Deep learning insights into cosmological structure formation

Luisa Lucie-Smith,[1,2,*] Hiranya V. Peiris,[2,3] Andrew Pontzen,[2] Brian Nord,[4,5,6] and Jeyan Thiyagalingam[7]

[1]*Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str. 1, 85748 Garching, Germany*
[2]*Department of Physics and Astronomy, University College London,*
Gower Street, London WC1E 6BT, United Kingdom
[3]*The Oskar Klein Centre for Cosmoparticle Physics, Department of Physics, Stockholm University,*
AlbaNova, Stockholm SE-106 91, Sweden
[4]*Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510, USA*
[5]*Department of Astronomy and Astrophysics, University of Chicago, Chicago, Illinois 60637, USA*
[6]*Kavli Institute for Cosmological Physics, University of Chicago, Chicago, Illinois 60637, USA*
[7]*Scientific Computing Department, Rutherford Appleton Laboratory, Science and*
Technology Facilities Council, Harwell Campus, Didcot OX11 0QX, United Kingdom

(Received 27 April 2022; accepted 29 February 2024; published 14 March 2024)

The evolution of linear initial conditions present in the early Universe into extended halos of dark matter at late times can be computed using cosmological simulations. However, a theoretical understanding of this complex process remains elusive; in particular, the role of anisotropic information in the initial conditions in establishing the final mass of dark matter halos remains a long-standing puzzle. Here, we build a deep learning framework to investigate this question. We train a three-dimensional convolutional neural network to predict the mass of dark matter halos from the initial conditions, and quantify in full generality the amounts of information in the isotropic and anisotropic aspects of the initial density field about final halo masses. We find that anisotropies add a small, albeit statistically significant amount of information over that contained within spherical averages of the density field about final halo mass. However, the overall scatter in the final mass predictions does not change qualitatively with this additional information, only decreasing from 0.9 dex to 0.7 dex. Given such a small improvement, our results demonstrate that isotropic aspects of the initial density field essentially saturate the relevant information about final halo mass. Therefore, instead of searching for information directly encoded in initial conditions anisotropies, a more promising route to accurate, fast halo mass predictions is to add approximate dynamical information based e.g. on perturbation theory. More broadly, our results indicate that deep learning frameworks can provide a powerful tool for extracting physical insight into cosmological structure formation.

DOI: 10.1103/PhysRevD.109.063524

I. INTRODUCTION

The formation of cosmic structures in the Universe is driven by the gravitational collapse of initially small perturbations in the density of matter, which grow over time into extended halos of dark matter. Computer simulations are the most accurate method available to compute the nonlinear evolution of dark matter over cosmic time [1–4]. Given the initial conditions and a cosmological model, N-body simulations follow the evolution of particles governed by the laws of gravity. Despite being able to compute the evolution of matter in the Universe, simulations alone do not provide a straightforward answer to how dark matter halos acquire their characteristic properties—such as mass, shape, inner profile and spin—from the initial density perturbations.

On the other hand, analytic theories of structure formation can provide a qualitative understanding of the connection between the early- and late-time Universe. By construction, all analytic frameworks present a far more simplified view of gravitational evolution relative to solving N-body dynamics, and therefore sacrifice some predictive accuracy but allow for a much clearer interpretation. Spherical collapse models provided us with the widely accepted idea that spherical overdensities encode the primary information about halo collapse [5,6]. Ellipsoidal collapse models yield a significant improvement over spherical collapse ones in predicting statistical quantities of the large-scale structure of the Universe, such as the halo mass function [7–11]. Ellipsoidal collapse

[*] luisals@mpa-garching.mpg.de

models do not directly use anisotropic features of the field in reaching their conclusion; instead, the models introduce free parameters within the spherical collapse framework, motivated by arguments about tidal shear effects. Those parameters are then fitted to numerical simulations. Therefore, whether or not anisotropic features of the initial density field have a role in establishing final halo masses remains a long-standing question in cosmological structure formation.

In previous work [12,13], we proposed a novel approach based on machine learning to gain new insights into physical aspects of the early Universe responsible for halo collapse. The approach consists of training a machine learning algorithm to learn the relationship between the early Universe and late-time halo masses directly from numerical simulations. The learning of the algorithm is based on a set of inputs, known as *features*, describing preselected physical aspects about the linear density field in the initial conditions. We trained the algorithm on spherical overdensities (motivated by spherical collapse models) and tidal shear information (motivated by ellipsoidal collapse models) in the local environment surrounding each dark matter particle in the initial conditions. Contrary to existing interpretations of the Sheth-Tormen ellipsoidal collapse model [10,11], we found that the addition of tidal shear information does not yield an improved model of halo collapse compared to a model based on density information alone [12,13]. This approach is limited by the need to explicitly construct a set of informative features, which relies on simplified analytic approximations of halo collapse. Due to this limitation, our previous work tackled a limited science question; the role of one specific anisotropic feature of the initial conditions (the tidal shear) in predicting final halo masses.

In this work, we extend our approach to a deep learning framework based on convolutional neural networks (CNNs) [14,15]. Unlike standard machine learning algorithms or analytic descriptions, CNNs do not require specification of preselected features from the data; instead, they are trained to extract information directly from raw data. This framework allows us to address a major long-standing issue in cosmology; the role of all anisotropic aspects of the initial density field in establishing final halo masses.

The structure of a CNN is closely similar to that of existing analytic halo collapse descriptions: the information in the initial conditions is compressed into a set of features, which are then combined in a nonlinear way to provide a halo mass prediction. Our CNN approach therefore yields a simplified description of halo collapse, and should not be expected to provide perfect predictive accuracy relative to detailed N-body simulations. However, since a CNN allows for the extraction of arbitrary features, it yields a model of halo collapse that far transcends the capabilities of current analytic approaches while following the same basic setup. Our approach can therefore be seen as a generalization of existing analytic approaches, which are limited to the extraction of spherical features from the initial conditions.

In Sec. II, we present an overview of our deep learning framework, followed by a description of the simulated data used to train the machine learning model and details on the CNN's architecture. We present the halo mass predictions returned by the model in Sec. III, comparing to expectations from previous work. We then move on to interpreting the learnt mapping between initial conditions and halo mass in Sec. IV, by testing the impact on the model's performance as we remove from the inputs certain physical aspects of the initial conditions. Finally, we test the robustness of our model in Sec. V, and conclude with a discussion on the implications of our work in Sec. VI.

II. THE DEEP LEARNING FRAMEWORK

N-body simulations are the most general and accurate available approach to compute the clustering evolution of matter at all scales (Fig. 1). Simulations start from early times, when the Universe was filled with small matter density perturbations that are described by a Gaussian random field. The simulations then follow the evolution of these fluctuations as they enter the nonlinear regime, through the emergence of self-gravitating dark matter halos wherein galaxies form. The final product of the simulation resembles the Universe we observe today, characterized by dark matter halos embedded in a web of filamentary large-scale structure.

Our aim is to develop a deep learning framework that can be used to learn about the physical connection between the early Universe and the mass of the final dark matter halos (Fig. 1). We focus on the mass of dark matter halos as it is the halos' primary characteristic, but our framework can similarly be applied to other halo properties. The CNN is trained to predict the final mass of the halo to which any given dark matter particle in the simulation belongs at the present time. The input to the CNN for a given particle is given by the initial density field in a cubic subregion of the initial conditions of the simulation, centered on the particle's initial position. The CNN is therefore trained to learn a particle-by-particle mapping from the initial conditions to the final mass of the halo to which each particle belongs. Our results are insensitive to the simulation box size and the choice of whether to use overdensity or gravitational potential as input, as discussed in Appendix A.

In cosmology, deep learning has become a popular method to learn mappings that require computationally expensive N-body simulations. Examples of CNN applications to simulations include estimating cosmological parameters from the dark matter or galaxy distribution [16–21], generating higher-resolution versions of low-resolution N-body simulations [22,23], as well as emulating the mapping between Zel'dovich-displaced and nonlinear density fields [24], or that between the dark

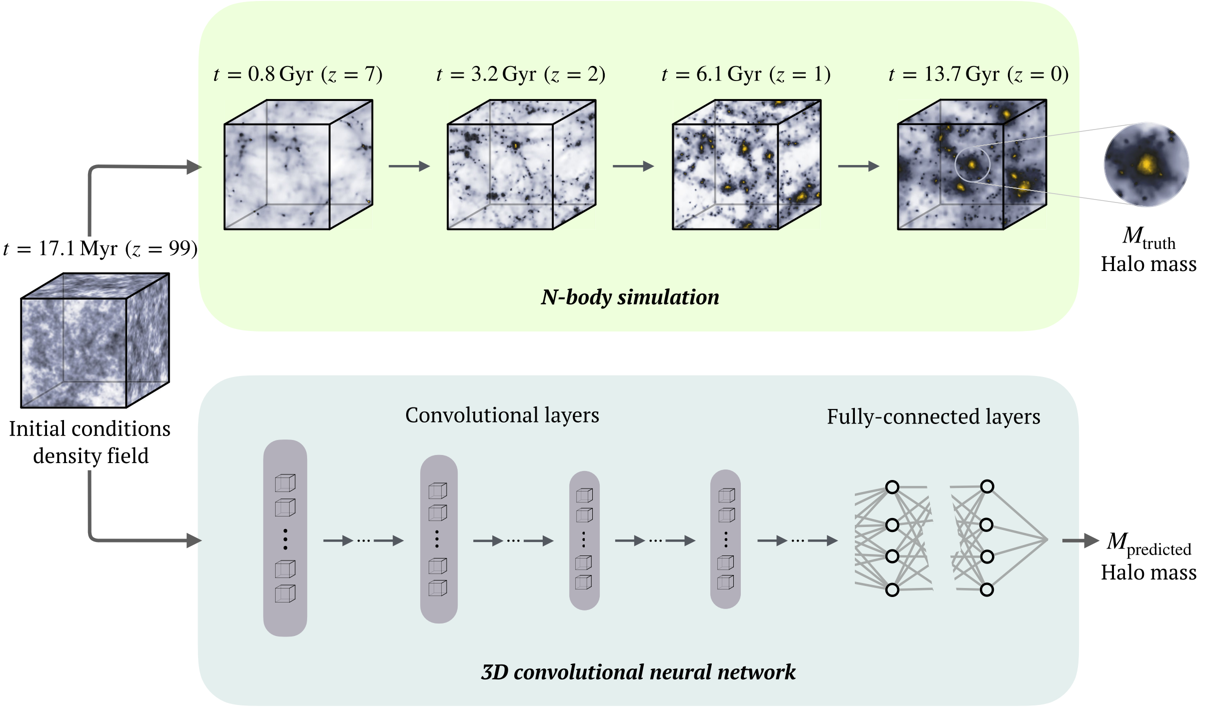

FIG. 1. N-body simulations of cosmological structure formation can accurately compute the gravitational evolution of dark matter over cosmic time, but do not provide a physical understanding of how cosmic structures arise from the initial conditions. We train a CNN model to learn the relationship between the initial density field and the final dark matter halos, given examples from N-body simulations. The inputs to the CNN are given by the initial density field surrounding each dark matter particle and the outputs are the mass of the dark matter halos to which each particle belongs at $z = 0$. The aim is to interpret the mapping learnt by the CNN in order to gain physical insights into dark matter halo formation.

matter and galaxy distributions [21,25]. The models are evaluated on global summary statistics such as two-point or three-point correlation functions [16,17,20,25,26]. Our work differs from such applications primarily in the aim: it is not to develop fast N-body surrogates, but rather to make use of deep learning to gain physical insight into the formation of cosmic structures within the simulations. Our CNN model returns particle-specific predictions, yielding a halo collapse model that can describe the nonlinear evolution of the density field from any initial location in the simulation.

