LReLU
Vector Features
Dense (50) +LReLU
Fully Connected Network
Dense (512) +LReLU +drop.
Dense (256) +LReLU +drop.
Dense (128) +LReLU +drop.
Dense (64) +drop.
Output (2)

Figure 3. Visual summary of the three ML models. The NN uses the vector features input (green) with the fully connected network for processing (orange). The standard CNN uses an image input with the image processing layers (blue) plus the fully connected network (orange). The hCNN joins these by concatenating the vector features with the final layer of the image processing; the result is fed into the fully connected layers. For further details about the NN, CNN, and hCNN, see Section 3.

2. 3 × 3 × 3 convolution with four filters
   leaky ReLU activation
   batch normalization.
3. 3 × 3 × 3 convolution with four filters
   leaky ReLU activation
   batch normalization.
4. Max pooling branch (in parallel with step 5):
   - (a) 5 × 5 × 1 max pooling.
   - (b) 3 × 3 × 3 convolution with four filters
     leaky ReLU activation
     batch normalization.
   - (c) 5 × 5 × 5 max pooling.
   - (d) 3 × 3 × 3 convolution with 32 filters
     leaky ReLU activation
     batch normalization.
   - (e) 5 × 5 × 5 max pooling, flattened.
5. Mean pooling branch (in parallel with step 4):
   - (a) 5 × 5 × 1 max pooling.
   - (b) 3 × 3 × 3 convolution with four filters
     leaky ReLU activation
     batch normalization.
   - (c) 5 × 5 × 5 max pooling.
   - (d) 3 × 3 × 3 convolution with 32 filters
     leaky ReLU activation
     batch normalization.
   - (e) 5 × 5 × 5 max pooling, flattened.
6. Concatenation of the max pool branch output (4e)
   and mean pool branch output (5e)
   leaky ReLU activation.
7. 1024 neurons, fully connected
   leaky ReLU activation
   30% dropout.
8. 512 neurons, fully connected
   leaky ReLU activation
   30% dropout.
9. 512 neurons, fully connected
   leaky ReLU activation
   30% dropout.
10. 256 neurons, fully connected
    leaky ReLU activation
    30% dropout.
11. 128 neurons, fully connected
    leaky ReLU activation
    30% dropout.
12. 64 neurons, fully connected
    linear activation
    30% dropout.
13. Two output neurons, one each for Ω_m and σ_8.

We use a mean absolute error loss function and the Adam Optimizer (Kingma & Ba 2014). In practice, we scale Ω_m and σ_8 linearly so that the range of training values lies between -1 and 1. The output predictions are scaled back to physically interpretable values according to the inverse of the same linear scaling. While this may not be an important detail for these particular cosmological parameters (σ_8 and Ω_m are of the same order of magnitude), problems can arise when training multiple outputs with significantly different value ranges (e.g., if H_0 in units of km s^{-1} Mpc^{-1} were added as a third output parameter). Details about the training scheme and learning rate are discussed in Section 3.4.

In our model, small-scale feature extraction is performed by several consecutive layers of 3D 3 × 3 × 3 convolutional filters. This feature extraction is followed by aggressive pooling in parallel max and mean pooling branches that each reduce the data cube to 32 neurons. The outputs of these branches are concatenated and followed by fully connected layers. We use a leaky rectified linear unit (ReLU; Nair & Hinton 2010; Xu et al. 2015) activation function throughout. The dropout, in which 30% of the neurons are ignored during training, reduces the likelihood of the model overfitting (Srivastava et al. 2014).

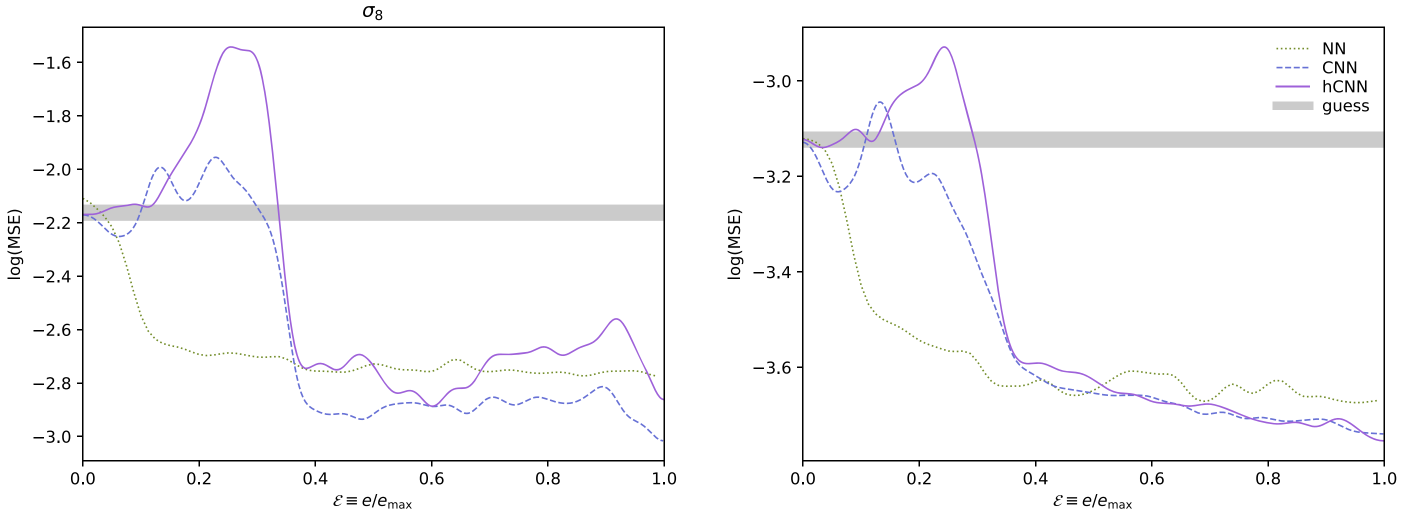


Figure 4. The MSE as a function of scaled epoch, $\mathcal{E}$. While the standard NN with a power spectrum as input (green dotted line) quickly settles to a low-error solution, the CNN (blue dashed line) and hCNN (purple solid line) have large fluctuations during the initial phase of training ($\mathcal{E} \lesssim 0.32$). Here the errors on the validation set predictions are regularly worse than a guess of the mean value (gray line) for both σ_8 (left) and Ω_m (right). The learning rate is decreased at $\mathcal{E} \approx 0.32$, and the CNN and hCNN settle into a low-error regime. To remove fluctuations that visually detract from the overall trends in error and slope, the curves shown in this figure have been smoothed with a Gaussian filter ($\sigma = 0.02$).

The model takes a 275 × 275 × 55 image as input and learns the filters, weights, and biases necessary to regress estimates of two cosmological parameters, the amplitude of matter fluctuations (σ_8) and the matter density parameter (Ω_m); each of the two output neurons maps to a cosmological parameter.

