The Astrophysical Journal, 935:30 (15pp), 2022 August 10 https://doi.org/10.3847/1538-4357/ac7aa3

Inferring Halo Masses with Graph Neural Networks

Pablo Villanueva-Domingo[1], Francisco Villaescusa-Navarro[2,3], Daniel Anglés-Alcázar[2,4], Shy Genel[2,5], Federico Marinacci[6], David N. Spergel[2,3], Lars Hernquist[7], Mark Vogelsberger[8], Romeel Dave[9,10,11], and Desika Narayanan[12,13]

[1] Instituto de Física Corpuscular (IFIC), CSIC-Universitat de València, E-46980, Paterna, Spain; pablo.villanueva.domingo@gmail.com
[2] Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY, 10010, USA; fvillaescusa@flatironinstitute.edu
[3] Department of Astrophysical Sciences, Princeton University, Peyton Hall, Princeton NJ 08544, USA
[4] Department of Physics, University of Connecticut, 196 Auditorium Road, U-3046, Storrs, CT 06269-3046, USA
[5] Columbia Astrophysics Laboratory, Columbia University, New York, NY, 10027, USA
[6] Dipartimento di Fisica e Astronomia "Augusto Righi," Università di Bologna, via Gobetti 93/2, I-40129, Bologna, Italy
[7] Center for Astrophysics—Harvard & Smithsonian, 60 Garden St, Cambridge, MA 02138, USA
[8] Kavli Institute for Astrophysics and Space Research, Department of Physics, MIT, Cambridge, MA 02139, USA
[9] Institute for Astronomy, University of Edinburgh, Royal Observatory, Edinburgh EH9 3HJ, UK
[10] Department of Physics & Astronomy, University of the Western Cape, Cape Town 7535, South Africa
[11] South African Astronomical Observatories, Observatory, Cape Town 7925, South Africa
[12] Department of Astronomy, University of Florida, Gainesville, FL, USA
[13] University of Florida Informatics Institute, 432 Newell Drive, CISE Bldg E251, Gainesville, FL, USA
Received 2021 December 9; revised 2022 June 17; accepted 2022 June 19; published 2022 August 10

Abstract

Understanding the halo–galaxy connection is fundamental in order to improve our knowledge on the nature and properties of dark matter. In this work, we build a model that infers the mass of a halo given the positions, velocities, stellar masses, and radii of the galaxies it hosts. In order to capture information from correlations among galaxy properties and their phase space, we use Graph Neural Networks (GNNs), which are designed to work with irregular and sparse data. We train our models on galaxies from more than 2000 state-of-the-art simulations from the Cosmology and Astrophysics with MachinE Learning Simulations project. Our model, which accounts for cosmological and astrophysical uncertainties, is able to constrain the masses of the halos with a ∼0.2 dex accuracy. Furthermore, a GNN trained on a suite of simulations is able to preserve part of its accuracy when tested on simulations run with a different code that utilizes a distinct subgrid physics model, showing the robustness of our method. The PyTorch Geometric implementation of the GNN is publicly available on GitHub (https://github.com/PabloVD/HaloGraphNet).

Unified Astronomy Thesaurus concepts: Cosmology (343); Dark matter distribution (356); Galaxy dark matter halos (1880); Hydrodynamical simulations (767); Neural networks (1933)

1. Introduction

In 1933 Fritz Zwicky found out that the mass of the Coma cluster should be much larger than the one from its luminous component (Zwicky 1933). That finding pointed to the existence of an unknown type of nonluminous matter: dark matter (DM). The requirement of unseen matter was later supported by the observation of rotation curves of galaxies (Rubin et al. 1978; Bosma 1978). Nowadays, there is overwhelming evidence for the existence of dark matter, although we are still ignorant of its fundamental properties (Young 2017).

We now believe that dark matter is the backbone of the distribution of matter in the universe: it concentrates in high-density regions called halos, which are connected by thin filaments with intermediate density and surrounded by gigantic regions with low density (voids). It is within halos that gas can cluster, cool, and form stars and galaxies (Somerville & Davé 2015). Dark matter halos are therefore the environment where galaxies reside.

Understanding the halo–galaxy connection is fundamental to improve our knowledge on the nature and properties of dark matter. There are two possible directions to take in the halo–galaxy connection. On one hand, given a dark matter halo and its properties and environment, predict the number and distribution of galaxies it hosts. This task is fundamental in order to create galaxy mocks with the correct clustering on all scales needed for forward modeling approaches (Wechsler & Tinker 2018). On the other hand, given a set of galaxies, it may be useful to determine some properties of their host halo such as its mass, spin, and concentration. This task is fundamental to derive cosmological constraints from the abundance of dark matter halos.

There has been an extensive program of weighing the masses of halos and clusters, using a wealth of techniques including gravitational lensing (Mandelbaum 2015; Huang et al. 2020, 2022), rotation curves of galaxies (Sofue & Rubin 2001; Sofue 2015), abundance matching (Behroozi et al. 2010), Sunyaev-Zeldovich effect (Grego et al. 2001), X-rays observations (Landry et al. 2013), velocity dispersion (Saro et al. 2013), and kinematics of satellite galaxies (Wojtak & Mamon 2013; Seo et al. 2020) among others; see, e.g., Old et al. (2015) for a comparison of different techniques for galaxy clusters.

All the above techniques do not make use of all available information. For instance, the abundance matching technique only considers the total stellar mass in the system, disregarding information about its clustering state. In this work, we attempt

to build a model that can use all available information from observations and/or simulations, e.g., phase-space information, stellar masses, galaxy sizes, to infer the mass of the halo hosting the galaxies. For this, we made use of neural networks and their capacity as universal function approximators.

Different machine-learning (ML) algorithms have been already used to perform this task. For instance, Convolutional Neural Networks (CNNs) have been already applied in order to predict dynamical galaxy cluster masses (Ho et al. 2019; Ntampaka et al. 2019; Gupta & Reichardt 2020; Yan et al. 2020; Kodi Ramanah et al. 2020; de Andres et al. 2021; Kodi Ramanah et al. 2021), as well as other ML techniques (Ntampaka et al. 2015, 2016; Haider Abbas 2019; Armitage et al. 2019; Green et al. 2019). Galaxy and subhalo masses can be inferred from different subhalo properties via multilayer perceptrons (Shao et al. 2022) or other ML algorithms (von Marttens et al. 2021), as well as from images with CNNs (de los Rios et al. 2021). Other works have employed different ML algorithms to predict the mass of a halo from the properties of the halo and galactic group (Man et al. 2019; Calderon & Berlind 2019; Lucie-Smith et al. 2020). These ML-based approaches have been shown to outperform other traditional techniques to infer halo masses. However, some of these works make use of N-body computations plus semianalytical galaxy formation models, rather than more accurate hydrodynamical simulations. Moreover, several of the features employed may not be easily observable, which could complicate their applicability to real data.

Although these approaches make use of the global properties of a halo as well as individual features of the galaxies, they do not incorporate explicitly the relationship between galaxies or subhalos, in the form of clustering in configuration space and/or distribution in phase space. In this article, we aim at predicting halo masses exploiting the halo–galaxy connection, using a novel method based on Graph Neural Networks (GNNs). This type of neural network shares the typical training procedure of other deep learning techniques, but it is applied to data structured in the form of mathematical graphs. To understand their significance, it is useful to compare them to other deep learning frameworks. CNNs are mostly employed with regular data (grids), such as images and 3D grids; CNNs automatically account for translational invariance. Recurrent neural networks, on their side, are designed to treat sequential data, such as chains of characters in a natural language or time series. However, GNNs can be applied when dealing with irregular data, where data points may have arbitrary relations.[14] That is the case of point clouds, as a galaxy catalog can be regarded. GNNs have been successfully employed in different fields such as chemistry, computer vision, natural language processing, social networks, or particle physics (Wu et al. 2019; Bronstein et al. 2021). There are already some applications of GNNs in cosmology, for instance in order to perform symbolic regression (Cranmer et al. 2019, 2020), to predict the redshift of galaxies (Beck & Sadowski 2019), or to allocate resources in an unsupervised way in order to select galaxies (Cranmer et al. 2021). GNNs have also been applied in other physics fields, such as particle physics (Shlomi et al. 2020), but still represent a novel, promising, and mostly unexplored way to extract information from irregular data.

[14] See Battaglia et al. (2018) for a comparison of the different deep learning components and their relational inductive biases.

We train our GNNs using galaxies from simulations of the Cosmology and Astrophysics with MachinE Learning Simulations (CAMELS) project (Villaescusa-Navarro et al. 2021) to extract information from the galaxy properties and their phase-space distribution. Since CAMELS contains thousands of state-of-the-art (magneto)hydrodynamic simulations with different values of the cosmological and astrophysical parameters, our method accounts for uncertainties in cosmology and astrophysics. Furthermore, since CAMELS contains two different suites of hydrodynamic simulations run with two different codes that employ different subgrid physics, we can quantify the robustness of our results to astrophysical uncertainties. Ultimately, we would like to build a tool that can infer the mass of galaxy systems, like our own Milky Way and Andromeda, just from the observed properties of those galaxies and their satellites. Knowing the mass of the host dark matter halos of those systems will allow us to make consistency checks within the ΛCDM model.

The article is structured as follows. We start by reviewing the basics on graphs and GNNs in Section 2. In Section 3, we describe the data we use to train our model together with an outline of the training procedure. The main results of this work are presented in Section 4. Some aspects of the interpretability of the GNNs are examined in Section 5, followed by a discussion of the main conclusions in Section 6.

2. Graph Neural Networks

In this section, we review the basics of GNNs, first summarizing the fundamentals of graphs, subsequently detailing how to build graphs from the galaxies of halos, and finally introducing how to build a neural network on a graph based on the message passing scheme. We refer the reader to Bronstein et al. (2021), Battaglia et al. (2018), Hamilton (2020) for comprehensive references on GNNs and geometric deep learning.

2.1. General Concepts on Graphs

We start by discussing some generic concepts of graphs and standard definitions, since some astrophysicists, cosmologists, and ML practitioners may not be familiar with the terminology. A *graph* can be defined as a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of *nodes*, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of *edges*. For two nodes of the graph $i, j \in \mathcal{V}$, there is an edge connecting them if $(i, j) \in \mathcal{E}$. Nodes i, j are thus coined as *neighbors*. The connectivity can be described by the *adjacency matrix* A_{ij}, which takes value of 1 if the pair (i,j) is connected by an edge, being 0 otherwise. A graph is *undirected* when $(i, j) \in \mathcal{E}$; then $(j, i) \in \mathcal{E}$ too. A *loop* is an edge, which connects a node to itself. We restrict our discussion to *simple* graphs, i.e., undirected graphs without loops, since these are enough for our purposes (although self-loops can be implicitly employed in some GNN architectures).