A. Simulations

We generated the training data from 20 dark matter-only N-body simulations produced with P-GADGET-3 [4,27], each consisting of a box of size $L = 50\ \mathrm{Mpc}\,h^{-1}$ (comoving) and $N = 256^3$ simulation particles evolving from $z = 99$ to $z = 0$. We tested the impact of the simulation box size by repeating the analysis using three simulations with box size $L = 200\ \mathrm{Mpc}\,h^{-1}$ and the same mass resolution ($N = 1024^3$). We found no significant change in the final predictions of the model, demonstrating that a box of size $L = 50\ \mathrm{Mpc}\,h^{-1}$ is sufficiently large to capture the relevant environmental effects for the halo population considered in this analysis. The results shown in the paper are for models trained on the $L = 50\ \mathrm{Mpc}\,h^{-1}$ cosmological boxes. We made use of pynbody [28] to analyze the information contained in the simulation snapshots. The simulations adopt a WMAP5 ΛCDM cosmological model; the cosmological parameters are given by $\Omega_\Lambda = 0.721$, $\Omega_{\rm m} = 0.279$, $\Omega_{\rm b} = 0.045$, $\sigma_8 = 0.817$, $h = 0.701$ and $n_s = 0.96$ [29]. Some of the simulations are part of a suite of existing simulations, which were performed at times where these cosmological parameters were up-to-date; the newer simulations were then run with the same set of old parameters for consistency. However, we do not expect our conclusions to change when updating the cosmological parameters to more recent constraints from observations [30], as demonstrated in previous work [12]. Each simulation is based on a different realization of a Gaussian random field drawn from the initial power spectrum of density fluctuations, generated using genetIC [31]. The simulation particles of the validation and testing data were randomly drawn from four additional, independent simulations to those used for training and their inputs/outputs were generated in the same way as for the training data.

Dark matter halos were identified at $z = 0$ using the SUBFIND halo finder [4], a friends-of-friends method with a linking length of 0.2, with the additional requirement that particles in a halo be gravitationally bound. We consider the entire set of bound particles that make up a halo and do not account for substructure within halos. The resolution and volume of the simulation limit the resulting range of halo masses; the lowest-mass halo has $M = 2.6 \times 10^{10} M_\odot$ and the highest $M = 4.1 \times 10^{14} M_\odot$. We restrict our analysis even further to the mass range $\log(M/M_\odot) \in [11, 13.4]$. This is because halos with mass $M \lesssim 10^{11} M_\odot$ contain less than ∼100 particles and are therefore not well-resolved in the simulation, whereas halos with mass $M \gtrsim 3 \times 10^{13} M_\odot$ are underrepresented as a result of the small volume of our simulations.

B. Inputs and outputs of the deep learning models

The training data consist of dark matter particles randomly drawn from the ensemble of particles in the simulations which belong to dark matter halos at $z = 0$; we consider all particles within halos, and not just those at the halo centers.

The final snapshots of the simulations ($z = 0$) were used to label each dark matter particle with its ground truth variable, given by the logarithmic mass of the dark matter halo to which each dark matter particle belongs. We only consider particles that make up dark matter halos at $z = 0$ in this analysis. The ground truths were rescaled to the range $[-1, 1]$ before training; this step sets a similar scale and dynamic range for the inputs and outputs of the model, which facilitates the model's training.

The density field in the initial conditions of the simulations ($z = 99$) was used to generate the deep learning inputs associated with each particle. In the initial conditions, the density field is given by a random realization $\delta(\mathbf{x}, t_{\mathrm{initial}})$ on a uniform 256^3 grid in the $(50\ \mathrm{Mpc}\, h^{-1})^3$ simulation volume. The input associated with any given particle is given by $\delta(\mathbf{x}, t_{\mathrm{initial}})$ in a cubic subregion of the full simulation centered on the particle's initial position. The density at every voxel of the cubic box is estimated from the positions of the particles in the initial conditions; specifically, we estimate the density at the location of each particle following an SPH procedure where the SPH kernel smoothing length depends on each particle's 32 nearest neighbors. Our results are insensitive to the exact number of nearest neighbors. This subvolume has size $L = 15\ \mathrm{Mpc}\, h^{-1}$ (comoving) and resolution $N = 75^3$.

The size of the sub-box was chosen to be large enough to capture large-scale information that is relevant to the algorithm to learn the initial conditions-to-halo mass mapping. In previous work, we trained a different machine learning algorithm to infer final halo masses in the same mass range based on precomputed features of the initial conditions density field [13]. We found that the machine learning model was able to learn relevant information from the smoothed density field up to a scale of $M_{\mathrm{smoothing}} \sim 10^{14} M_\odot$. Therefore, we chose a sub-box length $L_{\mathrm{box}} = 15\ \mathrm{Mpc}\, h^{-1}$, which encloses a total mass of $M \sim 4 \times 10^{14} M_\odot$, which is more than the largest relevant mass scale adopted in our previous work. We found that increasing the volume did not change the performance of the network. On the other hand, a smaller volume would lead to degradation in the predictions of particles in high-mass halos. The resolution of the sub-box was chosen such that the length of each voxel, l_{voxel}, is the same as the initial grid spacing in the simulation i.e., $l_{\mathrm{voxel}} = 0.2\ \mathrm{Mpc}\, h^{-1}$ (comoving). This is the highest possible choice of resolution. The training set inputs were rescaled to have 0 mean and standard deviation 1; the same rescaling was then applied to the validation and test sets.

C. The deep learning model

The deep learning model consists of a 3D CNN, made of six convolutional layers and three fully connected layers. Although CNNs are generally applied to two-dimensional images, we used three-dimensional kernels in the convolutional layers that can be applied to the 3D initial density field of the N-body simulation. The convolutions were performed with 32, 32, 64, 128, 128, 128 kernels for the six convolutional layers, respectively. The kernels have size $3 \times 3 \times 3$. All convolutional layers (but the first one) are followed by max-pooling layers; their output is then used as input to the nonlinear leaky rectified linear unit (LeakyReLU) [32] activation function. We refer the reader to Appendix B for more details on the CNN architecture. By training the network across many examples of particles across many simulations, the model learns to identify the aspects of the initial density field which impact the final mass of the resulting halos.

Training the deep learning model requires solving an optimization problem. The parameters of the model, $\boldsymbol{w}$, are optimized to minimize the loss function, $\mathcal{L}_w(\boldsymbol{M}_{\mathrm{true}}, \boldsymbol{M}_{\mathrm{pred}})$, which measures how closely the predictions, $\boldsymbol{M}_{\mathrm{pred}}$, are to their respective ground truths, $\boldsymbol{M}_{\mathrm{true}}$, for the training data. The model consists of a large number of parameters, thus making it highly flexible. As a result, CNNs are often prone to overfitting the training data, without generalizing well to unobserved test data. To overcome this, regularization techniques are employed by incorporating an additional penalty term into the loss function. We designed a custom loss function given by

$$\mathcal{L}_w(\boldsymbol{M}_{\mathrm{true}}, \boldsymbol{M}_{\mathrm{pred}}) = \mathcal{L}_{\mathrm{pred}}(\boldsymbol{M}_{\mathrm{true}}, \boldsymbol{M}_{\mathrm{pred}}) + \mathcal{L}_{\mathrm{reg}}(\boldsymbol{w}), \quad (1)$$

where $\mathcal{L}_{\mathrm{pred}}(\boldsymbol{M}_{\mathrm{true}}, \boldsymbol{M}_{\mathrm{pred}})$ is the predictive term, measuring how well the predicted values match the true target values, and $\mathcal{L}_{\mathrm{reg}}(\boldsymbol{w})$ is the regularization term. The predictive term can be reexpressed as $\mathcal{L}_{\mathrm{pred}} = -\ln[p(\boldsymbol{M}_{\mathrm{true}}, |\boldsymbol{w}, \mathcal{M})]$, where $p(\boldsymbol{M}_{\mathrm{true}}|\boldsymbol{w}, \mathcal{M})$ describes the probability distribution of ground truth values $\boldsymbol{M}_{\mathrm{true}}$, given the predicted values $\boldsymbol{M}_{\mathrm{pred}}$ of the training data returned by the 3D CNN model $\mathcal{M}$ with parameters $\boldsymbol{w}$. A common choice in the community is that of a Gaussian or Laplacian distribution, yielding the popular mean-squared-error or mean-absolute-error losses for regression. We found that a Cauchy distribution provides a better description of the data, as it contains broader tails than those of a Gaussian distribution. The scale parameter γ of the Cauchy distribution, which specifies the half-width at half-maximum, was optimized via backpropagation [33] during the training procedure of the CNN, similar to the way the model parameters $\boldsymbol{w}$ are optimized. The regularization term in (1), $\mathcal{L}_{\mathrm{reg}}(\boldsymbol{w})$, was designed to simultaneously (i) improve the optimization during training by preventing the algorithm from overfitting the training data and (ii) compress the neural network model into the smallest number of parameters without loss in performance. We refer the reader to Appendix C for more details on our custom loss function.

The parameters $\boldsymbol{w}$ were optimized during training via backpropagation, which consists of the chain rule for partial differentiation applied to the gradient of the loss with respect to the parameters. The training proceeds for thousands of iterations, each consisting of a forward and a backward pass. In the forward pass, the input runs through the network and reaches the output layer, and in the backward pass, the parameters of the network are updated to minimize the loss function evaluated for the training data. The algorithm was trained on 200,000 particles, randomly drawn from the ensemble of particles of 20 simulations based on different realizations of the initial conditions; it was then validated using 10,000 particles randomly drawn from a single simulation, and tested on 99,950 particles drawn from four additional independent simulations. The training set was subdivided into batches, each made of 64 particles. Batches were fed to the network one at a time, and each time the CNN updates its parameters according to the samples in that batch. Training was done using the `AMSGrad` optimizer [34], a variant of the widely used `Adam` optimizer [35], with a learning rate of 0.00005. The learning rate was optimized via cross-validation, together with α, the parameter weighting the regularization term in the loss function. Early stopping was employed to interrupt the training at the epoch where the validation loss reaches its minimum value.

