The Astrophysical Journal, 956:149 (29pp), 2023 October 20 https://doi.org/10.3847/1538-4357/acee6f

A Universal Equation to Predict Ω_m from Halo and Galaxy Catalogs

Helen Shao[1,2], Natalí S. M. de Santi[2,3], Francisco Villaescusa-Navarro[1,2], Romain Teyssier[1], Yueying Ni[4,5], Daniel Anglés-Alcázar[2,6], Shy Genel[2,7], Ulrich P. Steinwandel[2], Elena Hernández-Martínez[8], Klaus Dolag[8,9], Christopher C. Lovell[10,11], Lehman H. Garrison[2], Eli Visbal[12], Mihir Kulkarni[12], Lars Hernquist[4], Tiago Castro[13,14,15], and Mark Vogelsberger[16,17]

[1] Department of Astrophysical Sciences, Princeton University, 4 Ivy Lane, Princeton, NJ 08544 USA; hshao@princeton.edu
[2] Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA
[3] Instituto de Física, Universidade de São Paulo, R. do Matão 1371, 05508-900, São Paulo, Brasil
[4] Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA
[5] McWilliams Center for Cosmology, Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[6] Department of Physics, University of Connecticut, 196 Auditorium Road, U-3046, Storrs, CT 06269, USA
[7] Columbia Astrophysics Laboratory, Columbia University, New York, NY 10027, USA
[8] Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians-Universität München, Scheinerstr. 1, D-81679 München, Germany
[9] Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Straße 1, D-85741 Garching, Germany
[10] Institute of Cosmology and Gravitation, University of Portsmouth, Burnaby Road, Portsmouth, PO1 3FX, UK
[11] Centre for Astrophysics Research, School of Physics, Engineering & Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK
[12] Department of Physics and Astronomy and Ritter Astrophysical Research Center, University of Toledo, 2801 W Bancroft Street, Toledo, OH 43606, USA
[13] INAF-Osservatorio Astronomico di Trieste, Via G.B. Tiepolo 11, I-34143 Trieste, Italy
[14] INFN, Sezione di Trieste, Via Valerio 2, I-34127 Trieste TS, Italy
[15] IFPU, Institute for Fundamental Physics of the Universe, via Beirut 2, I-34151 Trieste, Italy
[16] Kavli Institute for Astrophysics and Space Research, Department of Physics, MIT, Cambridge, MA 02139, USA
[17] The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received 2023 March 1; revised 2023 July 18; accepted 2023 August 7; published 2023 October 18

Abstract

We discover analytic equations that can infer the value of Ω_m from the positions and velocity moduli of halo and galaxy catalogs. The equations are derived by combining a tailored graph neural network (GNN) architecture with symbolic regression. We first train the GNN on dark matter halos from Gadget *N*-body simulations to perform field-level likelihood-free inference, and show that our model can infer Ω_m with ∼6% accuracy from halo catalogs of thousands of *N*-body simulations run with six different codes: Abacus, CUBEP^{3}M, Gadget, Enzo, PKDGrav3, and Ramses. By applying symbolic regression to the different parts comprising the GNN, we derive equations that can predict Ω_m from halo catalogs of simulations run with all of the above codes with accuracies similar to those of the GNN. We show that, by tuning a single free parameter, our equations can also infer the value of Ω_m from galaxy catalogs of thousands of state-of-the-art hydrodynamic simulations of the CAMELS project, each with a different astrophysics model, run with five distinct codes that employ different subgrid physics: IllustrisTNG, SIMBA, Astrid, Magneticum, SWIFT-EAGLE. Furthermore, the equations also perform well when tested on galaxy catalogs from simulations covering a vast region in parameter space that samples variations in 5 cosmological and 23 astrophysical parameters. We speculate that the equations may reflect the existence of a fundamental physics relation between the phase-space distribution of generic tracers and Ω_m, one that is not affected by galaxy formation physics down to scales as small as $10\,h^{-1}$ kpc.

Unified Astronomy Thesaurus concepts: Cosmology (343); Cosmological parameters (339); Hydrodynamical simulations (767)

1. Introduction

ΛCDM is the current standard model in cosmology that describes the evolution and expansion of the Universe, where CDM denotes cold dark matter and Λ represents the cosmological constant. This model explains how primordial density perturbations in the early Universe were amplified by gravity and eventually lead to the formation of the large-scale structures that we observe today. To accomplish this, the model relies on several cosmological parameters that characterize the composition and other fundamental properties of our Universe. One of them is Ω_m, which quantifies the fractional energy density of total matter, and obtaining an accurate constraint for it is crucial for improving our understanding of the foundational physics that governs the Universe.

Historically, the statistics used to analyze the density and velocity fields of matter and galaxies have been useful probes for Ω_m (Peebles 1980; Davis et al. 1985; Angulo & Hahn 2022). This includes the analysis of redshift-space distortions of galaxy redshift surveys caused by virial and peculiar velocities that deviate from cosmic expansion (Kaiser 1987). Such distortions strongly affect the statistical properties of galaxy clustering because they break the symmetry in the line-of-sight direction. These anisotropies directly probe the growth factor, which depends on Ω_m as described in Sargent & Turner (1977), Tonegawa et al. (2020). Another useful statistic is the pairwise velocity metric defined for galaxies and galaxy clusters as the peculiar velocity difference of pairs along their radial separation vector. Its strong dependence on cosmology has allowed it to effectively provide constraints on various cosmological parameters including $\Omega_{\rm m}$ (Cen et al. 1994; Ma et al. 2015). These methods demonstrate that valuable cosmological information is embedded on the small scales ($\lesssim 5\,h^{-1}$ Mpc).

On large scales ($\gtrsim 10\,h^{-1}$ Mpc), methods that analyze cosmic flows (Dekel 1994) such as the skewness in the divergence of galaxy velocity fields (Bernardeau et al. 1995) have led to constraints on $\Omega_{\rm m}$ independent of the biasing relation between the distribution of galaxies and the underlying matter density field. A similar method is using the Zel'dovich approximation to recover the initial density fluctuation field from observed galaxy peculiar velocity and density fields. With this, one can then compute the one-point probability distribution function (IPDF), which is sensitive to $\Omega_{\rm m}$. Thus, one can tune the value of $\Omega_{\rm m}$ assumed for the observed density fields to fit the IPDF of the observed velocity field (Nusser & Dekel 1992, 1993).

In recent years, there have been significant advances in building detailed numerical simulations that accurately describe the distribution and dynamics of galaxies and dark matter. These include both *N*-body and state-of-the-art hydrodynamic simulations, and they have become powerful tools for constraining cosmological parameters such as $\Omega_{\rm m}$. However, the optimal method that can extract the maximum amount of information from this variety of data is still unknown for non-Gaussian density fields. Fortunately, the advent of revolutionary machine-learning techniques provides an alternative way to extract information from large amounts of data. By training neural networks to learn cosmology directly from generic fields, one can achieve tight constraints on the values of cosmological parameters without relying on summary statistics (Ravanbakhsh et al. 2017; Schmelzle et al. 2017; Gupta et al. 2018; Fluri et al. 2019; Ntampaka et al. 2020; Ribli et al. 2019; Villaescusa-Navarro et al. 2020, 2021a; Villanueva-Domingo & Villaescusa-Navarro 2022).

In particular, graph neural networks (GNNs), which are constructed to handle graph representations of irregular data structures, are especially useful for this purpose because of their unique ability to exploit relational knowledge between nodes in the graphs down to arbitrarily small scales (Battaglia et al. 2018; Hamilton 2020; Bronstein et al. 2021). Specifically, in our previous paper (Shao et al. 2022a), we showed that GNNs are able to infer $\Omega_{\rm m}$ with a 6% accuracy from halo catalogs of *N*-body simulations containing information about the spatial distribution and velocity modulus of the dark matter halos. More importantly, this network was shown to be robust across various *N*-body simulations that are run with different numerical codes, as well as various hydrodynamic simulations that each employ distinct subgrid physics models and astrophysical processes. This suggests that the GNN is employing a fundamental relation between the halo properties and $\Omega_{\rm m}$ that is not affected by numerical errors from the *N*-body simulations or baryonic effects. Moreover, in our companion paper de Santi et al. (2023), we show that GNNs are able to perform robust inference of $\Omega_{\rm m}$ from the 3D positions and 1D velocities galaxies of five different hydrodynamic simulation codes while marginalizing over cosmologies, astrophysical effects, subgrid physics models, and subhalo definitions. These results demonstrate the abundance of robust information contained in the phase-space distribution of halos and galaxies.

However, the learned relation is hard to understand because the GNN encodes information in high-dimensional latent space representations that are not associated with obvious physical interpretations. On the other hand, one can use techniques in symbolic regression to reveal the physics underlying neural networks via mathematical formulae. Symbolic regression algorithms can be trained to approximate any learned network by fitting analytic expressions to the input and output of neural network components. Such approximations may also generalize better to data that exists outside the range of the data distribution used for training because they possess stronger extrapolation properties than neural networks, whose complex functional forms have the tendency to overfit and learn uninformative priors used during training (Villaescusa-Navarro et al. 2020). This method has been recently used to rediscover physical laws in planetary motion, uncover new relations in matter overdensity fields, and more (Cranmer et al. 2019; Cranmer 2020; Wadekar et al. 2020; Villaescusa-Navarro et al. 2021b; Shao et al. 2022b; Bartlett et al. 2022; Delgado et al. 2022; Lemos et al. 2022; Wadekar et al. 2023).

Hence, in this paper, we attempt to understand the physical relations employed by the GNNs presented in Shao et al. (2022a), de Santi et al. (2023) by providing an explicit mathematical formula that approximates the learned networks. To achieve this, we follow a two-step method. First, we train a GNN on halo positions and velocity moduli to show that a model with reduced latent space dimensionality can recover the accuracy and robustness of the model discussed in Shao et al. (2022a). Its compressed architecture will aid the use of symbolic regression and decrease the complexity of the approximating expressions. In the second step, we train a symbolic regressor to find mathematical equations that approximate each component of the GNN model. We show that the discovered analytic expressions are able to preserve the accuracy and robustness of the relation found by the GNN by testing them on halos from thousands of *N*-body and hydrodynamic simulations of varying cosmological and astrophysical parameters. More surprisingly, we also demonstrate that the equations are able to predict the value of $\Omega_{\rm m}$ from galaxy catalogs of five different hydrodynamic simulations. This suggests that the equations may be independent of the complex connection between the spatial and velocity distributions of halos and galaxies. Finally, we attempt to interpret the physical meaning of the equations. Since the expressions reveal that the network is exploiting rotationally symmetric information encoded in the relative velocity modulus of the halo pairs on small scales $\sim 1.35\,h^{-1}$ Mpc, we draw connections to traditional techniques that rely on phase-space distributions for galaxies and halos to constrain $\Omega_{\rm m}$.

This paper is structured as follows. We first describe the data used for this project in Section 2. In Section 3, we describe the architecture of our GNN models, the symbolic regression algorithm, and the methods used to train, validate, and test both models. In Section 4, we present the results of our models and equations. We then provide a discussion of plausible physical interpretations of the equations in Section 5. Finally, we summarize the main findings in Section 6.

2. Data

We train our models using halo catalogs from high-resolution cosmological simulations that contain two halo properties. First, the halo positions, $\boldsymbol{r}$, are defined for the halo center using Cartesian coordinates in comoving-space. Second, the halo velocity modulus, V, is defined as the modulus of the 3D peculiar velocity vector computed with respect to the velocity of the simulation box. In this work, we focus on halo and galaxy catalogs at $z = 0$. We describe the methods to generate the halo and galaxy catalogs we use to train, validate, and test the model in Section 3.1.

2.1. Simulations

We follow the scheme used in Shao et al. (2022a) to test the accuracy and robustness of our models. This strategy is composed of two parts. First, we use cosmological N-body and hydrodynamic simulations that contain different Ω_{m} values organized in Latin hypercubes and varying initial random seed conditions to quantify the percentage constraints and level of precision achieved by the models. Specifically, Ω_{m} varies in the range

$$0.1 \leqslant \Omega_{\mathrm{m}} \leqslant 0.5 \qquad (1)$$

for both the N-body and hydrodynamic simulations. Note that these simulations also vary σ_8 in the range $0.6 \leqslant \sigma_8 \leqslant 1.0$. Furthermore, for the hydrodynamic simulations, we vary several astrophysical parameters; most of them just alter four astrophysical parameters controlling the efficiency of supernova and active galactic nucleus (AGN) feedback, but we also made use of a new set that varies 23 astrophysical parameters controlling most of the free parameters in the considered hydrodynamic code. The hydrodynamic simulations have been run with five different codes that not only solve the hydrodynamic equations using different methods but made use of different subgrid models. These simulations are part of the CAMELS project, and we refer the reader to Villaescusa-Navarro et al. (2021b, 2023) for further details.

Second, we use simulations that are generated with the same cosmologies and initial seeds for a control setup in which we can determine the robustness of the models when evaluated on halos generated with different codes. For this, we run 6 N-body simulations that have the same initial random seed and value of $\Omega_{\mathrm{m}} = 0.3175$ (all other cosmological parameters are shared among codes), but each is run with a different code. Additionally, we run 4 hydrodynamic simulations that have the same value of $\Omega_{\mathrm{m}} = 0.3$, initial random seed, and employ their fiducial subgrid physics model using 4 distinct codes.

For these two above steps, we employ thousands of N-body and hydrodynamic simulations that have volumes of $(25\ h^{-1}\ \mathrm{Mpc})^3$ and have been run with 11 different codes. We briefly describe these codes below, but for more detailed information, we refer the reader to Shao et al. (2022a) and the listed paper(s) for each code. Note that, at the end of the descriptions for each code, we include the number of simulations generated to contain the same cosmology and initial random seed as the other codes, and the number of simulations that contain varying cosmologies, initial seeds, and/or astrophysical parameters arranged in a Latin-hypercube (or Sobol sequence), respectively.

2.1.1. N-*body Codes*

The different N-body codes follow the evolution of dark matter particles (that represent the cold dark matter plus baryonic fluid) under the effect of self-gravity in a given expanding cosmological background using different numerical techniques and approximations. The six codes we use to run the N-body simulations are described briefly below.

1. *Abacus*. This code computes the long-range gravitational potential by decomposing the near-field and far-field forces in which the near-field forces are reduced to a r^{-2} summation (or an appropriately softened form), and the far-field forces are reduced to a discrete convolution over multipoles (Garrison et al. 2021). We run 51 simulations with Abacus: 1 simulation with a shared cosmology and initial random seed among codes and 50 simulations in a Latin-hypercube with varying values of Ω_{m} and σ_8.
2. *CUBEP^{3}M*. This code employs a particle-particle particle-mesh (P^{3}M) scheme, described in Harnois-Déraps et al. (2013), where long-range gravitational forces are computed via a two-level particle mesh calculation. We ran 51 CUBEP^{3}M simulations: 1 simulation with shared cosmology and initial random seed among codes and 50 simulations in a Latin-hypercube. For the simulation sharing the cosmology and initial random seed, we used the exact same initial particles as in the other codes, whereas the CUBEP^{3}M initial conditions, generated using the Zeldovich approximation, were used for the 50 simulations in the Latin-hypercube.
3. *Enzo*. This is an adaptive mesh refinement code, as described in Bryan et al. (2014), that solves the Poisson equation via a fast Fourier technique (Hockney & Eastwood 1988) on the root grid and a multigrid solver on the individual submesh. We only have one Enzo simulation, which shares the same cosmology and initial random seed with the other codes.
4. *Gadget*. This code utilizes a TreePM algorithm to compute short-range forces and Fourier techniques to calculate long-distance forces, as described in Springel (2005). We use the halo catalogs from these simulations to train the models. We run 1001 of the Gadget simulations: 1 simulation with shared cosmology and initial random seed among codes and 1000 simulations that have different values of Ω_{m}, σ_8, and initial random seed. We use the halo catalogs from these simulations to train the models.
5. *PKDGrav3*. This code computes forces using fast multipole method (Greengard & Rokhlin 1987) as described in Potter et al. (2017). We run 1001 N-body simulations with this code: 1 simulation with shared cosmology and initial random seed among codes and 1000 simulations with different values of Ω_{m}, σ_8, and initial random seed that are organized in a Latin-hypercube.
6. *Ramses*. This code uses the adaptive particle mesh technique described in Teyssier (2002). It solves Poisson's equation level by level using Dirichlet boundary conditions and a Multigrid relaxation solver. We have run 1001 Ramses simulations: 1 simulation with shared cosmology and initial random seed among codes, and 1000 simulations with different values of Ω_{m}, σ_8, and initial random seed that are organized in a Latin-hypercube.

2.1.2. *Hydrodynamic Codes*

The hydrodynamic simulations have been run using codes that solve the hydrodynamic equations with different numerical methods and employ distinct models to describe astrophysical processes such as star formation and feedback from supernova and AGN. The hydrodynamic simulations have been run with the codes Massively Parallel (MP)-Gadget, Arepo, OpenGadget, Gizmo, and SWIFT-EAGLE. In these simulations, which are part of the CAMELS project (Villaescusa-Navarro et al. 2021b), we vary the values of Ω_m, σ_8, the initial random seed, and several astrophysical parameters that we describe below. Instead of referring to these simulations by the name of the code used to run them, we will call them by name of the flagship simulations associated with them and their subgrid model; i.e., ASTRID, IllustrisTNG, Magneticum, SIMBA, and SWIFT-EAGLE respectively. We note that the SB28 simulations have been run with the Arepo code and employ the IllustrisTNG subgrid model, but since they vary 28 parameters, we use a special name for them. Below, we briefly describe the simulations from the different codes:

1. *ASTRID*. These simulations employ the MP-Gadget code to solve the gravity (with TreePM), hydrodynamics (with the pressure-entropy formulation of smoothed particle hydrodynamics, hereafter SPH), and astrophysical processes (Bird et al. 2022; Ni et al. 2022). We have run 1001 simulations with this code, which are 1 simulation with shared cosmology and initial random seed among codes, and 1000 simulations with different values of Ω_m, σ_8, four astrophysical parameters that control the efficiency of supernova and AGN feedback, and initial random seed that are organized in a Latin-hypercube.
2. *IllustrisTNG*. These simulations have been run with the Arepo code (Springel 2010; Weinberger et al. 2020), making use of a TreePM plus moving-mesh finite volume method (Weinberger et al. 2017; Pillepich et al. 2018a). We have run 1029 simulations with this code, which is 1 simulation with shared cosmology and initial random seed among codes, and 1000 simulations with different values of Ω_m, σ_8, four astrophysical parameters that control the efficiency of supernova and AGN feedback, and initial random seed that are organized in a Latin-hypercube. We also run 27 simulations using this code that only differs in the value of their initial random seed to study the effect of cosmic variance, which we refer to as the cataclysmic variable (CV) set. Finally, we have 1 simulation of this code containing a periodic comoving volume of $(205\ h^{-1}\ \mathrm{Mpc})^3$. This simulation is part of the IllustrisTNG-300 set (Naiman et al. 2018; Pillepich et al. 2018b; Marinacci et al. 2018; Nelson et al. 2018; Springel et al. 2018; Nelson et al. 2019), and we use it to quantify how our analytic expressions behave in the presence of supersample covariance effects.
3. *Magneticum*. This simulation is run with the code OpenGadget3 and implements the SPH-scheme following Beck et al. (2016). For more details, see Dolag et al. (2004), Jubelgas et al. (2004), Hirschmann et al. (2014), and Groth et al. (2023). We have run 51 Magneticum simulations, which are 1 simulation with shared cosmology and initial random seed among codes, and 50 simulations with different values of Ω_m, σ_8, four astrophysical parameters that control the efficiency of supernova and AGN feedback, and initial random seed that are organized in a Latin-hypercube.
4. *SIMBA*. These simulations have been run with the GIZMO code (Hopkins 2015) with a TreePM plus Meshes finite mass method; see Davé et al. (2019). We have run 1001 SIMBA simulations consisting of 1 simulation with shared cosmology and initial random seed among codes, and 1000 simulations with different values of Ω_m, σ_8, four astrophysical parameters that control the efficiency of supernova and AGN feedback, and an initial random seed that are organized in a Latin-hypercube.
5. *SB28*. These simulations have been run with Arepo and employ the IllustrisTNG model. They contain 1,024 simulations, and we place them in a different category as they vary the value of 5 cosmological parameters (Ω_m, Ω_b, h, n_s, σ_8), and 23 astrophysical parameters controlling most of the code free parameters. The values of the 28 parameters are organized in a Sobol sequence Sobol (1967).
6. SWIFT-EAGLE. These simulations have been run with the SWIFT-EAGLE code (Schaller et al. 2016, 2018) and employ a subgrid physics model that aims at mimicking the original Gadget-EAGLE model (Crain et al. 2015; Schaye et al. 2015), with some parameter and implementation differences (Borrow et al. 2022). The full model will be described in J. Borrow et al. (2023, in preparation). The suite contains 64 simulations varying the eight subgrid parameters that control stellar and AGN feedback on a Latin-hypercube (parameter ranges are given in square brackets):
   (a) $f_{\mathrm{E,min}}$, the minimal stellar feedback fraction, [0.18, 0.6];
   (b) $f_{\mathrm{E,max}}$, the maximal stellar feedback fraction, [5, 10];
   (c) $N_{\mathrm{H},0}$, pivot point in density that the feedback energy fraction plane rotates around, $[10^{-0.6}, 10^{-0.15}]$;
   (d) σ_n and σZ, energy fraction sigmoid width, controlling the density and metallicity dependence, [0.1, 0.65];
   (e) ε_f, coupling coefficient of radiative efficiency of AGN feedback, $[10^{-2}, 10^{-1}]$;
   (f) ΔT_{AGN}, AGN heating temperature, $[10^{8.3}, 10^{9.0}]$;
   (g) α, black hole accretion suppression and/or enhancement factor, [0.2, 1.1].