Given a node i, we denote its *feature vector* as $\boldsymbol{x}_i \in \mathbb{R}^{n_{\rm in}}$, encoding the relevant physical information about the node, with $n_{\rm in}$ as the number features. The *feature matrix* $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{|\mathcal{V}|})^T \in \mathbb{R}^{|\mathcal{V}| \times n_{\rm in}}$, where $|\mathcal{V}|$ is the number of nodes in the graph, comprises the feature vectors of all nodes. The *neighborhood* of the node $i \in \mathcal{V}$, denoted by $\mathcal{N}_i$, includes every node that shares an edge with i, and can thus be written as

$$\mathcal{N}_i = \{j \mid A_{ij} = 1\}. \quad (1)$$

Table 1
Summary of Galactic Properties Employed to Train the GNNs

Symbol	Feature
$\boldsymbol{p}$	3D comoving position
v	Modulus of relative velocity
M_*	Stellar mass
R_*	Stellar half-mass–radius

Note that $\mathcal{N}_i$ does not include the node i, since we dismiss loops. Analogously, we can define the set of feature vectors of the neighbors of the node i as

$$\mathcal{X}_i = \{\boldsymbol{x}_j \mid j \in \mathcal{N}_i\}. \tag{2}$$

A graph is *complete* if every pair of nodes is connected by an edge, having thus $\binom{|\mathcal{V}|}{2} = |\mathcal{V}|(|\mathcal{V}| - 1)/2$ edges in total.[15] The edges set can hence be written as $\mathcal{E} = \mathcal{V} \times \mathcal{V} \setminus \{(i, i) \in \mathcal{V}\}$ (excluding loops). As will be shown later, most of the halos employed here lead to complete graphs.[16]

2.2. Halos as Graphs

The general terminology discussed above allows us to set up our problem. Consider a halo h with mass M_h that hosts several galaxies. We build a graph, denoted by $\mathcal{G}_h$, by considering all the (central and satellite) galaxies of the halo as the nodes of the graph. The feature vector of a node i, $\boldsymbol{x}_i$, will include information about the corresponding galaxy, namely the 3D comoving position, $\boldsymbol{p}$, the stellar mass, M_*, the modulus of the velocity, v, and the stellar half-mass–radius (i.e., the comoving radius containing half of the stellar mass in the galaxy), R_*. These galactic properties are listed in Table 1. Positions are expressed with respect to the center of the halo (chosen as the point with minimum gravitational potential energy), while velocities are written relative to the center of the mass frame of all particles.[17] The mass of the halo is taken as the total mass of all components (dark matter, gas, stars, and black holes) within a virial radius enclosing a density 200 times the critical density, usually known as M_{200c}.

Unlike other graph applications, such as, e.g., social networks, we are dealing with the distributions of particles not physically connected among them (sometimes known as *point clouds* in graph contexts), and thus the edges are not predetermined. There is not an unique way to build the edges between the nodes. Since gravity is a long-range force, every galaxy is affected by the rest, and all the nodes should be connected in some way. However, gravitational effects, such as tidal forces, are stronger for shorter distances. Hence, one can expect that the dynamics and properties of a given galaxy will be more affected by nearby companions rather than by distant ones. Different approaches can be followed to connect the graphs and build the neighborhood of each node. In our case, we create the edges from a node taking into account the neighbors within a fixed certain radius, which is taken as a hyperparameter. Its specific value is given from the optimization outlined in Section 3.2. A similar procedure could be considering the k nearest neighbors for each node. However, this approach does not properly reflect the effect of clustering. It may miss some neighbors in a clustered region that should be connected, and include others much farther away, which in principle should have less influence in the node. We have checked that the performance slightly worsens using this approach instead of considering the nodes within a certain distance. It is worth it to remark that we only consider the galaxies that are part of the same halo, excluding thus interlopers and explicit relations between halos.

Among the available halos in the CAMELS catalogs, we work only with those containing more than one galaxy, excluding therefore the halos without satellites. The halos excluded, typically very low-mass halos, form trivial single-node graphs, where the advantages of graph-structured data with edge connections between nodes is absent and therefore are not very interesting for our purposes here. We have explicitly checked that by including those halos without satellites our results worsen. Note also that this fact limits the application of our method to real observations of halos with more than one known satellite galaxy.

2.3. Message Passing

GNNs employ the so-called neighborhood aggregation or *message passing* scheme.[18] Each node i receives a *message* from each of its neighbors j, appending its information to its own features. This makes it possible to create a *hidden* feature vector $\boldsymbol{h}_i$, which updates the node and contains aggregated data from its neighborhood. From the hidden vectors, our GNN is aimed at inferring the mass of the halo, which is a global property of the graph. The architecture is thus composed of two main steps: (1) update the nodes via message passing, and (2) extract the global information from the graph. We designate as *Graph Layer*, GL, to a generic message passing layer, which can be written as

$$\boldsymbol{h}_i = \mathrm{GL}(\boldsymbol{x}_i, \mathcal{X}_i) = \bigoplus_{j \in \mathcal{N}_i} \psi(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3}$$

where $\boldsymbol{h}_i$ is the output feature vector, $\bigoplus$ denotes a differentiable, permutation invariant *aggregation function*, such as the maximum, the mean, or the sum, while ψ is the *message function*, a differentiable function, like a Multilayer Perceptron (MLP), which depends upon the feature vector of the node, $\boldsymbol{x}_i$, as well as on those from its neighbors, $\boldsymbol{x}_j$, for $j \in \mathcal{N}_i$. The distinct ways to build ψ lead to different graph layers.[19]

The function ψ maps ψ: $\mathbb{R}^{n_{\mathrm{in}}} \rightarrow \mathbb{R}^{n_{\mathrm{out}}}$, with n_{in} and n_{out} as the number of initial and output features, respectively. As shown in Section 2.4, n_{in} may not correspond to the number of features considered for each node. A common advantage of GNNs is that they exploit the locality of data, because the closer nodes may present stronger relations than those farther away. Here,

[15] This can be reasoned as the number of pairs that can be chosen from a total of $|\mathcal{V}|$ elements.

[16] Complete graphs can be very relevant in ML applications, since the popular architecture transformers (Vaswani et al. 2017) can be regarded as a particular case of a GNN, a Graph Attention Network (Veličković et al. 2018), where the graph is complete (see Bronstein et al. 2021 for more details).

[17] We have checked that using only distances rather than 3D positions significantly worsens the performance of the GNN, because they do not provide a complete description of the spatial distribution of the system, which can be relevant for the halo–galaxy connection.

[18] There are other similar frameworks, such as the Graph Network formalism (Battaglia et al. 2018), which is mostly equivalent to our message passing scheme.

[19] One could also apply an additional MLP ξ to $\boldsymbol{h}_i$ and its original feature vector, $\boldsymbol{h}_i{}' = \xi([\boldsymbol{x}_i, \boldsymbol{h}_i])$, where the square brackets indicate array concatenation, although we dismiss this step for the sake of simplicity, and since we apply several successive graph layers, which leads to a similar effect.

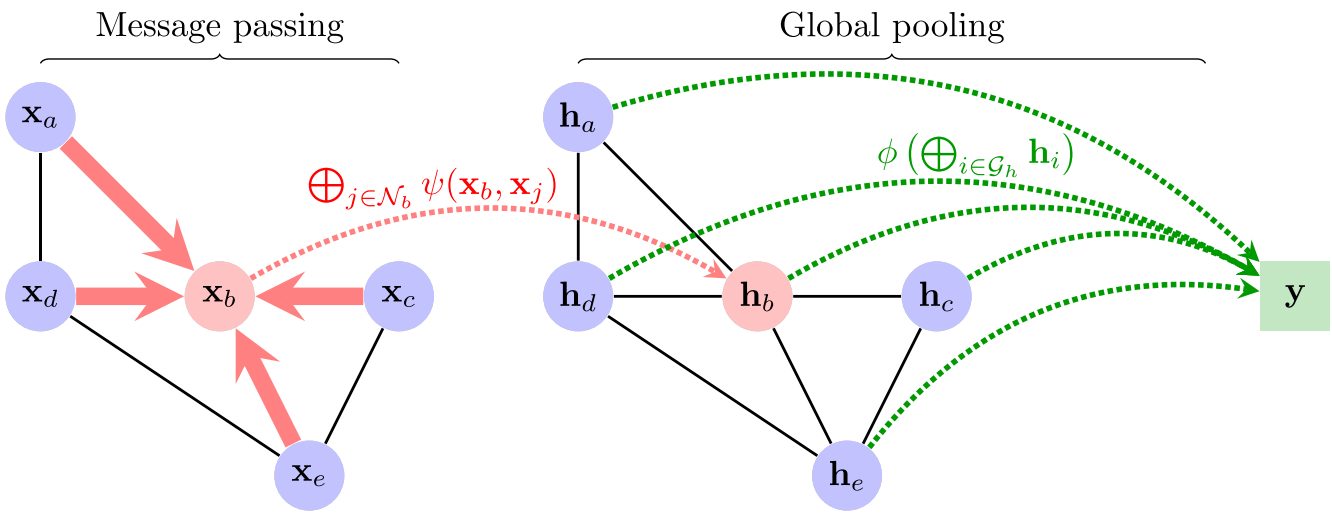

Figure 1. Sketch of a GNN with one message passing layer acting on a graph $\mathcal{G}_h$ composed of 5 nodes. During the message passing step, node b receives a message from each of its neighbors $j \in \mathcal{N}_b$, aggregating and updating its features to the hidden vector $\boldsymbol{h}_b$ (red). Once this step is done over all nodes, a global pooling step is performed, aggregating the hidden node features over all vertices of the graph $\mathcal{G}_h$, which leads to the global target $\boldsymbol{y}$ (green). Figure based on https://github.com/PetarV-/TikZ.

the locality is enforced by assuming that the functions ψ are the same for all nodes and graphs.[20]

The final GNN incorporates a last step to infer the global graph target, and thus can be of the form

$$\boldsymbol{y} = \phi\left(\bigoplus_{i\in\mathcal{G}_h}\boldsymbol{h}_i\right), \tag{4}$$

where ϕ is a differentiable function, chosen as a MLP with three hidden layers with 300 channels separated by ReLu activation functions, and $\boldsymbol{y}$ is the output of the network. In our case, since we perform likelihood-free inference to estimate the mean and standard deviation of the posterior of the mass of the halo, the target is given by a vector of two components $\boldsymbol{y} = [y, \sigma]$, containing the mean prediction y, and its expected standard deviation, σ. The targeted quantity is the logarithm of the mass of the halo, $y = \log_{10}[M_h/(M_\odot/h)]$. The likelihood-free approach to estimate the posterior mean and standard deviation is detailed in Section 3.2. The equation above can be easily modified to include some global graph quantities u, which may help to train the network, such as the number of nodes or the total stellar mass, writing $\boldsymbol{y} = \phi([\bigoplus_{i\in\mathcal{G}_h}\boldsymbol{h}_i, u])$. We include these global features in our graphs, although we have checked that removing them does not leave a significant impact in the results. Figure 1 shows a sketch of this basic architecture, illustrating the message passing scheme.

Permutation equivariance and invariance are key concepts in GNNs. We say that a function acting on the node feature matrix $f(\boldsymbol{X})$ is permutation invariant if for every permutation matrix $\boldsymbol{P}$, one has $f(\boldsymbol{PX}) = f(\boldsymbol{X})$, leaving thus an unchanged output. On the other hand, f is permutation equivariant if it transforms as $f(\boldsymbol{PX}) = \boldsymbol{P}f(\boldsymbol{X})$.[21] A message passing layer $\mathrm{GL}(\boldsymbol{x}_i, \mathcal{X}_i)$ should be permutation equivariant, since a reordering of the input nodes produces the same permutation in the outputs, although the output feature space can be different. However, any global quantity of the graph such as the final output of the network, the halo mass, must be permutation invariant, since its value should not depend on the ordering of the nodes. The final aggregation step, $\bigoplus_{i\in\mathcal{G}_h}\mathrm{GL}(\boldsymbol{x}_i, \mathcal{X}_i)$, can be regarded as a function $f(\boldsymbol{X})$ that fulfills by the construction permutation invariance, $f\colon \mathbb{R}^{|\mathcal{V}|\times n_{\rm out}} \to \mathbb{R}^{n_{\rm out}}$. In this case, the aggregation $\bigoplus$ may be termed as a global pooling layer, since it reduces dimensionality. Note that the aggregation method here does not have to be the same as in the message passing layer. Furthermore, it is possible to employ different aggregation operators, such as maximum, sum, and mean, and concatenate them, which is the approach used in this work. The last MLP ϕ makes use of $n_{\rm out}$ global features of the graph in order to extract a global property vector $\boldsymbol{y}$, the halo mass, and its expected standard deviation in our case, $\phi\colon \mathbb{R}^{n_{\rm out}} \to \mathbb{R}^2$. Finally, we can apply the multiple successive GNN layers, $\mathrm{GL}(\mathrm{GL}(\boldsymbol{x}_i, \mathcal{X}_i), \mathcal{H}_i)$, where $\mathcal{H}_i = \{\boldsymbol{h}_j \mid j \in \mathcal{N}_i\}$ denotes the updated feature vector of the neighbors. In such a case, the node after k updates will encode information from the nodes of its k-hop neighborhood. We take the number of message passing layers as an optimizable hyperparameter, although a single GNN layer is sufficient to provide the best results, as shown in Section 3.2.