III. PREDICTING THE MASS OF DARK MATTER HALOS FROM THE INITIAL CONDITIONS

We applied our trained CNN model to particles from independent simulations not used for training. The CNN predicts the mass of the halo to which the particles will belong at $z = 0$. The test set contains particles belonging to randomly selected dark matter halos with mass $\log(M/M_\odot) \in [11, 13.4]$ (Fig. 2(a)). This mass range is set by the resolution and volume of our simulations; halos of mass $\log(M/M_\odot) \lesssim 11$ are not well-resolved and those with mass $\log(M/M_\odot) \gtrsim 13.4$ are rare (and therefore underrepresented) in the small volume of our simulations. The halos in the test set do not only differ in mass; they also differ by factors such as their formation history and their large-scale environment, which contribute significantly to making the mapping between initial conditions and halo masses challenging. For example, halos of the same mass may have assembled smoothly through small accretion

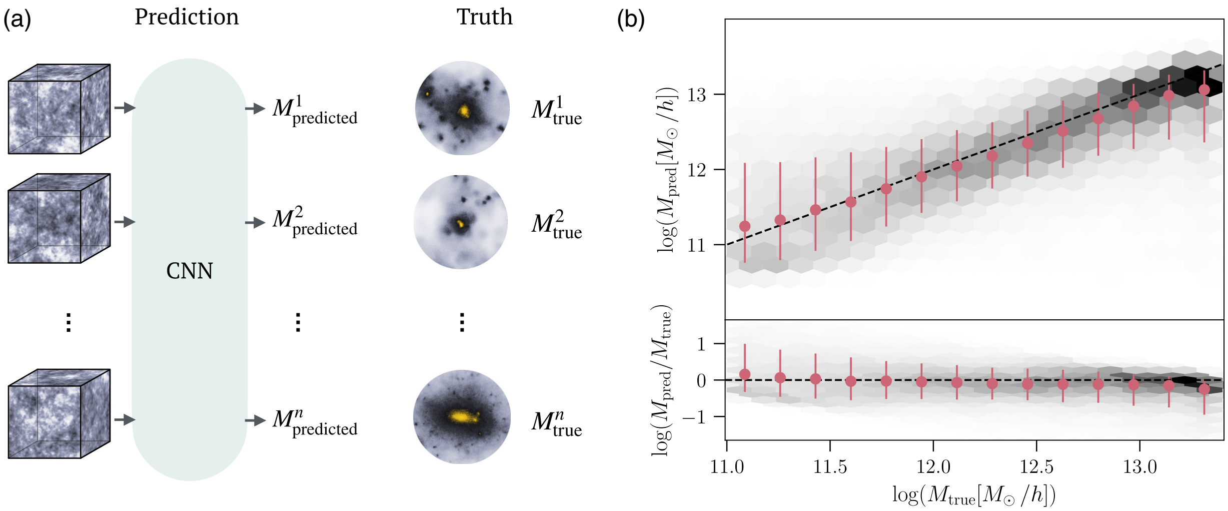

FIG. 2. (a) The CNN makes predictions for simulation particles that occupy different regions of the initial conditions of the simulation. These particles end up in halos which differ not only in their mass, but also in their formation history, large-scale environment, and amount of sub-structure within the halos. The CNN must identify from the initial density field the features that impact the final mass of the resulting halos. (b) Halo mass predictions returned by a CNN trained on the initial density field surrounding each dark matter particle's initial position. The predictions are shown against the ground truth halo mass values as a two-dimensional histogram in the top panel, while the bottom panel shows the residuals $\log(M_{\mathrm{pred}}/M_{\mathrm{true}})$. The error bars in the top (bottom) panel show the median and 68% confidence interval of the predictions (residuals) in bins of ground-truth mass values.

events or violently through mergers with other massive structures; they may have formed in isolated regions of the Universe or close to filaments and other massive objects. Particles that belong to the same halo may even have experienced significantly different dynamical histories; those in the inner region of halos are more likely to have been bound to the proto-halo patch from very early times, whereas those in the outskirts may have been more recently accreted onto the halo through late-time halo mergers, tidal stripping or accretion events. All this variability in the formation process of dark matter halos is not explicitly presented to the deep learning model; the CNN is faced with the task of finding features in the initial conditions which contain information about the complex, nonlinear evolution of halos.

Our problem setup is closely related to that of analytic models: features are extracted via convolutions from the initial conditions and combined nonlinearly to yield a halo mass prediction. Our CNN approach provides a major generalization over analytic models since the former is capable of extracting any arbitrary form of features from the inputs. Thus, we expect the CNN model to return halo mass predictions that are at least as accurate as those of state-of-the-art analytic approximations, or more accurate if there exists additional features of the initial conditions beyond those captured by analytic models that yield an improved description of halo collapse.

We compare the predictions made by the CNN to the true halo masses of the test set particles (Fig. 2(b)). The top panel shows the predicted against the true halo mass values as a two-dimensional histogram, while the bottom panel shows the residuals $\log(M_{\mathrm{pred}}/M_{\mathrm{true}})$. The errorbars in the top (bottom) panel show the median and 68% confidence interval of the predictions (residuals) in bins of ground-truth values. The black dashed line shows $y = x$ and represents the idealized case of 100% accuracy. The predictions' distributions are characterized by large variances and skewness throughout the whole mass range of halos, although the maxima of the posterior distributions are in the correct location. The variance in the distributions is larger for low-mass halos compared to high-mass halos. The extent of the variance in the predictions of the CNN is consistent with expectations from analytic models, such as the Sheth-Tormen ellipsoidal collapse model [10], which model a similar mapping between pre-selected features of the initial conditions and halo mass. We show and quantify the comparison between the CNN predictions and those of analytic models in Appendix E.

IV. INTERPRETING THE INFORMATION LEARNT BY THE CNN

Our goal it to interpret the mapping learnt by the deep learning model to understand which aspects of the initial conditions the CNN extracts to make its predictions. In particular, we wish to test whether anisotropic aspects of the density field play a major role in establishing final halo mass.

To do this, we modified the inputs to the CNN to remove any anistropic information about the 3D density field and re-trained the CNN. Given 20 concentric shells around the center of the box, the inputs were constructed by assigning to each voxel within a given shell the average density within that shell (Fig. 3(a)). The concentric shells were evenly spaced in radius r within the range $r \in [2, 75]$ voxels. Outside the largest shell that fits entirely within the box, voxels were assigned the average density within those parts of the concentric shell that intersect with the

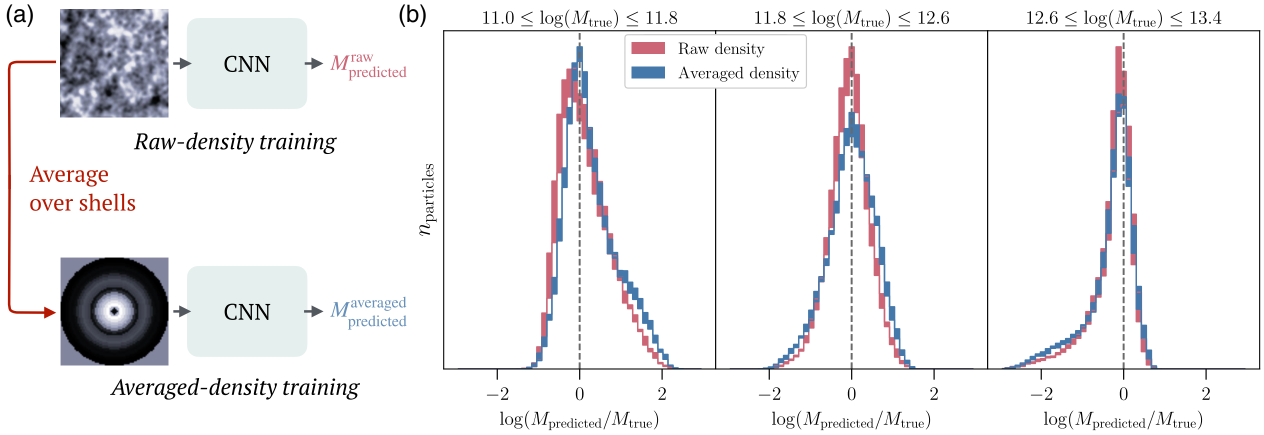

FIG. 3. (a) We re-train the model on inputs where the density in the initial conditions is averaged over shells so that any anisotropic information is removed. The two models each return a set of predictions for the test set particles. (b) We compare the predictions returned by the *raw-density* training set model and the *averaged-density* training set model. The histograms show the difference between the predicted and the true log halo mass for particles split into three mass bins of halos. The bands of the histograms capture the scatter in the predictions of each model trained with four different random seeds. The two models show similar residual distributions, except for a slightly smaller variance in the residual distribution of particles in the midmass range of halos.

box. This ensured that no additional information from outside the input sub-box region was used to construct the inputs. This procedure implies that each voxel of the 3D input sub-box only carries information about the spherically-averaged density. We call this model the *averaged-density* training set model, whereas the original one which uses the full initial density field as input is denoted as the *raw-density* training set model. The two separately-trained models were used to return their own halo mass predictions for the same set of test particles.

Figure 3(b) shows the comparison between the predictions of the two models, one trained on the raw initial density field and one trained only on spherically-averaged information. The bands of the histograms capture the scatter in the predictions of each model when trained using five different random seeds. We find that the two models return qualitatively similar predictions in the halo mass range $11 \leq \log(M/M_{\odot}) \leq 13.4$, for the same set of test particles.

To quantify the similarity between the predictions of the two models, we used the information-theoretic metric of mutual information (MI) between the predicted and ground truth halo mass values of the test set particles. In contrast to linear correlation measures such as the r-correlation, MI measures the full (linear and nonlinear) dependence between two variables. This allowed us to quantify and compare the amount of information captured by each model about the ground truth halo masses.

Mathematically, the MI between two continuous variables X and Y with values over $\mathcal{X} \times \mathcal{Y}$, $I(X, Y)$ is defined as

$$I(X, Y) \equiv \int_{\mathcal{X} \times \mathcal{Y}} p_{(X,Y)}(x, y) \ln \frac{p_{(X,Y)}(x, y)}{p_X(x) p_Y(y)} \, dx \, dy, \quad (2)$$

where $p_{(X,Y)}$ is the joint probability density distribution of X and Y, and p_X and p_Y are their marginal distributions, respectively. MI as defined by Eq. (2) is measured in natural units of information (nats). The MI was estimated using the publicly available software GMM-MI [36], which performs density estimation using Gaussian mixtures to estimate $p_{(X,Y)}$ in Eq. (2) and provides MI uncertainties through bootstrap.[1]

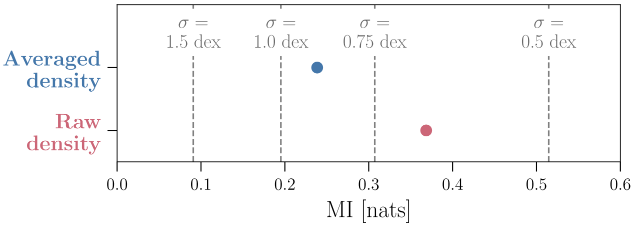

FIG. 4. Mutual information between predicted and ground truth halo mass values for the raw-density and averaged-density models. The horizontal gray lines show the value of the MI for mock predicted halo mass values constructed by adding Gaussian noise to the ground truth values with standard deviation of 1.5, 1., 0.75, 0.5 dex, respectively.

We find a value of $I(M_{\mathrm{pred}}, M_{\mathrm{truth}}) = 0.370 \pm 0.004$ for the raw density model and $I(M_{\mathrm{pred}}, M_{\mathrm{truth}}) = 0.240 \pm 0.002$ for the averaged density one, as shown in Fig. 4. The scatter in the MI between predicted and true halo mass values of the same model initiated with different random seeds is of order 10^{-3} for both the raw and averaged density models. The anisotropic components in the initial conditions add a statistically significant amount of information, increasing the MI by a factor of 1.5.

Despite being statistically significant, this increase in MI does not correspond to a useful qualitative improvement in predicting halo mass. To show this, in Fig. 4, the gray horizontal lines indicate the value of the MI for fictitious halo mass "models" that were constructed by adding Gaussian noise to the ground truth values with standard deviations of 1.5, 1.0, 0.75, 0.5 dex, respectively. By comparing the increase in MI due to the anisotropic components in the initial density field with these fictitious benchmark predictions, we see that while the scatter in predictive accuracy decreases by ~0.2 dex, the overall scatter still remains high at ~0.7 dex. These results confirm that while we have quantified a statistically significant amount of information contained in the anisotropic aspects of the initial density field about the final halo mass, it does not constitute sufficient information to provide a qualitative improvement in the initial conditions-to-halo mass mapping.

