THE UNIVERSITY *of* EDINBURGH

Multi-Epoch Machine Learning for Galaxy Formation

Robert J. McGibbon

Doctor of Philosophy
The University of Edinburgh
July 2023

Lay summary

A multitude of complex physical processes are involved in galaxy formation and evolution. In recent years computers have become powerful enough to simulate representative volumes of the Universe and are able to reproduce a number of properties of observed galaxies. However, many challenges still remain in the field. One significant issue is the range of sizes which need to be considered, as processes which are important in shaping galaxies vary from individual stars up to the scale of the Universe itself. Thus, a tradeoff arises between accurately modeling small-scale phenomena and simulating large cosmic volumes. Another obstacle lies in the complex interplay between the different processes involved, which can make it difficult to distinguish the specific factors responsible for determining a particular galaxy property. In this thesis I demonstrate how machine learning can help to alleviate some of these problems. Machine learning is a field that enables computers to discern patterns directly from data, bypassing the need for explicit human instruction. By applying machine learning techniques to the data generated by galaxy simulations, I aim to address the tensions mentioned above.

In the first part of my thesis I introduce a model that can be used to produce galaxy catalogs which span huge volumes of the universe. This method runs many times faster than a standard simulation. I show how my method is an improvement on previous work, and then use the catalog it generates to compare with observations of quasars at early times in the Universe.

The subsequent chapters explore the ability of machine learning to provide insights into a simulation. I present two novel methods to do this, and use them both to compare different simulations. In one instance I focus on unraveling the mechanisms driving the buildup of stellar mass in galaxies. In the second case I investigate the flow of gas into and out of galaxies, exploring its influence on the growth of black holes.

Abstract

In this thesis I utilise a range of machine learning techniques in conjunction with hydrodynamical cosmological simulations. In Chapter 2 I present a novel machine learning method for predicting the baryonic properties of dark matter only subhalos taken from N-body simulations. The model is built using a tree-based algorithm and incorporates subhalo properties over a wide range of redshifts as its input features. I train the model using a hydrodynamical simulation which enables it to predict black hole mass, gas mass, magnitudes, star formation rate, stellar mass, and metallicity. This new model surpasses the performance of previous models. Furthermore, I explore the predictive power of each input property by looking at feature importance scores from the tree-based model. By applying the method to the LEGACY N-body simulation I generate a large volume mock catalog of the quasar population at $z = 3$. By comparing this mock catalog with observations, I demonstrate that the IllustrisTNG subgrid model for black holes is not accurately capturing the growth of the most massive objects. In Chapter 3 I apply my method to investigate the evolution of galaxy properties in different simulations, and in various environments within a single simulation. By comparing the Illustris, EAGLE, and TNG simulations I show that subgrid model physics plays a more significant role than the choice of hydrodynamics method. Using the CAMELS simulation suite I consider the impact of cosmological and astrophysical parameters on the buildup of stellar mass within the TNG and SIMBA models. In the final chapter I apply a combination of neural networks and symbolic regression methods to construct a semi-analytic model which reproduces the galaxy population from a cosmological simulation. The neural network based approach is capable of producing a more accurate population than a previous method of binning based on halo mass. The equations resulting from symbolic regression are found to be a good approximation of the neural network.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Parts of this work have been published in McGibbon & Khochfar (2022).

Parts of this work have been published in Natarajan et al. (2023).

Parts of this work have been published in McGibbon & Khochfar (2023).

(*Robert J. McGibbon, July 2023*)

Acknowledgements

I am immensely grateful to my supervisor, Sadegh Khochfar, for his support and guidance throughout my years of study. His extensive knowledge and genuine enthusiasm for my project have been instrumental in the completion of this thesis. I feel incredibly privileged to have had the opportunity to collaborate with him.

I would like to extend my heartfelt appreciation to Priya Natarajan and Britton Smith for their valuable expertise, support, and engaging conversations. Additionally, I am indebted to my fellow students within the TMOX group who offered many helpful comments during our weekly meetings.

I must express my sincere thanks to those who helped put me in a position to carry out a PhD. Most particularly I want to acknowledge Dave Jess, Matt Probert, Alessandro Ferraro for supervising me during my undergraduate studies. I am also thankful for the financial support I received from an STFC Studentship.

I'd like to thank everyone who has made the Royal Observatory Edinburgh such a great place to work. I especially want to acknowledge Dennis, Léa, Massi, and Ryan, for fun conversation and constant support.

The greatest thanks must go to my family. To my Mum and my brother for their unconditional love. Finally to my late Dad, from whom I inherited a passion for physics and computing, I wish I could have shared my project with you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At the beginning of the previous century, the prevailing scientific consensus was that the Milky Way enclosed the entire Universe. However, scientists who proposed that the scale of the Universe was much more extensive became increasingly vocal. This culminated in the Great Debate in 1920 between Harlow Shapley and Heber Curtis, which centered on the nature of ”spiral nebulae”. Curtis argued that these nebulae were distinct galaxies, massive structures located far from our own galaxy. Curtis' ideas were soon validated by the observations of Edwin Hubble, who used Cephid variables (stars of known luminosity) to measure the distance to these nebulae. With these observations a new branch of astronomy was born, and in the years since there have been many great advances in our understanding of these distant objects. However, our knowledge of how galaxies form and evolve into the wide range of structures that we observe today is still incomplete.

Advances in computing power over the previous decades have allowed for significant gains in terms of our ability to model galaxy evolution, and hence increased our understanding of the interplay of the complex physical processes at work (Somerville & Davé, 2015; Vogelsberger et al., 2020). The rises in processing power and memory have allowed for simulations to be run that model galaxies directly and resolve individual parts, rather than treating them as a whole.

Over the past several years the field of machine learning has grown immensely (Pugliese et al., 2021). This too is due to increased computational power, which has made it much more feasible to extract information from large datasets. The field of machine learning is an area of computer science where machines (computer

models) learn to express functions without being explicitly programmed to do so. Applications of machine learning are widespread, from detecting tumors to self-driving cars. It is especially advantageous for the processing of data sets that are far too large for humans to evaluate by traditional methods. Due to the enormous number of observations, and the size of modern day simulations, astronomy is a "big data" field. Some telescopes due to be built in the next decade will produce petabytes of data per day. Consequently machine learning methods lend themselves well to astronomy, and indeed they have already seen widespread use in many areas within astrophysics.

The main focus of this thesis is the combination of these two methods. By using machine learning in conjunction with cosmological simulations we can gain two main advantages. The first is by using machine learning techniques to help alleviate some of the computational expenses associated with modelling galaxy evolution. The second is to gain greater understanding of the physics of complex processes involved in galaxy formation by interpreting machine learning models fitted to modern simulations.

1.1 Cosmology and galaxy formation

In this section I give a high level overview of our current understanding of how galaxies formed. I begin with the standard model of cosmology, because in order to understand galaxy formation we must first place it within a cosmological context.

1.1.1 The Lambda cold dark matter cosmological model

Model overview

The current prevailing cosmology within which galaxy formation occurs is the Lambda Cold Dark Matter (ΛCDM) model. If is founded based on the cosmological principle, which states that when viewed on large enough scales the distribution of matter in the Universe is both homogeneous (has the same properties at every point) and isotropic (looks the same in every direction). It assumes that Einstein's theory of general relativity is the correct theory of gravity for large scales. In this paradigm the energy budget of the Universe,

which is the source of spacetime curvature, is divided up into dark energy (in the form of a cosmological constant), baryonic matter (ordinary matter including protons, neutrons, and electrons which makes up gas, stars, and dust), and dark matter. Dark matter is assumed to be cold (has a velocity much lower than the speed of light), and collisionless (dark matter particles only interact with other particles through gravity, and potentially the weakly force). ΛCDM is able to successfully explain observed phenomena such as the cosmic microwave background, the accelerating expansion of the Universe, and large scale structure.

The origins of the ΛCDM model are based on observations taken by Edwin Hubble in the 1930s. He measured the velocity at which galaxies are moving relative to us, and also their distance based on the brightness of Cepheids (Hubble, 1929). He discovered that distant galaxies are moving away from us with a velocity that is proportional to their distance. This result is described by the equation known as Hubble's Law,

$$v = H_0 r \tag{1.1}$$

where v is the velocity of the galaxy, r is the proper (not comoving) distance to the galaxy, and $H_0 \approx 70\,\mathrm{km\,s^{-1}Mpc^{-1}}$ is the Hubble constant. This provided the first evidence for the expanding Universe, as all galaxies appear to be moving away from us. This implies that space itself is expanding, as otherwise the cosmological principle would be violated.

Therefore to relate positions at different times the scale factor $a(t)$ is used, which is defined to be equal to 1 at the present day, and <1 at earlier times when the Universe was smaller.

$$r(t) = a(t)x \tag{1.2}$$

where x is the comoving distance. By differentiating this equation we find that the Hubble constant is equal to the value of the Hubble parameter, $H(t) = \dot{a}/a$, evaluated at the present day.

The stretching of space also affects the light that we observe from distant objects. The redshift of an object, z, relates the observed wavelength λ_{obsv} to the wavelength of the emitted light λ_{emit} by

$$z = \frac{\lambda_{obsv} - \lambda_{emit}}{\lambda_{emit}} \tag{1.3}$$

As the effect on the wavelength of light is dependent on the amount of space the light has traveled through, the redshift is related to the scale factor at which the light was emitted by

$$a = \frac{1}{1+z} \tag{1.4}$$

The evolution of the scale factor describes the expansion history of the Universe. Its evolution is given by the Friedmann equation, which is derived by combining the conservation of energy, the cosmological principle, and the equations of general relativity (Liddle, 2003). It is given by

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{kc^2}{a^2} + \frac{\Lambda c^2}{3} \tag{1.5}$$

where c is the speed of light, G is the gravitational constant, ρ is the total energy density of the matter plus radiation in the Universe, k is the curvature (-1, 0, 1 for negatively curved, flat, positively curved universes respectively), and Λ is the cosmological constant. The critical density is defined such that we get zero curvature for a universe only composed of matter, which gives

$$\rho_c = \frac{3{H_0}^2}{8\pi G} \tag{1.6}$$

Current observational evidence suggests that the Universe is spatially flat. Density parameters are expressed as a fraction of the critical density, e.g. the matter density parameter is given by

$$\Omega_m = \frac{\rho_m}{\rho_c} \tag{1.7}$$

As the energy density of the different components scales according to different powers of the scale factor, the Universe began with a radiation dominated era for the first ~50,000 years after the big bang. It then moved to a matter dominated era when the energy density of matter exceeded that of radiation and dark energy. This lasted until the Universe was approximately 10 billion years old, at which point dark energy became dominant, as it remains today.

Primordial nucleosynthesis

Immediately after the big bang the Universe contained a plasma of free protons, neutrons and electrons. Due to the high temperatures, and available reactions, such as protons and electrons combining to form neutrons and neutrinos, at approximately one second after the big bang the neutron-to-proton ratio was close to 1:1. As the universe expanded it cooled, and as the temperature dropped the equilibrium shifted in favour of protons due to their lower mass. This caused the neutron-to-proton ratio to drop, but this was stopped by the formation of atomic nuclei, within which neutrons are stable. A full treatment of this process predicts that it will produce a mass fraction of approximately 75% hydrogen and 25% helium, with trace amounts of heavier elements such as lithium (Alpher et al., 1948). The first stars and galaxies were formed from this primordial hydrogen-helium gas. Heavier elements, referred to in astronomy as metals, are produced in stars by stellar nucleosynthesis processes (Burbidge et al., 1957).