2.4. Graph Layer Architecture

So far we have discussed the general shape of the GNN, but the explicit design of the message passing layer remains undetermined. There are many possible ways to construct the specific architecture of the GNN layer, which is defined by the message function ψ and the aggregation scheme $\bigoplus$. For instance, one could not take into account neighbors to update each node, and employ only the information of the node $\boldsymbol{x}_i$ to extract the hidden information $\boldsymbol{h}_i$. These types of architectures, a subset of GNNs, are known as DeepSets (Zaheer et al. 2017). No aggregation function is thus needed to update the node features, and the hidden layer can just be written as $\boldsymbol{h}_i = \psi(\boldsymbol{x}_i)$.

However, to fully exploit the relations among the nodes, it is useful to incorporate the neighborhood and edge information actually performing message passing. Here we assume an architecture based on the edge convolutional layer, coined as EdgeNet (Wang et al. 2019), where for each neighbor j of the

[20] Similarly, CNNs in, e.g., computer vision tasks guarantee locality by employing the same kernels over all different images.
[21] See Bronstein et al. (2021) for more details on invariances on graphs.

node i, the relative vectors $\boldsymbol{x}_i - \boldsymbol{x}_j$ are concatenated to the feature vector $\boldsymbol{x}_j$. The aggregation function is chosen as the maximum, writing thus the hidden layer as

$$\boldsymbol{h}_i = \max_{j \in \mathcal{N}_i} \psi([\boldsymbol{x}_i, \boldsymbol{x}_i - \boldsymbol{x}_j]), \tag{5}$$

where the square brackets indicate an array concatenation. In this case, the initial input number is given by $n_{\rm in} = 2n_{\rm feat}$, where $n_{\rm feat}$ is the number of features. The differentiable function ψ is taken as a MLP with three hidden layers, with 300, 300, and 100 hidden channels, separated by ReLu activation functions. We have checked that other choices of the number of channels do not improve the predictions of the net.

There are other popular architectures, such as PointNet (Qi et al. 2017a, 2017b), which employs only the relative spatial positions for the message passing, $\boldsymbol{p}_i - \boldsymbol{p}_j$, rather than the full feature vector, or the Graph Convolutional Network (Kipf & Welling 2017), where the neighbor information is simply incorporated by taking a linear combination of the features and summing over all the neighbors, imitating a convolution operation. We have checked that the EdgeNet (Equation (5)) outperforms the other architectures mentioned above via a hyperparameter optimization procedure, as commented in Section 3.2. See, e.g., Wu et al. (2019), Zhou et al. (2018) for a discussion of other different GNN architectures.[22]

3. Methods

In this section, we specify the details regarding the data employed and the training of the network.

3.1. The CAMELS Simulations

The CAMELS project (Villaescusa-Navarro et al. 2021, 2022) comprises a set of state-of-the-art hydrodynamic and N-body simulations, specially suited and designed to train and test ML algorithms. They include thousands of realizations varying two cosmological parameters, namely the matter density parameter Ω_m, and the variance of the linear field on 8 Mpc$/h$ at $z = 0$, σ_8, plus four astrophysical parameters controlling the efficiency of supernovae and active galactic nuclei (AGN) feedback. The rest of the cosmological parameters are kept fixed to standard values in flat cosmologies with cosmological constant: the baryonic fraction $\Omega_b = 0.049$, the reduced Hubble constant $h = 0.6711$, the tilt of the primordial power spectrum $n_s = 0.9624$, the equation of state of the dark energy (corresponding to a cosmological constant) $w = -1$, and the total neutrino mass $M_\nu = 0$ eV. Each simulation follows the evolution from $z = 127$ to $z = 0$ of 256^3 DM particles and 256^3 gas resolution elements within a periodic box of size 25 Mpc $/h$. The DM particle mass resolution is $\sim 10^8 M_\odot/h$. A collection of different cosmological fields in the form of 2D maps and 3D grids from the CAMELS simulations is available as the CAMELS Multifield Dataset (CMD),[23] intended to be a standard cosmological data set for ML applications (Villaescusa-Navarro et al. 2022), while the full data set has been recently made publicly available (Villaescusa-Navarro et al. 2022).

The CAMELS project includes two suites of simulations, with different astrophysics modeling and subgrid physics. On the one hand, a suite of magnetohydrodynamic simulations with the subgrid physics models of IllustrisTNG (Weinberger et al. 2017; Pillepich et al. 2018; Nelson et al. 2019) performed with the code Arepo[24] (Weinberger et al. 2020; see also Springel et al. 2018; Naiman et al. 2018; Marinacci et al. 2018 for more details). Its galaxy formation model is based on the predecessor Illustris (Vogelsberger et al. 2013, 2014). On the other hand, another suite employs the SIMBA subgrid physics model (Davé et al. 2019), making use of the code GIZMO[25] (Hopkins 2015). SIMBA is built on its precursor MUFASA (Davé et al. 2016) with the addition of supermassive black hole growth and feedback (Anglés-Alcázar et al. 2017). The hydrodynamics simulations are accompanied by N-body counterparts, run with the GADGET-III code (Springel 2005).[26]

Within each simulation suite, there are different sets, according to the configuration of astrophysical and cosmological parameters. The cataclysmic variable (CV) simulations (standing for Cosmic Variance) include 27 simulations sharing their astrophysical and cosmological parameters, fixed to standard values ($\Omega_m = 0.3$ and $\sigma_8 = 0.8$), and varying only the random seed to generate the initial conditions. The Latin-Hypercube (LH) set consists of 1000 simulations, varying all the astrophysical and cosmological parameters (together with the random seed) along a LH.[27] We make use of both sets for training our networks, in order to check whether the architectures considered are robust enough for the different cosmologies and baryonic feedback parameters. The halos and subhalos are identified using the SUBFIND algorithm (Springel et al. 2001). We define galaxies as subhalos that contain more than 20 star particles. Although the CAMELS suite contains data for several redshifts, only the simulations at $z = 0$ are considered in this work. We refer the reader to the CAMELS webpage[28] and Villaescusa-Navarro et al. (2021) for further details.

As an example of the differences between IllustrisTNG and SIMBA, Figure 2 shows the number of halos as a function of the number of galaxies that the halos contain for the simulations of the LH set. While most of the halos in the simulations only host a few galaxies, there are a number of them that contain hundreds of satellites. Note that, for a fixed halo mass, the SIMBA simulations contain more galaxies than the IllustrisTNG simulations. In general, SIMBA simulations usually contain many more galaxies than their IllustrisTNG counterpart, being an average of $\sim$1200 and $\sim$700 per simulation respectively, both at $z = 0$. Furthermore, SIMBA simulations tend to be more stochastic, and usually can produce more extreme scenarios (with larger effects of the baryonic feedback effects) than IllustrisTNG. These facts reflect the inherent differences between the two codes/subgrids models. We refer the reader to Villaescusa-Navarro et al. (2021) for a more detailed comparison between the different observables of both suites, such as clustering or halo population, and to Villaescusa-Navarro et al. (2022), Villanueva-Domingo & Villaescusa-Navarro (2022) for more information on the

[22] See also https://github.com/thunlp/GNNPapers for a comprehensive list of GNN references.
[23] https://camels-multifield-dataset.readthedocs.io
[24] https://arepo-code.org/
[25] http://www.tapir.caltech.edu/~phopkins/Site/GIZMO.html
[26] See, e.g., Vogelsberger et al. (2020) for a comparison of different models of galaxy formation in cosmological simulations.
[27] Besides CV and LH, other simulation sets are also included in CAMELS, but not employed in this work.
[28] https://camels.readthedocs.io

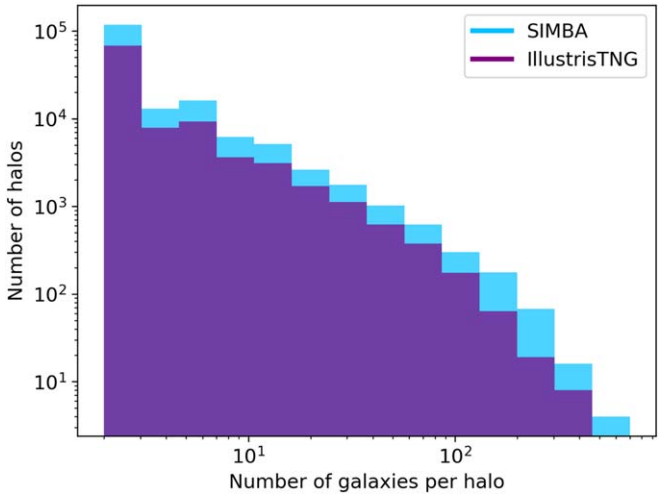

Figure 2. Number of halos sorted by the number of galaxies that they host in the IllustrisTNG (blue) and SIMBA (purple) suites, for the LH sets. On average, the halos of the SIMBA simulations contain more galaxies than the halos of the IllustrisTNG simulations, reflecting the large differences between the two simulation suites given their distinct subgrid physics models.

distribution and correlations between the different galactic features of the CAMELS catalogs.

3.2. Training Procedure

We construct our data set, the collection of graphs, based on the procedure outlined in Section 2.2. We restrict our analysis to the halos with more than one galaxy, as mentioned above. The models are trained on simulations of the CV and LH sets separately, where in each case we split the data set into training (70%), validation (15%), and testing (15%) sets. We employ an Adam optimizer (Kingma & Ba 2014) and L2 regularization. The batch size is set to 128, and the number of epochs is limited to 150. The GNN architecture is implemented following the prescription outlined in Section 2.3, concatenating one or several message passing layers and appending at the end a global pooling and MLP to infer the halo mass.

We perform a hyperparameter optimization in order to get the best values for the hyperparameters, following a bayesian model-based optimization procedure with the Tree Parzen Estimator (Bergstra et al. 2011), making use of the Python package Optuna[29] (Akiba et al. 2019). The hyperparameters considered for this optimization are the learning rate, the weight decay, the number of message passing layers, the distance to define neighborhoods, and the specific architecture (among those defined in Section 2.4). We perform at least 75 trials for each suite and set, where each trial is a specific choice of the values of the hyperparameters. The optimization procedure leads to different values for each simulation suite and set, with the learning rates ranging between 10^{-5} and 6×10^{-4}, the weight decay values between 10^{-8} and 10^{-7}, and the neighborhood distance between 2 and 20 Mpc/h. One unique message passing layer (i.e., one update of the hidden features of the nodes from Equation (3)) and the EdgeNet architecture (among those discussed in Section 2.4) are the optimal choices for all cases. The large values of the neighborhood radii obtained imply that most of the graphs created are complete, around 98% for both the CV and LH sets. That means that most of the galaxies are connected with each other within the halo, and most nodes can access information about each other's companions, from the graph, at one hop.