It is important to note here that parameter estimates from this CNN (as well as the models discussed in Sections 3.2 and 3.3) should not be interpreted as samples from a posterior from which a credible interval or region can be inferred. Instead, these deep learning techniques predict informative biased estimates of σ_8 and Ω_m. Uncertainty estimation methods for deep NNs are an active area of research (e.g., Lakshminarayanan et al. 2016; Kuleshov et al. 2018).

3.2. Standard NN

The standard NN uses only the fully connected layers, with the power spectrum as the only input, fed into steps 8–13 in the above architecture. It is shown in Figure 3. The model takes the binned power spectra as input and learns the weights and biases necessary to regress estimates of the cosmological parameters of interest.

3.3. hCNN

The hCNN takes advantage of a standard CNN but also utilizes information that is known to be important and meaningful. The power spectrum, which carries cosmological information, is folded in by inserting this information at step 8 in the standard CNN architecture. It should be noted that the use of incorporating physically meaningful parameters into a deep learning technique is not new to this work and has been used previously in astronomy (Dattilo et al. 2019), though it has not yet been widely adopted.

The hCNN model uses both the 275 × 275 × 55 images and the binned power spectra as input to learn Ω_m and σ_8. This architecture is shown in Figure 3.

3.4. Training

For training the CNN and hCNN, we adopt a two-phase training scheme. Our training approach takes advantage of a large step size during the initial phase of training to capture the diversity of cosmologies and HOD models, then transitions to a smaller step size during the second phase of training to improve the fit (see the Appendix for further discussion of this). We train for 550 epochs, 175 in the first phase and 375 in the second phase. The last 50 epochs will be used to select a model that meets criteria more nuanced than simply minimizing the loss function. It is discussed further in Section 4.2.1. Note that the NN is less sensitive to the details of training, such as step size and number of epochs), and trains significantly faster than models with convolutional layers. Therefore, the NN is trained for 800 epochs according to the details of phase one, described below.

We use the Adam Optimizer (Kingma & Ba 2014), which has a step size that varies as a function of epoch according to

$$\alpha(t) = \alpha_0 \frac{\sqrt{1 - \beta_2^t}}{1 - \beta_1^t}, \tag{3}$$

where α is the step size, t denotes a time step or epoch, α_0 is the initial step size,[8] and parameters β_1 and β_2 control the step size at each epoch. We adopt the default values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Phase one of training is 175 epochs with an initial step size of $\alpha_0 = 1.0 \times 10^{-5}$. We find that this first phase, with its larger initial step size, is necessary for the models to learn the diversity of cosmologies. Smaller learning rates tend to produce models with predictions that cluster near the mean values for σ_8 and Ω_m, while larger learning rates tend to produce models that fluctuate wildly in offset or overfit the training data. Near epoch 175, we find evidence in the CNN and hCNN that the learning rate is too large. This is characterized by swings in the tendency to over- or underpredict the validation set and can be seen in the large, fluctuating mean squared error (MSE) shown in

---

[8] The initial step size is denoted simply as “learning rate” in the Keras documentation.

Figure 4. The MSE is plotted as a function of scaled epoch, $\mathcal{E}$, defined as the epoch divided by the maximum number of training epochs.

We adopt the model at epoch 175 as a pretrained model and transition to a second phase of training with a lower learning rate. Phase two of training is an additional 375 epochs with an initial step size of $\alpha_0 = 0.2 \times 10^{-5}$. For clarity, we refer to the first training epoch of phase two as "epoch 176" for the remainder of this work. However, for the purposes of Equation (3) only, t is reset to zero. Figure 4 shows the effect of decreasing the learning rate: at $\mathcal{E} \approx 0.32$, the MSE decreases dramatically as the model settles into a stable fit that describes the validation data.

Overfitting is defined as the tendency of the model to produce excellent predictions on the testing set but fail on the validation set.[9] We find an increased learning rate or the use of max pooling, only both lead to overfitting. When the model is overfit, the validation set dramatically biases toward the mean (producing estimates that are pulled toward the mean parameters of the training sample), despite the fact that the training data are well described even at extreme values of σ_8 and Ω_m.

We caution, however, that we have not explored a full grid of hyperparameters for model optimization. It is likely that the two-phase training scheme could be avoided with carefully selected values of β_1 and β_2 of the Adam Optimizer to smoothly decrease step size. Likewise, we have not thoroughly vetted the tendency to overfit by increasing learning rate or removing mean pooling under many hyperparameter combinations. Such a comprehensive grid search is expensive and intractable with current computational resources. Therefore, the effects of learning rate and pooling described in this section should serve as a word of caution for those training other deep models but should not be overinterpreted.

4. Results

Here we present results from the validation set as a way of assessing the model's fit both near the median model and also toward extreme values of σ_8 and Ω_m. We also present results from the testing set to explore how the technique might generalize into the more realistic case where the cosmological model, galaxy formation details, and initial conditions are not explicitly known.

4.1. Validation Set Results

We define the prediction offset, b, as

$$b \equiv \langle |x_{\text{predicted}} - x_{\text{true}}| \rangle, \tag{4}$$

where $\rangle \cdot |$ denotes a mean and x is a placeholder for either σ_8 or Ω_m. Figure 5 shows the prediction offset as a function of scaled epoch, $\mathcal{E}$. During phase two of the training, the CNN and hCNN prediction offsets drop significantly, indicating that the lower learning rate is indeed reducing MSE and learning the spatial galaxy patterns that correlate with cosmological parameters.

While MSE and prediction offset both assess the typical offset of the validation set predictions, these statistics alone cannot tell the full story. It is also important to understand how the model might perform near the edges of the training set. For this, we assess the slope of a best-fit line through the true and predicted values of σ_8 and, separately, the best-fit line through the true and predicted values of Ω_m. A slope close to 1 indicates that the model fits well near the extreme values of σ_8 and Ω_m, while a slope of zero is indicative of a model biasing toward the mean. Overfit models will tend to have a larger MSE and prediction offset coupled with a smaller slope. Figure 5 shows the slope of this linear best-fit line. We can infer from the value of this fit, ~0.7–0.8 for both σ_8 and Ω_m, that the model may not predict well for σ_8 and Ω_m values near the edges of the training data and will likely bias toward the mean when presented with a cosmological parameter set far from the mean.

4.2. Testing Set Results

While it is an interesting academic exercise to discuss the results of the validation set, the universe, unfortunately, gives us one galaxy sample. This sample may differ from our training set in cosmological parameters and galaxy formation physics (and most certainly differs in initial conditions). If we aim to eventually use a CNN or hCNN to constrain cosmological models from an observed galaxy sample, is imperative to develop tools to assess ML models, going beyond a simple minimization of loss or performance on validation data. Though the model trains to minimize the mean absolute error, this is not necessarily the most interesting—or most useful—test statistic for a cosmological analysis of a large galaxy survey. Next, we lay out a technique for selecting a model with a small prediction offset.