Recombination

At this point the Universe still consisted of a plasma of the newly formed nuclei and electrons. However, approximately 370,000 years later the Universe had cooled (to ~3000K) such that it was now energetically favourable for electrons and protons to combine to form neutral hydrogen atoms. This point is known as recombination, although protons and electrons had not been combined before this time. It is also known as the ”surface of last scattering”, since photons could now travel freely in the absence of free electrons. The photons which decoupled from matter at this time have been redshifted (corresponding to a redshift of ~1100) into the microwave spectrum, and can be observed as the cosmic microwave background (CMB). The CMB was first detected by Penzias and Wilson when they discovered a background microwave signal they could not

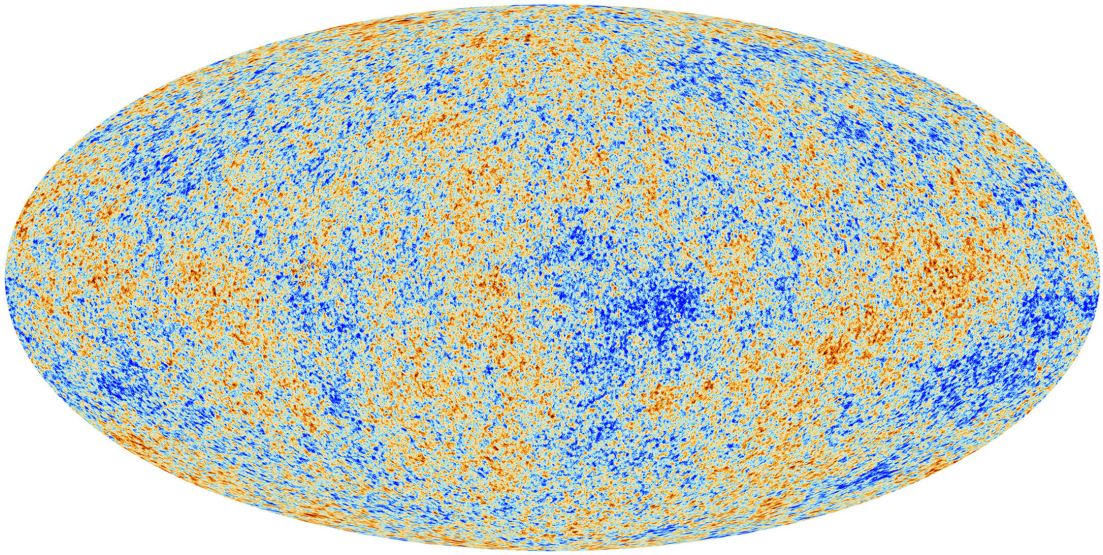

Figure 1.1 *A map generated using data from the Planck satellite of temperature fluctuations in the cosmic microwave background. Red colours indicate higher temperature regions. The variations have a root-mean-squared amplitude of 18μK, demonstrating that the Universe was highly uniform at the time of recombination.*

account for (Penzias & Wilson, 1965).

The CMB is described by a near perfect black-body spectrum, with a constant temperature of approximately 2.7K in every direction of the sky. However, it is not completely uniform, and has flucuations of one part in a hundred thousand, as shown in Figure 1.1. A succession of precision satellites have analysed the CMB in increasing detail - COBE (Smoot et al., 1992) in the 1990s, WMAP (Bennett et al., 2013) from 2001 to 2010, and most recently Planck (Planck Collaboration et al., 2016). The nature of the anisotropies gives us strong constraints on the properties of the Universe at this very early time, and so provides information about the curvature and matter content of the Universe.

Dark matter

Current measurements suggest that $\Omega_m \approx 0.3$. The baryonic matter that we are able to observe is not enough for this to be the case. Instead it is assumed the majority of the mass of matter is given by dark matter. This has been hypothesised ever since a number of observations at the start of the 20th century. In 1933 Fritz Zwicky measured the velocities of galaxies within the Coma Cluster (Zwicky, 1937). He calculated that the mass within the cluster (as estimated

based on the visible light) was far too small to explain the high orbital velocities of the galaxies. Without additional mass these galaxies should have escaped the cluster. During the 1970s Vera Rubin observed the rotation curves of spiral galaxies, and found that the observed matter within the systems was not enough in order to account for the high rotational velocities in the outskirts of the galaxies (Rubin et al., 1980). More evidence for dark matter comes from gravitational lensing. General relativity predicts that as matter curves spacetime, then the path along which light travels will be affected. This means that massive compact objects can magnify background galaxies, and sometimes even produce multiple images of them. By modelling the lensing effects of massive clusters an estimate of their mass can be obtained, and it is found to be significantly larger than the amount of visible matter (Massey et al., 2010).

Despite the strong evidence for the existence of dark matter it has never been directly detected, and we do not yet know what it is made of. Current candidates include WIMPs and axions. There have been many attempts at particles physics facilities, such as CERN, to produce such dark matter particles. The other option is direct detection, where large sensitive detectors are placed deep underground to shield them from other particles. These detectors are monitored for impacts with dark matter particles. However, such events are expected to be very rare, and at this time there has been no confirmed detection of a dark matter signal.

Dark energy

In 1998 two teams independently observed the luminosity and redshift of Type 1a supernova in distant galaxies (Perlmutter et al., 1999; Riess et al., 1998). Type 1a supernova are exploding white dwarfs which have exceeded their stability limit, and therefore all explode at a similar mass. This means their luminosity is known and so they are referred to as standard candles. Their distance can therefore be calculated based on their observed magnitude. Both teams discovered that the supernova were systematically fainter than expected given their redshift. This is explained by a universe where the expansion rate is increasing with time, rather than decreasing as would be the case in a matter dominated universe. Therefore there must be another component in the energy density of the Universe which opposes gravity. Observations of the CMB suggest the missing energy is approximately ~70% of the energy budget of the Universe. Like dark matter, the nature of this energy is unknown, and so it received the name dark energy. It

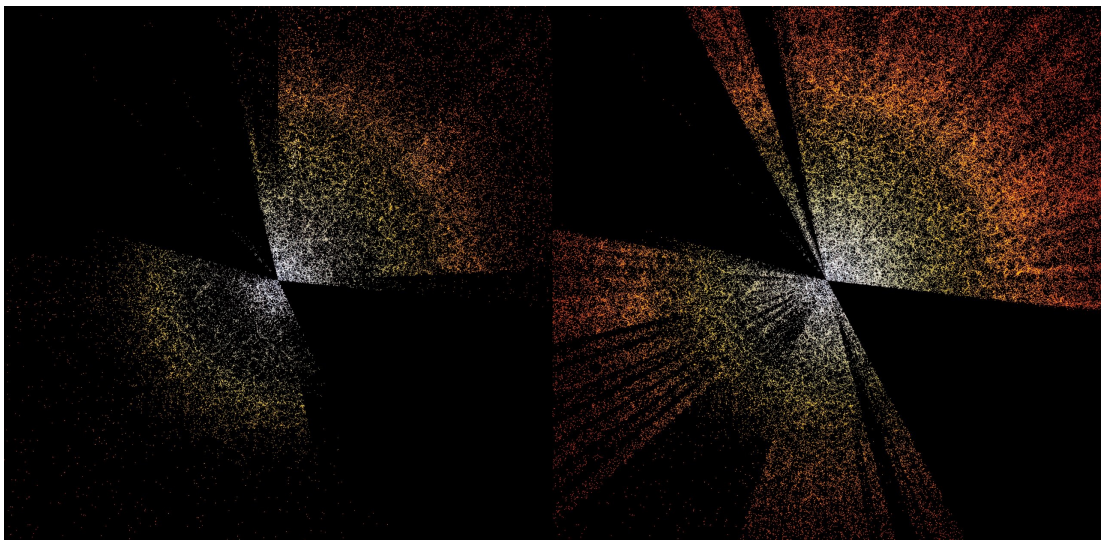

Figure 1.2 *The image on the left shows the 4 million galaxies and quasars mapped by Sloan survey from 2000 to 2020. The right images shows the 7.5 million objects observed by the DESI survey during its 7 first months of operation.*

has been suggested that it is a property of space itself. This energy would not be diluted as the universe expands and so is referred to as a cosmological constant, represented by the letter Λ. Most quantum field theories predict a value for the energy density of a vacuum which results from quantum fluctuations. However the measured cosmological constant is smaller than that calculated by quantum theory by a factor of $\sim 10^{120}$ (e.g. Martin, 2012).

Large Scale Structure (LSS) surveys

One of the most useful tools in recent years to help constrain properties of dark matter and dark energy has been large scale structure surveys. These observe the position and redshift of many galaxies, and combine these measurements within a given cosmological model to create a 3D map of the Universe. Some of the most significant LSS surveys carried out in the past decades include the Two-degree-Field Galaxy Redshift Survey (Colless et al., 2001), the Sloan Digital Sky Survey (SDSS) (York et al., 2000), and the Extended Baryon Oscillation Spectroscopic Survey (eBOSS) (Dawson et al., 2016).

The next generation of surveys is underway, which will acquire massive volumes of data. One of these is the Dark Energy Spectroscopic Instrument (DESI) (DESI Collaboration et al., 2016) that saw its first light in 2019 and is currently ongoing.

The galaxies observed in the first 7 months are shown in Figure 1.2. DESI is expected to run for 5 years and observe the redshift of 40 million quasars and galaxies. Other projects include the Euclid space telescope (Racca et al., 2016) which launched in July 2023, and the Legacy Survey of Space and Time (LSST) (Ivezić et al., 2019) which will be carried out at the Rubin Observatory. LSST aims to measure the redshift of billions of galaxies. For these upcoming surveys machine learning is an extremely promising tool, both for processing the data from the instruments, and for producing mock galaxy catalogs in order to compare predictions of models with the observations.

LSS surveys allow for the clustering of matter to be investigated. As can be seen in Figure 1.2, the distribution of galaxies is not uniform, and includes high and low density regions. This pattern of clusters, filaments, and voids is known as the cosmic web. Significant departures are observed in clustering from universes that only contain dark matter compared with those that contain baryons. One of the most successful validations of ΛCDM is the existence of baryon acoustic oscillations (BAOs). These result from the fact that in the early Universe dark matter was attracted towards the centre of overdensities due to the gravitational potential. However, photons and baryons moved outwards from the overdensities because of the pressure difference. As photons and baryons decoupled this pressure was relived. This left behind shells of baryonic matter at a wavelength that is well predicted by theory. BAOs were first measured by the 2dFRS and SDSS in 2005 (Cole et al., 2005; Eisenstein et al., 2005).

