

Exploring the halo-galaxy connection with probabilistic approaches

Natália V. N. Rodrigues^{1,2,3,*}, Natalí S. M. de Santi^{3,4,*}, Raul Abramo³, and Antonio D. Montero-Dorta⁵

¹ Instituto de Astrofísica de Canarias, Calle Via Láctea s/n, E-38205 La Laguna, Tenerife, Spain

² Departamento de Astrofísica, Universidad de La Laguna, E-38206 La Laguna, Tenerife, Spain

³ Departamento de Física Matemática, Instituto de Física, Universidade de São Paulo, Rua do Matão 1371, CEP 05508-090 São Paulo, Brazil

⁴ Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA

⁵ Departamento de Física, Universidad Técnica Federico Santa María, Avenida Vicuña Mackenna 3939, San Joaquín, Santiago, Chile

Received 4 December 2024 / Accepted 17 March 2025

ABSTRACT

Context. The connection between galaxies and their host dark matter halos encompasses a range of intricate and interrelated processes, playing a pivotal role in our understanding of galaxy formation and evolution. Traditionally, this link has been established through physical or empirical models. On the other hand, machine learning techniques are adaptable tools capable of handling high-dimensional data and grasping associations between numerous attributes. In particular, probabilistic models in machine learning capture the stochasticity inherent to these highly complex processes and relations.

Aims. We compare different probabilistic machine learning methods to model the uncertainty in the halo-galaxy connection and efficiently generate galaxy catalogs that faithfully resemble the reference sample by predicting joint distributions of central galaxy properties, namely stellar mass, color, specific star formation rate, and radius, conditioned to their host halo features.

Methods. The analysis is based on the IllustrisTNG300 magnetohydrodynamical simulation. The machine learning methods model the distributions in different ways. We compare a multilayer perceptron that predicts the parameters of a multivariate Gaussian distribution, a multilayer perceptron classifier, and the method of normalizing flows. The classifier predicts the parameters of a categorical distribution, which are defined in a high-dimensional parameter space through a Voronoi cell-based hierarchical scheme. The results are validated with metrics designed to test probability density distributions and the predictive power of the methods.

Results. We evaluate the model's performances under various sample selections based on halo properties. The three methods exhibit comparable results, with normalizing flows showing the best performance in most scenarios. The models not only reproduce the main features of galaxy properties distributions with high-fidelity, but can also be used to reproduce the results obtained with traditional, deterministic, estimators. Our results also indicate that different halos and galaxy populations are subject to varying degrees of stochasticity, which has relevant implications for studies of large-scale structure.

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

1. Introduction

We describe structure formation in terms of a layered approach. At the most basic level we have the density field, in dark matter and in baryons, which along with the gravitational potentials carry all the information about the initial conditions and the physical mechanisms that play a part in the growth of structures. At an intermediate level we have the halos and sub-halos, a summary statistic of the underlying density field which express and reflect in a vastly simpler fashion the fundamental cosmological parameters behind structure formation. And finally, at an observational level we have galaxies, quasars and other tracers of large-scale structure, which are the objects we actually employ when testing the physical models. It is critical, therefore, to understand how galaxies are related to their host halos and to the larger environment where they formed.

Machine learning has become a part of the standard toolbox used to explore that connection, not only because of its power to replicate the intricate relations between galaxies and

dark matter halos, but also as a means of investigating the physical mechanisms that shape this fundamental link. However, the number of factors that determine the properties that will emerge for each galaxy is enormous, leading to a degree of stochasticity that has been so far difficult to capture with existing approaches (Wechsler & Tinker 2018; Jo & Kim 2019; Stiskalek et al. 2022; de Santi et al. 2022; Jespersen et al. 2022; Rodrigues et al. 2023; Chuang et al. 2024).

This paper brings important improvements to predictions of galaxy properties based on their dark matter halos. It is built upon two previous studies in which we examined this relationship through the lens of machine learning, utilizing data from the IllustrisTNG300 simulation (de Santi et al. 2022; Rodrigues et al. 2023).

In the first work of this series (de Santi et al. 2022), we compare several methods, namely neural networks (NNs), extremely randomized trees (ERT), light gradient boosting machines (LGBM) and k-nearest neighbors (kNN), which input a set of halo properties and output the central galaxy properties. We show that these methods are able to reproduce key aspects of the relation between halo and galaxy properties. However, these are all deterministic estimators, which means that for a given

* Corresponding authors: nvilla-ext@iac.es,
natalidesanti@gmail.com

halo they are designed to return a single (expected) value for the central galaxy property. From a probabilistic standpoint, this is equivalent to focusing the predictions on the peak of the complete distribution, with the consequence of overpredicting the mode of the distribution, and underpredicting the tails. We initially tried to mitigate this problem by treating the problem as being due to an imbalanced dataset, which led us to employ the synthetic minority over-sampling technique for regression with Gaussian noise (SMOGN) (Kunz 2019; Branco et al. 2017). Although this technique allowed us to better reproduce the halo-galaxy and galaxy-galaxy properties, the improvement was not remarkable, and within that framework we are still constrained to predicting each galaxy property separately.

In the second work (Rodrigues et al. 2023), we concentrate on the same problem, but employing a non-deterministic approach in order to predict probability distributions associated with the galaxy properties, instead of single values. The method consists of binning the continuous galaxy properties and applying a NN classifier, dubbed NNclass, to predict the scores (weights) associated with each of the bins. Importantly, this scores add up to one, providing a proxy for a probability distribution. NNclass works as a proof of concept based on the notion of predicting distributions, instead of single values, in order to properly reproduce the scatter in the halo-galaxy connection. With this approach, one can generate many catalogs of galaxies by sampling from the predicted distribution. We managed to faithfully reproduce the shape of univariate and bivariate distributions. However, it is a simple method with key limitations.

First, it is not efficient in terms of predicting multivariate distributions in high dimensional spaces. In this work, we overcome this problem by introducing the Hierarchical Allocating Voronoi method (HiVAL), which allows to apply NNclass to predict the distributions of a larger number of galaxy properties jointly. There is a variety of ML methods in the literature designed to output distributions instead of single values, which could be used to capture the uncertainty in the relation between halo and galaxy properties. One example is to assume a (multivariate) Gaussian distribution. However, the distributions of galaxy properties can be very complex and even present multimodality. Although NNclass allows for some flexibility in the shape of the output distribution with the binning scheme, the method is still constrained to the choice of the binning and the bins can not be arbitrarily narrow, imposing a limitation on the resolution of the distribution.

As opposed to methods where we assume a parametric distribution to describe the target, generative models are designed to learn the underlying distribution of the data itself, in a non-parametric way. In particular, normalizing flows (NF) have proven to be excellent methods to model conditional distributions with complex shapes (Papamakarios et al. 2021), and even in the context of halo-galaxy connection (Lovell et al. 2023).

In this work, we apply machine learning methods to predict joint probability distributions of galaxy properties conditioned to the halo properties. Each method models the target (galaxy) distribution in a different way. We compare a multivariate Gaussian distribution (dubbed NNgauss), a categorical distribution (NNclass + HiVAL) and the distribution learned with NF. The paper is organized as follows. In Section 2 we describe the dataset, the simulation, the halo and galaxy properties considered in this work. In Sections 3 and 4, we introduce the ML methods and explain the training procedure. In Section 5 we present the metrics used to quantify the performance and com-

pare the methods. In Section 6 we present the results and summarize our main conclusions in Section 7.

2. Data

This work is based on data from the IllustrisTNG magnetohydrodynamical cosmological simulation (Pillepich et al. 2018a,b; Nelson et al. 2018, 2019; Marinacci et al. 2018; Naiman et al. 2018; Springel et al. 2018), which was generated using the AREPO moving-mesh code (Springel 2010). IllustrisTNG (or TNG for simplicity) is an improved version of the previous Illustris simulation (Vogelsberger et al. 2014a,b; Genel et al. 2014), featuring a variety of updated sub-grid models that were calibrated to reproduce observational constraints such as the $z = 0$ galaxy stellar mass function or the cosmic SFR density, to name but a few (see the aforementioned references for more information). The IllustrisTNG simulation adopts the standard Λ CDM cosmology (Planck Collaboration XIII 2016), with parameters $\Omega_m = 0.3089$, $\Omega_b = 0.0486$, $\Omega_\Lambda = 0.6911$, $H_0 = 100 h \text{ km s}^{-1} \text{ Mpc}^{-1}$ with $h = 0.6774$, $\sigma_8 = 0.8159$, and $n_s = 0.9667$.

Our analysis builds on the works of de Santi et al. (2022) and Rodrigues et al. (2023), where we employed a variety of ML techniques to reproduce the halo–galaxy connection. In those analyses, we used the largest box available in the TNG suite, TNG300, spanning a side length of $205 h^{-1} \text{ Mpc}$ with periodic boundary conditions. The reason for this choice was to minimize cosmic variance for our clustering measurements. For consistency with those works, and in order to allow comparison, we stick to the same dataset in the present work (even though clustering measurements are not presented this time). TNG300 contains 2500^3 DM particles of mass $4.0 \times 10^7 h^{-1} \text{ M}_\odot$ and 2500^3 gas cells of mass $7.6 \times 10^6 h^{-1} \text{ M}_\odot$. The adequacy of TNG300 in the context of clustering science has been extensively proven in a variety of analyses (see, e.g., Contreras et al. 2021; Gu et al. 2020; Hadzhiyska et al. 2020, 2021; Montero-Dorta et al. 2020, 2021a,b; Shi et al. 2020; Favole et al. 2021; Montero-Dorta & Rodriguez 2024).