A. Dependence on particles' radial position

We next compare the accuracy of the predictions across particles that reside in different locations inside the halos. Figure 5 shows the predictive accuracy of the models for three sets of particles that were split by their location inside the halos. Particles are split into those that live in the innermost region of halos (left panel) i.e., $r \leq 0.3\, r_{200\,\mathrm{m}}$, where r is the radius of the particle from the center of its host halos and $r_{200\,\mathrm{m}}$ is the halo virial radius, those in a midregion (middle panel) i.e., $0.3 < r/r_{200\,\mathrm{m}} \leq 0.6$ and those in the outskirts of halos (right panel) i.e., $r > 0.6\, r_{200\,\mathrm{m}}$. We find that the particles with the most

[1] The distribution of ground truth values contains two hard boundaries at $\log(M_{\mathrm{min}}/[M_{\mathrm{sol}} h^{-1}])$=11 and $\log(M_{\mathrm{max}}/[M_{\mathrm{sol}}\, h^{-1}]) =$ 13.4, which can be problematic for the Gaussian mixture density estimation of $p_{(X,Y)}$. Moreover, the fact that there exists fewer halos at the high mass end also introduces some discretization to the distribution of ground truth values at high mass. To correct for these effects, we add a small random noise and then apply an arctanh transformation to the ground truth values. Since mutual information is invariant under invertible, nonlinear transformations and the added noise is smaller than the discretization, this does not affect the value of the MI, but makes the density estimation procedure more robust.

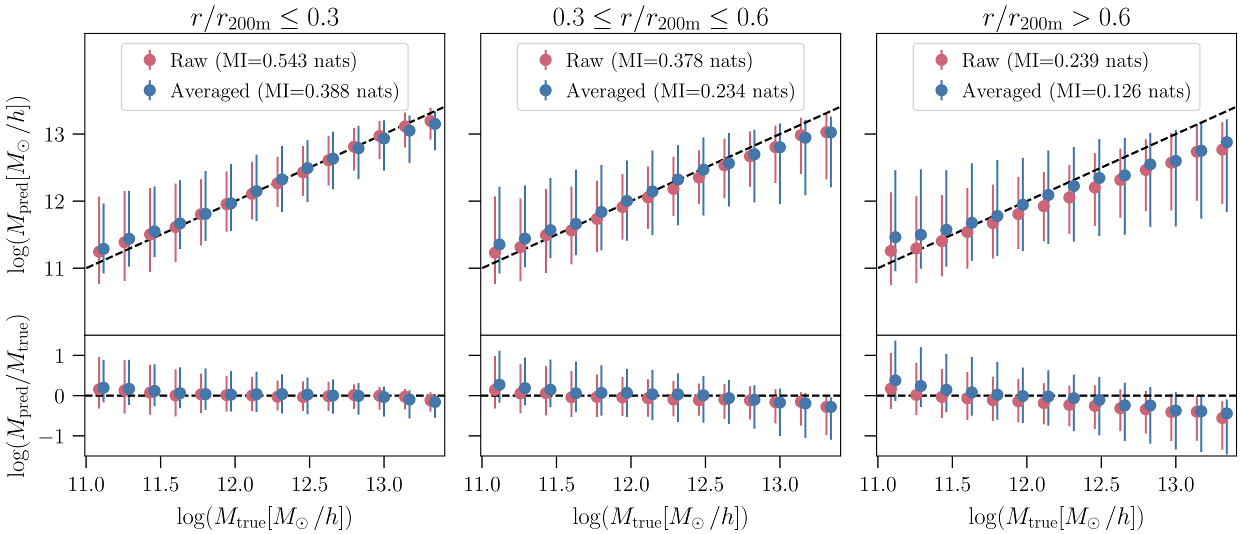

FIG. 5. Halo mass predictions for particles that reside in different locations inside the halos: those located in the inner region of the halo ($r \leq 0.3\, r_{200\ \mathrm{m}}$; left panel), those in a intermediate region ($0.3 < r/r_{200\ \mathrm{m}} \leq 0.6$); middle panel), and those in the outskirts of halos ($r > 0.6\, r_{200\ \mathrm{m}}$; right panel). The MI between predicted and ground truth halo mass values is indicated in the legend box of each panel. While the change in MI between raw-density and averaged-density models is statistically significant, the decrease in the scatter of the predictions is not sufficient to provide a qualitative improvement in the initial conditions-to-halo mass mapping.

accurate predictions are those in the innermost regions of the highest-mass halos. This is true for both the averaged-density and raw-density models. Particles that live in the outskirts of high-mass halos have larger bias and variance in their predictions; this is partially due to the fact that sometimes the input sub-box volume does not include the full extent of the region that will later collapse into a high-mass halo. Therefore, when the input sub-box is centered on an outskirt particle, its volume does not cover a large fraction of the proto-halo region, making it difficult for the CNN to infer the correct final halo mass. This explains why outskirts particles tend to yield larger errors than inner particles. For halos of smaller mass, the distribution of predictions for particles in different locations inside the halos share similar variances. One way to improve the predictions of outskirt particles in high-mass halos would be to use larger sub-box volumes as inputs. However, adopting a larger sub-box at the same resolution was not possible due to computational limitations in memory consumption on one Tesla V100 GPU. Similarly, decreasing the resolution to accommodate for a larger volume would cause the predictions to worsen for particles in low-mass halos. Despite this, the performance of the CNN remains similar to or better than the predictive accuracy of analytic models (see Appendix E).

In all panels, the MI values indicated in the legend box show a statistically significant gain in information when adding anisotropic aspects to the inputs; however, this again translates into a qualitatively small reduction in the scatter of the predictions. Therefore, our conclusion that the anisotropic information in the inputs does not yield a useful improvement in halo mass modeling is valid for all particles, regardless of their location inside the halo.

V. DEMONSTRATING THE ABILITY OF THE CNN TO EXTRACT FEATURES

One fundamental assumption behind this interpretation of our results is that the CNN is capable of capturing features across the range of different scales present in the input sub-box. If instead the CNN's ability to learn were limited, then we could not exclude the possibility that there exists additional information in the inputs which affects halo collapse, but which the CNN is unable to learn. This motivated us to perform tests showing that the same CNN model can return highly accurate predictions, when presented with the information to do so. In particular, we wanted to create a test as closely related to the real initial conditions-to-halo mass problem, which would specifically demonstrate the ability of the CNN to extract features from the density field, on all scales probed by the input sub-boxes, and return halo mass predictions that are consistent with expectations.

To do this, we tested the performance of the model in a scenario where we could compare the predictions of the CNN to our expectations. We trained the CNN to learn the mapping between the nonlinear density field at the present time ($z = 0$) and the mass of the resulting halos. This mapping is effectively given by an algorithm which first identifies the boundary of a halo based on a fixed density threshold, similar to the friends-of-friends algorithm used to identify halos in the simulation, and then computes the mass enclosed within such halo. To do this, the CNN must be able to simultaneously extract features at a number of different scales; from that of the boundary of the lowest mass halos up to that of the most massive ones. As this is a more straightforward mapping than that between the initial conditions density field and the final halo masses, we expect the CNN to return near-perfect predictions.

Similar to the $z = 99$ case, we provided the CNN with the nonlinear density field in cubic subregions of the simulation, centered at each particle's position. The inputs are given by the nonlinear density field $\delta(\mathbf{x}, t_{\rm final})$ at $z = 0$ in a subvolume centered at each particle's position. As for the $z = 99$ case, the density is estimated based on the particles' positions. We revisited our choices of box size and resolution of the 3D sub-box, as the scales of interest at $z = 0$ naturally differ from those in the initial conditions. We fixed the resolution to that used for the $z = 99$ case, $N = 75^3$, and chose a box size of $L = 1.5\ {\rm Mpc}\,h^{-1}$, which approximately corresponds to the virial radius of a halo with mass $M = 10^{14} M_\odot$. These choices resulted in a voxel length $l_{\rm voxel} \sim 30\ {\rm kpc}\,h^{-1}$, which is approximately equivalent to half the virial radius of a $M = 10^{10} M_\odot$ halo. Given that the box captures the virial radius of the largest and smallest halos probed by our simulations, we expect the input boxes to contain the information required by the algorithm to learn the density field-to-halos mapping. To summarize, the $z = 99$ and $z = 0$ settings use as inputs the density fields at $t = 17.1$ Myr and $t = 13.7$ Gyr after the big bang respectively, while keeping all other choices about the model's architecture and its hyperparameters identical.

Figure 6 shows the predictions of the CNN when trained on the $z = 0$ nonlinear density field. Just like in Fig. 2(b), the predictions are illustrated in the form of violin plots, showing the distributions of predicted halo masses in bins of true mass. As expected, the predictions show good agreement with the true halo mass labels, yielding a correlation coefficient $r = 0.96$, where $r = 1$ implies an exact linear relationship. The presence of a very low number of outliers, that make up the visible tails of the violin plots, is expected from any machine learning model trained from a finite dataset; the fraction of particles with predicted mass outside the 3σ interval of $\log(M_{\rm predicted}/M_{\rm true})$ is only 0.4%. Since the predictions are highly accurate throughout the full mass range of halos, the CNN must be able to identify the relevant features from the density field on all scales within the input sub-box and yield predictions within expected accuracy.

Crucially, the volume and resolution of the input sub-boxes were adapted to resolve the smallest and largest scales of interest at $z = 0$; therefore, this test confirms the ability of the CNN to extract features from the smallest to the largest accessible scales of the inputs. Consequently, we expect the same architecture to also have the capability to extract features on multiple scales within the $z = 99$ linear density field. However, since the features in the initial conditions have a much more complex relationship with halo mass, the predictions will naturally be less accurate than the $z = 0$ case.