Open questions

Despite the successes of the ΛCDM model, a number of challenges remain. These result from comparisons between numerous simulations and observations, and suggest that an extended cosmological model may be required. Some of the major challenges include:

- **Tensions in cosmological parameters** Multiple methods exist to measure the value of the Hubble constant. "Late universe" measurements rely on standard candles and produce a result of $H_0 = 73\ \mathrm{km\ s^{-1}Mpc^{-1}}$ (e.g. de Jaeger et al., 2020; Riess et al., 2021). "Early universe" measurements produce a value of $H_0 = 67.7\,\mathrm{km\,s^{-1}Mpc^{-1}}$ (e.g. Planck Collaboration et al., 2016; Ivanov et al., 2020). Late and early measurements of the S_8 parameter also differ. For a recent review see Abdalla et al. (2022).

- **Violations of isotropy** A hemispheric bias has been detected in observations of both the CMB (Planck Collaboration et al., 2014b) and the distribution of quasars (Secrest et al., 2021). Until an explanation is found this presents a challenge to the ΛCDM model as it contradicts the underlying assumption that the universe is isotropic and homogeneous on large scales.

- **Cusp-core problem** Halos within simulations have a density profile that increases steeply at small radii, and are referred to as cusps. Rotation curves of observed dwarf galaxies suggest that they have a flat central density, referred to as a core. This problem could be solved if baryonic feedback is able to produce outflows that can transfer energy to the collisionless dark matter.

Alternative proposed cosmologies include modified gravity, which is a replacement for dark matter that explains galaxy rotation curves by introducing a change in the law of gravity at small acceleration values (Nojiri et al., 2017). Other theories rely on more exotic forms of dark matter (Arun et al., 2017), e.g. self-interacting dark matter where the scattering of particles at the centre of halos can cause cores to be formed.

1.1.2 Galaxy formation

In this section I discuss some key observed properties of galaxies. I then describe how dark matter halos assemble, and how galaxies can form within those halos.

Observed properties of galaxies

Understanding the formation and evolution of galaxies must stem from observations. A common classification of galaxies is based on their morphology. This is commonly done according the Hubble sequence (or tuning-fork diagram), as shown in Figure 1.3.

Spiral (or disk) galaxies tend to have a mass in the range of 10^9 - $10^{12} M_\odot$ and are made up of two components. The first is a rotationally supported disk of stars, gas, and dust. The stellar population in the disk tend to be young, with star formation ongoing (Martinsson et al., 2013). This makes them appear blue in colour, as defined by the difference in magnitude between two photometric bands

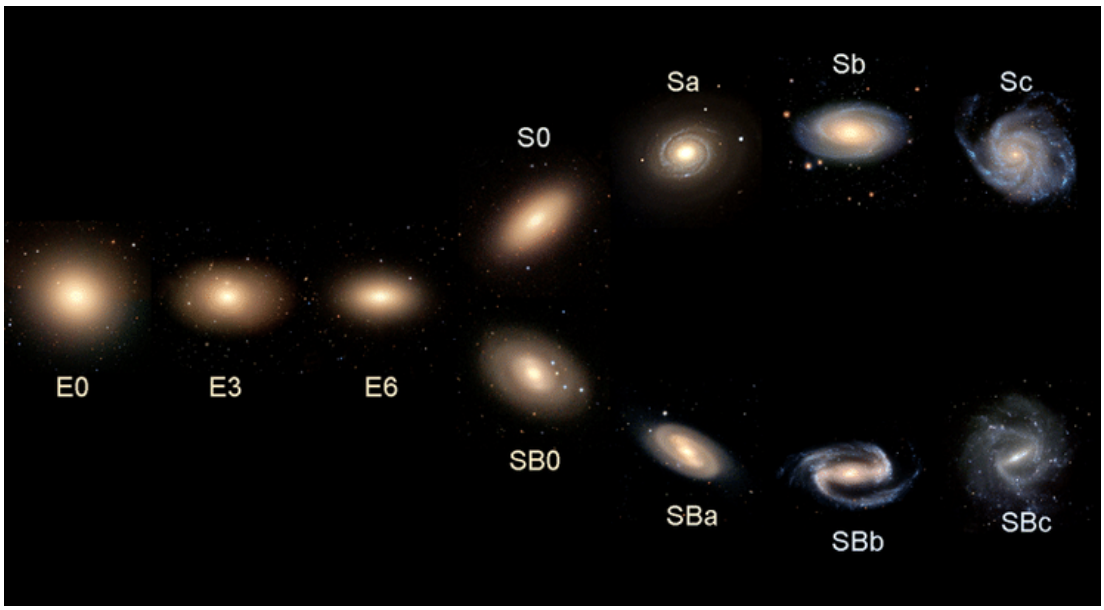

Figure 1.3 *Hubble tuning fork, showing ellipticals (E0 - E7), lenticulars (S0), spirals (Sa-Sc), and barred spirals (SBa - SBc). Taken from Cui et al. (2014)*

(Johnson, 1966). The stars have approximately circular orbits, and are usually metal-rich. There are also spiral arms within the the disk, which are most visible when observing young stars or molecular gas (Nishiyama & Nakai, 2001). The second main structure in a disk galaxy is its central bulge, a spherical collection of tightly packed stars found at the centre of the galaxy. These stars are older and lower metallicity than the stars in the disk. They have randomised orbits, such that the bulge is dispersion supported. Some spiral galaxies contain a central bar-like structure of stars, which connects the bulge to the spiral arms (Kruk et al., 2018). The presence of a bar determines which of the parallel branches on the right of the Hubble sequence a spiral galaxy will end up on.

On the left hand side of Figure 1.3 are elliptical galaxies. They are completely bulge dominated, with smooth brightness profiles (Chitre & Jog, 2002). E0 galaxies are spherical, with galaxies becoming more elongated as we move towards E7. Elliptical galaxies contain very little cold gas or dust, and hence have low star formation rates (SFRs) compared with spiral galaxies (Martig et al., 2013). Their stellar populations are old and metal-rich. The most massive galaxies are ellipticals, with masses up to $10^{13} M_{\odot}$.

S0 galaxies, also known as lenticulars, share characteristics of spirals and ellipticals. They tend to have the disk structure of a spiral galaxy, but with little ongoing star formation (van den Bergh, 2009).

Dwarf galaxies are a class of objects which do not fit within the Hubble sequence. They come in two types. Dwarf irregulars have highly asymmetric structures, and ongoing star formation (Parodi & Binggeli, 2003). Dwarf ellipticals have more regular structure and are found within galaxy clusters (Mistani et al., 2016), but tend to be quenched (Geha et al., 2012).

Gas in dark matter halos

Fluctuations in the density field that results from inflation will grow over time due to self-gravity (Peebles, 1980). Regions that are overdense will attract more matter, causing their density contrast to increase over time. The opposite will occur for underdense regions. The evolution of structure in the universe is a result of this gravitational instability. The overdensity at a point is defined as

$$\delta(x) = \frac{\rho(x) - \bar{\rho}}{\bar{\rho}} \tag{1.8}$$

The spatial distribution of baryons and dark matter is initially very similar. However, as the overdensity starts to collapse, the two components behave very differently. Within ΛCDM dark matter is collisionless, and so the particles can move freely. The spherical collapse model (Gunn & Gott, 1972; Lahav et al., 1991) is used to characterise the non-linear evolution of overdensities. In this model the density perturbation is modelled as a spherically symmetric fluctuation. The sphere will expand at a slightly slower rate than the rest of the universe due to the enhanced gravitational force, further increasing its density contrast. If the initial density is large enough then the expansion of the sphere will be halted, and the particles will virialize. Calculations show that the mean density of the sphere will be ~ 200 times the critical density of the universe. The virialized sphere is referred to as a halo.

Baryons are collisional and so pressure affects their dynamics. For a spherical symmetric overdensity the gas in the centre is in a hydrostatic state, where the pressure force balances the gravitational force. Cold gas falls into the halo until it hits the inner region. At this radius the infall velocity of the gas is dissipated into heat and a shock front develops, i.e. there is sharp change in the gas density and pressure.

According to the virial theorem the kinetic energy of a system is equal to half of

its potential energy. Equating the thermal energy per unit volume with half the potential energy per unit volume yields

$$\frac{3}{2}\frac{\rho}{\mu m_p}k_B T = \frac{\nu}{2}\rho\frac{GM}{R} \tag{1.9}$$

where ρ is the gas density, μ is the mean molecular weight, m_p is the proton mass, k_B is the Boltzmann constant, and the value of $\nu \sim 1$ is dependent on the density profile of the halo. The final term on the R.H.S. is equal to the square of the circular velocity of the halo, ${V_c}^2$. Thus we arrive at the following expression for the virial temperature of a halo

$$T_{vir} = \frac{\mu m_p}{3k_B}{V_c}^2 \tag{1.10}$$

For galaxy groups and clusters the virial temperature is hot enough that the gas emits X-rays, which can be easily detected. The characteristic temperature for galaxy-mass halos is $\sim 10^6$K, which is more difficult to observe.

Gas cooling

Stars are born inside dense clouds of gas, but in order to form these high density regions the gas must first cool and collapse to the centre of a halo. To do this the gas releases energy in the form of photons in a process known as radiative cooling. There are many mechanisms by which this can occur, such as scattering between electrons and nuclei (Bremsstrahlung Radiation) or collisions causing electrons to enter an excited state. The most relevant processes all depend on two-body interactions, which means their rate is dependent on the square of the gas density. For hydrogen rich gas the cooling rate is therefore defined as $C = \Lambda(T){n_H}^2$, where $\Lambda(T)$ is called the cooling function and depends on the temperature and composition of the gas, and n_H is the number density of hydrogen.

With the cooling function defined, we can calculate the time it will take for the gas to dissipate all its thermal energy. The cooling time is defined as

$$t_{\text{cool}} = \frac{3nk_bT}{2C} \tag{1.11}$$

The other relevant timescale is the free-fall time, which is the time taken for a free falling particle to reach the center of a halo. It is given by

$$t_{\text{ff}} = \sqrt{\frac{3\pi}{32G\rho}} \tag{1.12}$$

If the cooling time is less than the free-fall time, then gas will be able to fall freely towards the centre of the halo without being stopped by gas pressure. Therefore the condition $t_{\text{cool}} = t_{\text{ff}}$ is a criteria for determining whether gas can fall into the center of a halo and form dense regions.

Figure 1.4 shows the range of temperatures and densities for which cooling is effective. The cooling function of primordial gas drops off sharply below $10^4 K$. This means that cooling cannot take place efficiently in low mass halos, excepts at very high redshifts when the corresponding density is still high. For halos with masses from $10^9 - 10^{12} M_\odot$ cooling is efficient, so gas can collapse to the centre to form stars. All halos contain substructure (Wang et al., 2020), smaller gravitationally bound objects embedded within a parent halo. For the most massive halos gas collapses within its subhalos rather than falling to the halo centre. From this groups and clusters are formed, which contain large numbers of galaxies.

In reality the way gas collapses is more complicated and not spherical. The cosmic web is made up of filaments, and gas can flow along these to reach the central region of a halo without being heated. Simulations have shown that is an significant mechanism for halos to accrete gas (Kereš et al., 2005; Dekel et al., 2009; van de Voort et al., 2011).