Also for consistency with de Santi et al. (2022) and Rodrigues et al. (2023), the same halo and galaxy properties analyzed in those works are employed here. In terms of input halo properties we use:

- *Virial mass* ($M_{\text{vir}} [h^{-1} \text{ M}_\odot]$), which is computed by adding up the mass of all gas cells and DM particles contained within the virial radius R_{vir} (based on a standard collapse density threshold of $\Delta_c = 200$). In order to ensure that haloes are well resolved, we impose a mass cut $\log_{10}(M_{\text{vir}} [h^{-1} \text{ M}_\odot]) \geq 10.5$, corresponding to at least 500 dark matter particles.
- *Virial concentration* (c_{vir}), defined in the standard way as the ratio between the virial radius and the scale radius: $c_{\text{vir}} = R_{\text{vir}}/R_s$. Here, R_s is obtained by fitting the DM density profiles of individual haloes with a NFW profile (Navarro et al. 1997).
- *Halo spin* (λ_{halo}), for which we adopt the Bullock et al. (2001) definition: $\lambda_{\text{halo}} = |J|/\sqrt{2}M_{\text{vir}}V_{\text{vir}}R_{\text{vir}}$. Here, J and V_{vir} correspond to the angular momentum of the halo and its circular velocity at R_{vir} , respectively.
- *Halo age*, parametrized as the half-mass formation redshift $z_{1/2}$. This age parameter corresponds to the redshift at which half of the present-day halo mass has been accreted into a single subhalo for the first time. The formation redshift is measured following the progenitors of the main branch of the subhalo merger tree computed with SUBLINK, which is initialized at $z = 6$.

- The *overdensity* around haloes on a scale of $3 h^{-1} \text{ Mpc}$ (δ_3). The overdensity is defined as the number density of subhalos within a sphere of radius $R = 3 h^{-1} \text{ Mpc}$, normalized by the total number density of subhalos in the TNG300 box (e.g., Artale et al. 2018; Bose et al. 2019).

As for the galaxy properties that we want to reproduce, we again choose:

- The *stellar mass* ($M_* [h^{-1} \text{ M}_\odot]$), including all stellar cells within the subhalo. In order to ensure that galaxies are well resolved, we impose a mass cut $\log_{10}(M_* [h^{-1} \text{ M}_\odot]) \geq 8.75$, corresponding to at least 50 gas cells.
- The *colour* $g - i$, computed from the rest-frame magnitudes, which are obtained in IllustrisTNG by adding up the luminosities of all stellar particles in the subhalo (Buser 1978). This specific choice of colour is rather arbitrary. We have checked that using other combinations (i.e., $g - r$) provides similar results.
- The *specific star formation rate* (sSFR [$\text{yr}^{-1} h$]), which is the star formation rate (SFR) normalised by stellar mass. The SFR is computed by adding up the star formation rates of all gas cells in the subhalo. Note that around 14% of the galaxies at redshift $z = 0$ in TNG300 have SFR = 0. In order to avoid numerical issues, we have adopted the same approach as in de Santi et al. (2022), assigning to these objects artificial values of SFR, such that they end up distributed around $\log_{10}(\text{sSFR} [\text{yr}^{-1} h]) = -13.5 / \log_{10}(\text{SFR} [\text{M}_\odot \text{ yr}^{-1}]) = -3.2$, with a dispersion of $\sigma = 0.5$.
- The *galaxy size*, represented by the stellar (3D) half-mass radius ($R_{1/2}^{(*)} [h^{-1} \text{ kpc}]$) – i.e., the comoving radius containing half of the stellar mass in the subhalo.

We refer the reader to the TNG webpage¹ and main papers for more information on the determination of these quantities. Some more details can also be found in de Santi et al. (2022) and Rodrigues et al. (2023).

3. Methods

In this section we describe the models to predict galaxy properties based on the halo properties. All methods follow the general idea of predicting for each halo a conditioned probability distribution for the properties of its central galaxy. The model parameters are optimized such that the likelihood of observing the reference catalog (TNG300 training set) is maximized. Then, the loss function is the negative log-likelihood of the distribution. They differ from each other mainly in the way that the galaxy properties distribution (the likelihood) is modeled.

3.1. Neural network with Gaussian likelihood

In this method we assume a multivariate Gaussian to model the target distribution. We use TensorFlow² and TensorFlow Probability³ (Abadi et al. 2015) to train a multilayer perceptron (MLP) neural network to learn the parameters of the distribution, which are the mean values of each galaxy property and the covariance matrix. The Gaussian distribution of an instance characterized by a data vector \mathbf{x} is given by

$$p(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \quad (1)$$

where $\boldsymbol{\mu}$ is the vector of mean values whose components are the galaxy properties, $\boldsymbol{\Sigma}$ is the covariance matrix, and D is the number of dimensions (e.g. $D = 4$ if we predict the four galaxy properties jointly). The loss-function is then given by the negative log-likelihood of the Gaussian distribution. We refer to this method as NNgauss.

3.2. Neural network classifier

This method approximates the target probability distribution as a categorical distribution whose parameters are the probabilities of the classes⁴. We discretize the continuous values of galaxy properties into several bins that become classes (categories). The probabilities of the classes are computed by training, once again, a MLP using TensorFlow and TensorFlow Probability. The last layer of the MLP classifier has the softmax activation function, which normalizes the output values in such a way that the scores of all classes add up to one. In this sense, the output values are interpreted as probabilities associated with the classes (the parameters of the categorical distribution). The loss function is the categorical cross-entropy (CCE), which is the negative log-likelihood of the categorical distribution:

$$\text{CCE} = -\frac{1}{N} \sum_n^N \sum_k^K y_{nk} \log p_{nk}, \quad (2)$$

where N is the number of instances, K the number of classes, y_{nk} is the true class and p_{nk} is the assigned probability.

In principle, this strategy is flexible to model distributions with complex shapes, which could handle, for example, skewness and bimodality observed in the color or (s)SFR of galaxies. The narrower the bins, the higher the resolution, and the better we approach the continuous distribution. However the bins can not be arbitrarily small, otherwise there would not be a sufficiently large number of instances populating the bins to properly train the MLP. Therefore, the choice of the number of bins is a balance between the occupancy and the resolution or, in other words, the number of objects populating each bin and the size of the bin.

In our precursor work (Rodrigues et al. 2023), we have applied this technique and we managed to reproduce univariate and bivariate distributions of galaxy properties. However, the method was limited to low dimensions because of the way the bins (classes) were defined. In that paper, we simply grid the distributions in 50 equally spaced bins per dimension, which means that for a three dimensional problem, for example, one would have an order of 10^5 classes (bins). In this work, we introduce the Hierarchical Voronoi Allocation method (see Sect. 3.2) to discretize the distributions in such a way that more properties can be predicted jointly.

The strategy to sample from NNclass involves two steps. The MLP outputs the categorical distribution from which we sample the HiVAL domains (discrete values). Then we sample continuous values from the domains by defining a Normal distribution with mean value corresponding to a randomly chosen object belonging to the domain and variance corresponding to the variance of the domain. The list of objects belonging to the domains and their variances are both HiVAL outputs.

¹ <https://www.tng-project.org>

² <https://www.tensorflow.org>

³ <https://www.tensorflow.org/probability>

⁴ This is a generalization of the Bernoulli distribution.

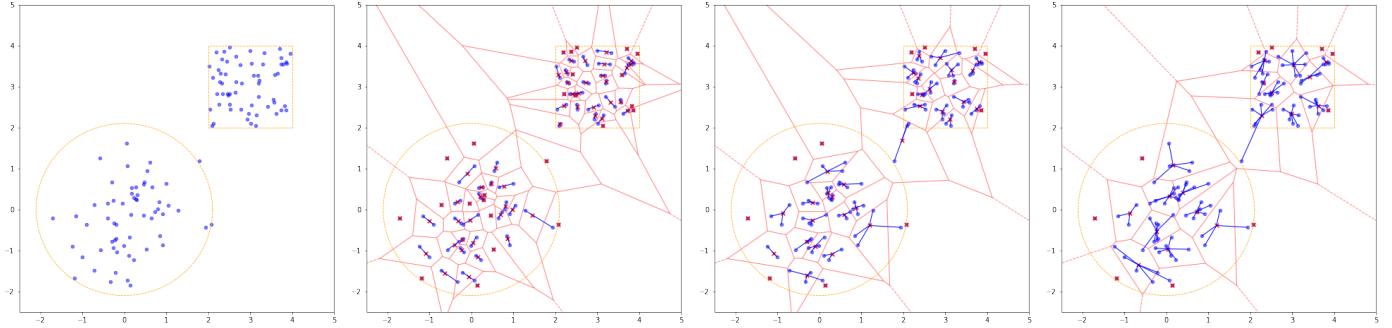


Fig. 1. Illustration of the HiVAI method. The first panel shows samples of two distributions as blue dots. The yellow lines are simply guides for the boundaries of the two disjoint distributions used in this example. The second panel (from left to right) shows the first iteration of HiVAI: the red crosses represent new particles (centers of cells) created by combining pairs of the original particles, and the red lines represent the boundaries of the cells. The blue lines indicate which of the original particles have been grouped together (the group being represented by the red cross). The third panel shows the second iteration of HiVAI: center of cells from the previous iteration, represented by red crosses, are once again combined in pairs to create new cells. The combined particles are connected by blue lines, therefore, in this iteration there are more particles connected as compared to previous one. The fourth panel shows the third iteration of HiVAI in this example. It follows the same reasoning as the previous panels.

Hierarchical Voronoi Allocation

In this section we describe a new tiling and allocation method for collecting discrete objects into cells (or classes), which we call the Hierarchical Voronoi Allocation method (HiVAI). Given a set of N objects in D dimensions, HiVAI works by (i) determining the Voronoi cells of the N objects, and then (ii) pairing neighbouring cells. In order to determine which neighbours are paired, we employ as criteria the D -dimensional volumes of the Voronoi cells, the $(D-1)$ -dimensional areas of the faces of the cells adjacent to their neighbors, as well as the distances between the neighbouring cells. Methods based on the Voronoi tessellation have been used before in the context of large-scale structure, such as the void-finding algorithm ZOBOV (Neyrinck 2008).

HiVAI is a self-organizing, deterministic method to group instances (the objects) into cells, which can be iterated as many times as needed. Each iteration of the method lowers the number of cells by a factor $\gamma \gtrsim 1/2$, and consequently increases the number of objects per cell by a factor $1/\gamma \lesssim 2$. HiVAI preserves the shape of the original (parent) distribution by progressively tiling the D -dimensional space while preserving the boundaries of the distribution. The edges of the distribution are frozen by means of identifying the objects that sit at the boundaries (whose Voronoi cells have infinite volumes) and blocking them from pairing up with neighbours.

