

PineTree: A generative, fast, and differentiable halo model for wide-field galaxy surveys

Simon Ding^{1,*}, Guilhem Lavaux¹, and Jens Jasche²

¹ Sorbonne Université, CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98 bis boulevard Arago, 75014 Paris, France

² The Oskar Klein Centre, Department of Physics, Stockholm University, Albanova University Center, SE 106 91 Stockholm, Sweden

Received 2 July 2024 / Accepted 1 August 2024

ABSTRACT

Context. Accurate mock halo catalogues are indispensable data products for developing and validating cosmological inference pipelines. A major challenge in generating mock catalogues is modelling the halo or galaxy bias, which is the mapping from matter density to dark matter halos or observable galaxies. To this end, N -body codes produce state-of-the-art catalogues. However, generating large numbers of these N -body simulations for big volumes, especially if magnetohydrodynamics are included, requires significant computational time.

Aims. We introduce and benchmark a differentiable and physics-informed neural network that can generate mock halo catalogues of comparable quality to those obtained from full N -body codes. The model design is computationally efficient for the training procedure and the production of large mock catalogue suites.

Methods. We present a neural network, relying only on 18 to 34 trainable parameters, that produces halo catalogues from dark matter overdensity fields. The reduction in network weights was realised through incorporating symmetries motivated by first principles into our model architecture. We trained our model using dark-matter-only N -body simulations across different resolutions, redshifts, and mass bins. We validated the final mock catalogues by comparing them to N -body halo catalogues using different N -point correlation functions.

Results. Our model produces mock halo catalogues consistent with the reference simulations, showing that this novel network is a promising way to generate mock data for upcoming wide-field surveys due to its computational efficiency. Moreover, we find that the network can be trained on approximate overdensity fields to reduce the computational cost further. We also present how the trained network parameters can be interpreted to give insights into the physics of structure formation. Finally, we discuss the current limitations of our model as well as more general requirements and pitfalls of approximate halo mock generation that became evident from this study.

Key words. methods: statistical – galaxies: abundances – galaxies: halos – galaxies: statistics – dark matter – large-scale structure of Universe

1. Introduction

Modern surveys like *Euclid* (Laureijs et al. 2011; Euclid Collaboration 2024b), DESI (DESI Collaboration 2016), LSST (LSST Science Collaboration 2009), PFS (Takada et al. 2014), SKA (Braun et al. 2019; Weltman et al. 2020), WFIRST (Green et al. 2012), and SPHEREx (Doré et al. 2014) are transforming cosmology into a data-driven research field with their unprecedented survey volume and instrument sensitivity. The constraints on cosmological parameters will shift from a noise-dominated regime to a systematics-dominated one. Thus, progress in the field will critically depend on our ability to model the data with ever-increasing accuracy and precision.

To this end, simulated observations or mock data are essential data products for developing new models and cosmological inference pipelines, as is demonstrated by *Beyond-2pt* Collaboration (2024). Another application utilises mocks to estimate the sample covariance matrix numerically for clustering statistics with the two-point and three-point correlation functions (like Percival et al. 2014). With the advance of machine-learning techniques in recent years, cosmologists have also explored various applications by leveraging simulations to

train deep neural networks (e.g. Ravanbakhsh et al. 2017; Fluri et al. 2019; Villanueva-Domingo & Villaescusa-Navarro 2022; Anagnostidis et al. 2022; Perez et al. 2023; Boonkirkord et al. 2023; Jamieson et al. 2023; Legin et al. 2024). Notably, the combination of machine learning with Bayesian statistics known as simulation-based inference (SBI) or implicit likelihood inference (ILI) has established itself as a promising framework in cosmological analysis (Hahn et al. 2023; Ho et al. 2024; Makinen et al. 2022; de Santi et al. 2023).

The state-of-the-art method of generating mock observables involves N -body codes such as Gadget (Springel et al. 2021), Ramses (Teyssier 2002), ENZO, Pkdgrav-3 (an earlier version described by Stadel 2001), and Abacus to trace the evolution of matter over cosmic time. Subsequently, halo finders like Friends-of-friends (Davis et al. 1985), SUBFIND (Springel et al. 2001), HOP (Eisenstein & Hut 1998), AdaptaHOP (Colombi 2013; Aubert et al. 2004), Amiga Halo Finder (Knollmann & Knebe 2009), ASOBF (Planelles & Quilis 2010), and ROCKSTAR (Behroozi et al. 2013) group the particles hierarchically into halos or galaxies. These powerful N -body simulations can be computationally expensive, especially if they aim to cover large survey volumes, include baryonic physics, or explore the effect of different astrophysical and cosmological parameters

* Corresponding author; simon.ding@iap.fr

like Euclid Collaboration (2024a); Villaescusa-Navarro et al. (2020); Nelson et al. (2019); Garrison et al. (2018); Ishiyama et al. (2021). Running these simulations requires millions of CPU hours and petabytes of storage. A growing need for simulation suites with large sets of independent samples aggravates the issue of computational resources. Work done by Blot et al. (2016) shows that more than 5000 independent mock observations are necessary in a Euclid-like setting to estimate the numerical covariance with adequate accuracy for the matter power spectrum. Similarly, the performance of machine-learning methods is directly dependent on the number of available simulations. In situations with sparse training data, deep neural networks are likely to over-fit, and thus may yield biased or overconfident results, as is discussed by Hermans et al. (2021) for ILI algorithms.

Several methods have been proposed to alleviate the reliance on running large sets of high-fidelity N -body simulations. For example, techniques like CARPool (Chartier et al. 2021; Chartier & Wandelt 2021) or ensembling (Hermans et al. 2021) reduce the required number of expensive N -body runs. An alternative approach involves building approximate simulators that are computationally faster than N -body codes, while maintaining sufficient accuracy. Blot et al. (2016) demonstrates that for matter power spectrum analysis in a Euclid-like scenario, the cosmological constraints are substantially weaker with scale cuts of $k \leq 0.2 h \text{ Mpc}^{-1}$, whereas only adopting linear covariance to include smaller scales results in overconfident parameter estimations. This effect will be exacerbated for other summary statistics such as k -th nearest neighbour (Yuan et al. 2023), density-split clustering (Paillas et al. 2023), skew spectrum (Hou et al. 2024), wavelet scattering transforms (Valogiannis et al. 2024; Régalo-Saint Blancard et al. 2024), cosmic voids (Hamaus et al. 2020; Contarini et al. 2023), or full field-level methods (Jasche & Wandelt 2013; Porqueres et al. 2023; Villaescusa-Navarro et al. 2021; Zhou & Dodelson 2023; Schmidt et al. 2019; Nguyen et al. 2024; Seljak et al. 2017; Jindal et al. 2023; Doerer et al. 2023), since these approaches are shown to be even more sensitive to non-linear and non-Gaussian signals in the data. Therefore, it is crucial to explore approximation methods to produce mocks that accurately capture non-linear effects to smaller scales ($k > 0.2 h \text{ Mpc}^{-1}$) as this will ultimately determine how much information and physical insights we shall be able to extract from upcoming surveys (Nguyen et al. 2021, 2024). A vital challenge in generating fast and realistic mock catalogues is to model the link between the matter density field and the resulting galaxies or their host halos, denoted as galaxy or halo bias.

The scope of this paper focuses on constructing dark matter (DM) halos, since around 80% of the total matter is made out of DM, and this hence dominates the dynamics. These halo catalogues are well-founded intermediate data products that can then be transformed into galaxy catalogues through methods such as ‘halo occupation distribution’ (e.g. Jing et al. 1998; Peacock & Smith 2000; Seljak 2000; Yang et al. 2003; Zheng et al. 2005; Yang et al. 2008, 2009; Zehavi et al. 2011, henceforth HOD), ‘subhalo abundance matching’ (e.g. Vale & Ostriker 2004; Behroozi et al. 2010; Moster et al. 2010; Reddick et al. 2013; Chaves-Montero et al. 2016; Lehmann et al. 2017; Stiskalek et al. 2021, hereinafter SHAM), ADDGALS (Wechsler et al. 2022), or semi-analytic models (e.g. Yung et al. 2022a,b).

In this work, we propose a generative and physics-informed neural network that we call PineTree (Physical and interpretable network for tracer emulation), which emulates the halo bias and can sample realistic DM halo catalogues in a computationally efficient way. The subsequent section will briefly review existing mock generation approaches and set the

context for our method. We shall then continue to describe the set-up of our model and how it is trained and used to generate mock catalogues in Sects. 3 and 4. This naturally leads to Sect. 5, in which we discuss our results, before we conclude with a summary and outline of follow-up projects and applications.

2. Overview of mock generation frameworks

The first analytical approaches to predict DM halo distributions – namely, the halo mass function, from DM density fields – were pioneered by Kaiser (1984); Bardeen et al. (1986) and later refined (e.g. through works by Bond 1987; Bond et al. 1988; Carlberg & Couchman 1989; Cole & Kaiser 1989; Mo & White 1996; Sheth & Tormen 1999) into the formalism known now as ‘peak background split’. In contrast, an alternative approach based on ‘excursion sets’ (e.g. Schaeffer & Silk 1985, 1988; Kaiser 1988; Cole & Kaiser 1988, 1989; Efstathiou et al. 1988; Efstathiou & Rees 1988; Narayan & White 1988) that was initially formulated by Press & Schechter (1974) as the Press-Schechter formula was developed. It was later extended by Bond et al. (1991) and further improved with the addition of ellipsoidal collapse (Sheth et al. 2001; Sheth & Tormen 2002). Using these physically derived schemes, it is possible as a first approximation to predict the halo mass function and, at the first order, the linear relation between the halo and matter density contrast,

$$\delta_h = b \delta_m, \quad (1)$$

with b as a scale-independent bias parameter. δ_m and δ_h denote the (dark) matter and halo relative density contrast, respectively.

However, the linear bias is only a valid approximation for the largest cosmological scales as it fails to account for any non-linear (Kravtsov & Klypin 1999) and scale-dependent (Smith et al. 2007) behaviour. Fry & Gaztanaga (1993) extended the analytical framework to quasi-linear scales through ‘local bias expansion’. The perturbative bias expansion expresses the halo number density (or galaxy number density) as a function of the matter density field and its higher-order derivatives like the tidal field. For an extensive review of the perturbative bias treatment, we refer the reader to Desjacques et al. (2018). Notably, the concept of ‘local-in-matter-density bias’ (Desjacques et al. 2018, hereinafter LIMD bias) has been extended to phenomenological bias models beyond the linear bias such as power-law parametrisations (Cen & Ostriker 1993; de la Torre & Peacock 2013) or other non-linear bias functions (Neyrinck et al. 2014; Repp & Szapudi 2020). The main idea of using LIMD bias models for mock generation is to produce a mean halo density field from a smoothed matter density field and then sample the halo field (e.g. through a Poisson distribution) to get a discrete count field.

Chan et al. (2012); Sheth et al. (2013) show – and this was highlighted again by Bartlett et al. (2024) – that the local bias is insufficient to predict the halo distribution correctly. Here, it is also important to emphasise that phenomenological bias methods need to be validated thoroughly since there is no theoretical guarantee of their accuracy, unlike the perturbative approach. Figure 1 depicts the breakdown of a local biasing scheme as well as the discrepancies in different validation metrics for which we fit the truncated power law (TPL) bias to a gravity-only N -body simulation following (Neyrinck et al. 2014). We show that the calibrated TPL can reproduce the overall halo count distribution of the simulation, effectively validating the one-point statistics. But when comparing the power spectra (two-point correlation) we can see a clear difference between the sampled mock fields

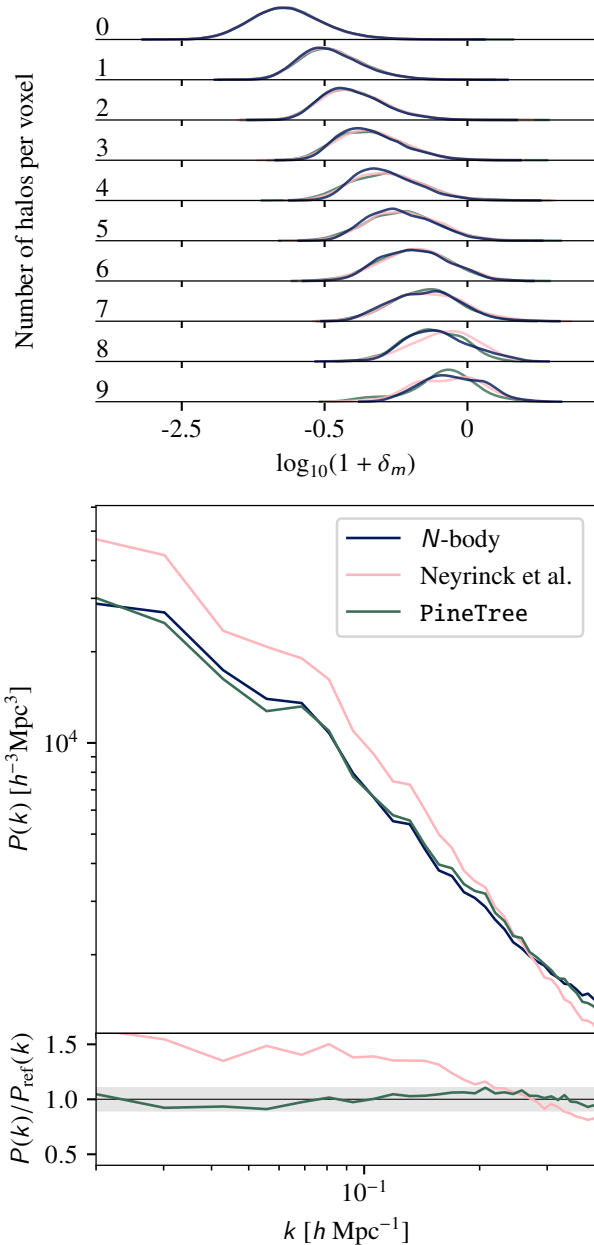


Fig. 1. Comparison between using one-point statistics (upper panel) and two-point statistics (lower panel) as validation metrics. The first panel on top shows a kernel density estimate plot of the number of voxels with a given halo count as a function of the underlying DM overdensity. Each line corresponds to the number of halos per voxel, as is indicated by the annotated text on the left. The distributions were produced using a Gaussian kernel with a bandwidth set according to Scott’s rule (Scott 2015). The dark blue, pink, and green curves show the number distribution obtained from N -body simulation, the TPL by Neyrinck et al. (2014), and PineTree respectively. In the bottom panel, we show the mean power spectrum from the generated fields and ratios with respect to the reference N -body simulation. The grey band indicates the 10% deviation threshold.

and the reference N -body simulation. Bartlett et al. (2024) show that this trend is true for any LIMD bias scheme. Therefore, it is imperative to validate the mocks across different statistical orders, especially if the scientific target is to generate realistic halos on the field level.