Formation of spiral and elliptical galaxies

How do we get the observed diversity of types of galaxies as described in Section 1.1.2? When halos form they attain an angular momentum, because of torques

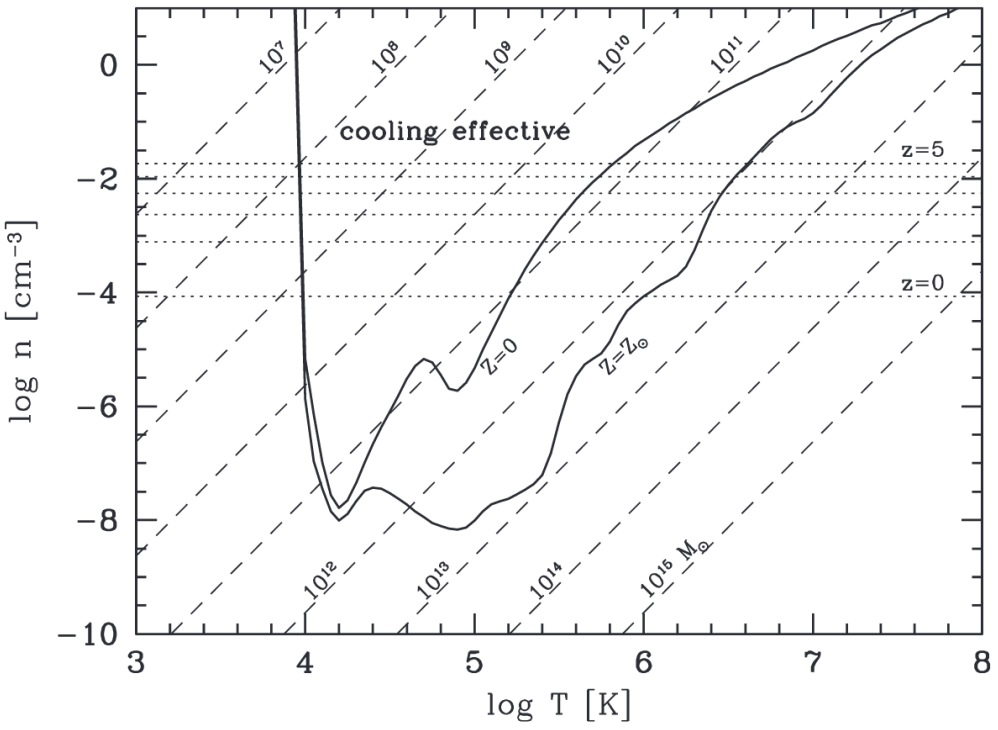

Figure 1.4 *Cooling diagram with solid curves indicates where $t_{cool} = t_{ff}$ in the n–T plane. Cooling is effective in the region above the curves. The upper and lower curves correspond to gas with zero and solar metallicity, respectively. The tilted dashed lines are lines of constant gas mass, while the horizontal dotted lines show the densities expected for virialized halos at different redshifts. Taken from Mo et al. (2010)*

caused by external gravitational forces. As gas cools and collapses towards the centre of a halo its angular momentum is conserved, and this prevents it from collapsing further. Frictional forces between particles act to move the gas onto approximately circular orbits where the angular momentum vector is normal to the plane of the gas. Thus a flat disk is formed. Within the disk giant molecular clouds (GMCs) form, with a high fraction of their hydrogen in H_2. Molecular hydrogen is crucial because it is still efficient at cooling at temperatures below 10^4K. The cloud will then undergo further gravitational collapse if it is massive enough to overcome its gas pressure. The cloud will fragment into protostellar clouds which will continue to collapse until they form stars. Thus spiral galaxies are formed with stars in a disk. The distribution of masses of the new stars is given by the initial mass function (IMF), although the exact shape of the IMF and its sensitivity to environmental conditions remains unknown (Bastian et al., 2010).

Elliptical galaxy formation is more difficult to explain. Unlike stars found in spiral galaxies, the stars in ellipticals have a high velocity dispersion. A clue to their formation is the fact that ellipticals galaxies preferentially live within groups and clusters, where a large number of galaxy-galaxy interactions occur. Mergers of halos occur when two halos combine to form one with a larger mass. Mergers are divided into two types. For major mergers the mass of the halos is similar, for minor mergers one halo is significantly larger than the other, with a mass ratio of 1:3 typically used as the dividing point. Both types of mergers occur when satellite subhalos and their galaxies experience dynamical friction (Boylan-Kolchin et al., 2008). This causes them to lose energy and infall into the centre of the host halo. For minor mergers the large scale properties of the accreting galaxy do not tend to change that much. However the picture is different for major mergers. In this case the disks of the two merging galaxies will be wiped out as the stellar population obtain a high velocity dispersion, forming an elliptical galaxy.

Stellar feedback

Figure 1.5 shows a diagram summarising the difference between the calculated halo mass function and the observed galaxy stellar mass function. We see that halos must not form a constant fraction of stars. Some of this is explained by the efficiency of gas cooling as discussed above, but another significant effect is feedback, which is when energy from stars or AGN is added to the gas.

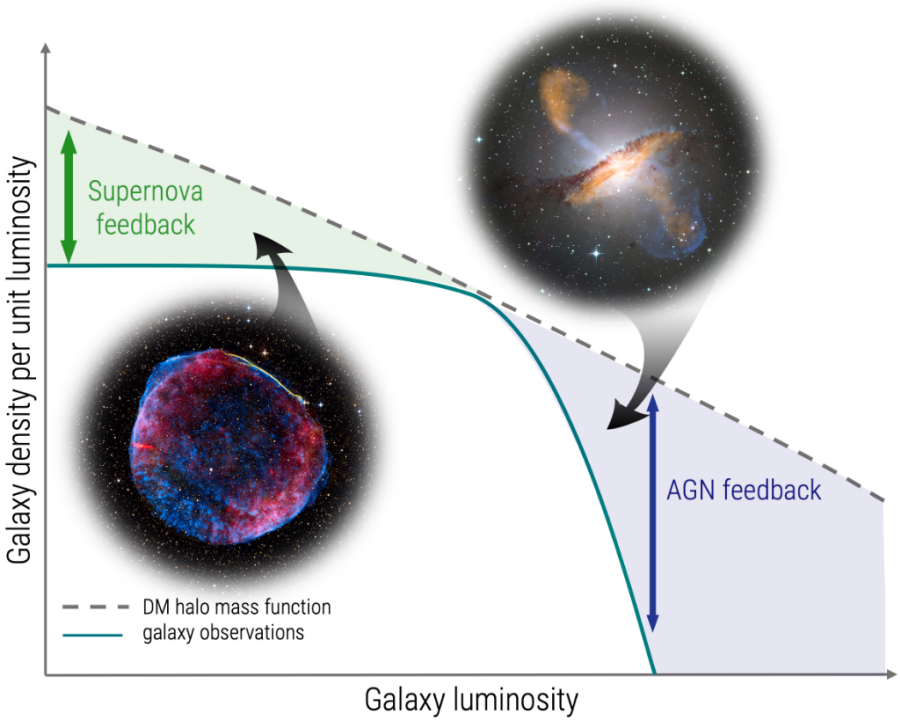

Figure 1.5 *Visualisation of the difference between the halo mass function and galaxy stellar mass function. For large halos the stellar-to-halo mass relation (SHMR) is decreased because of AGN feedback, for small halos it is due to supernova feedback. Taken from Piotrowska-Karpov (2022)*

Feedback can be categorized into two main types: preventive and ejective. Preventive feedback hinders star formation by preventing gas from accreting into the interstellar medium (ISM), while ejective feedback involves processes that remove gas from the ISM after it has already been accreted. This shapes the efficiency of star formation in different halos and different environments.

Stars emit electromagnetic radiation, especially young massive stars which produce large amounts of photons with high enough energy to photodissociate molecular hydrogen and ionise neutral atoms. This affects the surrounding gas, preventing molecular hydrogen from acting as a coolant, causing further star formation to cease. Radiation is also responsible for reionization, which is when the intergalactic medium (IGM) became fully ionized over the redshift range from $z \simeq 10-15$ to $z \simeq 5-6$ (Robertson, 2022; Bosman et al., 2022; Goto et al., 2021).

Supernova are extremely bright transient events, with a peak luminosity that can be comparable to the rest of their host galaxy, before fading over several weeks or months. Records exist of a supernova in the year 1006 AD (Winkler et al., 2003), with several possible observations predating that. The exact mechanisms are not completely understood, but there are two main types. Type 1a supernova are caused by the accretion of material onto a low mass white dwarf star from a binary companion (Hillebrandt & Niemeyer, 2000). Core-collapse supernova occur because massive stars have used up all their fuel and catastrophic collapse ensues (Smartt, 2009). Core-collapse supernova occur shortly after star formation due to the brief lifetime of massive stars, whereas Type 1a take a lot longer to explode. Both produce enormous amounts of energy, which heats the gas surrounding them, but can also drive winds that remove gas from the ISM (Veilleux et al., 2005).

Active Galactic Nuclei (AGN)

The light emitted by normal galaxies is dominated by stars. As stars can be modelled with black body radiation, the spectrum of a galaxy can be approximated by the superposition of Planck spectra over a relatively narrow temperature range. Some galaxies are observed which have a much broader energy distribution, and the emission arises from a small central region , known as an active galactic nucleus (Seyfert, 1943). It is thought that this is due to the accretion of mass onto a central supermassive black hole (SMBH), and it is now suggested that all galaxies have a black hole at their centre (Kormendy & Ho,

2013).

A wide range of objects are classified as AGN, and within that there are many different types. Our current model for understanding how these classes arise is that the black hole is surrounded by an accretion disk (Netzer, 2015). This disk contains dust, and so obscures our view of the central black hole. Any observations of the source will therefore be highly dependent on the viewing angle relative to the accretion disk. One common classification is based on the width of the spectral lines. The region close to the black hole is known the broad line region (BLR), as there is significant Doppler broadening due to the temperature of the gas and its velocity around the black hole. The region further from the black hole is the narrow line region (NLR), although the "narrow" lines still have considerable broadening. AGN with broad line emission are referred to as Type-1, those without are Type-2.

The formation mechanism of SMBHs is still unclear, especially for some of the high mass objects which are observed at high redshift (Wang et al., 2021). They grow over time by accreting surrounding material. A common model for the accretion rate, as introduced by Bondi (1952), is

$$\dot{M} \simeq \frac{\pi \rho G^2 M^2}{c_s^3} \tag{1.13}$$

where c_s is the sound speed of the gas and ρ is the gas density. The growth rate is restricted since if the luminosity is too high then there will be intense radiation driven winds. This is known as the Eddington limit. The energy that is emitted from accreted gas can have a significant effect on the environment around the black hole. Evidence for this comes from the fact that black holes are observed to follow a number of relations, such as correlations with bulge mass (Magorrian et al., 1998) and stellar velocity dispersion (Ferrarese & Merritt, 2000).