4.2.1. Model Selection

As highlighted in Figure 5, the models do not perform well at extreme values of σ_8 and Ω_m. This is unsurprising; ML models tend to interpolate much better than they extrapolate. In practice, one would want to train on a large range of simulated cosmologies extending well beyond a region containing the expected results. Furthermore, one would expect a bias toward the mean for any cosmology near the edges of the training sample. Because of this (and for the purposes of model selection only), we limit our analysis to the simulations enclosed in a 68% ellipse in the σ_8–Ω_m plane.[10]

In addition to limiting this analysis to the 27 simulations with σ_8 and Ω_m values closest to the mean cosmology, we also only assess the last 50 epochs of the CNN and hCNN trainings ($0.91 < \mathcal{E} \leqslant 1.0$). Importantly, we only use the validation data to assess models. Recall that the training data should not be used in such a way because the model has already explicitly seen this data. Likewise, the testing data should not be used to assess models because doing so would unfairly bias the results.

For each of the 27 simulations and at each epoch, we calculate the distance between the predicted and true cosmology according to the following: for each of the 16 validation mock observations per simulation, we predict σ_8 and Ω_m. The 68% error ellipse in the σ_8–Ω_m plane is calculated, as is the distance between the true cosmological parameters ($\Omega_{m,\text{true}}$ and $\sigma_{8,\text{true}}$) and the middle of the ellipse of the predicted

---

[9] The term "overfit" is occasionally used to describe a deep learning method identifying features in a cosmological simulation that do not describe actual observations, but we use the term in the more traditional sense.

[10] The selection of simulations used here is shown in a lighter shade of gray in Figure 8; the simulations shown in dark gray are near the edges of the σ_8–Ω_m plane, expected to have results that bias to the mean, and excluded from this particular analysis for this reason.

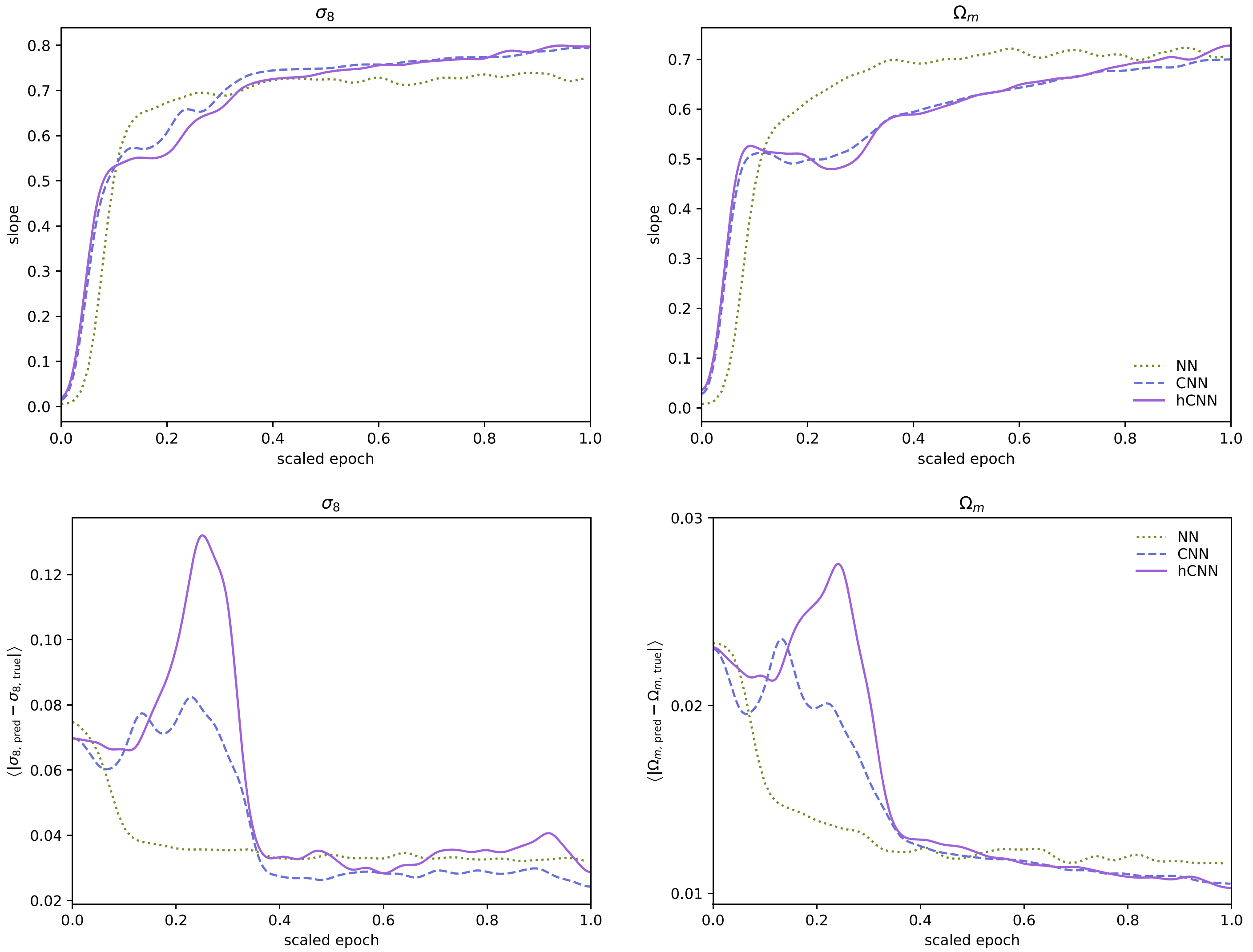


Figure 5. Top: slope of the best-fit line as a function of scaled epoch, $\mathcal{E}$. A slope of 1 indicates that the model captures the full range of σ_8 and Ω_m, while a slope of zero is indicative of the model predicting at or near the mean for all data in the validation set. As the models train, they increase the diversity of predictions. However, the slope never reaches a value of 1 for any model, indicating that the predictions will bias toward the mean for any mock observation with extreme values of σ_8 or Ω_m. Bottom: prediction offset, b, as a function of scaled epoch, $\mathcal{E}$. While the standard NN with power spectrum input (green dotted line) quickly settles to a solution with a low prediction offset, the CNN (blue dashed line) and hCNN (purple solid line) have large fluctuations during the initial phase of training ($\mathcal{E} \lesssim 0.32$). The learning rate is decreased at $\mathcal{E} = 0.32$, and the CNN and hCNN settle into a low-offset regime. To remove fluctuations that visually detract from overall trends in error and slope, the curves shown in this figure have been smoothed with a Gaussian filter ($\sigma = 0.02$).

cosmological parameters ($\Omega_{m,\mathrm{mid}}$ and $\sigma_{8,\mathrm{mid}}$). This distance, $\mathcal{Z}$, is calculated according to

$$\mathcal{Z} = \frac{(\Omega_{m,\mathrm{true}} - \Omega_{m,\mathrm{mid}})\cos\alpha + (\sigma_{8,\mathrm{true}} - \sigma_{8,\mathrm{mid}})\sin\alpha}{a^2} + \frac{(\Omega_{m,\mathrm{true}} - \Omega_{m,\mathrm{mid}})\sin\alpha - (\sigma_{8,\mathrm{true}} - \sigma_{8,\mathrm{mid}})\sin\alpha}{b^2}, \tag{5}$$

where α is the angle of the best-fit 68% ellipse, a is the length of the semimajor axis, and b is the length of the semiminor axis. Then $\mathcal{Z}$ is a 2D z-score, where $\mathcal{Z} = 1$ can be interpreted as the true value being on the edge of the 68% ellipse and $\mathcal{Z} = 0$ means that the true and mean predicted values are identical. We note that this choice favors accuracy over precision because larger error ellipses are more forgiving of large offsets between the predicted and middle predicted cosmological models.