We follow a likelihood-free bayesian inference approach to sample the expected standard deviation of the outputs. This procedure allows us to reproduce some properties of the posterior (its second centered moment in this case) without the need of a likelihood (Jeffrey et al. 2021). We follow Jeffrey & Wandelt (2020) and design our model to output two quantities: the mean and standard deviation of the halo mass posterior. The loss function needed to achieve that is given by $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$, where

$$\mathcal{L}_1 = \log\left[\sum_{i\in\text{batch}} (y_{\text{truth},i} - y_{\text{infer},i})^2\right] \tag{6}$$

and

$$\mathcal{L}_2 = \log\left[\sum_{i\in\text{batch}} ((y_{\text{truth},i} - y_{\text{infer},i})^2 - \sigma_i^2)^2\right]. \tag{7}$$

Note that we have included log functions to make sure that the two contributions are similar. For instance, if the errors are much smaller than the mean, the loss will be dominated by the mean, and the errors may not be accurately computed (see Villaescusa-Navarro et al. 2022 for further details).

It is worth emphasizing the symmetries fulfilled by the GNNs. By construction, GNNs are permutation invariant, guaranteed by neighborhood aggregation as discussed in Section 2.3. Furthermore, given that graphs are written in the center of mass rest frame, our framework is also translational invariant, since any displacement of the galaxy group would not alter the relative coordinates with respect to the center, and hence the graph would be the same. Moreover, for our special case of point clouds, our GNNs should be rotationally invariant, since an arbitrary rotation of all galaxies around the center of the halo should not change the global halo properties. We try to enforce this symmetry by performing random rotations on each graph at every training epoch, as a data augmentation procedure. This practice also helps in alleviating overfitting, given that the network becomes robust when tested in systems with arbitrary rotations. Overfitting also becomes absent in our training procedure with the proper selection of the hyperparameters.

We write and train the models making use of PyTorch Geometric[30] (Fey & Lenssen 2019). Our implementation of the GNNs, `HaloGraphNet`, is publicly available on GitHub[31] (Villanueva-Domingo 2020).

4. Results

In this section, the main results of the work are discussed. We first introduce a benchmark model to estimate halo masses from stellar masses, and next we detail the results of training and testing the GNNs in the different CAMELS simulation sets and suites considered, examining also their robustness over different astrophysical modeling.

[29] https://optuna.readthedocs.io

[30] https://pytorch-geometric.readthedocs.io

[31] https://github.com/PabloVD/HaloGraphNet

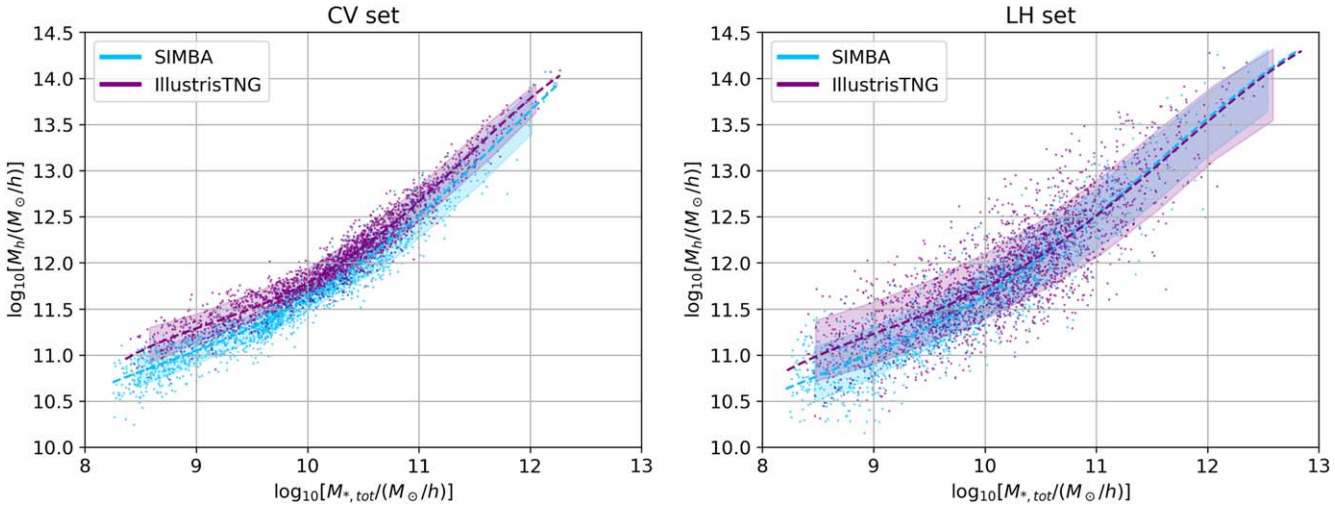

Figure 3. Halo masses with respect to the total stellar mass within, both for SIMBA and IllustrisTNG suites, for samples of 2000 halos from the CV (left) and LH (right) sets. Shaded areas denote standard deviation of points while dashed lines correspond to polynomial fits. The LH set covers a large astrophysical and cosmological parameter space, leading to a broader scatter in the masses with respect to the CV set, where parameters are fixed. While such fits can lead to accurate predictions of the halo mass for the CV set, it worsens in the LH set, given the larger dispersion.

4.1. Predictions from Stellar Masses

Before starting to discuss the accuracy of the GNN predictions, it may be useful to set a benchmark model to predict halo masses, to test the worthiness and degree of improvement of using GNNs. A traditional approach exploits the relation between stellar and halo mass, based on abundance matching (Behroozi et al. 2010; Wechsler & Tinker 2018). Figure 3 shows the mass of a halo M_h as a function of its total stellar mass $M_{*,tot} = \sum_i M_{*,i}$ (summing over all stellar particles in the central and satellite galaxies within the halo) in the IllustrisTNG and SIMBA suites, for the CV (left) and the LH (right) sets.[32] One can notice a clear correlation between the stellar and halo masses, although with a large scatter. This is specially noteworthy in the LH set, where multiple values of the cosmological and astrophysical parameters are considered, leading to completely different outcomes. Shaded areas denote the standard deviation of the points, which is around $\sim$0.2 dex in the CV case, but grows up to $\sim$0.3–0.4 dex in the LH set.

Taking advantage of the expressed correlation, we can build a naive estimator of the halo mass based only on the total stellar mass of galaxies. A simple fourth-degree polynomial fit is shown in dashed lines in Figure 3. This can be regarded as a benchmark model for comparing with our forthcoming models based on GNNs, in order to further evaluate their strength. To check its accuracy for predicting $y = \log_{10}[M_h/(M_\odot/h)]$, we employ the mean relative error ϵ,

$$\epsilon = \frac{1}{N}\sum_i^N \frac{|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}|}{y_{\mathrm{truth},i}}, \tag{8}$$

with N the number of test halos, as well as the correlation coefficient (or coefficient of determination) R^2, defined in the usual way as

$$R^2 = 1 - \frac{\sum_i^N (y_{\mathrm{truth},i} - y_{\mathrm{infer},i})^2}{\sum_i^N (y_{\mathrm{truth},i} - \bar{y}_{\mathrm{truth}})^2}, \tag{9}$$

with $\bar{y}_{\mathrm{truth}}$ the mean of true values. This naive estimator gives fairly accurate results in the CV set, with relative errors around 1% and a linear correlation coefficient of $R^2 \simeq 0.94$. However, when an analogous fit is attempted in the LH case, the relative error worsens down to $\sim$1.7% ($\sim$2.4%) for SIMBA (IllustrisTNG), with $R^2 \simeq 0.84$ ($R^2 \simeq 0.67$). Note that these errors are for the logarithm of the mass, y, rather than for the halo mass itself, which corresponds to relative errors in the mass between $\sim$50%–120%. This fact illustrates the nontrivial dependence of the halo and stellar masses on the different astrophysical and cosmological scenarios. Notice that the LH set contains some extreme scenarios that may be unlikely to describe the real universe. In any case, the most appropriate astrophysical parameters to describe our cosmos are still unclear, and thus, models able to robustly marginalize over baryonic feedback effects are required. Thus, a prediction of the halo mass from only the stellar mass when a broad range of astrophysical models are considered seems to be quite inaccurate. In the following, we shall show how a GNN is able to overcome this difficulty by considering further features and taking advantage of the graph structure of halos.

4.2. Inferring Halo Masses with GNNs

Here we first discuss the results of training a GNN to predict the halo masses using the galaxies from simulations of the CV set. As stated in Section 3.1, this set contains 27 simulations with fixed fiducial values for the cosmological and astrophysical parameters, only varying the random seed. The left panels of Figure 4 show the accuracy of the network in the CAMELS CV sets, where the top panel stands for the IllustrisTNG subgrid model, and the bottom one stands for the SIMBA suite. The vertical axis is the difference between the predicted and true logarithms of the halo mass, $\log_{10}(M_h/(M_\odot/h))$, with respect to the true value in the horizontal axis. Error bars have been estimated via likelihood-free inference, sampling the standard deviation of the posterior, as outlined in Section 3.2. One can see that the performance is fairly good in both cases, as the linear correlation coefficient of

[32] Note that other works consider the stellar mass of the central galaxy rather than the total one, which is employed here with the aim to take into account the contribution of the satellites.

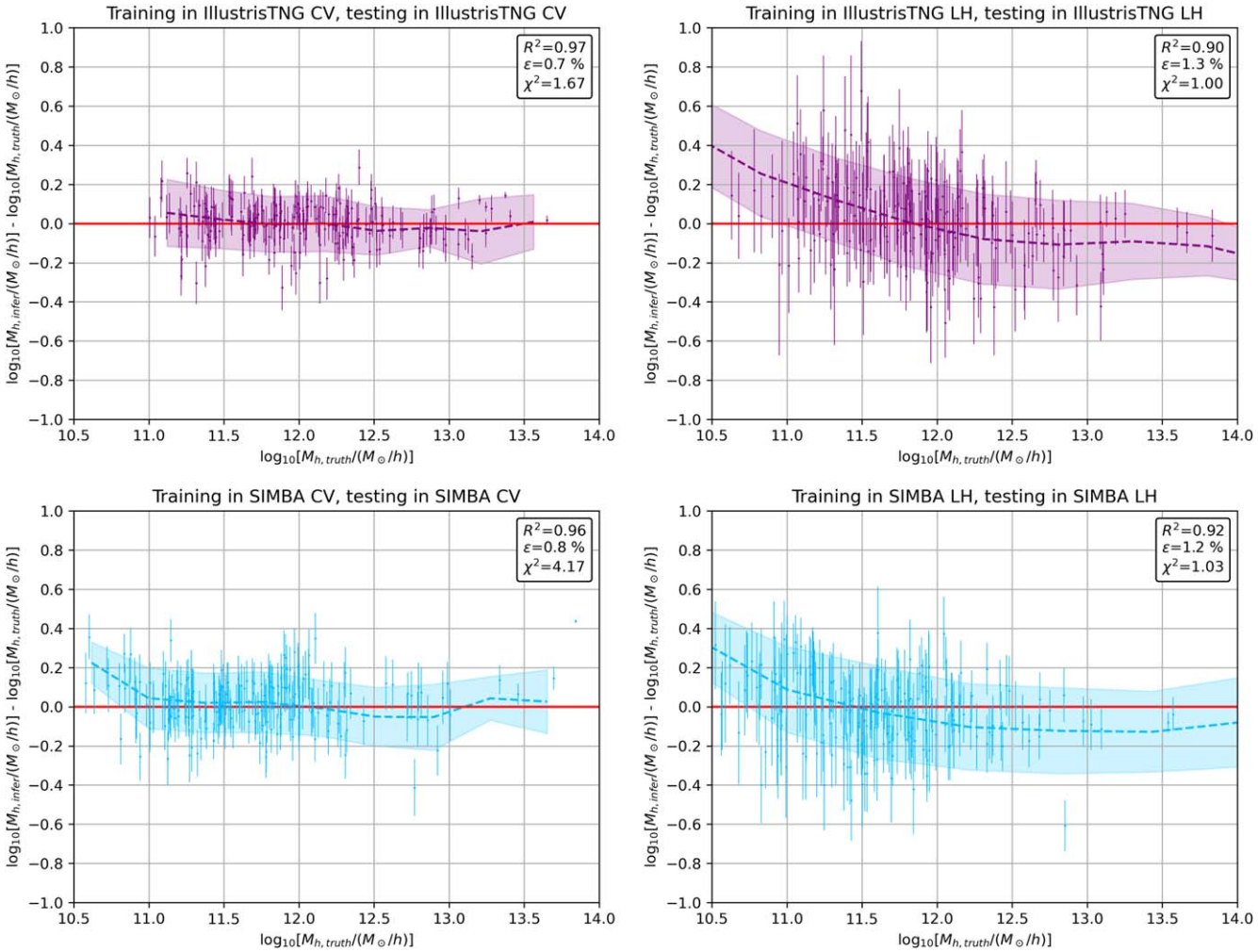

Figure 4. Predicted vs. true logarithm of halo masses $\log_{10}[M_h/(M_\odot/h)]$ for the CV (left) and LH sets (right), training in the IllustrisTNG suite (top) and in SIMBA (bottom). A sample of 200 halos in the test data set is shown in each case. Shaded regions and dashed lines correspond to real standard deviation and mean of test points, respectively. While in the CV set, astrophysical and cosmological parameters are fixed to fiducial values; the LH set comprises a broad range of astrophysical and cosmological scenarios. Even so, the GNN is still able to learn the halo–galaxy relation and predict masses in the LH case, only slightly worsening the prediction with respect to the CV case.