1.2 Numerical simulation of galaxy formation

Modelling galaxy formation and evolution is a demanding problem due to the multi-scale physics involved. Therefore computer simulations have become one of the main tools used to tackle the open questions in the field. In this section I

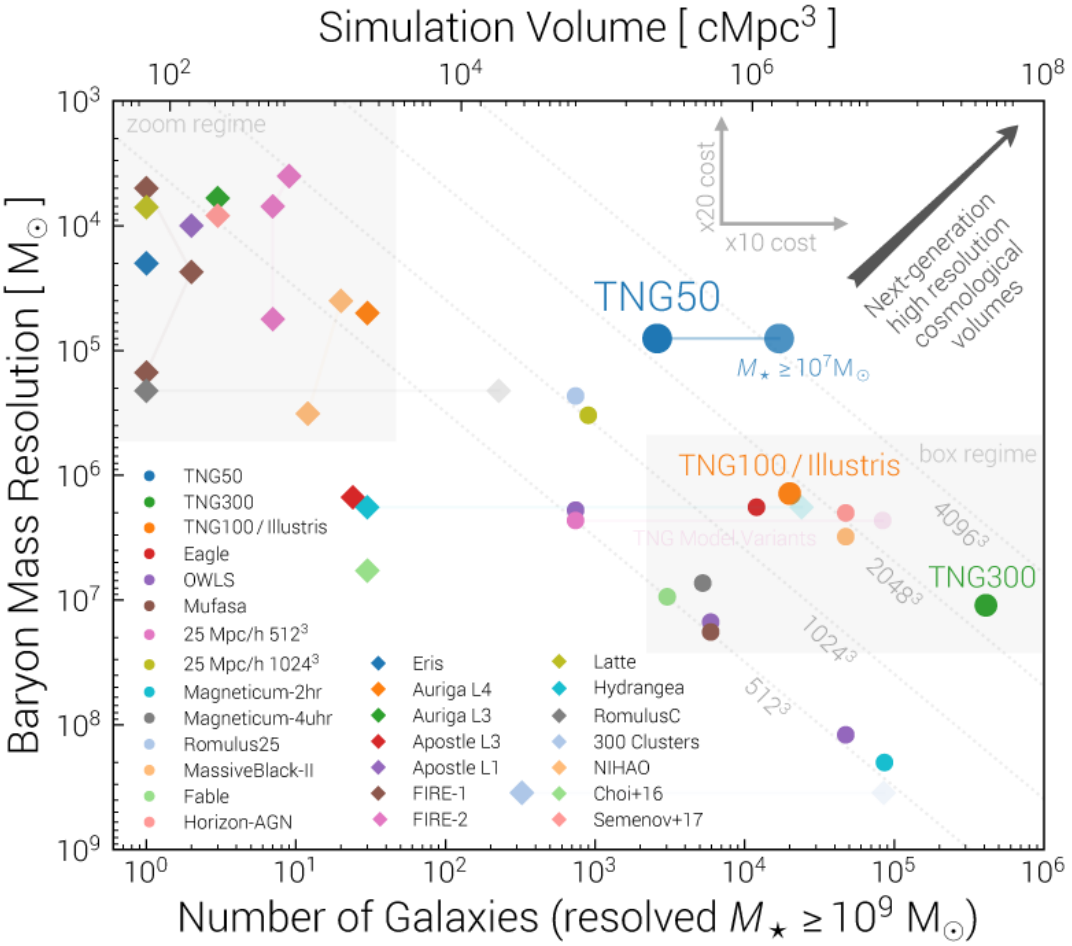

Figure 1.6 *Comparison of the volume simulated against mass resolution for a number of cosmological simulations. Taken from Nelson et al. (2019).*

start with discussion of simulating dark matter. As the majority of matter in the universe is dark, simulating dark matter describes the LSS of the universe, and is the foundation for both hydrodynamical simulations and semi-analytic models (SAMs). I then describe hydrodynamical simulations which directly model gas, along with the range of physical processes necessary for galaxy formation. I finish with a discussion of SAMs.

When deciding what kind of simulation to run, there are several factors that must be considered in order to determine an appropriate method. The two main competing factors are the computational cost of the method versus the accuracy of the method. For simulations the accuracy is often related to the resolution. This sets the level at which the solution is recovered, commonly the space and mass resolution. Increasing the resolution will increase the cost of running the simulation. For 3D simulations increasing the resolution by a factor of 2 will lead

to an increase in the number of particles or cells by a factor of 2^3. When this is combined with the scaling of the different solvers, increasing the resolution by a small factor can have a significant impact on the runtime and memory use of the simulation. There are also tradeoffs to be considered related to running a single expensive simulation with only one set of parameters compared with a suite of lower resolutions simulations that allow for exploration of the parameter space. A comparison of simulations is shown in Figure 1.6. More recent simulations are in the top right of the plot because of increased computational power. I will demonstrate in this thesis how machine learning is able to help alleviate the tension between these two competing factors.

1.2.1 N-body simulations

Simulating dark matter

Dark matter is assumed to obey the collisionless Boltzmann equation, which is coupled with Poisson's equation for gravity. However, the most common method for simulating the collisionless dynamics of dark matter is to use N-body simulations. A review can be found at Dehnen & Read (2011). These simulations follow N particles, each with a position, velocity, and mass. This can be thought of as a Monte Carlo technique, where a set of phase-space points is sampled from the initial phase-space density, and then are evolved through time. Therefore using a high number of particles is preferable to reduce noise.

Newtonian gravity is used since corrections from general relativity are negligible. Gravitational softening, where a constant term is added to the denominator of the force equation, is employed to avoid unphysical scattering when particles get too close. Calculating the gravitational force between each pair of particles in the simulation (a particle-particle scheme) would lead to a runtime that scales as N^2. As contributions from distant particles tend to be negligible, two alternative methods are commonly used. Both of these methods have a $N \log N$ runtime.

As with the direct summation approach, the tree approach solves the integral form of Poisson's equation. Tree codes work by dividing the volume to be simulated into cells. Particles in nearby cells are treated individually when the force is being calculated, but distant sub-volumes are treated as a single particle, with mass equal to the sum of particle masses within that cell, and location as the

center of mass of the cell (Barnes & Hut, 1986). The volume splitting is based on a tree structure. This is commonly done using an octree, where each cubic cell is repeatedly split into eight child cells. This results in a hierarchy of cubic nodes where the root node contains all particles. Interactions are calculated for all particles within a node, and then the interactions between nodes are included.

Another option is the particle-mesh method (Hockney & Eastwood, 1988). A grid is placed within the simulation volume, and the mass of each grid cell is calculated based on the number of neighbouring particles (weighted with a kernel function). The force from the grid of masses is calculated by solving Poisson's equation in Fourier space after carrying out a fast Fourier transform.

The generation of initial conditions relies on the fact that ΛCDM predicts Gaussian perturbations result from inflation. The perturbations can be evolved using linear theory until the start time of the simulation, typically $z \sim 100$. Particles are randomly placed in the simulation box, and the gravitational force is calculated. The direction of the force is reversed, and particles are moved until they reach a quasi-equilibrium state. The resulting uniform distribution of particles is known as a glass, and prevents there being any preferential directions (Baugh et al., 1995). The particles are displaced from their uniform configuration and assigned velocities based on the results of the linear theory approximation.

Most boxes employ periodic boundary conditions, as this simplifies the simulation while mimicking the large-scale homogeneity of the matter distribution in the universe. In order to run certain sections of a simulation at higher resolution zoom-in simulations can be used. In these simulations a region of interest is selected from a large volume low resolution simulation. The region is resimulated using particles with much lower mass than those in the large volume simulation. When running the zoom-in simulation the forces are calculated from the volume surrounding the zoom region using the original high mass particles. This means tidal forces due to external structure are included in the simulation of the region of interest.

Halo finding

To locate halos from a set of particle positions a halo finding algorithm is used. One of the most common finders is the Friend-of-Friends (FOF) algorithm (Davis et al., 1985). This allocates particles into the same group if the distance between

them is less than a threshold value, known as the linking length, b, which is defined as a fraction of the mean particle separation. By definition this means that each particle can be assigned to only one group. Different values of linking length will give different sizes of halo. If the linking length is set as $b = 0.2$ then the halos identified have a density of approximately 200 times the critical density.

One common issue with the FOF algorithm is that two distinct halos may be treated as a single object if there is a small chain of particles between them, known as a FOF bridge. A number of algorithms exist to locate gravitationally bound objects. This solves the previous problem, as well as allowing the complex substructure of halos to be identified.

SubFind (Springel et al., 2001) is one of the most common algorithms used to locate gravitationally bound structures within FOF groups. It first takes the list of particle positions from a FOF halo, and estimates the density at each of those positions by a kernel interpolation over the nearest neighbours. The particles are then processed in order of decreasing density. For each particle any neighbouring particles which have already been processed are identified. If a particle is isolated, then it starts growing a new subgroup around it. If all a particle's neighbours are members of a single subgroup then the particle is also added to that subgroup. If the particle's neighbours are members of two different subgroups, then a saddle point in the density contours has been identified, and the two subgroups are joined. In this process the subgroups form a nested hierarchy. Subgroups are then checked to ensure they are gravitationally bound. Any particles which are not bound to the subgroup are removed. If the subgroup still has at least N_{ngb} particles remaining then it is designated as a subhalo.

Another common choice is the Rockstar algorithm as introduced in Behroozi et al. (2013a). Initially it divides the simulation into groups by using the FOF algorithm with a large linking length. This allows the problem to be easily parallelised. From this point on the algorithm operates in a 6D phase space, where the distance between two particles is given by

$$d(p_1, p_2) = \left(\frac{|\vec{x}_1 - \vec{x}_2|^2}{\sigma_x^2} + \frac{|\vec{v}_1 - \vec{v}_2|^2}{\sigma_v^2} \right)^{1/2}, \tag{1.14}$$

where σ_x and σ_v are the particle position and velocity dispersions for the given group of particles being considered. A linking length is chosen such that 70% of particles will be linked with a least one other particle. Subgroups are then