For each epoch, the MSE as a function of epoch is calculated according to

$$\mathrm{MSE}(e) = \frac{1}{N_{\mathrm{sims}}}\sum_{i=1}^{N_{\mathrm{sims}}} \mathcal{Z}_i^2(e). \tag{6}$$

We select the epoch with the smallest MSE as the final model for the CNN and hCNN. Coincidentally, these "best" models are from training epochs that are rather close to each other, epochs 520 and 524 ($\mathcal{E} \approx 0.95$) for the CNN and hCNN, respectively. Selecting, instead, to define a 2D error ellipse that is averaged over all models and epochs selects the same hCNN model but prefers a CNN model with marginally tighter error bars and a more significant offset.

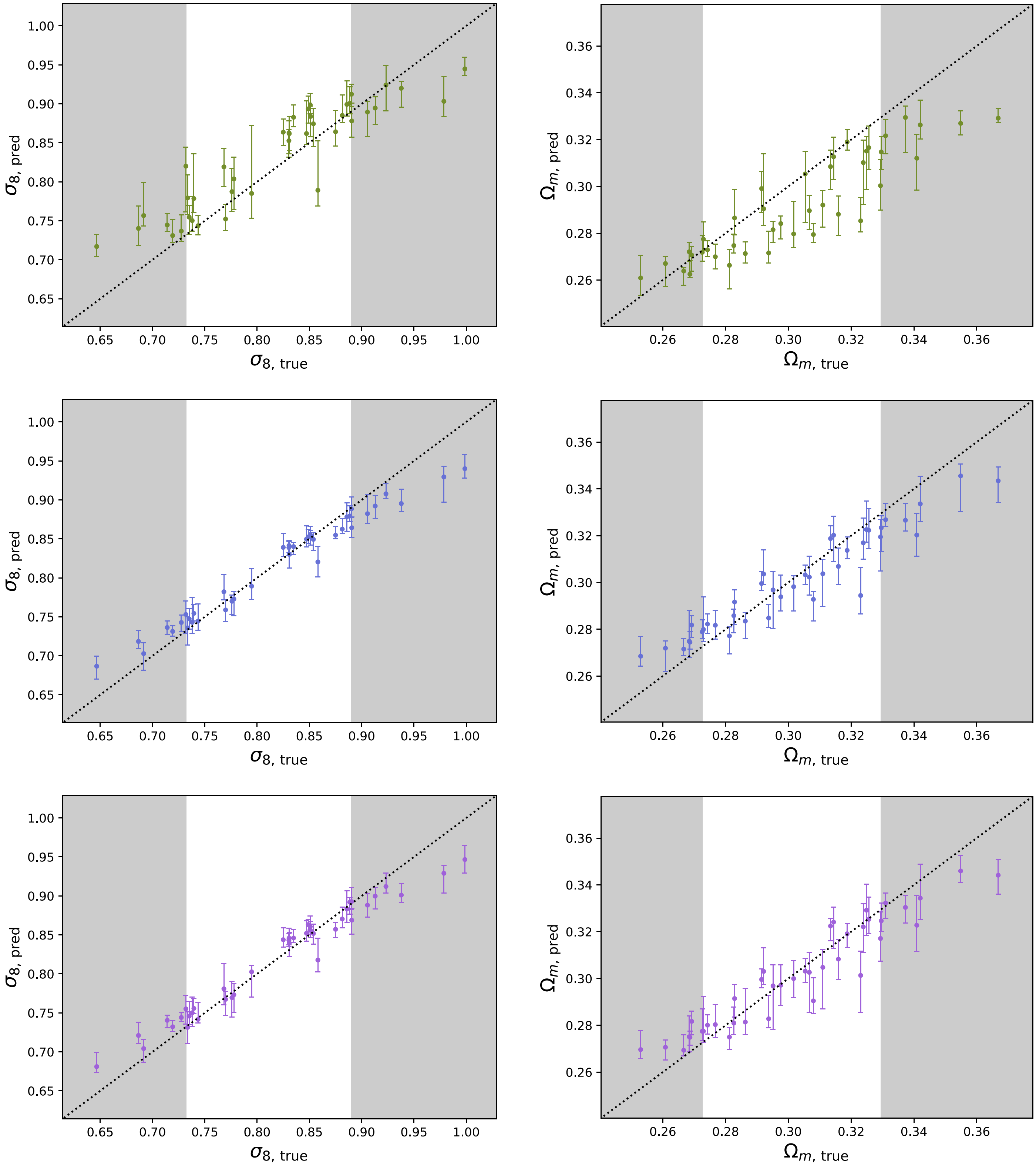


Figure 6. True and predicted values of σ_8 (left) and Ω_m (right) for the NN (green, top), CNN (blue, middle), and hCNN (purple, bottom). For the validation data of each of the 40 cosmological models, the median (circles) and middle 68% (error bars) are shown. While the predictions typically lie close to the one-to-one line (black dashed) near the central values of σ_8 and Ω_m, the bias toward the mean is more pronounced at extreme values. For illustrative purposes, σ_8 and Ω_m values below the 16th percentile and above the 84th percentile are set against a gray background, while the middle 1σ are shown against a white background. The CNN and hCNN predictions for the validation set display a significantly tighter scatter than the NN. This is unsurprising because the NN learns only from the power spectrum (see Figure 2), while the CNN and hCNN have more flexibility to learn from the unpreprocessed mock galaxy catalog.

Figure 6 shows the median and middle 68% predictions for each of the 40 cosmologies represented in the validation set at these epochs. As expected, the model visibly pulls toward the mean for outlying values of σ_8 and Ω_m. The CNN and hCNN produce tighter correlations between the true and predicted values than does the NN.