So far, we have only discussed methods utilising fast analytic halo bias schemes to generate mocks. An alternative to reduce (considerably) the computational cost is to accelerate the generation of the underlying DM density by approximate gravity solvers. This type of approach, also known as ‘Lagrangian methods’, involves the computation of an approximate matter distribution coupled with a deterministic halo formation prescription (see Monaco 2016 for a more comprehensive review). The first methods of this class such as PINOCCHIO Monaco et al. (2002b,a); Taffoni et al. (2002), PEAK-PATCH (Bond & Myers 1996), and PTHalos (Scoccimarro & Sheth 2002; Manera et al. 2012; Manera et al. 2014) adopted Lagrangian perturbation theory for the density field computation. Later works propose to utilise more accurate gravity solvers based on the particle-mesh scheme such as PMFAST (Merz et al. 2005), FastPM (Feng et al. 2016), COLA (Tassev et al. 2013; Koda et al. 2016; Izard et al. 2016; Ding et al. 2024), and its variants L-PICOLA (Howlett et al. 2015a) and sCOLA (Tassev et al. 2015; Leclercq et al. 2020). One downside of these algorithms is that they rely on Friends-of-friends or other particle-based halo finders, which can result in a computationally intense process when high accuracy is required (see for instance Blot et al. 2019; Monaco 2016). Still, these methods have been readily adopted in combination with HOD or SHAM to generate mock surveys, as is demonstrated by Howlett et al. (2015b); Drinkwater et al. (2010); Kazin et al. (2014); Ferrero et al. (2021).

The natural extension of these approaches is to combine approximate gravity solvers to produce smoothed density fields with simplified bias prescriptions, as has been done by PATCHY (Kitaura et al. 2013), QPM (White et al. 2013), EZmocks (Chuang et al. 2014), and Halogen (Avila et al. 2015). Notably, these methods include a stochastic component to address the stochasticity of halo bias (as discussed by e.g. Pen 1998; Tegmark & Bromley 1999; Dekel & Lahav 1999; Matsubara 1999; Sheth & Lemson 1999; Taruya & Soda 1999; Seljak & Warren 2004; Neyrinck et al. 2005; Hamaus et al. 2010; Baldauf et al. 2013). Despite utilising (local) bias schemes that can be difficult to calibrate, this type of mock generation approach has been successfully applied by Kitaura et al. (2015, 2016); Vakili et al. (2017) to produce large suites of accurate mock catalogues (see also Chuang et al. 2015; Monaco 2016 for a comparison of some of the methods).

In recent years, new techniques for mock generation have been explored to achieve even greater accuracy and computational efficiency. For instance, BAM (Balaguera-Antolínez et al. 2018, 2019, 2023; Kitaura et al. 2022; García-Farieta et al. 2024) proposes a non-parametric bias scheme that samples halos based on the underlying matter density field and its associated cosmic web-type information motivated by Yang et al. (2017); Fisher & Faltenbacher (2017). Meanwhile, Modi et al. (2018); Berger & Stein (2018); Zhang et al. (2019); Piras et al. (2023); Pandey et al. (2023); Pellejero Ibañez et al. (2024) have investigated the use of machine learning as bias emulators, since deep neural networks as universal function approximators are powerful tools to accurately model the non-linear behaviour on small scales. Yet, due to their large and degenerate parameter space, neural networks can be difficult and computationally expensive to set up and train, requiring large amounts of high-fidelity training samples. Moreover, large networks are black-box models that cannot be easily interpreted after training. The following section will introduce our mock generation pipeline, PineTree, which leverages machine-learning techniques to incorporate a stochastic and non-local bias scheme, while maintaining an interpretable

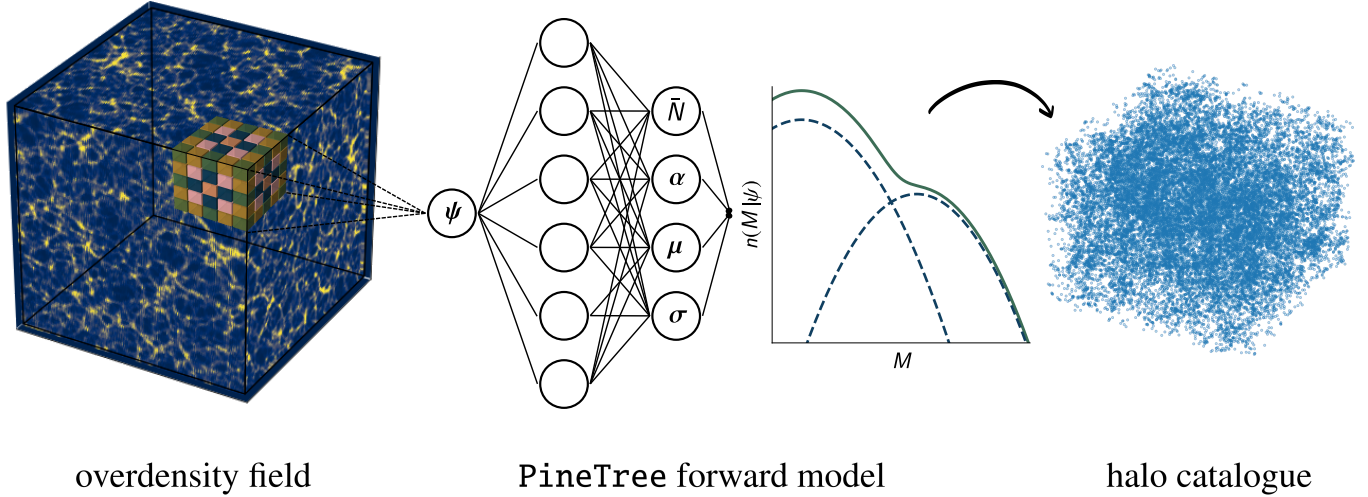


Fig. 2. Schematic view of the PineTree pipeline. The following steps are depicted from left to right: the input DM overdensity field was convolved with a rotation-symmetric kernel; a mixture density network computed the distribution parameters of a log-normal mixture distribution with the feature map, ψ ; halo masses were drawn from the emulated distributions to construct the final halo catalogue.

and reduced parameter space through a physics-motivated model architecture.

3. PineTree

PineTree is a computationally fast and generative model with a minimal set of trainable parameters that takes as an input a gridded DM overdensity field, $\delta_m(x)$, and outputs a halo count overdensity field, $\delta_h(x)$, or halo catalogue, $\{M_h(x)\}$, at the same gridding resolution. The mock generation pipeline builds upon the neural physical engine first described by Charnock et al. (2020) and is divided into three parts. The first subsection presents the model architecture and how it differs from conventional machine learning approaches. We then derive the objective function to optimise the model's parameters and conclude the section by describing how the trained model generates the final halo catalogues (for an overview of the set-up, see the schematic outline in Fig. 2). The code is written in Python and is implemented in the Jax¹ framework to enable GPU acceleration. A public version of the source code is hosted on Bitbucket².

3.1. Network architecture

The network architecture consists of two functional parts: a module to encode the environmental information of the input, and a neural network that approximates the halo mass function. As was mentioned in Sect. 2, we want the network to consider each cell and its surrounding environment. Thus, a network architecture building on a convolutional neural network is the natural choice. Furthermore, we want to encode our prior knowledge into the design of the network; namely, the principal physical forces of structure formation such as gravity are isotropic with regard to the underlying particle distribution. This means that our bias model should a priori be sensitive to neither the specific orientation nor the location of the DM overdensity field when predicting the resulting halo distribution. Hence, the network operation performed on the input field needs to be rotationally and translationally invariant.

3.1.1. Building environment summaries with a convolutional neural network

A standard convolution layer (LeCun et al. 1989) performs a discrete convolution on an input field, I , with a set of kernel weights, ω^{Kernel} , and is therefore, by definition, already translation-invariant. In the three-dimensional case, the layer operation with a kernel of size $M \times N \times L$ can be formally expressed as

$$\psi_{ijk} = (I * \omega^{\text{Kernel}})_{ijk} = \sum_m \sum_n \sum_l I_{i+m, j+n, k+l} \omega_{mnl}^{\text{Kernel}} + b, \quad (2)$$

where the set of indices, $\{ijk\}$, relate to a given voxel in the discretised three-dimensional field. In the above equation, the index m runs from $-(M-1)/2$ to $(M-1)/2$, and in the same way it also runs for the indices n and l to centre the convolutional kernel on the input voxel, I_{ijk} . In the machine-learning language, the output, ψ_{ijk} , is called a feature map and b is denoted as the neural bias parameter that needs to be learned alongside the kernel weights. In the context of this work, however, the term bias will henceforth refer to the DM halo connection, unless stated otherwise. The feature map, ψ_{ijk} , can be understood as the compressed information of the overdensity field patch around the central overdensity voxel, δ_{ijk} .

To enforce rotational invariance, we applied a multi-pole expansion to the convolutional kernel. Inspired by a conventional multi-pole expansion in which a generic function is decomposed as

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l C_l^m Y_l^m(\theta, \varphi), \quad (3)$$

with C_l^m denoting the constant coefficients and Y_l^m the spherical harmonics, we constructed a kernel that has shared weights for kernel positions with

$$C_l^m(r, \theta, \varphi) = r Y_l^m(\theta, \varphi), \quad (4)$$

where r is the Euclidean distance from the centre of the kernel to the position of the weight. Effectively, we decomposed a convolutional kernel with independent weights into a set of kernels

¹ <https://github.com/google/jax>

² <https://bitbucket.org/dingzimeng/pinetree>

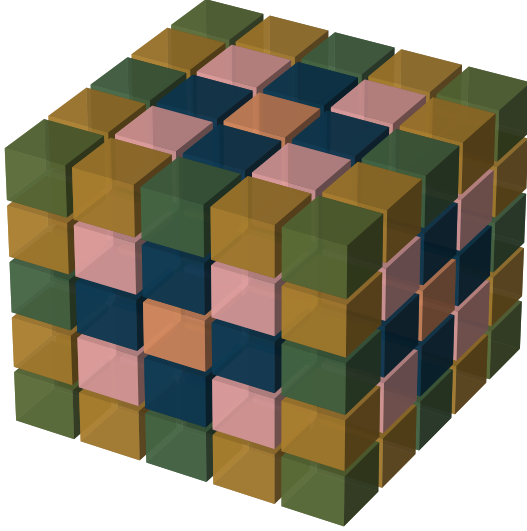


Fig. 3. Visualisation of a $5 \times 5 \times 5$ monopole kernel. Identical colours indicate shared weights resulting in a rotationally symmetric pattern.

with shared weights, for a given l and m , which is determined by the spherical harmonics; that is, kernel positions with the same $C_l^m(r, \theta, \varphi)$ according to Eq. (4). For a rotationally invariant kernel, we only considered the monopole contributions with $l = m = 0$. The output of the convolution operation then only encodes rotational and translational invariant features. In addition, using this multi-pole expansion approach, the number of independent weights is greatly reduced. For instance, a $5 \times 5 \times 5$ kernel will have only ten independent weights instead of 125. Figure 3 visualises the shared weights of a monopole kernel.

To reduce the parameter space and maximise the computational speed of the model, we adopted a single filter – the monopole kernel – for the convolutional network. Therefore, the feature map, ψ_i , for a patch around a central overdensity voxel, δ_i , is presented by a scalar. We note that our public implementation of the method has already been designed to make it possible to include higher-order multi-poles to handle direction-dependent effects (e.g. tidal forces), and potential improvements of utilising a multi-filter convolutional network approach will be explored in future works. Each multi-pole contribution will result in a separate kernel, so in general the output of the convolutional layer can also become a vector, ψ , with each entry corresponding to a distinct feature map associated with a different kernel. The relation to the multi-pole expansion gives us a direct way of interpreting the kernel weights of our model, indicating how important specific spatial locations are for the model prediction. In Sect. 5.4, we shall discuss the interpretation in detail for a trained monopole kernel.

3.1.2. The mixture distribution

The second part of the PineTree model is designed to digest the non-local information encoded in ψ and predict a probability function that can then be used to generate halo samples. In this case, we want to model the conditional halo mass function, $n(M|\{\delta\}_i)$; that is, the number of expected halos for mass, M , conditioned on an overdensity patch, $\{\delta\}_i$, for the i -th voxel. To ensure that the model can produce flexible and well-defined distributions, we chose a mixture of log-normal distributions, which are also frequently utilised in the machine learning community as density estimation networks. The formal expression of the

conditional halo number density then reads:

$$n(M|\{\delta\}_i) = \frac{\bar{N}_i}{V} \sum_{n=1}^{N_{\text{mix}}} \frac{\alpha_{ni}}{M \sqrt{2\pi\sigma_{ni}^2}} \exp\left[-\frac{(\ln M - \mu_{ni})^2}{2\sigma_{ni}^2}\right], \quad (5)$$

with N_{mix} as the number of mixture distributions, $\bar{N}_i$ the predicted mean halo number for voxel i , and V denoting the voxel volume. The conditioning on the overdensity patch enters through the previously computed feature map, ψ_i , via the mixture network parameters by:

$$\bar{N}_i = \exp[\omega^{\bar{N}} \psi_i + b^{\bar{N}}], \quad (6)$$

$$\alpha_{ni} = \text{softmax}(\omega_n^\alpha \psi_i + b_n^\alpha), \quad (7)$$

$$\sigma_{ni} = \exp[\omega_n^\sigma \psi_i + b_n^\sigma], \quad (8)$$

$$\mu_{ni} = \begin{cases} \omega_n^\mu \psi_i + b_n^\mu, & \text{if } n = 0, \\ \max[0, \omega_n^\mu \psi_i + b_n^\mu] + \mu_{n-1}, & \text{if } n > 0. \end{cases} \quad (9)$$

The parameters $\{\omega_n^{\bar{N}}, b_n^{\bar{N}}, \omega_n^\alpha, b_n^\alpha, \omega_n^\mu, b_n^\mu, \omega_n^\sigma, b_n^\sigma\}$ are defined globally for all voxels and the density dependency only enters through ψ_i . Furthermore, the activation function in Eq. (7) ensures the mixture distribution is properly normalised so that $\sum_n \alpha_{ni} = 1$ holds.