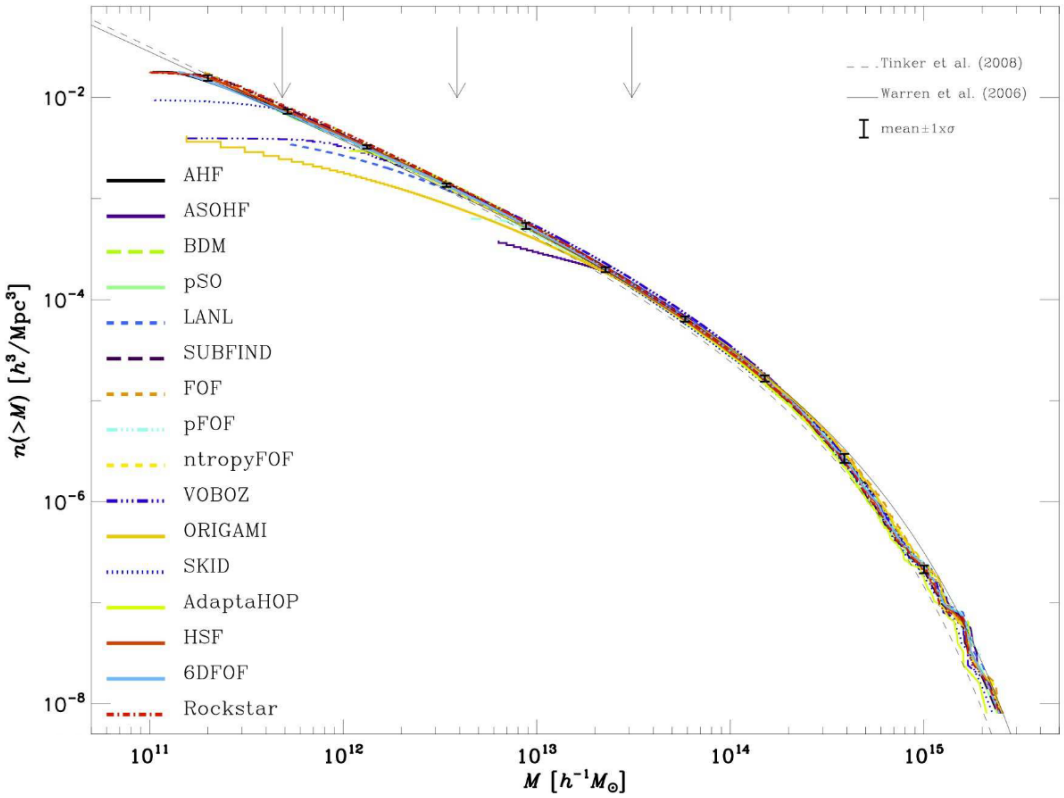

Figure 1.7 *Cumulative mass function of halos as identified by a number of different halo finding algorithms. All methods show good agreement for high mass halos, with the majority remaining in agreement for the full mass range. Taken from Knebe et al. (2011).*

identified as the groups of particles with a distance less than this linking length. Once subgroups have been found, the process is repeated on the subgroups themselves to identify substructure within. This remains until no subgroups are located with more than a minimum number of particles. An unbinding procedure similar to SubFind is then carried out to identify which subgroups should be classified as subhalos.

As different simulations apply different finders, it is important to ensure that they are consistent when trying to make comparisons between simulations. This is especially relevant for this thesis, as in some cases machine learning models are trained using halos from one finder, and then applied to halos from a different simulation identified using a different finder. I also compare the feature importance of models trained on different simulations, which also requires that the halos identified should be consistent. A comparison of different halo finding algorithms can be found in Knebe et al. (2011). They applied a number of different methods to the output of a single simulation, and compared the properties of the

halos located. As shown in Figure 1.7 they found that the mass function was consistent, as were a number of other properties which they considered.

Merger trees

It is useful to be able to track halos throughout a simulation. This allows studies to be carried out which see how halo properties (and the galaxies they host) evolve over time. As with halo finding a number of algorithms have been developed in order to track halos, and different simulations employ different methods. When two halos merge over time to form a single halo, the two merging halos are referred to as the progenitors of the combined halo, and the combined halo is referred to as the descendant of the merging halos. Since halos do not split up over time a given halo can have several progenitors, but will only have a single descendent. Here we summarise the merger tree finders used by the simulations used in this thesis. All of them build on the halo finders described in the previous section, in that halos are first identified for all snapshots output by the simulation, then halos are linked between these snapshots.

The LHaloTree algorithm (Springel et al., 2005) constructs mergers trees based on subhalos identified by SubFind. For a given halo, all halos in the subsequent snapshot are located which contain some of its particles. The particles are counted in a weighted fashion, with the weighting being given by the binding energy of the particles in the halo under consideration. This makes it easier to track halos which have fallen into a larger halo, and whose outer particles are being stripped. To allow for the possibility that halos may temporarily disappear for one snapshot, the descendant-finding process for snapshot S_n is carried out on snapshots S_{n+1} and S_{n+2}. If either there is a descendant found in S_{n+2} but not found in snapshot S_{n+1}, or if the descendant in S_{n+1} has several direct progenitors and the descendant in S_{n+2} has only one, then a link is made that skips the intervening snapshot.

The SubLink algorithm (Rodriguez-Gomez et al., 2015) also uses SubFind catalogs to generate merger trees. The method is similar to LHaloTree, but the score of the candidate descendent halos is given by $\sum_i R_i^{-1}$, where R_i is the binding energy rank of the particles from the halo under consideration. The other modification to the LHaloTree is that for each subhalo from snapshot S_n, a "skipped descendant" is identified at S_{n+2}, which is then compared to the "descendant of the descendant" at the same snapshot. If the two possible descendants at S_{n+2} are not the same object, the one obtained by skipping a

snapshot is kept. This processes is designed to allow for better tracking of flyby halos, which are passing through a larger structure but not becoming bound.

Consistent Trees (Behroozi et al., 2013b) is a more complex method than the particle-based algorithms described above. It enforces more strict conditions on the descendants being chosen in order to improve consistency of halo properties across timesteps. For a given halo at snapshot S_n, the method initially identifies the halo descendants that exist at S_{n+1} based on particle IDs. It then predicts what the position and the velocities of the descendants would have been at snapshot S_n. From this information it cuts connections between spurious descendants. It also creates new halos at snapshot S_n based on the predicted positions and velocities, although they are removed if no real progenitors are found for several timesteps.

1.2.2 Full physics cosmological simulations

Simulating baryonic matter is significantly more complicated than running dark matter only simulations. The majority of the baryonic matter in the universe is gas, so hydrodynamical simulations must be used to model fluid effects. Initial conditions are set by adding gas which is composed of hydrogen and helium. For a recent review of the use of hydrodynamical simulations within astrophysics see Vogelsberger et al. (2020).

Prominent examples of cosmological simulations include Illustris (Vogelsberger et al., 2014a,b; Genel et al., 2014; Sijacki et al., 2015), IllustrisTNG (Springel et al., 2018; Pillepich et al., 2018b; Naiman et al., 2018; Nelson et al., 2018; Marinacci et al., 2018), Simba (Davé et al., 2019), EAGLE (Schaye et al., 2015), HorizonAGN (Dubois et al., 2014), and FiBY (Johnson et al., 2013). A selection of images from simulations is shown in Figure 1.8.

Modelling hydrodynamics

As gas is made up of an uncountable number of particles, it's impossible to calculate the motion of each particle individually. Instead fluid elements are used, which represent an ensemble of physical particles, and track the statistical properties of the gas. In the Eulerian simulation approach the volume to be simulated in split into cells, with each cell representing a single fluid element,

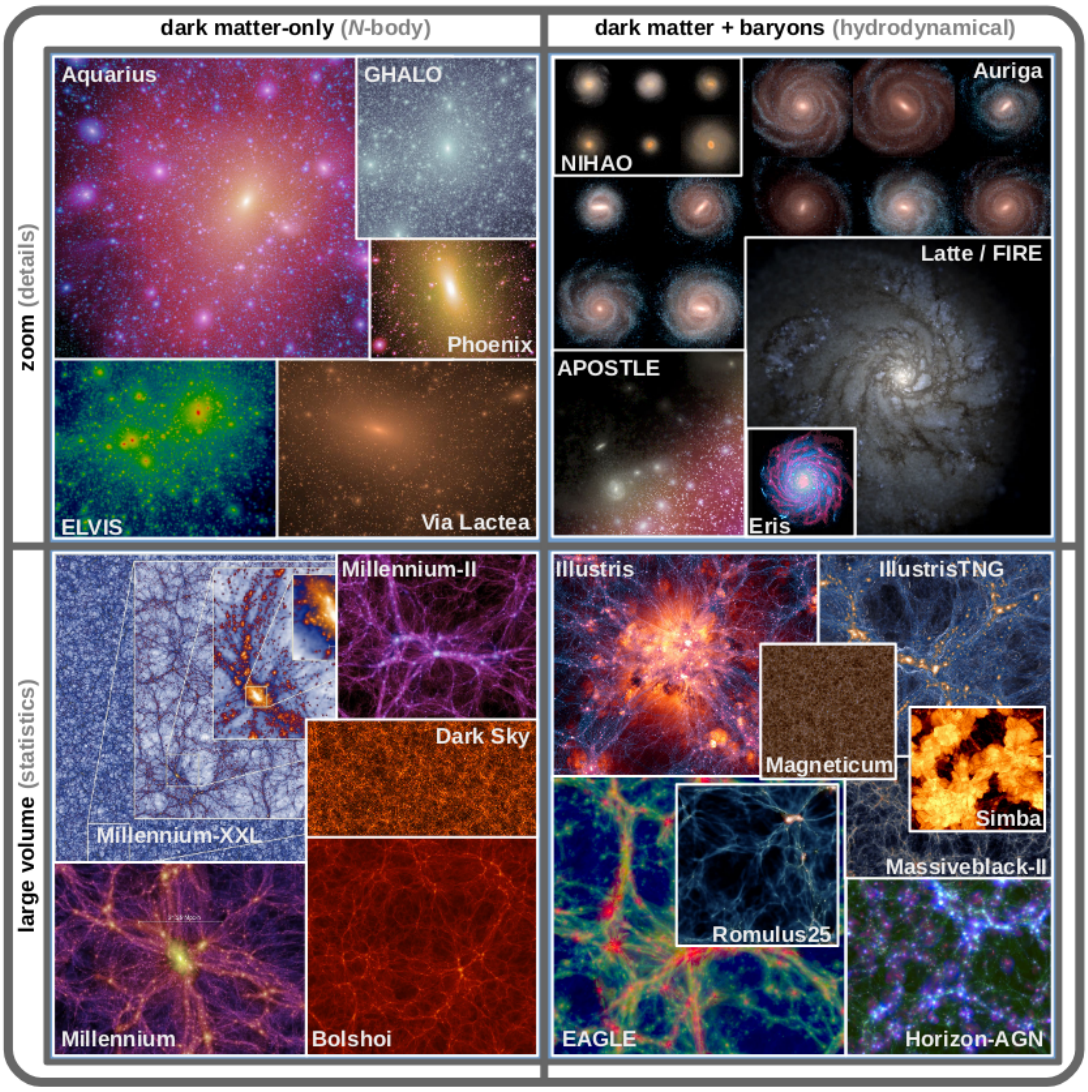

Figure 1.8 *Visual representations of some selected recent structure and galaxy formation simulations. Taken from Vogelsberger et al. (2020).*

and fluid flow is tracked through the faces of each cell. Resolution is determined by cell size. As the simulation evolves, certain regions will end up with a higher density of gas. As these regions are often areas of interest (such as sites of star formation), modern codes often make use of an approach called adaptive mesh refinement (AMR). Resolution is dynamically increased by decreasing the cell size in the sensitive region, effectively overlaying the original coarse grid with a finer one. One downside of this method is that fluids can only flow in the directions normal to the cell faces, which introduces edge effects for spherical flows. Examples of AMR simulation codes include ENZO (Bryan et al., 2014) and RAMSES (Teyssier, 2002).