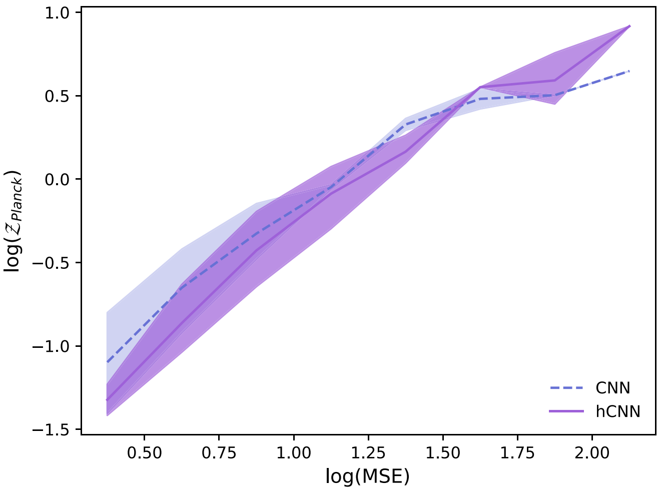


Figure 7. The MSE of the validation set, calculated in Equations (5) and (6), is tightly correlated with the testing set error $\mathcal{Z}_{Planck}$. Shown are the binned median and 68% scatter for the CNN (blue dashed line) and hCNN (purple solid line). The values tabulated here are restricted to epochs 501–550 ($0.91 < \mathcal{E} \leqslant 1.0$). The tighter correlation between low-MSE and low-$\mathcal{Z}_{Planck}$ models is mildly more pronounced for the hCNN, suggesting that the hCNN might be a more robust approach.

4.2.2. *Planck* Testing Set Results

Recall that the training set comprises mock observations built from 40 matched-phase cosmological simulations, while the validation set comprises mock observations from a unique portion ($z < 200h^{-1}$ Mpc) of those same simulations. In contrast, the testing set comprises mock observations from non-matched-phase simulations at the *Planck* cosmology that were populated with galaxies according to an HOD not yet seen by the trained model. With previously unseen cosmological parameters, HODs, and initial conditions, the *Planck* testing set is a fairer test of expected error under a realistic set of conditions. It is also important to note that, because the validation set is used to select a model, a completely separate and unseen testing set is needed to fairly assess the model.

In the previous section, we posited that the MSE of the validation set might serve as a fair proxy assessment for selecting the best model to apply to an unseen cosmology. Indeed, the validation MSE and the $\mathcal{Z}$ value for the *Planck* testing data (denoted $\mathcal{Z}_{Planck}$), are highly correlated, as shown in Figure 7. The Pearson R correlation coefficient is used to asses correlation between $\log(MSE)$ and $\log(\mathcal{Z}_{\rm Planck})$. Pearson R is given by

$$\mathcal{R} = \frac{\mathrm{cov}(X,\, Y)}{\sigma_X \sigma_Y},$$

where $\mathrm{cov}(X, Y)$ denotes the covariance of variables X and Y, σ denotes a standard deviation, $X = \log(\mathrm{MSE})$, and $Y = \log(\mathcal{Z}_{Planck})$. A Pearson R correlation coefficient of ± 1 indicates perfect correlation and 0 indicates no correlation in the linear regime. The $\log(MSE)$–$\log(\mathcal{Z}_{Planck})$ distribution has a Pearson R correlation coefficient of 0.88 for the CNN and a slightly tighter correlation of 0.93 for the hCNN. There is no strong evidence of evolution in the MSE–$\mathcal{Z}_{Planck}$ plane as a function of epoch; while low MSE is correlated with low $\mathcal{Z}_{Planck}$, the model is not taking a slow and steady march toward high or low MSE as it trains during epochs 501–550. The model's loss function should drive a decrease in mean absolute error across the 40 training cosmologies as it trains, while the MSE assesses a different measure of the goodness of fit.

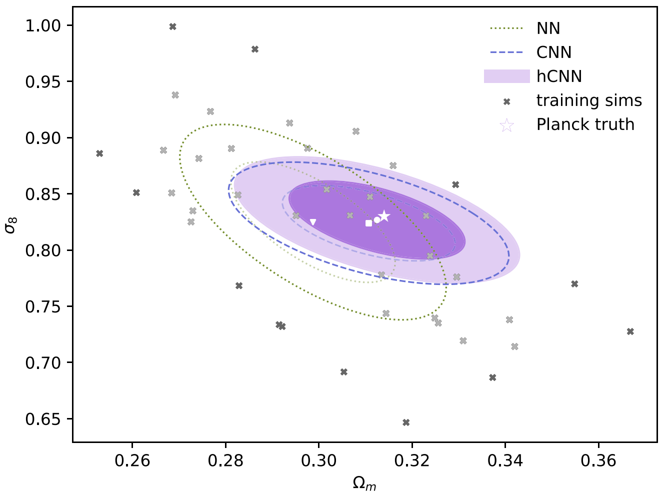


Figure 8. Testing set predictions of the NN (green dotted line), CNN (blue dashed line), and hCNN (purple shaded region); shown are the 68% and 95% error ellipses. The NN is heavily influenced by the degeneracy of the training simulations (gray crosses) in the σ_8–Ω_m plane and predicts cosmological parameters that are significantly biased toward the mean. The CNN and hCNN have tighter error ellipses and smaller offsets. The bias toward the mean is mildly smaller for the hCNN (white circle denoting the center of the error ellipse) compared to the CNN (white square).

Figure 8 shows the cosmological constraints for the NN, CNN, and hCNN. Despite the goodness of training suggested by the results in Figures 4–6, the NN never moves beyond predictions that are heavily influenced by the degeneracy of the training simulations. This is, perhaps, unsurprising. The power spectrum on which it is trained is calculated from a relatively small volume, $\sim 0.07\, h^{-3}$ Gpc^3, in contrast with the effective volume of ~ 6 Gpc^3 of the SDSS DR11 Baryon Oscillation Spectroscopic Survey (BOSS) observation (Gil-Marín et al. 2015). The volume of the mock observations used in this work is too small to isolate the baryon acoustic peak and reliably measure the acoustic scale. As a result, while the NN predicts reasonable σ_8 values, its predictions for Ω_m pull toward the mean Ω_m of training simulations.

Compared to the NN, the CNN and hCNN predictions are less biased toward the mean. The cosmological constraints in Figure 8, as well as the sample of low-$\mathcal{Z}_{Planck}$ models in Figure 7, suggest that the vector features included in the hCNN may make the model more robust, though the evidence for this is not strong.

Table 1 tabulates the simulation parameters and testing set results. For reference, we include the *Planck* testing set true values; recall that all simulations in the *Planck* suite of simulations were run at identical cosmologies, so the scatter of these values is zero. Table 1 also gives parameters that describe the distribution of the training data for reference. These include the training set mean σ_8 and Ω_m and the standard deviation of these and are used as a benchmark for how the distribution of simulated cosmologies compares to the error bars presented.

For the trio of ML models, the mean ($\bar{x}$), offset ($\bar{x} - x_{Planck}$), standard deviation of the predictions (denoted σ), and 1D z-score (offset$/\sigma$) are also given. The NN is the most offset of the trio, particularly in Ω_m, with the mean prediction $\sim 1.3\sigma$ away from the true value. From the prediction offset and error bars associated with the NN, we can conclude that the box volume is likely not large enough for the power spectrum to be

Table 1
Results Summary

	σ_8				Ω_m			
	Mean	Offset	σ	z	Mean	Offset	σ	z
Training Set	0.818	⋯	0.083	⋯	0.303	⋯	0.027	⋯
Planck testing set	0.830	⋯	⋯	⋯	0.314	⋯	⋯	⋯
NN	0.825	0.005	0.035	0.147	0.299	0.015	0.012	1.307
CNN	0.824	0.006	0.022	0.278	0.311	0.003	0.012	0.268
hCNN	0.827	0.003	0.023	0.144	0.312	0.002	0.012	0.121

diagnostic. Moving to larger mock observations that can more reliably measure the acoustic scale is likely to improve the NN technique.