$R^2 = 0.96$–0.97 confirms, with a relative error lower than $\sim$1% in y ($\sim$25%–40% in the mass itself). The dashed lines show the mean of test points, while the shaded region depicts their actual standard deviation, which extends up to $\sim$0.14 dex. Thus, most of the test predictions lie within this region, although there are some outliers. The neural network is thus able to accurately infer the halo mass given some features of its galaxies.

Nevertheless, the previous results only show that the GNN is capable of predicting the mass of a halo when the cosmology and astrophysics is known, i.e., when the parameters are fixed. However, the specific values of the relevant parameters for the real universe are not well known yet, especially the astrophysical ones. In order to marginalize over uncertainties in cosmology and astrophysics, we have trained our network in the LH simulation set, which includes 1000 simulations varying the cosmological and astrophysical parameters, together with the random seed (to also incorporate effects of sample variance), as noted in Section 3.1. The right panels of Figure 4 show the GNN predictions for training the GNN in IllustrisTNG (top) and SIMBA (bottom). One can see that the results only slightly worsen from the equivalent case in the CV simulation set, with $R^2 = 0.90 - 0.92$ and relative errors of $\sim$1%. Uncertainties are also larger than in the CV set, roughly by a factor of 2, meaning that the model trained in the LH set offers lower precision. This worsening is expected since the network needs to marginalize over the astrophysical and cosmological parameters, differently from the network trained on the CV set. It should be noted that a small bias appears at low masses, below $\lesssim 10^{11} M_\odot/h$, which deviates up to 0.4 dex (also present in the SIMBA CV case). This could be explained due to the smaller number of low-mass halos in the data sets, preventing the network to properly learn in that range. Moreover, the expected fewer satellites in low-mass halos could also affect the predictions, because the GNN counts on less satellites to extract information. In any case, these results indicate that our model is able to learn the mapping between astrophysics/cosmology and the halo mass, and thus marginalize over the value of the parameters present in the simulations to make an accurate prediction.

To further evaluate the predictive power of GNNs, it is worth comparing these results to those obtained from the naive fit based only on stellar mass outlined in Section 4.1. For both the CV and LH sets, the GNN predictions outperform those from the polynomial fit, presenting larger R^2 coefficients and lower

relative errors. The improvement is especially clear in the LH case, where the R^2 is significantly better than the benchmark method. Moreover, the scatter around the true values spans $\sim$0.14 and $\lesssim$0.2 dex, compared to the larger standard deviations from the naive fit, which are 0.2 and 0.3–0.4 dex respectively. Note that 0.2 dex is a factor up to $\sim$1.6 in the mass (rather than in the logarithm), while 0.3 dex is a factor $\sim$2, meaning a $\sim$100% error in the mass. These results demonstrate how, taking advantage of the graph structure and further galaxy features, it is possible to attain richer correlations and better results.

Nevertheless, it has to be emphasized that the linear correlation coefficient R^2 and the relative error cannot constitute a complete statistical summary for testing the accuracy of the GNN. It is because our network is also predicting the standard deviation of the target, for which an additional component to the loss has been included, as discussed in Section 3.2. Thus, neither R^2 nor the relative error quantify the sampling accuracy of the standard deviation. To test whether the uncertainties are reasonably predicted, it is useful to compute the χ^2, defined as

$$\chi^2 = \frac{1}{N}\sum_i^N \frac{(y_{\mathrm{truth},i} - y_{\mathrm{infer},i})^2}{\sigma_i^2}\,. \tag{10}$$

Note that minimizing the loss function contribution from Equation (7) tends to drive the χ^2 toward unity. This is actually the case in the LH set, where $\chi^2 = 1.00$ for IllustrisTNG and $\chi^2 = 1.03$ for SIMBA, indicating that uncertainties are accurately predicted. In the CV cases, however, while the mean predictions are more accurate, some errors are underestimated, leading to larger χ^2 values, around $\sim$4 in SIMBA. Moreover, since the test data set is smaller, a few outliers with too small standard deviations greatly impact the value of the χ^2.

One can also compute how many points in the test data set present an accurate uncertainty by counting the fraction that fulfill the conditions $|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}| \leqslant \sigma_i$ and $|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}| \leqslant 2\sigma_i$, i.e., how many points lie within one and two times the standard deviation of the posterior. We find that, for the LH cases, the fraction of points fulfilling the above conditions is 69% and 95% respectively for both suites. For the CV sets, these fractions only slightly deviate, 62% and 90% for IllustrisTNG, and 72% and 96% for SIMBA, respectively. For the Gaussian distributed errors, these fractions should be around 68% and 95% respectively. Note however that our calculation of the posterior mean and standard deviation does not make any assumption about the form of the posterior. Therefore, the numbers quoted above should be taken with caution, and a comparison with the Gaussian case should be done in a careful manner.

There is another way to figure out whether uncertainties are correctly sampled. The shaded regions in Figure 4 represent the actual standard deviation of the test points computed within several mass bins. Therefore, if the predicted uncertainties are accurately sampled, their mean value $\overline{\sigma}$ should correspond to those shaded regions. For the CV case, despite the GNN providing more accurate models, the mean uncertainties $\overline{\sigma}$ are underpredicted with respect to the actual scatter by $\sim$40% for IllustrisTNG and $\sim$20% for SIMBA. However, in the LH case, the uncertainties are better sampled, only deviating $\sim$3% and $\sim$7% for IllustrisTNG and SIMBA respectively. This implies that models trained in LH, despite predicting slightly less accurate results than those from the CV set, provide a better sample of the posterior uncertainties.

4.3. Robustness over Different Subgrid Physics Models

Subgrid physics, i.e., the models used to simulate unresolved astrophysical processes, such as the feedback from supernovae and black holes, can only be implemented in a phenomenological way, and therefore there is not a unique subgrid model that best represents reality. Thus, having a ML model robust over different subgrid scenarios would be needed in order to obtain predictions that do not depend on the particular type of simulation used to train the networks.

To check whether our GNN fulfills this requirement, we have taken the model previously trained on simulations from the IllustrisTNG suite, and we have tested it on the galaxies from the SIMBA simulations, which make use of a completely different subgrid physics model for AGN and SN feedback. The top left panel of Figure 5 shows the predictions for the halo mass in the CV set, i.e., trained in IllustrisTNG CV and tested in SIMBA CV. We see that the performance becomes worse, and actually a bias appears. This offset may arise from the fact that the IllustrisTNG and SIMBA models make different predictions for the halo–galaxy connection, as shown in Figure 3. This can be related with the fact that astrophysical parameters are not completely correlated and calibrated between both cases. While the CV set assumes fiducial values for the astrophysical and cosmological parameters, the default values in the IllustrisTNG suite do not correspond to the ones in SIMBA, since they refer to different quantities and physics. The absence of a one-to-one relation between both suites in the CV set may explain why the network fails to make a robust prediction.

The right panel of Figure 5 depicts the results of testing the network trained on galaxies of the IllustrisTNG LH set on galaxies of the SIMBA LH set. In this case, the bias present in the CV case disappears, and only a broad scatter holds. The absence of the offset can be attributed to the intrinsic marginalization of astrophysical effects carried out by the network (although a slight tilt is present in the trend). Moreover, the linear correlation coefficient only slightly decreases with respect to testing in IllustrisTNG (top right panel of Figure 4), and is actually better than in the CV counterpart (top left panel of Figure 5), in spite of dealing with a much broader astrophysical parameter space. The χ^2 also remains closer to unity than in the CV case, meaning that the uncertainties are better sampled in the LH set. The fraction of points fulfilling the conditions $|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}| \leqslant \sigma_i$ and $|y_{\mathrm{truth},i} - y_{\mathrm{infer},i}| \leqslant 2\sigma_i$ is 65% and 93% respectively, while much lower for the CV case (20% and 46% respectively). These facts imply that the models trained in the LH set generalize better than those trained in the CV one.

An analogous experiment has been carried out, employing the model trained in the SIMBA suite and testing it in IllustrisTNG. The results are shown in the bottom panels of Figure 5 for the CV set (left) and LH set (right). Note the offset in the CV case, which is similar but opposite (underpredicting the mass) to the one appearing in the top left panel of Figure 5. This is reasonable, since given that the IllustrisTNG model overpredicts the mass in SIMBA, the contrary should be expected when a SIMBA model is tested in IllustrisTNG. In the LH scenario, the predictions are slightly worse, decreasing the

