

# Improved Cosmic Displacement Field Estimation

Astrophysics Bachelor of Science Thesis

by

**Andy Nilipour**

Advisor: Nikhil Padmanabhan



DEPARTMENT OF ASTRONOMY  
YALE COLLEGE

May 5, 2025

# Improved Cosmic Displacement Field Estimation

**Andy Nilipour, Farnik Nikakhtar, Nikhil Padmanabhan**

Dept. of Physics, Yale University, New Haven, CT 06511, USA

E-mail: [andy.nilipour@yale.edu](mailto:andy.nilipour@yale.edu), [farnik.nikakhtar@yale.edu](mailto:farnik.nikakhtar@yale.edu),  
[nikhil.padmanabhan@yale.edu](mailto:nikhil.padmanabhan@yale.edu)

**Abstract.** Reconstruction of the primordial density field from the observed, late-time density field is an essential step in modern cosmological analyses of galaxy surveys, including with the Baryon Acoustic Oscillation (BAO) feature. However, standard reconstruction, which uses the Zel'dovich approximation, or first order Lagrangian perturbation theory (LPT), to estimate the displacement field mapping observed galaxies back to their initial positions, relies on large-scale structure physics and tend to struggle at smaller scales, where nonlinearities dominate. Several recent works have implemented deep learning models, trained on cosmological N-body simulations, to learn the initial density field from the late-time field. Expanding on this idea, we present a method to reconstruct the displacement field from the initial conditions using a convolutional neural network. We show that our model improves estimation of the displacement field compared to LPT, even to higher order. We also apply our framework to learn the optimal transport displacement field, which more closely approximates the true displacement field than the Zel'dovich approximation while still being sourced by a scalar potential. By enabling rapid forward modeling of initial conditions to a late-time displacement field, this work highlights the growing ability of machine learning to extract information at nonlinear scales and empower precision field-level inference.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>2</b>
2.1	LPT	2
2.2	<i>N</i> -Body and Particle Mesh Simulations	4
2.3	BAO Reconstruction	5
2.4	Optimal Transport Displacement Field	6
2.5	Convolutional Neural Networks	7
2.6	Previous Literature	8
<b>3</b>	<b>LPT Tests</b>	<b>11</b>
3.1	Data	11
3.2	LPT Code	12
3.3	Results	15
<b>4</b>	<b>CNN</b>	<b>18</b>
4.1	Data	18
4.2	Model	20
4.2.1	$128^3$ Grid	20
4.2.2	Scaling to Larger Grids	21
4.3	Training	21
4.4	Results	23
<b>5</b>	<b>Conclusions</b>	<b>29</b>
<b>A</b>	<b>Further Considerations</b>	<b>31</b>
A.1	HADES Deviations from Linear Theory Power Spectrum	31

---

## 1 Introduction

Although traditional analysis of cosmological galaxy surveys typically relies on summary statistics, such as power spectra, fully extracting information about the formation and evolution of large-scale structure from modern observations requires accurate methods of simulation [1–6]. *N*-body simulations, which numerically evolve the positions and velocities of millions to billions of particles under the nonlinear effects of gravity, are the most effective such simulations [7, 8]. However, *N*-body simulations are computationally expensive, so using them to constrain cosmological parameters by comparing the observed Universe to many simulated Universes is difficult. Several methods, such as Lagrangian Perturbation Theory (LPT) [9–12] and particle mesh (PM) codes [13–16], approximate the full evolution but tend to struggle at small scales, where nonlinearities dominate. Recent work has sought to use machine learning, either on its own or as an augmentation to less accurate methods, to accelerate this task of forward modeling, that is, predicting structure formation from initial conditions, as a cheap and accurate alternative to *N*-body simulations [17–22]

Here, we propose the use of a machine learning model to predict the displacement field, which maps the primordial density field to a late-time, evolved density field, from initial conditions. In the context of  $N$ -body simulations, the displacement field records the change in position of each particle from initial to final conditions. Thus, such a model can enable rapid forward modeling of the initial density field at almost no computational cost.

One important use of the displacement field is for measurements of the baryon acoustic oscillation (BAO) feature in galaxy spectroscopic surveys, which allow for constraints on the Hubble parameter and the accelerating expansion of the universe [23–26]. An essential step in standard BAO analysis is reconstruction, which attempts to reverse the smearing effects of nonlinear gravitational evolution on the BAO feature in the two-point correlation function [27]. Standard reconstruction requires using an estimated displacement field to move the observed galaxies and overdensities back to their initial positions. This method, and most subsequent reconstruction methods, estimate the displacement field from the observed nonlinear density field [28–31], typically with first- or second-order LPT. Another class of methods uses the framework of optimal transport theory to estimate the displacements, although still from the late-time density [32–34]. Machine learning has also been applied for such reconstruction [35–37]. Our approach, in which we predict the displacements from the initial density field rather than the evolved field, would allow for sampling and forward modeling of the initial conditions to perform reconstruction. Indeed, such methods have been explored with LPT or PM codes as the forward model [38–40]. The advantage of this forward modeling approach is that, in addition to powering BAO reconstruction, it can also enable field-level inference, allowing constraints on cosmological parameters from the full field rather than summary statistics such as the BAO scale or power spectrum.

In this work, we seek to learn how well the nonlinear displacement field can be estimated from the initial density field using theory (i.e., LPT) and theory augmented with machine learning. To this end, we train a convolutional neural network (CNN) to learn the nonlinear displacement field from the initial conditions and compare the results to the displacement field estimates from the first three orders of LPT. The paper is organized as follows. In Section 2, we summarize the theoretical and computational background behind the work and review related work in the literature. Section 3 describes our methods and results for testing LPT. Section 4 describes the architecture, training, and results of our CNN model. We conclude and discuss avenues for future investigation in Section 5 .

## 2 Background

### 2.1 LPT

In the framework of LPT, which follows particle trajectories rather than the evolution of density and velocity fields, the initial positions  $\mathbf{q}$  are mapped onto the final Eulerian positions  $\mathbf{x}$  by a displacement field  $\Psi(\mathbf{q}, t)$ , as

$$\mathbf{x}(\mathbf{q}, t) = \mathbf{q} + \Psi(\mathbf{q}, t). \quad (2.1)$$

The solution to  $\Psi(\mathbf{q}, t)$  is expanded perturbatively. The first-order solution, known as the Zel'dovich approximation [41], is given by

$$\nabla_{\mathbf{q}} \cdot \Psi^{(1)}(\mathbf{q}, t) = -D_1(t)\delta(\mathbf{q}), \quad (2.2)$$

where  $\delta(\mathbf{q})$  is the initial density field and  $D_1(t)$  is the linear growth factor [9]. The first- and second-order solutions are curl-free, so they allow for scalar potentials  $\phi^{(1)}$  and  $\phi^{(2)}$  obeying the Poisson equations

$$\nabla_{\mathbf{q}}^2 \phi^{(1)}(\mathbf{q}) = \delta(\mathbf{q}), \quad (2.3)$$

$$\nabla_{\mathbf{q}}^2 \phi^{(2)}(\mathbf{q}) = \sum_{i>j} \left[ \phi_{,ii}^{(1)}(\mathbf{q}) \phi_{,jj}^{(1)}(\mathbf{q}) - (\phi_{,ij}^{(1)}(\mathbf{q}))^2 \right], \quad (2.4)$$

where  $\phi_{,ij}^{(1)} = \frac{\partial^2 \phi^{(1)}}{\partial i \partial j}$ .

Then, the first- and second-order displacement fields are

$$\Psi^{(1)}(\mathbf{q}, t) = -D_1(t) \nabla_{\mathbf{q}} \phi^{(1)}, \quad (2.5)$$

$$\Psi^{(2)}(\mathbf{q}, t) = -D_1(t) \nabla_{\mathbf{q}} \phi^{(1)} + D_2(t) \nabla_{\mathbf{q}} \phi^{(2)}, \quad (2.6)$$

where  $D_2(t)$  is the second-order growth factor and is approximately  $D_2(t) \approx -\frac{3}{7} D_1^2(t) \Omega_m^{-2/63}$ .

Although the third-order displacement field is not curl-free, it can be expressed explicitly as

$$\Psi^{(3)}(\mathbf{q}, t) = \Psi^{(2)}(\mathbf{q}, t) + D_3(t) \left( \frac{1}{3} \nabla_{\mathbf{q}} \phi^{(3a)}(\mathbf{q}) - \frac{10}{21} \nabla_{\mathbf{q}} \phi^{(3b)}(\mathbf{q}) + \frac{1}{7} \nabla_{\mathbf{q}} \times \mathbf{A}^{(3c)}(\mathbf{q}) \right), \quad (2.7)$$

where the scalar and vector potentials obey the Poisson equations

$$\nabla_{\mathbf{q}}^2 \phi^{(3a)}(\mathbf{q}) = \det \left[ \phi_{,ij}^{(1)}(\mathbf{q}) \right], \quad (2.8)$$

$$\nabla_{\mathbf{q}}^2 \phi^{(3b)}(\mathbf{q}) = \frac{1}{2} \sum \left[ \phi_{,ii}^{(2)}(\mathbf{q}) \phi_{,jj}^{(1)}(\mathbf{q}) - \phi_{,ij}^{(2)}(\mathbf{q}) \phi_{,ji}^{(1)}(\mathbf{q}) \right], \quad (2.9)$$

$$\nabla_{\mathbf{q}}^2 \mathbf{A}^{(3c)}(\mathbf{q}) = \sum_i \left[ \nabla \phi_{,i}^{(2)}(\mathbf{q}) \times \nabla_{\mathbf{q}} \phi_{,i}^{(3)}(\mathbf{q}) \right], \quad (2.10)$$

and the third-order growth factor is approximately  $D_3(t) \approx D_1^3(t)$  [42].

Higher-order solutions have also been calculated, for example by [43]. There also exist recursive relations for these higher-order solutions [44, 45]. However, while this perturbative model is known to apply well at large scales, it is also known to generally become invalid at small scales, where nonlinear effects from gravity dominate and the motion of particles becomes nonperturbative [20]. This is true even for higher order LPT.

Differential operators can be computed with respect to both  $\mathbf{q}$  and  $\mathbf{x}$ ; for simplicity, we will henceforth refer to  $\nabla_{\mathbf{q}}$  as  $\nabla$ . We will also refer to fields in Fourier space as, for example,  $\tilde{\delta}(\mathbf{k})$ . In Fourier space, the gradient operator becomes  $\nabla \tilde{f}(\mathbf{k}) = i\mathbf{k} \tilde{f}(\mathbf{k})$  and likewise the divergence operator becomes  $\nabla \cdot \tilde{f}(\mathbf{k}) = i\mathbf{k} \cdot \tilde{f}(\mathbf{k})$ . We thus often move fields in and out of Fourier space using the Fast Fourier Transform (FFT) and inverse FFT (IFFT). We use the Python implementation of the FFTW library, PYFFTW<sup>1</sup>. The power spectrum of a field  $f$ , which encodes the variance of fluctuations in the field at a given spatial frequency, is  $P_f(k) = \left\langle \left| \tilde{f}(\mathbf{k}) \right|^2 \right\rangle$ .

---

<sup>1</sup><https://github.com/pyFFTW/pyFFTW>

## 2.2 *N*-Body and Particle Mesh Simulations

Most models for large-scale structure formation point to nearly uniform initial conditions. Thus, cosmological simulations for structure formation typically start with particles exactly on a grid with periodic boundary conditions, so that the initial density field is homogeneous, and perturbations in the initial velocity field drive subsequent evolution [46]. The initial displacements and velocities are generally derived from the gradient of a potential field, with the two most common cases being a Zel'dovich potential or a second-order LPT potential. For the former case, Equation 2.3 is solved for  $\tilde{\phi}^{(1)}(\mathbf{k})$  in Fourier space, where  $\tilde{\delta}(\mathbf{k})$  is generated from the linear theory power spectrum as a Gaussian density field. This is in turn used to solve Equation 2.4 for  $\tilde{\phi}^{(2)}(\mathbf{k})$  for the latter case. Then, for both types of initial conditions, the particles are displaced according to Equations 2.5 and 2.6, respectively, and are also given velocities that obey

$$\mathbf{v}(\mathbf{q}, t) = \frac{d\mathbf{x}}{dt} = -D_1(t)(af_1H)_{\text{init}}\nabla\phi^{(1)}(\mathbf{q}) + D_2(t)(af_2H)_{\text{init}}\nabla\phi^{(2)}(\mathbf{q}), \quad (2.11)$$

where the scaling factors are  $(af_iH)_{\text{init}} = a(d\ln D_i/d\ln a)(d\ln a/dt)$ , and the second term is only included for the second-order LPT initial conditions [47]. The simulation should start with initial conditions at some suitably small scale factor  $a$  (equivalently, a suitably large redshift  $z$ , since  $a = \frac{1}{1+z}$ ), where perturbations are nearly linear, but numerical effects often prevent simulations from starting at very early cosmic time [48].

*N*-body codes generate the initial conditions,  $\tilde{\delta}(\mathbf{k})$ , from the linear theory power spectrum, which is itself calculated via Boltzmann codes such as CAMB [49] or CLASS [50]. Typically, these codes are used to generate the matter power spectrum at  $z = 0$ , which is then rescaled to the initial conditions redshift,  $z_i$ , via

$$P_m(k, z_i) = \left( \frac{D_1(z_i)}{D_1(z=0)} \right)^2 P_m(k, z=0). \quad (2.12)$$

At the most fundamental level, all *N*-body codes perform two functions at each time step: they compute the gravitational field for the current configuration of particles, then move the particles by integrating the equations of motion. Although there are several methods for calculating the gravitational field, two common categories of algorithms are the hierarchical tree algorithms and the PM codes [8]. At each step, PM codes calculate the density field on a grid, solve Poisson's equation for the gravitational field on the same grid, then interpolate the gravitational field to the particle positions. The advantage of PM codes is that calculations on an exact grid can be simplified and dramatically accelerated by performing them in Fourier space, which also makes periodic boundary conditions automatic (and effectively mandatory). The primary disadvantage is that spatial resolution is limited, because the approximated gravitational force is generally inaccurate for scales below a few grid spacings.

On the other hand, the tree algorithms are not limited in resolution. These algorithms work by recursively subdividing space into cells such that particles in a cell far from where the force is to be calculated are grouped into one pseudoparticle, for which gravity is computed as a single multipole expansion of the force [8, 51, 52]. Though not as fast as PM codes, the tree algorithms are highly accurate in their force calculations. More recent codes have combined these two algorithms, using the tree method for short-range scales and the PM method for the long-range scales [53].

Modern *N*-body codes also generally incorporate fluid dynamics in addition to gravitational physics. Unlike gravity, which is fundamentally treated the same in all *N*-body codes,

the treatment of fluid dynamics can vary greatly. Two broad categories are the Eulerian methods, which computes hydrodynamic variables on a grid, and the Lagrangian methods, which follow hydrodynamic variables on particles that model fluid elements [8, 52].

For the full  $N$ -body simulations in this work, we use part of the HADES<sup>2</sup> simulation suite [54], which each contain  $512^3$  particles in a periodic box of length  $L = 1000 \text{ h}^{-1} \text{ Mpc}$  along each dimension. The HADES simulations use Zel'dovich initial conditions at  $z_i = 99$ , and a fiducial cosmology of  $\Omega_m = 0.3175$ ,  $\Omega_b = 0.049$ ,  $\Omega_\Lambda = 0.6825$ ,  $n_s = 0.9624$ ,  $h = 0.6711$ . These simulations are run with the GADGET-III code [52], which uses a hybrid tree and PM algorithm for gravity (TreePM, [53]) and a Lagrangian smoothed particle hydrodynamics (SPH) algorithm for fluid dynamics. The matter power spectra for the initial conditions are calculated with CAMB.

For faster model training, we also desire simulations with a smaller number of particles. To this end, we use the code PMWD<sup>3</sup>, which is a memory-efficient, GPU-compatible, differentiable PM code [16]. With PMWD, we generate simulations with  $128^3$  and  $256^3$  particles with the same cosmology, box size, and Zel'dovich initial condition parameters as the HADES simulations. Although certainly not as accurate as the HADES simulations, PMWD is still able to simulate the nonlinear effects of gravity well in just a matter of seconds.

### 2.3 BAO Reconstruction

Acoustic waves in the early universe imprinted a characteristic length scale in the matter distribution, observable in the microwave background and in galaxy clustering [55–57]. In large spectroscopic galaxy surveys, the BAO scale manifests as a peak in the two-point correlation function of various matter tracers at around  $r \sim 100 \text{ h}^{-1} \text{ Mpc}$  [58–62]. This BAO peak is a powerful cosmic standard ruler that can be used to probe the expansion history of the Universe [63, 64].

However, nonlinear gravitational evolution broaden and smear out the BAO peak in the correlation function, which impacts the precision of its measurement [63, 65–68]. To mitigate this loss of precision, [27] first proposed using the Zel'dovich approximation to approximate the displacement field and to undo the effects of gravity, thereby sharpening the BAO peak. This method, referred to as standard BAO reconstruction, has allowed for more precise measurements of the BAO distance scale in observations [69–72].

Standard reconstruction is fundamentally different from LPT, including at first-order (Zel'dovich approximation), as described in Section 2.1, because the source potential is provided by the evolved density field instead of the initial density field. In its most basic form, standard reconstruction estimates the displacement field using

$$\nabla_{\mathbf{x}} \cdot \Psi^{(1)}(\mathbf{x}) = -\delta(\mathbf{x}), \quad (2.13)$$

where all functions are computed with respect to the final observed galaxy positions  $\mathbf{x}$ , then shifts all galaxies by  $-\Psi^{(1)}(\mathbf{x})$ . Later reconstruction methods add iterative processes or higher order LPT, [28, 31], and while these methods work well at linear scales, in general, they are not fully correct applications of the physics of LPT, which relates the displacement field to the initial density field [30].