3.1.3. Final transformations

The final model was put together by first transforming the input overdensity, $\delta_m(x)$, into the logarithmic space $\log[1 + \delta_m(x)]$ and then compressing it with the monopole kernel by a convolutional layer. The extracted feature map was subsequently transformed by the ‘softplus’ activation function before finally being chained with the mixture density network. Thus, the full network describes a non-local and non-linear transformation from an overdensity environment patch to a halo number distribution at the environment centre.

It should be highlighted again that, due to the physically motivated design choices, the model only contains a very small parameter set compared to contemporary neural networks, which typically have hundreds of thousands to millions of free parameters. This was realised by utilising the symmetric single-kernel convolutional layer, and the shallow network that translates the feature map, ψ , into the mixture distribution parameters, as is described in Eqs. (6) to (9). For a kernel with size $3 \times 3 \times 3$ and a mixture density network of two components, the PineTree model has a total of 18 parameters, making it suitable to directly infer the model parameters from a given dataset in the context of Bayesian inference or using it as a Bayesian neural network. Section 5.4 discusses the effect of utilising different kernel sizes for which the number of model parameters can increase.

3.2. Loss function

The optimisation problem at hand is to find the best model parameters so that the sampled halo catalogues from our model match the halo catalogues obtained from a N -body halo finder. The evaluation function quantifying the distance of the prediction to the ground truth is called the objective or loss function. In line with the purpose of having an interpretable model, we used a physical likelihood function as the objective function, following the maximum likelihood principle to optimise the model parameters. This approach also enables the easy incorporation of PineTree into existing Bayesian forward modelling frameworks.

The probability distribution of the observed data, d , given the underlying forward model with parameter s is referred to as the likelihood, $\mathcal{P}(d|s)$. The explicit form of the likelihood depends on our assumption about the stochasticity of the observable. The likelihood, in the case of additive noise, can be formally written down by defining the measurement equation,

$$d = R(s) + n, \quad (10)$$

where the function $R(s)$ denotes the forward model that maps the signal to the expected observation and n denotes an additive noise. Using Eq. (10) and assuming a reasonable distribution for the measurement noise, $\mathcal{P}_N(n)$, (often Gaussian or Poissonian), the likelihood can then be written down as

$$\begin{aligned} \mathcal{P}(d|s) &= \int \mathcal{P}(d, n|s) dn = \int \mathcal{P}(n|s) \mathcal{P}(d|n, s) dn \\ &= \int \mathcal{P}(n|s) \delta_D(d - R(s)) dn \\ &= \mathcal{P}_N(n = d - R(s)), \end{aligned} \quad (11)$$

with δ_D denoting the Dirac delta function.

In this work, we assume that the number of halos per mass bin and per voxel is Poisson-distributed:

$$N_{i,M}^{\text{halo}} \sim \mathcal{P}_{\text{Poisson}}(\lambda_{i,M}), \quad (12)$$

where $\lambda_{i,M}$ denotes the mean (i.e. the Poisson intensity), and the indices i and M refer to the i -th voxel from the grid and the mass bin, $[M, M + \Delta M]$, respectively. Therefore, the mean halo number per voxel for each halo mass bin can be related to the conditional halo number density by

$$\lambda_{i,M} \equiv \langle N_{i,M} \rangle = V \int_M^{M+\Delta M} n(M|\delta_i) dM, \quad (13)$$

with V denoting the voxel volume. The Poisson log-likelihood for observing $N_{i,M}^{\text{obs}}$ is then given by

$$\ln \mathcal{P}_{i,M}(d = N_{i,M}^{\text{obs}} | \lambda_{i,M}) = N_{i,M}^{\text{obs}} \ln \lambda_{i,M} - \ln N_{i,M}^{\text{obs}}! - \lambda_{i,M}. \quad (14)$$

For the optimisation scheme, only the gradient with respect to $\lambda_{i,M}$ is relevant. Therefore, terms depending only on $N_{i,M}^{\text{obs}}$ can be discarded, and we shall omit them from now on. Furthermore, we assume that voxels as well as halo mass bins are conditionally independent, yielding the total log-likelihood:

$$\ln \mathcal{P}(N^{\text{obs}} | \lambda) = \sum_{i \in \text{voxels}} \sum_{M \in \text{mass bins}} N_{i,M}^{\text{obs}} \ln \lambda_{i,M} - \lambda_{i,M}. \quad (15)$$

If the halo mass bin width is decreased to the limit $\Delta M \rightarrow 0$, then the observed number of halos, $N_{i,M}^{\text{obs}}$, can either be one or zero, and the above sums can be rewritten as

$$\ln \mathcal{P}(N^{\text{obs}} | \lambda) = \sum_{j \in \text{halos}} \ln \lambda_{i(j), M(j)} - \sum_{i \in \text{voxels}} \int_{M_{\text{th}}}^{\infty} dM \lambda_{i,M}, \quad (16)$$

where the indices $i(j)$ and $M(j)$ correspond to the voxel and mass bin of halo j , respectively. Moreover, M_{th} denotes the lower mass threshold of the halos in the simulation.

In the optimisation context, following maximum likelihood, the loss function, $\mathcal{L}$, is given by the negative log-likelihood.

So, by inserting Eq. (5) and Eq. (13) into Eq. (16), the final expression that needs to be minimised reads:

$$\begin{aligned} \mathcal{L} &= - \sum_{j \in \text{halos}} \ln \left[\sum_{n=1}^{N_{\text{mix}}} \frac{\tilde{\alpha}_{ni(j)}}{\sqrt{2\pi\sigma_{ni(j)}^2}} \exp \left[-\frac{(\ln M_j - \mu_{ni(j)})^2}{2\sigma_{ni(j)}^2} \right] \right] \\ &\quad + \sum_{i \in \text{voxels}} \int_{M_{\text{th}}}^{\infty} \sum_{n=1}^{N_{\text{mix}}} \frac{\tilde{\alpha}_{ni(j)}}{M \sqrt{2\pi\sigma_{ni(j)}^2}} \exp \left[-\frac{(\ln M - \mu_{ni(j)})^2}{2\sigma_{ni(j)}^2} \right] dM \\ &= - \sum_{j \in \text{halos}} \ln \left[\sum_{n=1}^{N_{\text{mix}}} \frac{\tilde{\alpha}_{ni(j)}}{\sqrt{2\pi\sigma_{ni(j)}^2}} \exp \left[-\frac{(\ln M_j - \mu_{ni(j)})^2}{2\sigma_{ni(j)}^2} \right] \right] \\ &\quad + \sum_{i \in \text{voxel}} \sum_{n=1}^{N_{\text{mix}}} \frac{\tilde{\alpha}_{ni(j)}}{2} \text{erfc} \left(\frac{\ln M_{\text{th}} - \mu_{ni(j)}}{\sqrt{2\sigma_{ni(j)}^2}} \right). \end{aligned} \quad (17)$$

3.3. Sampling of halos

After obtaining the model parameters with the derived loss function from Eq. (17) and a suitable optimisation scheme, as is described in Sect. 4.2, the PineTree model could then be used to generate mock halo catalogues for a given overdensity field. The sampling procedure was divided into two parts: first, sampling the total number of halos per voxel, and thereafter the mass realisation of each halo. For the number count generation, we need to use the initial assumption that the halos are distributed according to a Poisson distribution (see Eq. (12)). Utilising the additive property of the Poisson distribution and the previous assumption of statistically independent mass bins, we can directly sample the halo count for all mass bins of one voxel rather than separately sampling each mass bin. With regards to the PineTree model, the predicted Poisson intensity per voxel is computed by

$$\lambda_i = \sum_{M \in \text{mass bins}} \lambda_{i,M} = \sum_{n=1}^{N_{\text{mix}}} \frac{\tilde{\alpha}_{ni}}{2} \text{erfc} \left(\frac{\ln M_{\text{th}} - \mu_{ni}}{\sqrt{2\sigma_{ni}^2}} \right), \quad (18)$$

which we subsequently use to generate a number realisation, $N_i^{\text{halo}} \sim \mathcal{P}_{\text{Poisson}}(\lambda_i)$. Finally, according to the sampled halo count for each voxel, the individual halo masses are drawn in parallel from the predicted mixture density,

$$P(M) = \sum_{n=1}^{N_{\text{mix}}} \frac{\alpha_{ni}}{M \sqrt{2\pi\sigma_{ni}^2}} \exp \left[-\frac{(\ln M - \mu_{ni})^2}{2\sigma_{ni}^2} \right]. \quad (19)$$

We note that the above expression only differs from the conditional halo number density from Eq. (5) by a factor of $\tilde{N}_i V^{-1}$.

4. Training data and configuration

Following the theoretical description of our model in Sect. 3, we detail the technical set-up and parameter choices that led to the results shown in the subsequent section. We first describe the N -body simulations that served as training and validation data before ending the section by reporting the specific model and training configurations that were tested.

Table 1. Range of tested PineTree configurations.

Setting	Used configurations
Kernel size	$3^3, 5^3, 7^3$
Voxel size (h^{-1} Mpc)	50, 25, 16, 8, 4
Redshift	0, 1
Gravity model	L-Gadget, 2LPT

4.1. Description of the simulation dataset

The N -body simulations were obtained by running a non-public variant of Gadget-2 (Springel 2005); namely, L-Gadget, which is a more computationally efficient version that only simulates collisionless matter. The initial conditions for the N -body simulations were computed by the initial condition generator, GenIC (Nishimichi et al. 2009, 2010; Valageas & Nishimichi 2011). We produced 60 dark-matter-only simulations with different random seeds for the initial conditions using Planck 2018 cosmology (Planck Collaboration VI 2020). To capture large-scale dynamics with good particle resolution for luminous red galaxy halos (e.g. van Uitert, Edo et al. 2015), we ran each simulation with a box size of $500^3 h^{-3} \text{ Mpc}^3$ and 512^3 particles, which results in a particle mass resolution of $8.16 \times 10^{10} h^{-1} M_{\odot}$. We set GenIC to use second-order Lagrangian perturbation theory (2LPT) to evolve the initial conditions until a redshift of $z = 24$, and the successive L-Gadget evolution to $z = 0$ and $z = 1$ was executed using a gravity softening of 0.05ℓ , where $\ell \equiv (V_{\text{box}}/N_{\text{part}})^{1/3}$ is the mean inter-particle separation, with V_{box} as the volume of the simulation and N_{part} denoting the total number of particles. Using the same initial condition seeds as for the L-Gadget simulations, another set of 60 snapshots for each redshift of $z \in \{0, 1\}$ was produced through GenIC only. Henceforth, we refer to this second set of simulations as 2LPT simulations. We used the cloud-in-cell (CIC) interpolation described in Hockney & Eastwood (1981) to obtain the final overdensity fields that were used as inputs for PineTree.

We validated the simulations by comparing the power spectra to the theoretical prediction from the Boltzmann solver CLASS (e.g. Lesgourgues 2011; Blas et al. 2011), which is shown in Appendix A. Finally, we used ROCKSTAR with a linking length of $b = 0.28$ and minimum halo particle set to 10 particles to produce the corresponding halo catalogues. In this work, we adopted the virial mass definition according to Bryan & Norman (1998), leading to a lower halo mass threshold of $3 \times 10^{12} h^{-1} M_{\odot}$ for each catalogue. Again, we checked the halo catalogues for potential inconsistencies by comparing the simulated halo mass function to theoretical predictions from Tinker et al. (2008) (see Appendix A).

4.2. Model configuration

In this work, we aim to study the scope of PineTree’s applicability. Thus, we conducted an extensive parameter study and summarised the different configuration choices in Table 1.

For all results presented in the following section, the number of log-normal distributions was always set to two. This was motivated by the intuition on halo mass functions provided by Press & Schechter (1974); Tinker et al. (2008) and insights from the set of N -body simulations (see Sect. 4.1). In addition to the listed configurations in Table 1, we tested further configurations, such as a different number of mixture components. However, these runs did not significantly improve our results and a complete

report on the specific settings with their outcome is given in Appendix B.1.

We opted for the gradient-based optimiser Adam by Kingma & Ba (2014) to achieve the optimisation. The learning rate was set to 0.001 for all results shown in this work. Other values and various learning rate schedulers were also tested without notable improvements. We considered the network parameters to have converged when the loss computed on the set of validation simulations saturated. Unless stated otherwise, the 60 simulations were split into a training set comprising 10 simulations and the validation set was made up of 30 different simulations that the model did not see during training. The remaining 20 simulations were used for testing and we shall justify our choice of only utilising 10 simulations for the training procedure in Sect. 5.5.