The way HiVAI works is pictured in Fig. 1. For that example, we took two 2D disjoint distributions: (i) a 2D multivariate Gaussian random distribution with mean $(0, 0)$ and diagonal variance $(0.7, 0.7)$, and (ii) a 2D uniform random distribution in the interval $(2-4, 2-4)$. We then sampled 60 objects from each one of the two distributions, for a total of 120 objects – see the top panel of Fig. 1. The 3σ region corresponding to the 2D Gaussian distribution is shown in that figure as the yellow circle, and the boundary of the uniform distribution is shown as the yellow square.

In the first iteration of HiVAI (second panel in Fig. 1), for each cell (in this case, for each object), the algorithm determines the Voronoi domains of the objects and identifies the neighbouring cells. If the object sits at the boundary of the distribution, its Voronoi cell volume is infinite and it is therefore blocked from pairing up with a neighbour. But if the cell has a finite volume, and if it has available finite-volume neighbours to pair up with, then the partner is chosen as the neighbour with the largest area in common with the original object. This choice ensures that the

cells created by the resulting pairs have the smallest area/volume ratio. The red crosses indicate the “center of mass” (CM) of the resulting cells after this first iteration: some are boundary cells (which are the unpaired particles at the boundaries), some sit between pairs of the original objects, and some cells may be single objects simply because there are no available partners anymore. For this example we have used “masses” (weights) of the CM which are inversely proportional to the size of the Voronoi cells, $m_i = V_i^{-1/D}$, but these weights can be tuned to the needs of the particular problem at hand.

In the second iteration of HiVAI (third panel in Fig. 1), the algorithm again starts by determining the Voronoi cells and identifying all the neighbours. Cells with infinite volume are blocked from pairing up with a neighbour, which preserves the boundaries of the original distribution. The finite-volume cells with available partners are then paired with the neighbour that has the largest area in common with that cell. The CM (red crosses) now indicate cells that can be made up of 1–4 particles. The third iteration (see fourth panel of Fig. 1) continues this process: the cells are now domains with 1–8 objects. In this example, after the third iteration we end up with 9 boundary domains (made up of single particles) and 18 domains with 3 to 8 objects.

We fit HiVAI with the same training set used to train the machine learning models (see Sect. 4). HiVAI returns, for each iteration, the position of the center of the domains (in parameter space coordinates), the list of training instances with the index of the domain they belong to, i.e., their classes – which is used as target to train NNclass –, and, complementary, the list of training set instances belonging to each domain, as well as their standard deviation, which characterize the dispersion of the domains. Several complementary scores are provided for additional analysis.

HiVAI approximates the density of the distribution of the data and, therefore, its probability distribution. It can be used to analyze the statistical significance of a given region (domain) in the parameter space by evaluating how it is populated, for example, if it has low or high density. For the purpose of predicting galaxy properties, we use HiVAI to model their distribution and combine it with NNclass to assign the scores associated to each HiVAI cell. HiVAI is implemented with the SciPy⁵ library (Virtanen et al. 2020).

⁵ <https://scipy.org>

3.3. Normalizing flows

NF is a ML algorithm designed to be a flexible way to learn distributions with NNs. It is a powerful generative method because it can handle conditional probabilities (Bingham et al. 2019) in high dimensions (still taking into account the curse of dimensionality Coccato et al. 2024).

The basic idea of NF is to use the change of variables formula for PDFs to transform a simple distribution (e.g. a Gaussian distribution) into the desired distribution. Mathematically, suppose we have two variables $X, Z \in \mathbb{R}^D$, with PDFs $p_X(x), p_Z(z) : \mathbb{R}^D \rightarrow \mathbb{R}$, where D is the number of dimensions. We can define the flow as a bijective map $f(Z) = X, f^{-1}(X) = Z$, which must be invertible and differentiable. Then, the two densities are related by:

$$p_X(x) = p_Z(z) \left| \det \left(\frac{\partial f(z)}{\partial z} \right) \right|^{-1} = p_Z(f^{-1}(x)) \left| \det \left(\frac{\partial f^{-1}}{\partial x} \right) \right|, \quad (3)$$

where $\left| \det \left(\frac{\partial f^{-1}}{\partial x} \right) \right| = J_{f^{-1}}(x)$ is the Jacobian of $f^{-1}(x)$. We choose $p_Z(z)$ to be a standard Normal distribution. Once we have f , we can easily sample from the base distribution $p_Z(z)$ and apply f to obtain a sample from the target distribution $p_X(x)$.

One way to learn the bijector that models the distributions with NFs, is to use coupling layers. A distribution $p_X(x)$ of dimension D is split into two parts A and B , with dimensions d and $(D-d)$. Then, the parameters of the transformation of part B are modeled by a NN that uses part A as input. Starting with $x \in \mathbb{R}^D$, we can split the variable into $(x^A, x^B) \in \mathbb{R}^d \times \mathbb{R}^{(D-d)}$. Then, the variables transform as

$$x^A = g(z^A) \quad (4)$$

$$x^B = h[z^B; \Theta(z^A)], \quad (5)$$

where g and h are both bijections, such as affine transformations or splines. The function g may also be the identity. The parameters of the coupling function h are defined in \mathbb{R}^{D-d} by the generic function $\Theta(z^A)$, which is modeled by a NN. The inverse transformation is given by

$$z^A = g^{-1}(x^A) \quad (6)$$

$$z^B = h^{-1}[x^B; \Theta(z^A)]. \quad (7)$$

Examples of this model are NICE (Dinh et al. 2014) and Real NVP (Dinh et al. 2016). In particular, neural spline flows model g and h as elementwise monotonic rational splines (Durkan et al. 2019; Dolatabadi et al. 2020).

A generalization to learn distributions in NFs is done by autoregressive flows (Kingma et al. 2016). This means that we split the data variable into more dimensions and the transformation for each dimension i is modeled by an autoregressive NN, resulting in a conditional probability distribution given by $p(x_i|x_1, \dots, x_{i-1})$. More specifically, we can define the autoregressive flow such that

$$x_1 = g(z_1) \quad (8)$$

$$x_i = h[z_i; \Theta(z_1, \dots, z_{i-1})], i = 2, \dots, D. \quad (9)$$

Alternatively, the inverse is

$$z_1 = g^{-1}(x_1) \quad (10)$$

$$z_i = h^{-1}[x_i; \Theta(z_1, \dots, z_{i-1})], i = 2, \dots, D. \quad (11)$$

Table 1. NNgauss hyperparameters range.

Learning rate	l2	Hidden layers	Hidden units
$[10^{-6}, 10^{-1}]$	$[10^{-5}, 10^{-1}]$	[1, 5]	[4, 512]

Notes. The learning rate and l2 regularization parameter are sampled in log-spaced steps.

As opposed to NNgauss and NNclass, NF is a complete generative method in the sense that it can model the joint distribution of all the variables in the problem (in our case, halo and galaxy properties), as

$$p_{x,y}(x, y) = p_{y|x}(y|x)p_x(x). \quad (12)$$

The loss function is then the negative log-likelihood: $-\frac{1}{N} \sum_n^N (\log p_{y|x}(y|x) + \log p_x(x))$.

For our purposes, we only use the conditioned distribution $p(y = \text{gal.}|x = \text{halos})$ to get the probability distribution of the galaxies hosting some set of given halos, such as the halos from our test set. Nevertheless, we are also modeling the halo properties distribution itself, following Eq. (12), and we can use $p(x = \text{halos})$ to handle missing features (see the github repository for more details).

In the present work, we use the spline coupling to model the halo distribution $p(x = \text{halo})$, and the conditional spline autoregressive to model the conditioned galaxy distribution $p(y = \text{gal.}|x = \text{halo})$. We implement NFs using the Pyro⁶ library (Bingham et al. 2019).

4. Training procedure

The ML methods have been trained, validated and tested considering the complete dataset of 174 527 instances (pairs of halos and galaxies) – see Sect. 2. We have split the initial catalog into training, validation and test subsets of 48%, 12%, and 40%, respectively. We use the same split as de Santi et al. (2022) and Rodrigues et al. (2023), for consistency. We use the OPTUNA⁷ package (Akiba et al. 2019) to perform a Bayesian optimization with Tree Parzen Estimator (TPE) (Bergstra et al. 2011) to select the hyperparameters for each of the different models. We do at least 100 trials for each model. In the following subsections we describe the set of explored hyperparameters.

4.1. NNgauss model selection

The explored hyperparameters are the number of layers, the number of neurons in each layer, and the value of the L2 regularization parameter. We also try different values for the initial learning rate of the Adam optimizer (Kingma & Ba 2014) and implement a schedule to reduce it when the validation loss stagnates for 20 epochs. We implement an early stopping schedule with a patience of 40 epochs. The final set of hyperparameters is the one that yields the lowest loss function in the validation set – see Tables 1 and 2.

4.2. NNclass model selection

We explore the same hyperparameters as NNgauss. One important additional aspect of NNclass model selection is the choice of HiVAL iteration, which is the number of classes (bins). This is

⁶ <https://docs.pyro.ai/en/stable/index.html>

⁷ <https://optuna.org>

Table 2. NNgauss best trial hyperparameters.

Learning rate	I2	Hidden layers
5.1×10^{-5}	6.4×10^{-4}	[134, 494, 256, 456, 380]

Table 3. NNclass hyperparameters range.

Learning rate	I2	Hidden layers	Hidden units
$[10^{-6}, 10^{-1}]$	$[10^{-5}, 10^{-1}]$	[1, 5]	[4, 512]

Notes. The learning rate and I2 regularization parameter are sampled in log-spaced steps.

Table 4. NNclass best trial hyperparameters.

Learning rate	I2	Hidden layers	#classes
1.0×10^{-3}	1.9×10^{-5}	[225, 468, 125, 402, 33]	857

Table 5. NF hyperparameters range.