Another class of reconstruction algorithms relies on forward modeling of the initial conditions to the evolved, nonlinear field [12, 38–40]. These Bayesian approaches typically sample

---

<sup>2</sup><https://franciscovillaescusa.github.io/hades.html>

<sup>3</sup><https://github.com/eelregit/pmw>

sets of initial density fields and evolve them forward to explore which initial conditions are consistent with the observed final density field. Since the forward model mapping the initial field to the final field must be called many times, it should be computationally cheap and, in general, differentiable. Thus, most of these methods rely on some form of LPT or PM models. However, LPT tends to exhibit poor accuracy at small scales, and PM models require sufficiently small grid sizes and time steps to match  $N$ -body codes, which can also become computationally expensive. A potential alternative is with a CNN, which, once trained, can map from an initial to final field for very little computational cost.

## 2.4 Optimal Transport Displacement Field

Before discussing CNNs and our proposed method in more detail, we will briefly describe a different type of reconstruction algorithm that uses the mathematical framework of optimal transport (OT) to estimate the displacement field from the observed galaxy distribution. The OT problem defines a metric, the Wasserstein distance, as the minimum cost to transport mass between two probability distributions [73–76]. In the context of large-scale structure, OT can provide a framework to exactly reconstruct the displacement field given the final positions with only the assumptions that the displacement field is the gradient of a scalar potential and that the initial conditions are uniform [77, 78]. As discussed in Sections 2.1 and 2.2, the first assumption is explicitly true in first- and second-order LPT, and most models of the early Universe suggest the second assumption is true as well, hence why cosmological simulations generally start with a homogeneous initial density field and nonlinear evolution driven by the initial velocity field.

Although the true nonlinear displacement field is not sourced by a single scalar potential, the displacement field estimated by the OT algorithm (henceforth referred to as the OT displacement field, or  $\Psi^{(\text{OT})}(\mathbf{q}, t)$ ) is well-correlated with the true  $\Psi(\mathbf{q}, t)$  [46]. As with standard reconstruction, OT reconstruction starts from a set of evolved positions  $\mathbf{x}$  (e.g., observed galaxy positions or the final particle positions of an  $N$ -body simulation) and returns a set of corresponding initial positions  $\mathbf{q}_{\text{OT}}(\mathbf{x})$ . The algorithm is deterministic and cosmology-independent, yielding a unique solution for any set of  $\mathbf{x}$ . Recent versions of the OT reconstruction algorithm, the so-called semi-discrete OT methods, demonstrate remarkable increases in speed, from  $\mathcal{O}(N^3)$  to  $\mathcal{O}(N \log N)$ , allowing for practical application to BAO reconstruction [32, 33, 79].

It is important to note that, necessarily,  $\Psi^{(\text{OT})} \neq \Psi^{(1)}$  and  $\Psi^{(\text{OT})} \neq \Psi^{(2)}$ , even though all three are sourced by a scalar field. If we refer to the scalar field from which  $\Psi^{(\text{OT})}$  is sourced as the OT potential,  $\phi^{(\text{OT})}(\mathbf{q})$ , then, by definition,  $\mathbf{x} = \mathbf{q}_{\text{OT}}(\mathbf{x}) + \nabla\phi^{(\text{OT})}$ . However, moving the particles from their initial positions by  $\nabla(-D_1(t)\phi^{(1)}(\mathbf{q}))$  or  $\nabla(-D_1(t)\phi^{(1)}(\mathbf{q}) + D_2(t)\phi^{(2)}(\mathbf{q}))$  would put them at their Zel'dovich-evolved or second-order LPT-evolved positions, not their exact nonlinear positions  $\mathbf{x}$ . Importantly, this means that OT reconstruction does not explicitly attempt to reconstruct  $\delta(\mathbf{q}) = \nabla^2\phi^{(1)}(\mathbf{q})$ , even though a better estimate of  $\delta(\mathbf{q})$  is typically the goal of BAO reconstruction. However,  $\nabla^2\phi^{(\text{OT})}(\mathbf{q})$  is generally well correlated with  $\delta(\mathbf{q})$ , at least on BAO scales [46].

Regardless, it is still interesting to consider how well the OT displacement field can be estimated from the initial conditions. It can be directly computed from an observed galaxy distribution, so a cheap forward model for  $\Psi^{(\text{OT})}(\mathbf{q}, t)$  can enable better reconstruction and field-level inference.

## 2.5 Convolutional Neural Networks

With the advent of powerful computational resources alongside a large influx of data in all fields, deep learning has recently become a common tool for learning complex functions from data, leading to models that often exceed human performance and traditional algorithms [80].

The basic model of deep learning is the neural network, which consists of layers that sequentially perform some mathematical transformation on data. In a fully-connected neural network, the output of each layer is

$$\mathbf{y}_{i+1} = s(\mathbf{W}_i \mathbf{y}_i + \mathbf{b}_i), \quad (2.14)$$

where  $\mathbf{y}_i$  is the  $n_i$ -dimensional input vector of layer  $i$  (and  $\mathbf{y}_0$  is the input data to the entire model),  $\mathbf{W}_i$  is an  $n_i \times n_{i-1}$  weight matrix,  $\mathbf{b}_i$  is an  $n_i$ -dimensional vector of additive biases, and  $s$  is a nonlinear activation function. The activation function is essential to the performance of the neural network; without it, the composition of layers would ultimately result in a linear operator. By training the model on pairs of input and output data, with the goal of minimizing a certain cost function, the weight and bias parameters of each layer can be iteratively updated, typically through gradient backpropagation [81], so that the neural network “learns” to transform input data into desired output data [37].

For high-dimensional, complex data, a fully-connected neural network will require many layers, each with many features. Training such a model is generally intractable. For data on a grid, such as images, CNNs resolve this issue with convolutional layers which hierarchically extract local features rather than treating each pixel as an independent feature [82]. The input to a convolutional layer,  $\mathbf{y}_i$ , consists of  $n_i$ -dimensional vectors on a  $d$ -dimensional grid, so that  $\mathbf{y}_i$  has dimension  $(N_1, \dots, N_d, n_i)$ , with  $N_i$  denoting the lengths of the each grid axis. For example, a displacement vector field on a  $256^3$  grid would have dimension  $(256, 256, 256, 3)$  while the divergence of such a field would have dimension  $(256, 256, 256, 1)$ . For intermediate layers, the vector dimension  $n_i$  is typically referred to as the number of features or filters of the layer. Then, the output of a convolutional layer is

$$\mathbf{y}_i^{(j)} = s \left( \sum_k \mathbf{w}_i^{(k,j)} * \mathbf{y}_{i-1}^{(j)} + \mathbf{b}_i^{(j)} \right), \quad (2.15)$$

where  $j, k$  are vector indices ranging from 1 to  $n_i$  or  $n_{i-1}$ , respectively,  $\mathbf{w}_i^{(k,j)}$  and  $\mathbf{b}_i^{(j)}$  are  $d$ -dimensional grids of kernel weight and bias parameters, respectively, and  $*$  represents a cross-correlation or convolution operator [37].

CNNs also commonly include pooling layers, which reduce spatial dimensions and thus computational complexity, as well as fully-connected layers at the end [83]. However, for our application, in which the input and output data have the same spatial dimension, such layers are not implemented.

The extent of connectivity between pixels in the input and output grid to the CNN is referred to as the receptive field. The receptive field is determined by the number of convolutional layers as well as the kernel size, which is the length of each dimension of  $\mathbf{w}_i^{(k,j)}$ . A typical kernel size is small relative to the input or output data so that the receptive field is also small. In other words, each grid point of the output data is only affected by nearby grid points of the input data [37]. For our use case, this is beneficial, because LPT already performs well at large scales, so we only hope to improve it at smaller scales.

## 2.6 Previous Literature

[84] provides a broad review of deep learning in cosmology and galaxy surveys, but their section on large-scale structure is brief. Moreover, given the rapid development of the field, many recent results are not included. Thus, in this section, we review some of the recent literature in both modeling of the displacement field and applications of deep learning to large-scale structure formation.

An early application of neural networks to reconstruction is presented by [85], who propose a differentiable forward model from the linear density field to the halo mass field, which allows for maximum likelihood approaches to data fitting through gradient backpropagation. Halo fields are typically incompatible with gradient-based methods, because halos are often discretized and defined with non-differentiable methods. Their forward model is divided into two components: first, the linear density field is evolved to the nonlinear density field with a differentiable PM simulation, and then a neural network is trained to learn the mapping from the nonlinear density field to the halo mass field. They further break down the neural network into two models, one that predicts the halo positions and one that predicts the halo masses. Despite relatively simple neural network architectures, with two hidden layers in each fully-connected network, they are able to reconstruct the initial density field from the final halo mass field with high correlation out to  $k \sim 0.1 - 0.2 \text{ h}^{-1} \text{ Mpc}$ , and, when applied to BAO reconstruction, they find improvement over standard reconstruction as well.

[86] apply CNNs to a similar task, mapping the nonlinear dark matter density field output of  $N$ -body simulations to the spatial distribution of galaxies that are outputted by full hydrodynamic simulations, which incorporate hydrodynamic forces and astrophysical processes in addition to gravity, making them much more costly than  $N$ -body simulations. This task encounters a problem in that the galaxy distribution is extremely sparse, which makes training difficult. The solution to this problem is similar to the two-model neural network architecture of [85]: they first train a classifier CNN that predicts the presence or absence of galaxies in each output voxel (like the halo position predictor of [85]), then use that output to train a second CNN to predict the number of galaxies in each voxel (like the halo mass predictor of [85]). Importantly, the second CNN only outputs a galaxy number prediction for voxels that are predicted to contain at least one galaxy by the first CNN. They test multiple variations of CNN architectures and find improvements over standard methods of linking the galaxy distribution to the dark matter distribution.

[18] attempt to emulate the entire  $N$ -body technique with Generative Adversarial Networks (GANs) [87], which is a machine learning framework in which a generator neural network learns to generate new data sufficiently similar to the real training data to deceive a second discriminator neural network. In effect, their model attempts to quickly generate realistic evolved  $z = 0$  density fields. However, this is the only output of the model; there is, for example, no corresponding initial density field, making applications to reconstruction non-obvious. And, although the generated fields are visually indistinguishable from true  $N$ -body fields, statistics such as the power spectrum are not fully in agreement with real simulation data.

[21] implement a similar GAN model, but they use a more informed generator model. Whereas [18] generate evolved density fields from latent vectors, which means that they have no control over the input to the generator model, [21] provide a lognormal density field, which are cheap approximations to simulated density fields, as input. As a whole, their generator CNN augments the lognormal approximation to match the nonlinear density field more closely. They find that using a standard loss function, such as the mean squared error,

is insufficient, so they also add a discriminator model, turning it into a GAN framework. Ultimately, their GAN model is able to produce realistic nonlinear density fields with overall statistical properties that differ from the true values by less than ten percent at most scales. [22] expand this model to encode a cosmology and redshift dependence. In addition, they turn the discriminator model into a physics-informed one. Typically, the discriminator model of a GAN consists of a standard feature-based CNN that classifies generator outputs as either emulated or real. They add another path within the discriminator model that calculates the difference between the power spectrum of the input lognormal density field, which are the same as the power spectrum of the corresponding  $N$ -body simulation by construction, and the power spectrum of the emulated density field, thus penalizing the generator for not maintaining the power spectrum of its input.

The most relevant works to our proposed model are [17, 19, 88], who all attempt to model the  $N$ -body displacement field using CNNs. In particular, they use a more complex U-Net model [89], which typically consists of a contractive path of successive down-sampling convolutional layers that reduce the data dimensionality followed by an expansive path of up-sampling convolutional layers that increase the data dimensionality back to the original size, as well as concatenation procedures that adds the features from the contractive path to the expansive path at each resolution level. Such models result in many layers and thus possibly a huge number of model parameters. [17] uses such a neural network to learn the displacement field output of a PM model with  $32^3$  particles in a  $L = 128 \text{ h}^{-1} \text{ Mpc}$  box, with the Zel'dovich approximation displacement field as the input. Their model, which contains a total of 15 convolutional layers, quite accurately predicts the nonlinear PM output at most scales. [88] expands this idea to learn the displacement field outputs of full  $N$ -body simulations with  $512^3$  particles in an  $L = 1000 \text{ h}^{-1} \text{ Mpc}$  box from the Quijote simulation suite [90], again with the Zel'dovich displacement field as their input. In comparison to a benchmark code, COLA (COmoving Lagrangian Acceleration) [91], which, like PM codes, is a fast  $N$ -body approximation, their model output is well-correlated to the true nonlinear displacement field to much smaller scales, with percent-level correlations to  $k \sim 1 \text{ h Mpc}^{-1}$ . [19] again builds off these two previous works to produce NECOLA (Neural Enhanced COLA), which takes as input the COLA displacement field and predicts the nonlinear  $N$ -body displacement field. Thus, while [17, 88] train their models to learn corrections to the Zel'dovich approximation, NECOLA learns corrections to COLA displacements to match the  $N$ -body displacements. Training on the Quijote simulations again shows that NECOLA outperforms even the model of [88].

[92] expand such work in a different direction by training a CNN (or more accurately, two CNNs) to learn the complete  $N$ -body phase space evolution, which includes not only the displacement field but also the velocity field. Upon training their model, they test it on physically idealized cases in which the solution can be solved for analytically or perturbatively, including a spherically symmetric density field or isolated and coupled plane waves. Given that such idealized test scenarios are far from the CNN training data, which consists entirely of random Gaussian initial density fields, the level of success they observe with their model at all scales indicates that the CNN is encoding some amount of physics. An important feature of their CNN architecture is a global residual operation, by which the output of the CNN is an estimate of the difference between the  $N$ -body displacement field and the Zel'dovich displacement field. Since CNNs are local models, doing so allows the output to naturally achieve better performance at large scales, where the Zel'dovich approximation is accurate on large scales. [20] add cosmology dependence through additional model parameters that

encode  $\Omega_m$ , allowing the model to effectively interpolate between a range of cosmological parameters. In a similar manner, [93] extend the model to include redshift dependence with new model parameters that encode  $D(z)$ , which allows for calculation of particle velocities as the time derivative of the displacements.

Rather than emulate the  $N$ -body simulation process at the field level, [94] train a CNN to predict individual evolved dark matter halo masses. Thus, the input to their model is the initial density field in a region around a dark matter particle, and the output is the mass of the dark matter halo containing the initial particle at  $z = 0$ . That the model performs well at predicting halo masses indicates that the CNN is successfully able to identify the features in the initial density field that impact halo evolution. They also perform tests in which they modify the input fields to remove anisotropic information by performing spherical averaging at each voxel, but do not find that the model predictions are significantly less precise. They are thus able to conclude that the anisotropies of the initial density field do not contain much information relevant to the final halo masses that is not already encoded in the isotropic features, indicating that improvements in models of halo collapse should be driven by the addition of dynamical information rather than just the initial conditions.

The above works all, at some level, hope to achieve fast and accurate forward models for large-scale structure formation. Machine learning has also been applied in the inverse direction, which more closely resembles standard reconstruction. For example, [35, 37, 95] all use CNNs to learn the initial density field directly from the evolved density field. [95] takes as input the raw, nonlinear fields, and their reconstructions achieve stronger correlation with the linear density field than standard reconstruction at some scales. [35, 37] both take as input the reconstructed density field, which results in much tighter correlation to the linear density field at all scales. They note that the improvement in using the reconstructed density field over the unreconstructed density field is due to the fact that, as mentioned in Section 2.5, CNNs can only learn local features within the receptive field, so reconstruction is necessary for large-scale information. [37] applies only standard reconstruction for the CNN input data, whereas [35] tests different reconstruction algorithms, although they find that the CNN output is similar regardless of which specific reconstruction is used.

While [35, 37, 95] focus on the dark matter density field, [96] consider a hybrid reconstruction technique for both dark matter halos and galaxies, which are both biased tracers of the density field. As in [35, 37], they first apply standard reconstruction, then train a CNN model to learn corrections at nonlinear scales. They implement a subgrid technique where they only train the model to learn on smaller subcubes of the full data. This helps mitigate GPU memory constraints without sacrificing performance, since standard reconstruction performs well at large scales and thus the model only needs to improve reconstruction at small scales. Additionally, as input to the CNN, they provide both the late-time tracer field and the reconstructed density field, in case there is information that only one of the two contains. Another unique feature of their model is the loss function, which is calculated as a mean squared error in Fourier space that is weighted such that modes in the range  $0.08 \leq k \leq 0.5 \text{ h Mpc}^{-1}$  are boosted to further prioritize nonlinear scales. Their model not only successfully extends the range of  $k$  to which correlation between the reconstructed and true density fields is strong, but also can be scaled to arbitrarily large volumes without retraining due to their subgrid method.

[36] similarly attempt the inverse problem using the displacement field. Interestingly, they use the model of [92], which, as described above, is a CNN that outputs the nonlinear displacement field from the initial density field, to produce their training data. Then, they use

the exact same model architecture of [92] to predict the linear Zel'dovich displacement field (from which the initial density field can be derived) from the nonlinear displacement field. In effect, they produce a new model to learn the inverse function of the original. However, their CNN also performs well on some test data from full  $N$ -body simulations, indicating that it is able to generalize from emulated training data.