Where Eulerian methods track the flow across cell boundaries, Lagrangian

methods follow fluid parcels that move with the gas flow. In this approach the fluid is discretised by mass rather than by space. For cosmological simulations smoothed particle hydrodynamics (SPH) is used (Gingold & Monaghan, 1977). The density and pressure at an arbitrary point $\mathbf{r}$ are obtained by summing contributions from the N_k nearest particles to that point.

$$\rho(\mathbf{r}) = \sum_{i=1}^{N_k} m_i W(\mathbf{r} - \mathbf{r}_j, h) \tag{1.15}$$

$$P(\mathbf{r}) = \sum_{i=1}^{N_k} m_i \frac{P_i}{\rho_i} W(\mathbf{r} - \mathbf{r}_j, h) \tag{1.16}$$

Here $W(\mathbf{r}, h)$ is a smoothing kernel with scale length h chosen such that N_k particles are contained within the region. The motion of a particle can then be determined by taking the gradient of the pressure. This method provides natural resolution adaptation, as areas with high densities have a large number of particles by definition. However, it has difficulties in capturing sharp discontinuities, such as shocks. Simulation codes which employ SPH include GADGET (Springel, 2005; Springel et al., 2021) and SWIFT (Schaller et al., 2023).

Lagrangian-Eulerian methods attempt to combine the strengths of the Lagrangian and Eulerian methods. This is done by using a deformable mesh to define the cells across whose faces the fluid flows. The mesh is recomputed as particles move. An example code which uses this method is AREPO (Springel, 2010), which uses a Voronoi tessellation to divide the space around particles into cells.

The need for subgrid models

Many of the relevant physical processes that need to be modelled occur below the typical resolution limits of cosmological simulations. Therefore various ”sub-grid physics” prescriptions are employed to model them. An example of this is star formation. Since the typical mass of a star is far below the mass resolution of cosmological simulations, a star particle represents a population of stars rather than an individual star. Once certain conditions on the gas are met the subgrid model decides to create a star particle.

Subgrid models tend to be based on idealised versions of the physical process,

and usually have a number of tunable parameters. The value of these parameters is usually set based on what will allow the simulation to reproduce observations, rather than from some physical argument. They can also introduce degeneracies between different subgrid models that can be difficult to untangle. In the following subsections I detail a number of common subgrid prescriptions.

Star formation and feedback

Radiative cooling must be implemented in simulation codes in order for baryons to dissipate their energy. As metal line cooling can dominate for typical enriched warm-hot gas, it is important to tracking the enrichment of gas with heavy elements. Most codes now track a set of individual elements to more accurately model cooling rates (e.g. Oppenheimer & Davé, 2008; Wiersma et al., 2009). Post-reionization simulations also assume a uniform UV background to calculate the cooling rates (e.g. Haardt & Madau, 2012). In order to follow cooling in detail the resolution of the simulation must be high enough to distinguish gas phases.

Simulating cold gas directly is difficult because of the short numerical integration timescales required for solving the equations of motion in dense regions. However, observations indicate that the star formation efficiency in molecular gas is nearly universal, with approximately 1% of the gas being converted into stars per free fall time (Bigiel et al., 2011; Krumholz et al., 2012), and so this significantly simplifies the modelling. Simulations require that the density of a gas element exceeds some critical value before it can form stars. The choice of density threshold used varies between simulations, and is an example of a tunable parameter within subgrid models. Some simulations require additional criteria to be fulfilled such as converging flows (Stinson et al., 2006), or that the gas is self-shielding (Hopkins et al., 2018).

Gas particles which satisfy the criteria for star formation are assigned a SFR based on a Schmidt (1959) law,

$$\frac{dM_*}{dt} = \epsilon \frac{M_g}{t_{ff}} \tag{1.17}$$

The conversion efficiency parameter ϵ is calibrated to matched the observed relation from Kennicutt (1998). To prevent an overabundance of low mass star particles being generated, the gas is converted into star particles using stochastic sampling, based on the calculated SFR.

The interaction of stars with their immediate environment is incredibly important within cosmological simulations as without it the properties of gas are drastically different (e.g. Vogelsberger et al., 2013) and many more stars would form than current observations show. As with star formation, the scales over which these processes occur is below the resolution limit of simulations. One key contribution of stars is increasing the metallicity of gas, as all metals are produced within stars. Star particles enrich nearby gas elements based on metal yield models derived from stellar evolution calculations.

As discussed in Section 1.1.2, stars, especially massive ones, inject large amounts of energy and momentum into the ISM. A wide range of subgrid models exist to implement this, but the main divide is based on whether they deposit energy thermally or kinetically. Thermal models directly increase the temperature of neighbouring gas particles. However, due to artificial overcooling, some additional mechanisms are required to prevent the energy being immediately radiated away. These include disabling radiative cooling temporarily in the heated cells (Stinson et al., 2006), or injecting energy stochastically so that particles receive such a large temperature boost that it significantly increases the cooling time (Dalla Vecchia & Schaye, 2012). Kinetic models induce outflows by injecting momentum into the ISM (Springel & Hernquist, 2003; Pillepich et al., 2018a). This is done through the use of hydrodynamically-decoupled wind particles. Outside of the dense ISM wind particles recouple, allowing them to deposit their mass, momentum, metals, and thermal energy content. These models are parameterized by a wind velocity, v_{wind}, and a mass loading factor $\eta = \dot{M}_{out} \ / \ \dot{M}_{*}$, where M_{out} is the wind mass outflow. Recently a number of models for stellar feedback have been developed which include other feedback channels, such as stellar winds and radiation pressure from massive stars (e.g. Agertz et al., 2013).

Black hole growth and feedback

Since the processes from SMBHs originate are still poorly understood they are impossible to model explicitly in cosmological simulations. The standard approach is to place a black hole in halos above a certain mass (typically $\sim 10^{10} M_{\odot}$). This is known as seeding. In most cases a fixed mass seed is used (e.g. Schaye et al., 2015; Weinberger et al., 2017), but for others the mass depends on the gas properties (Tremmel et al., 2017).

Black holes can grow by two processes. One is by merging with other black holes.

This occurs as black holes experience dynamical friction which will cause them to move towards the centre of their host galaxy. If there are two black holes at the centre of a galaxy then dynamical interactions will bring them closer together, and general relativistic effects will cause the final merger. Simulations tend to merge black holes instantly once they are close enough to each other. The other process by which they grow is by accretion of gas. This is typically done based on Eddington-limited Bondi-Hoyle accretion (e.g. Schaye et al., 2015; Weinberger et al., 2017), but the accretion rate is sometimes artificially increased in order to better match observations (e.g. Booth & Schaye, 2009). The other option is a torque driven model (Hopkins & Quataert, 2011; Davé et al., 2019), as the Bondi model assumes that the accreting gas has negligible angular momentum.

The energy and momentum that results from the accretion of a SMBH can couple to the gas in the galaxy. This is important as it provides a limit on the black hole growth. Black holes with a high accretion rate will lower the density of the surrounding gas, in turn causing the accretion rate to drop. Two modes of feedback are commonly implemented within cosmological simulations. In quasar mode the feedback affects the surrounding gas with either energy or momentum being directly injected, similar to supernova feedback. This mode is associated with radiatively efficient accretion. Jet mode feedback was introduced to reproduce observation of jets extending out of galaxies (Blandford et al., 2019). It is triggered when the accretion rate drops below a certain critical value. The impact of jet feedback on galaxy evolution is still unclear. This highlights one of the major challenges with cosmological simulations, in that the jets can extend for tens of kpc, but the scales of black hole accretion are on the order of ~AU.

1.2.3 Semi-Analytic Models

Running full hydrodynamical simulations is very computationally expensive compared with running dark matter N-body simulations. SAMs are a cheaper alternative as they combine dark matter only simulations with analytic approximations for baryons. They work by assuming that at the moment a halo forms it will contain a baryon fraction equal to the cosmic mean. Furthermore, the baryons will have approximately the same spatial distribution and similar angular momentum as the dark matter. The evolution of the baryons from this point is dependent on the physical processes discussed in the previous sections. These are modelled by using a set of coupled differential equations, where the parameters

and forms of these equations can be based on approximations of the physics that is occurring, or else tuned to reproduce observations. The effect of halo mergers on baryons is also accounted for in these formulations. Thus semi-analytic modelling allows for mock galaxy catalogs to be created that span large volumes, which can then be compared with large observational surveys. A comparison of different SAMs can be found in Knebe et al. (2015).

1.3 Machine Learning

Machine learning is a field of computer science, statistics, and optimization theory that covers a wide range of problems. As such an all-encompassing definition is difficult. However, in general it can be thought of as a set of algorithms that can automatically detect patterns in data. These algorithms can then provide insight into information contained within the data set, or they can use patterns that were identified to produce a model that can make predictions on future data. Commonly used resources for an introduction to the field of machine learning include T. J. Hastie & Friedman (2005), Bishop (2006), Murphy (2012), Sarah Guido (2016), Mehta et al. (2019).

Machine learning algorithms come in two main categories: supervised and unsupervised. Supervised learning algorithms are trained using data that is labelled. This means that each vector of input features in the training set has a corresponding label. The label can be a scalar or a vector. The model learns relationships between the input features and the label, and from this it is possible to make predictions of the labels of unseen data. In general the training data consists of (X, y), where X is the set of input features, and y are the labels to be predicted. The model then learns an approximate mapping of $X \rightarrow y$. Supervised learning itself can be split into a further two classes. In regression problems the output label can take on a continuous range of values. An example would be predicting the stellar mass of a galaxy. In classification problems each data point in the training data is a member of a certain class. The output vector y consists of N binary features, where N is the number of classes. The algorithm then predicts which class new data points will be a member of. An example would be classifying an image of a galaxy as containing a bar or not. Common examples of supervised learning algorithms include linear models, support vector machines, and decision trees. Most supervised learning algorithms can be adapted for either classification or regression problems.

In order to assess how well a supervised model performs it is necessary to have a test data set. This data set must not have been used to train the model as otherwise the model could simply memorise labels of the training data and perfectly predict the test data points. The test data set must also be large enough to be representative of the data as a whole. Therefore before training of a model starts, the data set is split into a training and a test set. The ratio of the split depends on the problem being solved. Depending on the algorithm being used, sometimes the train data must be further split into two sets. A validation set is needed whenever hyperparameters are being tuned. A hyperparameter is a value that is set before the model is trained, as opposed to the parameters of a model which are the values learned during training. As an example a random forest classifier is made up of N decisions trees. The value N is a hyperparameter. In summary, the model parameters are learned using the training set, the hyperparameters are selected based on the performance of the model on the validation data, and the test data gives a score that is representative of how the model will work for unseen data. The metric used to evaluate the performance of the model is dependent on the problem to be solved.

The second family of machine learning algorithms are unsupervised. In unsupervised learning, the data is fed into the algorithm, and the algorithm extracts information from it. Similar to how supervised learning is split in two, there are two main cases of unsupervised learning. Clustering algorithms separate the data into distinct groups. An example would be feeding in positions of galaxies and having them grouped in order to find galaxy clusters. The other major class of unsupervised learning algorithms are used to create a new representation of the data. These are often referred to as unsupervised transformations. The most common use case is dimensionality reduction. Some examples of prevailing unsupervised learning algorithms are t-SNE, DBSCAN, and principal component analysis.