Learning rate	Bins	Bound	Layers & units	Batch size
$[10^{-4}, 10^{-2}]$	[8, 64]	$K = [1, 5]$	[1, 10] & [8, 256]	$[10, 10^4]$

Notes. The learning rate and batch size are sampled in log-spaced steps.

related to the balance between resolution and occupation. Narrow cells lead to better resolution, but the higher the number of cells, the lower their mean occupancy, and the harder it is for the MLP to properly classify the objects.

In principle, the number of classes can be set as a hyperparameter to be explored by OPTUNA. However, we could not rely on the value of the loss function to select the best model because we would be comparing different likelihoods. Moreover, choosing a different metric that grasps information from the D -dimensional space may not be trivial. Alternatively, we choose a few number of classes based on some prior knowledge of occupancy and resolution (i.e., not too large or too low number of classes), and run OPTUNA on them independently. Afterwards, we compare the generated samples of continuous values and choose the one with the best performance in terms of the metrics discussed in Sect. 5, giving preference to the model with best conditioning power. The explored range and the final set of hyperparameters are outlined in Tables 3 and 4.

4.3. NF model selection

For both transformations (i.e., spline coupling and conditional spline autoregressive), the explored hyperparameters are the following. The number of segments in the splines (bins), the interval within which the splines are defined (bound = $[-K, K]$), the number of hidden layers and units of the NNs that output the parameters of the splines, the learning rate of the Adam optimizer, and the batch size. We consider the final, best set, of hyperparameters the one that yields the lowest loss function on the validation set – see Tables 5 and 6. We also vary the split dimension of the spline coupling between [1, 2].

5. Metrics

In this work, we train models to predict probability distributions of galaxy properties based on halo properties to quantify the uncertainty in this relation. As a consequence, we expect to generate catalogs of galaxy properties that faithfully reproduce the

Table 6. NF best trial hyperparameters.

Learning rate	Bins	Bound	Hidden layers	Batch size	Split
7.7×10^{-3}	12	$K = 1$	[219, 219, 219]	7620	2

reference by sampling from these distributions. Therefore, we select metrics to validate and measure the quality of the predictions by quantifying the similarity between the distributions, the calibration of the predicted distributions, and the conditioning power of the models to ensure that the right galaxies are populating the right halos.

5.1. Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test (hereafter K-S test) quantifies the difference between two distributions $[f_1(x_1), f_2(x_2)]$ in a non-parametric way by measuring the maximum distance between the cumulative distribution functions (CDFs)

$$D = \max(|F_1(x_1) - F_2(x_2)|), \quad (13)$$

where $F_1(x_1) = \text{CDF } [f_1(x_1)]$, $F_2(x_2) = \text{CDF } [f_2(x_2)]$. Similarly, one can measure the 2D K-S test for bivariate distributions (Ivezić et al. 2014; Peacock 1983; Fasano & Franceschini 1987).

The K-S test is sensitive to location, scale, and the shape of the underlying distributions, but it is more sensitive to differences near the center than in the tails. It is often used for hypothesis testing to check whether two samples comes from the same distribution and focuses on the maximum difference between the CDFs, making it less sensitive to subtle details.

In this work, we have implemented our own 1D K-S test schedule and used the Taillon (2018) repository to compute the 2D K-S test.

5.2. Wasserstein distance

The Wasserstein distance, also known as the earth mover's distance, quantifies the cost of transforming one distribution into another. Mathematically, given two probability distributions u and v defined over \mathcal{R} , the 1-Wasserstein distance is given by

$$W(u, v) = \inf_{\pi \in \Gamma(u, v)} \int_{\mathcal{R} \times \mathcal{R}} d\pi(x, y) |x - y| \quad (14)$$

where $\Gamma(u, v)$ is the set of joint probability distributions π whose marginal distributions are $u(x)$ and $v(y)$ (Ramdas et al. 2015; Virtanen et al. 2020). This means that we are looking for the joint distribution $\pi(x, y)$ with marginals $u(x), v(y)$ that minimizes the expected distance between x and y when the values (x, y) are sampled from $\pi(x, y)$.

The Wasserstein distance provides a metric that takes into account the entire structure of the distribution, allowing for a meaningful and smooth representation of the similarities and differences between them. The Wasserstein distance is an important metric in the context of optimal transport and have been widely used to train generative models, like Wasserstein GANs (generative adversarial networks; Arjovsky et al. 2017).

We compute the 1D Wasserstein distance based on Virtanen et al. (2020) and the 2D version using Flamary et al. (2021).

5.3. Coverage test

The coverage test is a metric to verify the accuracy of generative posterior estimators $\hat{p}(\theta|x)$, for a high number of dimensions of θ , using coverage checks, and is often used in the context

of simulation-based inference (SBI). Here we use the Tests of Accuracy with Random Points (TARP)⁸ (Lemos et al. 2023).

At this point, it is useful to explicitly frame our problem using SBI language for clarity. A simulation is a pair $\{\theta_i, x_i\}$, which in our case are the pairs of central galaxies and host halos given by TNG300. The posterior is the probability of observing θ given x , which in our case is modeled by the methods presented in Sect. 3. For each simulation (halo) i , we can sample many values θ_i (galaxy properties) from this estimator and compare with the true value θ_i^* from TNG300. Depending on how the samples compare with the true value, according to the coverage test, we can claim that the estimator $\hat{p}(\theta|x)$ is accurate.

The basic idea of this score is to compute the expected coverage probability ECP ($\hat{p}, \alpha, \mathcal{G}$) averaged over the data distribution x , where \hat{p} is the posterior estimator, \mathcal{G} is a credible region generator, and α is a credibility level. ECP is computed using samples from the reference distribution $p(\theta|x)$ and samples from the estimated posterior distribution $\hat{p}(\theta|x)$.

More specifically, for each true value θ_i^* , it takes the posterior estimation $\hat{p}_i(\theta_i|x^*)$ and sample a reference point⁹ θ_i^r from a uniform distribution $\mathcal{U}(0, 1)$ – all the samples are transformed to have values within 0 and 1. Then, it measures the distance of each θ_i to θ_i^r as $d(\theta_i, \theta_i^r)$ and counts the number of samples that are inside a radius R_i defined by the distance between θ_i^* and θ_i^r . This means that for each simulation i , it computes the fraction f_i of the sampled points falling within a ball centered at θ_i^r . The ECP is then defined as the average of $f_i < 1 - \alpha$ over the simulations (data points), for credibility levels ranging from 0 to 1.

A good quality posterior distribution for the whole range of θ gives a one-to-one correspondence of coverage to credibility level. Deviations from the one-to-one correspondence indicates that the posterior estimator is underfitting, overfitting, or biased. The algorithm is fully described in Lemos et al. (2023).

5.4. Simulation based calibration

Similarly to the TARP coverage test described above, one can check whether the predicted distributions are well calibrated with the simulation-based calibration (SBC) rank statistic (Talts et al. 2020). Given a halo, we generate samples for the hosted galaxy properties and rank them with respect to the corresponding reference value (from TNG300). In other words, we count how many values fall below the reference. When analyzing many halos, we expect this statistic to follow a uniform distribution, meaning that the distribution is neither too narrow (overconfident), too broad (underconfident), or biased. The results are presented in Appendix A.

5.5. Pearson correlation coefficient

We use the Pearson correlation coefficient (PCC)

$$\text{PCC} = \frac{\text{cov}(y^{\text{pred.}}, y^{\text{true}})}{\sigma_{y^{\text{pred.}}} \sigma_{y^{\text{true}}}} \quad (15)$$

to quantify the correlation between the predictions and the reference sample and, therefore, evaluate how well the galaxy properties are conditioned to the halo properties (i.e., evaluate the predictive power of model). This metric is particularly informative for single-point estimation, as we can verify if the expected values are in agreement with the reference.

We evaluate the PCC score in two scenarios. First, we generate a catalog of predicted values $y^{\text{pred.}}$ by sampling a single time

from the predicted distribution and compare the values with the reference y^{true} . We repeat this process several times to have an estimate of the variance over multiple realizations – see Fig. 4. Second, we consider $y^{\text{pred.}}$ as the expected value, which means that we take the mean over several generated catalogs and compare it with the reference y^{true} . We refer to this as the single point estimate catalog – see Fig. 6.

6. Results

In this section, we compare the best OPTUNA trials of NNgauss, NNclass and NF. We start Sect. 6.1 by comparing the predictions in the complete test set, and then, in Sect. 6.2, we compare the model predictions for some halo populations.

Here we show the results computed based on the prediction of the four dimensional distribution of all galaxies properties predicted jointly. However, one could train the models to predict distributions with whatever desired combination of galaxy properties, such as single properties, pairs of properties, etc. The advantage of predicting the joint distribution is that it captures correlations between galaxy properties (Rodrigues et al. 2023).

Unless noted otherwise, all metrics are computed with the complete test set and 1000 catalogs randomly sampled from the predicted distributions for each instance (halo or galaxy) to calculate the standard deviation. The exceptions are the 2D K-S test and 2D Wasserstein distance, which are computed using a randomly drawn subset of 30×10^4 from the test catalog, and 5 samples per instance to calculate the standard deviation.

Additional results are presented in Appendix A, such as the simulation based calibration analysis introduced in Sect. 5.4 and the Tables with the explicit values for the metrics (PCC, K-S test and Wasserstein distance).

6.1. Complete test sample

This section presents the results computed with the complete test sample. We start with a visual inspection of contour plots of a single generated catalog in Fig. 2. It shows the diagrams that are typically analyzed in the context of astrophysics, but one could plot any of the possible combinations of the four galaxy properties by marginalizing over the remaining ones. We see that NNclass and NF faithfully reproduce the shape of the target distributions, both in one and two dimensions, proving their flexibility to model complex shapes. NNgauss shows good agreement for stellar mass and radius, but shows biased contours for the color and sSFR (especially for 3- σ), which have a strong bimodal feature.

Fig. 3 shows the coverage test of the joint distribution of all four properties on the left-hand side, and, once again, of univariate and bivariate marginalized distributions on the right-hand side. We show the same pairs of properties as Fig. 2. We compute the coverage test with 1000 samples for each instance in the test set¹⁰. Typically, all methods present well calibrated distributions. The main exceptions are NNclass stellar mass prediction, which indicates that it is underfitting, and NNgauss sSFR prediction. We verified that NNclass presents this underfitting feature even when the stellar mass is predicted alone.