Whereas [96] attempts to learn the initial density field from late-time observations of biased tracers, several works also use machine learning to better reconstruct the late-time density and velocity fields from such observations. [97] attempt to use simulated halo catalogs to reconstruct the full density and velocity fields with a CNN. With their reconstructed fields, they calculate the velocity-density coupling parameter  $\beta$ , which relates the peculiar velocity field to the density field. Although they find a systematic bias in the estimated  $\beta$  values, agreement with the fiducial value is generally within  $1\sigma$  indicates that the reconstructed density and velocity fields can provide useful constraints on cosmological parameters. In a similar manner, [98] trains a CNN model to reconstruct the velocity field in real space from the dark matter halo spatial distribution in redshift space, building off work by [99] to reconstruct the velocity field from the dark matter field and by [100] to reconstruct the peculiar velocity field from the dark matter halo distribution. Rather than dark matter halos, [101] uses both the spatial distribution and the radial peculiar velocities of galaxies to reconstruct the dark matter density field, with the goal of performing such reconstruction on sparse peculiar velocity surveys. On the other hand, [102] uses only the galaxy spatial distribution from dense redshift surveys to reconstruct both the dark matter density and velocity fields.

As noted above, our work is similar to the models of [17, 19, 88, 92] in that we attempt to predict the displacement field with deep learning. However, we address other open questions as well. Rather than augment an approximate  $N$ -body code like COLA, we augment theory only, in the form of LPT. We test whether adding second-order LPT as an input to the network improves performance. We also consider how simple we can make our CNN architecture while still achieving strong correlations at a wide range of scales. While [88] and [19] use 180 and 100 Quijote simulation boxes for training, respectively, [37] uses 15 AbacusSummit [103, 104] simulations and [35] uses just eight Quijote simulations. Thus, we also use a small number of training data for accelerated training and comparable results. To better accommodate both a simpler model and less training data, we also test whether it is easier for a CNN to learn to predict the divergence of the  $N$ -body displacement field, which is a scalar field.

### 3 LPT Tests

We first assess how well LPT can estimate the nonlinear displacement field.

#### 3.1 Data

We perform these tests on the HADES  $N$ -body simulations, described in Section 2.2. The suite we utilize contains 20 boxes of dimension  $L = 1000 \text{ h}^{-1} \text{ Mpc}$  with  $512^3$  particles. Each box starts with unique initial conditions generated from the same linear theory power spectrum (i.e., the same underlying cosmology). The data snapshots, containing all particle positions and velocities, are provided at redshifts of  $z = 99$  (the initial conditions),  $z = 3$ ,  $z = 2$ ,  $z = 1$ ,  $z = 0.5$ , and  $z = 0$ .

### 3.2 LPT Code

The nonlinear displacement of each particle, which we refer to as  $\Psi^{(\text{True})}$ , is calculated as

$$\Psi^{(\text{True})}(\mathbf{q}, z) = \mathbf{x}(\mathbf{q}, z) - \mathbf{q}, \quad (3.1)$$

where  $\mathbf{q}$  are the initial particle positions at  $z = 99$  and  $\mathbf{x}(\mathbf{q}, z)$  are the evolved positions at some later redshift. Periodic boundary conditions of the box are taken into account, so that the maximum displacement for a particle is  $500 \text{ h}^{-1} \text{ Mpc}$  in each component.

Computing both the nonlinear displacement field and the initial density field from the initial snapshot of particle positions requires placing the particles on a grid. Common mass assignment schemes are the Nearest Grid Point (NGP), Cloud-in-Cell (CIC), and Triangular Shaped Cloud (TSC) mass-assignment schemes. Employing these methods introduces artifacts into the density field power spectrum, which can theoretically be removed with correction factors. However, even with these corrections, we find that the power spectra of the initial density field from all of these mass assignment schemes are slightly inconsistent with the linear theory power spectrum, calculated with `CLASS`<sup>4</sup> [50]. We thus use the initial velocity information to better compute the initial density field.

We trace back the particles, moving in the negative velocity direction, until they each lie exactly on a grid point, so that the gridded displacement field can be easily calculated by assigning to each grid point the full nonlinear displacement of the respective particle. However, moving particles back onto an exact grid enforces a homogeneous density, so that the initial density field is lost. Instead, since the initial conditions snapshot includes the particle velocities, we calculate the gridded initial velocity field, then utilize that to get the initial density field.

Since HADES uses Zel'dovich initial conditions, combining Equation 2.11 and 2.3 gives the divergence of the initial velocity field as

$$\nabla \cdot \mathbf{v}_{\text{init}}(\mathbf{q}) = -(af_1 H)_{\text{init}} \delta(\mathbf{q}). \quad (3.2)$$

In Fourier space, which we move in and out of using the Fast Fourier Transform (FFT) and inverse FFT (IFFT), the modes of the initial velocity field are thus related to those of the initial density field as  $\tilde{\mathbf{v}}_{\text{init}}(\mathbf{k}) = \frac{i\mathbf{k}\tilde{\delta}(\mathbf{k})}{k^2}$ . Combining this with Equation 2.2 gives the Zel'dovich displacement at an evolved redshift  $z$  as

$$\tilde{\Psi}^{(1)}(\mathbf{k}, z) = \frac{D_1(z)\tilde{\mathbf{v}}(\mathbf{k})}{D_1(z=99)(af_1 H)_{\text{init}}}. \quad (3.3)$$

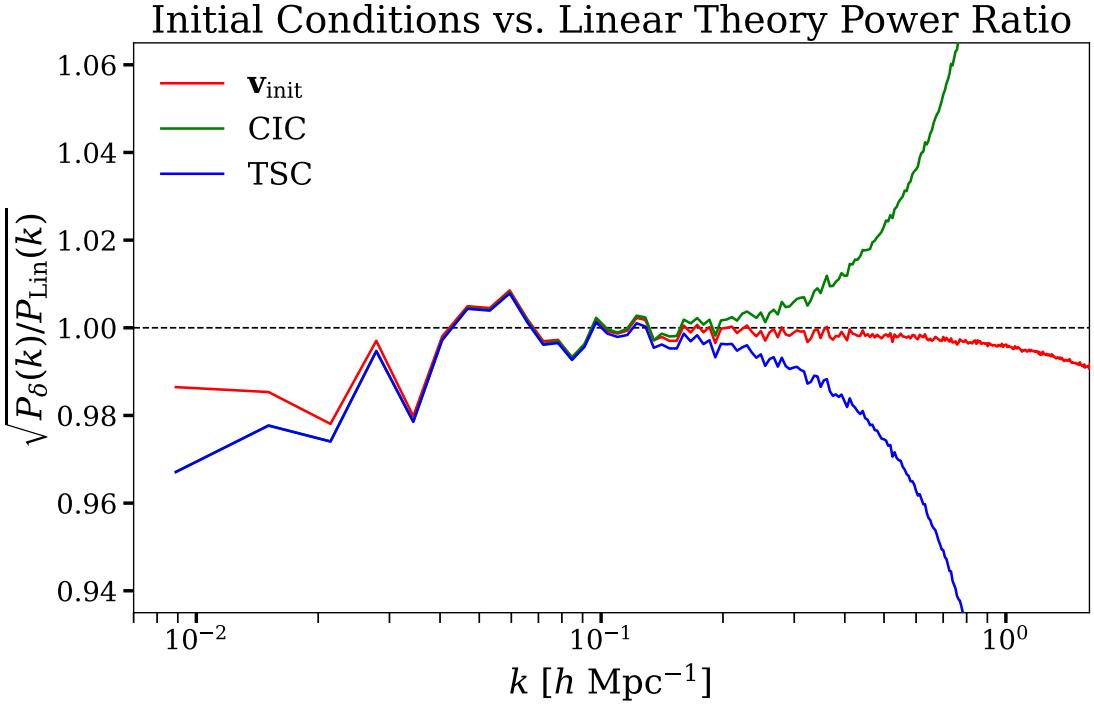
We calculate the Zel'dovich displacement and initial density fields from the gridded initial velocity field using the above equations.

In theory, the power spectrum of the initial density field calculated in this manner should match the linear theory power spectrum at a redshift of  $z = 99$ , particularly when averaging over many simulation boxes, which reduces cosmic noise. Figure 1 shows that there is indeed a good match across most scales, with a power spectrum ratio near unity, and that the numerical mass assignment schemes introduce artifacts at scales approaching the 1D Nyquist frequency,

$$k_N = \pi \frac{N_{\text{dim}}}{L_{\text{box}}}, \quad (3.4)$$

---

<sup>4</sup><http://class-code.net/>



**Figure 1.** Square root of the ratio of the initial density field power spectrum, calculated from the gridded velocity field (red) or a numerical mass assignment scheme (CIC - green, TSC - blue), to the linear theory power spectrum. The shaded gray area represents  $|\sqrt{P_\delta/P_{\text{Lin}}} - 1| \leq 0.01$ . The maximum frequency is the Nyquist frequency,  $k_N = 1.61 \text{ h Mpc}^{-1}$ . Note that using an NGP mass assignment would result in a completely homogeneous density field.

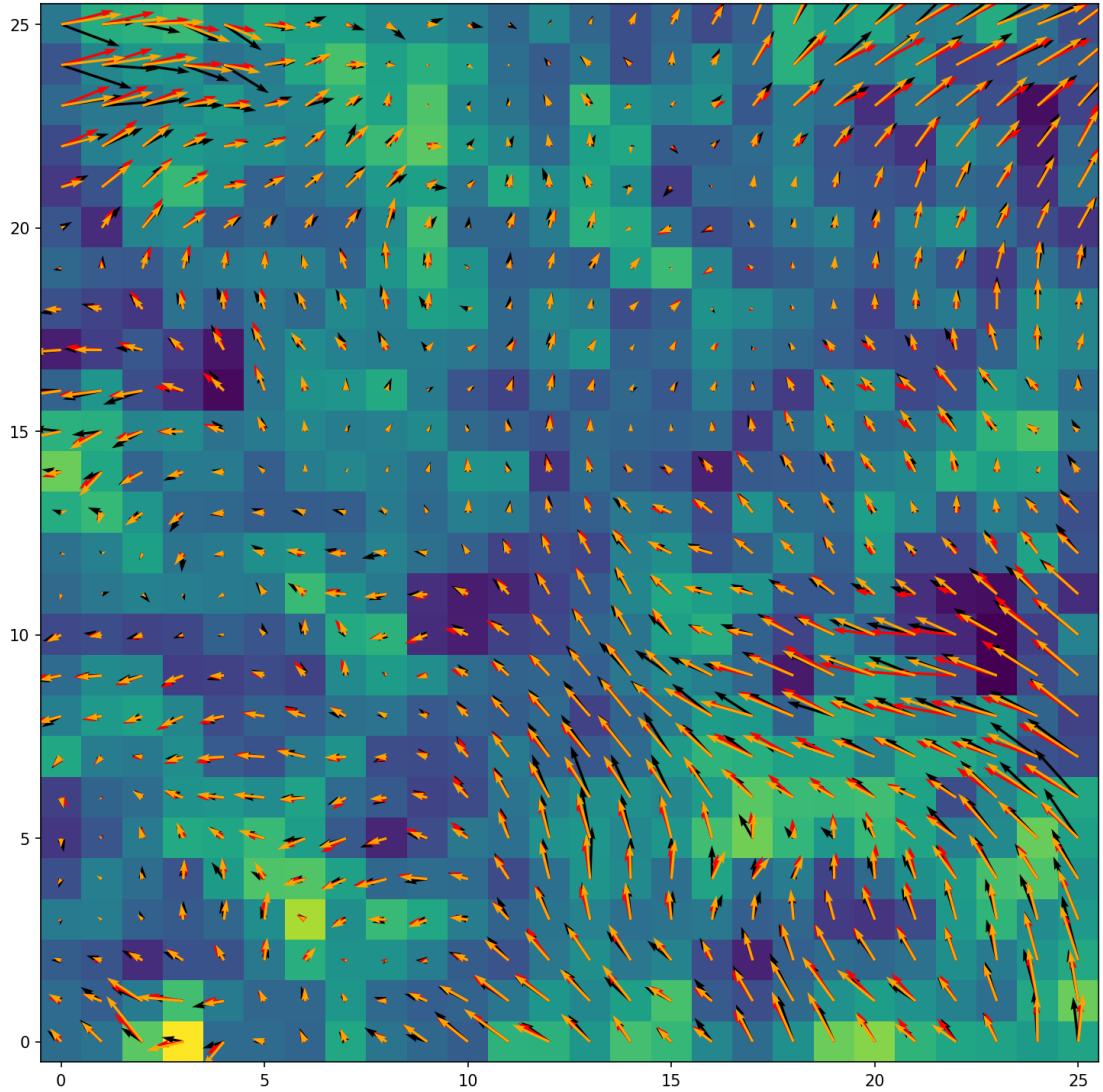
where  $N_{\text{dim}}$  is the grid dimension and  $L_{\text{box}}$  is the physical box length; for the HADES simulations, where  $N_{\text{dim}} = 512$  and  $L_{\text{box}} = 1000 \text{ h}^{-1} \text{ Mpc}$ , the Nyquist frequency is about  $k_N = 1.61 \text{ h Mpc}^{-1}$ . However, even for the gridded initial velocity field method, there is a strange artifact at  $k \gtrsim 1 \text{ h Mpc}^{-1}$ , where the ratio begins to fall below unity. This is discussed in more detail in Appendix A.

Solving Equation 2.3 in Fourier space gives  $\phi^{(1)}(\mathbf{k})$ , from which we can recursively solve Equations 2.4, 2.8, and 2.9 for the second- and third-order scalar potentials, then the second-order LPT displacement field and the curl-free component of the third-order field.

Lastly, we compute the OT displacement field. The OT displacement field can be calculated directly from observations, and it has the advantage of being sourced by a scalar potential, so it is also beneficial to develop a forward model for its cheap estimation from the initial conditions. For each HADES box, the semidiscrete OT reconstruction algorithm produces a set of estimated initial particle positions,  $\mathbf{q}_{\text{OT}}$ , from the output  $z = 0$  snapshot of final particle positions,  $\mathbf{x}$ . The algorithm has been previously run and its output saved for the 20 HADES boxes that we consider [79]. We calculate the OT displacement of each particle as

$$\Psi^{(\text{OT})}(\mathbf{q}) = \mathbf{x} - \mathbf{q}_{\text{OT}}(\mathbf{x}), \quad (3.5)$$

again taking into account periodic boundary conditions. Unlike for the  $\Psi^{(\text{True})}$ , for which we can take the initial positions  $\mathbf{q}$  to lie on a grid, neither  $\mathbf{x}$  nor  $\mathbf{q}_{\text{OT}}$  can be placed exactly on a



**Figure 2.** Corner section in the  $z = 0$  plane of true displacement field (black arrows), first-order LPT displacement field (red arrows), and second-order LPT displacement field (orange arrows). The background plot shows the initial density field,  $\delta(\mathbf{q})$ .

grid, so we must utilize a numerical mass assignment scheme. To keep consistency with the other displacement fields, we grid with respect to  $\mathbf{q}_{\text{OT}}$ . However, we find boxes where NGP, CIC, and TSC mass assignment schemes all lead to some grid points with zero displacement (i.e., there is no particle close enough to those grid point), which is problematic. We thus use a piecewise cubic spline (PCS) for mass assignment, which works well.

For each of the above displacement fields, we also take a divergence to get a scalar field,

and we refer to these as

$$\Theta^{(\text{True})}(\mathbf{q}, t) \equiv \nabla \cdot \Psi^{(\text{True})}(\mathbf{q}, t), \quad (3.6)$$

$$\Theta^{(1)}(\mathbf{q}, t) \equiv \nabla \cdot \Psi^{(1)}(\mathbf{q}, t) = -D_1(t)\delta(\mathbf{q}), \quad (3.7)$$

$$\Theta^{(2)}(\mathbf{q}, t) \equiv \nabla \cdot \Psi^{(2)}(\mathbf{q}, t), \quad (3.8)$$

$$\Theta^{(3)}(\mathbf{q}, t) \equiv \nabla \cdot \Psi^{(3)}(\mathbf{q}, t), \quad (3.9)$$

$$\Theta^{(\text{OT})}(\mathbf{q}) \equiv \nabla \cdot \Psi^{(\text{OT})}(\mathbf{q}). \quad (3.10)$$

For each simulation box, we calculate the displacement fields (except the OT displacement field) up to both  $z = 0$  and  $z = 1$ , the latter of which may be more relevant for observational surveys. We therefore calculate nine displacement vector fields plus the divergences of all of these fields per box. We average all results over the 20 HADES simulation boxes.

We note that the second- and third-order LPT terms contain quadratic or higher nonlinearities, and so multiplying numerically, as is required to solve the relevant Poisson equations, will introduce aliasing [42]. This can be mitigated by zero-padding and performing convolutions in Fourier space; however, this dramatically increases the memory required for each field. Furthermore, the effect of aliasing on the displacement field is small, especially at most relevant scales. Thus, we do not perform any de-aliasing.