VI. DISCUSSION AND CONCLUSIONS

We have presented a deep learning framework, capable of learning final halo masses directly from the linear density field in the initial conditions of an N-body simulation. The overall goal of our work is to learn about

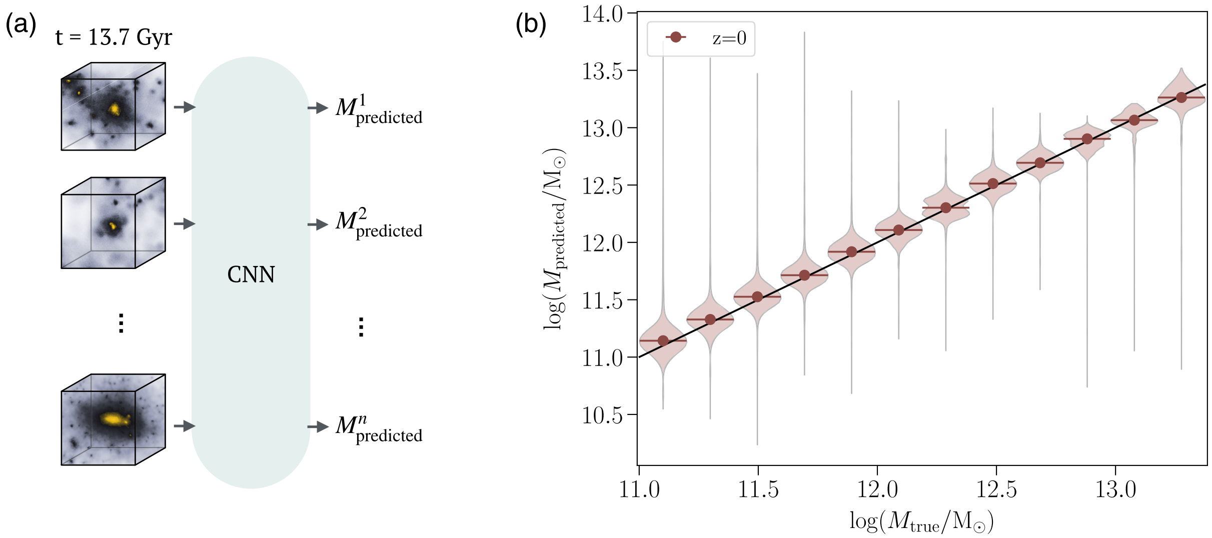

FIG. 6. Halo mass predictions returned by a CNN trained on the nonlinear density field at $z = 0$. The predictions are shown in the form of violin plots i.e., distributions (and their medians) of predicted halo masses of particles within evenly-spaced bins of true logarithmic halo mass. The predictions are in excellent agreement with their respective ground truth halo masses, yielding a correlation coefficient $r = 0.96$.

physical aspects of the early Universe which impact the formation of late-time halos using the results of deep learning, without the need to featurize the initial conditions. To do this, we require a deep learning framework that allows for the interpretability of its learning; for example, in understanding the features assembled by the convolutional layers and how these map onto the final predictions. In this work, we removed part of the information from the inputs and retrained the CNN to test the impact of this on the accuracy of the final predictions.

This allowed us to quantify in full generality the amounts of information in the isotropic and anisotropic components of the initial density field about final halo masses. We found a small, albeit statistically significant amount of additional information in the anisotropic component over that contained in the isotropic component of the initial density field; however, this corresponds to only a 0.2 dex decrease in a scatter of 0.9 dex in predictive accuracy. Thus, the addition of anisotropic information does not yield qualitative improvements in the initial conditions-to-halo mass mapping in the range $\log(M/M_{\odot}) \in [11, 13.4]$. In practice, the information in the initial conditions gleaned by the deep learning model to infer halo masses is equivalent to that captured by spherical averages over the initial density field. Our conclusions do not change if we train on the initial potential field instead of the density field, demonstrating that long-range gravitational effects do not significantly affect the local process of halo collapse (see Appendix A). A crucial test of robustness of our framework was to demonstrate that the deep learning model can effectively extract spatially local features on all scales probed by the input sub-boxes and yield robust halo mass predictions that match expectations for a simpler test-case scenario.

The idea of removing or changing parts of the data and retraining has previously been used in the deep learning community as part of a data engineering step. For example, adversarial examples are visually imperceptible perturbations added to the data that yield catastrophic failures in the CNN predictions [37]; these are often used to test the robustness of CNNs. In the context of astrophysics, inputs are often modified to include different levels of noise in simulated data [38,39]; this is also done to test the robustness of the model against noise. In our work, modifications to the input data are made to remove specific physical aspects from the initial density field; the aim is to verify whether the removed information plays a role in inferring the final output. To our knowledge, this is the first time these techniques have been adopted for generating a physical interpretation of a CNN model within a cosmological setting.

Our results lead to a reevaluation of the current understanding of gravitational collapse based on more traditional analytic and semianalytic approaches. Existing studies based on the ellipsoidal collapse model [10,11,40–42] incorporate tidal shear effects either indirectly or in the form of free parameters calibrated to numerical simulations. Peak-patch theories [8,43,44] do not quantify the impact of the added tidal shear information on the predictive power for halo mass compared to spherically-averaged density alone. Therefore, the models do not in themselves demonstrate a significant role for tidal shear in determining final halo mass. Our work focuses on providing a direct test of the importance of anisotropic information, and not just tidal shear, and shows that this does not in fact play a significant role in establishing final halo masses in the range $\log(M/M_{\odot}) \in [11, 13.4]$. Consequently, building better theoretical models of halo collapse requires incorporating information beyond the initial conditions, such as approximations to the gravitational evolution of the density field. Our results illustrate the promise of deep learning frameworks as powerful tools for extracting new insights into cosmological structure formation.

The software used to generate the simulations are available at [45] to generate the initial conditions, and at [46] to run the N-body simulations. The parameter files for generating the initial conditions and simulations can be made available from the authors upon reasonable request. The data used in this work, including simulations and training/validation/test sets, can be freely downloaded from Google Cloud Storage [47]. The code used to conduct the analysis is publicly available at [48].

ACKNOWLEDGMENTS

We thank Davide Piras for assistance with GMM-MI. L. L. S. thanks Corentin Cadiou, Joao Caldeira, Andres Felipe Alba Hernandez, Eiichiro Komatsu, Michael Maire, Manuel Valentin and Samuel Witte for useful discussions. L. L. S. thanks Stephen Stopyra for help in generating the simulations. H. V. P. thanks Justin Alsing and Daniel Mortlock for useful discussions. L. L. S. acknowledges the hospitality of the Fermi National Accelerator Laboratory, where part of this work was completed, and thanks Josh Frieman and Brian Nord for kindly hosting. H. V. P. acknowledges the hospitality of the Aspen Center for Physics, which is supported by National Science Foundation Grant No. PHY1607611. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 818085 GMGalaxies. L. L. S. was also supported by the Science and Technology Facilities Council. H. V. P. was partially supported by the research project grant "Fundamental Physics from Cosmological Surveys" funded by the Swedish Research Council (VR) under No. Dnr 2017-04212. This project has received support from the research project grant "Understanding the Dynamic Universe" funded by the Knut and Alice Wallenberg Foundation under No. Dnr KAW 2018.0067. A. P. was supported by the Royal Society. This work was partially supported by the Visiting Scholars Award Program of the Universities Research Association, and by funding from the UCL Cosmoparticle Initiative. This work used computing facilities provided by the UCL Cosmoparticle Initiative; and we thank the HPC systems manager Edd Edmondson for his committed support. This work was also partially supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant No. EP/T001569/1, particularly the "AI for Science" theme within that grant and The Alan Turing Institute, and by the Benchmarking for AI for Science at Exascale (BASE) project under the EPSRC Grant No. EP/V001310/1. Work supported by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

L. L.-S. led the project; project conceptualization, methodology, software, and obtained, validated, and interpreted results; writing—original draft, editing, final draft and funding acquisition. H. V. P. and A. P. contributed to the project conceptualization, methodology, investigation, validation and interpretation, and supervision; writing—editing and funding acquisition. B. N. contributed to the methodology and investigation; writing—editing and provided resources. J. T. contributed to the review and provided resources.

APPENDIX A: THE GRAVITATIONAL POTENTIAL VS THE DENSITY FIELD AS INPUT

The initial density field contains all the information to fully describe the initial conditions of the Universe. Other fields, such as the potential field, its gradient or its Hessian, can be derived directly from the density field in the whole simulations box via the Poisson equation. The solution to the Poisson equation is a convolution, meaning that information about the potential field is accessible to the CNN from the density field. However, the one-to-one correspondence between potential and density fields is only strictly valid in the whole simulation box. Since our inputs are limited to subregions of the simulations, the information contained within the density field is not *exactly* the same as that carried by the potential field within that same subregion. In particular, the density field within a sub-volume of the simulation excludes information about long-range gravitational effects which would instead be accessible through the potential field.

To test whether long-range gravitational effects contain relevant information for the CNN to predict final halo masses, we trained the CNN on the gravitational potential field instead of the density field. We computed the potential field across the entire simulation by solving the Poisson equation, and replaced the initial density field within each input sub-box with the gravitational potential field within that same region. We retrained the CNN model using this new set of inputs and compared the predictions returned by the CNN when trained on the density field and when trained on the gravitational potential field. Figure 7 compares the distributions of predicted halo masses within bins of true halo masses for the two cases. Each distribution is estimated using a kernel density estimate over the set of discrete mass values predicted by the model. We found that the predictions from the two models are consistent, despite small variations in the predicted distributions at the high-mass end. We quantified the significance of these differences by computing the MI [Eq. (2)] between predicted and ground truth halo mass values, for both the potential and density models. The former yields $I(\log M_{\rm pred}, \log M_{\rm true}) = 0.399 \pm 0.004$, which is similar to the MI of the density model $I(\log M_{\rm pred}, \log M_{\rm true}) = 0.370 \pm 0.004$. Therefore, we conclude that small variations in the predictions from the two models present at the high-mass end do not significantly impact the overall predictivity of the model.

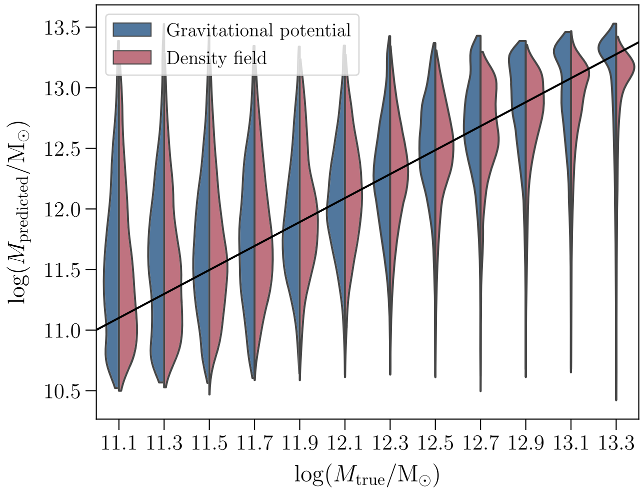

FIG. 7. We compare the predictions returned by the CNN when trained on the initial density field and when trained on the gravitational potential field. The two models yield consistent predictions, meaning that long-range gravitational effects do not impact the final mass of halos and that any information needed about the potential can be retrieved from the density field by the CNN.

This test demonstrates that long-range gravitational effects do not impact the final mass of halos in any significant way. Any information needed about the potential can be retrieved from the density field by the CNN, despite the fact that this information is expected to reside on larger scales. We therefore expect our inputs—the density field within subvolumes of the simulation—to capture all relevant information in the initial conditions about final halo mass.

APPENDIX B: CNN ARCHITECTURE

Our deep learning architecture consists of six convolutional layers, all but the first one followed by max-pooling layers, and three fully connected layers. The convolutions were performed with 32, 32, 64, 128, 128, 128 kernels for the six convolutional layers, respectively—all with a stride of 1 and zero-padding. The initial weights of the kernels in a layer were set following the Xavier initialization technique [49], which randomly draws values from a uniform distribution bounded between $\pm\sqrt{6/(n_i + n_{i+1})}$, where n_i and n_{i+1} are the number of incoming and outgoing network connections to that layer. The kernels have size $3 \times 3 \times 3$ in all convolutional layers, meaning that the first layer learns features on scales of 0.6 $\mathrm{Mpc}\,h^{-1}$. As more convolutional layers are stacked on top of each other, the algorithm becomes sensitive to features at increasing scales. In this way, both local and global information are able to propagate through the network. We applied a a nonlinear activation function to every feature map given by a leaky rectified linear unit (LeakyReLU) [32]

$$f(x) = \begin{cases} x & \text{for } x \geq 0, \\ \beta \times x & \text{for } x < 0, \end{cases} \tag{B1}$$

with $\beta = 0.03$. A LeakyReLU activation, with β of order 10^{-2}, is a common choice that has proved successful in many deep learning applications [50]. The feature maps are then fed to max-pooling layers, which reduce their dimensionality by taking the average over $2 \times 2 \times 2$ nonoverlapping regions of the feature maps. The CNN is inherently translationally invariant due to the combination of convolutional and pooling layers in the architecture; we do not incorporate any further symmetries in the network.