5. Results

To assess the quality of the PineTree predictions, we compared the sampled mock catalogues with the halo catalogues from the dedicated validation set generated by L-Gadget and rockstar. As is described in Sect. 2 and highlighted in Fig. 1, it is important to validate by comparing the results with different levels of N -point correlation functions. The main metrics used in this work comprise the halo mass function, $n(M)$, power spectrum, $P(k)$, and reduced bispectrum, $Q_{k_1, k_2}(\theta)$, that probe the 1-point, 2-point, and 3-point correlation, respectively. The power spectrum and reduced bispectrum were computed using the package Pylians3 (Villaescusa-Navarro 2018). We begin this section by reporting the model efficiency with different runs following Table 1. We then discuss the insights that can be gained by examining the trained model weights and conclude with a benchmark on computational time.

5.1. Best-case reconstruction

The runs with PineTree in this subsection aim to determine the best possible configuration for emulation and set up a reference to compare with subsequent experiments. Therefore, we started with the DM overdensities from L-Gadget at a redshift of $z = 0$ and performed a grid search over the different field resolutions according to Table 1, but with a fixed kernel size of $3 \times 3 \times 3$.

We find that the best-performing setting is for mock samples generated at the $7.81 h^{-1} \text{ Mpc}$ voxel resolution and for halos with a virial mass, M_{vir} , between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$. Figure 4 shows the three main metrics on the aforementioned best-configuration case. For the marginal halo mass function shown in the first panel, we find good agreement between both catalogues to percent-level accuracy from the mass threshold up to halos of $1 \times 10^{15} h^{-1} M_{\odot}$, where the variance of the high mass halos starts to dominate. A similar performance, with the two ensembles not deviating more than one sigma from each other, can be seen in the next two panels, which display the comparison using the power spectrum and the reduced bispectrum. In addition, we report that the conditional halo mass function for a set of overdensity bins agrees between the predicted mock and validation catalogues, as is depicted in Fig. 5. We note that the shape of the predicted mass function by PineTree can change significantly over the range of underlying overdensities and we can see that the second log-normal distribution becomes apparent for high-overdensity regions.

Computing the power spectrum for a given halo count field averages over the phases, and hence the comparison between the mock and validation catalogues yields only the difference in the

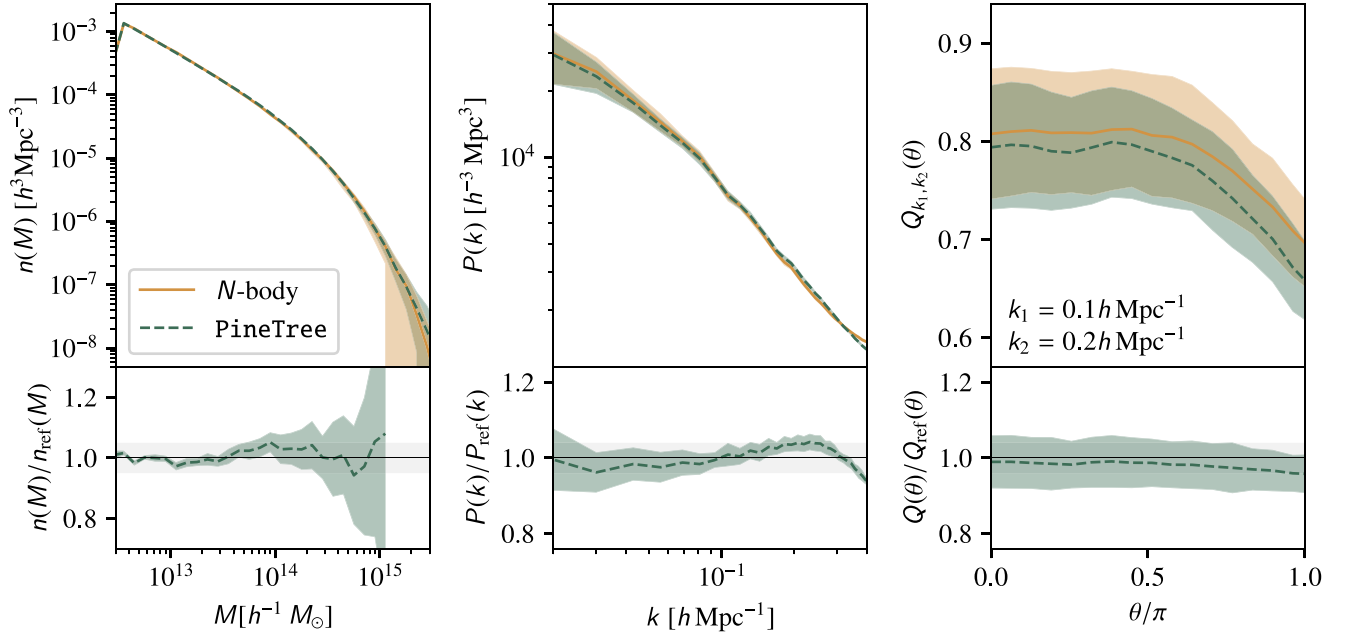


Fig. 4. Marginal halo mass function, power spectrum, and reduced bispectrum from the left to right panel for halo density fields with a virial mass between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$ at a $7.81 h^{-1}$ Mpc voxel resolution. All curves were obtained from 30 catalogues, with the dashed green and the solid orange line representing the ensemble mean of PineTree mocks and validation data, respectively. The 1σ line is outlined by the coloured shaded regions. The bottom subplot in each panel shows the ratio of the respective summary statistic between the mock and validation set where the shaded grey region is the 5% deviation threshold.

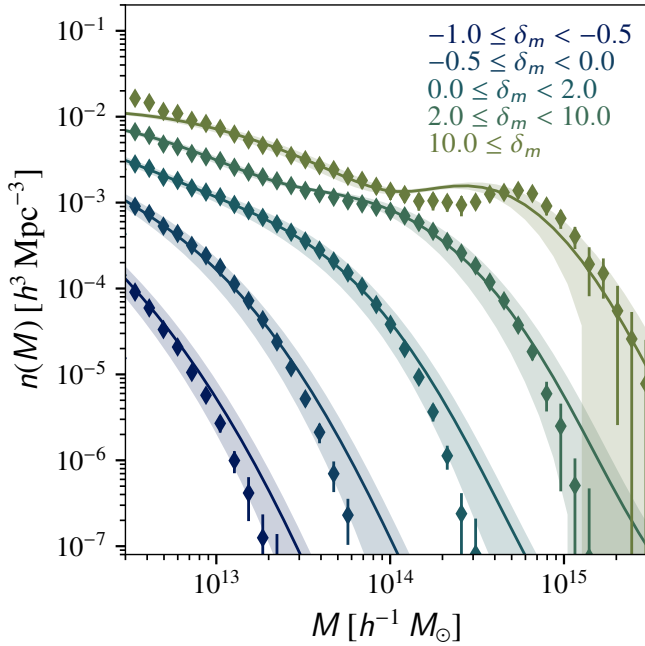


Fig. 5. Mean conditional halo mass functions predicted by the PineTree model in solid lines and the halo mass function estimated from N -body simulations depicted in diamonds. The different colours correspond to different underlying DM overdensity bins that are specified by the legend in the figure. Error bars correspond to the 1σ deviation and are presented as solid vertical lines or as shaded regions for the N -body estimates and PineTree predictions, respectively.

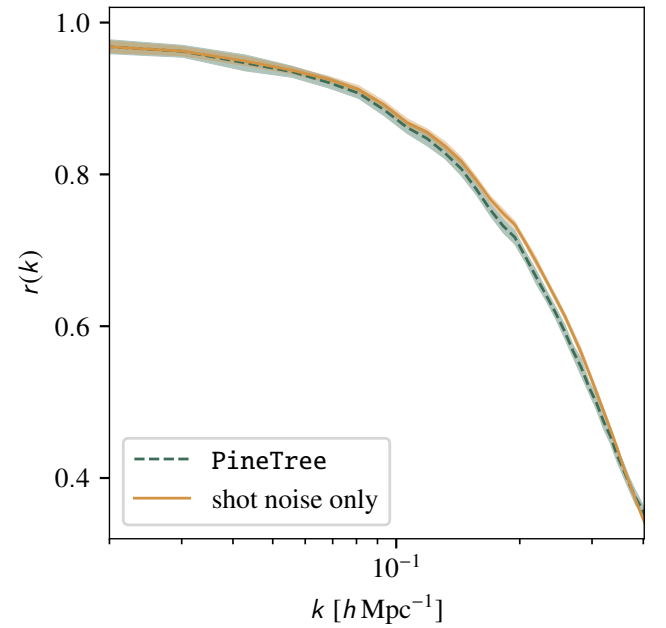


Fig. 6. Cross-correlation coefficient comparison for halos with masses between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$ at a $7.81 h^{-1}$ Mpc voxel resolution from 30 validation samples. The dashed green line shows the mean for the cross-correlation coefficient from the mock fields, while the mean of idealised fields with only Poisson noise contributions is depicted in solid orange. The shaded coloured area shows the 1σ region, respectively.

auto-correlation strength per k -bin. We want to further assess if the PineTree halos are realistic on the field level. Therefore, to complement the validation at the two-point correlation level, we computed the cross-correlation coefficient between each mock

and validation halo count field via

$$r_{\text{mr}}(k) = \frac{P_{\text{mr}}(k)}{\sqrt{P_{\text{mm}}(k)P_{\text{rr}}(k)}}, \quad (20)$$

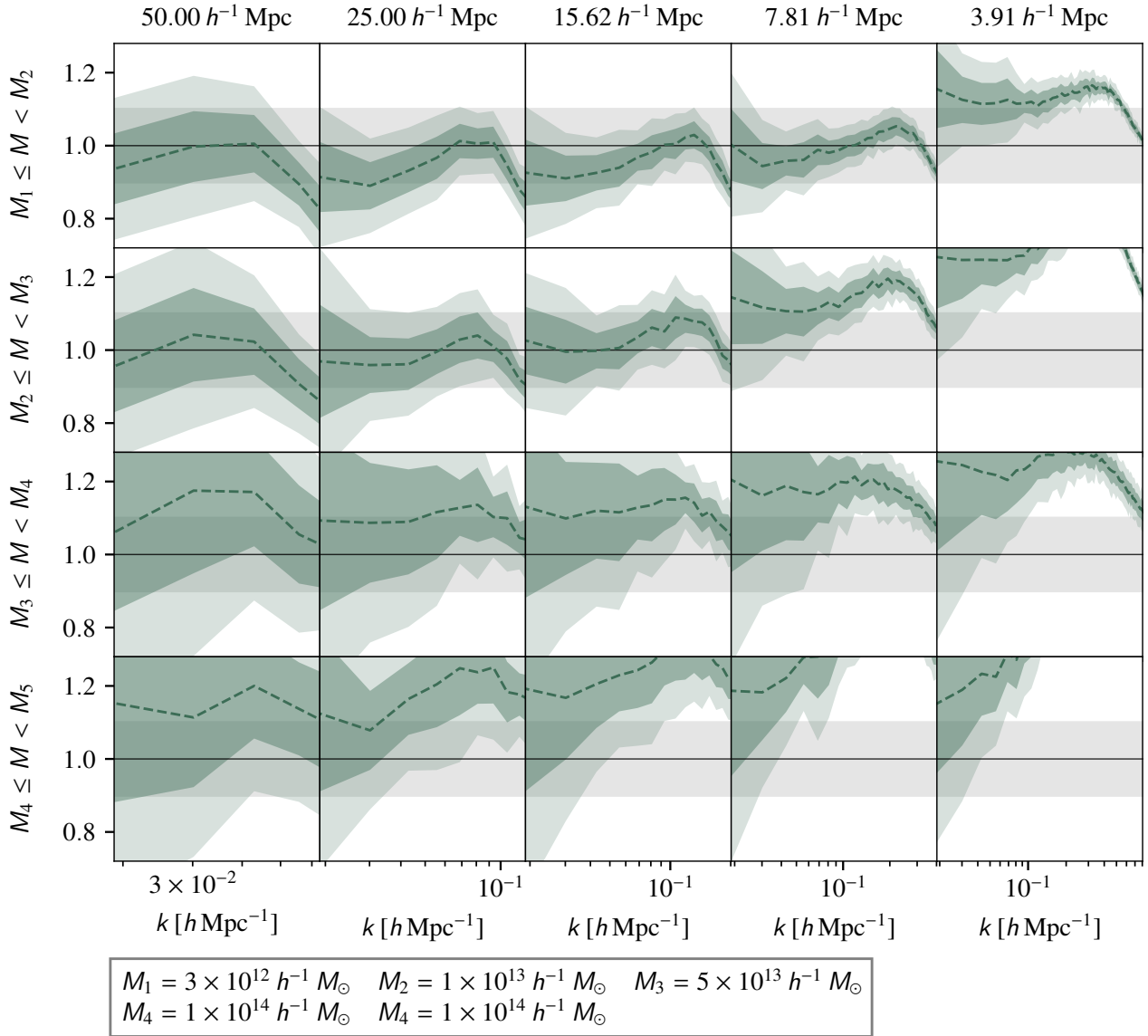


Fig. 7. Ratios between the power spectrum of the PineTree mocks and the validation catalogues for different mass bins and at voxel resolutions. For the subplots shown from left to right, the voxel size is decreasing according to the Table 1 and also indicated by the annotations in the top row. From top to bottom, each row shows the ratio for increasing mass bins. As with all previous figures, the quantities shown here were all computed using an ensemble of 30 fields.

with the indices ‘m’ and ‘r’ as labels for the mock and reference fields. In Fig. 6, we show the cross-correlation coefficient computed for the 30 mocks and validation samples and we compare it to the best-case scenario in which we assume that the decorrelation is only due to Poisson noise (also called shot noise). The cross-correlation coefficient with only shot noise contributions was obtained by assuming a given reference halo count field as ground truth and adding an uncorrelated Poisson noise, ε , as a scale-independent term given by

$$P_{\varepsilon}(k) = \frac{V}{\bar{N}}, \quad (21)$$

where V is the volume of the voxel and $\bar{N}$ is the mean number of halos from the reference halo catalogue. We can see that the actual cross-correlation coefficient traces the optimal case very closely, concluding that the positions of the predicted halos correspond very well with the assumed ground truth from the full simulation.