In addition to computing the power spectra for each field, we calculate the normalized correlation coefficients between various fields, which is given by

$$r_{AB}(k) = \frac{\langle \tilde{A}(\mathbf{k})\tilde{B}(\mathbf{k})^* \rangle}{\sqrt{P_A(k)P_B(k)}}. \quad (3.11)$$

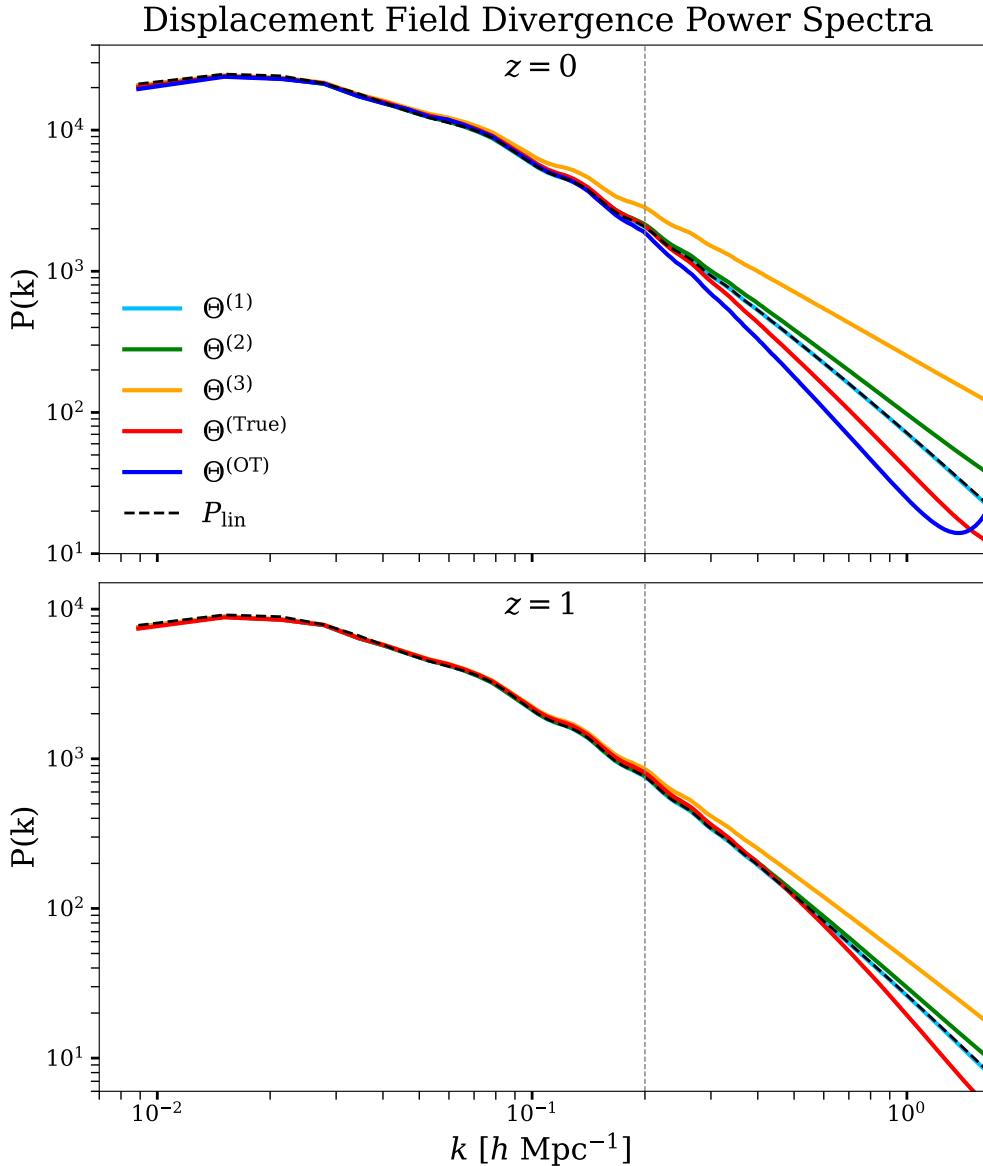
While the ratio of power spectra primarily assesses correlation in amplitude at each spatial frequency, the normalized correlation coefficients instead gives correlation in phase.

### 3.3 Results

Figure 2 compares a small 2D slice of the true displacement field to the Zel'dovich and second-order LPT displacement fields. Though the arrows do not show the third dimension, there appears to be a slight qualitative improvement in estimating the true displacement field when increasing from first- to second-order LPT.

Figure 3 compares the power spectra of the divergences of the various displacement fields at  $z = 0$  and  $z = 1$ , as well as the linear theory power spectrum, calculated with CLASS. We see that all the power spectra agree at the largest of scales, but each of the estimator power spectra diverges, at varying speeds, from the ground truth as we approach smaller scales. Also, the power spectra are all closer together, even at small scales, for  $z = 1$ , which is expected since the particles have had less time to evolve through gravity in this case.

A clearer comparison is shown in Figures 4 and 5, which display the square root of the divergence power spectrum ratios between pairs of displacement fields or the linear theory power spectrum. In the panel depicting the ratio between the estimated displacement fields and the true displacement fields, we can see that the Zel'dovich and second-order LPT power spectra agree in amplitude with the true power spectrum to within a few percent for  $k \lesssim 0.2 \text{ h Mpc}^{-1}$  at  $z = 0$ , and for slightly smaller scales at  $z = 1$ . Interestingly, it appears that the correlation in amplitude tends to worsen upon increasing the order of LPT. For both the  $z = 0$  and  $z = 1$  case, there is also an unexpected slope to the ratio at large scales, where the ratio does not remain flat around unity but rather starts above unity at the lowest  $k$

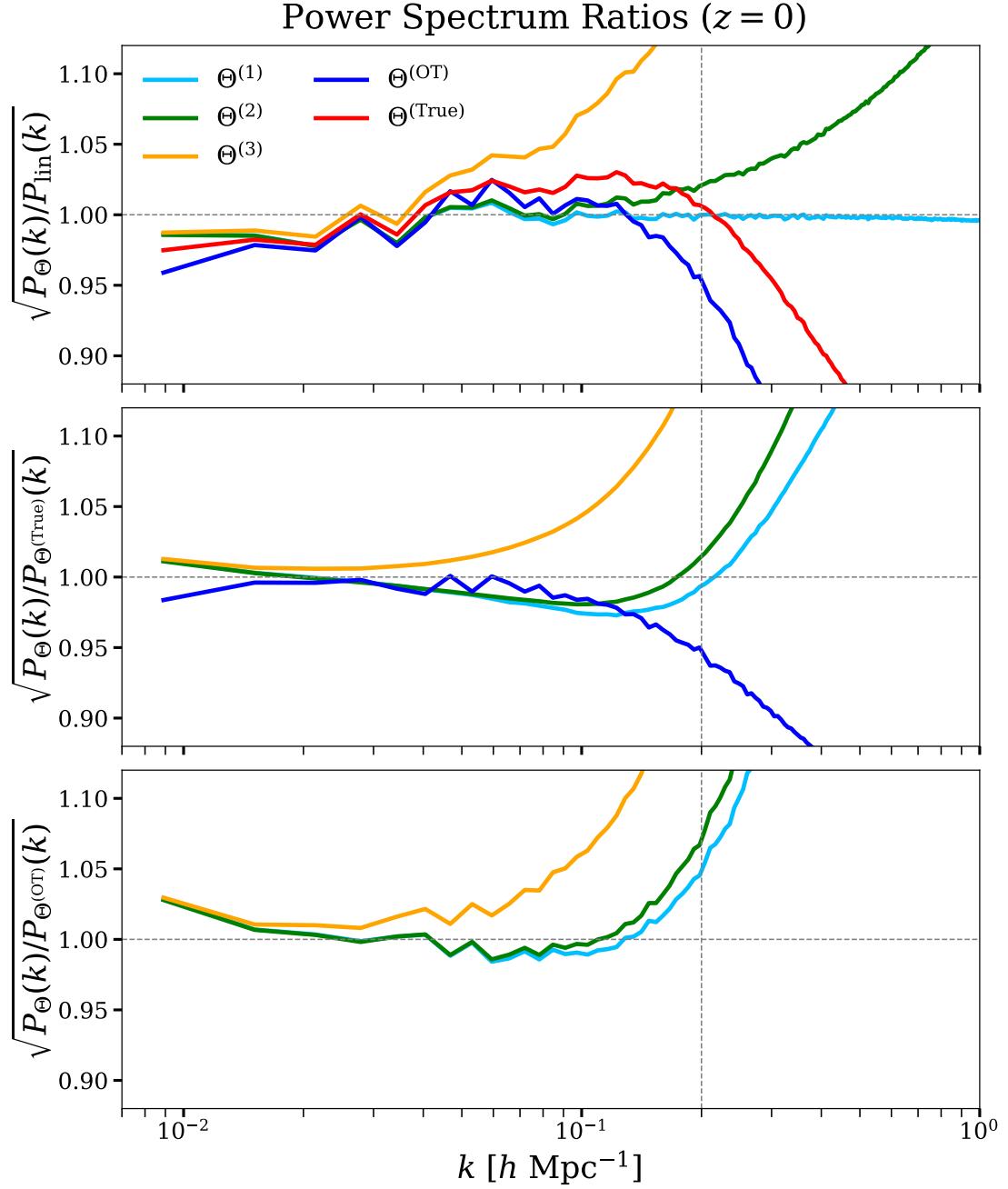


**Figure 3.** Power spectra of the divergences of the true, Zel’dovich, second- and third-order LPT, and OT displacement fields for both  $z = 0$  (top) and  $z = 1$  (bottom). The linear theory power spectrum is also shown for comparison. The vertical dashed line marks  $k = 0.2 \text{ h Mpc}^{-1}$ . Note that the OT displacement field is only calculated for the  $z = 0$  evolved field.

and decreases as  $k$  increases until the nonlinear scales are reached, at which point it quickly diverges upward. This feature is most prominent in the Zel’dovich and second-order LPT displacement fields.

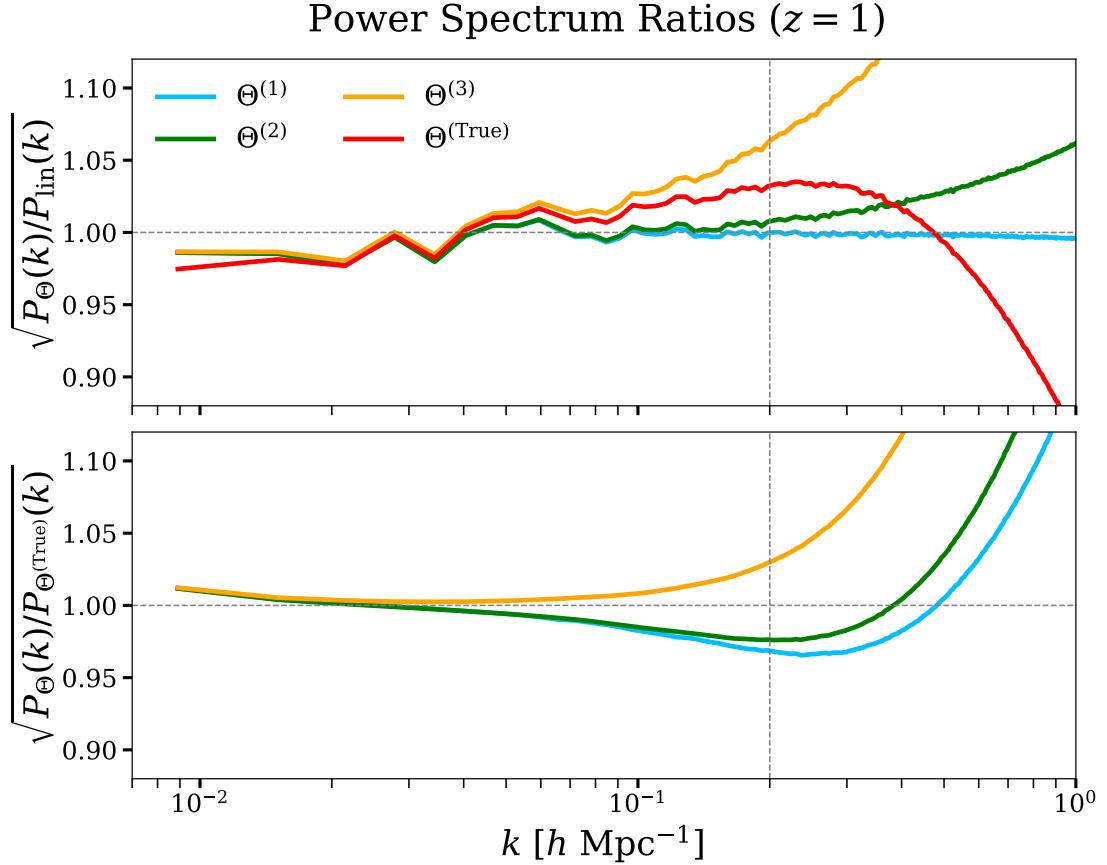
Comparing the middle and bottom panels of Figure 4 reveals that the LPT fields are worse at estimating the OT field than the true field, with the divergence in power ratio occurring at larger scales.

Figure 6 shows the normalized correlation coefficients between pairs of displacement field divergences. We again see that the correlation is strong at large scales, then falls off at small



**Figure 4.** Square root of the ratio of the different displacement field divergence power spectra to the linear theory power spectra (top),  $P_{\Theta(\text{True})}(k)$  (middle), and  $P_{\Theta(\text{OT})}(k)$  (bottom) for  $z = 0$ . For the ratios with  $P_{\Theta(\text{True})}(k)$  and  $P_{\Theta(\text{OT})}(k)$ , we take the ratio of the averages rather than the average of the ratios. The vertical dashed line marks  $k = 0.2 \text{ h Mpc}^{-1}$  and the horizontal dashed line marks a ratio of unity. The shaded gray area represents  $\left| \sqrt{P_{\Theta}} / P_{\text{ref}} - 1 \right| \leq 0.01$ .

scales. Here, we can clearly see improvement as go from the Zel'dovich approximation to second-order LPT, where the range of  $k$  at which the correlation is near unity extends much further into the range of nonlinear scales for both  $z = 0$  and  $z = 1$ . This is also true as we



**Figure 5.** Same as the top two panels of Figure 4, but for  $z = 1$ , excluding the OT displacement field.

go from second- to third-order LPT, but the marginal improvement is much smaller. We also see that the strongest correlation is between the true and OT displacement fields, although the correlation between the LPT and OT displacement fields is weaker than that between the LPT and true fields.

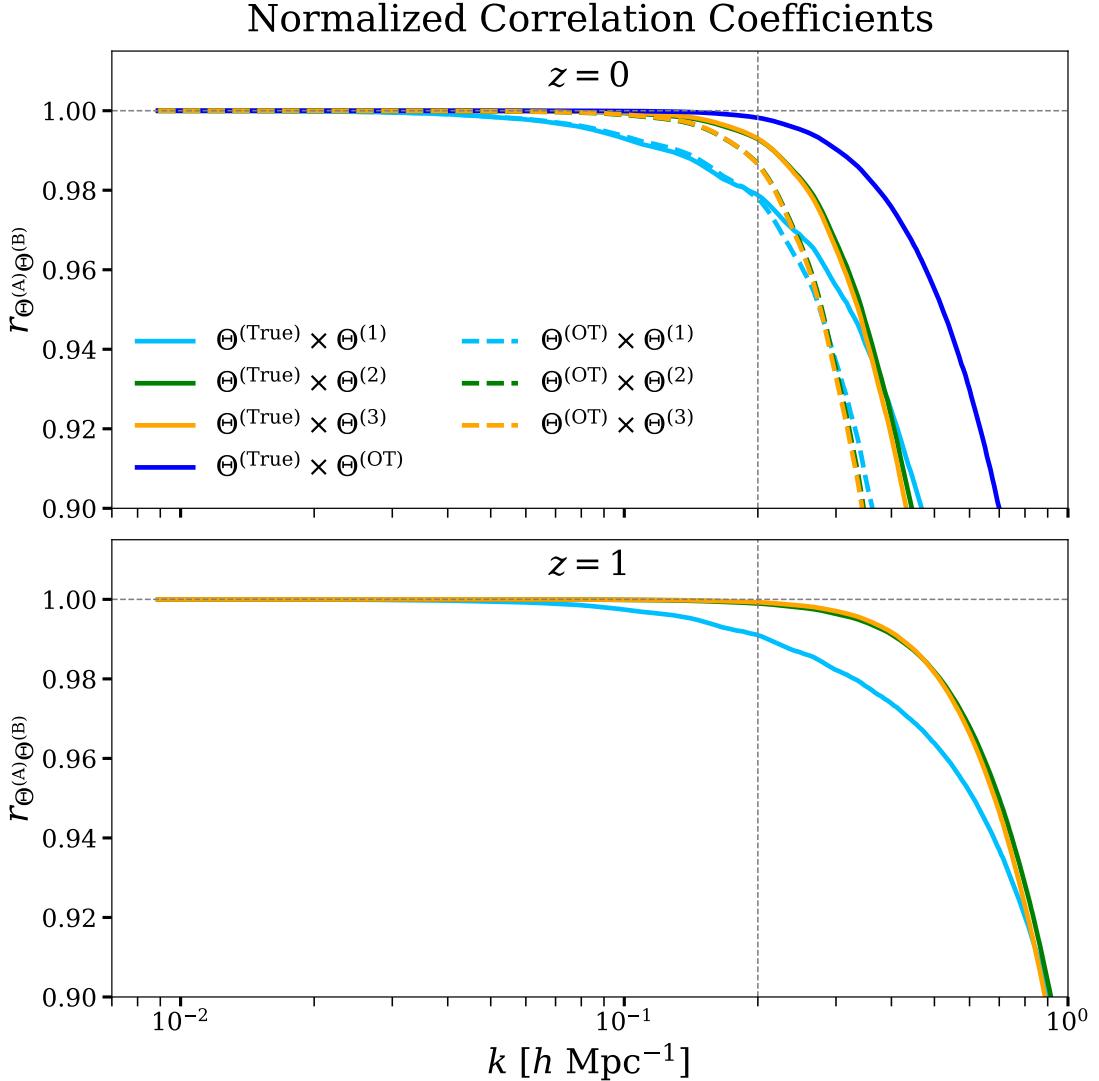
Taken as a whole, these results suggest that while LPT performs well at estimating the displacement field at low  $k$ , it breaks down at high  $k$ , and adding additional terms to the expansion may not improve convergence at these nonlinear scales. The question that naturally arises is then whether there are ways to improve the displacement field estimation, such as by augmenting LPT with machine learning.

## 4 CNN

In this section, we describe the CNN model that we train to better predict the true displacement field.