After the sixth convolutional layer and subsequent pooling layer, the output is flattened into a one-dimensional vector and fed to a series of three fully connected layers, each made of 256 and 128 and 1 neuron, respectively. The nonlinear activation function of the first two layers is the same ReLU activation [(B1)] as that used in the convolutional layers, whereas the last layer has a linear activation in order for the output to represent halo mass. The weights were initialized using the same Xavier initialization technique used for the kernel weights of the convolutional layers. Regularization in the convolutional and fully connected layers was incorporated in the form of priors over the parameters of the model, as explained in the next subsection.

We chose the architecture that returned the best performance (i.e., the lowest loss score on the validation set after convergence) amongst many, but not all, alternative models with different choices of architecture-specific and layer-specific hyperparameters. We investigated the change in the validation loss in response to the following modifications; adding batch-normalization layers, introducing dropout, varying the amount of dropout, adding/removing convolutional layers and/or fully connected layers, increasing/decreasing the number of kernels/neurons in each convolutional/fully connected layer, changing the weight initializer, and changing the convolutional kernel size. In all cases, we found that the final loss score either increased or showed no change compared to that of the architecture retained in this work. We leave further hyperparameter exploration, including changes to the optimizer, the addition of skip connections, and other variations in the architecture, to future work.

APPENDIX C: THE LOSS FUNCTION

A neural network can be viewed as a probabilistic model $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$, where given an input $\mathbf{x}$, a neural network assigns a probability to each possible output y, using the set of parameters $\boldsymbol{w}$. The parameters are learned via maximum likelihood estimation (MLE); given a set of training examples $\mathcal{D} = \{x_i, d_i\}_{i=1}^{N}$, the optimal weights are those that minimize the negative log-likelihood, $\ln[p(\mathcal{D}|\boldsymbol{w})] = \sum_{i=1}^{N} \ln[p(d_i|x_i, \boldsymbol{w})]$, more generally called the loss function $\mathcal{L}$ in the machine learning community. The issue is that deep neural networks are generally overparametrized; it has in fact been demonstrated that there exists a major redundancy in the parameters used by a deep neural network [51]. This means that when minimizing the negative log-likelihood with a deep neural network model, one almost always encounters the problem of overfitting. The algorithm tends to fit the samples of the training data $\mathcal{D}$ extremely well but fails to learn patterns that are generalizable to unseen data. To overcome this issue, one modifies the loss function of the neural network in such a way that prevents the algorithm from overfitting and improves its generalizability. This is known as regularization.

We introduce regularization by adopting priors over the weights. Following Bayes' theorem, the goal of the neural network then becomes to maximize the posterior distribution $p(\boldsymbol{w}|\mathcal{D}) = p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$, rather than the likelihood $p(\mathcal{D}|\boldsymbol{w})$. The loss function, $\mathcal{L}$, is then given by

$$\mathcal{L} = -\ln\left[p(\boldsymbol{w}|\mathcal{D})\right] = -\ln\left[p(\mathcal{D}|\boldsymbol{w})\right] - \ln\left[p(\boldsymbol{w})\right], \tag{C1}$$

where the first is the likelihood term, or predictive term $\mathcal{L}_{\mathrm{pred}}$, and the second is the prior term, or regularization term $\mathcal{L}_{\mathrm{reg}}$, as in (1). If $\boldsymbol{w}$ are given a Gaussian prior, this yields L2 regularization; if $\boldsymbol{w}$ are given a Laplacian prior, then one obtains L1 regularization. The advantage of this form of regularization is that it can be incorporated in terms of priors on the weights. There exist many other regularization techniques, including for example dropout, but we choose to focus on those that have a direct Bayesian interpretation.

Technically, the parameters optimized during training include not just the weights, but also the biases. These consist of a constant value that is added to the product of inputs and weights for every kernel (neuron) in a convolutional (fully connected) layer. These parameters add little flexibility to the model and are therefore typically not responsible for overfitting. Therefore, we choose not to consider setting priors on the biases as they do not require regularization.

1. The choice of the likelihood function

We denote $d = d(\mathbf{x})$ as the ground truth variable rescaled to $[-1, 1]$ and $y = y(\mathbf{x}, \boldsymbol{w})$ as the prediction returned by the CNN model with weights $\boldsymbol{w}$, for a given set of rescaled inputs $\boldsymbol{x}$. The likelihood function describes the distribution of ground truth values d for a given value of predicted output y, returned from the neural network model with weights $\boldsymbol{w}$. A typical choice in the field is that of a Gaussian or Laplacian likelihood, yielding the popular mean-squared-error or mean-absolute-error losses. For our problem, we found that a Gaussian distribution is a poor description of the training data: the distributions of d for fixed values of y contain long tails, especially when y is close to the boundaries $y = -1$ and $y = 1$, that a Gaussian distribution fails to account for. Not accounting for these tails led to biased predictions, especially towards the boundaries. Instead, we choose a Cauchy distribution function for the likelihood, characterized by the scale parameter γ, which has broader tails than a Gaussian and a well-defined form for its conditional distribution function.

The negative log-likelihood then becomes

$$-\ln\left[p(d|y,S)\right] = \frac{1}{N}\sum_{i=1}^{N}\left[\ln(\gamma) + \ln\left[1 + \left(\frac{d_i - y_i}{\gamma}\right)^2\right] + \ln\left[\arctan\left(\frac{d_{\max} - y_i}{\gamma}\right) - \arctan\left(\frac{d_{\min} - y_i}{\gamma}\right)\right]\right], \tag{C2}$$

for a Cauchy likelihood function with scale parameter γ, under a top-hat selection function S over the ground truth variable, $p(S|d) = \Theta(d_{\max} - d)\Theta(d - d_{\min})$, where Θ is the Heaviside step function. The latter arises from the fact that, by construction, the rescaled ground truth is restricted to $d_{\min} \leq d \leq d_{\max}$, where $d_{\min} = -1$ and $d_{\max} = 1$. This selection function was needed in order to correctly model the loss at the boundaries. The first two terms in (C2) arise from the Cauchy likelihood; the first is effectively a prior on γ which is insensitive to the predictions y, and the second measures the difference between predicted and ground truth values weighed by the scale parameter γ. The third term in (C2) comes from accounting for the selection function S. The normalization factor $1/N$ is not part of the negative log-likelihood but is typically introduced in the loss function so that the loss is insensitive to the size of the training set (or, of the batch size if performing batch gradient descent when training). The scale parameter γ determines the half-width at half-maximum of the Cauchy distribution. Since the optimal value of γ is not known *a priori*, we optimize that parameter during training using backpropagation, together with the rest of the weights and biases optimized by the network. To test the robustness of simultaneously optimizing loss function hyperparameters and network weights, we retrained the network with γ fixed to its best-fit value and found no significant change in the performance of the network. The above likelihood term in the loss function satisfies our desiderata of having a heavy-tailed probability distribution function and accounting for the restricted range of ground truth d.

The expression in (C2) is valid under the condition that $d, y \in [-1, 1]$. However, since the activation function in the last layer is given by the unbounded linear function $\sigma(z) = z$, the predictions can technically take any value $y \in \mathbb{R}$. To solve this, we introduced a superexponential function, denoted as $f(y)$, in the regime $|y| \geq 1$, to counter balance the Cauchy limits at the boundaries and sharply disfavor predictions outside the interval $[-1, 1]$. The function is continuously matched to the Cauchy distribution at the boundaries $y = \pm 1$. The likelihood term of the loss function $\mathcal{L}_{\text{pred}}$ is then given by a piecewise function conditional on y,

$$\mathcal{L}_{\text{pred}} = \frac{1}{N}\sum_{i=1}^{N}\left[-\ln p(d_i|y_i, S)\Theta(|y| + 1) + f(y_i)\Theta(|y| - 1)\right]. \tag{C3}$$

Figure 8 compares the form of the likelihood, given the optimized value for γ returned by the model, compared to the empirical distribution of ground truth values at fixed slices in y, the predicted variable returned by the trained CNN, for the training set samples. The Cauchy likelihood provides a good fit to the empirical likelihood distribution of the model. However, we note that for small values of y, the fit could have been improved by adopting a two-tailed Cauchy likelihood function. Further flexibility could be provided by using a student's t-distribution. We leave this to future work.

2. The choice of priors: regularization and model compression

We adopt weight priors that can simultaneously (i) improve the optimization during training by preventing overfitting and (ii) compress the neural network model into the least number of parameters without loss in performance. Regularization and model compression are very much related; these tasks can be achieved simultaneously by minimizing a properly defined cost function. We selected weight priors that penalize large values (for regularization)

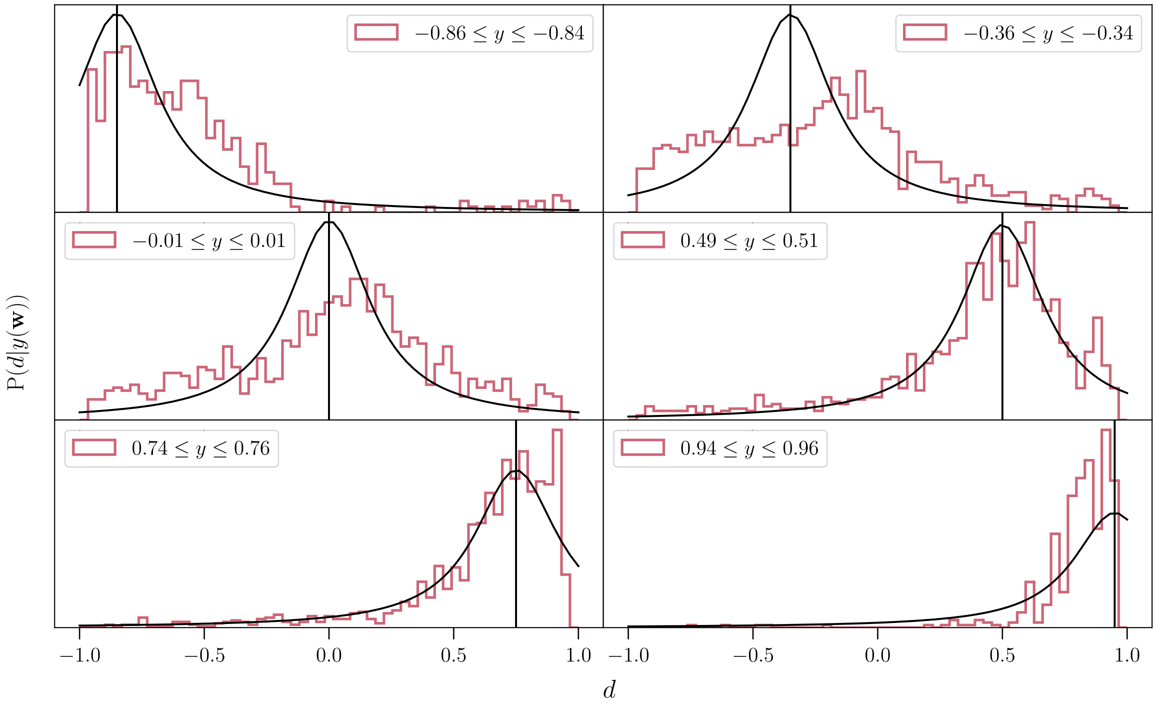

FIG. 8. Distribution of d for fixed slices in y, where d is the ground truth logarithmic halo mass variable rescaled to the range $[-1, 1]$ and y is the predicted value returned by the CNN in rescaled units. The black line is the likelihood function given the value of γ at the epoch where the validation loss reaches its minimum, $\gamma = 0.26$.

and induce sparsity (for model compression). To regularize the network, we adopted weight priors that promote smaller values, as these typically lead to more generalizable solutions. We chose Gaussian priors for the weights of the convolutional layers and Laplacian priors for the weights of the fully connected layers, which penalize the sum of the squared values or the sum of the absolute values of the weights, respectively. The choice of Laplacian prior has the additional benefit that it induces sparsity on the weights by driving most weights to be zero; a Laplacian prior thus combines the idea of model compression and regularization. For model compression, our aim is to induce a more compact network with the smallest number of nonzero neurons in the fully connected layers. To do this, we adopted the group Lasso formulation [52] which imposes group-level sparsity, meaning that all the variables in that group are either simultaneously set to 0, or none of them are. For the case of fully connected layers, a group is equivalent to an entire neuron.