Figure 7 shows the validation of PineTree mocks at power spectrum level for the different resolutions from Table 1 and across four mass bins. We can see that the agreement with respect to the N -body ground truth deteriorates for higher mass bins as well as higher resolutions. There are likely two main reasons for this; namely, the initial Poisson and a conditionally independent voxel assumption. Using a Poisson likelihood has the advantage that it can be written down in an analytic fashion (see Eq. (17)), but as a consequence, the variance of the predicted halo distribution is always equal to the mean. However, studies by Casas-Miranda et al. (2002) indicate that for high-overdensity regions, and therefore especially for high masses, the Poisson assumption breaks down. The exact extent and impact of the non-Poisson contributions are not in the scope of this article and will be investigated in future works. The second assumption of voxel independence is most likely to break down for high-resolution set-ups. Because PineTree samples halos independently for each voxel, the clustering behaviour of halos

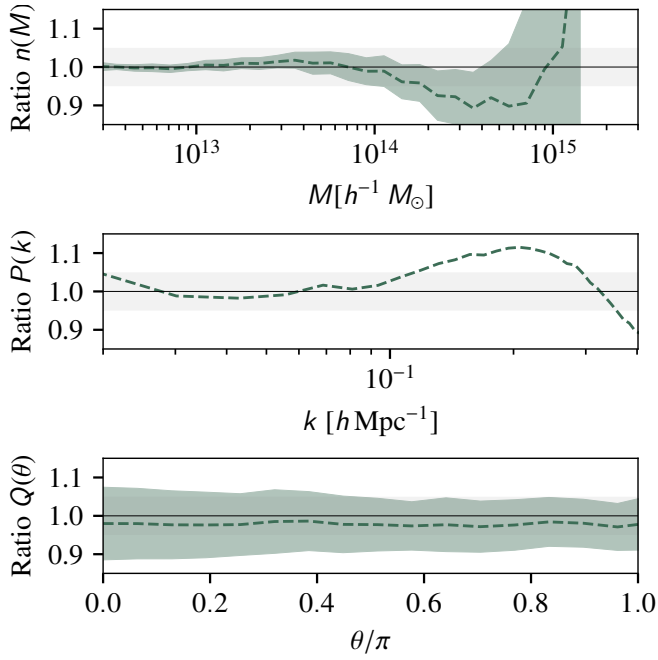


Fig. 8. Simplified version of Fig. 4 showing the ratios of the halo mass function, power spectrum, and reduced bispectrum but for input overdensity fields from a 2LPT gravity model. The respective quantities were computed for halos with a virial mass between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$ at a $7.81 h^{-1} \text{Mpc}$ voxel resolution.

on small scales is neglected. In other words, if a very massive halo is sampled in one voxel, the neighbouring voxels will not be aware of this and the sampling behaviour is not adjusted. This effect is then especially apparent at finer gridding, since nearby voxels will tend to have similar underlying DM overdensities. A more quantitative study on the impact of conditional dependencies with respect to grid resolution is planned and possibilities of addressing this issue will be discussed later in the conclusion. We see a similar deterioration in agreement for the cross-correlation coefficient and the reduced bispectrum.

5.2. Approximate gravity model

The results in the previous subsection were predictions based on L-Gadget overdensity fields. In practice, we would like to substitute the input fields with overdensity fields from approximate gravity solvers to save even more computational time. Figure 8 shows the results from PineTree retrained using the density fields produced by a 2LPT model (as is described in Sect. 4.1). The figure is a simplified version of Fig. 4 and shows only the ratios for the halo mass function, the power spectrum, and the reduced bispectrum with respect to the reference halo catalogue. We can report that the precision of the prediction is almost identical to the set-up with the L-Gadget overdensity fields, showing that PineTree is able to interpolate correctly from the approximate DM fields to the high-resolution N -body halos.

5.3. Redshift dependency

The variation discussed in this subsection is the application of PineTree on L-Gadget DM overdensities and halo catalogues at a redshift of $z = 1$ (as is described in Sect. 4.1). We find that

the retrained model performs very similar to the set-up at a redshift of $z = 0$ with notably only a slightly worse agreement for the reduced bispectrum. Therefore, we derive that PineTree can potentially be used to generate halo mocks at different redshifts. Future studies of the model are planned to incorporate the redshift as a conditional parameter. The results for this set-up are shown in Fig. B.2.

5.4. Model interpretation

As was mentioned in Sect. 3, the architecture of the network is interpretable and the trained weights can yield insights into the optimisation task itself. In particular, we can draw intuition from the symmetric convolutional kernel on the effective environment size needed as an input for the network to predict the halo distribution for a given voxel.

The kernel as described in Sect. 3.1.1 is based on a multipole expansion and more specifically all set-ups use only the monopole. Figure 9 visualises a slice through each of the three kernels with varying sizes according to Table 1. All of the depicted kernels are from training runs in which the fields are gridded with a voxel size of $7.81 h^{-1} \text{Mpc}$. We see that the absolute values of the kernel weights strongly decrease the further it is located from the centre of the kernel. Because the magnitude of each weight in a convolutional kernel indicates the relative importance of a given location of that patch, we can quantify the information content from an overdensity region needed to predict the subsequent halo properties. We report that the central kernel itself accounts for over 40% of the total kernel weight.

Since the loss function for PineTree is a likelihood, derived in Eq. (11), we can directly compare the set-ups with varying kernel sizes through the difference in log-likelihood computed on the validation set. Figure 10 depicts the comparison of the difference in the log-likelihood, $\Delta \log \mathcal{L}$, for four different kernel sizes with the $3 \times 3 \times 3$ kernel as baseline. As was expected, the local kernel performs the worst and there is a strong improvement when adopting the baseline model that takes neighbouring voxels into account. Increasing the kernel size to $5 \times 5 \times 5$ still yields a significant improvement in the log-likelihood, while adopting even larger kernels does not improve the mean difference in log-likelihood beyond the standard deviation computed on the validation samples. Figure 11 additionally visualises the performance across the N -point validation metrics. For the $9 \times 9 \times 9$ kernel, we observe over-fitting that would need to be mitigated via more training data or the early-stopping technique often utilised in the machine learning community, but this would digress from the goal of this work. Therefore, we conclude that PineTree performs best in our resolution and parameterisation study for fields gridded at a voxel size of $7.81 h^{-1} \text{Mpc}$ and with a kernel size of $7 \times 7 \times 7$.

Another notable finding lies within the central 3×3 patch of the kernel shown in Fig. 9. We can see that the signs between the central kernel weight and its adjacent weights are opposite to each other. Consequently, a given matter density voxel and its surrounding environment patch adjacent positive overdensity regions will lower the contribution to the feature vector, ψ . In contrast, voids with negative overdensity values will enhance the magnitude of the feature vector instead. Figure 12 shows that with increasing ψ the resulting conditional halo mass function shifts right towards high mass halos and also increases in amplitude. The result agrees with our physical intuition that extended overdensity regions yield less massive halos, while regions where the overdensity is accumulated into a more condensed

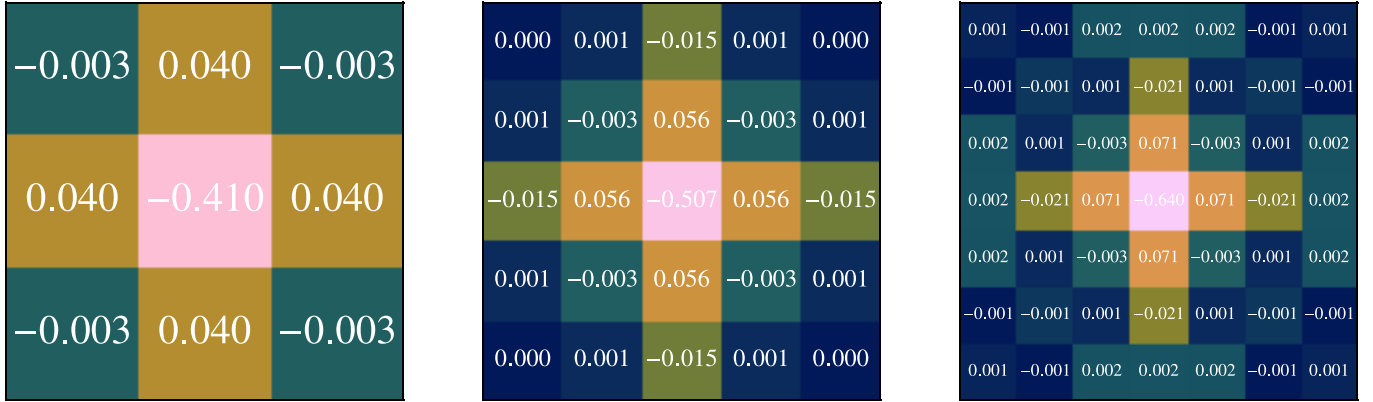


Fig. 9. Trained kernel weights from the central slice through the monopole kernel of PineTree. Each panel each from left to right shows a different kernel with sizes of 3^3 , 5^3 , and 7^3 , respectively. The weights from the kernels are from optimisation runs at a $7.81 h^{-1}$ Mpc voxel resolution and the underlying colour indicates the absolute value of each weight.

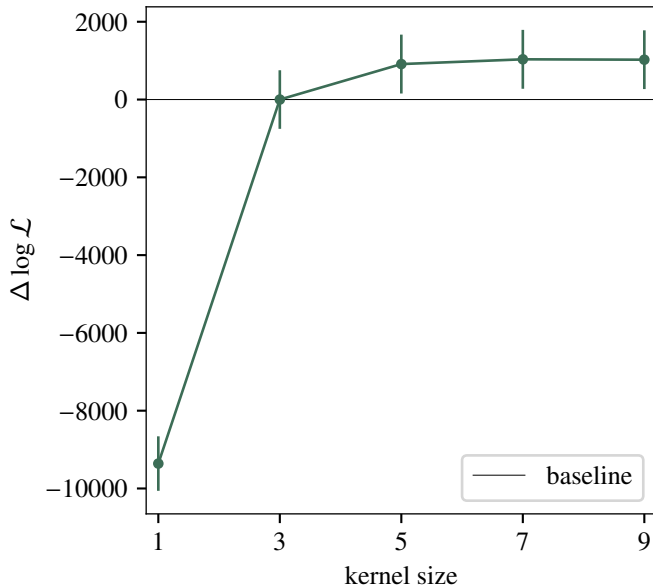


Fig. 10. Mean log-likelihood difference computed from the 30 validation simulations for PineTree with varying kernel sizes. The horizontal black line indicates the baseline model and represents the $3 \times 3 \times 3$ kernel set-up. Error bars for each point indicate the 1σ deviation obtained from the validation set.

region will give rise to the most massive halos in our universe (or simulation). Moreover, one can observe that the kernels in Fig. 9 appear to be axis-aligned rather than purely isotropic, which is due to the unphysical density estimation through the CIC kernel, showing that PineTree is able to adjust for the alignment artefacts even with its constraint kernel set-up. We show in Appendix B.3 that the kernel weights are distributed more isotropically when using the computationally more expensive smoothed-particle hydrodynamics (SPH) interpolation (first described by Lucy 1977; Gingold & Monaghan 1977).

5.5. Computational cost

One of the key differences between our modelling approach and conventional machine-learning methods is the significantly reduced number of parameters. This leads to a model that is computationally efficient to train, both in terms of the required number of training samples and in computational time. We

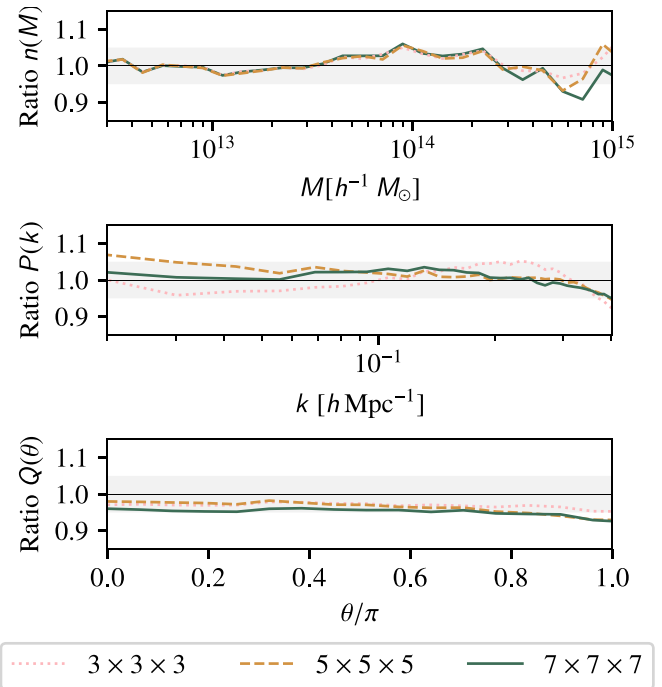


Fig. 11. Simplified version of Fig. 4 showing the halo mass function, power spectrum, and reduced bispectrum ratios but for different kernel sizes, as is specified in Table 1. The respective quantities were computed for halos with a virial mass between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$ at a $7.81 h^{-1}$ Mpc voxel resolution.

report as an example the number of training simulations needed to saturate the validation loss of PineTree for the 3^3 kernel set-up at a $7.81 h^{-1}$ Mpc voxel size resolution. The final mean validation log-likelihoods computed for the 30 validation samples after training for a given training sample size are listed in Table 2. Again, we have made use of the likelihood (see Eq. (11)) to assess the training run via the difference in log-likelihood with the baseline model using only one simulation for training. Already, the mean log-likelihood difference between one and five training simulations is significantly smaller than the standard deviation computed on the validation set and the difference becomes even more negligible when increasing the training set size. This shows that PineTree can generalise well

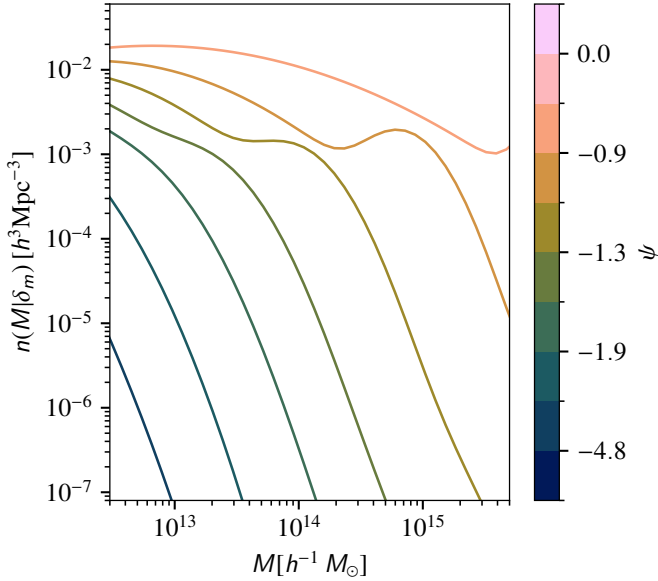


Fig. 12. Predicted conditional halo mass functions from PineTree for different values (indicated by the colour bar) of the feature map, ψ . The scalar is, as is described in Sect. 3.1.1, the output of the symmetric convolutional layer.