### 4.1 Data

The primary challenges of training such a deep learning model are GPU memory constraints. Training on a large number of particles requires dividing the simulation cubes into smaller



**Figure 6.** Normalized correlation coefficients between different displacement field divergences at  $z = 0$  (top) and  $z = 1$  (bottom). At  $z = 0$ , the dashed lines denote correlations between LPT displacement fields and the OT displacement field, whereas solid lines denote correlations between the nonlinear displacement field and either LPT or the OT displacement field. At  $z = 1$ , we do not compute an OT displacement field. The vertical dashed line marks  $k = 0.2 \text{ h Mpc}^{-1}$  and the horizontal dashed line marks a value of unity, which is the maximum of the normalized correlation coefficient.

subcubes. To demonstrate that we can learn the displacement field with a CNN in principle, we start with smaller simulations, then show that our approach can be easily scaled up.

We generate simulations with PMWD, described in Section 2.2, with both  $128^3$  and  $256^3$  particles. We keep the background cosmology the same as in the HADES simulations, fix the box size at  $L = 1000 \text{ h}^{-1} \text{ Mpc}$ , and use Zel'dovich initial conditions. We only consider simulations evolving from  $z = 99$  to  $z = 0$ .

The model we implement, described below, is flexible in what it takes as input and output. Thus, for each simulation we save the initial density field, the first- and second-order

LPT potentials, the true displacement field, and the first- and second-order LPT displacement fields. These are all generally calculated in the same manner as with the HADES simulations. To test that our code to generate the LPT displacement fields works properly, we also generate Zel'dovich and second-order LPT “initial conditions” at  $z = 0$  with PMWD, and find that they agree.

## 4.2 Model

At its core, our model takes as input an  $N^3 \times n$  grid, where  $N$  is the number of grid points along each spatial dimension and  $n$  is the number of scalar fields in the input.

### 4.2.1 $128^3$ Grid

For the smaller case of the simulations with  $128^3$  particles, we are able to perform training on the full cubes. The input grid passes through nine convolutional layers, with each intermediate layer having 32 filters. All but the final convolutional layer are followed with a hyperbolic tangent activation function, and all layers have a kernel size of 5. A convolution with a kernel size of 5 reduces the size of the grid by 4 in each spatial dimension, so the output has dimension  $(N - 36)^3 \times n$ . To reconstruct the full output, we expand the input field with periodic boundary conditions. Thus, the input has dimension  $164^3 \times n$  and the output has dimension  $128^3 \times n$ .

We focus on two possibilities, both with  $n = 3$ . In the first model, we input the second-order LPT displacement field, so that the three input scalar fields are the three components of the vector field. We refer to this as the vector model. In the second model, the three scalar field inputs are the initial density field and the first- and second-order LPT potentials. We refer to this as the scalar model.

For the vector model, we train to learn the true vector displacement field, so that the output of the model is an  $128^3 \times 3$  grid. For the scalar model, we train to learn the divergence of the true displacement field, so that the output of the model is an  $128^3 \times 1$  grid.

There is ultimately much freedom in what exactly the input fields are. We naturally choose to map vector fields to vector fields and scalar fields to scalar fields. In both of our models, we choose to provide fields derived from second-order LPT because we want our model to be physically informed. We could just as easily input, for example, the Zel'dovich displacements into the vector model, or replace the second-order LPT potential with its Laplacian in the scalar model. Exploring which physically motivated inputs enable the model to perform best is a possible avenue of future investigation.

In the final convolutional layer, we concatenate the three scalar input fields to the output of the previous layer, so that the number of filters becomes 35. This concatenation, inspired by the skip connections of U-Net models, potentially helps retain some of the large-scale features that LPT is able to predict well and that may be lost after multiple convolutional layers, which are local functions. After the previous eight convolutional layers, the grid spatial dimensions have been reduced from  $164^3$  to  $132^3$ , so we remove the appropriate boundary pixels of the input fields before concatenation.

The exact hyperparameters of our model, such as the kernel size and number of layers, are chosen heuristically; finer tuning may be required to achieve optimal performance. Our model architecture is summarized in Table 1.

Layer	Kernel Size	Num. Filters	Activation	Input
1	5	32	tanh	$164^3 \times 3$ $(\delta(\mathbf{q}), \phi^{(1)}(\mathbf{q}), \phi^{(2)}(\mathbf{q}))$ or $\Psi^{(2)}(\mathbf{q})$
2	5	32	tanh	$160^2 \times 32$
3	5	32	tanh	$156^2 \times 32$
:	:	:	:	:
8	5	32	tanh	$136^2 \times 32$
9	5	32	—	$132^2 \times 35$ Filters + Input
Output	—	—	—	$128^3 \times 1$ or $128^3 \times 3$ $\nabla \cdot \hat{\Psi}(\mathbf{q})$ or $\hat{\Psi}(\mathbf{q})$

**Table 1.** Summary of CNN architecture for simulations with  $128^3$  particles.

#### 4.2.2 Scaling to Larger Grids

We encounter GPU memory constraints even when attempting to model simulations with  $256^3$  particles. We thus use a subcube method to scale to these larger simulations, allowing us to keep the exact same model architecture as summarized in Table 1.

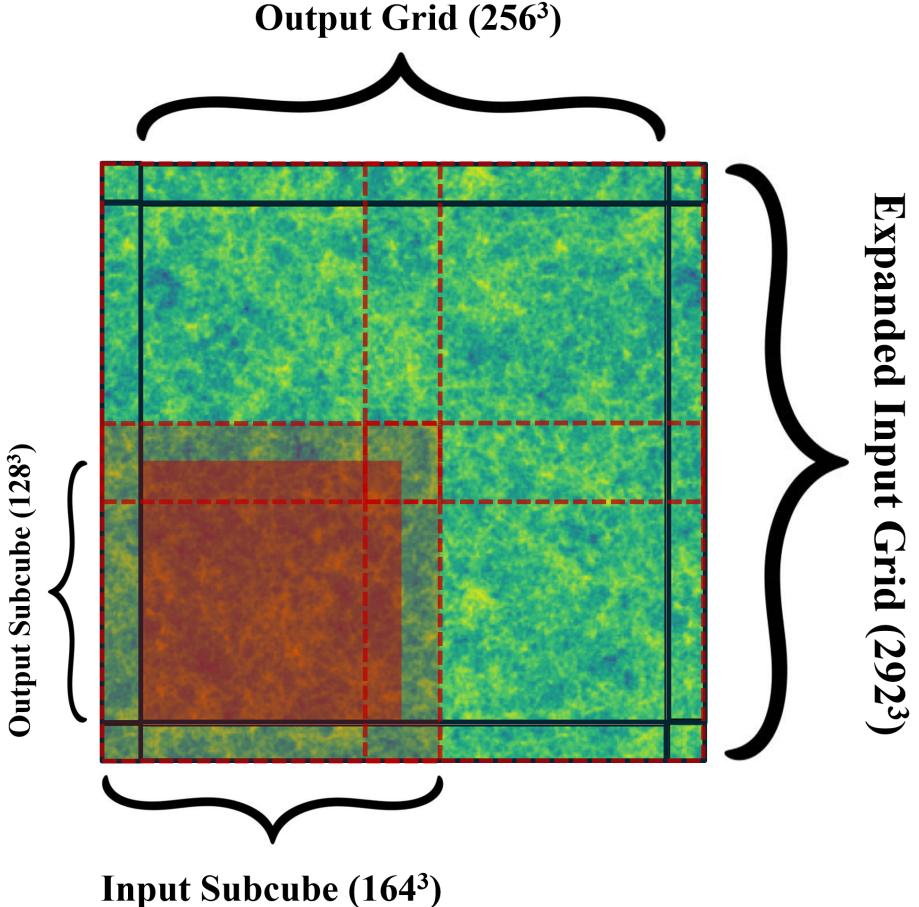
At each iteration of training, we select a subcube of shape  $164^3 \times 3$  from the input field and the corresponding subcube of shape  $128^3 \times n$  from the output field. In practice, to choose a random subcube while keeping the periodic boundary conditions of the full simulation, we first roll the input and output grids in each spatial dimension by a random integer value (possibly different in each dimension, but the same for the input and output grids). We then expand the input spatial grid with periodic boundary conditions, giving it shape  $292^3$ . We select the corner subcube of this grid with shape  $164^3$ . The corner subcube of the (non-expanded) output grid with shape  $128^3$  is the corresponding output subcube.

The model is trained only on these subcubes, which are produced on the fly at each iteration. Once the model is trained, we can use it to reconstruct the full output field with spatial shape  $256^3$  by reconstructing eight non-overlapping subcubes. To do so, we must construct the eight corresponding, overlapping subcubes of the expanded input fields. To do so, we again expand the input spatial grid to have shape  $292^3$ . We then select subcubes of this grid with shape  $164^3$ . There are eight such subcubes with that each share exactly one corner with the full grid. The overlap between such subcubes is at most 36 in each dimension. The model outputs of these eight subcubes each have spatial shape  $128^3$  and can be combined to reconstruct the estimate of the full output field. A visual schematic of this procedure in 2D is shown in Figure 7.

This procedure can, in principle, be easily generalized to larger simulations as well.

### 4.3 Training

We train our model in JAX [105] using the Flax [106] and Optax [107] libraries. Training involves learning the optimal convolutional kernel weights and bias parameters through gradient



**Figure 7.** Schematic of the procedure to reconstruct full  $256^3$  output grids from smaller subcubes, showing a 2D slice of the spatial dimensions. The input grid is expanded, with periodic boundary conditions, from  $256^3$  to  $292^3$ . It is divided into eight overlapping subcubes of size  $164^3$ , which are inputted into the CNN. Four of these are visible in the 2D slice as the dashed red squares, with the one at the bottom left corner shaded in light red. The outputs are eight non-overlapping subcubes, each of size  $128^3$ , which combine to create the full output grid. The one at the bottom left corner is shaded in dark red.

backpropagation on a loss function. We use the mean squared error (MSE) loss function,

$$\mathcal{L} = \frac{1}{N_{\text{sim}}} \sum_i \frac{1}{N_{\text{grid}}} \sum_q \left( \hat{f}_i(\mathbf{q}) - f_i(\mathbf{q}) \right)^2, \quad (4.1)$$

where  $\hat{f}(\mathbf{q})$  and  $f(\mathbf{q})$  are the model prediction and true values, respectively, of either  $\Psi(\mathbf{q})$  for the vector model or  $\nabla \cdot \Psi(\mathbf{q})$  for the scalar model;  $N_{\text{grid}}$  is the number of grid points in the output field, which are indexed by  $q$ ; and  $N_{\text{sim}}$  is the number of training simulations, indexed by  $i$ . In practice, we do not compute the loss over all simulations, but rather use a stochastic gradient descent (SGD) with momentum [108], which reduces training loss at each step by computing the gradient of the loss over a smaller random subset of the full training set. For the  $128^3$  grid, the smaller subset is one random training simulation; for larger grids, it is a random subcube of a random training simulation. Thus, in all cases, at each training step,

we load in a full simulation into CPU memory, but only load in a  $164^3 \times 3$  grid as input and a  $128^3 \times n$  grid as output into GPU memory.

This loss function is standard in many neural networks. However, [19, 88] both modify the loss function and instead use

$$\mathcal{L} = \ln \left( L_\delta L_\Psi^\lambda \right), \quad (4.2)$$

where  $L_\delta$  is the MSE loss on the density field,  $L_\Psi$  is the MSE loss on the displacement field, and  $\lambda$  is a weight parameter. Thus,  $L_\Psi$  is equivalent to our loss function for the vector model, whereas  $L_\delta$  is entirely different. Using such a loss function requires adding an extra field, the nonlinear density field at  $z = 0$ , into the output. Furthermore, this requires calculating the estimated nonlinear density field from the CNN estimate of the displacement field. For the scalar model, this is feasible but possibly expensive, since this would require calculating the displacement field from its divergence. Even for the vector model, this requires computing the evolved particle positions  $\mathbf{x}$  and calculating the density field with a mass assignment scheme. [19, 88] use the CIC scheme, which is differentiable. While implementing such a loss function may have a benefit, we choose to keep a simpler MSE loss function, which we find performs well.

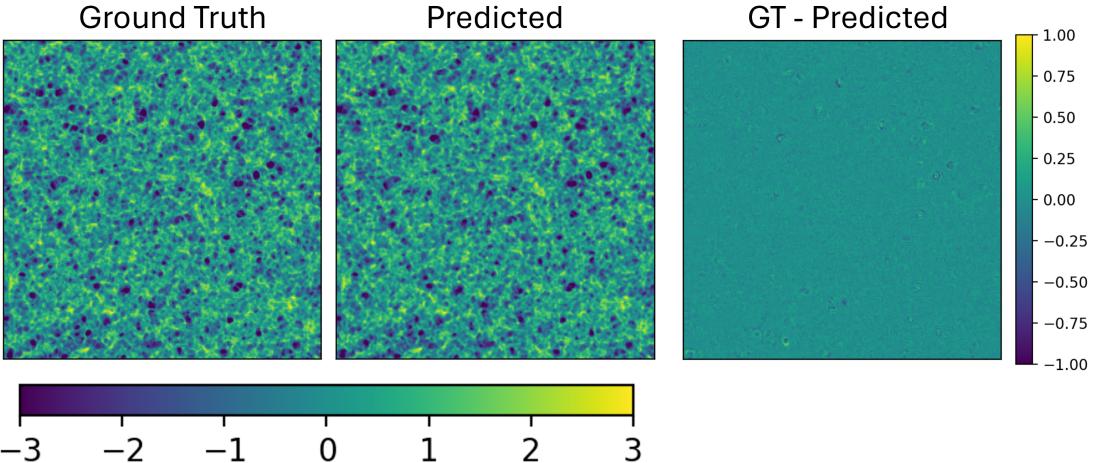
Before performing training, the training data set is normalized such that the mean value over all pixels in the training set is zero and the variance is unity. The normalization values are recorded so that validation and testing data can be normalized as well. We use a learning rate of 0.001 and a SGD momentum parameter of 0.9. For the simulations with  $128^3$  particles, we train both a vector and a scalar model (henceforth referred to as the  $128^3$  vector and scalar models, respectively). The simulations with  $256^3$  particles are used only to show that our model is scalable, so we train only a scalar model (the  $256^3$  scalar model). For all models, we train on two simulations. For the scalar models, we train for 10000 steps. For the vector model, we train for 5000 steps, although this is only to accelerate the training, and it may be preferable to train for more. To ensure that the model is not overfitting the training data, we check the loss on a validation simulation every 50 steps and find that it aligns with the training loss well. Each model was trained on a GPU, either a NVIDIA RTX A5000 or an NVIDIA A100, on the Yale Grace cluster for approximately three hours. Heuristically, it seems that any GPU with at least about 15 GB of VRAM is sufficient to train the model without encountering memory constraints.

#### 4.4 Results

The trained models are tested on 17 simulations that differ in the initial conditions from the training and validation simulations. For all models, we compare the divergences of the true and estimated displacement fields. For the scalar models, this is just the output. For the vector model, we could also compare the individual components of the displacement fields, but we choose to focus on just the divergence in this work.

Figure 8 shows a 2D slice of a randomly selected  $256^3$  scalar model output. Compared to the true displacement field divergence in the left panel, the model prediction is visually indistinguishable. The residuals, shown in the right panel at a narrower color scale, are nearly flat, which is promising.