The biased prediction of NNgauss is likely due to the bimodal shape in sSFR. Notice that for $g - i$ it also slightly deviates from the diagonal. This could perhaps be improved with a mixture of Gaussian distributions, such as Mixture Density Networks (Bishop 1994; see Lima et al. 2022 for an application in photo-z estimation). However NF is already a very general

⁸ See the code used at <https://github.com/Cielo-Institute/tarp>

⁹ Not to be confused with the true value θ_i^* from TNG300.

¹⁰ In the case of NNclass, we sample 40 classes and then 25 continuous values for each classes.

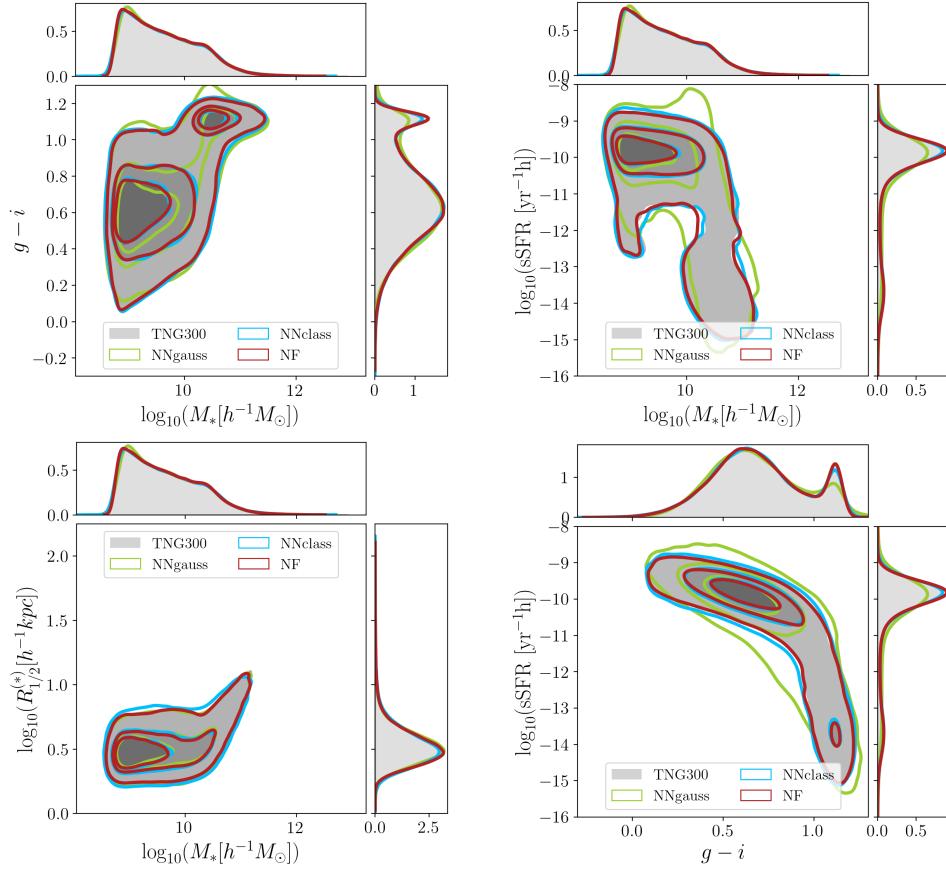


Fig. 2. Contour plots of pairs of galaxy properties predicted by each model (NNGauss, NNclass, and NF) compared with the truth distribution of the test set from TNG300. The samples are generated by marginalizing the complete four-dimensional distribution.

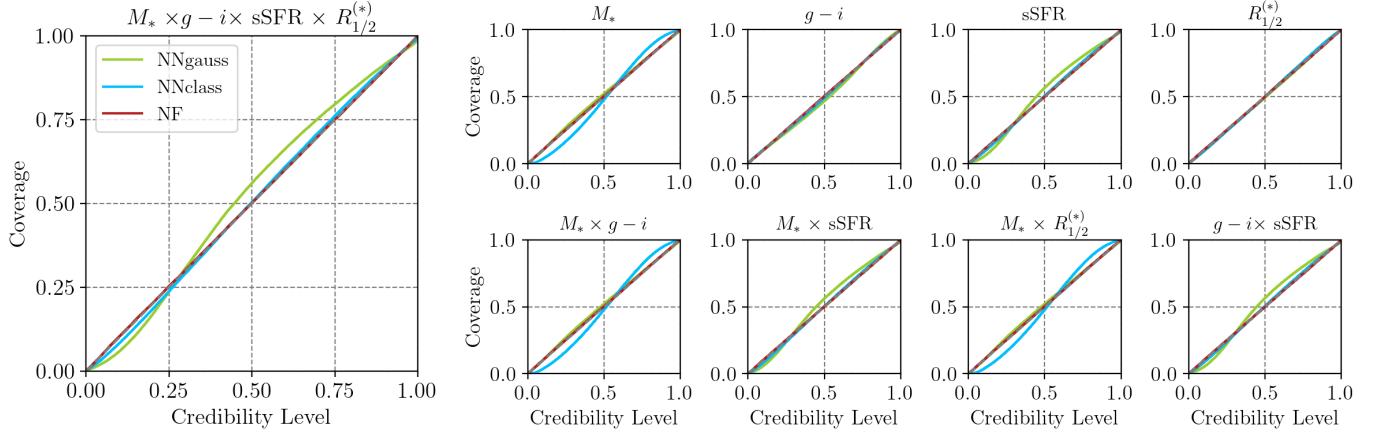


Fig. 3. Left: TARP coverage test of the joint distribution of all properties. Right: TARP coverage test over marginal distributions; top panels show individual properties and bottom panels shows pairs of properties.

method and overcomes the aforementioned limitations of both NNclass and NNGauss.

Fig. 4 shows the metrics introduced in Sect. 5 computed over the generated catalogs. We take the mean and standard deviation of the scores computed over multiple generated catalogs to obtain the error bars. For the 2D scores, we use a randomly selected subset of the test sample (due to computational time).

The PCC is an important metric to verify if the predicted distributions are well-conditioned. Suppose we apply a model that randomly assigns values and simply mimics the scatter and shape

of the distribution (i.e., the frequencies of the target values). Even such a model would provide good Wasserstein distance and K-S test on the complete test set. However, since the distributions are not well conditioned, there would be no correlation between true and predicted values, and the PCC score would be poor. On the other hand, if we aim to model the uncertainty in such a way that the sampled catalogs faithfully resemble the reference, it is important to evaluate if we are properly modeling the underlying statistical process that generated the reference catalog, which can be done through the K-S test and the Wasserstein distance.

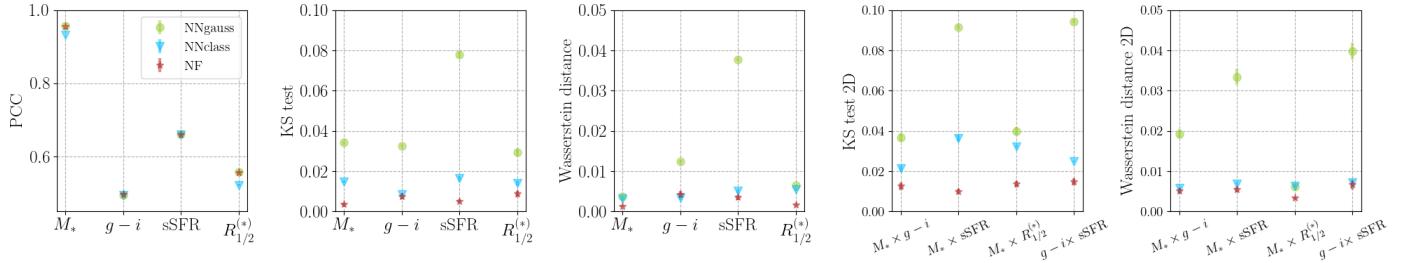


Fig. 4. Comparison between the machine learning methods NNgauss, NNclass and NF in terms of the metrics introduced in Sect. 5. The error bars correspond to the standard deviation over multiple catalogs sampled from the predicted distributions. The 2D scores are computed with a random sub-sample from the test set.

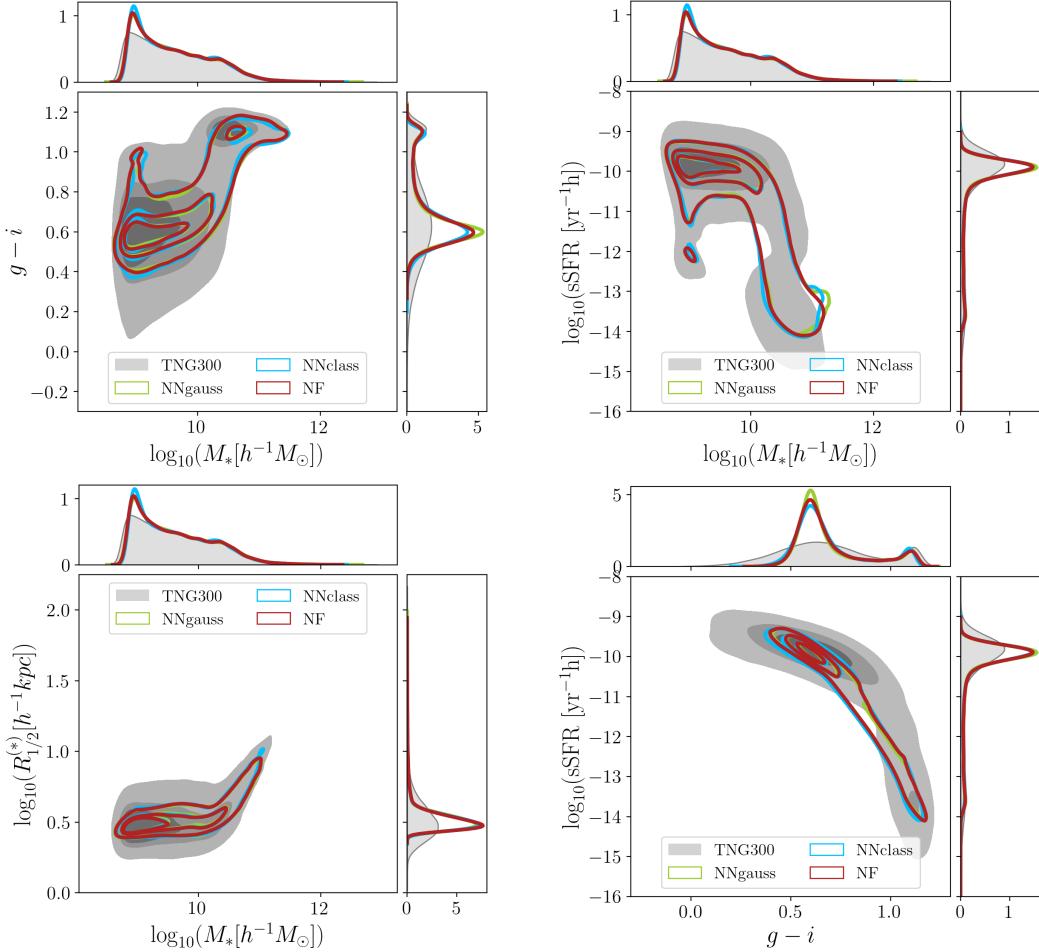


Fig. 5. Contour plots of the single-point estimate catalog of the models (NNgauss, NNclass and NF) compared with the truth distribution of the test set from TNG300. This catalog corresponds to the average over 1000 samples of the predicted distribution.