Table 2. Difference in log-likelihood depending on the number of training samples.

Number of training samples	$\mu(\Delta \log \mathcal{L})$	$\sigma(\Delta \log \mathcal{L})$
1	0	753.02
5	9.05	753.58
10	10.48	753.08
20	10.84	753.09
30	10.75	753.18

with a small number of training samples, and we were very conservative in our choice to train all our optimisation runs with the ten aforementioned training simulations.

We conclude this section by reporting the computational time for training as well as running the model to produce halo catalogues to give an idea of the computational speed of our model. The specific hardware used in this work was a 64-core AMD EPYC 7702 machine with 1 TB of RAM (hereafter referred to as the CPU machine) and a Nvidia V100 with 32 GB of RAM (denoted later on as the GPU machine). The training was exclusively executed on the GPU machine, and the training time mainly depended on the resolution on which the model was trained, as is reported in Table 3, as it determined the total number of voxels for which the likelihood needs to be evaluated. For instance, at $3.91 h^{-1} \text{ Mpc}$ voxel resolution, we needed to compute over the likelihood for 10×128^3 voxels, as opposed to the $50 h^{-1} \text{ Mpc}$ resolution run, where the training sample consisted only of 10×10^3 voxels. Due to the limited number of parameters, it would be possible to reduce the training time for high-resolution runs by only using sub-boxes of the training simulations, but we postpone a detailed study on optimising training time for PineTree for future work. Finally, we report the numbers for sampling halo catalogues using the trained model. The sampling for a $500 h^{-1} \text{ Mpc}$ simulation gridded at a resolution of 128^3 voxels took about 3.2 seconds for the CPU and 3.7 seconds on the GPU. This would result in an estimated cost of 100 GPU

Table 3. Training time in GPU hours for different gridding resolutions.

Number of GPU hours	Voxel size ($h^{-1} \text{ Mpc}$)
15	50
16	25
17	16
50	8
100	4

hours of training plus 63 000 CPU hours to produce 1000 realisations of $4 h^{-1} \text{ Gpc}$ sized mock halo simulations, which is over an order of magnitude faster than PINNOCHIO (1 024 000 CPU hours), EZmocks (822 000 CPU hours), or PATCHY (822 000 CPU hours) reported by Monaco (2016).

6. Conclusions

This paper has presented the computationally efficient neural network PineTree as a new approach to generating mock halo catalogues that are accurate on a percent level across various N -point correlation metrics. We built upon the work of Charnock et al. (2020), for which we further benchmarked and tested the limits of the model for a wide range of set-ups.

We performed an extensive resolution and parameter search to find that, in the current iteration, PineTree performs best at a voxel resolution of $7.81 h^{-1} \text{ Mpc}$ and for halo masses between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$. Here, we stress again that the metric utilised as the loss function during training is independent from the validation metrics; namely, the halo mass function, power spectrum, cross-correlation, and bispectrum. Therefore, this demonstrates the robustness of our model's predictions. Similarly, the prediction accuracy translates well into set-ups with snapshots at a redshift of $z = 1$ and also for input density fields computed with a 2LPT algorithm. Notably, the use of PineTree in conjunction with approximate density fields makes it attractive as an emulator for future works to generate large mock simulation suites that can then be utilised; for instance, to compute numerical covariances.

Subsequently, we discussed how the physics-informed model architecture facilitates interpretability. The symmetric convolutional kernel and the simple network set-up enable insights into the effective environment size needed to predict the halo masses optimally on a given gridding resolution. In addition, the kernel reveals how the voxels in a density environment patch affect the resulting prediction and how much information each cell holds. Furthermore, the physical Poisson likelihood employed as a loss function allows the seamless integration of PineTree as the bias model into a galaxy clustering inference pipeline. A natural follow-up study will leverage this alongside the model's reduced parameter space and differentiability within an inference with the algorithm BORG (Jasche & Wandelt 2013; Jasche et al. 2015; Lavaux & Jasche 2016; Jasche & Lavaux 2019; Lavaux et al. 2019).

Finally, we have shown in this work that PineTree is not only computationally efficient for sampling halo catalogues but also for training. This makes the pipeline easily adaptable to different model set-ups without the need for large training datasets.

We see that the model's predictions start to break down for halo masses greater than $1 \times 10^{14} h^{-1} M_{\odot}$ and high gridding resolutions, notably at $3.91 h^{-1} \text{ Mpc}$. The cause of this

breakdown is related to the assumption of the underlying halo distribution. First, DM halos follow a Poisson distribution, and second, each voxel is independently distributed. Future iterations of PineTree can address these issues through the replacement of the loss and sampling procedure by a distribution-agnostic neural density estimator, as was done by Pandey et al. (2023), or by incorporating a more flexible likelihood such as the generalised Poisson distribution (Consul & Jain 1973). To account for the halo conditional distribution across neighbouring voxels, a mass conservation scheme or multi-scale approach could be explored.

There are numerous potential applications of PineTree to be tested. Future extensions to this work could be the prediction of other halo properties like velocities or the direct generation of mock galaxy catalogues. Another interesting aspect is the conditioning of PineTree on cosmological parameters.

Acknowledgements. We thank Metin Ata, Deaglan Bartlett, Nai Boonkongkird, Ludvig Doerer, Matthew Ho, Axel Lapel, Lucas Makinen, Stuart McAlpine, and Stephen Stopyra for useful discussions related to this work. We also thank Deaglan Bartlett, Ludvig Doerer, and Stuart McAlpine for their helpful feedback to improve the manuscript. This work has made use of the Infinity cluster hosted by Institut d’Astrophysique de Paris. We thank the efforts of S. Rouberol for running the cluster smoothly. This work was granted access to the HPC resources of Joliot Curie/TGCC (Très Grand Centre de Calcul) under the allocation AD010413589R1. This work was enabled by the research project grant ‘Understanding the Dynamic Universe’ funded by the Knut and Alice Wallenberg Foundation under Dnr KAW 2018.0067. JJ acknowledges support from the Swedish Research Council (VR) under the project 2020-05143 – ‘Deciphering the Dynamics of Cosmic Structure’. GL and SD acknowledge the grant GCEuclid from ‘Centre National d’Etudes Spatiales’ (CNES). This work was supported by the Simons Collaboration on ‘Learning the Universe’. This work is conducted within the Aquila Consortium (<https://aquila-consortium.org>).

References

- Anagnostidis, S., Thomsen, A., Kacprzak, T., et al. 2022, arXiv e-prints [arXiv:2211.12346]
- Aubert, D., Pichon, C., & Colombi, S. 2004, *MNRAS*, **352**, 376
- Audren, B., & Lesgourgues, J. 2011, *J. Cosmol. Astropart. Phys.*, **2011**, 037
- Avila, S., Murray, S. G., Knebe, A., et al. 2015, *MNRAS*, **450**, 1856
- Balaguera-Antolínez, A., Kitaura, F.-S., Pellejero-Ibáñez, M., Zhao, C., & Abel, T. 2018, *MNRAS*, **483**, L58
- Balaguera-Antolínez, A., Kitaura, F.-S., Pellejero-Ibáñez, M., et al. 2019, *MNRAS*, **491**, 2565
- Balaguera-Antolínez, A., Kitaura, F.-S., Alam, S., et al. 2023, *A&A*, **673**, A130
- Baldauf, T., Seljak, U. c. v., Smith, R. E., Hamaus, N., & Desjacques, V. 2013, *Phys. Rev. D*, **88**, 083507
- Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, *ApJ*, **304**, 15
- Bartlett, D. J., Ho, M., & Wandelt, B. D. 2024, *ApJ*, submitted [arXiv:2405.00635]
- Behroozi, P. S., Conroy, C., & Wechsler, R. H. 2010, *ApJ*, **717**, 379
- Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013, *ApJ*, **762**, 109
- Berger, P., & Stein, G. 2018, *MNRAS*, **482**, 2861
- Beyond-2pt Collaboration (Kause, E., et al.) 2024, arXiv e-prints [arXiv:2405.02252]
- Blas, D., Lesgourgues, J., & Tram, T. 2011, *J. Cosmol. Astropart. Phys.*, **2011**, 034
- Blot, L., Corasaniti, P. S., Amendola, L., & Kitching, T. D. 2016, *MNRAS*, **458**, 4462
- Blot, L., Crocce, M., Sefusatti, E., et al. 2019, *MNRAS*, **485**, 2806
- Bond, J. R. 1987, in *Cosmology and Particle Physics*, ed. I. Hinchliffe, 22
- Bond, J. R., & Myers, S. T. 1996, *ApJS*, **103**, 1
- Bond, J. R., Szalay, A. S., & Silk, J. 1988, *ApJ*, **324**, 627
- Bond, J. R., Cole, S., Efstathiou, G., & Kaiser, N. 1991, *ApJ*, **379**, 440
- Boonkongkird, C., Lavaux, G., Peirani, S., et al. 2023, arXiv e-prints [arXiv:2303.17939]
- Braun, R., Bonaldi, A., Bourke, T., Keane, E., & Wagg, J. 2019, arXiv e-prints [arXiv:1912.12699]
- Bryan, G. L., & Norman, M. L. 1998, *ApJ*, **495**, 80
- Carlberg, R. G., & Couchman, H. M. P. 1989, *ApJ*, **340**, 47
- Casas-Miranda, R., Mo, H. J., Sheth, R. K., & Boerner, G. 2002, *MNRAS*, **333**, 730
- Cen, R., & Ostriker, J. P. 1993, *ApJ*, **417**, 415
- Chan, K. C., Scoccimarro, R., & Sheth, R. K. 2012, *Phys. Rev. D*, **85**, 083509
- Charnock, T., Lavaux, G., Wandelt, B. D., et al. 2020, *MNRAS*, **494**, 50
- Chartier, N., & Wandelt, B. D. 2021, *MNRAS*, **509**, 2220
- Chartier, N., Wandelt, B., Akrami, Y., & Villaescusa-Navarro, F. 2021, *MNRAS*, **503**, 1897
- Chaves-Montero, J., Angulo, R. E., Schaye, J., et al. 2016, *MNRAS*, **460**, 3100
- Chuang, C.-H., Kitaura, F.-S., Prada, F., Zhao, C., & Yepes, G. 2014, *MNRAS*, **446**, 2621
- Chuang, C.-H., Zhao, C., Prada, F., et al. 2015, *MNRAS*, **452**, 686
- Cole, S., & Kaiser, N. 1988, *MNRAS*, **233**, 637
- Cole, S., & Kaiser, N. 1989, *MNRAS*, **237**, 1127
- Colombi, S. 2013, AdaptaHOP: Subclump Finder, Astrophysics Source Code Library [record ascl:1305.004]
- Colombi, S., Chodorowski, M. J., & Teyssier, R. 2007, *MNRAS*, **375**, 348
- Consul, P. C., & Jain, G. C. 1973, *Technometrics*, **15**, 791
- Contarini, S., Pisani, A., Hamaus, N., et al. 2023, *ApJ*, **953**, 46
- Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, *ApJ*, **292**, 371
- de la Torre, S., & Peacock, J. A. 2013, *MNRAS*, **435**, 743
- de Santi, N. S. M., Shao, H., Villaescusa-Navarro, F., et al. 2023, *ApJ*, **952**, 69
- Dekel, A., & Lahav, O. 1999, *ApJ*, **520**, 24
- DESI Collaboration (Aghamousa, A., et al.) 2016, arXiv e-prints [arXiv:1611.00036]
- Desjacques, V., Jeong, D., & Schmidt, F. 2018, *Phys. Rep.*, **733**, 1
- Ding, J., Li, S., Zheng, Y., et al. 2024, *ApJS*, **270**, 25
- Doerer, L., Jamieson, D., Stopyra, S., et al. 2023, arXiv e-prints [arXiv:2312.09271]
- Doré, O., Bock, J., Ashby, M., et al. 2014, arXiv e-prints [arXiv:1412.4872]
- Drinkwater, M. J., Jurek, R. J., Blake, C., et al. 2010, *MNRAS*, **401**, 1429
- Efstathiou, G., & Rees, M. J. 1988, *MNRAS*, **230**, 5
- Efstathiou, G., Frenk, C. S., White, S. D. M., & Davis, M. 1988, *MNRAS*, **235**, 715
- Eisenstein, D. J., & Hut, P. 1998, *ApJ*, **498**, 137
- Euclid Collaboration (Castander, F. J., et al.) 2024a, A&A, submitted
- Euclid Collaboration (Mellier, Y., et al.) 2024b, A&A, in press, <https://doi.org/10.1051/0004-6361/202450810>
- Feng, Y., Chu, M.-Y., Seljak, U., & McDonald, P. 2016, *MNRAS*, **463**, 2273
- Ferrero, I., Crocce, M., Tutusaus, I., et al. 2021, *A&A*, **656**, A106
- Fisher, J. D., & Faltenbacher, A. 2017, *MNRAS*, **473**, 3941
- Fluri, J., Kacprzak, T., Lucchi, A., et al. 2019, *Phys. Rev. D*, **100**, 063514
- Fry, J. N., & Gaztanaga, E. 1993, *ApJ*, **413**, 447
- García-Farieta, J. E., Balaguera-Antolínez, A., & Kitaura, F.-S. 2024, arXiv e-prints [arXiv:2405.10319]
- Garrison, L. H., Eisenstein, D. J., Ferrer, D., et al. 2018, *ApJS*, **236**, 43
- Gingold, R. A., & Monaghan, J. J. 1977, *MNRAS*, **181**, 375
- Green, J., Schechter, P., Baltay, C., et al. 2012, arXiv e-prints [arXiv:1208.4012]
- Hahn, C., Eickenberg, M., Ho, S., et al. 2023, *PNAS*, **120**, e2218810120
- Hamaus, N., Seljak, U. c. v., Desjacques, V., Smith, R. E., & Baldauf, T. 2010, *Phys. Rev. D*, **82**, 043515
- Hamaus, N., Pisani, A., Choi, J.-A., et al. 2020, *J. Cosmology Astropart. Phys.*, **2020**, 023
- Hermans, J., Delaunoy, A., Rozet, F., et al. 2021, arXiv e-prints [arXiv:2110.06581]
- Ho, M., Bartlett, D. J., Chartier, N., et al. 2024, Open J. Astrophys., accepted [arXiv:2402.05137]
- Hockney, R. W., & Eastwood, J. W. 1981, *Computer Simulation Using Particles*
- Hou, J., Dizgah, A. M., Hahn, C., et al. 2024, *Phys. Rev. D*, **109**, 103528
- Howlett, C., Manera, M., & Percival, W. 2015a, *Astron. Comput.*, **12**, 109
- Howlett, C., Ross, A. J., Samushia, L., Percival, W. J., & Manera, M. 2015b, *MNRAS*, **449**, 848
- Ishiyama, T., Prada, F., Klypin, A. A., et al. 2021, *MNRAS*, **506**, 4210
- Izard, A., Crocce, M., & Fosalba, P. 2016, *MNRAS*, **459**, 2327
- Jamieson, D., Li, Y., de Oliveira, R. A., et al. 2023, *ApJ*, **952**, 145
- Jasche, J., & Wandelt, B. D. 2013, *MNRAS*, **432**, 894
- Jasche, J., & Lavaux, G. 2019, *A&A*, **625**, A64
- Jasche, J., Leclercq, F., & Wandelt, B. D. 2015, *J. Cosmology Astropart. Phys.*, **2015**, 036
- Jindal, V., Liang, A., Singh, A., Ho, S., & Jamieson, D. 2023, arXiv e-prints [arXiv:2303.13056]
- Jing, Y. P., Mo, H. J., & Börner, G. 1998, *ApJ*, **494**, 1
- Kaiser, N. 1984, *ApJ*, **284**, L9
- Kaiser, N. 1988, *Symp. Int. Astron. Union*, **130**, 43
- Kazin, E. A., Koda, J., Blake, C., et al. 2014, *MNRAS*, **441**, 3524
- Kingma, D. P., & Ba, J. 2014, arXiv e-prints [arXiv:1412.6980]
- Kitaura, F.-S., Yepes, G., & Prada, F. 2013, *MNRAS*, **439**, L21
- Kitaura, F.-S., Gil-Marín, H., Scóccola, C. G., et al. 2015, *MNRAS*, **450**, 1836
- Kitaura, F.-S., Rodríguez-Torres, S., Chuang, C.-H., et al. 2016, *MNRAS*, **456**, 4156