More important than a visual inspection are the power spectra and their derived summary statistics. The following are all averaged over the test simulations. Figure 9 compares the power spectra of the divergences of the displacement field output from both the  $128^3$  scalar

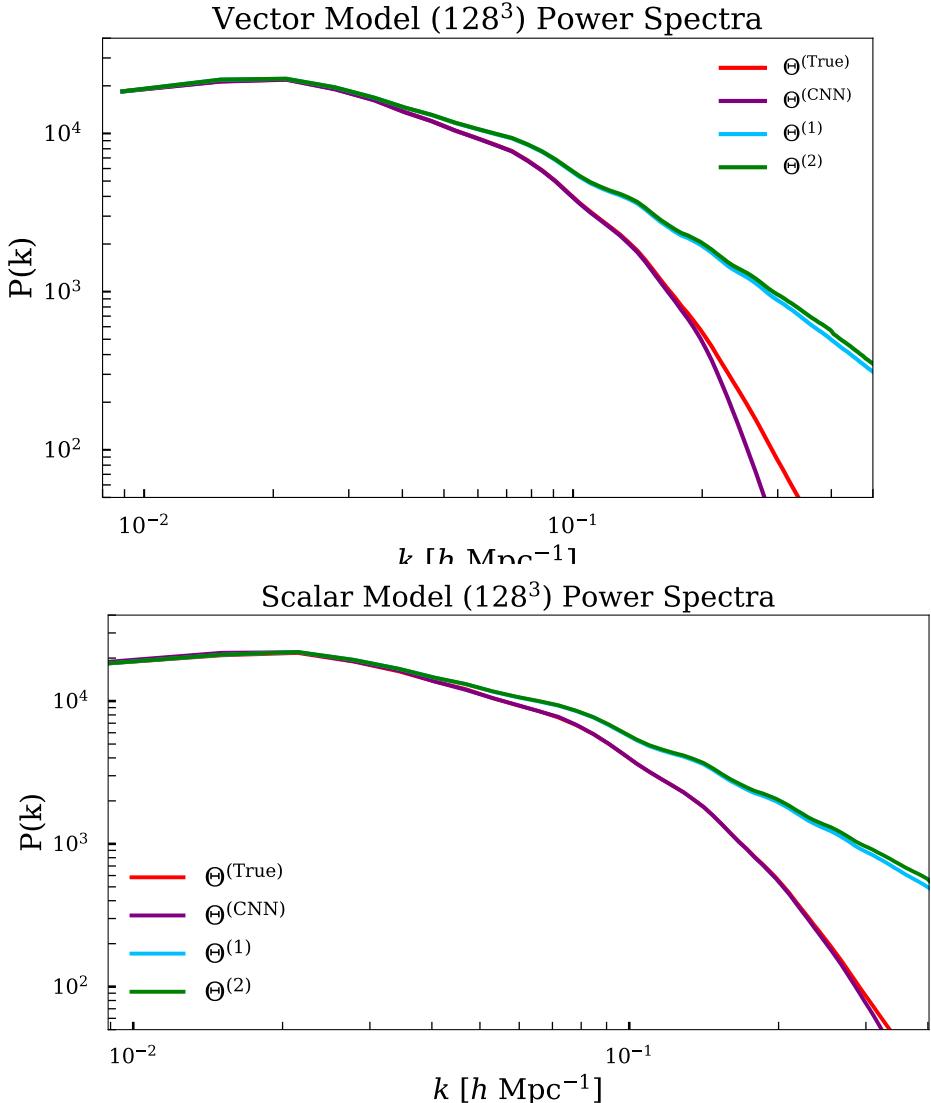


**Figure 8.** 2D slice of the divergence of the true displacement field (left), the divergence of the  $256^3$  scalar model estimate of the displacement field (middle), and the residuals between the two (right), at an arbitrary, but consistent, color scale. These are shown for a randomly selected test simulation with  $256^3$  particles.

and vector models to the true value, as well as that estimated by first- and second-order LPT. Figure 10 compares the square root of the ratio between the power spectra of each of the estimated displacement field divergences and the true displacement field divergence, and Figure 11 compares the normalized cross correlation between the estimates and truth. In all cases, we see that the CNN output far outperforms first- and second-order LPT. In the  $128^3$  vector model, we see reconstructed displacement fields with divergences that are highly correlated with the true value ( $r_{\Theta^{\text{True}} \Theta^{\text{CNN}}}(k) > 0.99$ ) for  $k \lesssim 0.15 \text{ h Mpc}^{-1}$ , which is slightly less than half the Nyquist frequency of  $k_N = 0.402 \text{ h Mpc}^{-1}$  for these simulations. By comparison, first- and second-order LPT only maintain such a strong correlation out to  $k \lesssim 0.06 \text{ h Mpc}^{-1}$ . Furthermore, we observe  $\left| \sqrt{P_{\Theta^{\text{CNN}}}(k)/P_{\Theta^{\text{True}}}(k)} - 1 \right| \leq 0.01$  out to similarly nonlinear scales, but for both first- and second-order LPT,  $\left| \sqrt{P_{\Theta^{\text{(LPT)}}}(k)/P_{\Theta^{\text{True}}}(k)} - 1 \right| \leq 0.01$  only for  $k \lesssim 0.015 \text{ h Mpc}^{-1}$ . This demonstrates that our model extends the range of scales for which the amplitude is tightly correlated by about a factor of 10 compared to LPT.

In the  $128^3$  scalar model, we observe tighter correlations at even more nonlinear scales, with both  $r_{\Theta^{\text{True}} \Theta^{\text{CNN}}}(k) > 0.99$  and  $\left| \sqrt{P_{\Theta^{\text{CNN}}}(k)/P_{\Theta^{\text{True}}}(k)} - 1 \right| \leq 0.01$  out to  $k \sim 0.2 \text{ h}^{-1} \text{ Mpc}$ . Whether this is because the divergence of the displacement field, which is a scalar field, is easier to learn or because the model architecture and training is better suited to this case (note, for example, we used double the number of steps for training the scalar model) is an open question. Regardless, it is promising that both models provide such significant improvements over LPT.

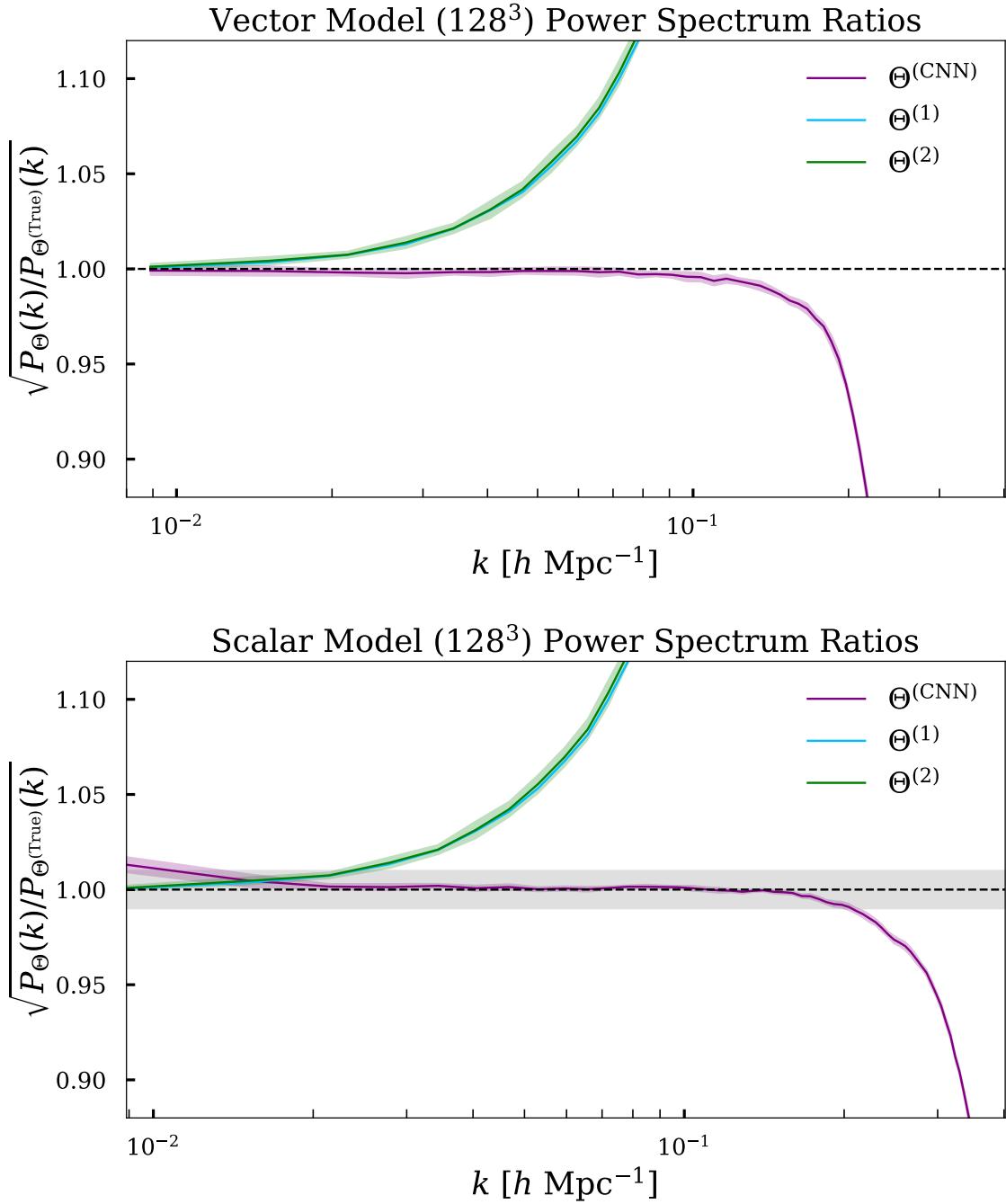
Figures 12 and 13 show the same power spectrum ratio and normalized correlation coefficient plots but for the  $256^3$  scalar model. In this case, we again see that the CNN outperforms LPT at small scales. The  $256^3$  scalar model shows a tight correlation with the true displacement field divergence value in both phase ( $r_{\Theta^{\text{True}} \Theta^{\text{CNN}}}(k) > 0.99$ ) and amplitude ( $\left( \left| \sqrt{P_{\Theta^{\text{CNN}}}(k)/P_{\Theta^{\text{True}}}(k)} - 1 \right| \leq 0.01 \right)$  for  $k \lesssim 0.3 \text{ h Mpc}^{-1}$ , which is approximately double the  $k$  values to which the  $128^3$  vector model maintains similarly strong correlation. Note that



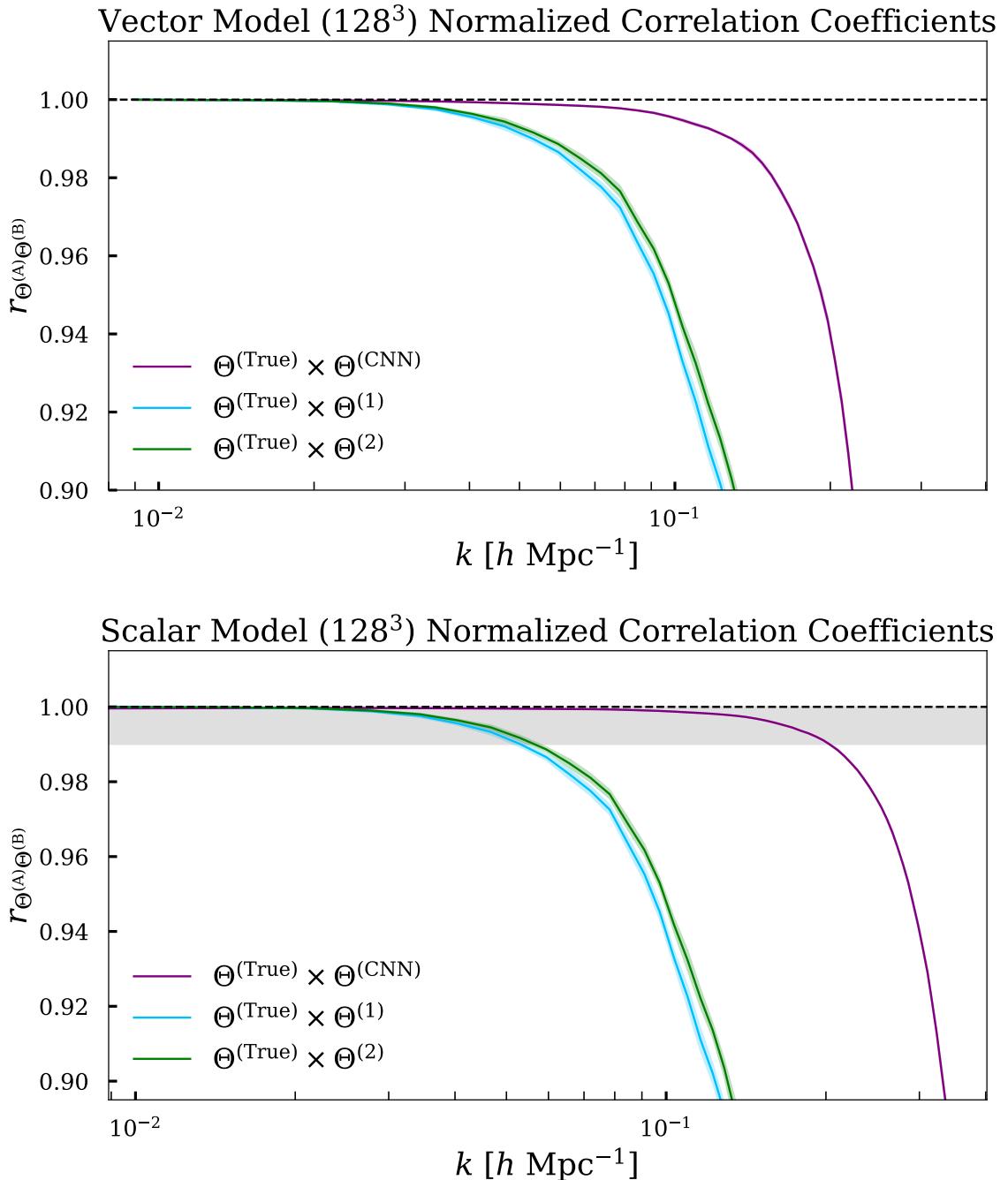
**Figure 9.** Power spectra of the divergences of the true, CNN prediction, Zel'dovich, and second-order LPT displacement fields for the  $128^3$  vector CNN model (top) and the  $128^3$  scalar model (bottom). The power spectra are averaged over 17 test simulations.

the Nyquist frequency also doubles as we go from  $128^3$  to  $256^3$  grid points. In this case, LPT only maintains such tight correlations out to  $k \lesssim 0.1 h \text{ Mpc}^{-1}$ . This demonstrates that our model and training scheme is easily scalable to larger simulations.

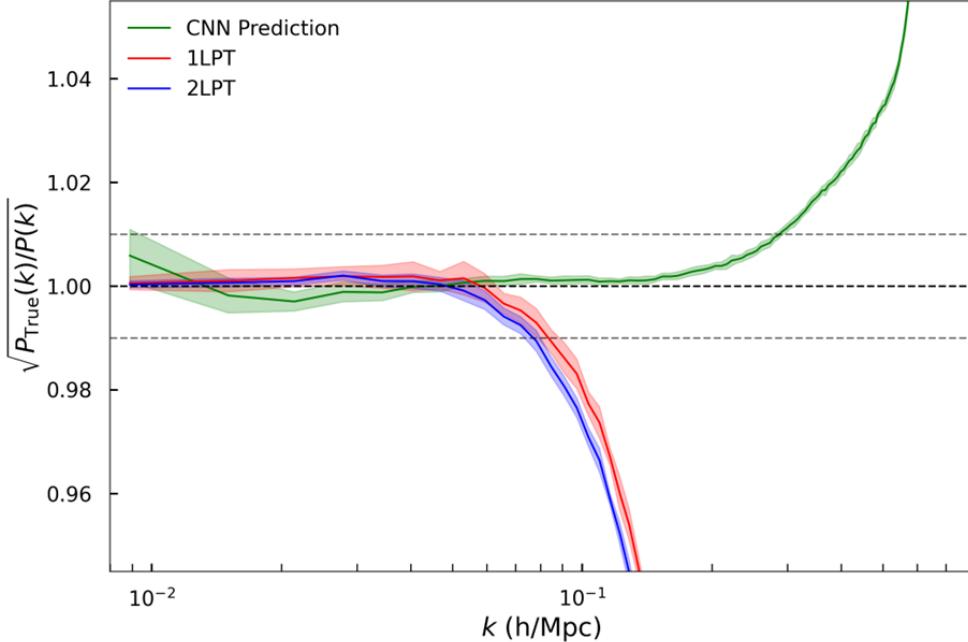
There are some noticeable deviations from a power spectrum ratio of unity at the largest scales for both the  $128^3$  scalar model, seen in the bottom panel of 10, and the  $256^3$  scalar model, seen in Figure 12. In the  $128^3$  scalar model, the ratio rises above unity at low  $k$ , and in the  $256^3$  scalar model, the ratio fluctuates above and below. Although these effects are all at the subpercent level, the most likely cause is the fact that CNNs are local models with a limited receptive field per pixel. That this effect is not observed in the  $128^3$  vector model may suggest that there is somehow more large-scale information encoded locally in the



**Figure 10.** Square root of the ratio of the power spectra of the divergence of each estimated displacement field to that of the true displacement field for the  $128^3$  vector CNN model (top) and the  $128^3$  scalar model (bottom), averaged over 17 test simulations. We take the average of the ratios rather than the ratio of the averages. The shaded area around each line represents a  $1\sigma$  standard deviation range around the mean. The horizontal dashed line marks a ratio of unity and the shaded gray area represents  $|\sqrt{P_\Theta}/P_{\Theta^{(\text{True})}} - 1| \leq 0.01$ .



**Figure 11.** Normalized correlation coefficients between the divergence of each estimated displacement field to and that of the true displacement field for the  $128^3$  vector CNN model (top) and the  $128^3$  scalar model (bottom), averaged over 17 test simulations. The shaded area around each line represents a  $1\sigma$  standard deviation range around the mean. The horizontal dashed line marks a value of unity, which is the maximum value of the normalized correlation coefficient. The shaded gray area represents  $r_{\Theta^{(\text{True})} \Theta^{(B)}} \geq 0.99$ .

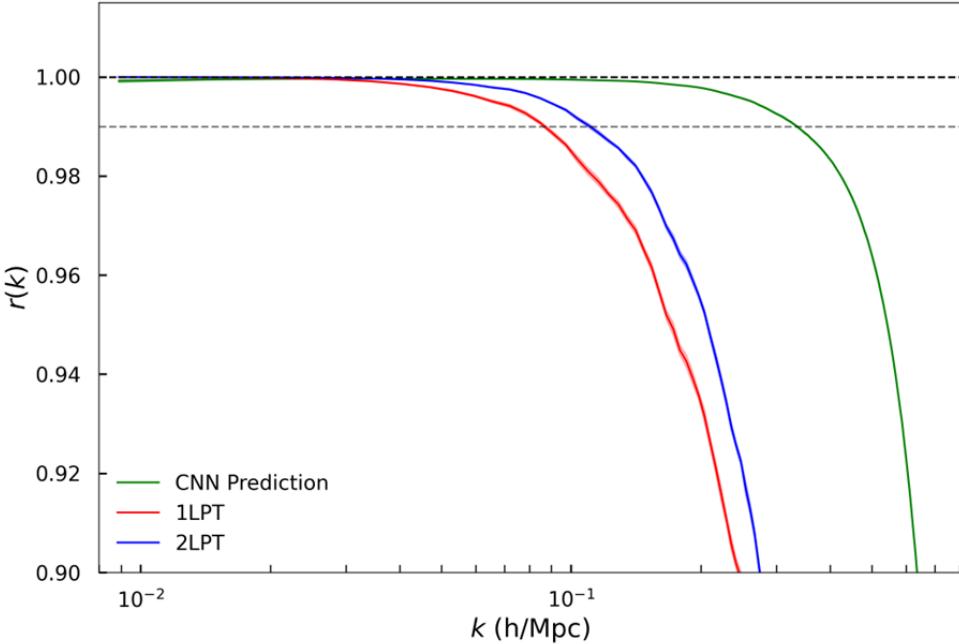


**Figure 12.** Same as 10, but for the  $256^3$  scalar CNN model. Here, the dark horizontal dashed line marks a ratio of unity and the lighter horizontal dashed lines mark  $\left| \sqrt{P_\Theta / P_{\Theta^{(\text{True})}}} - 1 \right| = 0.01$ .

second-order LPT displacement vector field than in the LPT potentials.