3. Methods

In Shao et al. (2022a), we found that GNNs are not only able to infer Ω_m with a 5.6% precision but are also robust across different N-body and hydrodynamic codes, suggesting that the learned relation might be physically fundamental. In this work, we build upon this previous study to understand the found relation and search for an analytic formula that can approximate the mapping from the halo positions and velocities, $\boldsymbol{r}$ and V, to the cosmological parameter, Ω_m. To accomplish this, we make use of both GNNs and symbolic regression algorithms. We refer the reader to Cranmer et al. (2019, 2019) for similar methodologies devised to extract symbolic relations from trained neural networks.

We begin by training a GNN with the goal of obtaining a low-dimensional latent space network to learn a relation between the input halo properties and Ω_m that can approximate the previously found model. We can do this by fixing certain hyperparameters of the GNN so that it has a reduced architecture depth and width. This step is key to aiding the search for analytic expressions when we use symbolic regression to approximate the GNN, as we later explain. We then evaluate this GNN model on halo catalogs from the N-body and hydrodynamic codes described in the previous section to ensure that the sparse architecture is able to achieve comparable precision and accuracy to the model obtained in Shao et al. (2022a). Finally, we use symbolic regression to fit mathematical formulae to each component of the architecture in the trained GNN model to obtain approximate analytic equations. To improve their interpretability, we also make modifications motivated by physical principles, such as preserving the symmetries present in the data and model and simplifying the found expressions. We refer the reader to Figure 1, which depicts this methodology schematically.

In the following sections, we describe in detail the ingredients we use to perform this procedure: (1) the method for constructing the halo (training, validating, and testing) and galaxy catalogs (testing), (2) the graph data used to train the GNN, (3) the GNN architecture and training procedure, (4) the data and procedure used to train the symbolic regressor, and finally, (5) the metrics used to evaluate the accuracy and precision of the models.

3.1. Halo and Galaxy Catalogs

Here, we describe the procedures for constructing the halo and galaxy catalogs that we use to train, validate, and test the GNN and symbolic expressions.

1. *Halo catalogs for training and validating*. For training and validation, we use halo catalogs from the Gadget simulations. For each simulation, we generate 10 halo catalogs by taking all halos with masses larger than $M_{\rm X}$, where $M_{\rm X}$ is a randomly chosen number between $100\, m_{\rm p}$ and $500\, m_{\rm p}$. Here, m_p is the mass of a single dark matter particle. As explained in Shao et al. (2022a), using different dark matter particle thresholds is key to achieving a model that is robust to different simulations. These halo catalogs are generated by running ROCKSTAR (Behroozi et al. 2013) on snapshots from the numerical simulations described above.
2. *Halo catalogs for testing*. We use all N-body simulations described in the previous section and two hydrodynamic simulations: IllustrisTNG and SIMBA. For each simulation, we generate 5 halo catalogs for the five different dark matter particle thresholds: {100, 200, 300, 400, 500}. Note that, for hydrodynamic simulations, the mass of a dark matter halo contains contributions from various mass sources. Hence, instead of considering only the amount of dark matter mass to make our mass cuts, we define m_p as the effective particle mass: $m_p = \frac{1}{N_c}\Omega_{\rm m} V \rho_c$, where V is the volume of the simulation, ρ_c is the Universe's critical density today, and $N_c = 256^3$ is the effective number of particles. These halo catalogs are generated by running ROCKSTAR (Behroozi et al. 2013) on snapshots from the numerical simulations described above. However, for one test where we gauge the robustness of the train models to different halo definitions, we run SUBFIND (Dolag et al. 2009) to generate halo catalogs from the Gadget N-body and Illustris-TNG simulations.
3. *Galaxy catalogs for testing*. We use galaxy catalogs from all the hydrodynamic simulations described in the previous section. We define a galaxy as a subhalo (can be either a central or satellite) that contains a stellar mass of at least $N \times m_*$ where $N \in 3,\ 4,\ 5,\ 6,$ and $m_* = 1.3 \times 10^7\, h^{-1}\, M_\odot$. For each simulation, we construct four catalogs, each using a different N. We limit the range of the stellar mass thresholds to be no larger than $6 \times m_*$ because we find that using larger cuts results in catalogs with galaxy number densities that are smaller than the number densities (from the halo catalogs) used to train the network and equations. We find that using catalogs with number densities that are outside the training range can lead to inaccurate predictions. These galaxy catalogs are generated by running ROCKSTAR (Behroozi et al. 2013) on snapshots from the six hydrodynamic simulations described above, with the exception of the catalogs from the SWIFT-EAGLE simulations, which were generated using the halo finder VELOCIRAPTOR (Cañas et al. 2019; Elahi et al. 2019).

3.2. GNNs

The methods described in this section closely follow those presented in Shao et al. (2022a) to infer $\Omega_{\rm m}$. We emphasize the key changes that we implement in this work are as follows: (1) using only the summation operator as the aggregation function and (2) reducing the depth and width of the GNN architecture with constrained hyperparameter optimization. These steps decrease the complexity of the model and allow for easier interpretation of the learned relations.

3.2.1. Model Input: Halo Graphs

The input of the GNN is a graph defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. The nodes represent the halos (or galaxies), and an edge is created between two nodes if their distance is smaller than the linking radius, $r_{\rm link}$. This property is considered a hyperparameter that we optimize during training, as we explain later. Thus, two nodes $i, j \in \mathcal{V}$ are referred to as neighbors if they are connected via an edge, $(i, j) \in \mathcal{E}$. As in Shao et al. (2022a), we do not consider self-loops and account for periodic boundary conditions when computing distances and angles between nodes.

The nodes and the edges can have different properties associated with them, which we denote as $\boldsymbol{v}_i^{(n)}$ and $\boldsymbol{e}_{ij}^{(n)}$, respectively. The architecture of the GNN models may consist of multiple layers that take a graph as the input and outputs an updated graph. For this reason, we denote the node and edge features at the n^{th} layer with the superscript n.

The initial node feature, represented by $\boldsymbol{v}_i^{(0)}$, that we use is the halo velocity modulus, V. Since the velocities are defined with respect to the simulation box, the node features preserve Galilean invariance. The edge features between nodes i and j at the nth layer are represented by $\boldsymbol{e}_{ij}^{(n)}$, and they contain information about the spatial distribution of halos. To ensure that the model preserves the rotational and translational invariance of the data, we use the following vector for the edge features:

$$\boldsymbol{e}^{(0)} = [\alpha_{ij},\ \beta_{ij},\ \gamma_{ij}] \tag{2}$$

where

$$\alpha_{ij} = \frac{\boldsymbol{r}_i - \boldsymbol{c}}{|\boldsymbol{r}_i - \boldsymbol{c}|} \cdot \frac{\boldsymbol{r}_j - \boldsymbol{c}}{|\boldsymbol{r}_j - \boldsymbol{c}|}, \tag{3}$$

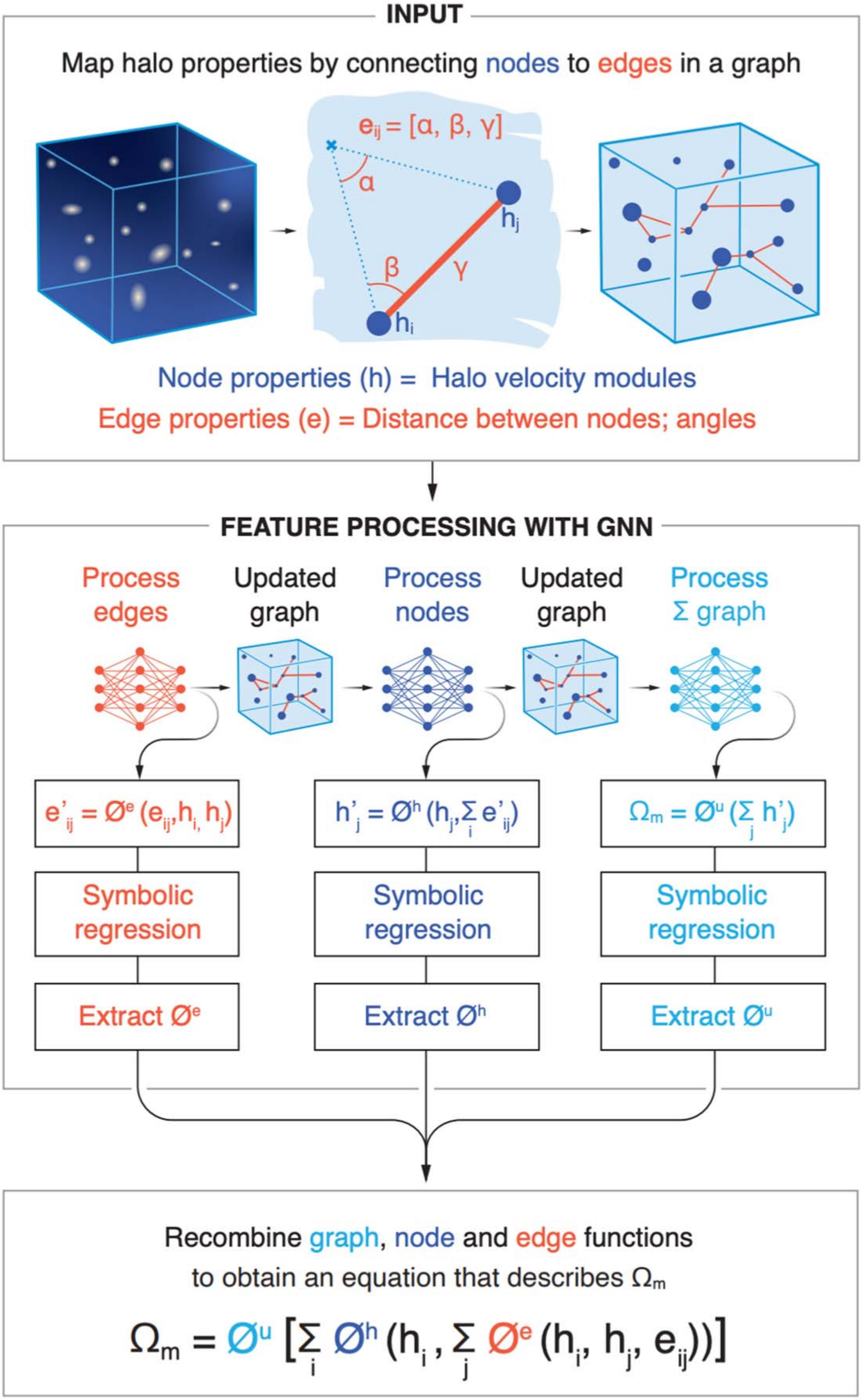
INPUT
Map halo properties by connecting nodes to edges in a graph
$e_{ij} = [\alpha, \beta, \gamma]$
α
β
γ
h_j
h_i
Node properties (h) = Halo velocity modules
Edge properties (e) = Distance between nodes; angles
FEATURE PROCESSING WITH GNN
Process edges
Updated graph
Process nodes
Updated graph
Process Σ graph
$e'_{ij} = Ø^e(e_{ij}, h_i, h_j)$
$h'_j = Ø^h(h_j, \sum_i e'_{ij})$
$\Omega_m = Ø^u(\sum_j h'_j)$
Symbolic regression
Symbolic regression
Symbolic regression
Extract $Ø^e$
Extract $Ø^h$
Extract $Ø^u$
Recombine graph, node and edge functions to obtain an equation that describes Ω_m
$\Omega_m = Ø^u\left[\sum_i Ø^h\left(h_i, \sum_j Ø^e(h_i, h_j, e_{ij})\right)\right]$

Figure 1. This is a schematic of our methodology, which is explained in Section 3. We begin by constructing a graph from a halo catalog using halo positions and velocity moduli. We then feed the graphs to a GNN and train it to perform parameter inference for Ω_m. After training the model, we use symbolic regression to extract the equations from each component of the GNN architecture. Finally, we assemble the equations into one expression and use it to predict Ω_m from halos and galaxies of various *N*-body and hydrodynamic simulations.

$$\beta_{ij} = \frac{\boldsymbol{r}_i - \boldsymbol{c}}{|\boldsymbol{r}_i - \boldsymbol{c}|} \cdot \frac{\boldsymbol{d}_{ij}}{|\boldsymbol{d}_{ij}|}, \tag{4}$$

$$\gamma_{ij} = \frac{|\boldsymbol{d}_{ij}|}{r_{\text{link}}}, \tag{5}$$

with $\boldsymbol{d}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ being the relative distance between the nodes i and j, and $\boldsymbol{c}$ is the centroid of the halo and/or galaxy distribution. α_{ij} defines the angle between the positions of node i and its neighbor node j, while β_{ij} describes the angle between positions of node i and the separation between nodes i and j. Note that we have normalized the distance, $\boldsymbol{d}_{ij}$, by dividing it with the linking radius, r_{link}, to have dimensionless edge features. We refer the reader to Villanueva-Domingo & Villaescusa-Navarro (2022) for more details on this construction.

3.2.2. Architecture

The architecture of our GNN model closely follows COSMOGRAPHNET[18] (Villanueva-Domingo 2022), presented in Villanueva-Domingo & Villaescusa-Navarro (2022) and used in Shao et al. (2022a). However, our model only includes one message-passing layer and a final aggregation layer. We arrived at this architecture by experimenting with different numbers of hidden layers to optimize the simplicity of the model while maintaining the precision and accuracy of its predictions. We explain this in more details in Section 3.2.3.

In the message-passing layer, information from the input node and edge features are encoded with multilayer perceptrons (MLP) and recursively exchanged and aggregated between each node's neighbors and edges. Afterwards, the node and edge features are updated. This creates hidden feature vectors that are ultimately used to predict the target parameter. For this reason, we denote the edge and node features that are input to the message-passing layer (the initial halo properties) with the superscript (0) and the output (hidden) features by the message-passing layer with the superscript (1).

For our compressed GNN, we restricted to two hidden features for each node and edge because the number of hidden features scales in proportion to the number of analytic expressions needed to approximate the network, as we explain later.

For the message-passing layer, the input of the edge model is the initial features of the node i, the neighboring node j, and their shared edge. In this case, the initial node features are V as defined in Section 2, and the initial edge features are $e^{(0)}$ as defined in Equation (2). This information is passed through an MLP, denoted by ϕ^e, and the outputs are the updated hidden edge features:

$$\boldsymbol{e}_{ij}^{(1)} = \phi^e([\boldsymbol{v}_i^{(0)}, \boldsymbol{v}_j^{(0)}, \boldsymbol{e}^{(0)}]). \tag{6}$$

This hidden edge feature, along with the initial node feature of node i, is then passed to the node model, where another MLP, denoted by ϕ^v, outputs the hidden node features:

$$\boldsymbol{v}_i^{(1)} = \phi^v\left(\left[\boldsymbol{v}_i^{(0)}, \sum_{j\in\mathcal{N}_i} \boldsymbol{e}^{(1)}\right]\right). \tag{7}$$

Here, we use a permutationally invariant aggregation function —the summation—to aggregate the node features of the neighbor nodes $j \in \mathcal{N}_i$ that are connected to node i. In Shao et al. (2022a), the aggregation function used was a concatenation of the maximum, summation, and mean operators. In this work, we reduce this function to just the summation to decrease the complexity of the learned relations. This choice is motivated by the fact that the summation can serve as a proxy for the other two operators. Using only one aggregation operator as opposed to three decreases the number of hidden channels by a factor of 3 and thus reduces the number of equations we find for our model.

The final layer in the architecture aggregates the hidden node features output by the message-passing layer to make the prediction $\boldsymbol{y}$:

$$\boldsymbol{y} = \phi^u\left(\left[\sum_{i\in\mathcal{G}} \boldsymbol{v}_i^{(1)}\right]\right), \tag{8}$$

where $\sum_{i\in\mathcal{G}}$ operates over all nodes in the graph, and ϕ^u is another MLP that extracts the target information.

3.2.3. Training Procedure

We train and test the models using graphs constructed from halo catalogs of the Gadget simulations. For each simulation, we construct 10 catalogs using the procedure described in Section 3.1 to marginalize over the halo number density. Once trained, the model is tested using catalogs from all simulations. For Gadget, we split the simulations into training (80%), validation (10%), and testing (10%) data sets before creating halo catalogs for each simulation. For the other codes, we use the entirety of the data set for testing.

We standardize the values of input node features as

$$\tilde{x} = \frac{x - \mu}{\delta}, \tag{9}$$

where μ and δ denote the mean and standard deviation of the feature x. However, we explain in later sections that the value of δ must be tuned for when evaluating the symbolic equations. We also normalize the values of the target cosmological parameter, Ω_{m}:

$$\bar{\Omega}_{\text{m}} = \frac{\Omega_{\text{m}} - \min(\Omega_{\text{m}})}{\max(\Omega_{\text{m}}) - \min(\Omega_{\text{m}})}, \tag{10}$$

where the minimum and maximum values of the ranges of Ω_{m} are listed in Equation (1).

As we did in Shao et al. (2022a), we train the GNN to perform likelihood-free inference, so the output of the model is $\boldsymbol{y} = [\mu_i, \sigma_i]$, where μ_i is the posterior mean, and σ_i is the posterior standard deviation of Ω_{m}. To achieve this, we employ the following loss function:

$$\mathcal{L} = \log\left(\sum_{j\in\text{batch}} (\theta_{i,j} - \mu_{i,j})\right)^2 + \log\left(\sum_{j\in\text{batch}} ((\theta_{i,j} - \mu_{i,j})^2 - \sigma_{i,j}^2)\right)^2 \tag{11}$$

[18] https://github.com/PabloVD/CosmoGraphNet

where the sums are performed over the halo catalogs in the batch. Further details on this can be found in Jeffrey & Wandelt (2020), Villaescusa-Navarro et al. (2022).