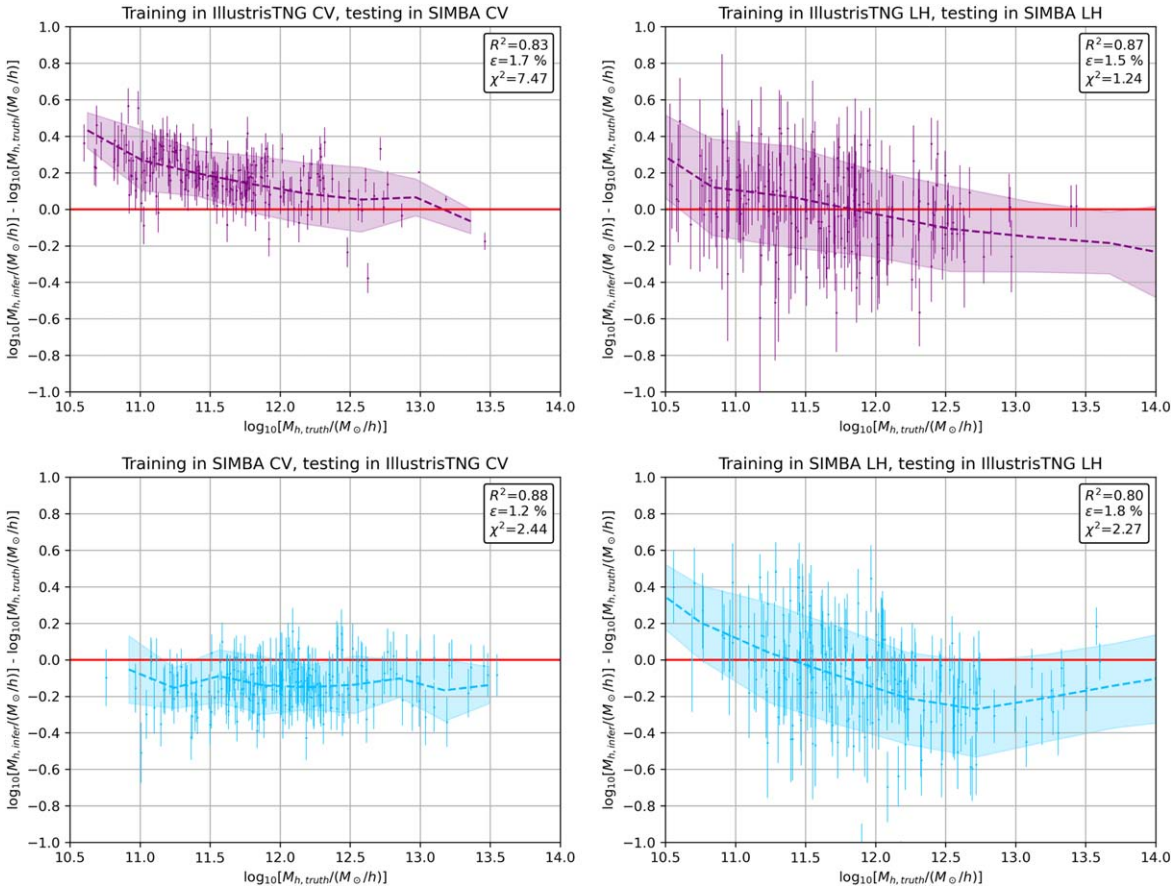

Figure 5. Same as Figure 4 but either using a model trained with the IllustrisTNG suite and tested with the SIMBA simulations (top) or trained in SIMBA and tested in the IllustrisTNG suite (bottom), for CV (left) and LH (right) sets. A model trained in a given suite worsens its behavior when tested in the other one, appearing biased in the CV case. However, in the LH set, it is possible to find a mapping between the parameter space of both subgrid physics models, alleviating such biases.

truth-prediction correlation down to $R^2 = 0.8$, and also poorer than in the CV case. Both sets present larger χ^2 values and a deviation from the Gaussian counterpart, given that only a fraction of $\sim$50% and $\sim$80% points lie within one and two times the posterior standard deviation, respectively.

It is noteworthy that, for the LH set, a model trained in IllustrisTNG generalizes better in SIMBA than the opposite case. An interpretation of this outcome is problematic. SIMBA usually covers more extreme astrophysical scenarios than IllustrisTNG, presenting also more galaxies per halo (see Figure 2), facts that could have an impact in this different cross testing. SIMBA simulations usually show a more stochastic behavior, presenting more scatter in some properties such as galaxy scaling relations. This can be due to the small box size compared to the large scale effect of the SIMBA AGN jets (Davé et al. 2019; Villaescusa-Navarro et al. 2021). Nevertheless, although we observe a higher variance and some outliers, most of the predictions only deviate up to 0.4 dex from the ground truth. In all cases, the mean relative error in the logarithm of the mass lies below 2%. The GNNs are able to recover the true mass in the majority of the cases within the standard deviation uncertainty. This shows that the models are relatively accurate even when they are applied to the simulations with a different subgrid physics modeling, manifesting their robustness.

5. Interpretability of the GNN

Neural networks represent superb tools to deal with multiple problems, but the factors that determine their behavior and performance are difficult to understand. It is always desirable to gain some interpretability of our ML models, in order to determine which features and inputs are the most relevant for predicting the output. To do so, we make use of the Python library Captum[33] (Kokhlikyan et al. 2020), a package designed for model interpretability and attribute selection. Specifically, we employ the saliency method for computing the gradients of the outputs with respect to the inputs and features employed (Simonyan et al. 2013). In that way, the larger gradients, and hence larger saliency values, imply that the prediction is more sensitive to variations of that variable.

We have computed the gradients with respect to the features employed in the training of the GNN, averaged over all galaxies and taken the absolute value. This approach gives us a saliency value for each property, which is shown in Figure 6 for the IllustrisTNG suite in the CV (blue) and LH (red) sets. For both sets, we can notice that, as could be expected, each coordinate of the position presents a very similar importance, due to implicit rotational invariance present in the problem. Regarding the CV case, the most relevant features to determine the output appear to be the stellar mass, M_* (as expected from the tight correlation shown in left panel of Figure 3) and,

[33] https://captum.ai/

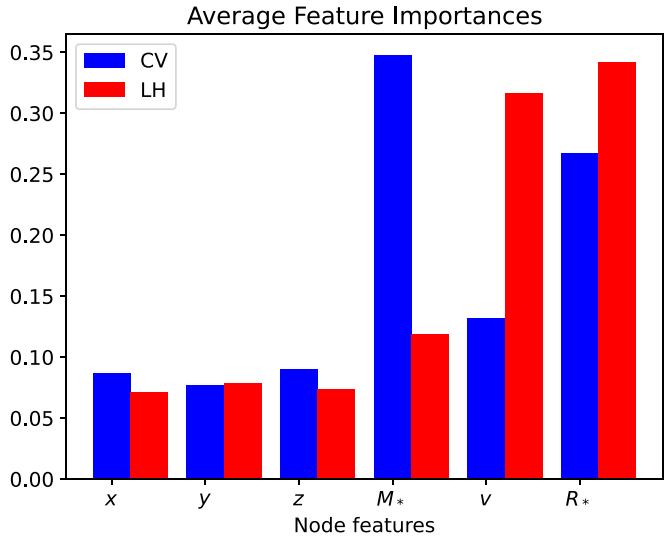

Figure 6. Saliency values of the galaxy features employed to train the GNN. Larger values imply that the network prediction is more influenced by variations of that feature, indicating that such properties are more relevant for the result.

subsequently, the stellar half-mass–radius, R_*. However, in the LH case, the network focuses more on R_* and v rather than the stellar mass. One can interpret this change of learning behavior from the larger scatter in stellar mass shown in Figure 3 in LH with respect to the CV set. In other words, the large variety of astrophysical and cosmological scenarios in the LH set may require the network to base its predictions on features that do not exhibit such large scatter. While the predictor mostly relies on the M_*-M_h correlation when the scatter is small, the velocities and galaxy size become better tracers otherwise. Nevertheless, one has to be cautious with these interpretations, since the saliency values may not give a complete picture. For instance, the spatial position importance may be underestimated since its weight could be split into the three coordinates.

It is common in computer vision tasks to calculate the saliency map, which indicates those pixels in an image that are most relevant for the final output (see, e.g., Villanueva-Domingo & Villaescusa-Navarro 2021 for an application in cosmology). In our case, we are dealing with graphs rather than images. Therefore, for a given halo, it is possible to compute the *saliency graph*, which shows the nodes whose features are more relevant for predicting the halo mass. Saliency values can be computed using the same procedure as above, but taking the absolute value and averaging over all features at each node. Examples of such saliency graphs for different halos are depicted in Figure 7, where the color indicates the saliency value, and the size is proportional to the stellar mass. Neighbors are connected by lines. Chosen samples present relative errors lower than 1.5% to ensure that their saliency graphs are meaningful. The top row stands for the CV set and bottom for LH, both in IllustrisTNG. In the CV set, as one would naively expect, the central galaxies provide the most relevant nodes for the output. These galaxies are also those with larger stellar masses. However, given that the stellar mass is an informative property, as seen in Figure 6, halos with relatively massive satellites can also show other relevant nodes besides the central one. On the other hand, in the LH set, the stellar mass is less informative, as seen in Figure 6; the network may focus more on some low-mass satellites rather than in central galaxies, which become less important. This illustrates the importance of the satellite population in our method. Hence, one has to be cautious when applying these models to galactic systems, since excluding some relevant satellites may have an impact in the predictions and induce a bias in the results.

It is thus pertinent to ask ourselves which satellites leave a greater impact on the halo mass. Figure 8 depicts the saliency value of each satellite galaxy as a function of distance to the center, excluding central galaxies. Point size is proportional to the stellar mass of the node. One can notice a trend where the closer galaxies become more relevant than those farther away. Moreover, this tendency seems to be relatively independent on their stellar mass. Furthermore, in real applications, the membership of galaxies lying at the boundaries of the system may be less secure, being not clear whether a far galaxy is a satellite or not. Figure 8 shows that removing or including such candidates would not have a significant impact in the results of the network, given the decreasing saliency value with the distance. This plot employs the IllustrisTNG CV data set, although similar qualitative conclusions can be extracted from the other cases.

These tests provide us with enlightening information about how GNNs learn and predict their outputs, as well as which are the most relevant components to understand the halo–galaxy connection. It has to be noted that the previous interpretations have to be taken with caution, because the saliency maps can be sometimes dominated by noise in the gradients and lose meaningfulness. Further development is required in order to obtain a better understanding of the training procedure and the emergent properties of halos from galaxies.

6. Discussion

6.1. Main Conclusions

Constraining the total mass of dark matter halos from galaxies still presents a challenge from both theoretical and observational perspectives, given the large contribution of the dark matter component. In this work, we have presented a new method based on artificial intelligence designed to infer the total mass of a halo from the properties of the galaxies it hosts. The point cloud arrangement of halo–galaxy catalogs has been exploited in order to structure halos as mathematical graphs, where galaxies constitute the nodes, connected by proximity. This organization of data makes it possible to employ GNNs, naturally suited to operate with graphs, to extract global permutation invariant quantities, as is the case of the halo mass. The models are fed with different observable galactic features, such as the position, velocity, stellar mass, and stellar half-mass–radius.

We have made use of the large collection of state-of-the-art hydrodynamic simulations from the CAMELS project to train our networks, which include thousands of simulations covering different astrophysical scenarios. Training GNNs over this data set allows us to achieve precise models capable of predicting the mass of a halo with remarkable accuracy. Here we outline some of the key conclusions of our method:

1. Our models are able to accurately infer the mass of a halo when trained and evaluated in simulations with fixed astrophysical and cosmological parameters, with a precision better than $\sim$1% (in terms of the logarithm of the halo mass).

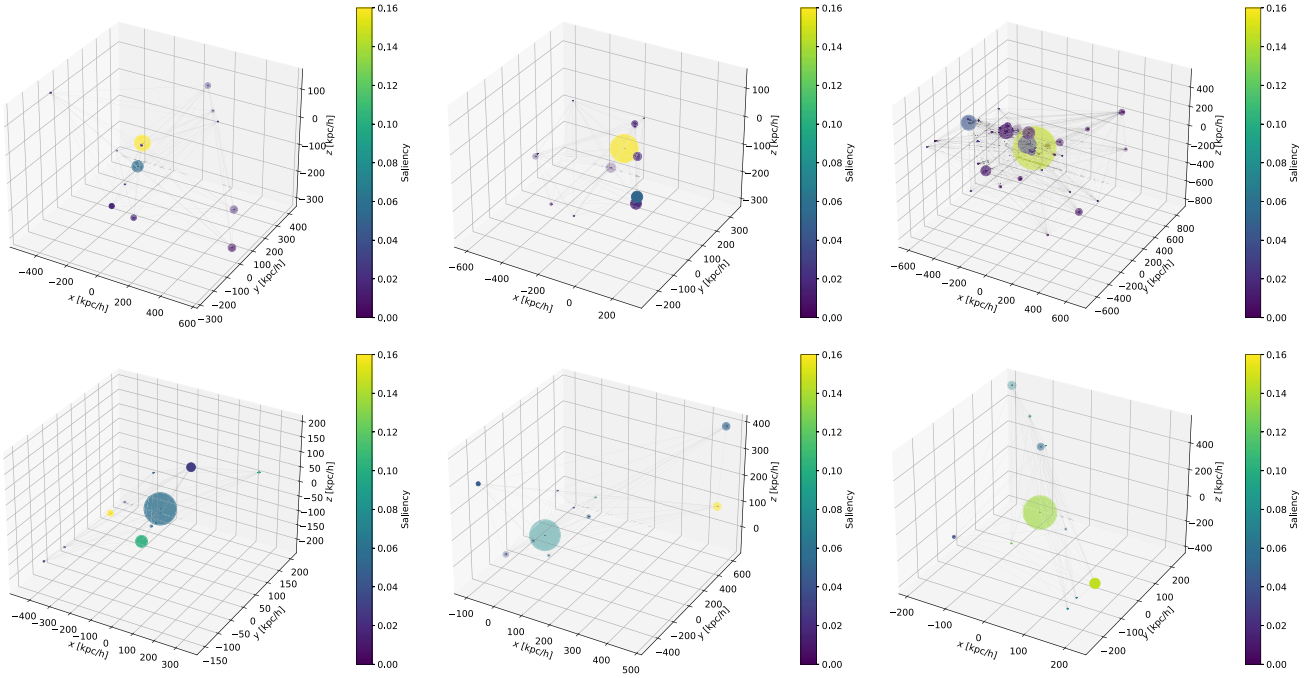

Figure 7. Saliency graphs for six different halos from the IllustrisTNG CV (top) and LH (bottom) data sets. Colors denote the saliency attributes, meaning that galaxies with larger values are more relevant for the prediction. Sizes of the nodes are proportional to the stellar mass of the galaxies. In the CV set, central galaxies usually have the greatest impact on the prediction; although if there are massive satellites, they can also contribute significantly (e.g., right panel). In the LH case, however, more attention is often focused on smaller satellites rather than on central galaxies.