We see that, although NNclass shows good performance in terms of the K-S test and Wasserstein distance (comparable to NF results), it is slightly worse than the other methods in terms of PCC. This is perhaps related to the natural limitation of the discretization, which is susceptible to the occupancy and resolution trade-off, and affects the conditioning power of the method. NNgauss, on the other hand, shows great performance in terms of PCC, similar to NF, although it is not the best assumption to fairly reproduce the details of the shape of the distribution, specially for color and sSFR that have bimodality, as confirmed by the lower performance in terms of the K-S test and Wasserstein distance.

One can also build a single-point estimate catalog based on the expected values by taking the average over multiple samples of the distributions. This should be equivalent to a traditional, deterministic, machine learning estimator like those applied in de Santi et al. (2022). Fig. 5 shows the contour plots of the average over 1000 samples. As expected, all methods repeat the previous achievements (Raw and SMOGN) resulting in predictions along the peak of the distributions.

Fig. 6 shows the PCC of the single-point estimate catalogs computed with NNclass, NNgauss, NF, and, for comparison, the deterministic models from de Santi et al. (2022): Raw (corresponding to the predictions of the stacked mod-

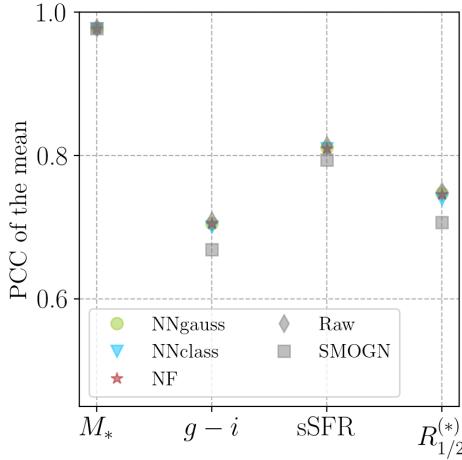


Fig. 6. PCC of the single-point estimate catalog for each galaxy property. The single-point estimate of NNgauss, NNclass and NF is computed by taking the average over 1000 samples of the predicted distributions.

els NN, kNN, LGBM, and ERT taken from the raw dataset) and SMOGN (related to the predictions of the stacked models NN, kNN, LGBM, and ERT computed with the SMOGN augmented dataset). A similar exercise is done in Appendix A from Rodrigues et al. (2023). Due to the intrinsic scatter, we do not expect a high PCC with a single realization of our predicted distributions, as shown in Fig. 4. However, if we take the mean over the 1000 realizations and thus estimate the expected value, the PCC score should be comparable with the deterministic methods.

We see that with all methods we can recover results as good as the Raw model, which demonstrates the flexibility of the probabilistic approach in recovering what is achieved with deterministic estimators. We omit the other metrics in this single-point estimation analysis since visual inspection of Fig. 5 already shows that the scatter is not well reproduced in this case.

6.2. Halo populations

In the previous section we discussed the performance of the models in the complete test set. In this section we evaluate the performance on sub-samples of the test set, corresponding to some chosen halo populations (i.e., some selection based on halo properties). This analysis is useful to investigate in more detail the conditioning power of the methods, if they provide the correct populations of galaxies given a population of halos, and to see how well the distributions of different tracers are recovered. They may be more or less degenerate, and therefore show different shapes and scatters, depending on how tightly they are constrained to the halo properties.

To select halo populations, we use HiVAL domains defined over halo parameter space. As an illustrative case, we do the selection based on the halo mass and age, simply because for 2D distributions we can visualize the HiVAL diagram, but a similar analysis can be done based on a larger number of halo properties¹¹.

In the top panel from Fig. 7, we have selected HiVAL domains containing low-mass, intermediate age, halos (total of 10265 instances). The top, left-hand panel shows the HiVAL diagram

build upon halo mass and age and highlights the domains containing the selected halo population. The top, right-hand panel shows how the color – stellar mass diagram of the galaxies population the selected halos. Once again, we emphasize that other combinations of properties could be computed, but for the purposes of illustration we keep only the color-mass diagram. Gray regions correspond to the reference TNG300 sample and colored contours correspond to the machine learning models predictions. These halos typically host low-massive, blue galaxies. We see that all models accurately reproduce the properties of the galaxies from these halos.

The bottom panels from Fig. 7 show a similar analysis, but with a different population (total of 1674 instances). As shown in the HiVAL diagram on the bottom, left-hand panel, these halos have intermediate mass and are typically older. This intermediate halo mass regime presents a lot of degeneracy in color, which is likely related to quenching. We see that both blue and red galaxies can occupy those halos. Due to their flexibility, NNclass and NF can accurately predict the range of galaxy colors, while NNgauss does not capture this bimodal feature. Stellar mass, on the other hand, is tightly constrained in this selection and presents a very narrow distribution. Both NF and NNgauss are in good agreement with TNG300 for stellar mass, but NNclass shows strong underfitting.

7. Conclusions

As a follow-up of previous works (de Santi et al. 2022; Rodrigues et al. 2023), we explore machine learning methods to model the relation between halo and central galaxy properties based on the Illustris TNG300 simulation.

In this work, we investigate methods that model the uncertainty due to the intrinsic stochastic nature of the halo-galaxy connection, and from which one can generate galaxy catalogs that faithfully resemble the reference (TNG300) based on some set of halo properties. This idea has been developed as a proof of concept in Rodrigues et al. (2023) and in the present paper we apply more robust methods that overcome the main limitations of that previous work.

We compare different approaches to model the joint distribution of galaxy properties: a neural network combined with a multivariate Gaussian (NNgauss), a neural network classifier combined with a categorical distribution (NNclass), and normalizing flows (NF). In particular, we introduce the HiVAL method to define statistically significant regions in a given parameter space based on the density of instances via Voronoi tesselation, which is used to model the target distribution of NNclass.

Each method presents weaknesses and strengths manifested in terms of the different evaluated metrics. We investigate their quality in three main aspects. First, the calibration of the predicted distribution, quantified by the TARP coverage test and by the simulation based calibration rank statistics (see Appendix A); second, the predictive power of the models, quantified in terms of PCC and through an analysis of single-point estimation; and third, the level of fidelity of the predicted distribution with respect to the reference, quantified in terms of the K-S test and the Wasserstein distance. A summary of the values of the scores is presented in the Appendix A.

NNgauss shows good predictive power, with results well correlated with the reference – as compared to previous works – but it does not reproduce all the features of the distributions, in particular because they present multimodality. NNclass is our first step towards a method able to model a distribution with flexible shape that deviates from Gaussianity. However, it is not optimal

¹¹ We reinforce that to train NNclass we have only used HiVAL in the galaxy parameter space. Not to be confused with the application of HiVAL on the distribution of halo properties.

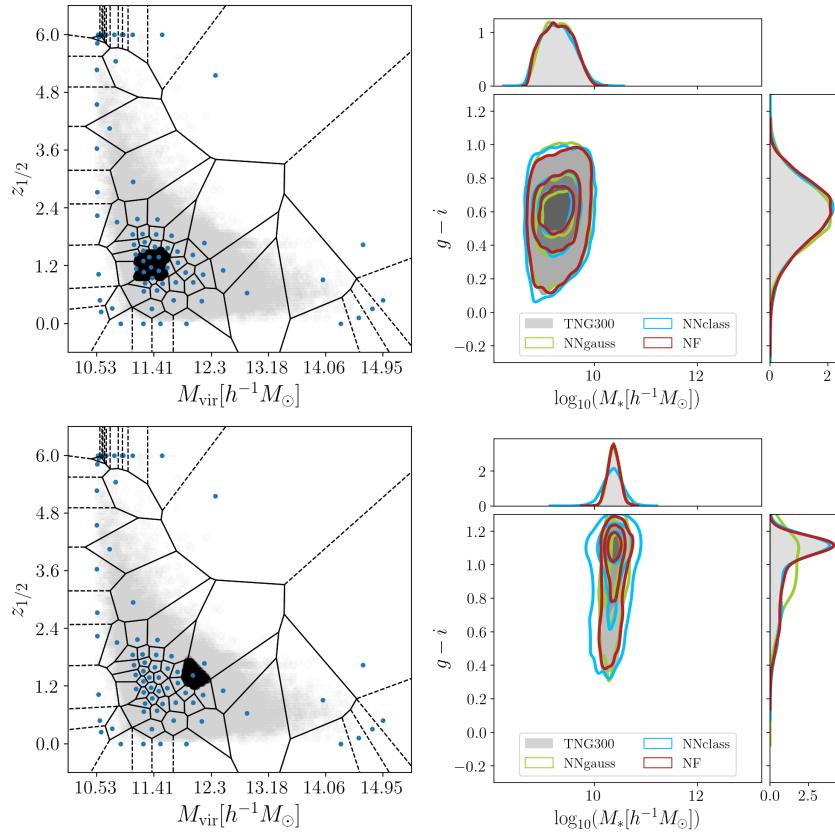


Fig. 7. Left: two population of halos selected based on HiVAL using halo mass and age. Right: color-mass diagram of the galaxies hosted by the corresponding halo populations obtained with TNG300 (reference, gray), NNgauss (green), NNclass (blue) and NF (red).

due to the discretization of continuous variables, which compromises the predictive power of the method and leads to an underfitting distribution for stellar mass – which should be the best constrained property in halo-galaxy connection, due to its well known high correlation with halo mass. Nevertheless, the satisfactory results in terms of K-S test and the Wasserstein distance suggests that flexible, non-parametric distributions are a promising approach, and motivate the use of NF. NF achieves satisfactory results on all the metrics here evaluated, proving its quality to model the intrinsic statistical process that generated the reference TNG300, and in terms of predictive power. Moreover, the single-point estimation analysis shows that the models are consistent with the deterministic estimators. Therefore, this methodology is a promising auxiliary tool to drive studies and analysis in the context of the halo-galaxy connection, and can serve as a complementary approach to existing methods, such as sub-halo abundance matching (SHAM; see e.g. Favole et al. 2021; Ortega-Martinez et al. 2024) and semi-analytical models (SAM; see e.g. White & Frenk 1991; Guo et al. 2013).