- Kitauro, F.-S., Balaguera-Antolínez, A., Sinigaglia, F., & Pellejero-Ibáñez, M. 2022, *MNRAS*, **512**, 2245
- Knollmann, S. R., & Knebe, A. 2009, *ApJS*, **182**, 608
- Koda, J., Blake, C., Beutler, F., Kazin, E., & Marin, F. 2016, *MNRAS*, **459**, 2118
- Kravtsov, A. V., & Klypin, A. A. 1999, *ApJ*, **520**, 437
- Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv e-prints [arXiv:1110.3193]
- Lavaux, G., & Jasche, J. 2016, *MNRAS*, **455**, 3169
- Lavaux, G., Jasche, J., & Leclercq, F. 2019, *MNRAS*, submitted [arXiv:1909.06396]
- Leclercq, F., Faure, B., Lavaux, G., et al. 2020, *A&A*, **639**, A91
- LeCun, Y., Boser, B., Denker, J. S., et al. 1989, *Neural Computat.*, **1**, 541
- Legin, R., Ho, M., Lemos, P., et al. 2024, *MNRAS*, **527**, L173
- Lehmann, B. V., Mao, Y.-Y., Becker, M. R., Skillman, S. W., & Wechsler, R. H. 2017, *ApJ*, **834**, 37
- Lesgourgues, J. 2011, arXiv e-prints [arXiv:1104.2932]
- LSST Science Collaboration (Abell, P. A., et al.) 2009, arXiv e-prints [arXiv:0912.0201]
- Lucy, L. B. 1977, *AJ*, **82**, 1013
- Makinen, T. L., Charnock, T., Lemos, P., et al. 2022, *Open J. Astrophys.*, **5**, 18
- Manera, M., Scoccimarro, R., Percival, W. J., et al. 2012, *MNRAS*, **428**, 1036
- Manera, M., Samushia, L., Tojeiro, R., et al. 2014, *MNRAS*, **447**, 437
- Matsubara, T. 1999, *ApJ*, **525**, 543
- Merz, H., Pen, U.-L., & Trac, H. 2005, *New Astron.*, **10**, 393
- Mo, H. J., & White, S. D. M. 1996, *MNRAS*, **282**, 347
- Modi, C., Feng, Y., & Seljak, U. 2018, *J. Cosmology Astropart. Phys.*, **2018**, 028
- Monaco, P. 2016, *Galaxies*, **4**, 53
- Monaco, P., Theuns, T., & Taffoni, G. 2002a, *MNRAS*, **331**, 587
- Monaco, P., Theuns, T., Taffoni, G., et al. 2002b, *ApJ*, **564**, 8
- Moster, B. P., Somerville, R. S., Maulbetsch, C., et al. 2010, *ApJ*, **710**, 903
- Narayan, R., & White, S. D. M. 1988, *MNRAS*, **231**, 97
- Nelson, D., Springel, V., Pillepich, A., et al. 2019, *CompAC*, **6**, 1
- Neyrinck, M. C., Hamilton, A. J. S., & Gnedin, N. Y. 2005, *MNRAS*, **362**, 337
- Neyrinck, M. C., Aragón-Calvo, M. A., Jeong, D., & Wang, X. 2014, *MNRAS*, **441**, 646
- Nguyen, N.-M., Schmidt, F., Lavaux, G., & Jasche, J. 2021, *J. Cosmology Astropart. Phys.*, **2021**, 058
- Nguyen, N.-M., Schmidt, F., Tucci, B., Reinecke, M., & Kostić, A. 2024, arXiv e-prints [arXiv:2403.03220]
- Nishimichi, T., Shirata, A., Taruya, A., et al. 2009, *PASJ*, **61**, 321
- Nishimichi, T., Taruya, A., Koyama, K., & Sabiu, C. 2010, *J. Cosmology Astropart. Phys.*, **2010**, 002
- Paillas, E., Cuesta-Lazaro, C., Zarrouk, P., et al. 2023, *MNRAS*, **522**, 606
- Pandey, S., Modi, C., Wandelt, B. D., & Lavaux, G. 2023, in *NeurIPS 2023 AI for Science Workshop*
- Peacock, J. A., & Smith, R. E. 2000, *MNRAS*, **318**, 1144
- Pellejero Ibáñez, M., Angulo, R. E., Jamieson, D., & Li, Y. 2024, *MNRAS*, **529**, 89
- Pen, U.-L. 1998, *ApJ*, **504**, 601
- Percival, W. J., Ross, A. J., Sánchez, A. G., et al. 2014, *MNRAS*, **439**, 2531
- Perez, L. A., Genel, S., Villaescusa-Navarro, F., et al. 2023, *ApJ*, **954**, 11
- Piras, D., Joachimi, B., & Villaescusa-Navarro, F. 2023, *MNRAS*, **520**, 668
- Planck Collaboration VI. 2020, *A&A*, **641**, A6
- Planelles, S., & Quilis, V. 2010, *A&A*, **519**, A94
- Porqueres, N., Heavens, A., Mortlock, D., Lavaux, G., & Makinen, T. L. 2023, arXiv e-prints [arXiv:2304.04785]
- Press, W. H., & Schechter, P. 1974, *ApJ*, **187**, 425
- Ravanbakhsh, S., Oliva, J., Fromenteau, S., et al. 2017, arXiv e-prints [arXiv:1711.02033]
- Reddick, R. M., Wechsler, R. H., Tinker, J. L., & Behroozi, P. S. 2013, *ApJ*, **771**, 30
- Régalo-Saint Blancard, B., Hahn, C., Ho, S., et al. 2024, *Phys. Rev. D*, **109**, 083535
- Repp, A., & Szapudi, I. 2020, *MNRAS*, **493**, 3449
- Schaeffer, R., & Silk, J. 1985, *ApJ*, **292**, 319
- Schaeffer, R., & Silk, J. 1988, *ApJ*, **332**, 1
- Schmidt, F., Elsner, F., Jasche, J., Nguyen, N. M., & Lavaux, G. 2019, *J. Cosmology Astropart. Phys.*, **2019**, 042
- Scoccimarro, R., & Sheth, R. K. 2002, *MNRAS*, **329**, 629
- Scott, D. W. 2015, *Multivariate Density Estimation: Theory, Practice, and Visualization* (John Wiley & Sons)
- Seljak, U. 2000, *MNRAS*, **318**, 203
- Seljak, U., & Warren, M. S. 2004, *MNRAS*, **355**, 129
- Seljak, U., Aslanyan, G., Feng, Y., & Modi, C. 2017, *J. Cosmology Astropart. Phys.*, **2017**, 009
- Sheth, R. K., & Lemson, G. 1999, *MNRAS*, **304**, 767
- Sheth, R. K., & Tormen, G. 1999, *MNRAS*, **308**, 119
- Sheth, R. K., & Tormen, G. 2002, *MNRAS*, **329**, 61
- Sheth, R. K., Mo, H. J., & Tormen, G. 2001, *MNRAS*, **323**, 1
- Sheth, R. K., Chan, K. C., & Scoccimarro, R. 2013, *Phys. Rev. D*, **87**, 083002
- Smith, R. E., Scoccimarro, R., & Sheth, R. K. 2007, *Phys. Rev. D*, **75**, 063512
- Springel, V. 2005, *MNRAS*, **364**, 1105
- Springel, V., White, S. D. M., Tormen, G., & Kauffmann, G. 2001, *MNRAS*, **328**, 726
- Springel, V., Pakmor, R., Zier, O., & Reinecke, M. 2021, *MNRAS*, **506**, 2871
- Stadel, J. G. 2001, *Cosmological N-body Simulations and Their Analysis* (University of Washington)
- Stiskalek, R., Desmond, H., Holvey, T., & Jones, M. G. 2021, *MNRAS*, **506**, 3205
- Taffoni, G., Monaco, P., & Theuns, T. 2002, *MNRAS*, **333**, 623
- Takada, M., Ellis, R. S., Chiba, M., et al. 2014, *PASJ*, **66**, R1
- Taruya, A., & Soda, J. 1999, *ApJ*, **522**, 46
- Tassev, S., Zaldarriaga, M., & Eisenstein, D. J. 2013, *J. Cosmology Astropart. Phys.*, **2013**, 036
- Tassev, S., Eisenstein, D. J., Wandelt, B. D., & Zaldarriaga, M. 2015, arXiv e-prints [arXiv:1502.07751]
- Tegmark, M., & Bromley, B. C. 1999, *ApJ*, **518**, L69
- Teyssier, R. 2002, *A&A*, **385**, 337
- Tinker, J., Kravtsov, A. V., Klypin, A., et al. 2008, *ApJ*, **688**, 709
- Vakili, M., Kitauro, F.-S., Feng, Y., et al. 2017, *MNRAS*, **472**, 4144
- Valageas, P., & Nishimichi, T. 2011, *A&A*, **527**, A87
- Vale, A., & Ostriker, J. P. 2004, *MNRAS*, **353**, 189
- Valogiannis, G., Yuan, S., & Dvorkin, C. 2024, *Phys. Rev. D*, **109**, 103503
- van Uitert, Edo, Cacciato, Marcello, Hoekstra, Henk, & Herbonnet, Ricardo. 2015, *A&A*, **579**, A26
- Villaescusa-Navarro, F. 2018, Pylans: Python libraries for the analysis of numerical simulations, Astrophysics Source Code Library [record ascl:1811.008]
- Villaescusa-Navarro, F., Hahn, C., Massara, E., et al. 2020, *ApJS*, **250**, 2
- Villaescusa-Navarro, F., Genel, S., Angles-Alcazar, D., et al. 2021, arXiv e-prints [arXiv:2109.10360]
- Villanueva-Domingo, P., & Villaescusa-Navarro, F. 2022, *ApJ*, **937**, 115
- Wechsler, R. H., DeRose, J., Busha, M. T., et al. 2022, *ApJ*, **931**, 145
- Weltman, A., Bull, P., Camera, S., et al. 2020, *PASA*, **37**, e002
- White, M., Tinker, J. L., & McBride, C. K. 2013, *MNRAS*, **437**, 2594
- Yang, X., Mo, H. J., & van den Bosch, F. C. 2003, *MNRAS*, **339**, 1057
- Yang, X., Mo, H. J., & van den Bosch, F. C. 2008, *ApJ*, **676**, 248
- Yang, X., Mo, H. J., & van den Bosch, F. C. 2009, *ApJ*, **695**, 900
- Yang, X., Zhang, Y., Lu, T., et al. 2017, *ApJ*, **848**, 60
- Yuan, S., Abel, T., & Wechsler, R. H. 2023, *MNRAS*, **527**, 1993
- Yung, L. Y. A., Somerville, R. S., Ferguson, H. C., et al. 2022a, *MNRAS*, **515**, 5416
- Yung, L. Y. A., Somerville, R. S., Finkelstein, S. L., et al. 2022b, *MNRAS*, **519**, 1578
- Zehavi, I., Zheng, Z., Weinberg, D. H., et al. 2011, *ApJ*, **736**, 59
- Zhang, X., Wang, Y., Zhang, W., et al. 2019, arXiv e-prints [arXiv:1902.05965]
- Zheng, Z., Berlind, A. A., Weinberg, D. H., et al. 2005, *ApJ*, **633**, 791
- Zhou, A. J., & Dodelson, S. 2023, *Phys. Rev. D*, **108**, 083506