Our model is implemented in PyTorch (Paszke et al. 2019), PyTorch Geometric (Elahi & Lenssen 2019). We use the AdamW optimizer (Loshchilov & Hutter 2017) with beta values equal to 0.9 and 0.999. We train the network using a batch size of 8 for 500 epochs. The hyperparameters for our model are as follows: (1) the learning rate, (2) the weight decay, and (3) the linking radius. We use the OPTUNA code (Angulo & Hahn 2019) to perform Bayesian optimization and find the best value of these hyperparameters for each model. As mentioned earlier, we aim to reduce the depth and width of our GNN architecture to obtain a *compressed* network, so we restrict to only one layer and two hidden neurons. For each model, we run 100 trials, where each trial consists of training the model using selected values of the hyperparameters. We perform the optimization of the hyperparameters required to achieve the lowest validation loss possible and use early stopping to save only the model with a minimum validation error.

3.3. Symbolic Regression

While neural networks can provide precise and accurate approximations of complex relations in the data, interpreting them is often challenging because they employ a large number of parameters to make predictions. Therefore, it is desirable to extract mathematical expressions that characterize, or approximate, the relation learned by the neural network because it is easier to understand the physics of the found relationships in such forms. Moreover, analytic equations have been found to generalize better, than neural networks, to data with characteristics not presented in the training set, which can give us more robust predictions and possibly illuminate fundamental properties of the model (Shao et al. 2022b).

For this purpose, we first train a symbolic regression algorithm designed to approximate functions with analytic formulae. We then modify the expressions using reasoning based on physical principles—such as that the model should preserve rotational and translational symmetries of the data—to improve the interpretability of the equations and reduce their complexity. In this section, we describe the symbolic regression algorithm we use and the procedure for fitting functions to components of the learned GNN.

We use the package PYSR (Cranmer 2023) to train a symbolic regression algorithm with the ability to fit mathematical formulas to the learn GNN relations. This package implements genetic programming, which searches for the optimal analytic expression creating combinations between the sets of given operators and input variables. The found expressions of each so-called generation are evaluated, and the most accurate ones *survive* to the next generation. Throughout this iterative process, *mutations* and *crossovers* take place to explore the entire equation space and find an accurate expression.

However, a key limitation of symbolic regression is that its tractability and accuracy are restricted to low-dimensional spaces of input data. To circumvent this, we limit the size of the latent space produced by the GNN, as described in Section 3.2.2. Using the learned parameters and relations from the low-dimensional GNN architecture, we search for equations that characterize the model by approximating the individual MLPs used in the node model, edge model, and final layer described in Equations (6), (7), and (8), respectively. We emphasize that, since there is only one message-passing layer, we only need to approximate one node model MLP and one edge model MLP. Moreover, for each of the node and edge models, we search for two equations because there are two hidden features. The data and procedure used to obtain these equations are described below.

1. *Approximating edge model.* To approximate the edge model, we train a symbolic regressor to map from the input variables, $\boldsymbol{x}^e$, to the target variables, $\boldsymbol{y}^e$, defined as follows:

$$\boldsymbol{x}^e = (v_i^{(0)}, v_j^{(0)}, \alpha_{ij}, \beta_{ij}, \gamma_{ij}); \tag{12}$$

$$\boldsymbol{y}^e = (e_1^{(1)}, e_2^{(1)}). \tag{13}$$

The input variables are the initial features of the nodes and their neighbors, as well as the initial edge features as described in Section 3.2.1. The corresponding target variables are the edge features of the MLP in the edge model defined in Equation (6). Since the GNN employs only two hidden features for each message-passing layer, we denote the first component of the edge feature as $e_1^{(1)}$ and the second component as $e_2^{(1)}$. To obtain this data, we randomly select 10 ($\boldsymbol{x}^e$, $\boldsymbol{y}^e$) pairs from each graph in the training set. This selection is done to ensure that we have a representative sample of the training set without using every node pair of all graphs, which would result in too large of a data set.

2. *Approximating node model.* Similarly, to approximate the node model, the input variables, $\boldsymbol{x}^n$, and the target variables, $\boldsymbol{y}^n$, of the symbolic regressor, are as follows:

$$\boldsymbol{x}^n = \left(v_i^{(0)}, \sum_{j\in\mathcal{N}_i} e_1^{(1)}, \sum_{j\in\mathcal{N}_i} e_2^{(1)}\right); \tag{14}$$

$$\boldsymbol{y}^n = (v_1^{(1)}, v_1^{(1)} + v_2^{(1)}). \tag{15}$$

As seen above, the inputs are the initial node feature and the neighborhood-wise sums of the hidden edge features because the output of the edge model is aggregated using the summation operator before being passed onto the node model. The corresponding target variables are the hidden node features of the MLP in the node model defined in Equation (7). We denote the first and second hidden node features as $v_1^{(1)}$ and $v_2^{(1)}$, respectively. However, instead of directly finding an equation for the second node feature, $v_2^{(1)}$, we instead search for a formula for the sum $v_1^{(1)} + v_2^{(1)}$. This is because we find that the change of variables allows us to obtain more accurate approximations than with the original target variable. Ultimately, to obtain the expression of $v_2^{(1)}$, we subtract from it $v_1^{(1)}$. To obtain this data, we randomly sample 10 ($\boldsymbol{x}^n$, $\boldsymbol{y}^n$) pairs from each graph in the training set as we did with the edge model data.

3. *Approximating final MLP.* Lastly, to approximate the MLP in the final aggregation layer, the input and target variables are as follows:

$$\boldsymbol{x}^u = \left(\sum_{i\in\mathcal{G}} v_1^{(1)}, \sum_{i\in\mathcal{G}} v_2^{(1)}\right); \tag{16}$$

$$\boldsymbol{y}^u = \mu_i. \tag{17}$$

Here, the inputs are the graph-wise sums of the hidden node features because the output of the node model is aggregated using the summation operator before being passed onto the final MLP. The corresponding target is the mean posterior. We do not attempt to find an expression for the posterior standard deviation as it is solely a component of the parameter inference methodology and does not contribute to additional physical understanding. We obtain this data from each graph in the training set. Note that this time there is no need to select a subsample of nodes from each graph because x^u and y^u are global properties of the graph, so we can use every graph in the training set.

In each of the above approximation steps, the symbolic regression algorithm searches for analytic expressions that can map from the given input variables to the desired target. For the training, the regressor is allowed to employ the binary operators "ADD," "SUB," "MULT," "DIV," "POW"[19]; and the unary operators "1/X" (the inverse of a variable), "ABS," "LOG," "LOG10," "SQRT." We employ a standard mean squared error (MSE) loss function to optimize the fitting defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_{\mathrm{true}} - y_{\mathrm{pred}})^2, \tag{18}$$

where $\boldsymbol{y}_{\mathrm{pred}}$ denotes the predicted value of the target variable, and $\boldsymbol{y}_{\mathrm{true}}$ is the corresponding true value. The model was trained for 100,000 trials with a batch size of 64.

During training, the algorithm outputs a list of equations found by the regressor. For each equation, PySR provides three values to quantify the fit of the equation: its complexity, MSE, and score. The complexity of the equation takes into account the number of operators, constants, and variables used. The MSE and the complexity are combined into an overall metric that gives the equation's score, akin to Occam's Razor (Cranmer 2023). Specifically, the algorithm sorts the found equations from the least to the most complex, and for each equation, it computes the fractional decrease in MSE relative to the next (more complex) equation. The score is maximized if this fractional decrease is large. We evaluate several candidate equations on a test set for each hidden feature before selecting one that optimizes the trade-off between complexity and accuracy with these metrics in mind. We note that we have experimented with manipulating the inputs to the symbolic regressor, such as using sums and differences of the velocity moduli of the halo and its neighbor as the input variables, but the regressor struggled to output high-accuracy expressions. We have also tried to restrict the complexity of the symbolic expression outputs, but this significantly reduced the precision of the predictions.

3.4. Performance Metrics

For the graph i, with the true value of the considered parameter $y_{\mathrm{truth,i}}$, our models output the posterior mean, $y_{\mathrm{infer,i}}$, and standard deviation σ_i. To evaluate the accuracy and precision of our models, we follow Villanueva-Domingo & Villaescusa-Navarro (2022), Shao et al. (2022a), and employ four different metrics:

1. the mean relative error, ϵ, defined as
$$\epsilon = \frac{1}{N}\sum_{i}^{N}\frac{|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}|}{y_{\mathrm{truth},i}}, \tag{19}$$
where N is the number of halo catalogs in the test set;
2. the coefficient of determination, R^2, defined as
$$R^2 = 1 - \frac{\sum_{i}^{N}(y_{\mathrm{truth},i} - y_{\mathrm{infer},i})^2}{\sum_{i}^{N}(y_{\mathrm{truth},i} - \bar{y}_{\mathrm{truth}})^2}; \tag{20}$$
3. the root mean squared error, RMSE, defined as
$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{\mathrm{truth},i} - y_{\mathrm{infer}})^2}; \tag{21}$$
4. the Chi squared, χ^2, defined as
$$\chi^2 = \frac{1}{N}\sum_{i=1}^{N}\frac{(y_{\mathrm{truth,i}} - y_{\mathrm{infer,i}})^2}{\sigma_i^2}. \tag{22}$$
Note that a value of χ^2 that is close to one suggests that the standard deviations are accurately predicted. On the other hand, a larger or lower value indicates that the uncertainties are underestimated or overestimated, respectively.

Note that the sums in all expressions above run over the graphs in the test set.

4. Results

In this section, we present the results we obtain from training the GNN model. We then show the analytic approximations that were found using symbolic regression.

4.1. GNN Results

We first train a GNN with a single message-passing layer and fix the number of hidden features to two. Using Bayesian optimization of the hyperparameters, we find that the optimal linking radius is ~1.35 h^{-1} Mpc, which describes the characteristic length scale of the model. When we evaluate the trained model on a test set of Gadget simulations, we find that it is able to attain very accurate predictions of Ω_{m} with a mean relative error of 6% and a χ^2 of 1.37. This indicates that both the posterior mean and standard deviations are accurately inferred. These results are depicted in the left panel of Figure 2. Hence, we see that the accuracy of the model is not significantly compromised by the reduction in the dimensions of its latent space with respect to the model used in Shao et al. (2022a), which was ~5.6%. In the following two sections, we present the results for testing the equations on halos from the six different N-body simulations and four hydrodynamic simulations. We then present the predictions for the model tested on galaxies from six different hydrodynamic simulation suites.

[19] The listed operators perform addition, subtraction, multiplication, and division. "POW" takes the power of X to the input variable, where X is any number.

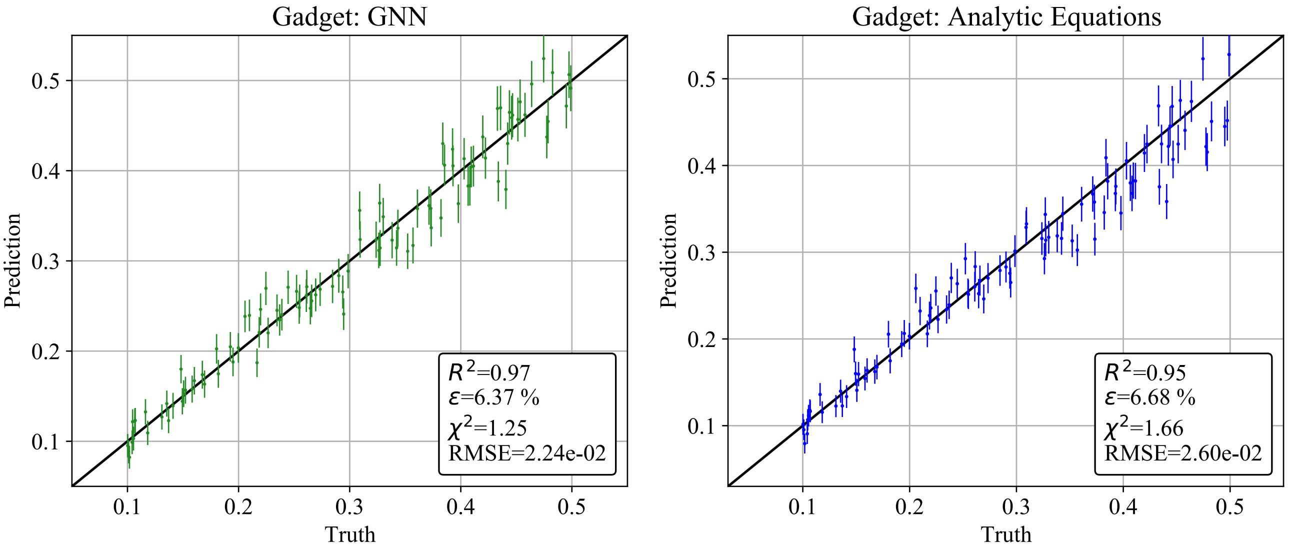

Figure 2. This figure compares the accuracies of the GNN model with the analytic formulae obtained from symbolic regression and the modified equations using physical principles. Left: we first train a GNN with a compressed latent space representation to perform likelihood-free inference for the cosmological parameter Ω_m with halo catalogs containing the positions and velocity moduli of the halos. Evidently, the model is able to achieve very high accuracy with a mean relative error of only ∼6.4% when evaluated on the test set of Gadget simulations. Despite its reduced dimensionality, this accuracy is comparable to the model found in Shao et al. (2022a). Right: we then use symbolic regression to extract analytic expressions for each MLP in the message-passing and final aggregation layers of the GNN. After modifying them to reduce their complexity and to preserve the symmetries of the model, we evaluate the expressions on the Gadget test set. As shown, the expressions are able to maintain the accuracy of the GNN, with an error of only ∼6.7%, indicating that the equations are close approximations for the learned GNN relations.

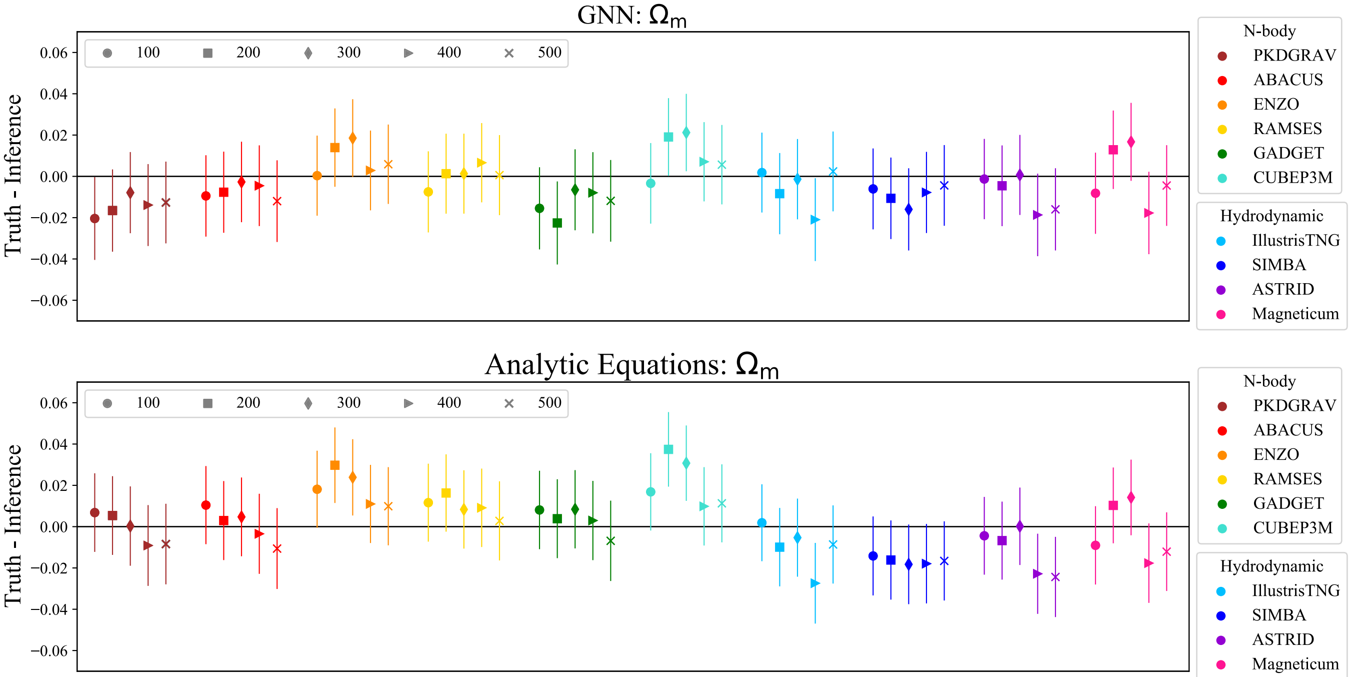

Figure 3. Top: we train a GNN model with a compressed latent space to perform likelihood-free inference for the cosmological parameter Ω_m. The inputs to the model are halo catalogs from Gadget that only carry information about halo positions and peculiar velocity moduli. Once trained, we test the model on halo catalogs from different *N*-body and hydrodynamic simulations as indicated in the legend. We note that simulations of the same type, either *N*-body or hydrodynamic, are run with the same initial conditions, cosmology (and fiducial astrophysics for the hydrodynamic simulations). For each simulation, we generate 5 catalogs. Each halo catalog contains all halos with masses above Nm_p, where m_p is the particle mass, and N can be 100, 200, 300, 400, or 500 (see legend). The *y*-axis represents the difference between the truth and the inference. As can be seen, this model exhibits surprising extrapolation properties and is robust to all simulation codes despite only containing one message-passing layer and two latent features. Bottom: same as above but for the analytic equations obtained using symbolic regression and modified to preserve rotational and translational symmetries in the data, as described in Section 3.3. As can be seen, the formulae maintain the robustness of the GNN model and achieve a very similar accuracy compared to the GNN.

4.1.1. Halos

We first find that the model is robust to different *N*-body codes despite being trained on halo catalogs from only the Gadget simulations, agreeing with the results discussed in Shao et al. (2022a). The simulations we use for this test are Abacus, Ramses, PKDGrav3, Enzo, and CUBEP^{3}M, which share the same cosmology and initial conditions but employ different numerical methods, as described in Section 2. As shown in the top panel of Figure 3, the model obtains similar

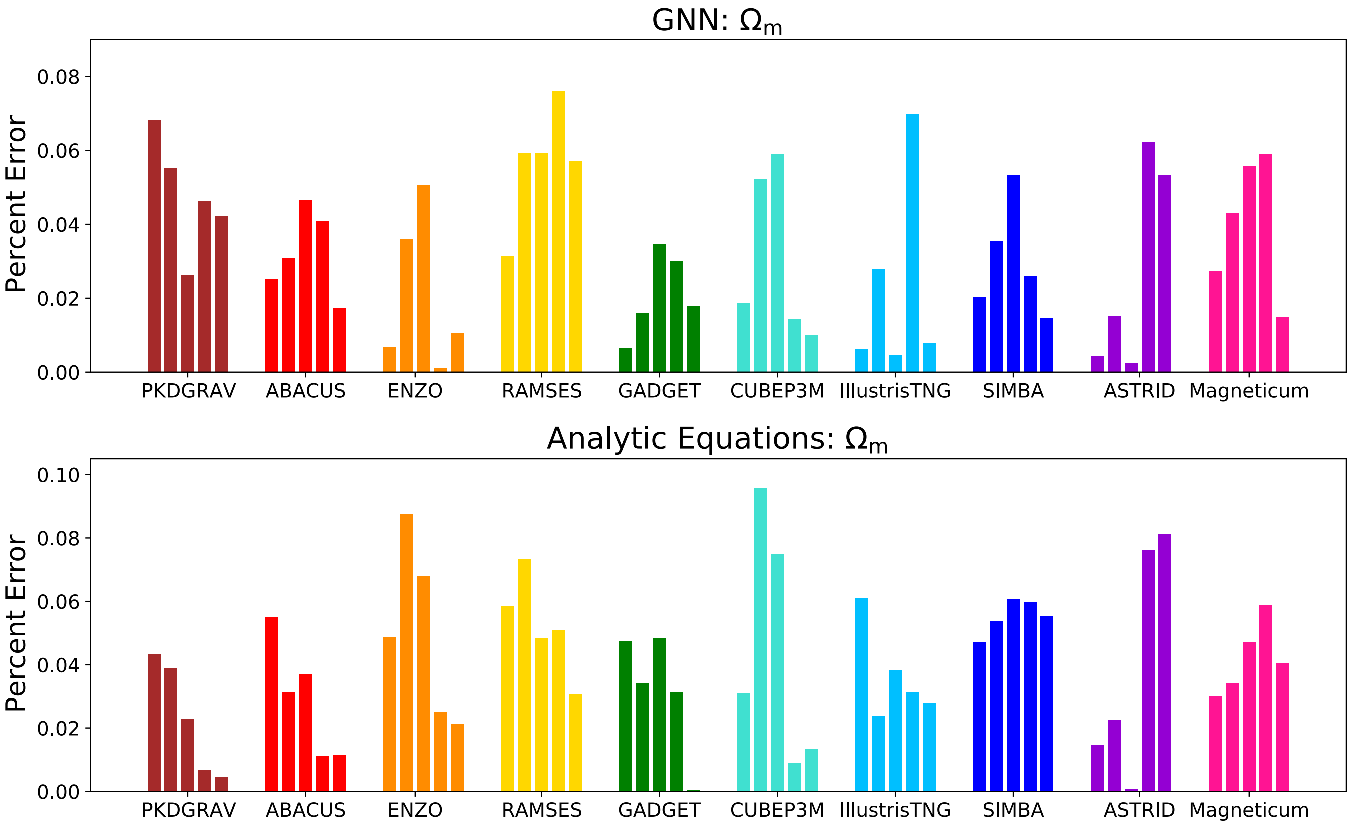

Figure 4. These bar plots follow the same format as Figure 3 except we plot the percent error on the *y*-axes, defined as the ratio of the absolute difference between the truth and predicted to the truth. The five bars of each color represent the different catalogs created for each simulation and are ordered based on the minimum mass thresholds used to construct that catalog, following the same order presented in Figure 3. We note that, while the percent errors obtained by the analytic expressions for certain simulations, such as CubeP^{3}M, appear seemingly larger compared to the other codes, this disparity in one simulation does not reflect the accuracy of the equations because we have performed tests on hundreds of simulations from each code and obtained similar accuracies for them (see Figure 8.)

constraints on Ω_m for these simulations. The corresponding percent errors are shown in Figure 4. We present more detailed results of this test in Appendix A, where the figures depict the accuracies of the model when tested on 50 catalogs of different cosmologies from each simulation code, see Figure 7.