NF perform better at capturing the shape of the distribution but are inherently more complex models that generally require greater computational resources compared to a MLP that predicts the parameters of a given distribution. In this study, we did not observe significant differences in computational cost between the models. However, it is worth noting that, as the number of features increases and the complexity of the problem grows, the computational cost may become relevant. In this context, it is interesting to see that NNgauss performs well across various metrics.

In order to improve the predictions, one could consider exploring different sets of halo properties. This will be par-

ticularly relevant for satellite galaxies, as their formation and evolution involve many complex processes. Future work will be devoted to exploring informative features to predict satellite galaxy properties, combined with the probabilistic approach here proposed. For example, recent works have applied deep learning to optimize feature extraction, and thus uncover deterministic correlations hidden in the scatter (see, e.g. Chuang et al. 2024; Wu et al. 2024).

One possible application of this machinery is to populate dark matter only simulations with TNG300-like galaxies, which reduces the computational cost of running full hydrodynamical simulations. However, baryonic interactions from the hydrodynamical simulations may impact on the dark matter and halo properties (Gebhardt et al. 2024). One could use the dark-matter only counterpart from TNG300 (with the same initial conditions) to retrain the models and to investigate the effects of baryons in more detail (see, e.g. Lovell et al. 2021). We leave this application for future work.

In Rodrigues et al. (2023) we have demonstrated that the probabilistic model better populates the correct halos with the correct galaxies through an analysis of clustering statistics (specifically with the power spectrum). In particular, we have quantified how the uncertainty in our halo-galaxy connection model impacts the power spectrum estimation. We expect that with the methods presented here, we will better quantify this error and investigate possible implications for large-scale structure studies.

Another interesting and related application is regarding the inverse problem of inferring halo properties from galaxy properties, which gives a more straightforward application to real observations. The exact same machinery could be re-trained to

input galaxy properties and output halo properties and thus have a direct modeling of $p(\text{halo}|\text{gal})$.

Our training and validation sets are all based on TNG300. In order to apply this to real observations, it is important to stress that we are naturally constrained to whatever limitations the simulation may have, and some of the stochasticity that we are modeling comes from numerical artifacts of the simulations (Genel et al. 2019). Comparisons between simulations with different galaxy formation models should be helpful and informative to evaluate (and possibly extend) the range of applicability of our models (e.g. different sub-grid physics as explored in CAMELS; Villaescusa-Navarro et al. 2021).

Finally, we conclude by emphasizing that we have selected a few examples to showcase some of the possible analysis that can be done with this machinery and to drive some discussions in the context of halo-galaxy connection. The pipeline developed in this work is flexible to predict properties individually and jointly, and can be applied in a variety of contexts.

Data availability

The material presented in this paper is available in the following GITHUB repository: [nvillanova/halo_galaxy_connection_probML](https://github.com/nvillanova/halo_galaxy_connection_probML)¹².

Acknowledgements. NVNR acknowledges financial support from CAPES and the IAC for hospitality during the period in which part of this work has been done, financed by a PrInt-PDSE grant. NSMS acknowledges financial support from FAPESP, grants 2019/13108-0 and 2022/03589-4. RA is supported by FAPESP and CNPq. ADMD thanks Fondecyt for financial support through the Fondecyt Regular 2021 grant 1210612. The training, validation and test of the NFs have been carried out using graphic processing units from Simons Foundation, Flatiron Institute, Center of Computational Astrophysics.

References

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, <https://www.tensorflow.org>
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. 2019, ArXiv e-prints [arXiv:1907.10902]
- Arjovsky, M., Chintala, S., & Bottou, L. 2017, ArXiv e-prints [arXiv:1701.07875]
- Artale, M. C., Zehavi, I., Contreras, S., & Norberg, P. 2018, *MNRAS*, **480**, 3978
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. 2011, in *Advances in Neural Information Processing Systems*, eds. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Curran Associates, Inc.), 24
- Bingham, E., Chen, J. P., Jankowiak, M., et al. 2019, *J. Mach. Learn. Res.*, **20**, 28:1
- Bishop, C. 1994, *Mixture Density Networks*, Workingpaper, Aston University, USA
- Bose, S., Eisenstein, D. J., Hernquist, L., et al. 2019, *MNRAS*, **490**, 5693
- Branco, P., Torgo, L., & Ribeiro, R. P. 2017, *Proc. Mach. Learn. Res.*, **74**, 36
- Bullock, J. S., Dekel, A., Kolatt, T. S., et al. 2001, *ApJ*, **555**, 240
- Buser, R. 1978, *A&A*, **62**, 411
- Chuang, C.-Y., Jespersen, C. K., Lin, Y.-T., Ho, S., & Genel, S. 2024, *ApJ*, **965**, 101
- Coccaro, A., Letizia, M., Reyes-González, H., & Torre, R. 2024, *Symmetry*, **16**, 942
- Contreras, S., Angulo, R. E., & Zennaro, M. 2021, *MNRAS*, **504**, 5205
- de Santi, N. S. M., Rodrigues, N. V. N., Montero-Dorta, A. D., et al. 2022, *MNRAS*, **514**, 2463
- Dinh, L., Krueger, D., & Bengio, Y. 2014, ArXiv e-prints [arXiv:1410.8516]
- Dinh, L., Sohl-Dickstein, J., & Bengio, S. 2016, ArXiv e-prints [arXiv:1605.08803]
- Dolatabadi, H. M., Erfani, S., & Leckie, C. 2020, ArXiv e-prints [arXiv:2001.05168]
- Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. 2019, ArXiv e-prints [arXiv:1906.04032]
- Fasano, G., & Franceschini, A. 1987, *MNRAS*, **225**, 155
- Favole, G., Montero-Dorta, A. D., Artale, M. C., et al. 2021, *MNRAS*, **509**, 1614
- Flamary, R., Courty, N., Gramfort, A., et al. 2021, *J. Mach. Learn. Res.*, **22**, 1
- Gebhardt, M., Anglés-Alcázar, D., Borrow, J., et al. 2024, *MNRAS*, **529**, 4896
- Genel, S., Vogelsberger, M., Springel, V., et al. 2014, *MNRAS*, **445**, 175
- Genel, S., Bryan, G. L., Springel, V., et al. 2019, *ApJ*, **871**, 21
- Gu, M., Conroy, C., Diemer, B., et al. 2020, ArXiv e-prints [arXiv:2010.04166]
- Guo, Q., White, S., Angulo, R., et al. 2013, *MNRAS*, **428**, 1351
- Hadzhiyska, B., Bose, S., Eisenstein, D., Hernquist, L., & Spergel, D. N. 2020, *MNRAS*, **493**, 5506
- Hadzhiyska, B., Bose, S., Eisenstein, D., & Hernquist, L. 2021, *MNRAS*, **501**, 1603
- Ivezic, Ž., Connolly, A., VanderPlas, J., & Gray, A. 2014, *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data*, Princeton Series in Modern Observational Astronomy (Princeton University Press)
- Jespersen, C. K., Cranmer, M., Melchior, P., et al. 2022, *ApJ*, **941**, 7
- Jo, Y., & Kim, J.-H. 2019, *MNRAS*, **489**, 3565
- Kingma, D. P., & Ba, J. 2014, ArXiv e-prints [arXiv:1412.6980]
- Kingma, D. P., Salimans, T., Jozefowicz, R., et al. 2016, ArXiv e-prints [arXiv:1606.04934]
- Kunz, N. 2019, SMOGN, <https://github.com/nickkunz/smogn>
- Lemos, P., Coogan, A., Hezaveh, Y., & Perreault-Levasseur, L. 2023, ArXiv e-prints [arXiv:2302.03026]
- Lima, E., Sodré, L., Bom, C., et al. 2022, *Astron. Comput.*, **38**, 100510
- Lovell, C. C., Wilkins, S. M., Thomas, P. A., et al. 2021, *MNRAS*, **509**, 5046
- Lovell, C. C., Hassan, S., Villaescusa-Navarro, F., et al. 2023, *Machine Learning for Astrophysics*, 21
- Marinacci, F., Vogelsberger, M., Pakmor, R., et al. 2018, *MNRAS*, **480**, 5113
- Montero-Dorta, A. D., & Rodriguez, F. 2024, *MNRAS*, **531**, 290
- Montero-Dorta, A. D., Artale, M. C., Abramo, L. R., et al. 2020, *MNRAS*, **496**, 1182
- Montero-Dorta, A. D., Artale, M. C., Abramo, L. R., & Tucci, B. 2021a, *MNRAS*, **504**, 4568
- Montero-Dorta, A. D., Chaves-Montero, J., Artale, M. C., & Favole, G. 2021b, *MNRAS*, **508**, 940
- Naiman, J. P., Pillepich, A., Springel, V., et al. 2018, *MNRAS*, **477**, 1206
- Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, *ApJ*, **490**, 493
- Nelson, D., Pillepich, A., Springel, V., et al. 2018, *MNRAS*, **475**, 624
- Nelson, D., Springel, V., Pillepich, A., et al. 2019, *Comput. Astrophys. Cosmol.*, **6**, 2
- Neyrinck, M. C. 2008, *MNRAS*, **386**, 2101
- Ortega-Martinez, S., Contreras, S., & Angulo, R. 2024, *A&A*, **689**, A66
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. 2021, *J. Mach. Learn. Res.*, **22**, 1
- Peacock, J. A. 1983, *MNRAS*, **202**, 615
- Pillepich, A., Nelson, D., Hernquist, L., et al. 2018a, *MNRAS*, **475**, 648
- Pillepich, A., Springel, V., Nelson, D., et al. 2018b, *MNRAS*, **473**, 4077
- Planck Collaboration XIII. 2016, *A&A*, **594**, A13
- Ramdas, A., Garcia, N., & Cuturi, M. 2015, ArXiv e-prints [arXiv:1509.02237]
- Rodrigues, N. V. N., de Santi, N. S. M., Montero-Dorta, A. D., & Abramo, L. R. 2023, *MNRAS*, **522**, 3236
- Shi, J., Wang, H., Mo, H., et al. 2020, *ApJ*, **893**, 139
- Springel, V. 2010, *MNRAS*, **401**, 791
- Springel, V., Pakmor, R., Pillepich, A., et al. 2018, *MNRAS*, **475**, 676
- Stiskalek, R., Bartlett, D. J., Desmond, H., & Anbjagane, D. 2022, *MNRAS*, **514**, 4026
- Tailon, G. 2018, 2DKS, <https://github.com/Gabinou/2DKS>
- Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. 2020, ArXiv e-prints [arXiv:1804.06788]
- Villaescusa-Navarro, F., Anglés-Alcázar, D., Genel, S., et al. 2021, *ApJ*, **915**, 71
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *Nat. Meth.*, **17**, 261
- Vogelsberger, M., Genel, S., Springel, V., et al. 2014a, *MNRAS*, **444**, 1518
- Vogelsberger, M., Genel, S., Springel, V., et al. 2014b, *Nature*, **509**, 177
- Wechsler, R. H., & Tinker, J. L. 2018, *ARA&A*, **56**, 435
- White, S. D. M., & Frenk, C. S. 1991, *ApJ*, **379**, 52
- Wu, J. F., Jespersen, C. K., & Wechsler, R. H. 2024, *ApJ*, **976**, 37