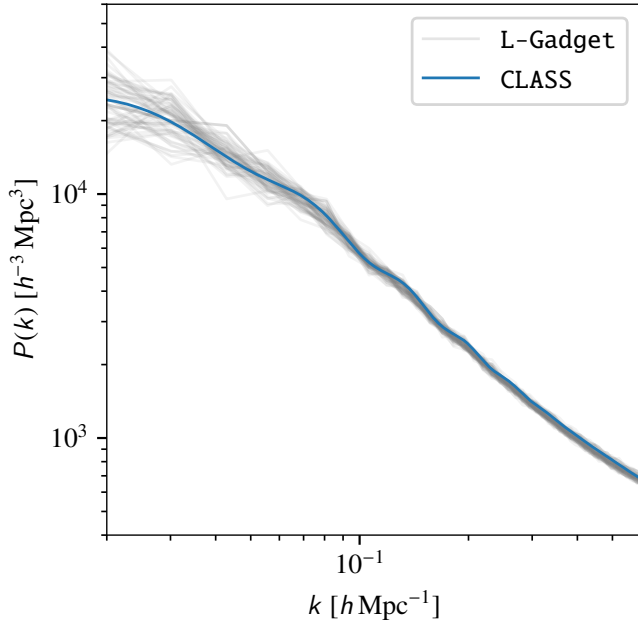


Fig. A.1. Matter power spectrum computed from by the Boltzmann solver CLASS with non-linear corrections at redshift $z = 0$ depicted by the solid blue line and the matter power spectra from the N -body simulations obtained from L-Gadget in light grey. Each grey line corresponds to one realisation from the simulation.

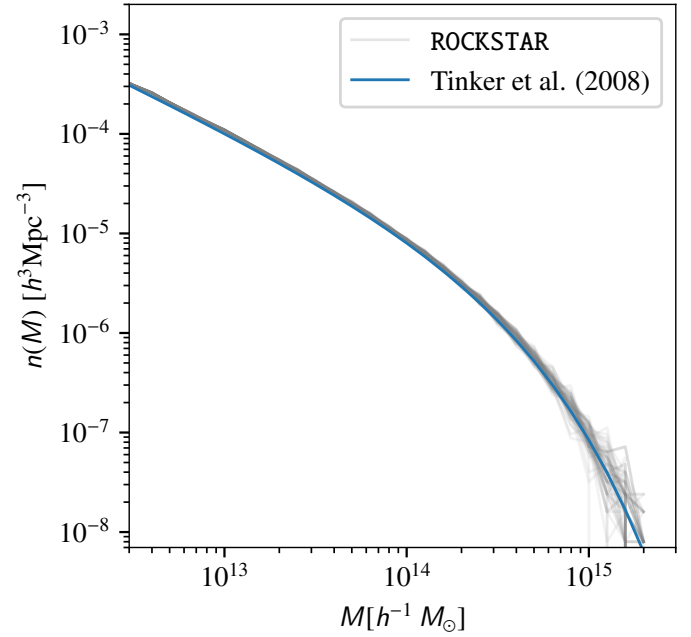


Fig. A.2. Halo mass function comparison between Tinker et al. (2008, in solid blue) and the N -body simulations (in light grey) produced by L-Gadget and ROCKSTAR as halo finder. The halo mass function is for halos adopting the virial mass definition (Bryan & Norman 1998) at redshift $z = 0$. Each grey line corresponds to one realisation from the simulation.

Appendix A: Simulation validation

As described in Sect. 4.1, we compare the matter power spectrum and halo mass function with theoretical predictions to validate the simulations we generated with L-Gadget and ROCKSTAR. Figure A.1 shows that the matter power spectra of the 60 simulations match the non-linear theoretical prediction computed by CLASS (as described in Audren & Lesgourgues 2011) very well for the Planck 2018 cosmological parameters (Planck Collaboration VI 2020). Similarly, the halo mass function reported by Tinker et al. (2008) is in good agreement with the halo mass functions from the final ROCKSTAR halo catalogues as shown in Fig. A.2. Therefore, we conclude that the simulation runs completed as expected and do not contain inherent errors that could bias our results in Sect. 5 obtained from the optimisations with PineTree.

Appendix B: Additional PineTree optimisation runs

B.1. Mixture component testing

Following the description of Sect. 4.2, we ran additional optimisations for different numbers of mixture components N_{mix} in PineTree namely two, three, and four. Since the loss function is derived from a proper probability density (see Sect. 3.2), we use the difference in the log-likelihood values computed on the validation simulations as the model comparison metric. Figure B.1 depicts the log-likelihood difference $\Delta \log \mathcal{L}$ for the three different runs with respect to the baseline model with two mixture components as used throughout Sect. 5. We optimised the different model set-ups on snapshots at redshift $z = 0$ and field discretisation at a voxel resolution of $7.81 h^{-1} \text{ Mpc}$. We report that even though the models with more mixture components slightly improve the log-likelihood compared to the baseline model, the improvements are not significant as the mean difference in log-

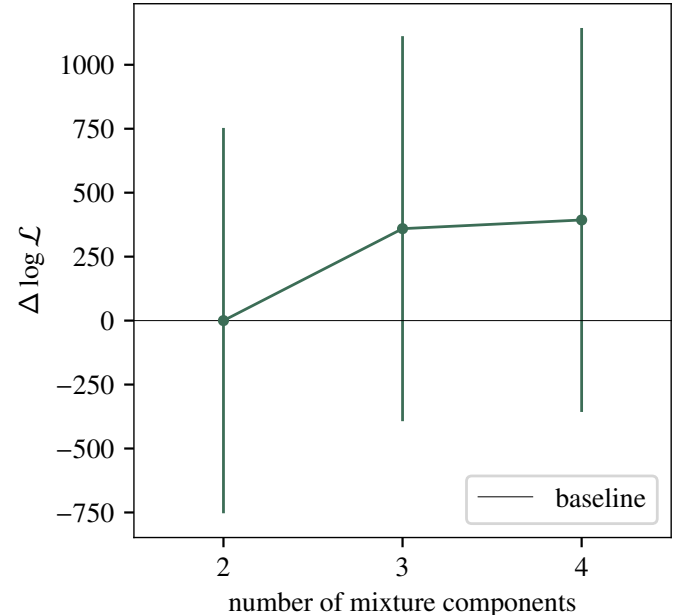


Fig. B.1. Mean difference in log-likelihood computed from the 30 validation simulations for different numbers of mixture components. The black horizontal line indicates the baseline model and represents the PineTree model with two mixture components. Error bars for each point indicate the 1σ deviation obtained from the validation set.

likelihood is still within the 1σ standard deviation of the baseline. Hence, we limited our main resolution and parameter study to PineTree configurations with two mixture components.

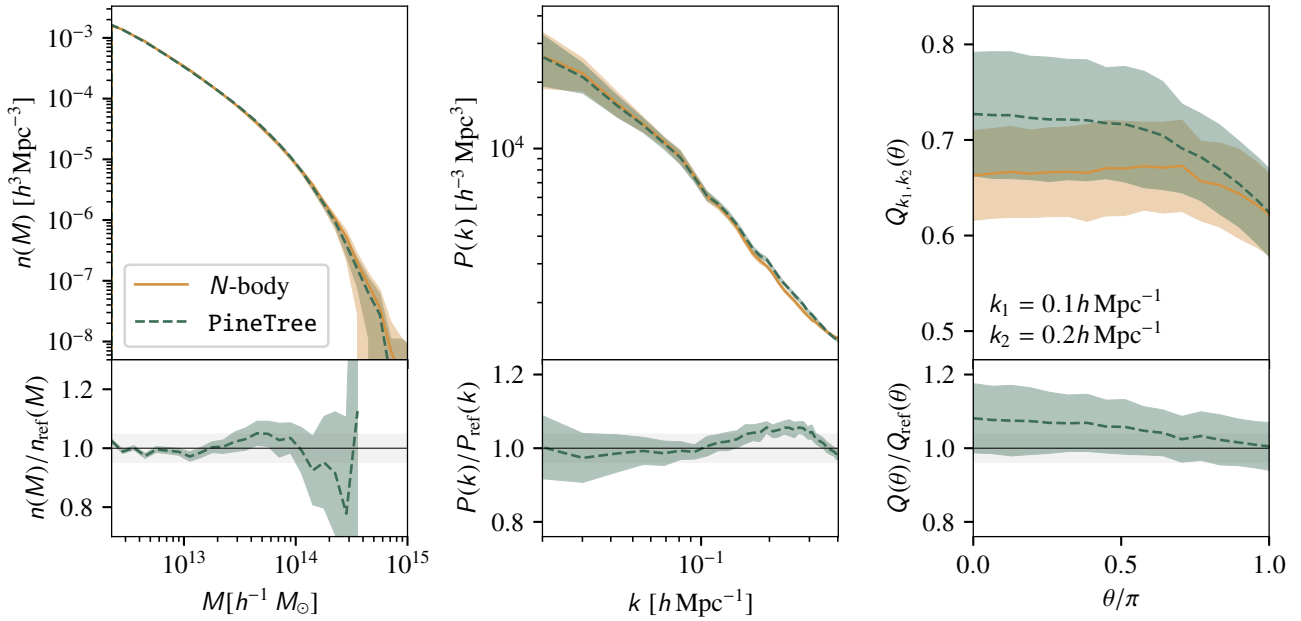


Fig. B.2. Same as Fig. 4 but with halos at $z = 1$. Again, the panels from left to right show the marginal halo mass distribution, the power spectrum and the reduced bispectrum with the network mocks plotted in green while the reference from the N -body simulations are shown in orange. The bottom subplot of each panel depicts the ratio of the respective summary statistics. The power spectrum and reduced bispectrum are computed using halos with a virial mass between $3 \times 10^{12} h^{-1} M_{\odot}$ and $1 \times 10^{13} h^{-1} M_{\odot}$.

B.2. Redshift $z = 1$ results

Following the results of Sect. 5.1, we optimise PineTree with snapshots at redshift $z = 1$ at a voxel resolution of $7.81 h^{-1} \text{ Mpc}$. Figure B.2 shows the same three key validation metrics after training. Comparing it to Fig. 4, the run at redshift $z = 0$, only the prediction for the reduced bispectrum is noticeably worse.

B.3. Density fields from smoothed-particle hydrodynamics

As pointed out in Sect. 5.4, the optimisation with the overdensity fields computed via CIC leads to kernels that show axis-aligned effects (see Fig. 9). To validate that this observed behaviour is not due to the model architecture itself, we retrain the same PineTree model as described in Sect. 5.1 at $7.81 h^{-1} \text{ Mpc}$ voxel resolution, but with overdensity fields from L-Gadget estimated by the SPH algorithm. We applied the SPH implementation developed and used by Colombi et al. (2007). Since the SPH interpolation scheme preserves the isotropic properties of the underlying particle distribution better than the CIC approach, we expect the change in the mass assignment scheme to be reflected by the resulting convolutional kernel. Figure B.3 shows the trained kernel weights for the central slice. Comparing them to Fig. 9, we notice that the kernel weights appear more isotropic. Hence, we conclude that the axis-aligned weights as shown in Fig. 9 are caused by non-isotropic artefacts from the CIC interpolation.

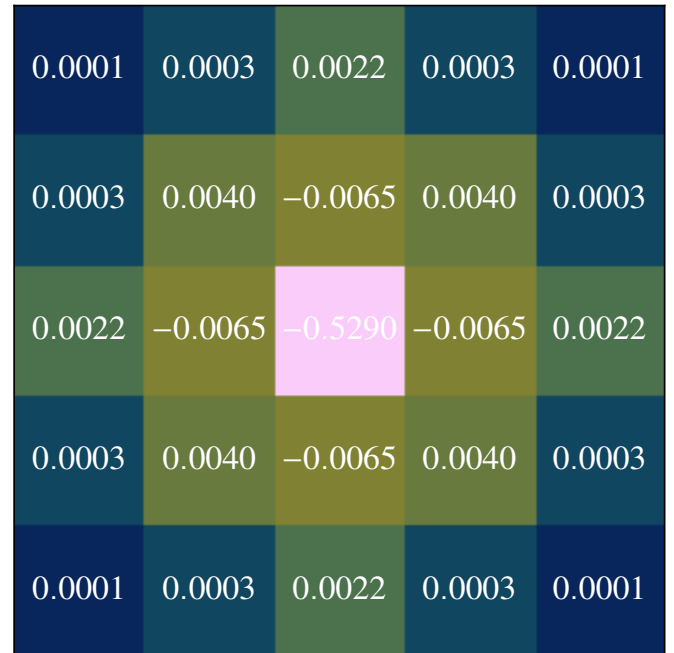


Fig. B.3. Central slice through a $5 \times 5 \times 5$ monopole kernel of PineTree (similar to Fig. 9). The weights are obtained through an optimisation run at $7.81 h^{-1} \text{ Mpc}$ voxel resolution and DM overdensity fields computed using an SPH algorithm. The underlying colour indicates the absolute value of each weight.