The CNN and hCNN, on the other hand, both predict σ_8 to within 3% and Ω_m to within 4%. The CNN and hCNN error bars are similarly sized, but the hCNN exhibits a prediction offset that is smaller than the CNN by about a factor of 2. However, the prediction offset in both the CNN and hCNN are small, and further studies on larger mock observations are needed to make strong claims about the potential advantages of the hCNN architecture.

5. Discussion and Conclusion

We have presented a trio of ML approaches for learning estimates of σ_8 and Ω_m from a mock 3D galaxy survey. The NN uses the binned power spectrum as input and is processed through a fully connected NN architecture. The CNN uses a spatially binned 3D galaxy distribution; this is processed through a series of convolutions and pooling, and finally through a fully connected network. The hCNN merges the two.

The methods are trained and tested on a sample of mock surveys built on the `AbacusCosmos` suite of cosmological *N*-body simulations, and the mock surveys include a variety of galaxy formation scenarios through the implementation of generalized HODs. The full training sample spans a large parameter space: six cosmological parameters and six HOD parameters.

We describe a number of best practices for preventing the 3D CNN or 3D hCNN from memorizing structure and producing overly optimistic results on the validation data. Most important is setting aside an independent portion of all simulations as a validation set to assess the goodness of fit. This validation set should ideally draw from the same portion of the box to prevent the deep network from memorizing correlated structure across simulations stemming from simulations with matched initial phases. Other best practices include recentering the box, aggressive pooling to restrict the models' knowledge of slab-size length scales, subsampling the galaxy catalog to prevent the model from learning from the aggregate number of galaxies within a volume, and employing the standard suite of axial flips and rotations to account for rotational invariance.

We have shown that the validation set MSE is a useful proxy for selecting a best-fit model for estimating cosmological parameters, even when presented with previously unseen cosmological and HOD parameters.

The model is limited by the availability of simulated data; it is trained and tested on relatively small volumes ($\sim 0.07\ h^{-3}$ Gpc^3, which is $1/20$ of the simulation box volume). Furthermore, we train with only 40 training simulations at a variety of cosmologies that vary in $\Omega_{CDM}\ h^2$, $\Omega_b\ h^2$, σ_8, H_0, w_0, and n_s, which have been populated with galaxies according to a flexible HOD with six parameters. Yet even within these limitations—the small volumes and large cosmological and HOD parameter space—we have shown that it is possible to robustly train a model that can learn σ_8 and Ω_m directly from a catalog of galaxies.

Developing more realistic mock observations that span the cosmological and galaxy formation parameter space is an essential next step for applying 3D hCNNs to observational data. These extensions to the existing mock observations include adopting more diversity in cosmological parameters, taking advantage of larger training mock observations, employing additional flexibility in galaxy models, and modeling real survey embeddings. As such training data become available, 3D hCNNs have the potential to become a powerful tool for extracting cosmological information from next-generation spectroscopic surveys.

We thank our anonymous referee for helpful feedback on this work. We also thank Alexei Efros, Melanie Fernandez, Zoltán Haiman, Paul La Plante, José Manuel, Szymon Nakoneczny, Junier Oliva, Barnabás Póczos, Siamak Ravanbakhsh, Dezsö Ribli, Alexey Vikhlinin, and Javier Zazo for their helpful feedback on this project. D.J.E. is supported by U.S. Department of Energy grant DE-SC0013718 and as a Simons Foundation Investigator. S.Y. and L.G. were supported by the Simons Foundation. M.N. is supported by the Harvard Data Science Initiative. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

Appendix
On the Life Cycle of CNNs

Traditionally, CNNs are trained to minimize a loss function such as MSE or absolute error, yet it is not obvious that this is an ideal approach for astronomical and cosmological applications. In this section, we present more on the life cycle of our CNN and show additional plots that have been useful in interpreting fits and designing our two-phase training scheme.

While figures showing traditional metrics can be diagnostic, they can be difficult to interpret for models that regress more than one parameter. Such traditional figures include error as a function of epoch (e.g., Figure 4) and one-to-one scatter of true and predicted values (e.g., Figure 6). It is concerning that typical early stopping routines rely on these test statistics to determine when a model is well fit because using such diagnostics blindly can lead to unexpected or overly pessimistic results.

Figure 9 shows the validation data 2D predictions as a function of epoch. Unsurprisingly, at epochs as early as 5, the model has learned to predict a mean value but cannot differentiate among models. This is encouraging and expected;