Moreover, the model is robust to different hydrodynamic codes. When tested on halo catalogs from the IllustrisTNG, SIMBA, Astrid, and Magneticum simulations, the GNN is able to achieve similar precision and accuracy compared to the predictions for the *N*-body codes, as seen in the top panel of Figure 3. This demonstrates that the model is robust even to hydrodynamics, varying astrophysical parameters, and different subgrid physics models. This agrees with the results from Shao et al. (2022a) and shows that, even with a reduced latent space dimensionality, the model could possibly still be learning a fundamental relation between the halo properties and Ω_m. However, note that by reducing the size of the latent space the precision of the predictions decreases slightly, which is expected.

Another test that we performed to gauge the extent of the robustness of the GNN is evaluating our model on halos generated using a halo finder that is different (SUBFIND) from the one used during training (ROCKSTAR). We find that the model is able to extrapolate to these halos, and we present the details of this test in Figure 10 in Appendix B.

4.1.2. Galaxies

We also asked if the network would extrapolate to galaxy distributions after being trained on only the positions and velocities of *N*-body halos. Hence, we test the GNN on galaxy catalogs from the following hydrodynamic simulations: Astrid, IlustrisTNG, Magneticum, SB28, SIMBA, and SWIFT-EAGLE. As per the halo catalogs employed in the previous sections, the galaxy catalogs used to perform the following tests contain the galaxy positions and velocity moduli.

We find that the GNN is unable to accurately predict Ω_m for all galaxy catalogs of each simulation. We include the results in Figure 11 of Appendix C. This is not surprising given that the GNN was trained on *N*-body simulations and hence was not given any information regarding the intricate astrophysical and baryonic processes in galaxy distributions. Moreover, the halo–galaxy connection is known to be a complex and challenging relation (Moster et al. 2018; Behroozi et al. 2019).

4.2. Analytic Approximations

Here, we present the equations extracted from the trained GNN model using the symbolic regression method explained in Section 3.3. The formulae for each of the hidden edge and node features, as well as for the predicted posterior mean from the final MLP, are listed in Table 1. The listed RMSE values are computed by individually replacing the corresponding component in the GNN architecture with each expression while keeping all other components of the GNN unchanged and evaluating them on halo catalogs of the Gadget test set. The computed RMSE values are used to gauge the error that each approximate equation introduces.

Table 1
This Table Lists the Analytic Formulae Obtained Using Symbolic Regression for Each Component of the Learned GNN Model: The Edge Model, Node Model, and the MLP in the Final Aggregation Layer

GNN Component	Formula	RMSE
Edge Model: $e_1^{(1)}$	$1.32\lvert v_i - v_j + 0.21\rvert + 0.12(v_i - v_j) - 0.12(\gamma_{ij} + \beta_{ij} - 1.73)$	0.03
Edge Model: $e_2^{(1)}$	$\lvert 1.62(v_i - v_j) + 0.45\rvert + 1.98(v_i - v_j) + 0.55$	0.04
Node Model: $v_1^{(1)}$	$1.21^{v_i}(0.77^{3.29\sum_{j\in\mathcal{N}_j} e_1^{(1)} + \sum_{j\in\mathcal{N}_j} e_2^{(1)}}) + 0.12$	0.02
Node Model: $v_1^{(1)} \ + \ v_2^{(1)}$	$0.78 - \sqrt{\log(0.16^{\sum_{j\in\mathcal{N}_j} e_2 + \sum_{j\in\mathcal{N}_j} e_1 - 0.41 v_i - 1.05})} + 1.45$	0.03
Final MLP: $\mu_{\Omega_{\rm m}}$	$4 \times 10^{-4} \cdot (-5.5\sum_{i\in\mathcal{G}} v_2^{(1)} \ + \ 2.21\sum_{i\in\mathcal{G}} v_1^{(1)} +$ $\lvert 0.96\sum_{i\in\mathcal{G}} v_2^{(1)} \ + \ 0.82\sum_{i\in\mathcal{G}} v_1^{(1)}\rvert) - 0.103$	0.03

Notes. The last column lists the RMSE values of the analytic expressions when they are individually substituted into the GNN architecture. This evaluation is done by replacing the corresponding MLP in the edge model, node model, or final aggregation layer with the symbolic approximation while keeping all other components of the GNN unchanged. When these approximations replace all components of the GNN architecture, the RMSE of the predictions is 0.026, as shown in Figure 2. We note that the edge equations have been modified based on physical motivations to preserve the symmetries of the data. Specifically, we modified the edge equations to depend only on relative velocity moduli $v_i - v_j$, rather than individual halo velocity modulus terms. This is done to enforce the parity between the information from the velocity of a halo and its neighbor. Compared to the predictions shown in Figure 12, we see that using these modified equations improves the overall accuracy of the predictions. The way to use these equations is as follows. First, given a halo and/or galaxy catalog, a mathematical graph is constructed by considering the halos and/or galaxies as nodes and linking nodes by edges if their distance is smaller than $r_{\rm link} = 1.35\ h^{-1}$ Mpc (see Section 3.2.1 for details). Second, the feature of node i is defined as $v_i = (|\vec{v}_i| - \mu)/\delta$, where $|\vec{v}_i|$ is the velocity modulus of halo and/or galaxy i, $\mu = 189$ km s^{-1}, and δ is a free parameter with units of kilometers per second that needs to be adjusted for galaxy catalogs (see Section 4.2.2 and Table 2 for more details). Third, the edge features β_{ij} and γ_{ij} between nodes i and j are computed using Equations (4) and (5), respectively. Fourth, the updated edge features of the graph are computed using the below first two equations. Fifth, the updated node features are computed using the below third and fourth equations. Finally, from the updated graph, we can estimate $\Omega_{\rm m}$ by using the below fifth equation.

It is important to note that the variables v_i and v_j in the equations represent the initial edge features or velocity moduli. As explained in Section 3.2.3, these variables were normalized by the mean and standard deviation of the velocity modulus for the halos from the training set to ensure that all terms in the equations are dimensionless. Hence, the velocity modulus terms in the equations are $v_i = \frac{v_i - \mu}{\delta}$, and $v_j = \frac{v_j - \mu}{\delta}$, where $\mu = 189$ km s^{-1} is a fixed value that was the computed mean velocity modulus for all halos in the training set, and δ is treated as a free parameter. For testing on halo catalogs, we set $\delta = 129$ km s^{-1}, which is equal to the value used during training and was the standard deviation computed for all halos in the training set. On the other hand, for testing on galaxy catalogs, we tune δ to fit to each hydrodynamic simulation set as listed in Section 2 because we find that using the value $\delta = 129$ km s^{-1} leads to inaccurate predictions. This is not surprising given that this value was computed for N-body halos, which would not be expected to extrapolate to galaxies. Hence, it is possible that tuning it for different simulations can account for the halo–galaxy bias. We discuss this in more detail in Section 4.2.2.

We also note that the presented edge equations were modified to include terms that depend only on the relative velocity moduli of the halos and their neighbors. This was done to simplify the equations and improve their interpretability. Moreover, including only the relative velocity modulus as opposed to arbitrary linear combinations of v_i and v_j (see equations in Table 2) enforces the symmetry between the information from the velocity of a halo and its neighbor. Furthermore, as described in Section 2, the halo velocity moduli that appear in all the equations are defined with respect to the simulation box, implying that the equations also preserve Galilean invariance. We note that this modification improves the accuracy of the equations compared to the original expressions found by the symbolic regression algorithm. We include more details on this result, as well as the original equations found by the symbolic regression algorithm, in Appendix D. In the following discussions, we only refer to the modified equations.

Table 2
Optimized Values of the Free Parameter, δ, Used in the Analytic Expressions

Simulation	δ	Simulation	δ
N-body codes	129.2	SB28	100.0
ASTRID	126.5	SIMBA	122.5
Illustris-TNG	99.6	SWIFT-EAGLE	114.5
Magneticum	147.2	…	…

Notes. We list the values for the six different hydrodynamic sets, ASTRID, Illustris-TNG, Magneticum, SB28, SIMBA, and SWIFT-EAGLE. These values were obtained using linear least squares optimization with SCIPY-OPTIMIZE as described in Section 4.2.2 to achieve robustness across various simulation codes. We also include the δ used for testing on N-body halos for comparison.

The accuracy of the equations when evaluated on the halo catalogs of the Gadget simulations is shown in the right panel of Figure 2. As can be seen, these analytic approximations achieve similar mean relative error (6.7%) and RMSE (2.6×10^{-2}) as the GNN, suggesting that they are accurate representations of the trained network. We emphasize that our analytic formula predicts the posterior mean while the error bars (posterior standard deviation) are obtained from the GNN discussed in Section 4.1.

In the following two sections, we present the results for testing the equations on halos from the six different N-body simulations and four distinct hydrodynamic simulation codes, as well as galaxies from six different hydrodynamic simulation sets.

4.2.1. Halos

We first test the robustness of the analytic equations by evaluating them on halos of the different N-body simulations, as we did with the GNN. We find the analytic formulae to be

Table 3
This Table Follows the Format of Table 1 and Lists the Original Analytic Formulas Found by the Symbolic Regression Algorithm

GNN Component	Formula	RMSE
Edge Model: $e_1^{(1)}$	$1.32\|1.05v_i - v_j + 0.21\| - 0.12v_j - 0.12(\gamma_{ij} + \beta_{ij} - 1.73)$	0.028
Edge Model: $e_2^{(1)}$	$\|1.53(v_i - 1.06v_j) + 0.45\| + 1.93(v_i - 1.02v_j) + 0.55$	0.035
Node Model: $v_1^{(1)}$	$1.21^{v_i}(0.77^{3.29\sum_{j\in\mathcal{N}_j} e_1^{(1)} + \sum_{j\in\mathcal{N}_j} e_2^{(1)}}) + 0.12$	0.02
Node Model: $v_1^{(1)} + v_2^{(1)}$	$0.78 - \sqrt{\log(0.16^{\sum_{j\in\mathcal{N}_j} e2 + \sum_{j\in\mathcal{N}_j} e1 - 0.41v_i - 1.05})} + 1.45$	0.03
Final MLP: μ	$4 \times 10^{-4} \times (-5.5\sum_{i\in\mathcal{G}} v_2^{(1)} + 2.21\sum_{i\in\mathcal{G}} v_1^{(1)} +$ $\|0.96\sum_{i\in\mathcal{G}} v_2^{(1)} + 0.82\sum_{i\in\mathcal{G}} v_1^{(1)}\|) - 0.103$	0.03

Notes. The key difference is that the edge model equations originally obtained by the algorithm depend on terms v_i and v_j, which are the individual halo velocity moduli. As in the equations discussed in Section 4.2. the velocities used in the equations here have also been normalized to aid the model training and to ensure that they are dimensionless: $v_i = \frac{v_i - \mu}{\delta}$, $v_j = \frac{v_j - \mu}{\delta}$. For testing these equations on halo catalogs, we use the fixed values $\mu = 189$ km s^{-1}, and $\delta = 129$ km s^{-1} computed from the mean and standard deviation of the velocity moduli for all halos in the training set. The accuracies of these equations are shown in Figure 12.

accurate across all simulations, with predictions of comparable mean relative errors as depicted in the lower panel of Figure 3. We note that, in some cases, the analytic expressions are able to extrapolate better than the GNN due to their known improved generalization abilities (e.g., see Shao et al. 2022b). For instance, certain numerical artifacts that appear in the predictions made by the GNN for boundary cases, such as halo catalogs generated with 100 or 500 minimum particle thresholds, are not present in the predictions made by the analytic expressions. We elaborate on this in Appendix A where we present results for testing the model on simulations of different cosmologies for various N-body codes, as shown in Figure 7. Again, this suggests that the found formulae might represent fundamental relations between the halo properties and the cosmological parameter, Ω_m, as they are not affected by the additional astrophysical processes such as gas cooling and AGN feedback. Similar to the GNN, we perform a second robustness test using halo catalogs generated with SUBFIND and find that the equations reach comparable accuracies. See Appendix B for more details and plots. For all these tests, the depicted errorbars are represent the inferred posterior standard deviation values obtained by the GNN model trained on the halo catalogs since we do not find an expression for this value, as discussed in Section 3.3.

4.2.2. Galaxies

We also test the equations on galaxy catalogs from the six hydrodynamic simulation suites: Astrid, IllustrisTNG, Magneticum, SB28, SIMBA, and SWIFT-EAGLE. We emphasize that this is not a trivial task as the GNN and the corresponding equations were trained using dark matter halos from *N*-body simulations that do not contain any information about the intergalactic dynamics or baryonic processes present in hydrodynamic simulations. There is also a complex galaxy–halo connection, which can, for instance, be reflected in the relative abundances of halos and galaxies where larger halos can contain multiple galaxies while smaller halos may not contain any. These biases can possibly leave a significant imprint in the relations between the relative position and velocity terms of the equations found for halos. For these tests, we follow the definitions of galaxies and stellar mass thresholds discussed in Section 3.1 in constructing the galaxy catalogs where we include both central and satellite galaxies.

We present the results for evaluating the equations on galaxy catalogs from the different hydrodynamic simulations in Figure 5. Each panel is labeled with the corresponding simulation suite. For simplicity, we present the predictions for only the galaxy catalogs generated with the stellar mass threshold of $4 \times m_*$ for a fixed m_* denoting the mass of an individual stellar particle as described in Section 3.1. However, we find that the equations are able to perform with similar accuracies for catalogs constructed with different mass cuts, which we discuss further in Appendix E. Moreover, since the simulations from the SWIFT-EAGLE suite are run with the same value of Ω_m, we plot the difference between the true ($\Omega_m = 0.3$) and the predicted values on the *y*-axis for these catalogs. We note that the presented errorbars for all simulations are the inferred posterior standard deviation values obtained by the model trained and tested on galaxy catalogs discussed in de Santi et al. (2023), since the equations predict only the first moment of the posterior for Ω_m (see Section 3.3). To quantify the error of the predictions made by the analytic equations and the estimated uncertainties, we compute the listed validation statistics.

There are are several important features to note for evaluating the equations on galaxy catalogs from the different hydrodynamic simulations. First, for each simulation, we tune the parameter δ to improve the accuracy of the predictions. As discussed in Section 4.2, this parameter appears in the equation as a normalization of the velocity modulus terms v_i and v_j, and its value varies for different hydrodynamic simulations when testing on galaxies. We tune this normalization because we noticed that, using the original value $\delta = 129$ km s^{-1}, the standard deviation of the velocity moduli for all halos in the training set resulted in predictions that deviated from the truth in terms of a slope and bias, which varies for each simulation. Thus, in Table 3, we list the values of δ that we optimize for each simulation using nonlinear least squares with SCIPY-OPTIMIZE[20] for the catalogs constructed using the $4 \times m_*$ stellar mass threshold. We also compare these found values with the δ used to evaluate on halo catalogs in the table and in later discussions.