That the effect seems worsened as we increase the grid size from  $128^3$  to  $256^3$  is likely because of the CNN receptive field. Indeed, by not altering the architecture between the  $128^3$  and  $256^3$  models, we inherently reduce the physical receptive field of the  $256^3$  models. This is because the receptive field in units of grid points remains fixed, since the kernel size and number of layers is fixed; however, grid spacing in the simulations with  $256^3$  particles is half that in the simulations with  $128^3$  particles, because the number of grid points in each dimension is double while the box size remains the same at  $L = 1000 h^{-1}$  Mpc. It may thus prove beneficial to increase either the kernel size or the number of layers, or perform strided convolutions, to maintain fidelity at large scales.

Despite being the closest to our work in principle, it is difficult to directly compare our models to that of [88], which is significantly more complex in its architecture, since they use the Zel'dovich displacement as input and  $512^3$  particles in the same  $L = 1000 h^{-1}$  Mpc box size. Additionally, they train the model to learn only the corrections to the Zel'dovich displacement (i.e.,  $\Psi^{(\text{True})} - \Psi^{(1)}$ ), which may help at larger scales. However, from their Figure 1, it appears that they are able to maintain tight correlation between the CNN estimated displacement field and the true displacement field out to  $k \lesssim 0.8$ , although this is difficult to confirm given the scale of their plot. If we extrapolate from our scalar models, in which the maximum  $k$  at which there is tight correlation increases from  $\sim 0.15 h \text{ Mpc}^{-1}$  to  $\sim 0.3 h \text{ Mpc}^{-1}$  as we increase from the  $128^3$  model to the  $256^3$  model, we might expect similar performance to [88] for a  $512^3$  model.



**Figure 13.** Same as 11, but for the  $256^3$  scalar model. Here, the dark horizontal dashed line marks a value of unity and the lighter horizontal dashed line marks  $r_{\Theta^{(\text{True})}\Theta^{(\text{B})}} = 0.99$ .

## 5 Conclusions

In this work, we have first shown the LPT, even to higher-orders, is unable to accurately predict the nonlinear displacement field at small scales. Then, to resolve this, we have trained deep learning models to augment LPT at these nonlinear scales. Our models far outperform LPT alone in estimating either the displacement vector field or its divergence, and at different resolutions, demonstrating that there is more small-scale information encoded in the initial conditions than LPT itself is able to extract.

There are a plethora of directions to take future continuations of this work. One of our primary initial goals was to estimate the OT displacement field or potential, which can be calculated from an observed galaxy distribution. A cheap forward model for the OT field could prove valuable for improved reconstruction and field-level inference. Apart from model specifics, training to learn the OT field was to be the main distinguishing factor between our work and previous work using CNNs to learn the displacement field. We thus plan to train a similar CNN to estimate the OT potential from the initial conditions. Although producing the OT potential from PMWD simulations is computationally cheap, it might also be beneficial to scale the model to the HADES simulations, with  $512^3$  particles, since we already have OT displacement fields computed for the HADES simulations, as shown in Section 3,

We also hope to test the model on simulations generated with different cosmologies. This may manifest itself in a variety of ways. It would be worthwhile to first try to run the models we have already trained on test simulations with different cosmological parameters, though it is unlikely that the model will perform well on such a test, especially if the parameters are changed significantly from the fiducial values used for training. A second option would

be to simply add simulations with different cosmologies to the training data. These could be easily generated with PMWD, in which case we could also modify the transfer code used to generate the initial conditions, or we could use the Quijote suite of  $N$ -body simulations [90], which are the successors to HADES and which contain a set of simulations generated with a Latin hypercube of varying cosmological parameters. A more complex method would be to modify the CNN architecture itself and add parameters than encode, for example,  $\Omega_m$ , in a manner similar to [20].

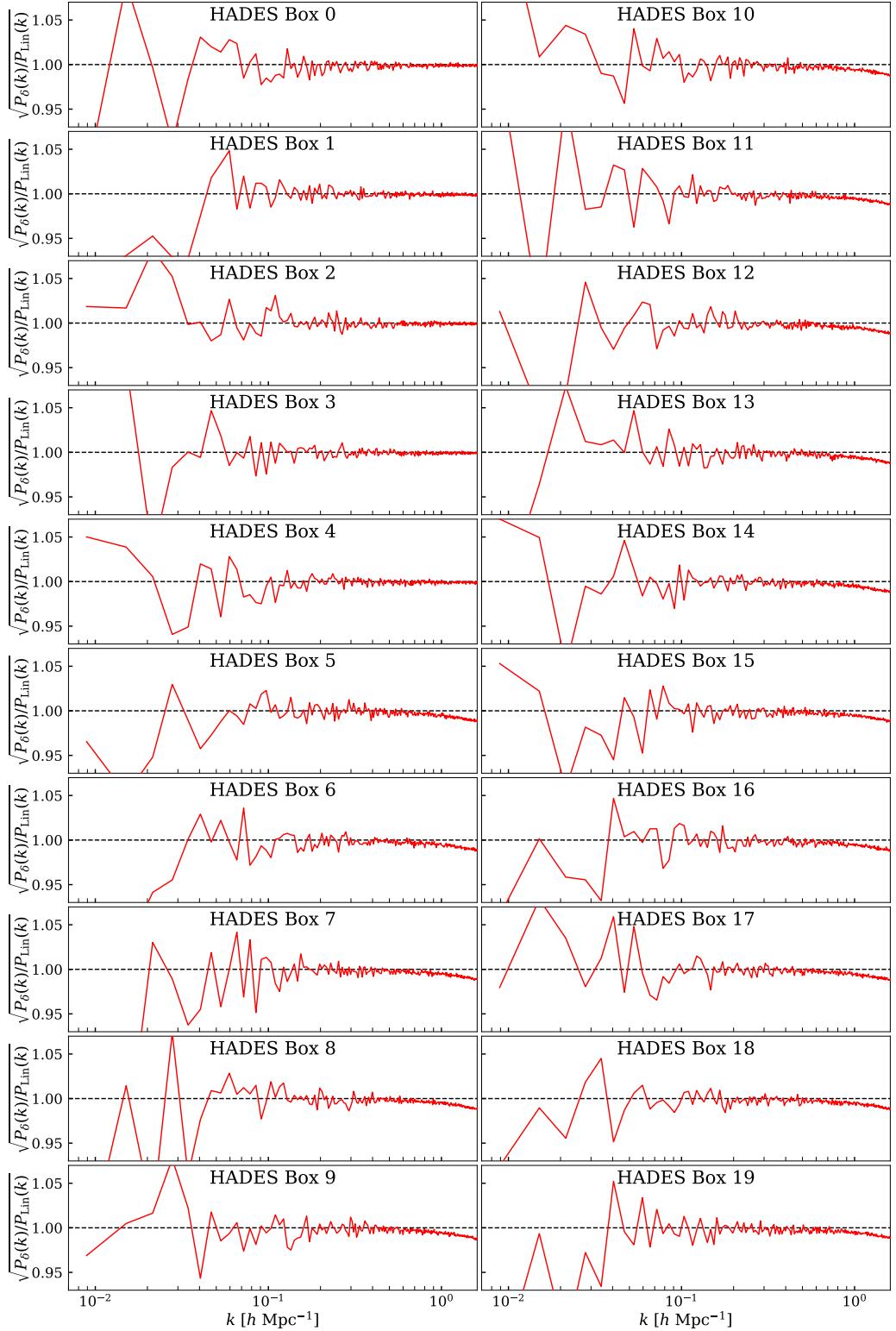
As always with deep learning, adjustments can always be made the model architecture. Unlike many of the models discussed in Section 2.6, we attempt to keep our CNN architecture as simple as possible. However, there are still ways to possibly improve performance at both small and large scales without drastically increasing the number of model parameters. Increasing the number of layers or the kernel size could help improve performance at large scales for simulations with a greater number of particles. Rather than a concatenation step at the input of the final convolutional layer, we could implement a global residual operation similar to that of [88, 92], in which case the output of the CNN will not be an estimate for the displacement field or its divergence but rather an estimate for the correction from the Zel'dovich (or second-order LPT) displacement field to the true displacement field. This may also help the model on large scales. It may also be useful to consider what the best physically motivated input fields would be. In this work, we use  $(\delta(\mathbf{q}), \phi^{(1)}(\mathbf{q}), \phi^{(2)}(\mathbf{q}))$  for the scalar model and  $\Psi^{(2)}(\mathbf{q})$  for the vector model, but other fields may offer some improvement. For example, it may be beneficial to replace  $\phi^{(2)}(\mathbf{q})$  with its Laplacian, which is more immediately connected to  $\Psi^{(2)}(\mathbf{q})$ . Testing the addition of the third-order LPT displacement or its scalar and vector potentials would also be interesting; if there is no improvement, it may indicate that the relevant information for the true displacement field is saturated in the lower orders of the LPT expansion. We could also modify the loss function to, for example, be weighted in Fourier space, as do [96]. This would penalize errors at small scales more than those at large scales.

We would also like to be able to better understand what the model is learning. Neural networks are generally thought of as black box functions, but there have been developments in providing model explanations, for example, through heatmaps that visualize which areas of an input image are most important to a model output [109]. Another approach would be to perform tests of the model on idealized initial conditions with analytic solutions, which is done by [92]. An entirely different idea would be to change the model to have a structure based on perturbative physics. In such a model, we might treat the filters of a convolutional layer as physical fields in their own right. Then, rather than nonlinearities in the neural network arising from the activation function, we could multiply the “fields” element-wise in real space between convolutional layers, thus mimicking the structure of higher-order LPT, in which derivatives of lower-order potentials are multiplied.

## A Further Considerations

### A.1 HADES Deviations from Linear Theory Power Spectrum

In the initial density fields of the 20 HADES boxes, calculated from the gridded initial velocity field as described in Section 3.2, several show an unexpected deviation from the linear theory power spectrum at the smallest of scales. Figure 14 shows that while boxes 0 through 4 maintain strong agreement between the initial density field power spectrum and the linear theory power spectrum out to the Nyquist frequency of  $k = 1.61 \text{ h Mpc}^{-1}$ , the rest of the boxes have a slight loss of power in the initial density field at small scales, at  $k \gtrsim 1 \text{ h Mpc}^{-1}$ . This explains the similar downward curve at small scales observed in the average ratio in Figure 1. We are unable to determine any reason for this. This is especially puzzling given the sequential ordering of the simulation boxes.



**Figure 14.** Same as Figure 1, but for each individual HADES box rather than the average over all boxes. Only the initial density field computed from the gridded velocity field (the red line of Figure 1) is shown.

## References

- [1] M. Colless, G. Dalton, S. Maddox, W. Sutherland, P. Norberg, S. Cole et al., *The 2dF Galaxy Redshift Survey: spectra and redshifts*, *Monthly Notices of the Royal Astronomical Society* **328** (2001) 1039.
- [2] D.H. Jones, M.A. Read, W. Saunders, M. Colless, T. Jarrett, Q.A. Parker et al., *The 6dF Galaxy Survey: final redshift release (DR3) and southern large-scale structures*, *Monthly Notices of the Royal Astronomical Society* **399** (2009) 683.
- [3] D.J. Eisenstein, D.H. Weinberg, E. Agol, H. Aihara, C. Allende Prieto, S.F. Anderson et al., *SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way, and Extra-Solar Planetary Systems*, *The Astronomical Journal* **142** (2011) 72.
- [4] L. Amendola, S. Appleby, A. Avgoustidis, D. Bacon, T. Baker, M. Baldi et al., *Cosmology and fundamental physics with the Euclid satellite*, *Living Reviews in Relativity* **21** (2018) 2.
- [5] Z. Ivezic, S.M. Kahn, J.A. Tyson, B. Abel, E. Acosta, R. Allsman et al., *LSST: From Science Drivers to Reference Design and Anticipated Data Products*, *The Astrophysical Journal* **873** (2019) 111.
- [6] DESI Collaboration, B. Abazajian, J. Aguilar, S. Ahlen, S. Alam, D.M. Alexander et al., *Overview of the Instrumentation for the Dark Energy Spectroscopic Instrument*, *The Astronomical Journal* **164** (2022) 207.
- [7] M. Davis, G. Efstathiou, C.S. Frenk and S.D.M. White, *The evolution of large-scale structure in a universe dominated by cold dark matter*, *The Astrophysical Journal* **292** (1985) 371.
- [8] E. Bertschinger, *Simulations of Structure Formation in the Universe*, *Annual Review of Astronomy and Astrophysics* **36** (1998) 599.
- [9] F. Bernardeau, S. Colombi, E. Gaztanaga and R. Scoccimarro, *Large-Scale Structure of the Universe and Cosmological Perturbation Theory*, *Physics Reports* **367** (2002) 1.
- [10] P. Monaco, T. Theuns and G. Taffoni, *The pinocchio algorithm: pinpointing orbit-crossing collapsed hierarchical objects in a linear density field*, *Monthly Notices of the Royal Astronomical Society* **331** (2002) 587.
- [11] R. Scoccimarro and R.K. Sheth, *PTHALOS: a fast method for generating mock galaxy distributions*, *Monthly Notices of the Royal Astronomical Society* **329** (2002) 629.
- [12] F.S. Kitaura and S. Hess, *Cosmological structure formation with augmented lagrangian perturbation theory*, *Monthly Notices of the Royal Astronomical Society* **435** (2013) L78.
- [13] M. White, J.L. Tinker and C.K. McBride, *Mock galaxy catalogues using the quick particle mesh method*, *Monthly Notices of the Royal Astronomical Society* **437** (2014) 2594.
- [14] Y. Feng, M.-Y. Chu, U. Seljak and P. McDonald, *FASTPM: a new scheme for fast simulations of dark matter and haloes*, *Monthly Notices of the Royal Astronomical Society* **463** (2016) 2273.
- [15] C. Modi, F. Lanusse and U. Seljak, *FlowPM: Distributed TensorFlow implementation of the FastPM cosmological N-body solver*, *Astronomy and Computing* **37** (2021) 100505.
- [16] Y. Li, L. Lu, C. Modi, D. Jamieson, Y. Zhang, Y. Feng et al., *pmwd: A Differentiable Cosmological Particle-Mesh \$N\$-body Library*, Nov., 2022. 10.48550/arXiv.2211.09958.
- [17] S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen et al., *Learning to predict the cosmological structure formation*, *Proceedings of the National Academy of Science* **116** (2019) 13825.
- [18] N. Perraudin, A. Srivastava, A. Lucchi, T. Kacprzak, T. Hofmann and A. Réfrégier, *Cosmological N-body simulations: a challenge for scalable generative models*, *Computational Astrophysics and Cosmology* **6** (2019) 5.

- [19] N. Kaushal, F. Villaescusa-Navarro, E. Giusarma, Y. Li, C. Hawry and M. Reyes, *NECOLA: Toward a Universal Field-level Cosmological Emulator*, *The Astrophysical Journal* **930** (2022) 115.
- [20] D. Jamieson, Y. Li, R.A. de Oliveira, F. Villaescusa-Navarro, S. Ho and D.N. Spergel, *Field-level Neural Network Emulator for Cosmological N-body Simulations*, *The Astrophysical Journal* **952** (2023) 145.
- [21] D. Piras, B. Joachimi and F. Villaescusa-Navarro, *Fast and realistic large-scale structure from machine-learning-augmented random field simulations*, *Monthly Notices of the Royal Astronomical Society* **520** (2023) 668.
- [22] P. Bhambra, B. Joachimi, O. Lahav and D. Piras, *PSI-GAN: a power-spectrum-informed generative adversarial network for the emulation of large-scale structure maps across cosmologies and redshifts*, *Monthly Notices of the Royal Astronomical Society* **536** (2025) 3138.
- [23] W.J. Percival, S. Cole, D.J. Eisenstein, R.C. Nichol, J.A. Peacock, A.C. Pope et al., *Measuring the Baryon Acoustic Oscillation scale using the Sloan Digital Sky Survey and 2dF Galaxy Redshift Survey*, *Monthly Notices of the Royal Astronomical Society* **381** (2007) 1053.
- [24] F. Beutler, C. Blake, M. Colless, D.H. Jones, L. Staveley-Smith, L. Campbell et al., *The 6dF Galaxy Survey: baryon acoustic oscillations and the local Hubble constant*, *Monthly Notices of the Royal Astronomical Society* **416** (2011) 3017 [[1106.3366](#)].
- [25] A. de Mattia, V. Ruhlmann-Kleider, A. Raichoor, A.J. Ross, A. Tamone, C. Zhao et al., *The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: measurement of the BAO and growth rate of structure of the emission line galaxy sample from the anisotropic power spectrum between redshift 0.6 and 1.1*, *Monthly Notices of the Royal Astronomical Society* **501** (2021) 5616.
- [26] X. Chen, Z. Ding, E. Paillas, S. Nadathur, H. Seo, S. Chen et al., *Extensive analysis of reconstruction algorithms for DESI 2024 baryon acoustic oscillations*, *arXiv e-prints* (2024) [[arXiv:2411.19738](#)] [[2411.19738](#)].
- [27] D.J. Eisenstein, H.-J. Seo, E. Sirk and D.N. Spergel, *Improving Cosmological Distance Measurements by Reconstruction of the Baryon Acoustic Peak*, *The Astrophysical Journal* **664** (2007) 675.
- [28] S. Tashev and M. Zaldarriaga, *Towards an optimal reconstruction of baryon oscillations*, *Journal of Cosmology and Astroparticle Physics* **2012** (2012) 006.
- [29] M. Schmittfull, Y. Feng, F. Beutler, B. Sherwin and M.Y. Chu, *Eulerian BAO reconstructions and N-point statistics*, *Physical Review D* **92** (2015) 123522.
- [30] R. Hada and D.J. Eisenstein, *An iterative reconstruction of cosmological initial density fields*, *Monthly Notices of the Royal Astronomical Society* **478** (2018) 1866.
- [31] H.-J. Seo, A. Ota, M. Schmittfull, S. Saito and F. Beutler, *Iterative reconstruction excursions for Baryon Acoustic Oscillations and beyond*, *Monthly Notices of the Royal Astronomical Society* **511** (2022) 1557.
- [32] B. Levy, R. Mohayaee and S. von Hausegger, *A fast semidiscrete optimal transport algorithm for a unique reconstruction of the early Universe*, *Monthly Notices of the Royal Astronomical Society* **506** (2021) 1165.
- [33] S. von Hausegger, B. Lévy and R. Mohayaee, *Accurate Baryon Acoustic Oscillations Reconstruction via Semidiscrete Optimal Transport*, *Physical Review Letters* **128** (2022) 201302.
- [34] F. Nikakhtar, R.K. Sheth, N. Padmanabhan, B. Lévy and R. Mohayaee, *Displacement field analysis via optimal transport: Multitracer approach to cosmological reconstruction*, *Physical Review D* **109** (2024) 123512 [[2403.11951](#)].