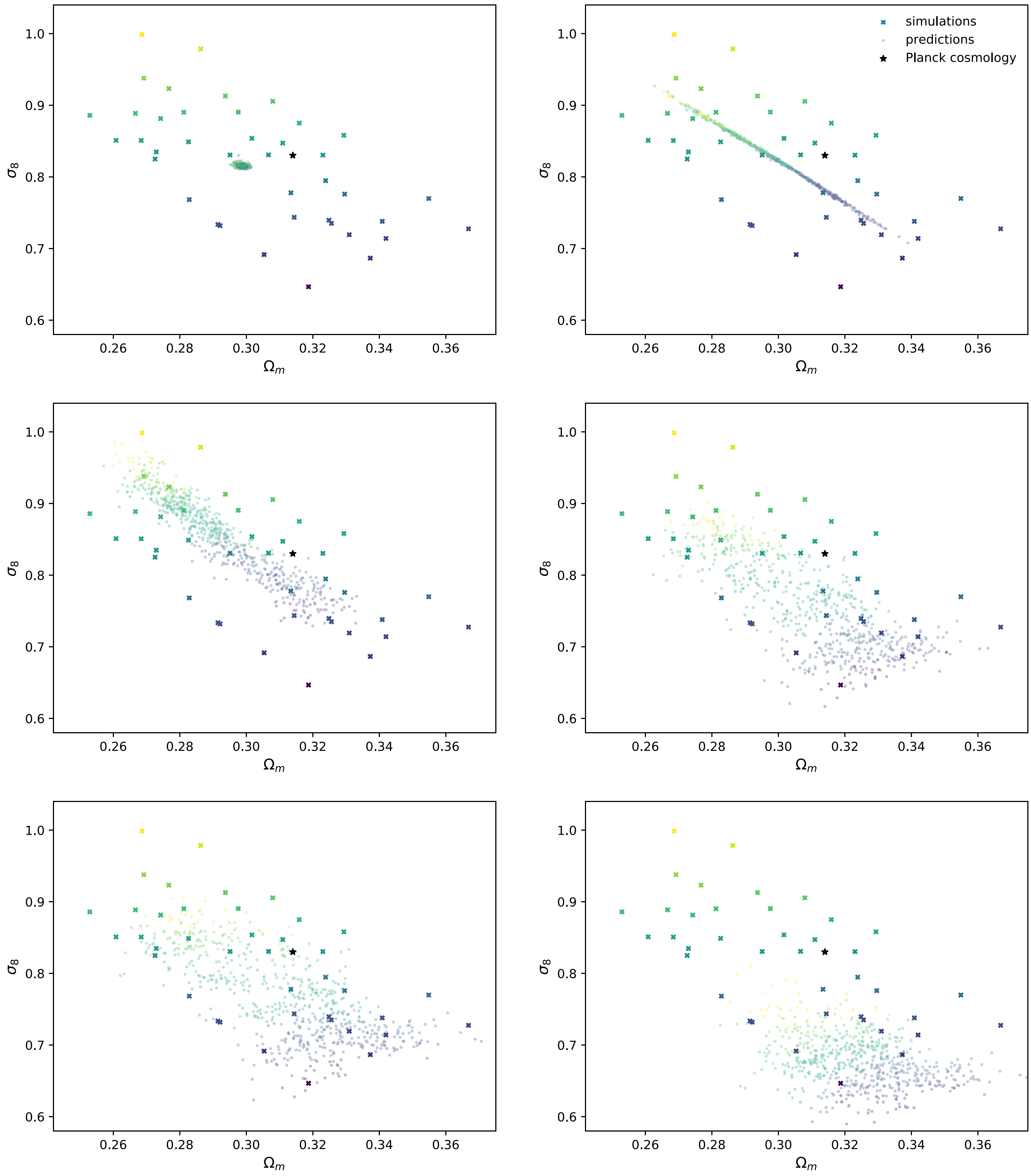


Figure 9. Life cycle of CNNs. Training data (crosses) are colored according to their σ_8 values, and predictions on the validation data (circles) are likewise colored according to their true (not predicted) σ_8 values. Early in training, the model learns reasonable values for σ_8 and Ω_m, eventually learning a tight degeneracy in this space, and finally achieving a more diverse representation of the simulations. Shown are, from top left to bottom right, epochs 5, 30, 100, 175, 219, and 220 in phase 1 of the training scheme.

the model, which is initialized to completely random weights and biases, learned reasonable values for σ_8 and Ω_m in the first few epochs.

The model predictions at epoch 30, though, are a surprising extension of this prediction of the mean. In Figure 4, the error as a function of epoch slowly and steadily decreases for the first few epochs, then begins to oscillate. At epoch ∼30, this initial plunge has come to an end, and an error-based early stopping scheme might suggest that these results are sufficient. A one-to-one plot of true and predicted σ_8 and Ω_m will tell a similar story—the results bias toward the mean, and the scatter is larger than is to be desired, but the model has clearly learned trends in the data and a diversity of σ_8 and Ω_m values. Yet, when viewed as a scatter plot in the σ_8–Ω_m plane (top right panel of Figure 9), it is clear that the CNN has learned a 2D version of predicting the mean: it has produced predictions that spread along the degeneracy direction of the training simulations, with the predictions arranged in a sensible way (i.e., the predictions of the high-σ_8 simulations are indeed at high σ_8).

It is only by delving into a "high-error" regime that the CNN starts to make progress beyond this tight degeneracy. Between epochs 30 and 175, we see large oscillations in MSE. Epoch 100 is shown as an example epoch in this region. Despite the fact that Figure 4 shows the error increasing and oscillating in this epoch range, something important and meaningful is happening under the surface. The model is starting to produce more diversity in predictions, expanding the range of predictions in the direction orthogonal to the degeneracy of the simulations. At epoch 175, the predictions are still biased toward the mean but at least span a wider spectrum of possibilities.

Here we can take an alternate timeline and continue with phase 1 of the training scheme for a few more epochs. Recall that, in the training scheme presented in the main text, we transition to a lower learning rate at epoch 175. At epochs 219 and 220 in this alternate timeline, we begin to see the oscillations in prediction offset. While the results for epoch 219 look reasonable, the results for epoch 220 are offset to very low σ_8; such large swings in prediction offset hint that the step size is too large.

Another alternative timeline transitions from phase 1 (high learning rate) to phase 2 (lower learning rate) as early as epoch 30, with disastrous results. The epoch 30 model has not yet learned much beyond the degeneracy of the simulations, and when it is moved to a much smaller learning rate, it fails to learn a diversity of predictions in the σ_8–Ω_m plane, instead producing predictions along a tight curve for many epochs.

While they are certainly valuable, traditional methods for understanding how well a CNN has fit can be difficult to interpret, particularly when assessing models trained to predict multiple parameters. Employing early stopping routines that assess a single statistical measurement of error can lead to models that have not yet learned a range of predictions in the parameter space. Appropriately assessing the diversity of predictions, identifying epochs to stop training, and developing intuition for training deep models will be an essential step toward properly using these powerful tools in astronomical and cosmological applications.

ORCID iDs

Michelle Ntampaka https://orcid.org/0000-0002-0144-387X
Lehman H. Garrison https://orcid.org/0000-0002-9853-5673

References

Abadi, M., Barham, P., Chen, J., et al. 2016, in 12th USENIX Symp. on Operating Systems Design and Implementation (OSDI 16) (Berkeley, CA: USENIX), 265

Alam, S., Ata, M., Bailey, S., et al. 2017, MNRAS, 470, 2617

Amendola, L., Appleby, S., Bacon, D., et al. 2013, LRR, 16, 6

Aragon-Calvo, M. A. 2019, MNRAS, 484, 5771

Bacon, D. J., Refregier, A. R., & Ellis, R. S. 2000, MNRAS, 318, 625

Behroozi, P., Wechsler, R., & Wu, H.-Y. 2012, Rockstar: Phase-space Halo Finder, Astrophysics Source Code Library, ascl:1210.008

Beltz-Mohrmann, G. D., Berlind, A. A., & Szewciw, A. O. 2019, MNRAS, 491, 5771

Berger, P., & Stein, G. 2019, MNRAS, 482, 2861

Berlind, A. A., & Weinberg, D. H. 2002, ApJ, 575, 587

Cacciato, M., van den Bosch, F. C., More, S., Mo, H., & Yang, X. 2013, MNRAS, 430, 767

Cole, S., Percival, W. J., Peacock, J. A., et al. 2005, MNRAS, 362, 505

Dattilo, A., Vanderburg, A., Shallue, C. J., et al. 2019, AJ, 157, 169

de Haan, T., Benson, B. A., Bleem, L. E., et al. 2016, ApJ, 832, 95

de Jong, R. S., Barden, S., Bellido-Tirado, O., et al. 2014, Proc. SPIE, 9147, 91470M

Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441

DES Collaboration, Abbott, T. M. C., Abdalla, F. B., et al. 2018, PhRvD, 98, 043526

DESI Collaboration, Aghamousa, A., Aguilar, J., et al. 2016, arXiv:1611.00036

Eisenstein, D. J., Zehavi, I., Hogg, D. W., et al. 2005, ApJ, 633, 560

Fluri, J., Kacprzak, T., Lucchi, A., et al. 2019, PhRvD, 100, 063514

Fukushima, K., & Miyake, S. 1982, Competition and Cooperation in Neural Nets (Berlin: Springer), 267