Second, after tuning this parameter, we find that the equations are able to predict Ω_m with mean relative errors of 15.35% for ASTRID, 12.85% for Illustris-TNG, 6.89% for Magneticum, 16.17% for SB28, 8.50% for SIMBA, and 4.08%

[20] https://docs.scipy.org/doc/scipy/reference/optimize.html

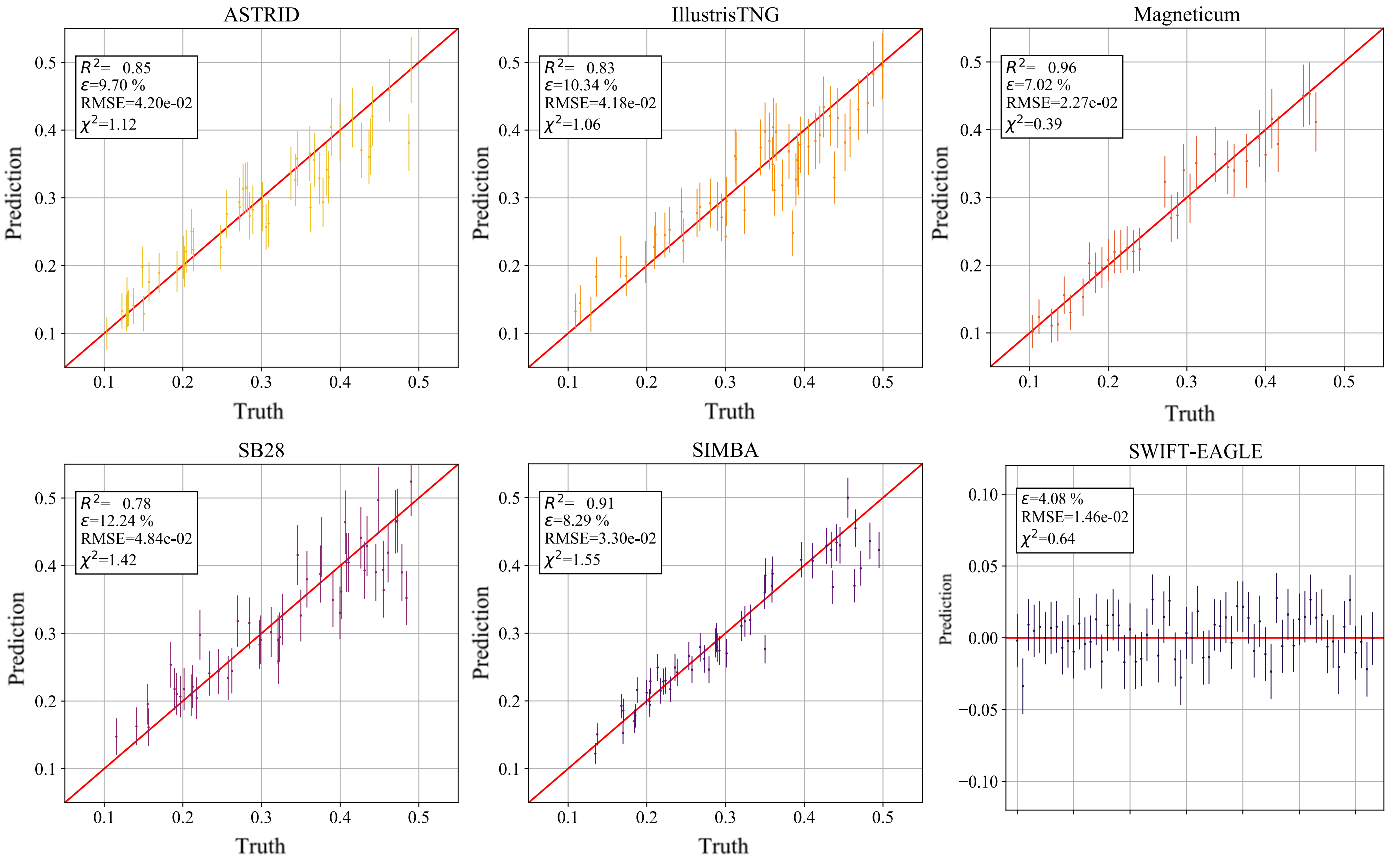

Figure 5. We test the analytic equations that were trained for halo catalogs of *N*-body simulations on thousands of galaxies from 6 different hydrodynamic simulation sets, Astrid, IllustrisTNG, Magneticum, SB28, Simba, and SWIFT-EAGLE, to predict the value of Ω_m and plot the predicted against truth for each simulation. To conserve space, we only present results for the tests performed on catalogs constructed with a stellar mass threshold of $4 \times m_*$ where m_* is a fixed mass for an individual stellar particle as described in Section 3.1, but we reach similar accuracies for catalogs constructed with other mass cuts. We also include only 50 randomly selected catalogs for each simulation set for the clarity of the figures, but the reported metrics were computed for all simulations in the suites. Note that, for the bottom right panel, which depicts the predictions for the SWIFT-EAGLE simulation set, we use simulations that are generated with the same value of $\Omega_m = 0.3$. Thus, we plot the difference between the truth and the prediction on the *y*-axis for these catalogs. As depicted in Figure 6, a large fraction of the catalogs, particularly for the ASTRID, IllustrisTNG, and SB28 simulations, contain galaxy number densities that are outside the range of the number densities exhibited by the halo catalogs used during training of the network and equations. Hence, in this plot, we remove these outliers and find that the mean relative errors of the predictions significantly decrease (see Figure 6 for comparison). These results exhibit a relatively high accuracy with mean errors that average around $\epsilon \sim 9.4\%$, comparable to the accuracies obtained by our companion paper de Santi et al. (2023) with model trained on galaxy properties. This further demonstrates the robustness of the equations as well as their ability to use halo properties to extrapolate to galaxy distributions. This is a surprising result given the various astrophysical processes exhibited by the hydrodynamic simulations and the complex mapping between galaxies and halos. We note that the presented errorbars for all simulations are the inferred posterior standard deviation values obtained by the model trained and tested on galaxy catalogs discussed in de Santi et al. (2023), since the equations predict only the first moment of the posterior.

for SWIFT-EAGLE, across the four stellar mass thresholds. Evidently, the predictions for the galaxy catalogs from ASTRID, SB28, and Illustris-TNG exhibit significantly larger error than those for the halo catalogs. This can be explained by two reasons. One, there are additional astrophysical processes and dynamics present in the thousands of hydrodynamic simulations that can interfere with the equations' extrapolation ability. Given that the equations can only encode information regarding the gravitational interactions between halos from *N*-body simulations, the effects of these various astrophysical parameters may impede on the accuracy of the predictions. Moreover, there are likely to be significantly more outliers for simulations such as SB28, where we vary 28 cosmological and astrophysical parameters at a time. This is also true for the ASTRID simulations, which encompass a wider range of galaxy properties and are able to encapsulate the variations found in the other simulation suites. A more detailed discussion of the wide range of characteristics in the ASTRID simulations can be found in our companion papers, de Santi et al. (2023) and in Y. Ni et al. (2023, in preparation).

Two, there is a large fraction of the galaxy catalogs that contain galaxy number densities outside the scope of the halo number densities seen by the GNN and equations during training. For instance, the number of halos in catalogs from the Gadget simulations used for training ranges from ~1000 to 6,000. However, there are galaxy catalogs that contain fewer than 500 galaxies at this stellar mass threshold. These outliers are particularly dominant in the IllustrisTNG, Astrid, and SB28 simulations, which leads to underpredicted values of Ω_m. This effect can be seen in Figure 6, which contains the same plots as Figure 5 but with each scatter-point colored according to the galaxy number density that the catalog contains. The colorbars accompanying each plot indicate the range of the galaxy number densities present in the catalogs. As it can be seen, in the catalogs with significantly lower (higher) galaxy number densities compared to those seen in training, the value of Ω_m is

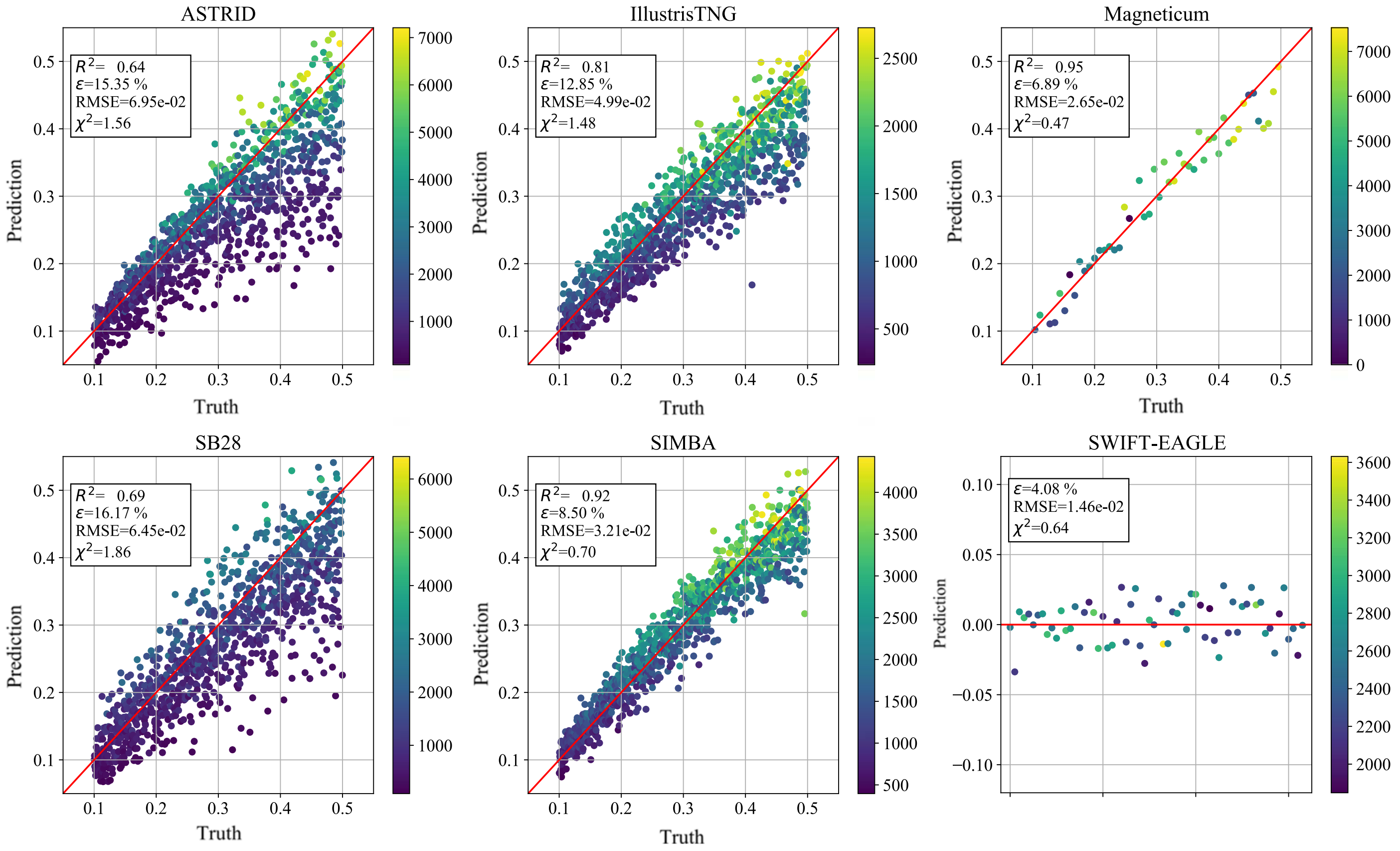

Figure 6. This figure follows the format of Figure 5. Here, each scatterpoint (representing one galaxy catalog) is colored according to the number of galaxies the catalog contains. The colorbar depicts the range of galaxy number density present in the catalogs for the corresponding stellar mass threshold of each column. As it can be seen, a significant portion of the galaxy catalogs from simulations such as Astrid, Illustris-TNG, and SB28 contains much smaller or larger galaxy number densities than the number densities seen during training, which were within the range of (1000, 6000). These catalogs account for the the relatively larger errors in these predictions, which is expected because the halo number density acts as an uninformative prior during the training of the GNN and equations. When we omit these outlier catalogs, we obtain smaller scatter in the results, as shown in Figure 5.

often underpredicted (overpredicted), which contributes to the large scatter. On the other hand, if one removes these outliers, the mean relative errors significantly decrease. Hence, Figure 5 depicts the results for only the catalogs with galaxy number densities that fall within the range of (1000, 6000). Restricting to these catalogs decreases the mean relative errors to the following: 9.76% for Astrid, 10.34% for IllustrisTNG, 7.02% for Magneticum, 12.24% for SB28, 8.29% for SIMBA, and 4.08% for SWIFT-EAGLE. Thus, we conclude that the equations are able to extrapolate to galaxies with accuracies that are comparable to those attained for the halo catalogs from hydrodynamic simulations. These results are also comparable to those obtained by our companion paper de Santi et al. (2023) where we trained a model directly on galaxy properties. We note that the effect of the number density being an uninformative prior during the learning process can be diminished by broadening the range of halo number densities used to train the network and equations, but we leave this for future work.

After accounting for the aforementioned details, we conclude that the equations are able to accurately predict $\Omega_{\rm m}$ for galaxy catalogs. We emphasize that the ability of the equations to achieve a reasonable inference of $\Omega_{\rm m}$, being trained on halo catalogs from $N-$body codes, is a surprising result because it is expected that baryonic effects will affect the abundance and clustering of galaxies in a complex and unknown manner. This is particularly astounding for simulations such as those from the SB28 suite that covers a vast volume in parameter space with many regions not covered by the training set (e.g., cosmological parameters like h, n_s, and $\Omega_{\rm b}$). Furthermore, the equations work really well for SWIFT-EAGLE catalogs that were created running a different halo and/or subhalo finder than the one used for training. Furthermore, the equations are robust to the nontrivial galaxy–halo connection as they can map the information learned about the halo position and velocity fields to those for galaxies. We emphasize that, while the network was designed to be robust to different halo masses by the marginalization procedure discussed in Section 3.2.3, this does not explain the ability of the symbolic expressions to be able to generalize to galaxies, and the astrophysical effects were not foreseen by the design of the marginalization procedure that made the model robust to different halo masses. Hence, the ability of the equations to remain robust to these variations provides strong indication that they may be relying on fundamental relations in the galaxy and halo phase-space distribution that encodes effective information on $\Omega_{\rm m}$. Another possibility is that the equations are extracting information on scales unaffected by astrophysical dynamics. In the next section, we explore possible interpretations of these equations in more detail.

5. Discussion

Here, we discuss some speculative interpretations of the found equations. We attempt to only explain the formulae for the edge models because their functional forms are simpler than those for the node models. The edge model also solely employs physical information about the halo positions and velocity moduli, so they are responsible for directly leveraging the clustering and distribution of the halos. This aligns with the analysis from Cranmer et al. (2019), where it was argued that the relations used in the edge models of GNNs are analogous to describing the force laws between pairs of particles in physical systems. We will elaborate on how the edge equations found in this work may also reflect the physical relations pertaining to the halo and galaxy populations. The node model, on the other hand, exhibits a more complex form because it introduces nonlinearities to the formulae and makes use of information pertaining to the aggregate features from all neighboring halos. However, this should not suggest that the equations for the node model contain information that is less important than those for the edge model.

5.1. Relative Peculiar Velocity Modulus

In both edge model equations, $e_1^{(1)}$ and $e_2^{(1)}$, the information regarding the velocities of the halos appear in terms in the form of $(v_i - v_j)$, which indicates that the model is taking advantage of the relative velocity moduli of the halos and their neighbors. This dependence also preserves the parity between the information content of a halo and that of its neighbor. The ability for the edge model in the GNN to employ relational information between pairs of bodies of a system has been a recognized advantage (Cranmer et al. 2019; Cranmer 2020) toward understanding the physical principles underlying the model predictions. We believe that, in this case, using the relative velocities allows the models to gauge the local gravitational forces where the relative velocity moduli between two halos can serve as a proxy for the depth of the potential wells in the bound system. This is reasonable since larger relative speeds of interacting bodies can result from the presence of stronger attractive forces between them. From this, the model may be learning a representation of the masses of the halos. An analogous discussion in Cen et al. (1994) reached similar conclusions pertaining to the pairwise peculiar velocities and speeds, which were found to have strong dependence on Ω_m at the same small scale as that used by the models in this work ($\lesssim 5\,h^{-1}$ Mpc).

We also speculate that the presence of these terms reflects the strong dependence of Ω_m on the information available in the cosmic velocity fields (Dekel 1994; Bernardeau et al. 1995). For instance, Bernardeau et al. (1995) discusses a derived relation between the moments of the scalar field of the peculiar velocity divergence and Ω_m that is independent of the biasing between the distribution of galaxies and the underlying dark matter density field. It is possible that the found expressions in this work reflect a similar relationship because our models have been trained using the scalar halo velocity modulus and demonstrate an accuracy that is not significantly affected by the presence of astrophysical and baryonic effects. We speculate that the network and equations may be correcting for the nonlinearities of the galaxy velocity fields on smaller scales by considering the galaxy distribution and number densities. Specifically, the equations may be obtaining stochastic velocities from the relative positions of galaxies using the baryonic physics present in the hydrodynamic simulations. This information, coupled with the input pairwise velocity moduli, may then be used to compute the contribution of the galaxy velocities from the bulk flows that trace the large-scale structure of the Universe. Since the bulk flows are a consequence of the mass continuity equation, which relates the large-scale density and growth rate, the equations are able to extract cosmological information on Ω_m. A similar argument was made in the formulation of the cosmic virial theorem from Peebles (1976, 1980), which constructs a relation between the mean square relative peculiar velocity computed for galaxy pairs and the galaxy correlation functions. Hence, we emphasize the importance of leveraging both the positions and velocities of the halos and/or galaxies in the analytic expressions. This aligns with previous findings that using only the positions or only the velocities fails to achieve accurate inference (Villanueva-Domingo et al. 2022). Our companion paper, de Santi et al. (2023), also reaches similar conclusions about the amount of information contained in the galaxy phase-space. Moreover, we have found that introducing additional halo properties such as the halo mass and maximum circular velocity eliminates the generalization of the expressions to various simulation codes (Shao et al. 2022a), which further indicates the robustness of the information contained in peculiar velocities for inferring Ω_m.

5.2. Velocity Normalization

Here we also discuss the implications of tuning the normalization of the velocity modulus terms, δ, for galaxies from each simulation set. Previous findings in Juszkiewicz et al. (1999, 2000) indicate that the halo–galaxy distribution bias can induce biases in pairwise velocity statistics defined using the radial separation between galaxies. Thus, we speculate that the normalization of the velocity modulus terms v_i and v_j in our equations reflect a similar correction to account for the fact that the spatial clustering of galaxies may not trace that of the matter field. In that case, it would be expected for the values of δ to differ for various galaxy populations. Since the optimal value of δ varies across different hydrodynamic codes, we hypothesize that this parameter relates the kinematics of the galaxy velocities to their abundances. For instance, as seen in Table 3, the value of δ is largest for the Magneticum simulations, which have been found to contain significantly higher galaxy number densities compared to the other codes (de Santi et al. 2023). Consequently, the disparity in optimal δ values can possibly reflect the variations in the abundances of satellites in simulations of difference codes since the peculiar motions of satellites are more sensitive to small-scale dynamics, and their presence would thus contribute to a larger spread in the dispersion of the peculiar velocity. On the other hand, the mean galaxy number densities are smallest for IllustrisTNG and SB28, which can explain why δ is smallest for these two simulations (see Table 3). We leave for future work to further investigate the role of δ in the context of galaxy abundances, populations, and cosmological inference.

5.3. Spatial Distribution and Clustering

Next, we discuss the implications of halo clustering and spatial distribution in the found edge equations. In the first edge equation, e_1, the presence of the terms β and γ reflects the spatial distribution of the halos in the catalogs. Specifically, the variable $\gamma \in (0, 1]$ describes the distance between two halos where its range is restricted due to its normalization by the linking radius, $r_{\text{link}} \sim 1.35\,h^{-1}$ Mpc, as described in

Section 3.2.1. Thus, a smaller γ would indicate a denser distribution of halos. Meanwhile, the variable $\beta \in [-1, 1]$ describes the angular orientation of a halo with respect to its neighbor and can provide information about the shape of the distribution, e.g., the filamentary structure of the cosmic web. Both parameters are used by the model to learn about the presence of large-scale structures such as superclusters and filaments.

6. Conclusions

In this work, we have found an analytic expression that approximates the relation employed by a GNN that was trained to infer $\Omega_{\rm m}$ from dark mater halo catalogs. This was motivated by the results of Shao et al. (2022a), which found that GNNs are able to perform accurate field-level inference of $\Omega_{\rm m}$ using halo catalogs from various *N*-body and hydrodynamic simulations. These results imply that the found relation could be a fundamental one as it is not affected by varying numerical errors, astrophysical processes, subgrid physics, or even halo definitions. This motivates us to gain a better understanding of the learned relation by approximating it with symbolic equations that are more physically interpretable than a neural network.

To derive the analytic approximations, we followed a two-step approach. We first simplified the model that was used in the previous work to obtain a GNN with reduced latent space dimensionality. The intention for this step was to maintain the accuracy and precision of the model discussed in Shao et al. (2022a) while building a less complex, and hence more easily interpretable, architecture. We train our *compressed* model on catalogs that only contain the positions and peculiar velocity moduli of dark matter halos from *N*-body simulations. Next, we trained a symbolic regressor to fit equations to each component of the trained GNN (see Figure 1).