¹² https://github.com/nvillanova/halo_galaxy_connection_probML

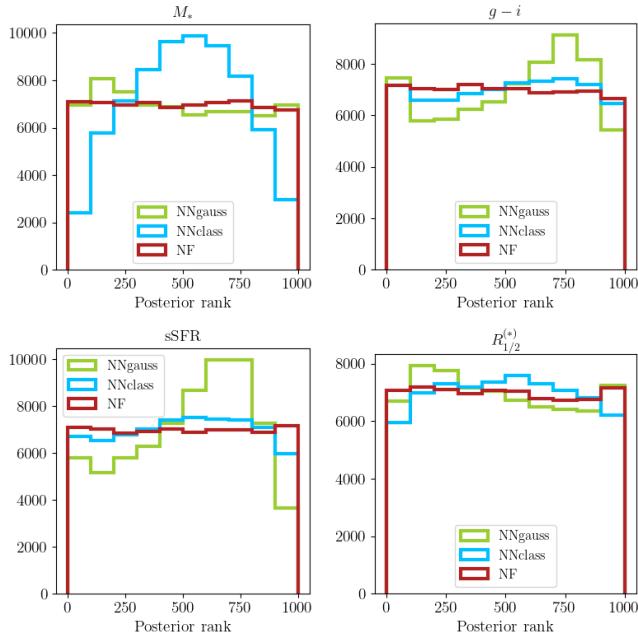


Fig. A.1. Simulation based calibration rank statistics (see §5.4) computed for each galaxy property and for each ML model.

Appendix A: Additional results

A.1. Simulation based calibration

Fig.A.1 shows the SBC rank statistics for each galaxy property and for each ML method. As discussed in §5.4, a good model should present this score following a uniform distribution. It is computed with 1000 samples per instance, for all test set instances. By visual inspection, we see that NF shows good agreement for all properties, once more validating the quality of the predicted distribution. The observed pattern for NNclass stellar mass prediction suggests that the model is underfitting, which is in agreement with the TARP coverage test result shown in Fig.3. For NNgauss we see once again more pronounced deviations indicating a biased prediction for color and sSFR, which are the properties that present bimodality.

A.2. Tables

This Section contains the Tables comparing the values of the scores presented in §6. The best values for each score and each galaxy property are highlighted in bold.

Table A.1 contains the PCC scores shown in Figure 6. The best scores for all galaxy properties are found for NNgauss, followed by NF predictions. Table A.2 contains the PCC scores shown in Figure 4. In this case, NNgauss provides the best predictions for stellar mass, sSFR, and radius while NF better predicts galaxy color.

Tables A.3 and A.4 contain the values of the 1D and 2D K-S test, respectively (see Fig.4). We have implemented our own 1D K-S test schedule and used the [Taillon \(2018\)](#) repository to compute the 2D K-S test. For the K-S test the best scores follow for NF predictions.

Tables A.5 and A.6 contain the values of the 1D and 2D Wasserstein distance, respectively (see Fig.4). We compute the 1D Wasserstein distance based on [Virtanen et al. \(2020\)](#) and the 2D version using [Flamary et al. \(2021\)](#). Once again NF provides the best scores but now for the 2D Wasserstein, while it holds

the best values for the 1D version, unless for color predictions, when NNclass produces the best results.

Table A.1. Pearson correlation coefficient between the reference and predicted sample for each galaxy property.

	PCC avg	M_*	$g - i$	sSFR	$R_{1/2}^{(*)}$
NNgauss	0.9781	0.7056	0.8112	0.7477	
NNclass	0.9769	0.7010	0.8101	0.7394	
NF	0.9779	0.7057	0.8105	0.7468	

Notes. The predicted value is the average over 1000 samples per instance in the test set.

Table A.2. Pearson correlation coefficient between the reference and predicted sample for each galaxy property.

PCC	M_*	$g - i$	sSFR	$R_{1/2}^{(*)}$
NNgauss	0.9578 \pm 0.0002	0.4938 \pm 0.0023	0.6611 \pm 0.0023	0.5583 \pm 0.0021
NNclass	0.9320 \pm 0.0004	0.4939 \pm 0.0022	0.6591 \pm 0.0027	0.5223 \pm 0.0026
NF	0.9554 \pm 0.0002	0.4971 \pm 0.0023	0.6597 \pm 0.0024	0.5557 \pm 0.0022

Notes. The mean and standard deviation are computed with 1000 samples per instance in the test set.

Table A.3. 1D K-S test computed for each galaxy property.

1D K-S test	M_*	$g - i$	sSFR	$R_{1/2}^{(*)}$
NNgauss	0.0342 \pm 0.0009	0.0326 \pm 0.0007	0.0780 \pm 0.0015	0.0294 \pm 0.0017
NNclass	0.0147 \pm 0.0008	0.0085 \pm 0.0005	0.0164 \pm 0.0010	0.0140 \pm 0.0009
NF	0.0034 \pm 0.0008	0.0073 \pm 0.0014	0.0050 \pm 0.0008	0.0086 \pm 0.0016

Notes. 1D scores are computed with the complete test set catalog, and 1000 samples per instance to calculate the standard deviation.

Table A.4. 2D K-S test computed for each pair of galaxy properties.

2D K-S test	$M_* \times g - i$	$M_* \times \text{sSFR}$	$M_* \times R_{1/2}^{(*)}$	$g - i \times \text{sSFR}$
NNgauss	0.0368 \pm 0.0022	0.0915 \pm 0.0010	0.0399 \pm 0.0019	0.0944 \pm 0.0009
NNclass	0.0213 \pm 0.0015	0.0362 \pm 0.0015	0.0323 \pm 0.0006	0.0249 \pm 0.0004
NF	0.0125 \pm 0.0019	0.0100 \pm 0.0011	0.0136 \pm 0.0015	0.0147 \pm 0.0019

Notes. 2D scores are computed using a randomly drawn subset of 3×10^4 instances from the test set, and 5 samples per instance to compute the standard deviation.

Table A.5. 1D Wasserstein distance computed for each galaxy property.

1D Wasserstein	M_*	$g - i$	sSFR	$R_{1/2}^{(*)}$
NNgauss	0.0037 \pm 0.0002	0.0123 \pm 0.0004	0.0377 \pm 0.0004	0.0064 \pm 0.0003
NNclass	0.0031 \pm 0.0003	0.0033 \pm 0.0004	0.0051 \pm 0.0004	0.0054 \pm 0.0003
NF	0.0012 \pm 0.0002	0.0042 \pm 0.0009	0.0034 \pm 0.0006	0.0017 \pm 0.0003

Notes. 1D scores are computed with the complete test set catalog, and 1000 samples per instance to calculate the standard deviation.

Table A.6. 2D Wasserstein distance computed for each pair of galaxy properties.

2D Wasserstein	$M_* \times g - i$	$M_* \times \text{sSFR}$	$M_* \times R_{1/2}^{(*)}$	$g - i \times \text{sSFR}$
NNgauss	0.0193 \pm 0.0012	0.0334 \pm 0.0020	0.0062 \pm 0.0004	0.0399 \pm 0.0021
NNclass	0.0057 \pm 0.0001	0.0069 \pm 0.0002	0.0064 \pm 0.0002	0.0071 \pm 0.0001
NF	0.0051 \pm 0.0007	0.0054 \pm 0.0007	0.0033 \pm 0.0003	0.0067 \pm 0.0012

Notes. 2D scores are computed using a randomly drawn subset of 3×10^4 instances from the test set, and 5 samples per instance to compute the standard deviation.