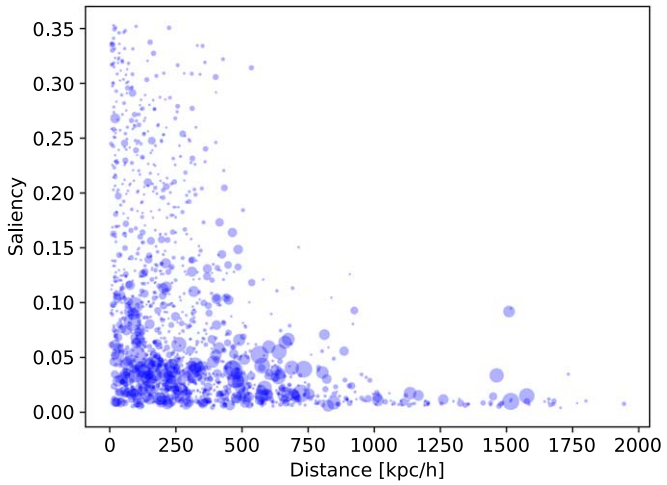

Figure 8. Saliency for each satellite galaxy depending on its distance to the halo center, excluding the central ones. Point diameter indicates the stellar mass of the nodes. There is a general tendency where closer galaxies to the center present larger saliency values, and therefore, are more relevant for predicting the output.

2. The networks have also been trained in simulations with different astrophysical and cosmological scenarios, successfully marginalizing over a broad astrophysical parameter space, and learning the connection between the halo mass and the properties of the galactic components. This provides accurate predictions of the total halo mass, with relative errors (in the logarithm of the halo mass) around ∼1%.
3. We have proven that the trained networks in a simulation suite still provide relatively precise predictions when tested in simulations with a different subgrid physics model, only increasing the mean relative error up to ∼2%. This illustrates the robustness of this method with respect to the astrophysical modeling.
4. Our results strongly rely on the velocity and size of the satellite galaxies, demonstrating the importance of other galactic features beyond the stellar mass.
5. We have performed likelihood-free bayesian inference, providing additionally an estimate for the standard deviation without knowing the actual likelihood.
6. Hyperparameter optimization has been carried out to maximize the performance of the networks.

6.2. Comparison to Previous Works

It is not straightforward to compare our results to previous works, given that the data sets employed, problem setups and features considered can be notably different. In the following, we compare our approach to other ML methods estimating the halo mass in previous literature, while emphasizing that a proper comparison between two techniques would require the use of common data and a problem statement.

1. Calderon & Berlind (2019) apply several ML techniques, such as standard MLPs, to predict the halo mass from galaxy groups data. They train their models in semi-analytical galaxy catalogs calibrated to SDSS data, outperforming traditional methods to infer the dynamical mass. Their 1σ scatter region spans ∼0.4–0.6 dex (a fractional difference of ∼3%–5%), while for the same masses, our GNN reduces down to ∼0.14 dex for the CV set and ∼0.2 dex for the LH one. In terms of the relative error in the mass (rather than in the logarithm), their scatter gets up to ∼250%–400%, while our models reduce it down to 40% and 60% for CV and LH respectively. Furthermore, our models make use of fewer

galaxy features (6 rather than their 9), and the LH simulations cover a large volume of the astrophysical and cosmological parameter space.

2. Man et al. (2019) face a similar problem of predicting halo masses from group galaxy properties, splitting their data sets in red and blue groups, according to the color of the central galaxies. They train a random forest estimator, obtaining scatter comparable to ours (note the $\sqrt{2}$ factor of difference in their definitions), but employing 9 features with a semianalytical galaxy formation model with fixed parameters.
3. von Marttens et al. (2021) predict galactic properties, such as the total mass, training different ML models in the TNG100 simulation. As in our case, the galaxies considered present stellar masses above $\gtrsim 10^8 M_\odot/h$, although they predict the subhalo mass rather than the halo one. They employ up to 15 features, including photometric, kinematic, and structural properties. When all features are included, they are able to get accurate results, with correlation of $R^2 = 0.92$. This case can be compared to our IllustrisTNG CV set, since it presents the fixed astrophysical and cosmological parameters,[34] where we obtain a better correlation of $R^2 = 0.97$ even using only 6 features. Moreover, we are also capable of obtaining a similar accuracy in the LH set compared to their models in TNG100.
4. Lucie-Smith et al. (2020) train CNNs on the density field of dark matter halos to infer their masses, with an accuracy comparable to the one of our network train on the CV set. However, note that this is a very different task, since that method relies on the 3D dark matter density field rather than on observable features, as is the case of our GNNs. Moreover, the authors employ N-body simulations with fixed cosmology rather than full hydrodynamic simulations. In any case, one has to be cautious at comparing these different approaches, due to the distinct data sets, assumptions, and features considered, being appropriate only for illustrating the potential of GNN models.

6.3. Caveats and Future Prospects

While the ML method presented here shows reasonable accuracy, it however has some caveats that have to be emphasized. For instance, it may not be obvious whether one galaxy belongs to a halo or not, in cases where two galactic groups are close together. This can be exacerbated in real observations, which take place in redshift space. This may cause the appearance of interlopers in a halo that could be counted as its own satellites, distorting the results. The effect of the presence of other halos in the environment when building the graph is also disregarded, which could have an impact for some close halos gravitationally bound. A way to deal with the influence of surrounding groups may be to include some global feature regarding the amount of galaxies or halos within a certain distance from the halo that is evaluated.

Note that, since only the halos with more than one galaxy have been considered, this method should only be applied to the halos that contain multiple satellites (above the assumed mass threshold). In those cases where the existence of satellites is not clear, the method may not be reliable. ML approaches could also be employed to identify the satellites around a given central galaxy, i.e., training a network to predict whether a subhalo is part of a given halo or not.

It is important to highlight that, given the small size of the simulation boxes, the amount of very massive galaxies may be undersampled. The fraction of galaxies with stellar mass above $10^{11} M_\odot/h$ is $\lesssim 1\%$ for both suites, implying a relatively low amount of those large objects. Training in larger simulation boxes with a higher amount of large halos would be advisable to ensure the reliability of our method when applied to very massive galaxies.

Moreover, we have only trained with halos at $z = 0$, but it would be also convenient to derive models capable of inferring halo masses at earlier times. New difficulties may arise in that case, such as redshift-space distortions, or the fact that we could expect fainter galaxies at higher redshift (because for a given luminosity threshold, the number density of galaxies decreases with redshift), reducing thus the size of the training data set.

The cross-tests discussed in Section 4.3 are aimed to illustrate the robustness of the network under different conditions regarding the baryonic feedback modeling. While our results are still relatively accurate even when they are tested in a different subgrid physics model than the one used for training, it would be desirable to further maximize their robustness. While training on IllustrisTNG and SIMBA simultaneously could enhance the performance, there is no guarantee that the network would be more robust under other conditions. It is possible that the model learns some kind of bimodal distribution of simulations, identifying first whether the halo corresponds to an IllustrisTNG or a SIMBA simulation, and then predicting the halo mass. A further improvement on this would be to train on IllustrisTNG plus SIMBA and cross-test in a third different suite. Such an additional suite with equivalent features to those of IllustrisTNG and SIMBA is still missing, although the CAMELS collaboration is already working to include other suites using different hydrodynamic codes and baryonic feedback models.

The results presented here are obtained by restricting ourselves to a reduced set of several observable quantities. However, it is possible in principle that using further variables, the results could improve. Additional correlations with the halo mass could arise if supplementary features are considered to train the GNNs. For instance, previous works like von Marttens et al. (2021) have shown the importance of photometric variables, such as luminosities at different wavelengths, for inferring the total halo mass. Additionally, considering, e.g., radii enclosing 20% and 80% of the mass (or light) may provide further information about the concentration and morphology of the halo. Among the other appealing galactic properties that could be contemplated are the gas mass, the neutral hydrogen (HI) mass, metallicities, velocity dispersion, etc. Furthermore, it would be desirable to explore more complex GNN architectures to find whether more accurate results can be achieved, reducing the scatter and the predicted uncertainties. The training of GNNs on galactic properties combined with other observables such as lensing could also enhance its predictive power.

The framework developed in this article has been proposed to infer the total mass of a given halo. However, since it is

[34] Although TNG100 has a box size of $\sim$100 Mpc$/h$, larger than the 25 Mpc$/h$ CAMELS box size, the CV set includes 27 simulations varying the random seed.

based on extracting global quantities from galaxy features, it could be generally applied to predict any other global quantity of the halo, such as its concentration, spin, characteristic age, etc. Depending on the specific global quantity considered, different galaxy or subhalo features may be required in order to maximize accuracy, which in principle could differ from those employed here. Moreover, one could apply GNN architectures to other problems, such as edge prediction. For instance, given a set of galaxies, a GNN could be trained to identify those that conform separate halos, as a ML alternative to friend-of-friends algorithms. Furthermore, note that our GNNs are trained to marginalize over the baryonic feedback and cosmological parameters, but they cannot identify the specific parameters underlying the simulation of a given halo. However, a GNN can be explicitly trained to infer the values of the cosmological parameters from a simulation employing its galactic population, a task that has been tackled with CAMELS catalogs in Villanueva-Domingo & Villaescusa-Navarro (2022).

It would be desirable to synthetize the predictive power of the GNNs into an analytical formula, via symbolic regression, as has been done already in other cosmological contexts (Cranmer et al. 2019, 2020; Wadekar et al. 2020; Shao et al. 2022). The analytical formulae derived in this way would depend upon the messages exchanged between the nodes. Nonetheless, obtaining a model susceptible of being replicated by a concise formula presents further difficulties. Performing symbolic regression requires interpretable models. One way to achieve this is to enforce sparse weights, obtained, e.g., by applying L1 regularization, which can affect the accuracy of the network. Sometimes, increasing the accuracy of a network reduces its extrapolation properties, and the other way around (Shao et al. 2022). Hence, deriving simple analytical formulae via symbolic regression from these GNNs remains as a desirable but challenging step for future work.