We summarize the main results of this work below:

1. We train a compressed GNN architecture composed of only 1 message-passing layer and 2 hidden features on halo catalogs from the Gadget *N*-body simulations. We find that it is able to achieve precise constraints on $\Omega_{\rm m}$ with a mean relative error of $\epsilon \sim 6.5\%$, similar to the GNN model with larger latent space dimensionality as discussed in Shao et al. (2022a), which achieved a mean relative error of $\epsilon \sim 5.6\%$.
2. The compressed GNN model, trained on Gadget simulations, is also robust across thousands of halo catalogs generated from five different *N*-body codes—Abacus, CUBEP^{3}M, Enzo, PKDGrav3, Ramses—and four different hydrodynamic codes that employ different galaxy formation implementations—Astrid, IllustrisTNG, Magneticum, SIMBA. This model reproduces the results of Shao et al. (2022a) where the nontriviality of this robustness was discussed.
3. We use symbolic regression to find equations that approximate the different MLPs that our GNN model is comprised of. These analytic equations can approximate the learned relation between $\Omega_{\rm m}$ and the input halo properties with a mean relative error of $\epsilon \sim 6.7\%$ when evaluated on halos from Gadget *N*-body simulations. We then evaluate the equations on thousands of *N*-body and hydrodynamic simulations run with the different codes listed above. Thus, we demonstrate that the equations are able to reproduce the preciseness and robustness of the GNN, concluding that they are successful approximations of the learned network.
4. We further find that the equations are able to extrapolate better than the GNN in certain cases. Specifically, we test on galaxy catalogs from six different hydrodynamic simulation suites and find that while the equations are able to predict the value of $\Omega_{\rm m}$ accurately while the GNN is unable to. This is a surprising feat given that the equations were trained only on halo properties from *N*-body simulations and were not given any information regarding the complex baryonic effects and astrophysical feedback processes present in galaxy interactions. This also demonstrates that the equations may be exploiting a relation between positions, velocities, and $\Omega_{\rm m}$ that is independent of the halo–galaxy connection.
5. To obtain good accuracies in the galaxy catalogs, we need to tune one single free-parameter, δ, which is the normalization of the velocity modulus terms used in the analytic expressions. The value of δ appears to be sensitive to the characteristics of the considered galaxy population. We leave, for future work, studying its physical role as well as the best strategy to constrain it—such as fitting it using a subset of data, marginalizing over its values, or others.
6. As in our companion paper de Santi et al. (2023), we find some robustness to supersample covariance effects, although further work is needed to properly assess it, taking into account the setup we used to train our models. Further details are presented in Appendix F.
7. We attempt to provide a physical interpretation of the equations for the edge component of the GNN, which could reflect physical laws and forces between interacting objects represented by the nodes of the graph. Specifically, the equations demonstrate an explicit dependence on the pairwise velocity modulus and relative positions of halos and/or galaxies at separation distances $\lesssim 1.35\,h^{-1}$ Mpc. These dependencies illustrate how the rotational and translational symmetries present in the data are maintained and exploited by the model. Moreover, the dual reliance on the spatial and velocity fields of the halos indicates that there is robust information embedded in the phase-space distribution of halos, perhaps reflecting some underlying physical law like the continuity equation. We draw speculative connections to past works that have analyzed similar information in observational fields at the same scales, such as the pairwise velocity and speed statistics as analyzed in Cen et al. (1994), Juszkiewicz et al. (1999, 2000), and the use of cosmic velocity fields as seen in Dekel (1994), Bernardeau et al. (1995).

Finally, we briefly discuss the potential applications of our methods to real data. First, we note that currently, GNNs with large connectivity are unable to scale to simulations of large volumes since they can only be feasibly trained on graphs of fewer than ∼10,000 input nodes[21] (halos or galaxies). Larger input data would considerably reduce the training process's efficiency and become computationally prohibited regarding the required memory. As outlined in the caption of Table 1, to apply the found equations to galaxies, one would first need to obtain the positions and velocity moduli of the galaxies and

[21] We are using A100 GPUs with 40 Gb of RAM.

compute the edge properties that are taken as inputs to the equations. On the other hand, we note that the input variables required for the equations are generally not observable. To address this, we refer the reader to our companion paper de Santi et al. (2023) where we have trained GNNs on the 3D positions and 1D velocities of galaxies. These results indicate that robust cosmological information can be extracted from galaxy fields and are thus more applicable for observational data. However, we also note that substantial further steps need to be taken before adapting the above-mentioned procedures to observed galaxies. This includes an investigation of the effects of supersample covariance (which we show, in Appendix F, can be at least partially accounted for by the models), error propagation of the uncertainties in the input variables, and selection biases that may appear in the data from specific astrophysical probes of the galaxy properties. Since the goal of this study was to probe the idea of implementing symbolic regressors for GNN predictions rather than developing a model to use with real data, we leave for the future to investigate these various avenues. While there are many additional steps that need to be taken, we emphasize that finding an analytical universal equation will likely help us to further understand the complex physics underlying the spatial and velocity distributions of galaxies, which can ultimately be useful for parameter inference, one of the most important tasks for upcoming cosmological surveys.

7. Code Availability

The code used for this work is available at https://github.com/HelenShao/halo_galaxy_GNNs/.

Acknowledgments

We thank Lucy Reading-Ikkanda for creating Figure 1. We thank Ravi Sheth, Oren Slone, David Spergel, Ben Wandelt, Michael Strauss, Oliver Philcox, Gigi Guzzo, Marina Silvia Cagliari, and Miles Cranmer for the enlightening discussions. N.S.M.S. acknowledges financial support from FAPESP, grants 2019/13108-0 and 2022/03589-4. The CAMELS project is supported by the Simons Foundation and NSF grant AST 2108078. E.V. is supported by NSF grant AST-2009309 and NASA grant 80NSSC22K0629. E.H. acknowledge supported by the grant agreements ANR-21-CE31-0019/490702358 from the French Agence Nationale de la Recherche/DFG for the LOCALIZATION project. K.D. acknowledges support by the COMPLEX project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement ERC-2019-AdG 882679 as well as by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2094-390783311. T.C. is supported by the INFN INDARK PD51 grant and the FARE MIUR grant ClustersXEuclid' R165SBKTMA. The research in this paper made use of the SWIFT-EAGLE open-source simulation code (http://www.swiftsim.com, Schaller et al. 2018) version 1.2.0.

Appendix A
Additional *N*-body and Hydrodynamic Simulations

In Section 4, we presented the performance of the GNN and analytic expressions on the different *N*-body and hydrodynamic simulations run with the same cosmologies and initial conditions. Here, we present additional results demonstrating the accuracy and robustness of both model predictions for both $\Omega_{\rm m}$ and σ_8 for different minimum halo particle thresholds. For these plots, we evaluate the models on 50 simulations containing different cosmologies and initial conditions for four different *N*-body codes: Abacus, CUBEP^{3}M, PKDGrav, and Ramses (in Figure 8). We also test the models on 1000 simulations from two hydrodynamic codes, IllustrisTNG and SIMBA, but plot the results for 50 randomly selected simulations to conserve space (in Figure 9). As before, we perform these tests using halo catalogs created with different minimum halo particle thresholds as indicated in the plots. Each of these plots depicts the predictions plotted against the truth minus the inference.

As it can be seen, the GNN is able to infer $\Omega_{\rm m}$ accurately for all *N*-body simulations with similar mean relative errors of ∼7%, and the analytic expressions have comparable accuracies of ∼8% (see Figures 7 and 8).

For the hydrodynamic simulations IllustrisTNG and SIMBA, we obtain concurring results where both the GNN and analytic expressions are able to attain mean relative errors of ∼7% (Figure 9). An interesting note is that the GNN predictions for halo catalogs constructed with 100 or 500 minimum particle thresholds exhibit tail biases due to the effects of the prior distribution, as seen in the right panels of Figure 9. These numerical artifacts are not present in the inferences made by the analytic expressions due to their better known generalization capabilities.

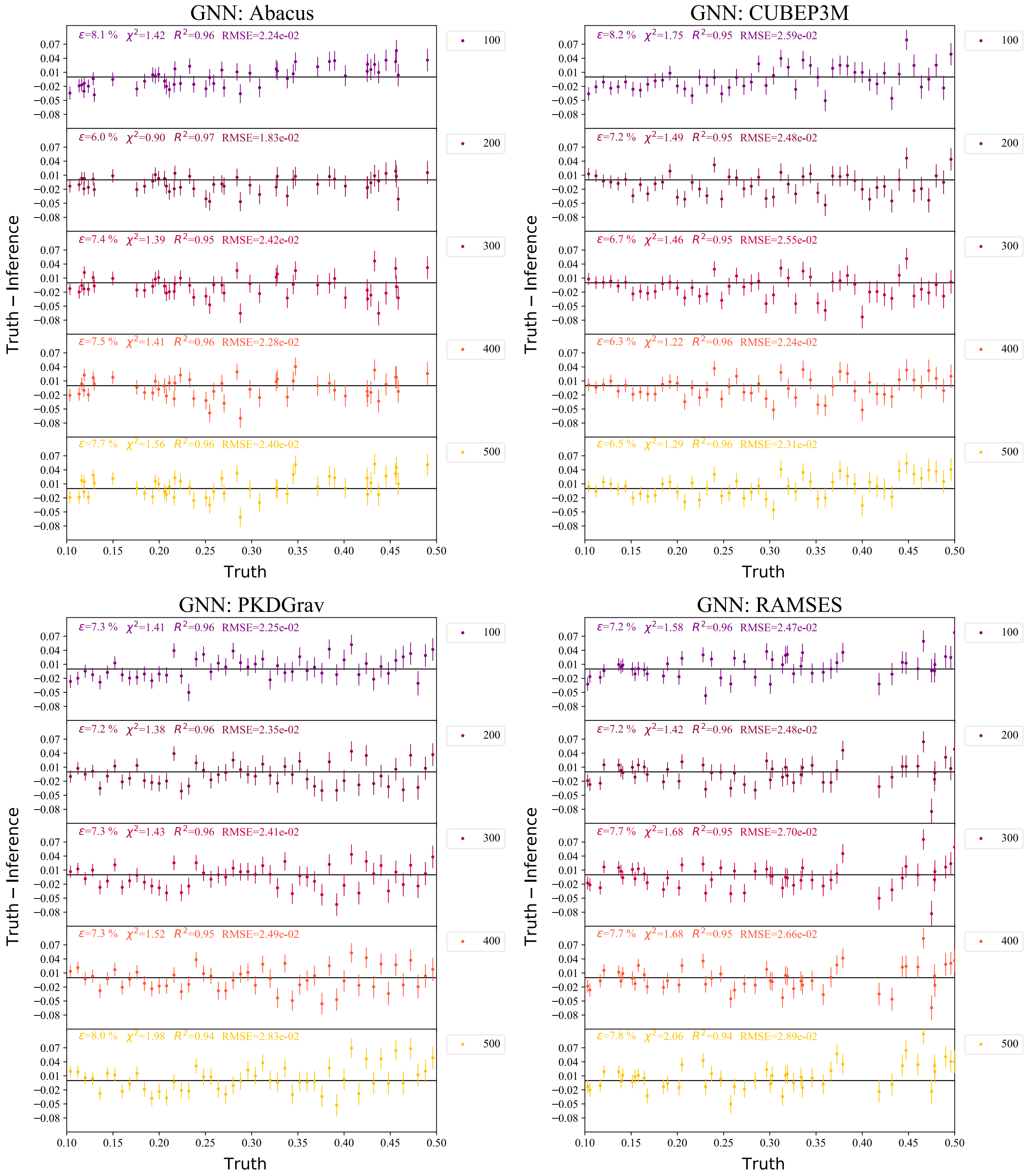

Figure 7. We train a GNN with a low-dimensional latent space to infer Ω_m from catalogs of the Gadget *N*-body simulations using the halo relative positions and velocity moduli. We then evaluate this model on different *N*-body simulations, Abacus, CUBEP^{3}M, PKDGrav3, and Ramses, using catalogs created with particle thresholds indicated next to the plots. As can be seen, the model is able to extrapolate well to different *N*-body codes and is able to predict with similar accuracy compared to that of the halo catalogs from Gadget.

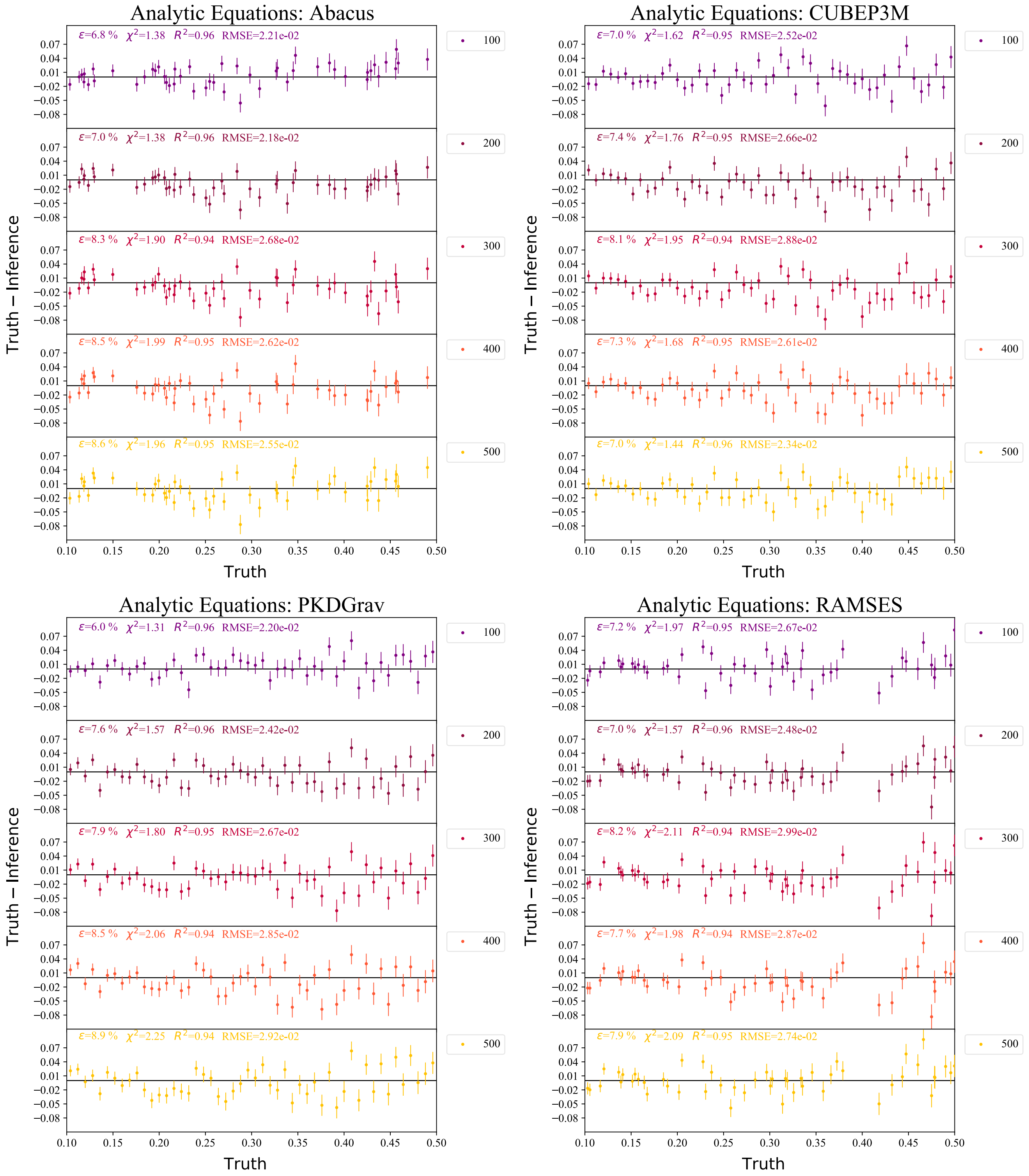

Figure 8. This follows the format as Figure 7 but for the analytic equations discussed in Section D. The equations were found using symbolic regression and modified using physical principles to preserve the rotational and translational symmetries of the data. As can be seen, the equations maintain the accuracy and robustness exhibited by the GNN in Figure 7, indicating that the formulae offer good approximations to the model.

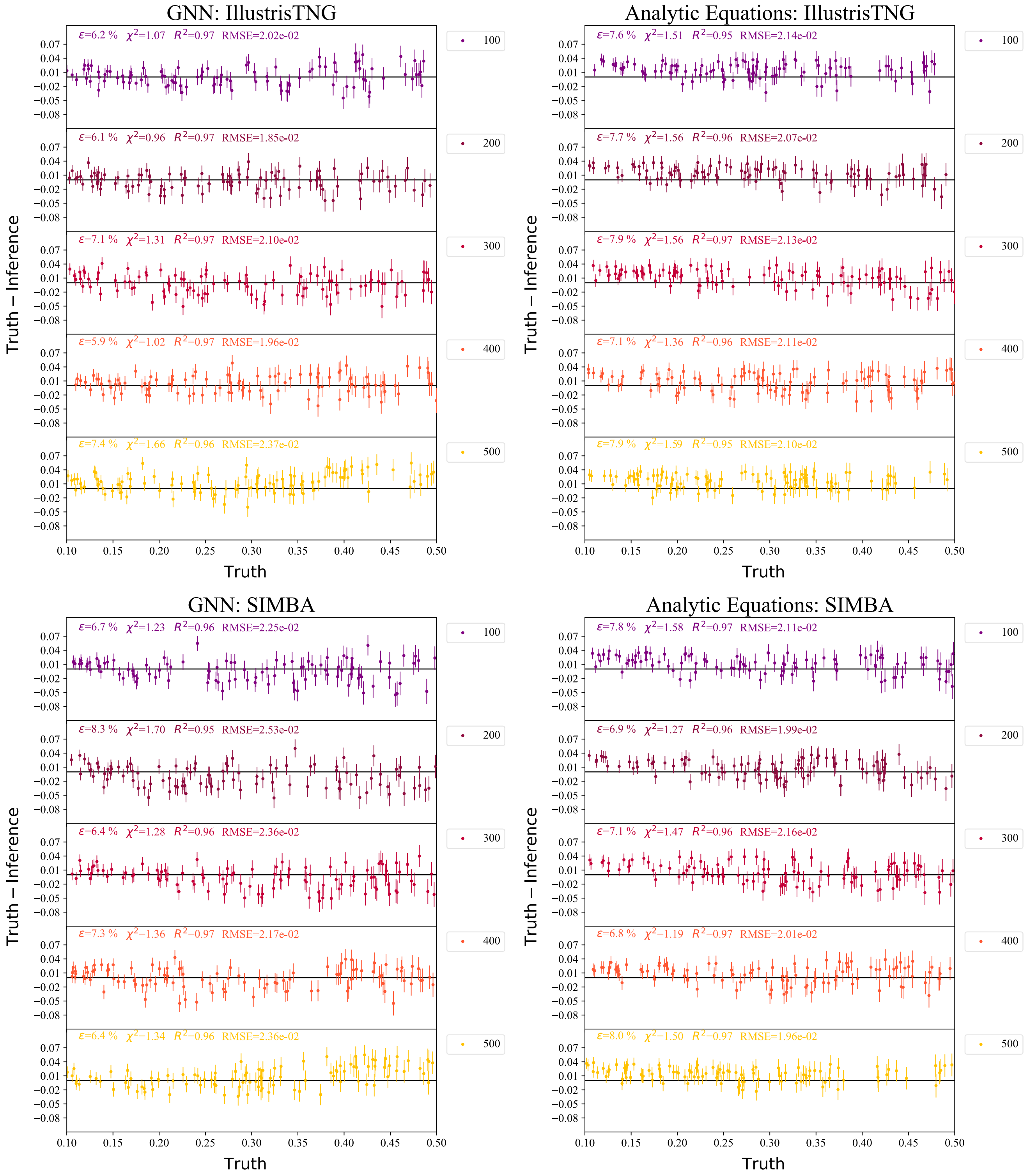

Figure 9. Similar to Figures 7 and 8, we test the GNN and the analytic equations on halo catalogs generated from the SIMBA and IllustrisTNG hydrodynamic simulations. For clarity, we plot the predictions for 50 randomly selected catalogs in each panel. It can be seen that both models remain robust to the additional astrophysical effects present in these simulations, indicating that they are employing a possibly fundamental relation between the relevant halo properties and $\Omega_{\rm m}$. Moreover, the analytic equations are able to capture this as their accuracies for all hydrodynamic simulations, which are similar to those of the GNN.