The log priors over the weights become

$$\ln \mathrm{p}(\boldsymbol{w}) = \alpha\left[\sum_{l \in l_c} \sum_{p=1}^{P_l} (w_p^{(l)})^2 + \sum_{l \in l_d} \sum_{q=1}^{Q_l} |w_q^{(l)}| + \sum_{l \in l_d} \sum_{i=1}^{N_{l-1}} \left[\sum_{j=1}^{N_l} w_{ij}^2\right]^{1/2}\right], \tag{C4}$$

where the first term is a Gaussian prior over each of the P_l weights of each convolutional layer l_c, $w_p^{(l)}$, the second term is a Laplacian prior over each of the Q_l weights of each fully connected layer l_d, $w_q^{(l)}$, and the third term is a group Lasso prior over the set of weights that determine the connections between a single neuron in the $(l-1)$th layer and all the neurons in the lth layer. The idea of group-level sparsity can also be applied to convolutional layers, where a single group is given by the collection of weights from a single kernel of the layer. This can be thought of as a feature selection method, in that it removes entire kernels (and thus the feature represented by that kernel) within each convolutional layer. Given that our network is relatively small, we chose not to perform feature selection; we leave this for future work.

The prior term $\ln[\mathrm{p}(\boldsymbol{w})]$ is added to the likelihood term in the loss function, as in (C1). The regularization parameter α in (C4) weighs the prior term relative to the likelihood term in the loss function. Its value sets the balance between an overly-complex model (which overfits and has a high variance) and an overly simple model (which underfits and has a high bias). We optimized this parameter, in combination with the learning rate, using cross-validation.

APPENDIX D: TRAINING AND OPTIMIZATION

The algorithm was trained on 200,000 particles, randomly drawn from the ensemble of particles of 20 simulations based on different realizations of the initial conditions. We validated the model using 10,000 particles from a single simulation, and tested it on 99,950 particles from four independent simulations. We did not perform data augmentation to increase the size of the training set since we had available a large number of training samples and therefore opted for testing the impact of adding new (independent) samples instead. We found no improvement in the performance of the algorithm as we added to the training set an additional 300,000 particles from another independent simulation, implying that our choices were sufficient to yield a training set representative of the mapping between initial conditions and halos. We further investigated changes to the training set, such as re-balancing the training set to have the same number of particles in low- and high-mass halos, and found no change in the accuracy of the predictions. The training set was subdivided into batches, each made of 64 particles. Batches were fed to the network one at a time, and each time the CNN updates its parameters according to the samples in that batch.

Training was done using the `AMSGrad` optimizer [34], a variant of the widely-used `Adam` optimizer [35], with a learning rate of 0.00005. The learning rate was optimized via cross-validation, together with α, the parameter weighting the regularization term in the loss function. The number of trained parameters in the network is 2,108,258. Figure 9 shows the loss function (*upper panel*) evaluated for the training and validation sets, and the value of the parameter γ in the likelihood term of the loss (*lower panel*) as a function of the number of iterations. Early stopping was employed to interrupt the training at the epoch where the validation loss reaches its minimum value; the early-stopping iteration is shown as a vertical gray line in Fig. 9. The final weights of the CNN and the optimized value of γ are given by those characterizing the model at the end of the early stopping iteration.

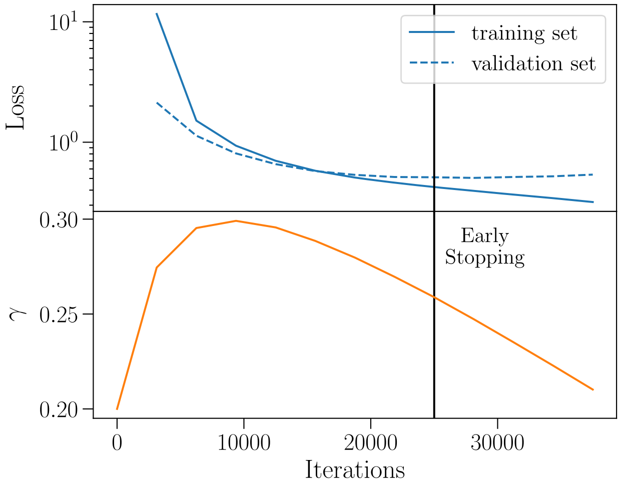

FIG. 9. Top panel: The loss function evaluated for the training set and the validation set at each batch iteration. Early stopping was employed to interrupt the training at the epoch where the validation loss reaches its minimum. The weights of the CNN at the early-stopping iteration were retained. Bottom panel: The half-width at half-maximum parameter of the Cauchy likelihood function, γ, as a function of iteration.

Validation and testing was performed on particles from an independent simulation based on a different realization of the initial density field to those used for training. Although the validation set does not directly enter the training process of the algorithm, it is indirectly used to test the response of the algorithm to changes in the architecture, and to determine the stopping point for training. Validating and testing on independent realizations ensures that the algorithm is not overfitting patterns specific to the simulations used for training. Instead, it ensures that the CNN is learning physical connections between the initial conditions and the final halos which are generalizable to any realization of the initial density field.

APPENDIX E: A COMPARISON WITH ANALYTIC MODELS

We compared the accuracy of the CNN predictions against that of analytic models which also provide final halo mass predictions from the initial conditions. This serves as a validation test for our CNN model. We expect the CNN model to return halo mass predictions that are at least as accurate as those of state-of-the-art analytic approximations, since both CNN models have access to the spherically-averaged density around each particle which is what is used by the analytic models to predict the final halo mass.

We compared the CNN to the extended Press-Schechter (EPS) [6] and Sheth-Tormen (ST) [11] analytic halo-collapse models. According to EPS, the fraction of density trajectories with a first upcrossing of a density threshold barrier $\delta_{\rm th}$ is equivalent to the fraction of haloes of mass M. The density threshold barrier $\delta_{\rm th}$ adopted by Bond *et al.* [6] is that of spherical collapse; $\delta_{\rm th}(z) = (D(z)/D(0))\delta_{\rm sc}$, where $\delta_{\rm sc} \approx 1.686$. The predicted halo mass of each test particle is given by the smoothing mass scale at which the particle first upcrosses the density threshold barrier.

In the ST formalism, EPS theory is extended by adopting a “moving” collapse barrier rather than the spherical collapse barrier. The ST collapse barrier $b(z)$ varies as a function of the mass variance $\sigma^2(M)$ and is given by

$$b(z) = \sqrt{a}\delta_{\rm sc}(z)\left[1 + \left(\beta\frac{\sigma^2(M)}{a\delta_{\rm sc}^2(z)}\right)^{\gamma}\right], \qquad \text{(E1)}$$

where $\delta_{\rm sc}(0) \approx 1.686$ and the best-fit parameters found in Sheth *et al.* [10] are $\beta = 0.485$, $\gamma = 0.615$ and $a = 0.707$. Similar to the EPS case, the predicted halo mass of each test particle is given by the smoothing mass scale at which the particle first upcrosses the threshold barrier given by

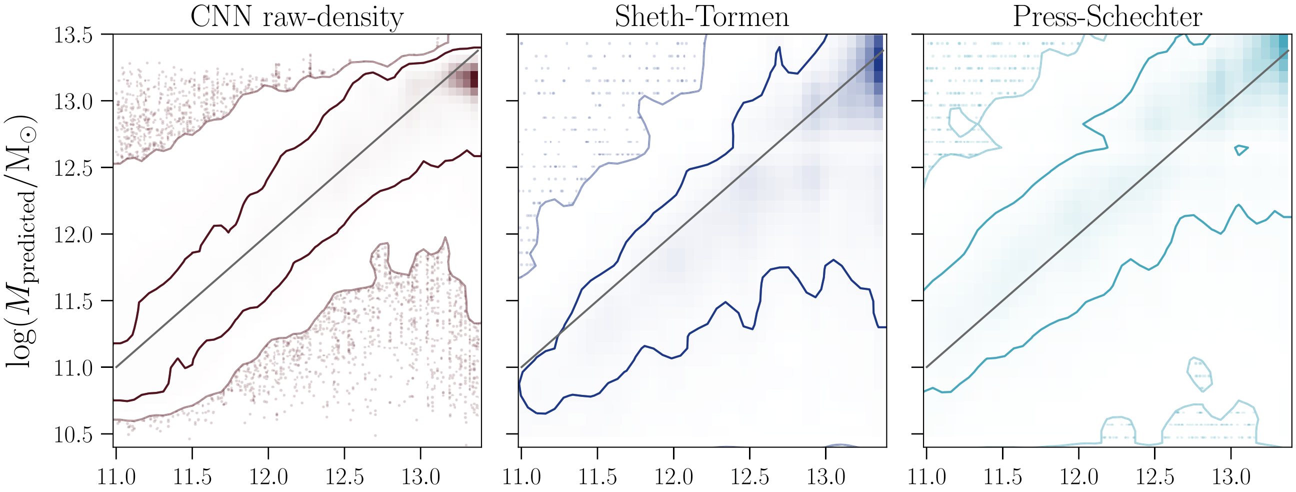

FIG. 10. Two-dimensional histograms and contours containing 68% and 95% of the joint probability of the predicted vs true halo masses for the analytic and CNN models. We compare the raw-density CNN predictions (left panel) with those from the Sheth-Tormen (middle panel) and extended Press-Schechter (right panel) analytic models. The predictions are qualitatively similar, but with tighter confidence regions for the CNN case. This validates our results from the CNN as we find no evidence of any lack of predictive performance from the CNN compared to established analytic models of the same mapping.

Eq. (E1). Note that 30% (58%) of particles in the test set have trajectories that never cross the EPS (ST) collapse barrier for the smoothing mass scales we consider, and so these do not have an associated mass prediction. This selection bias will be relevant when making quantitative comparisons with the CNN models in terms of MI, as described below.

We computed the EPS and ST predicted halo masses for the particles in the test set used to test the CNN. Figure 10 shows the predicted halo masses as a function of true halo masses for the analytic and CNN models. We show two-dimensional histograms and the contours containing 68% and 95% of the joint probability. All models show qualitatively similar predictions, but with tighter confidence regions for the CNN predictions. This is especially notable in the bottom-right region of the middle and right panels, where the analytic models' predictions extend to much lower mass values than the CNN predictions. The CNN shows a slight tendency to underpredict the mass of high-mass halos as these are close to the edge of the training set mass range; although our loss function was designed to mitigate this common bias effect in CNNs, it does not entirely remove it. The ST predictions are shifted towards lower mass values compared to the EPS predictions, for fixed true halo mass. This is because the ST collapse barrier takes larger δ values than the EPS barrier at fixed smoothing mass scale; as a result, the same particle will cross the collapse barrier at lower smoothing mass scales for ST compared to EPS. This in turn yields a lower halo mass prediction for ST compared to EPS.