- [35] X. Chen, F. Zhu, S. Gaines and N. Padmanabhan, *Effective cosmic density field reconstruction with convolutional neural network*, *Monthly Notices of the Royal Astronomical Society* **523** (2023) 6272.
- [36] V. Jindal, A. Liang, A. Singh, S. Ho and D. Jamieson, *Predicting the Initial Conditions of the Universe using a Deterministic Neural Network*, Mar., 2023. 10.48550/arXiv.2303.13056.
- [37] C.J. Shallue and D.J. Eisenstein, *Reconstructing cosmological initial conditions from late-time structure with convolutional neural networks*, *Monthly Notices of the Royal Astronomical Society* **520** (2023) 6256.
- [38] J. Jasche and B.D. Wandelt, *Bayesian physical reconstruction of initial conditions from large-scale structure surveys*, *Monthly Notices of the Royal Astronomical Society* **432** (2013) 894.
- [39] H. Wang, H.J. Mo, X. Yang, Y.P. Jing and W.P. Lin, *ELUCID—Exploring the Local Universe with the Reconstructed Initial Density Field. I. Hamiltonian Markov Chain Monte Carlo Method with Particle Mesh Dynamics*, *The Astrophysical Journal* **794** (2014) 94.
- [40] U. Seljak, G. Aslanyan, Y. Feng and C. Modi, *Towards optimal extraction of cosmological information from nonlinear data*, *Journal of Cosmology and Astroparticle Physics* **2017** (2017) 009.
- [41] Y.B. Zel'dovich, *Gravitational instability: An approximate theory for large density perturbations.*, *Astronomy and Astrophysics* **5** (1970) 84.
- [42] M. Michaux, O. Hahn, C. Rampf and R.E. Angulo, *Accurate initial conditions for cosmological N-body simulations: minimizing truncation and discreteness errors*, *Monthly Notices of the Royal Astronomical Society* **500** (2021) 663.
- [43] T. Buchert, *Lagrangian theory of gravitational instability of friedman–lemaître cosmologies – a generic third-order model for non-linear clustering*, *Monthly Notices of the Royal Astronomical Society* **267** (1994) 811  
[<https://academic.oup.com/mnras/article-pdf/267/4/811/4056911/mnras267-0811.pdf>].
- [44] C. Rampf, *The recursion relation in Lagrangian perturbation theory*, *Journal of Cosmology and Astroparticle Physics* **2012** (2012) 004 [[1205.5274](#)].
- [45] C. Rampf and O. Hahn, *Shell-crossing in a  $\Lambda$ CDM Universe*, *Monthly Notices of the Royal Astronomical Society* **501** (2021) L71 [[2010.12584](#)].
- [46] F. Nikakhtar, R.K. Sheth, N. Padmanabhan, B. Lévy and R. Mohayaee, *Displacement field analysis via optimal transport: Multitracer approach to cosmological reconstruction*, *Physical Review D* **109** (2024) 123512.
- [47] R. Scoccimarro, *Transients from initial conditions: a perturbative analysis*, *Monthly Notices of the Royal Astronomical Society* **299** (1998) 1097.
- [48] M. Crocce, S. Pueblas and R. Scoccimarro, *Transients from initial conditions in cosmological simulations*, *Monthly Notices of the Royal Astronomical Society* **373** (2006) 369.
- [49] A. Lewis, A. Challinor and A. Lasenby, *Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models*, *The Astrophysical Journal* **538** (2000) 473.
- [50] D. Blas, J. Lesgourgues and T. Tram, *The Cosmic Linear Anisotropy Solving System (CLASS). Part II: Approximation schemes*, *Journal of Cosmology and Astroparticle Physics* **2011** (2011) 034.
- [51] J. Barnes and P. Hut, *A hierarchical  $O(N \log N)$  force-calculation algorithm*, *Nature* **324** (1986) 446.

- [52] V. Springel, *The cosmological simulation code GADGET-2*, *Monthly Notices of the Royal Astronomical Society* **364** (2005) 1105.
- [53] G. Xu, *A New Parallel N-Body Gravity Solver: TPM*, *The Astrophysical Journal Supplement Series* **98** (1995) 355.
- [54] F. Villaescusa-Navarro, A. Banerjee, N. Dalal, E. Castorina, R. Scoccimarro, R. Angulo et al., *The Imprint of Neutrinos on Clustering in Redshift Space*, *The Astrophysical Journal* **861** (2018) 53.
- [55] P.J.E. Peebles and J.T. Yu, *Primeval Adiabatic Perturbation in an Expanding Universe*, *The Astrophysical Journal* **162** (1970) 815.
- [56] R.A. Sunyaev and Y.B. Zeldovich, *Small-Scale Fluctuations of Relic Radiation*, *Astrophysics and Space Science* **7** (1970) 3.
- [57] D.J. Eisenstein and W. Hu, *Baryonic Features in the Matter Transfer Function*, *The Astrophysical Journal* **496** (1998) 605.
- [58] S. Cole, W.J. Percival, J.A. Peacock, P. Norberg, C.M. Baugh, C.S. Frenk et al., *The 2dF Galaxy Redshift Survey: power-spectrum analysis of the final data set and cosmological implications*, *Monthly Notices of the Royal Astronomical Society* **362** (2005) 505.
- [59] D.J. Eisenstein, I. Zehavi, D.W. Hogg, R. Scoccimarro, M.R. Blanton, R.C. Nichol et al., *Detection of the Baryon Acoustic Peak in the Large-Scale Correlation Function of SDSS Luminous Red Galaxies*, *The Astrophysical Journal* **633** (2005) 560.
- [60] C. Zhao, A. Variu, M. He, D. Forero-Sánchez, A. Tamone, C.-H. Chuang et al., *The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: cosmological implications from multitracer BAO analysis with galaxies and voids*, *Monthly Notices of the Royal Astronomical Society* **511** (2022) 5492.
- [61] J. Moon, D. Valcin, M. Rashkovetskyi, C. Saulder, J.N. Aguilar, S. Ahlen et al., *First detection of the BAO signal from early DESI data*, *Monthly Notices of the Royal Astronomical Society* **525** (2023) 5406.
- [62] T.M.C. Abbott, M. Adamow, M. Aguena, S. Allam, O. Alves, A. Amon et al., *Dark Energy Survey: A 2.1% measurement of the angular baryonic acoustic oscillation scale at redshift  $z_{\text{eff}}=0.85$  from the final dataset*, *Physical Review D* **110** (2024) 063515.
- [63] A. Meiksin, M. White and J.A. Peacock, *Baryonic signatures in large-scale structure*, *Monthly Notices of the Royal Astronomical Society* **304** (1999) 851.
- [64] H.-J. Seo and D.J. Eisenstein, *Baryonic Acoustic Oscillations in Simulated Galaxy Redshift Surveys*, *The Astrophysical Journal* **633** (2005) 575.
- [65] D.J. Eisenstein, H.-J. Seo and M. White, *On the Robustness of the Acoustic Scale in the Low-Redshift Clustering of Matter*, *The Astrophysical Journal* **664** (2007) 660.
- [66] M. Crocce and R. Scoccimarro, *Nonlinear evolution of baryon acoustic oscillations*, *Physical Review D* **77** (2008) 023533.
- [67] N. Padmanabhan and M. White, *Calibrating the baryon oscillation ruler for matter and halos*, *Physical Review D* **80** (2009) 063508.
- [68] H.-J. Seo, J. Eckel, D.J. Eisenstein, K. Mehta, M. Metchnik, N. Padmanabhan et al., *High-precision Predictions for the Acoustic Scale in the Nonlinear Regime*, *The Astrophysical Journal* **720** (2010) 1650.
- [69] N. Padmanabhan, X. Xu, D.J. Eisenstein, R. Scalzo, A.J. Cuesta, K.T. Mehta et al., *A 2 percent distance to  $z = 0.35$  by reconstructing baryon acoustic oscillations - I. Methods and application to the Sloan Digital Sky Survey*, *Monthly Notices of the Royal Astronomical Society* **427** (2012) 2132.

- [70] X. Xu, A.J. Cuesta, N. Padmanabhan, D.J. Eisenstein and C.K. McBride, *Measuring DA and H at z=0.35 from the SDSS DR7 LRGs using baryon acoustic oscillations*, *Monthly Notices of the Royal Astronomical Society* **431** (2013) 2834.
- [71] A.J. Ross, F. Beutler, C.-H. Chuang, M. Pellejero-Ibanez, H.-J. Seo, M. Vargas-Magaña et al., *The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: observational systematics and baryon acoustic oscillations in the correlation function*, *Monthly Notices of the Royal Astronomical Society* **464** (2017) 1168.
- [72] H. Gil-Marín, J.E. Bautista, R. Paviot, M. Vargas-Magaña, S. de la Torre, S. Fromenteau et al., *The Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: measurement of the BAO and growth rate of structure of the luminous red galaxy sample from the anisotropic power spectrum between redshifts 0.6 and 1.0*, *Monthly Notices of the Royal Astronomical Society* **498** (2020) 2492.
- [73] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, *Histoire de l'Académie Royale des Sciences de Paris* (1781) 666.
- [74] L. Kantorovich, *On the translocation of masses*, *Comptes Rendus (Doklady) de l'Academie des Sciences de l'URSS* **37** (1941) 199.
- [75] C. Villani and others, *Optimal transport: old and new*, vol. 338, Springer (2008).
- [76] S. Kolouri, S.R. Park, M. Thorpe, D. Slepcev and G.K. Rohde, *Optimal Mass Transport: Signal processing and machine-learning applications*, *IEEE Signal Processing Magazine* **34** (2017) 43.
- [77] U. Frisch, S. Matarrese, R. Mohayaee and A. Sobolevski, *A reconstruction of the initial conditions of the Universe by optimal mass transportation*, *Nature* **417** (2002) 260.
- [78] Y. Brenier, U. Frisch, M. Hénon, G. Loeper, S. Matarrese, R. Mohayaee et al., *Reconstruction of the early Universe as a convex optimization problem*, *Monthly Notices of the Royal Astronomical Society* **346** (2003) 501.
- [79] F. Nikakhtar, R.K. Sheth, B. Lévy and R. Mohayaee, *Optimal Transport Reconstruction of Baryon Acoustic Oscillations*, *Physical Review Letters* **129** (2022) 251101.
- [80] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, *Nature* **521** (2015) 436.
- [81] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning representations by back-propagating errors*, *Nature* **323** (1986) 533.
- [82] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard et al., *Handwritten digit recognition with a back-propagation network*, in *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, NIPS'89, (Cambridge, MA, USA), pp. 396–404, MIT Press, 1989.
- [83] N. Aloysius and M. Geetha, *A review on deep convolutional neural networks*, in *2017 International Conference on Communication and Signal Processing (ICCP)*, pp. 0588–0592, Apr., 2017, DOI.
- [84] M. Huertas-Company and F. Lanusse, *The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys*, *Publications of the Astronomical Society of Australia* **40** (2023) e001.
- [85] C. Modi, Y. Feng and U. Seljak, *Cosmological reconstruction from galaxy light: neural network based light-matter connection*, *Journal of Cosmology and Astroparticle Physics* **2018** (2018) 028.
- [86] X. Zhang, Y. Wang, W. Zhang, Y. Sun, S. He, G. Contardo et al., *From Dark Matter to Galaxies with Convolutional Networks*, Feb., 2019. 10.48550/arXiv.1902.05965.

- [87] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al., *Generative Adversarial Networks*, June, 2014. 10.48550/arXiv.1406.2661.
- [88] R. Alves de Oliveira, Y. Li, F. Villaescusa-Navarro, S. Ho and D.N. Spergel, *Fast and Accurate Non-Linear Predictions of Universes with Deep Learning*, Nov., 2020. 10.48550/arXiv.2012.00240.
- [89] O. Ronneberger, P. Fischer and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, May, 2015. 10.48550/arXiv.1505.04597.
- [90] F. Villaescusa-Navarro, C. Hahn, E. Massara, A. Banerjee, A.M. Delgado, D.K. Ramanah et al., *The Quijote Simulations*, *The Astrophysical Journal Supplement Series* **250** (2020) 2.
- [91] S. Tashev, M. Zaldarriaga and D.J. Eisenstein, *Solving large scale structure in ten easy steps with COLA*, *Journal of Cosmology and Astroparticle Physics* **2013** (2013) 036.
- [92] D. Jamieson, Y. Li, S. He, F. Villaescusa-Navarro, S. Ho, R. Alves de Oliveira et al., *Simple lessons from complex learning: what a neural network model learns about cosmic structure formation*, June, 2022. 10.48550/arXiv.2206.04573.
- [93] D. Jamieson, Y. Li, F. Villaescusa-Navarro, S. Ho and D.N. Spergel, *Field-level emulation of cosmic structure formation with cosmology and redshift dependence*, *Journal of Cosmology and Astroparticle Physics* **2025** (2025) 072.
- [94] L. Lucie-Smith, H.V. Peiris, A. Pontzen, B. Nord and J. Thiyagalingam, *Deep learning insights into cosmological structure formation*, *Physical Review D* **109** (2024) 063524.
- [95] T.-X. Mao, J. Wang, B. Li, Y.-C. Cai, B. Falck, M. Neyrinck et al., *Baryon acoustic oscillations reconstruction using convolutional neural networks*, *Monthly Notices of the Royal Astronomical Society* **501** (2021) 1499.
- [96] L. Parker, A.E. Bayer and U. Seljak, *Initial Conditions from Galaxies: Machine-Learning Subgrid Correction to Standard Reconstruction*, Apr., 2025. 10.48550/arXiv.2504.01092.
- [97] F. Qin, D. Parkinson, S.E. Hong and C.G. Sabiu, *Reconstructing the cosmological density and velocity fields from redshifted galaxy distributions using V-net*, *Journal of Cosmology and Astroparticle Physics* **2023** (2023) 062.
- [98] X. Xiao, J. Ding, X.L. Luo, S.K. Lan, L. Xiao, S. Liu et al., *AI-Powered Reconstruction of Dark Matter Velocity Fields from Redshift-Space Halo Distribution*, Nov., 2024. 10.48550/arXiv.2411.11280.
- [99] Z. Wu, Z. Zhang, S. Pan, H. Miao, X. Luo, X. Wang et al., *Cosmic Velocity Field Reconstruction Using AI*, *The Astrophysical Journal* **913** (2021) 2.
- [100] Z. Wu, L. Xiao, X. Xiao, J. Wang, X. Kang, Y. Wang et al., *AI-assisted reconstruction of cosmic velocity field from redshift-space spatial distribution of haloes*, *Monthly Notices of the Royal Astronomical Society* **522** (2023) 4748.
- [101] S.E. Hong, D. Jeong, H.S. Hwang and J. Kim, *Revealing the Local Cosmic Web from Galaxies by Deep Learning*, *The Astrophysical Journal* **913** (2021) 76.
- [102] P. Ganeshiah Veena, R. Lilow and A. Nusser, *Large-scale density and velocity field reconstructions with neural networks*, *Monthly Notices of the Royal Astronomical Society* **522** (2023) 5291.
- [103] N.A. Maksimova, L.H. Garrison, D.J. Eisenstein, B. Hadzhiyska, S. Bose and T.P. Satterthwaite, *ABACUSSUMMIT: a massive set of high-accuracy, high-resolution N-body simulations*, *Monthly Notices of the Royal Astronomical Society* **508** (2021) 4017.
- [104] L.H. Garrison, D.J. Eisenstein, D. Ferrer, N.A. Maksimova and P.A. Pinto, *The ABACUS cosmological N-body code*, *Monthly Notices of the Royal Astronomical Society* **508** (2021) 575.

- [105] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin et al., *JAX: composable transformations of Python+NumPy programs*, 2018.
- [106] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner et al., *Flax: A neural network library and ecosystem for JAX*, 2024.
- [107] DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce et al., *The DeepMind JAX Ecosystem*, 2020.
- [108] B.T. Polyak, *Some methods of speeding up the convergence of iteration methods*, *USSR Computational Mathematics and Mathematical Physics* **4** (1964) 1.
- [109] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*, Oct., 2016.  
10.48550/arXiv.1610.02391.