Gao, L., Springel, V., & White, S. D. M. 2005, MNRAS, 363, L66

Garrison, L. H., Eisenstein, D. J., Ferrer, D., et al. 2018, ApJS, 236, 43

Garrison, L. H., Eisenstein, D. J., & Pinto, P. A. 2019, MNRAS, 485, 3370

Gil-Marín, H., Noreña, J., Verde, L., et al. 2015, MNRAS, 451, 539

Gupta, A., Matilla, J. M. Z., Hsu, D., & Haiman, Z. 2018, PhRvD, 97, 103515

Hand, N., Feng, Y., Beutler, F., et al. 2018, AJ, 156, 160

He, S., Li, Y., Feng, Y., et al. 2019, PNAS, 116, 13825

Hearin, A. P., Zentner, A. R., van den Bosch, F. C., Campbell, D., & Tollerud, E. 2016, MNRAS, 460, 2552

Hikage, C., Oguri, M., Hamana, T., et al. 2019, PASJ, 71, 43

Hildebrandt, H., Köhlinger, F., van den Busch, J. L., et al. 2020, A&A, 633, A69

Ho, M., Rau, M. M., Ntampaka, M., et al. 2019, ApJ, 887, 25

Huchra, J. P., Geller, M. J., de Lapparent, V., & Corwin Harold G., J. 1990, ApJS, 72, 433

Huertas-Company, M., Gravet, R., Cabrera-Vives, G., et al. 2015, ApJS, 221, 8

Jeong, D. 2010, PhD thesis, Univ. Texas at Austin

Ji, S., Xu, W., Yang, M., & Yu, K. 2013, ITPAM, 35, 221

Kaiser, N., Wilson, G., & Luppino, G. A. 2000, arXiv:astro-ph/0003338

Kamnitsas, K., Ledig, C., Newcombe, V. F. J., et al. 2016, arXiv:1603.05959

Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980

Kobayashi, Y., Nishimichi, T., Takada, M., & Takahashi, R. 2020, PhRvD, 101, 023510

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems 25, ed. F. Pereira et al. (Red Hook, NY: Curran Associates, Inc.), 1097

Kuleshov, V., Fenner, N., & Ermon, S. 2018, arXiv:1807.00263

Kwan, J., Heitmann, K., Habib, S., et al. 2015, ApJ, 810, 35

La Plante, P., & Ntampaka, M. 2018, ApJ, 880, 110

Lakshminarayanan, B., Pritzel, A., & Blundell, C. 2016, arXiv:1612.01474

Lanusse, F., Ma, Q., Li, N., et al. 2018, MNRAS, 473, 3895

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. 1999, Shape, Contour and Grouping in Computer Vision (Berlin: Springer), 319

Mantz, A. B., von der Linden, A., Allen, S. W., et al. 2015, MNRAS, 446, 2205

Mathuriya, A., Bard, D., Mendygral, P., et al. 2018, arXiv:1808.04728

More, S., van den Bosch, F. C., Cacciato, M., et al. 2013, MNRAS, 430, 747

Naidoo, K., Whiteway, L., Massara, E., et al. 2020, MNRAS, 491, 1709

Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning (ICML) 10, ed. J. Furnkranz & T. Joachims (Madison, WI: Omnipress), 807

Ntampaka, M., ZuHone, J., Eisenstein, D., et al. 2019, ApJ, 876, 82

Pan, S., Liu, M., Forero-Romero, J., et al. 2019, arXiv:1908.10590

Peacock, J. A., & Smith, R. E. 2000, MNRAS, 318, 1144

Peel, A., Lalande, F., Starck, J.-L., et al. 2019, PhRvD, 100, 023508

Percival, W. J., Baugh, C. M., Bland-Hawthorn, J., et al. 2001, MNRAS, 327, 1297

Perraudin, N., Defferrard, M., Kacprzak, T., & Sgier, R. 2019, A&C, 27, 130

Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2014a, A&A, 571, A16

Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2014b, A&A, 571, A24

Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016, A&A, 594, A13

Ravanbakhsh, S., Oliva, J., Fromenteau, S., et al. 2017, arXiv:1711.02033

Ribli, D., Ármin Pataki, B., Zorrilla Matilla, J. M., et al. 2019a, MNRAS, 490, 1843
Ribli, D., Pataki, B. Á., & Csabai, I. 2019b, NatAs, 3, 93
Schmelzle, J., Lucchi, A., Kacprzak, T., et al. 2017, arXiv:1707.05167
Schmidhuber, J. 2014, arXiv:1404.7828
Scoccimarro, R., Sheth, R. K., Hui, L., & Jain, B. 2001, ApJ, 546, 20
Shectman, S. A., Landy, S. D., Oemler, A., et al. 1996, ApJ, 470, 172
Simonyan, K., & Zisserman, A. 2014, CoRR, arXiv:1409.1556
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, The Journal of Machine Learning Research, 15, 1929
Takada, M., Ellis, R. S., Chiba, M., et al. 2014, PASJ, 66, R1
Tegmark, M., Blanton, M. R., Strauss, M. A., et al. 2004, ApJ, 606, 702
van den Bosch, F. C., More, S., Cacciato, M., Mo, H., & Yang, X. 2013, MNRAS, 430, 725
Van Waerbeke, L., Mellier, Y., Erben, T., et al. 2000, A&A, 358, 30
Vikhlinin, A., Kravtsov, A. V., Burenin, R. A., et al. 2009, ApJ, 692, 1060
Wang, K., Mao, Y.-Y., Zentner, A. R., et al. 2019, MNRAS, 488, 3541
Wechsler, R. H., Zentner, A. R., Bullock, J. S., Kravtsov, A. V., & Allgood, B. 2006, ApJ, 652, 71
Weinberg, D. H., Mortonson, M. J., Eisenstein, D. J., et al. 2013, PhR, 530, 87
Wittman, D. M., Tyson, J. A., Kirkman, D., Dell'Antonio, I., & Bernstein, G. 2000, Natur, 405, 143
Xu, B., Wang, N., Chen, T., & Li, M. 2015, arXiv:1505.00853
York, D. G., Adelman, J., Anderson, J. E., et al. 2000, AJ, 120, 1579
Yuan, S., Eisenstein, D. J., & Garrison, L. H. 2018a, MNRAS, 478, 2019
Yuan, S., Eisenstein, D. J., & Garrison, L. H. 2018b, GRAND-HOD: GeneRalized ANd Differentiable Halo Occupation Distribution, Astrophysics Source Code Library, ascl:1812.011
Yuan, S., Eisenstein, D. J., & Leauthaud, A. 2019, arXiv:1907.05909
Zhang, X., Wang, Y., Zhang, W., et al. 2019, arXiv:1902.05965
Zheng, Z., Berlind, A. A., Weinberg, D. H., et al. 2005, ApJ, 633, 791
Zheng, Z., & Weinberg, D. H. 2007, ApJ, 659, 1