In Tables I and II, we quantitatively compare the performance of the CNN models with the analytic ones in terms of the MI between predicted and ground truth halo mass values. When making a quantitative comparison, one must consider that particles whose trajectories do not cross the EPS (or ST) collapse barrier do not have an associated halo mass prediction. By contrast, the CNN will always return a mass prediction for every particle used for testing. Therefore, we compare the performance of EPS to that of the CNN models for only those subset of particles in the test set that have a predicted mass value according to EPS, and report the MI values in Table I. We do the same for ST and report the MI values in Table II. We find that the raw density and averaged density models perform better than state-of-the-art analytic models for the same set of particles, as the MI of the CNN models is higher than that of the analytic models.

This test validates our results as it confirms that the CNN is at least as accurate as the analytic models.

TABLE I. MI (in nats) between predicted and ground truth halo mass values for the raw-density, averaged-density and EPS models. The MI computation includes only test set particles which have a prediction under the EPS model.

Raw density	Averaged density	EPS
0.409 ± 0.004	0.271 ± 0.004	0.257 ± 0.005

TABLE II. MI (in nats) between predicted and ground truth halo mass values for the raw-density, averaged-density and ST models. The MI computation includes only test set particles which have a prediction under the ST model.

Raw density	Averaged density	ST
0.411 ± 0.004	0.299 ± 0.004	0.260 ± 0.005

[1] G. Efstathiou, M. Davis, S. D. M. White, and C. S. Frenk, Numerical techniques for large cosmological N-body simulations, Astrophys. J. Suppl. Ser. **57**, 241 (1985).

[2] A. Jenkins, C. S. Frenk, S. D. M. White, J. M. Colberg, S. Cole, A. E. Evrard, H. M. P. Couchman, and N. Yoshida, The mass function of dark matter haloes, Mon. Not. R. Astron. Soc. **321**, 372 (2001).

[3] J. F. Navarro, C. S. Frenk, and S. D. M. White, A universal density profile from hierarchical clustering, Astrophys. J. **490**, 493 (1997).

[4] V. Springel, N. Yoshida, and S. D. M. White, GADGET: A code for collisionless and gasdynamical cosmological simulations, New Astron. **6**, 79 (2001).

[5] W. H. Press and P. Schechter, Formation of galaxies and clusters of galaxies by self-similar gravitational condensation, Astrophys. J. **187**, 425 (1974).

[6] J. R. Bond, S. Cole, G. Efstathiou, and N. Kaiser, Excursion set mass functions for hierarchical Gaussian fluctuations, Astrophys. J. **379**, 440 (1991).

[7] A. G. Doroshkevich, The space structure of perturbations and the origin of rotation of galaxies in the theory of fluctuation, Astrofiz. **6**, 581 (1970).

[8] J. R. Bond and S. T. Myers, The peak-patch picture of cosmic catalogs. I. Algorithms, Astrophys. J. Suppl. Ser. **103**, 1 (1996).

[9] R. K. Sheth and G. Tormen, Large-scale bias and the peak background split, Mon. Not. R. Astron. Soc. **308**, 119 (1999).

[10] R. K. Sheth, H. J. Mo, and G. Tormen, Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes, Mon. Not. R. Astron. Soc. **323**, 1 (2001).

[11] R. K. Sheth and G. Tormen, An excursion set model of hierarchical clustering: Ellipsoidal collapse and the moving barrier, Mon. Not. R. Astron. Soc. **329**, 61 (2002).

[12] L. Lucie-Smith, H. V. Peiris, A. Pontzen, and M. Lochner, Machine learning cosmological structure formation, Mon. Not. R. Astron. Soc. **479**, 3405 (2018).

[13] L. Lucie-Smith, H. V. Peiris, and A. Pontzen, An interpretable machine-learning framework for dark matter halo formation, Mon. Not. R. Astron. Soc. **490**, 331 (2019).

[14] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[15] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. **2**, 1 (2009).

[16] S. Ravanbakhsh, J. Oliva, S. Fromenteau, L. C. Price, S. Ho, J. Schneider, and B. Póczos, Estimating cosmological parameters from the dark matter distribution, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning—Volume 48, ICML'16* (2016), p. 2407–2416, JMLR.org.

[17] A. Mathuriya, D. Bard, P. Mendygral, L. Meadows, J. Arnemann, L. Shao, S. He, T. Kärnä, D. Moise, S. J. Pennycook, K. Maschhoff, J. Sewall, N. Kumar, S. Ho, M. F. Ringenburg, Prabhat, and V. Lee, Cosmoflow: Using deep learning to learn the universe at scale, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC '18* (IEEE Press, 2018).

[18] S. Pan, M. Liu, J. Forero-Romero, C. G. Sabiu, Z. Li, H. Miao, and X.-D. Li, Cosmological parameter estimation from large-scale structure deep learning, Sci. China Phys. Mech. Astron. **63**, 110412 (2020).

[19] F. Villaescusa-Navarro, B. D. Wandelt, D. Anglés-Alcázar, S. Genel, J. M. Zorrilla Mantilla, S. Ho, and D. N. Spergel, Neural networks as optimal estimators to marginalize over baryonic effects., Astrophys. J. **928**, 44 (2022).

[20] M. Ntampaka, D. J. Eisenstein, S. Yuan, and L. H. Garrison, A hybrid deep learning approach to cosmological constraints from galaxy redshift surveys, Astrophys. J. **889**, 151 (2020).

[21] B. P. Moster, T. Naab, M. Lindström, and J. A. O'Leary, Galaxynet: Connecting galaxies and dark matter haloes with deep neural networks and reinforcement learning in large volumes, Mon. Not. R. Astron. Soc. **507**, 2115 (2021).

[22] D. Kodi Ramanah, T. Charnock, F. Villaescusa-Navarro, and B. D. Wandelt, Super-resolution emulator of cosmological simulations using deep physical models, Mon. Not. R. Astron. Soc. **495**, 4227 (2020).

[23] Y. Li, Y. Ni, R. A. C. Croft, T. Di Matteo, S. Bird, and Y. Feng, Ai-assisted superresolution cosmological simulations, Proc. Natl. Acad. Sci. U.S.A. **118**, e2022038118 (2021).

[24] S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen, and B. Póczos, Learning to predict the cosmological structure formation, Proc. Natl. Acad. Sci. U.S.A. **116**, 13825 (2019).

[25] X. Zhang, Y. Wang, W. Zhang, Y. Sun, S. He, G. Contardo, F. Villaescusa-Navarro, and S. Ho, From dark matter to galaxies with convolutional networks., arXiv:1902.05965.

[26] D. Kodi Ramanah, T. Charnock, and G. Lavaux, Painting halos from cosmic density fields of dark matter with physically motivated neural networks, Phys. Rev. D **100**, 043515 (2019).

[27] V. Springel, The cosmological simulation code GADGET-2, Mon. Not. R. Astron. Soc. **364**, 1105 (2005).

[28] A. Pontzen, R. Roškar, G. Stinson, and R. Woods, pynbody: N-Body/SPH analysis for PYTHON, Astrophysics Source Code Library (2013).

[29] J. Dunkley, E. Komatsu, M. R. Nolta, D. N. Spergel, D. Larson, G. Hinshaw, L. Page, C. L. Bennett, B. Gold, N. Jarosik, J. L. Weiland, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, G. S. Tucker, E. Wollack, and E. L. Wright, Five-Year Wilkinson Microwave Anisotropy Probe Observations: Likelihoods and Parameters from the WMAP Data, Astrophys. J. Suppl. Ser. **180**, 306 (2009).

[30] Planck Collaboration and others *et al.*, Planck 2018 results. VI. Cosmological parameters, Astron. Astrophys. **641**, A6 (2020).

[31] S. Stopyra, A. Pontzen, H. Peiris, N. Roth, and M. P. Rey, Genetic–a new initial conditions generator to support genetically modified zoom simulations, Astrophys. J. Suppl. Ser. **252**, 28 (2021).

[32] V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10* (Omnipress, USA, 2010), pp. 807–814.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, Nature (London) **323**, 533 (1986).

[34] S. J. Reddi, S. Kale, and S. Kumar, On the convergence of adam & beyond, International Conference on Learning Representations (2018).

[35] D. Kingma and J. Ba, Adam: A method for stochastic optimization, *International Conference on Learning Representations* (2014).

[36] D. Piras, H. V. Peiris, A. Pontzen, L. Lucie-Smith, N. Guo, and B. Nord, A robust estimator of mutual information for deep learning interpretability, Mach. Learn. **4**, 025006 (2023).

[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, Intriguing properties of neural networks, arXiv:1312.6199.

[38] J. Fluri, T. Kacprzak, A. Refregier, A. Amara, A. Lucchi, and T. Hofmann, Cosmological constraints from noisy convergence maps through deep learning, Phys. Rev. D **98**, 123518 (2018).

[39] J. Schmelzle, A. Lucchi, T. Kacprzak, A. Amara, R. Sgier, A. Réfrégier, and T. Hofmann, Cosmological model discrimination with Deep Learning, arXiv:1707.05167.

[40] P. Monaco, T. Theuns, and G. Taffoni, The pinocchio algorithm: Pinpointing orbit-crossing collapsed hierarchical objects in a linear density field, Mon. Not. R. Astron. Soc. **331**, 587 (2002).

[41] E. Castorina, A. Paranjape, O. Hahn, and R. K. Sheth, Excursion set peaks: The role of shear, arXiv:1611.03619.

[42] A. Paranjape, R. K. Sheth, and V. Desjacques, Excursion set peaks: A self-consistent model of dark halo abundances and clustering, Mon. Not. R. Astron. Soc. **431**, 1503 (2013).

[43] J. R. Bond and S. T. Myers, The peak-patch picture of cosmic catalogs. II. Validation, Astrophys. J. Suppl. Ser. **103**, 41 (1996).

[44] G. Stein, M. A. Alvarez, and J. R. Bond, The mass-peak patch algorithm for fast generation of deep all-sky dark matter halo catalogues and itsn-body validation, Mon. Not. R. Astron. Soc. **483**, 2236 (2018).

[45] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*(2010), pp. 249–256, http://proceedings.mlr.press/v9/glorot10a.html.

[46] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, arXiv:1811.03378.

[47] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, Predicting parameters in deep learning, NIPS'13, in *Proceedings of the 26th International Conference on Neural Information Processing Systems* (2013), pp. 2148–2156.

[48] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, Group sparse regularization for deep neural networks, Neurocomputing;Variable Star Bulletin **241**, 81 (2017).

[49] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, *JMLR Workshop and Conference Proceedings* (Chia Laguna Resort, Sardinia, Italy, 2010), pp. 249–256.

[50] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, arXiv:1811.03378.

[51] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, Predicting parameters in deep learning, in *Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, NIPS'13* (Curran Associates Inc., Red Hook, NY, USA, 2013), pp. 2148–2156.

[52] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, Group sparse regularization for deep neural networks, Neurocomputing;Variable Star Bulletin **241**, 81 (2017).