Our mass inference method based on GNNs is developed for galactic systems composed by central and satellite galaxies, but it could also be applied to galaxy clusters, given that the data description as graphs would be equivalent. While we believe our method will be perfect to perform mass estimates of galaxy clusters, the CAMELS simulations may not be the best training data set given the small volume that they cover, which translates into a lack of galaxy clusters, as already pointed out by previous work Wadekar et al. (2022).

Besides galaxy clusters, the description as graph-structured data makes this kind of network suitable for other types of astrophysical systems, which are characterized by point clouds. Examples of these may include globular clusters, stellar populations within a galaxy, or even planetary systems. Any point distribution could take advantage of the graph structure presented here for halos, in order to derive global quantities of such systems. An unexplored window remains open to apply all the power of GNNs to astrophysics.

Given that halo masses can be accurately predicted from numerical simulations by using the method shown here, the natural step forward would be applying this kind of model to real data. Once a GNN model is trained in simulations, one can predict the mass of a real halo using the observed kinematic and internal features of some real galactic systems as input data. Note that this procedure presents further challenges. Spectroscopic measurements would be required to infer the velocity, redshift information, and 3D positions of every satellite and central galaxy. The satellite population of distant halos could not be completely observable given their faintness, which could induce some biases in the results. Moreover, additional problems may arise when considering several redshifts at once, such as dealing with redshift-space distortions. However, our local galactic neighborhood provides us with a simple scenario to apply our method, where the previous difficulties can be handled. This task is carried out in a companion paper (Villanueva-Domingo et al. 2021), where GNNs trained with CAMELS simulations are employed to predict the masses of the halos containing the Milky Way and Andromeda. In that article, independent predictions for both halos are presented, which are consistent with other standard methods for estimating the dynamical masses of our galaxy and its companion. This represents a success of applying ML models trained with numerical simulations on real data, illustrating the power of artificial intelligence to enhance our knowledge of the universe.

We thank Miles Cranmer for enlightening discussions at early stages of this work. The work of P.V.D. is supported by CIDEGENT/2018/019, CPI-21-108. D.A.A. was supported in part by NSF grants AST-2009687 and AST-2108944, and by the Flatiron Institute, which is supported by the Simons Foundation. The training of the GNNs has been carried out using GPUs from the Tiger cluster at the Princeton University.

Data Availability

The models and implementation of GNNs in PyTorch Geometric, `HaloGraphNet`, are available on GitHub[35] (Villanueva-Domingo 2020). Details on the CAMELS simulations can be found in https://www.camel-simulations.org.

ORCID iDs

Pablo Villanueva-Domingo https://orcid.org/0000-0002-0936-4279
Francisco Villaescusa-Navarro https://orcid.org/0000-0002-4816-0455
Daniel Anglés-Alcázar https://orcid.org/0000-0001-5769-4945
Shy Genel https://orcid.org/0000-0002-3185-1540
Federico Marinacci https://orcid.org/0000-0003-3816-7028
David N. Spergel https://orcid.org/0000-0002-5151-0006
Lars Hernquist https://orcid.org/0000-0001-6950-1629
Mark Vogelsberger https://orcid.org/0000-0001-8593-7692
Romeel Dave https://orcid.org/0000-0003-2842-9434
Desika Narayanan https://orcid.org/0000-0002-7064-4309

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. 2019, arXiv:1907.10902
Anglés-Alcázar, D., Davé, R., Faucher-Giguère, C.-A., Özel, F., & Hopkins, P. F. 2017, MNRAS, 464, 2840
Armitage, T. J., Kay, S. T., & Barnes, D. J. 2019, MNRAS, 484, 1526
Battaglia, P. W., Hamrick, J. B., Bapst, V., et al. 2018, arXiv:1806.01261
Beck, R., & Sadowski, P. 2019, in Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019) (Red Hook, NY: Curran Associates, Inc.)
Behroozi, P. S., Conroy, C., & Wechsler, R. H. 2010, ApJ, 717, 379
Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. 2011, in Advances in Neural Information Processing Systems, ed. J. Shawe-Taylor et al., Vol. 24 (Red Hook, NY: Curran Associates, Inc.)

[35] https://github.com/PabloVD/HaloGraphNet

Bosma, A. 1978, PhD thesis, Groningen University
Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. 2021, arXiv:2104.13478
Calderon, V. F., & Berlind, A. A. 2019, MNRAS, 490, 2367
Cranmer, M., Melchior, P., & Nord, B. 2021, arXiv:2106.09761
Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., 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
Davé, R., Thompson, R., & Hopkins, P. F. 2016, MNRAS, 462, 3265
de Andres, D., Cui, W., Ruppin, F., et al. 2021, EPJWC, 257, 00013
de los Rios, M. E., Petač, M., Zaldivar, B., et al. 2021, arXiv:2111.08725
Fey, M., & Lenssen, J. E. 2019, Fast Graph Representation Learning with PyTorch Geometric, v2.0.2. https://github.com/pyg-team/pytorch_geometric
Green, S. B., Ntampaka, M., Nagai, D., et al. 2019, ApJ, 884, 33
Grego, L., Carlstrom, J. E., Reese, E. D., et al. 2001, ApJ, 552, 2
Gupta, N., & Reichardt, C. L. 2020, ApJ, 900, 110
Haider Abbas, M. 2019, arXiv:1912.05316
Hamilton, W. L. 2020, Synthesis Lectures on Artificial Intelligence and Machine Learning, 14 (San Rafael, CA: Morgan & Claypool), 1
Ho, M., Rau, M. M., Ntampaka, M., et al. 2019, ApJ, 887, 25
Hopkins, P. F. 2015, MNRAS, 450, 53
Huang, S., Leauthaud, A., Bradshaw, C., et al. 2022, MNRAS, in press
Huang, S., Leauthaud, A., Hearin, A., et al. 2020, MNRAS, 492, 3685
Jeffrey, N., Alsing, J., & Lanusse, F. 2021, MNRAS, 501, 954
Jeffrey, N., & Wandelt, B. D. 2020, arXiv:2011.05991
Kingma, D. P., & Ba, J. 2014, arXiv:1502.03167
Kipf, T. N., & Welling, M. 2017, arXiv:1609.02907
Kodi Ramanah, D., Wojtak, R., Ansari, Z., Gall, C., & Hjorth, J. 2020, MNRAS, 499, 1985
Kodi Ramanah, D., Wojtak, R., & Arendse, N. 2021, MNRAS, 501, 4080
Kokhlikyan, N., Miglani, V., Martin, M., et al. 2020, arXiv:2009.07896
Landry, D., Bonamente, M., Giles, P., et al. 2013, MNRAS, 433, 2790
Lucie-Smith, L., Peiris, H. V., Pontzen, A., Nord, B., & Thiyagalingam, J. 2020, arXiv:2011.10577
Man, Z.-Y., Peng, Y.-J., Shi, J.-J., et al. 2019, ApJ, 881, 74
Mandelbaum, R. 2015, in Galaxy Masses as Constraints of Formation Models, ed. M. Cappellari & S. Courteau, Vol. 311 (Cambridge: Cambridge Univ. Press), 86
Marinacci, F., Vogelsberger, M., Pakmor, R., et al. 2018, MNRAS, 480, 5113
Naiman, J. P., Pillepich, A., Springel, V., et al. 2018, MNRAS, 477, 1206
Nelson, D., Springel, V., Pillepich, A., et al. 2019, ComAC, 6, 2
Ntampaka, M., Trac, H., Sutherland, D. J., et al. 2015, ApJ, 803, 50
Ntampaka, M., Trac, H., Sutherland, D. J., et al. 2016, ApJ, 831, 135
Ntampaka, M., ZuHone, J., Eisenstein, D., et al. 2019, ApJ, 876, 82
Old, L., Wojtak, R., Mamon, G. A., et al. 2015, MNRAS, 449, 1897
Pillepich, A., Springel, V., Nelson, D., et al. 2018, MNRAS, 473, 4077
Qi, C. R., Su, H., Mo, K., & Guibas, L. J. 2017a, arXiv:1612.00593
Qi, C. R., Yi, L., Su, H., & Guibas, L. J. 2017b, arXiv:1706.02413
Rubin, V. C., Ford, W. K. J., & Thonnard, N. 1978, ApJL, 225, L107
Saro, A., Mohr, J. J., Bazin, G., & Dolag, K. 2013, ApJ, 772, 47
Seo, G., Sohn, J., & Lee, M. G. 2020, ApJ, 903, 130
Shao, H., Villaescusa-Navarro, F., Genel, S., et al. 2022, ApJ, 927, 85
Shlomi, J., Battaglia, P., & Vlimant, J.-R. 2020, MLST, 2, 021001
Simonyan, K., Vedaldi, A., & Zisserman, A. 2013, arXiv:1312.6034
Sofue, Y. 2015, PASJ, 67, 75
Sofue, Y., & Rubin, V. 2001, ARA&A, 39, 137
Somerville, R. S., & Davé, R. 2015, ARA&A, 53, 51
Springel, V. 2005, MNRAS, 364, 1105
Springel, V., Pakmor, R., Pillepich, A., et al. 2018, MNRAS, 475, 676
Springel, V., White, S. D. M., Tormen, G., & Kauffmann, G. 2001, MNRAS, 328, 726
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, arXiv:1706.03762
Veličković, P., Cucurull, G., Casanova, A., et al. 2018, arXiv:1710.10903
Villaescusa-Navarro, F., Anglés-Alcázar, D., Genel, S., et al. 2021, ApJ, 915, 71
Villaescusa-Navarro, F., Genel, S., Angles-Alcazar, D., et al. 2022, ApJS, 259, 61
Villaescusa-Navarro, F., Genel, S., Anglés-Alcázar, D., et al. 2022, arXiv:2201.01300
Villanueva-Domingo, P. 2020, HaloGraphNet, v1.0, Zenodo, doi:10.5281/zenodo.5676528
Villanueva-Domingo, P., & Villaescusa-Navarro, F. 2021, ApJ, 907, 44
Villanueva-Domingo, P., & Villaescusa-Navarro, F. 2022, arXiv:2204.13713
Villanueva-Domingo, P., Villaescusa-Navarro, F., Genel, S., et al. 2021, ApJ, 907, 44
Vogelsberger, M., Genel, S., Sijacki, D., et al. 2013, MNRAS, 436, 3031
Vogelsberger, M., Genel, S., Springel, V., et al. 2014, Natur, 509, 177
Vogelsberger, M., Marinacci, F., Torrey, P., & Puchwein, E. 2020, NatRP, 2, 42
von Marttens, R., Casarini, L., Napolitano, N. R., et al. 2021, arXiv:2111.01185
Wadekar, D., Thiele, L., Villaescusa-Navarro, F., et al. 2022, arXiv:2201.01305
Wadekar, D., Villaescusa-Navarro, F., Ho, S., & Perreault-Levasseur, L. 2020, arXiv:2012.00111
Wang, Y., Sun, Y., Liu, Z., et al. 2019, arXiv:1801.07829
Wechsler, R. H., & Tinker, J. L. 2018, ARA&A, 56, 435
Weinberger, R., Springel, V., Hernquist, L., et al. 2017, MNRAS, 465, 3291
Weinberger, R., Springel, V., & Pakmor, R. 2020, ApJS, 248, 32
Wojtak, R., & Mamon, G. A. 2013, MNRAS, 428, 2407
Wu, Z., Pan, S., Chen, F., et al. 2019, arXiv:1901.00596
Yan, Z., Mead, A. J., Van Waerbeke, L., Hinshaw, G., & McCarthy, I. G. 2020, MNRAS, 499, 3445
Young, B.-L. 2017, FrPhy, 12, 121201
Zaheer, M., Kottur, S., Ravanbakhsh, S., et al. 2017, arXiv:1703.06114
Zhou, J., Cui, G., Hu, S., et al. 2018, arXiv:1812.08434
Zwicky, F. 1933, AcHPh, 6, 110