Appendix B Robustness to Different Halo Finder: SUBFIND

In Section 4, we discussed the accuracy and robustness of the GNN and analytic expressions when evaluated on various simulation codes. Here, we present another test for the robustness of these models where we evaluate the GNN and the analytic approximations on halo catalogs generated using a different halo finder (SUBFIND) than the one used for training (ROCKSTAR). SUBFIND identifies halos by determining local peaks in the 3D density field and separating them using saddle points. The overdense regions and their

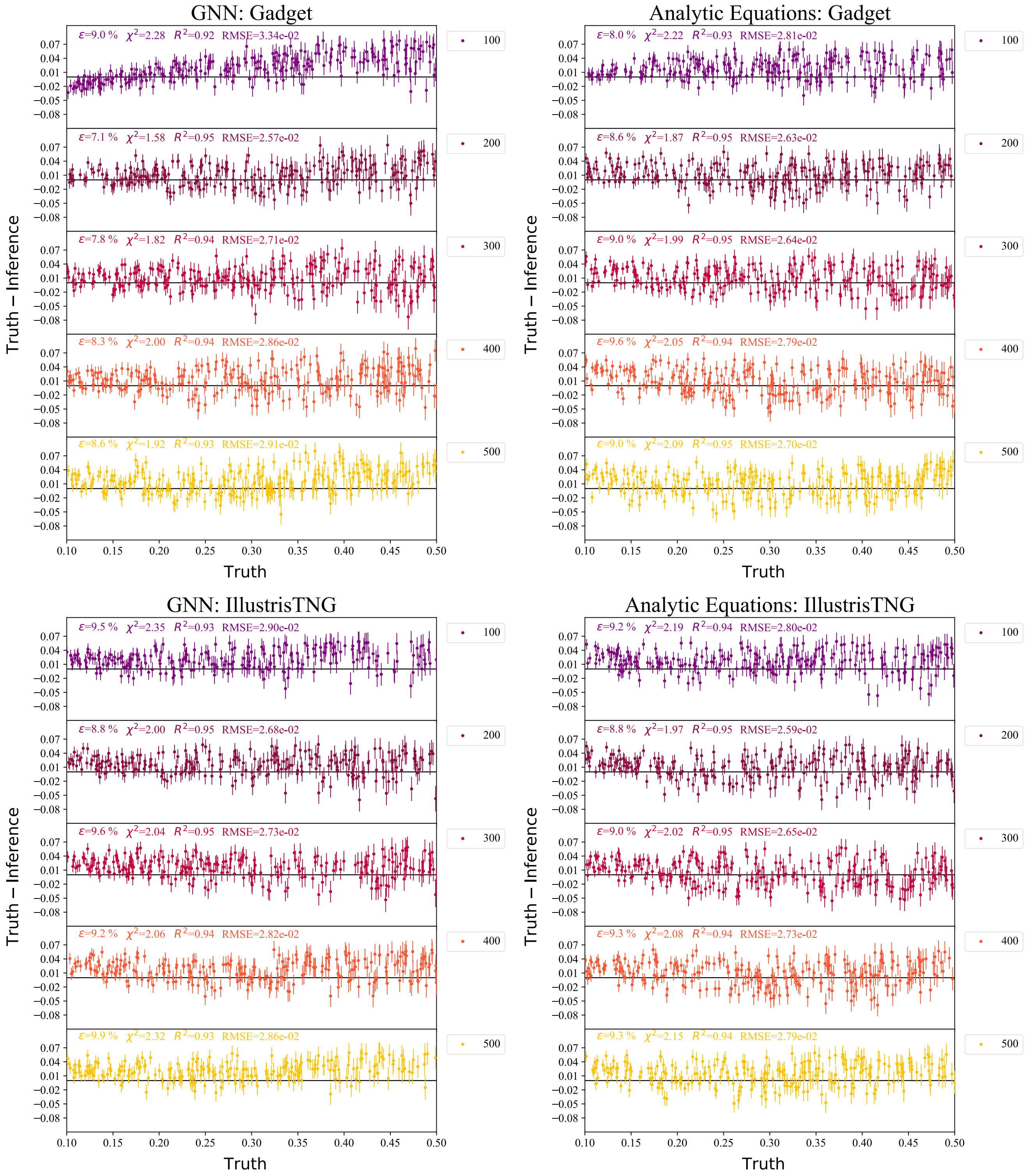

Figure 10. We trained a GNN using halo catalogs generated with the halo finder ROCKSTAR to infer Ω_m, and approximated the learned model with analytic equations using symbolic regression. The top plots show the accuracy of the model and the analytic approximations when evaluated on halo catalogs from the *N*-body Gadget simulations using a different halo finder SUBFIND constructed with the varying minimum particle thresholds as described earlier. It is overall able to accurately extrapolate to the different halo finder with ∼8% mean relative error across the different catalogs. On the other hand, while the analytic expressions have a slightly larger error of ∼9%, they do not exhibit the noticeable biases present in the predictions from the GNN, demonstrating the known improved extrapolation properties of analytic expressions over neural networks. The bottom plots depict the same test as above but for halo catalogs from the IllustrisTNG hydrodynamic simulations.

surroundings are then examined for subhalos, which are gravitationally self-bound regimes. Those that are not bound are attached to their neighboring overdensities with whom they share saddle points. SUBFIND operates on all particle types in the simulations, dark matter and baryonic alike (Dolag et al. 2009).

To perform these tests, we consider the total mass of the halo contained in a sphere with a mean density that is 200 times the mean density of the Universe at redshift $z = 0$. Same as the previous tests, we construct halo catalogs with varying minimum particle thresholds in the range of [100, 500], as explained in Section 2.

First, we perform this test for the 1000 halo catalogs from the *N*-body Gadget simulations. As it can be seen in the top plots of Figure 10, both the GNN and the analytic expressions provide accurate predictions of $\Omega_{\rm m}$ overall, with mean relative errors of ∼8.8% and ∼9.2%, respectively, across the different halo catalogs. However, there are some interesting features to emphasize. First, while the GNN predictions exhibit an offset and a significant lower-tail bias for the halo catalog generated with a minimum particle threshold of 100, this is a boundary case considering the interval of the minimum particle thresholds used to construct the catalogs. Moreover, the identification of lower mass halos can vary across different halo finders, and this can influence the predictions more strongly than the presence of more massive halos, which are more likely to be commonly identified in both halo finders. Second, it can be seen that the analytic approximation demonstrates higher accuracy for this boundary case, which is another indication of the better extrapolation capabilities of analytic equations over neural networks.

Likewise, we performed the same test with halo catalogs from the IllustrisTNG hydrodynamic simulations. The results for this are shown in the bottom plots of Figure 10. As it can be seen, the mean relative errors are similar between the GNN and the analytic expressions, averaging to be ∼9.5% across the different halo catalogs. While these metrics indicate a slightly decreased precision of the predictions, this can be attributed to additional baryonic effects present. Nevertheless, the overall accuracy further demonstrates the generalization ability of the trained network as it is able to extrapolate to both additional hydrodynamic simulations and varying halo definitions. This agrees with the results discussed in Sections 4.2.2 and Appendix C, where it was also found that both the network and the symbolic approximations are able to obtain robust predictions for catalogs generated from the SWIFT-EAGLE simulations, which employ a different halo and/or subhalo finder (VELOCIRAPTOR).

Appendix C
Additional Plots: Testing GNN on Galaxies

As discussed in Section 4.1.2, we trained a GNN on halo catalogs and tested the learned network on galaxies from six different hydrodynamic simulation suites: Astrid, IllustrisTNG, Magneticum, SB28, Simba, and SWIFT-EAGLE. Here, we present the results for these predictions. As it can be seen in Figure 11, the GNN is unable to accurately predict the values of $\Omega_{\rm m}$ as all the predictions exhibit a bias deviating from the true values. This is common across all simulations, which is expected given that there is a nontrivial connection between halo and galaxy distributions. On the other hand, as explained in Section 4.2.2, the analytic equations that approximate the GNN can be tuned to avoid this error.

We believe that these biases are due to the effects of the halo–galaxy connection in addition to the differences in the abundance of galaxies found in the catalogs used for testing and that of halos found in the training data set. As discussed in Section 4.2.2, the network is unable to extrapolate to number densities outside of the training range. In the case of galaxy catalogs, as shown in Figure 6, there are many catalogs with galaxy number densities that fall below the range of the halo number densities seen during training (1000, 6000). However, the underpredicted values of $\Omega_{\rm m}$ cannot be solely attributed to the abundance of galaxies. As discussed in our companion paper, de Santi et al. (2023), the full range of galaxy number densities is exhibited for all values of $\Omega_{\rm m}$. Hence, there is no strong correlation between $\Omega_{\rm m}$ and the number of galaxies in each catalog. This agreement further demonstrates that the biases present in the network predictions are attributed to the intrinsic characteristics of the galaxy population.

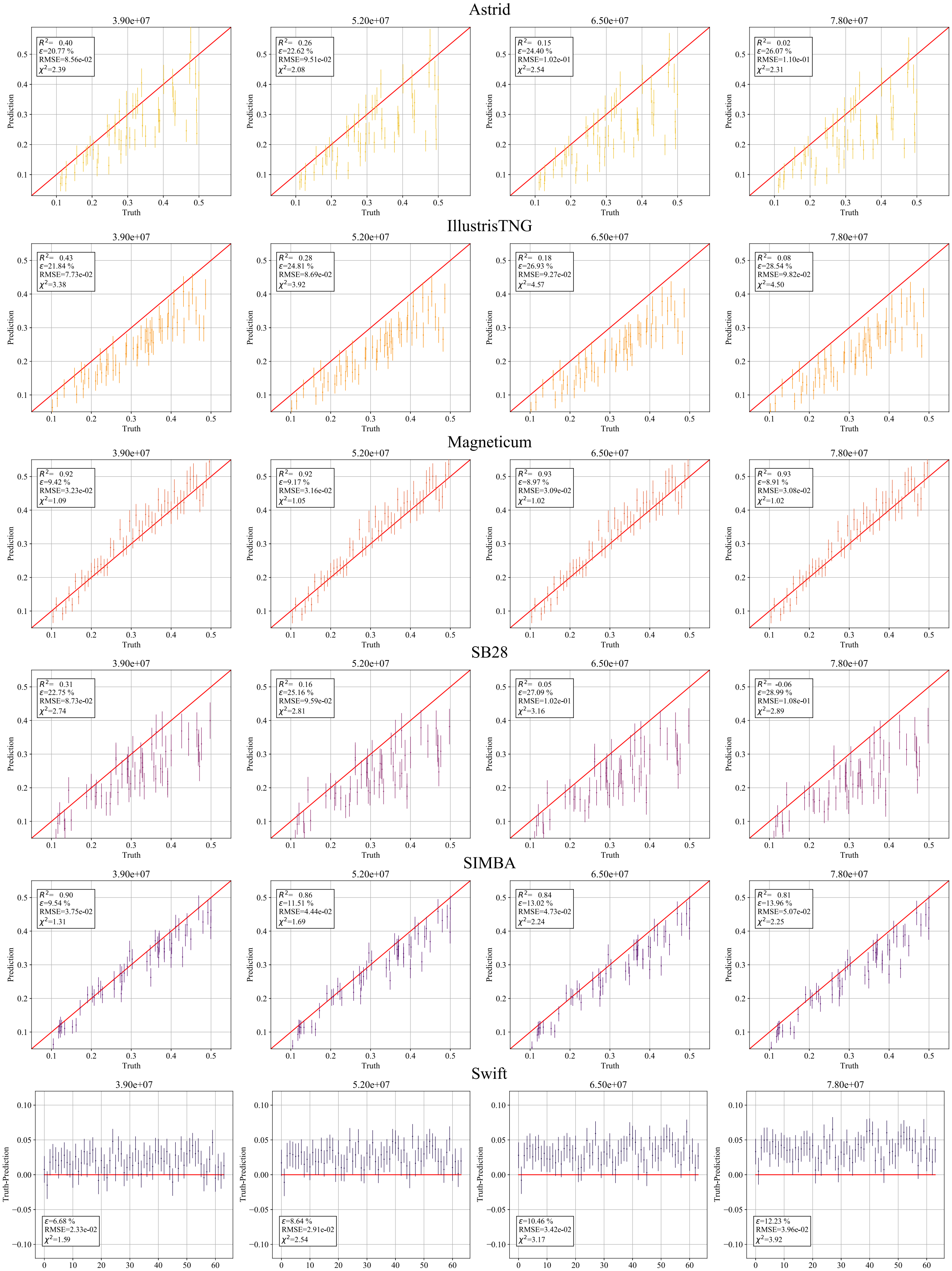

Figure 11. This plot shows the predictions of the GNN trained on halo catalogs from Gadget *N*-body simulations being tested on galaxies from six different hydrodynamic simulations as listed for each row. To construct the galaxy catalogs, we follow the procedure discussed in Section 3.1 and use four different stellar mass thresholds, which are labeled for each column. For clarity, we plot the predictions for 50 randomly selected catalogs in each panel. As can be seen, the GNN is unable to accurately predict the values of Ω_m as all the predictions exhibit a bias deviating from the true values. This is common across all simulations, which is expected given that there is a nontrivial connection between halo and galaxy distributions. However, as explained in Section 4.2.2, the analytic equations that approximate the GNN can be easily tuned to avoid this error.

Appendix D
Original Symbolic Regression Equations

In Section D, we presented the equations obtained by the symbolic regression algorithm that were then modified based on motivations of physical principles. Here, in Table 2, we present the equations originally found by the symbolic regression algorithm that we trained following the procedure described in Section 3.3. Note that the edge equations found by the algorithm contained dependencies on the individual velocity modulus of halos in the form of linear combinations of v_i and v_j. These terms break the parity between a halo and its neighbor. Moreover, we adapted terms that explicitly reflect differences between the velocity moduli due to the known statistics between pairwise velocities and $\Omega_{\rm m}$,

We also show the accuracy of these equations when evaluated on halo catalogs from the Gadget test set simulations in Figure 12. As it can be seen, these formulas are able to achieve a mean relative error of ~7.1%, which is slightly higher than the error of the modified equations, possibly indicating that the imposed symmetries offer an important constraint on the predictions and play a significant role in achieving accurate inferences (see Figure 2).

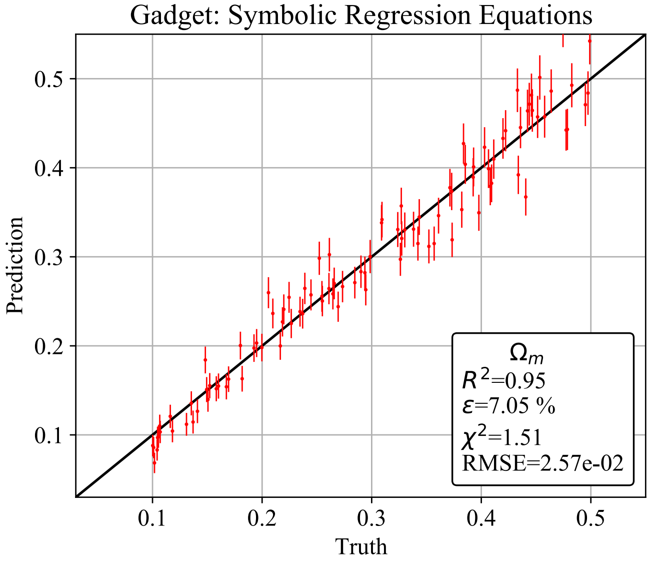

Figure 12. This plot shows the predictions of the original equations found by the symbolic regression algorithm evaluated on the halo catalogs of the Gadget test set. As can be seen, while it achieves similar accuracy to the GNN model, with a mean relative error of 7.1%, it is not as accurate as the modified expressions, which had an error of 6.6%. See Figure 2.

Appendix E
Varying Stellar Mass Thresholds

In this section, we discuss the results for testing the analytic equations discussed in Section 4.2 on galaxy catalogs constructed with different minimum stellar mass thresholds: $N \times m_*$ for $N \in \{3, 4, 5, 6\}$ where m_* is a fixed mass for a single stellar particle. As explained in Section 3.1, the use of different mass cuts during training of the model and equations enables the models to marginalize over the halo and/or galaxy number densities found in each simulation due to different halo and/or galaxy mass functions. Here, we test whether the equations are robust to this using the simulations from ASTRID and SWIFT-EAGLE. To do this, we use the same δ values optimized for catalogs of the single mass threshold $4 \times m_*$ as discussed in Section 4.2.2.

First, we present the results for ASTIRD, which are shown in Figure 13. As it can be seen, the accuracies of the equations are not largely affected by the different mass thresholds used, as expected. Second, we perform these tests for galaxy catalogs from SWIFT-EAGLE, as shown in Figure 14. Here, we explain the apparent trend of increasing scatter in the predicted $\Omega_{\rm m}$ values as the stellar mass threshold increases with the fact that all the SWIFT-EAGLE simulations were run with the same random seed. Hence, the predictions should be considered as highly correlated, which causes the small bias for the catalog with a larger stellar mass threshold for a fixed δ value.

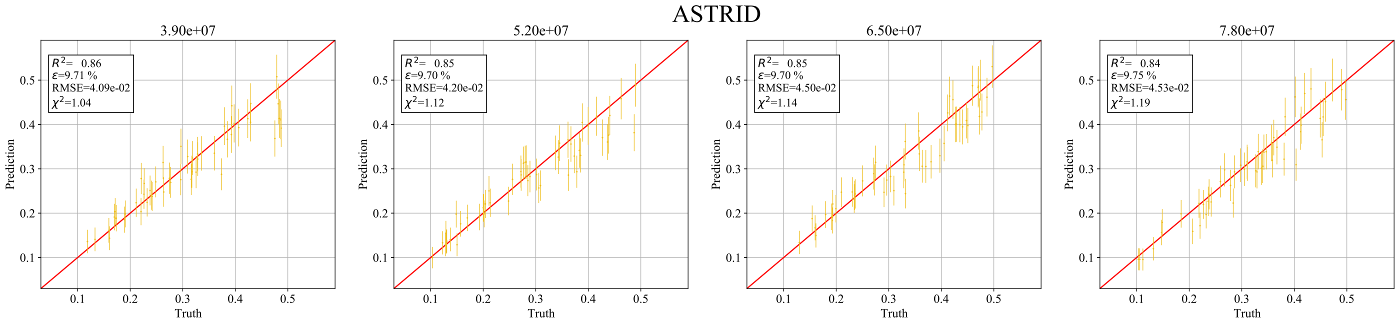

Figure 13. We evaluate the analytic equations discussed in Section 4.2 on galaxy catalogs from the Astrid simulation set constructed using four different minimum stellar mass thresholds: $N \times m_*$ for $N \in \{3, 4, 5, 6\}$ where m_* is a fixed mass for a single stellar particle. Each column is labeled with the corresponding mass cut. As it can be seen, the accuracies of the equations are preserved for the different mass thresholds demonstrating that the model has marginalized over the number density of galaxies.

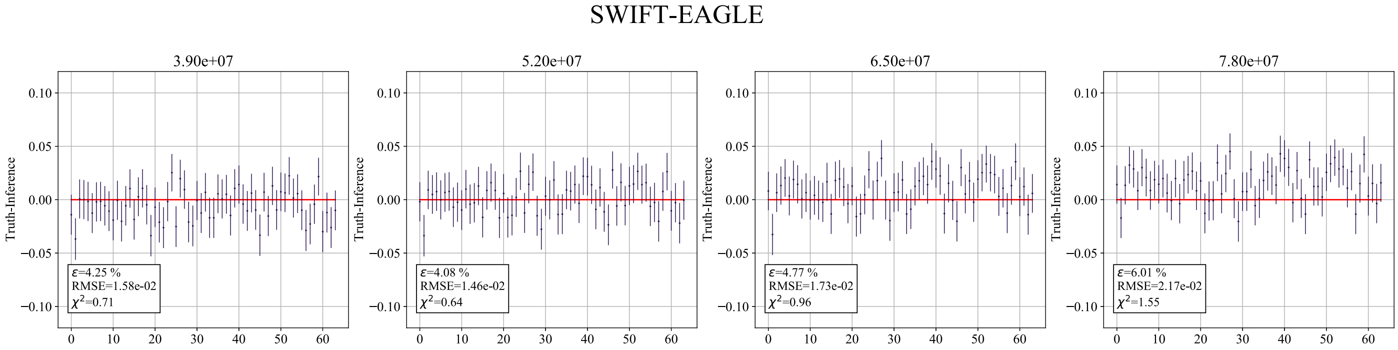

Figure 14. We evaluate the analytic equations discussed in Section 4.2 on galaxy catalogs from the SWIFT-EAGLE simulation set constructed using four different minimum stellar mass thresholds: $N \times m_*$ for $N \in \{3, 4, 5, 6\}$ where m_* is a fixed mass for a single stellar particle. Each column is labeled with the corresponding mass cut. We note that, since the SWIFT-EAGLE simulations were generated using the same initial random seed, there is a high correlation between the galaxy catalogs of the different stellar mass thresholds for this simulation set that is responsible for the trend of decreasing accuracy as the stellar mass threshold increases.

Appendix F
Testing with Supersample Covariance

Here we demonstrate that the analytic equations discussed in Section 4.2 are robust to the effects of supersample covariance. Quantifying how the analytic equations behave in response to supersample covariance is a critical step toward being able to apply them to the observational data from surveys that are sampled with finite volume. This is because, in galaxy surveys, the short-wavelength modes that contain information on the nonlinear dynamics are coupled to long-wavelength, or supersample, modes that extend beyond the survey volume (Hu & Kravtsov 2003; Hamilton et al. 2006; Takada & Hu 2013). This results in sample variances that dominate the nonlinear regime (Takada & Bridle 2007; Sato et al. 2009; Yu 2012; de Putter et al. 2012). In the analysis that we have performed so far, we have not taken into consideration this effect because we have used simulations with periodic boundary conditions, which are not influenced by background modes that extend outside the simulation box.

To test for these effects, we evaluate the analytic equations on galaxy catalogs constructed from $(25\,h^{-1}\ Mpc)^3$ subvolumes randomly selected from the IllustrisTNG-300 simulation to match the size of the simulation boxes used for training. As described in Section 2.1.2, this simulation has a total volume of $(205\,h^{-1}\ Mpc)^3$ and was run with the cosmology $\Omega_m = 0.3089$. It is important to note that, unlike the simulations used for training, we do not impose periodic boundary conditions on the subvolumes used in this test in order to account for the supersample modes.

We present the results of this test in the top panel of Figure 15. Each plot in the figure depicts the differences between the truth and predicted Ω_m made by the analytic equations for 100 randomly selected subvolumes. Following the same procedure used to perform the previous tests on galaxies, we construct four catalogs for each subvolume using the stellar mass thresholds discussed in Section 3.1. Each column is thus labeled with the corresponding stellar mass cut used. We note that the predictions across all catalogs exhibit a common offset from the truth, which we account for by introducing to the final MLP equation an additive constant of $b = -0.19$ found using χ^2 minimization. This common bias is explained by the fact that the equations are being evaluated on subvolumes that do not contain periodic boundary conditions but were trained only on simulations that contain periodic boundary conditions. After correcting for this, the analytic expressions are able to achieve mean relative errors of ∼11.1%.

To confirm that this offset is indeed the consequence of the removal of periodic boundary conditions, we evaluate the analytic expressions on galaxy catalogs constructed from the 27 IllustrisTNG simulations of the CV set as described in Section 2. These simulations were run with the same cosmology of $\Omega_m = 0.3$. We present the results for these simulations in the lower panel of Figure 15, which follow the same format the one above. We find that the predictions for these simulations possess the same offset found in the IllustrisTNG-300 subvolumes. After correcting for this with

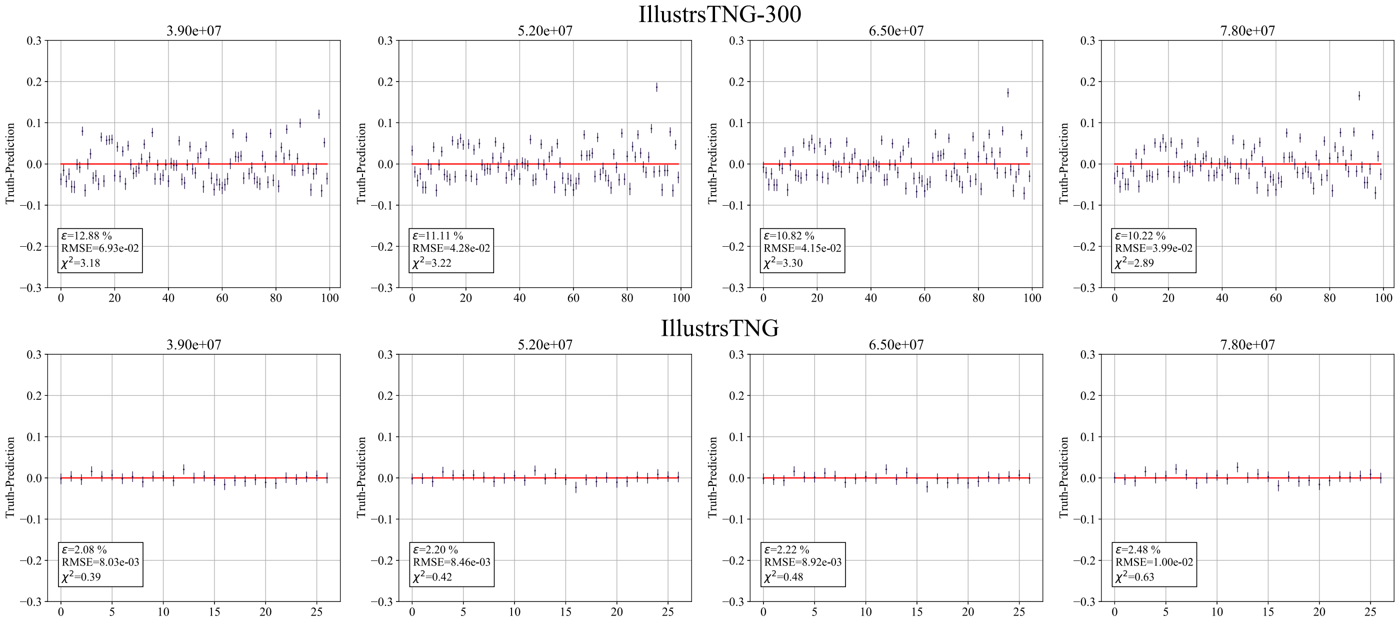

Figure 15. Top: we quantify the behavior of the analytic expressions, discussed in Section 4.2, in the presence of supersample covariance. We test the analytic expressions on 100 $(25\,h^{-1}\,\mathrm{Mpc}))^3$ subvolumes randomly selected from the IllustrisTNG-300 simulation without imposing periodic boundary conditions. The simulation contains a total volume of $(205\,h^{-1}\,\mathrm{Mpc}))^3$ and was run with a cosmology of $\Omega_m = 0.308$. The plots depict the difference between the true Ω value and the predicted for the galaxy catalogs constructed using each of the four stellar mass thresholds as indicated at the top of each column. For all catalogs, the predictions are corrected for their negative offset from the truth by introducing a bias in the analytic expression for the final MLP, $b = -0.19$, which shifts all predictions upwards by a constant. After adjusting for this common offset, the predictions exhibit mean relative errors of ∼11.1%, comparable to the predictions for galaxy catalogs from other hydrodynamic simulation codes as discussed in Section 4.2.2. The common offset can be attributed to the fact that the equations were trained only on simulation volumes of $(25\,h^{-1}\,\mathrm{Mpc})^3$ with periodic boundary conditions and are now being tested on simulations without such conditions. We confirm this reasoning with the results shown in the bottom panel. Bottom: this panel follows the same format as the one above. We show that the analytic equations behave similarly when evaluated on 27 IllustrisTNG simulations from the CV set (see Section 2) after removing periodic boundary conditions. These simulations were run with the same cosmology $\Omega_m = 0.308$. All predictions for galaxy catalogs constructed from these simulations possess a negative offset equal to the one found for the predictions from IllustrisTNG-300 subvolumes, which was adjusted for by introducing a bias to the final MLP equation: $b = -0.19$. After doing so, the predictions exhibit only a mean relative error of ∼2.2%. These results indicate that the analytic equations are able to account for the effects of supersample covariance if one simply shifts the predictions by a constant bias, b, due to the presence of periodic boundary conditions in the training data.

the bias parameter, $b = -0.19$, in the analytic expressions, we achieve mean relative errors of ∼2.2%. This indicates that the offset in the predictions is attributed to the fact that the equations were trained using periodic boundary conditions. This result agrees with the findings of our companion paper, de Santi et al. (2023), where a similar offset was found that is common to all predictions made by a GNN model trained on simulations with periodic boundary conditions and tested on simulations without it. Hence, we conclude that the analytic expressions are able to take into account the effects due to supersample covariance, which is key for applying them to the observational data from surveys that contain finite volume.

ORCID iDs

Helen Shao https://orcid.org/0000-0002-0152-6747
Natalí S. M. de Santi https://orcid.org/0000-0002-4728-6881
Francisco Villaescusa-Navarro https://orcid.org/0000-0002-4816-0455
Romain Teyssier https://orcid.org/0000-0001-7689-0933
Yueying Ni https://orcid.org/0000-0001-7899-7195
Daniel Anglés-Alcázar https://orcid.org/0000-0001-5769-4945
Shy Genel https://orcid.org/0000-0002-3185-1540
Ulrich P. Steinwandel https://orcid.org/0000-0001-8867-5026
Elena Hernández-Martínez https://orcid.org/0000-0002-1329-9246
Christopher C. Lovell https://orcid.org/0000-0001-7964-5933
Lehman H. Garrison https://orcid.org/0000-0002-9853-5673
Eli Visbal https://orcid.org/0000-0002-8365-0337
Mihir Kulkarni https://orcid.org/0000-0002-9789-6653
Lars Hernquist https://orcid.org/0000-0001-6950-1629
Tiago Castro https://orcid.org/0000-0002-6292-3228
Mark Vogelsberger https://orcid.org/0000-0001-8593-7692

References

Angulo, R. E., & Hahn, O. 2022, LRCA, 8, 1
Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. 2019, in KDD '19, Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (New York, NY: Association for Computing Machinery), 2623
Bartlett, D. J., Desmond, H., & Ferreira, P. G. 2022, arXiv:2211.11461
Battaglia, P. W., Hamrick, J. B., Bapst, V., et al. 2018, arXiv:1806.01261
Beck, A. M., Murante, G., Arth, A., et al. 2016, MNRAS, 455, 2110
Behroozi, P., Wechsler, R. H., Hearin, A. P., & Conroy, C. 2019, MNRAS, 488, 3143
Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013, ApJ, 762, 109
Bernardeau, F., Juszkiewicz, R., Dekel, A., & Bouchet, F. R. 1995, MNRAS, 274, 20
Bird, S., Ni, Y., Di Matteo, T., et al. 2022, MNRAS, 512, 3703
Borrow, J., Schaller, M., Bahé, Y. M., et al. 2022, MNRAS, in press
Bronstein, M. M., Bruna, J., Cohen, T., & Velickovic, P. 2021, arXiv:2104.13478
Bryan, G. L., Norman, M. L., O'Shea, B. W., et al. 2014, ApJS, 211, 19
Cañas, R., Elahi, P. J., Welker, C., et al. 2019, MNRAS, 482, 2039
Cen, R., Bahcall, N. A., & Gramann, M. 1994, ApJL, 437, L51
Crain, R. A., Schaye, J., Bower, R. G., et al. 2015, MNRAS, 450, 1937
Cranmer, M. 2023, arXiv:2305.01582
Cranmer, M. D., Sanchez-Gonzalez, A., Battaglia, P. W., et al. 2020, arXiv:2006.11287
Cranmer, M. D., Xu, R., Battaglia, P., & Ho, S. 2019, arXiv:1909.05862
Davé, R., Anglés-Alcázar, D., Narayanan, D., et al. 2019, MNRAS, 486, 2827
Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, ApJ, 292, 371
de Putter, R., Wagner, C., Mena, O., Verde, L., & Percival, W. J. 2012, JCAP, 2012, 019
de Santi, N. S. M., Shao, H., Villaescusa-Navarro, F., et al. 2023, ApJ, 952, 69
Dekel, A. 1994, ARA&A, 32, 371
Delgado, A. M., Wadekar, D., Hadzhiyska, B., et al. 2022, MNRAS, 515, 2733
Dolag, K., Borgani, S., Murante, G., & Springel, V. 2009, MNRAS, 399, 497
Dolag, K., Jubelgas, M., Springel, V., Borgani, S., & Rasia, E. 2004, ApJL, 606, L97
Elahi, P. J., Cañas, R., Poulton, R. J. J., et al. 2019, PASA, 36, e021
Fey, M., & Lenssen, J. E. 2019, Fast Graph Representation Learning with PyTorch Geometric, v2.0.2, https://github.com/pyg-team/pytorch_geometric
Fluri, J., Kacprzak, T., Lucchi, A., et al. 2019, PhRvD, 100, 063514
Garrison, L. H., Eisenstein, D. J., Ferrer, D., Maksimova, N. A., & Pinto, P. A. 2021, MNRAS, 508, 575
Greengard, L., & Rokhlin, V. 1987, JCoPh, 73, 325
Groth, F., Steinwandel, U. P., Valentini, M., & Dolag, K. 2023, MNRAS, 526, 616
Gupta, A., Matilla, J. M. Z., Hsu, D., & Haiman, Z. 2018, PhRvD, 97, 103515
Hamilton, A. J. S., Rimes, C. D., & Scoccimarro, R. 2006, MNRAS, 371, 1188
Hamilton, W. L. 2020, Graph Representation Learning, Vol. 14 (Cham: Springer), 1
Harnois-Déraps, J., Pen, U.-L., Iliev, I. T., et al. 2013, MNRAS, 436, 540
Hirschmann, M., Dolag, K., Saro, A., et al. 2014, MNRAS, 442, 2304
Hockney, R. W., & Eastwood, J. W. 1988, Computer Simulation using Particles (Bristol: Hilger)
Hopkins, P. F. 2015, MNRAS, 450, 53
Hu, W., & Kravtsov, A. V. 2003, ApJ, 584, 702
Jeffrey, N., & Wandelt, B. D. 2020, in 34th Conf. Neural Information Processing Systems (NeurIPS), https://hal.archives-ouvertes.fr/hal-03047530
Jubelgas, M., Springel, V., & Dolag, K. 2004, MNRAS, 351, 423
Juszkiewicz, R., Ferreira, P. G., Feldman, H. A., Jaffe, A. H., & Davis, M. 2000, Sci, 287, 109
Juszkiewicz, R., Springel, V., & Durrer, R. 1999, ApJL, 518, L25
Kaiser, N. 1987, MNRAS, 227, 1
Lemos, P., Jeffrey, N., Cranmer, M., Ho, S., & Battaglia, P. 2022, arXiv:2202.02306
Loshchilov, I., & Hutter, F. 2017, arXiv:1711.05101
Ma, Y.-Z., Li, M., & He, P. 2015, A&A, 583, A52
Marinacci, F., Vogelsberger, M., Pakmor, R., et al. 2018, MNRAS, 480, 5113
Moster, B. P., Naab, T., & White, S. D. M. 2018, MNRAS, 477, 1822
Naiman, J. P., Pillepich, A., Springel, V., et al. 2018, MNRAS, 477, 1206
Nelson, D., Pillepich, A., Springel, V., et al. 2018, MNRAS, 475, 642
Nelson, D., Springel, V., Pillepich, A., et al. 2019, ComAC, 6, 2
Ni, Y., Di Matteo, T., Bird, S., et al. 2022, MNRAS, 513, 670
Ntampaka, M., Eisenstein, D. J., Yuan, S., & Garrison, L. H. 2020, ApJ, 889, 151
Nusser, A., & Dekel, A. 1992, ApJ, 391, 443
Nusser, A., & Dekel, A. 1993, ApJ, 405, 437
Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv:1912.01703
Peebles, P. J. E. 1976, Ap&SS, 45, 3
Peebles, P. J. E. 1980, The Large-scale Structure of the Universe (Princeton, NJ: Princeton Univ. Press)
Pillepich, A., Springel, V., Nelson, D., et al. 2018a, MNRAS, 473, 4077
Pillepich, A., Nelson, D., Hernquist, L., et al. 2018b, MNRAS, 475, 648
Potter, D., Stadel, J., & Teyssier, R. 2017, ComAC, 4, 2
Ravanbakhsh, S., Oliva, J., Fromenteau, S., et al. 2017, Proc. 33rd Int. Conf. Mach. Learn., 48 (New York, NY: PMLR), 2407, https://proceedings.mlr.press/v48/ravanbakhshb16.html
Ribli, D., Pataki, B. Á., Zorrilla Matilla, J. M., et al. 2019, MNRAS, 490, 1843
Sargent, W. L. W., & Turner, E. L. 1977, ApJL, 212, L3
Sato, M., Hamana, T., Takahashi, R., et al. 2009, ApJ, 701, 945
Schaller, M., Gonnet, P., Chalk, A. B. G., & Draper, P. W. 2016, in Proc. Platform for Advanced Scientific Computer Conf. (New York, NY: Association for Computing Machinery), 2
Schaller, M., Gonnet, Pedro, Draper, Peter, et al., 2018, SWIFT: SPH With Inter-dependent Fine-grained Tasking, Astrophysics Source Code Library, ascl:1805.020
Schaye, J., Crain, R. A., Bower, R. G., et al. 2015, MNRAS, 446, 521
Schmelzle, J., Lucchi, A., Kacprzak, T., et al. 2017, arXiv:1707.05167
Shao, H., Villaescusa-Navarro, F., Villanueva-Domingo, P., et al. 2022a, arXiv:2209.06843
Shao, H., Villaescusa-Navarro, F., Genel, S., et al. 2022b, ApJ, 927, 85
Sobol', I. M. 1967, USSR Comput. Math. Math. Phys., 7, 86
Springel, V. 2005, MNRAS, 364, 1105

Springel, V. 2010, MNRAS, 401, 791

Springel, V., Pakmor, R., Pillepich, A., et al. 2018, MNRAS, 475, 676

Takada, M., & Bridle, S. 2007, NJPh, 9, 446

Takada, M., & Hu, W. 2013, PhRvD, 87, 123504

Teyssier, R. 2002, A&A, 385, 337

Tonegawa, M., Park, C., Zheng, Y., et al. 2020, ApJ, 897, 17

Villaescusa-Navarro, F., Genel, S., Angles-Alcazar, D., et al. 2021a, arXiv:2109.10360

Villaescusa-Navarro, F., Anglés-Alcázar, D., Genel, S., et al. 2021b, ApJ, 915, 71

Villaescusa-Navarro, F., Genel, S., Anglés-Alcázar, D., et al. 2023, ApJ, 265, 54

Villaescusa-Navarro, F., Genel, S., Anglés-Alcázar, D., et al. 2022, ApJS, 259, 61

Villaescusa-Navarro, F., Wandelt, B. D., Anglés-Alcázar, D., et al. 2020, arXiv:2011.05992

Villanueva-Domingo, P. 2022, PabloVD/CosmoGraphNet, v1.0, Zenodo, doi:10.5281/zenodo.6485804

Villanueva-Domingo, P., & Villaescusa-Navarro, F. 2022, ApJ, 937, 115

Villanueva-Domingo, P., Villaescusa-Navarro, F., Anglés-Alcázar, D., et al. 2022, ApJ, 935, 30

Wadekar, D., Thiele, L., Hill, J. C., et al. 2023, MNRAS, 522, 2628

Wadekar, D., Villaescusa-Navarro, F., Ho, S., & Perreault-Levasseur, L. 2020, arXiv:2012.00111

Weinberger, R., Springel, V., Hernquist, L., et al. 2017, MNRAS, 465, 3291

Weinberger, R., Springel, V., & Pakmor, R. 2020, ApJS, 248, 32

Yu, H.-R., Harnois-Dé raps, J., Zhang, T.-J., & Pen, U.-L. 2012, MNRAS, 421, 832