Deep learning is a branch of machine learning. It consists of using neural networks with many layers, and has the advantage that it can extract features directly from raw data, unlike many classical machine learning techniques. For example when using decision trees to classify images, features must first be extracted, but convolutional neural networks can be fed the entire image. Research into it and its uses have increased drastically in the past ten years, in part due to the fact that models can be trained and deployed using GPUs. Most recent breakthroughs in machine learning have come in deep learning, often allowing super-human

performance levels (e.g. Silver D., 2017). Deep neural networks come in many different varieties, and can be used for both supervised and unsupervised tasks. One drawback however is that the typical number of parameters within a neural network makes them so complex that reasons for the decisions they make, and the relationships they learn cannot be extracted in a human-interpretable way. Given how highly non-linear large neural networks are it is often impossible to even determine which input features are providing the most information. This is now an active research area within machine learning (Erhan et al., 2009).

1.3.1 Machine learning in astronomy

The initial utilization of machine learning techniques within astronomy was applying them to observation problems. One of the earliest applications involved the use of neural networks to categorize objects from photometric catalogs as either stars or galaxies (Odewahn et al., 1992). Machine learning techniques were then used to classify stellar spectra (von Hippel et al., 1994) and to determine the photometric redshift of galaxies (Collister & Lahav, 2004). In more recent times, since the field of machine learning has took off, machine learning has been used for a wide variety of applications within astronomy. For recent reviews see Fluke & Jacobs (2020); Baron (2019); Djorgovski et al. (2022); Smith & Geach (2023).

There have been numerous applications of machine learning relating to cosmological simulations. The first main category of uses is to train a model on simulations and then apply it to observational data. This can be useful for inferring properties which are easy to measure in simulations but are difficult to observe, such as estimating the mass of objects (Villanueva-Domingo et al., 2021b; Carlesi et al., 2022), and determining which galaxies have recently undergone mergers (Ferreira et al., 2022). The other category is simulation-only applications. Examples include producing super-resolution N-body simulations (Schaurecker et al., 2021; Ni et al., 2021; Li et al., 2021), emulating power spectra (Agarwal et al., 2014; Jennings et al., 2019), and generating merger trees (Robles et al., 2022). Further uses of machine learning which are specifically relevant to this thesis are discussed in detail in the introduction sections of Chapters 2 and 3. For a list of publications applying machine learning to cosmology see `https://github.com/georgestein/ml-in-cosmology`.

1.4 Thesis outline

The subsequent three chapters constitute the scientific contents of this thesis. Chapters 2 and 3 are based upon published journal articles.

In Chapter 2 I develop a new method for predicting the baryonic properties of dark matter only subhalos. I show how this method offers an improvement in accuracy compared with previous approaches. I examine the trained machine learning models to learn about nature vs nurture. I apply the method to generate a mock catalog, which is then compared with quasar observations.

In Chapter 3 I extend the method by passing baryonic properties as input features. This allows me to investigate the evolution of galaxy properties in different simulations, and in various environments within a single simulation. Using the CAMELS simulation suite I consider the impact of cosmological and astrophysical parameters on the buildup of stellar mass.

In Chapter 4 I apply a combination of neural networks and symbolic regression methods to construct a semi-analytic model which reproduces the galaxy population from a cosmological simulation. The neural network based approach is capable of producing a more accurate population than a previous method of binning based on halo mass. The equations resulting from symbolic regression are found to be a good approximation of the neural network.

In Chapter 5 I present a summary of my conclusions and discuss potential future extensions to this work.

Chapter 2

Generating mock galaxy catalogs

The material in this chapter was originally published in McGibbon & Khochfar (2022) and Natarajan et al. (2023).

2.1 Introduction

The ability to compare the output of N-body simulations with the observed large scale structure of the universe allows for determination of cosmological parameters and provides insight into the mechanisms of galaxy formation within halos (e.g. Somerville & Davé, 2015). However, large N-body simulations cannot be directly compared with observations of the universe as we only observe luminous baryonic matter. There are several common methods used to combine baryonic physics with N-body simulations.

The ideal approach is to run a full hydrodynamical simulation, which includes fluid elements alongside the dark matter particles as discussed in Section 1.2.2. However, running these types of simulations is incredibly computationally expensive to run. Some of the upcoming surveys mentioned in Section 1.1.1 such as Euclid and LSST will cover $\sim$ Gpc^3 volumes, and require similarly sized mock galaxy catalogs to make comparisons. It's still not possible to run hydro simulations of this size at a reasonable resolution, so other methods are needed.

Another approach is to take the halo catalogs resulting from an N-body simulation and "paint on" galaxies. The simplest way to do this is via subhalo abundance

matching (e.g. Vale & Ostriker, 2004; Moster et al., 2010; Grylls et al., 2020; Neistein et al., 2011). It assumes that each halo hosts one central galaxy, each subhalo hosts one satellite galaxy, and that the highest mass halo hosts the most massive galaxy, the second highest mass halo hosts the second highest mass galaxy, and so on. The galaxy stellar masses are set such that the stellar mass function is recovered. Another method is to use the halo occupation distribution (HOD) approach (e.g. Berlind & Weinberg, 2002; Hadzhiyska et al., 2020). Here the number of galaxies within a halo are usually determined by an empirical formula which takes the halo mass as its input variable. Recent HOD models also account for secondary halo properties, such as halo concentration or environment (e.g. Paranjape et al., 2015; Hadzhiyska et al., 2021)). A more sophisticated technique is to use semi-analytic models, as discussed in Section 1.2.3

A recently developed procedure for generating galaxy catalogs is to make use of machine learning. The first way to utilise machine learning algorithms is to predict the number of galaxies within a friends-of-friends halo. This method is similar to the HOD method, except a machine learning model is trained to predict the number of galaxies rather than by fitting a formula. Xu et al. (2013) were among the first to try this approach. They used support vector machines and k-nearest-neighbour regression algorithms. A series of papers (Zhang et al., 2019; Yip et al., 2019) has used convolutional neural networks that take density fields as input to predict the number of galaxies. Recently Delgado et al. (2022) used a combination of random forests and symbolic regression to examine the galaxy-halo connection in IllustrisTNG. Xu et al. (2021) also used random forests and examined their feature importance.

The other method of using machine learning is to learn the relationship between the baryonic properties themselves and the dark matter properties of the host halo, an approach first considered by Kamdar et al. (2016a,b). In one work they used data from the Illustris hydrodynamic simulation to train their models, and in another they trained on the Munich SAM. They used various classical machine learning algorithms and found that the extremely randomized tree (ERT) algorithm performed best. Agarwal et al. (2018) also investigated a range of algorithms by training on the MUFASA simulation and found that the ERT was the best. They included information about the local environment around the halos as input to the models and found that this improved predictions. Jo & Kim (2019) used the number of mergers a halo had undergone as an input feature. They applied their model to a large N-body simulation and compared the

resulting galaxy catalog with one generated by SAMs applied to the same N-body simulation. They found disagreement between the machine learning approach and the SAM galaxy population, but this was to be expected as the parameters of the SAM were not tuned to the hydrodynamical simulation they trained on. Machado Poletti Valle et al. (2021) used similar techniques to predict the shape of gas within halos, and Eide et al. (2020) used machine learning to predict the black hole mass of high redshift halos, but included baryonic properties as input features. An alternative machine learning technique was used in Moster et al. (2021). Rather than training directly on hydrodynamical simulations, they used reinforcement learning to train a neural network. Training in this way means there does not need to be a direct mapping between halos and galaxies, the network is only tasked to reproduce mass functions. This means their model can be trained on observations. They found that the halo growth rate was an important feature for making predictions. The most recent work includes Kasmanoff et al. (2020) who used convolutional neural networks that take density maps as input for prediction of stellar mass, Moews et al. (2021) who use an equilibrium model as input to the machine learning models to help improve predictions, and Lovell et al. (2022) who train their model using zoom-in simulations alongside a larger periodic box. In Icaza-Lizaola et al. (2021) and Icaza-Lizaola et al. (2023) a sparse regression model was used to model the relation between galaxy stellar masses and their host halos.

In this chapter I use machine learning algorithms to predict the baryonic properties of dark matter subhalos. I train the model on a state-of-the-art hydrodynamical simulation. Rather than using halo properties from redshift zero combined with summary features for the halo's history such as number of mergers or formation time, I directly use the evolutionary information of halo properties over a wide range of redshifts. This model could be used to create galaxy catalogs from any N-body simulation that has merger trees. I show how including the full growth history of the halo significantly improves the performance of the machine learning models. I examine the features that are selected as important and show how they can be used to gain insight into galaxy formation mechanisms.

This approach of probing feature importance over time can disentangle the physical drivers of galaxy properties and inform observational survey strategies. It allows me to examine whether correlations that have been observed between galaxies and their host halos are indeed because they are directly linked, or if the correlation is simply the result of a deeper connection at a higher redshift. As my

model includes information on both a galaxies initial conditions and its evolution, it is ideal for providing insight into the "nature vs nurture" debate (e.g. De Lucia et al., 2012, 2019; Winkel et al., 2021), and thus I address the question within simulations.

2.2 A novel machine learning method for generating large volume mock galaxy catalogs

In this section I provide a summary of the hydrodynamical simulation used to train my models and also an overview of the machine learning algorithms I used. I discuss how the data from the simulation must be transformed before it is possible to pass it to the models. I then compare the accuracy of the different models when predicting galaxy properties.

2.2.1 Training data

IllustrisTNG (Springel et al., 2018; Pillepich et al., 2018b; Naiman et al., 2018; Nelson et al., 2018; Marinacci et al., 2018) is a suite of hydrodynamical cosmological simulations run with the moving mesh code AREPO (Springel, 2010). Each simulation includes all significant physical processes to track the evolution of dark matter, cosmic gas, luminous stars, and supermassive blackholes from a starting redshift of $z = 127$ to the present day $z = 0$. All the simulations are run with a flat cosmology consistent with Planck Collaboration et al. (2016): $\Omega_{\mathrm{m},0} = 0.3089, \Omega_{\Lambda,0} = 0.6911, \Omega_{\mathrm{b},0} = 0.0486, \sigma_8 = 0.8159, n_{\mathrm{s}} = 0.9667$, and h =0.6774. For further details regarding the specific implementation of the IllustisTNG simulations, including all relevant subgrid models, I refer the reader to Chapter 3 of this thesis.

For this work I use the TNG100 simulation which has a simulation volume of $(75\ h^{-1}\mathrm{Mpc})^3$. The TNG100 was run from the same initial condition for three resolutions. For this work I use the highest resolution run available, named TNG100-1. This run has 1820^3 dark matter particles with $m_{\mathrm{DM}} = 7.5{\times}10^6 M_\odot$ and 1820^3 hydrodynamic cells with $m_{gas} = 1.4 \times 10^6 M_\odot$ at $z = 127$. Halos are found first with the FOF algorithm (Davis et al., 1985), then subhalos are identified using the SubFind subhalo finder (Springel et al., 2001). Two sets of mergers

trees are available. For this work I use those generated by the LHaloTree algorithm (Springel et al., 2005). The outputs of the simulation are saved in 100 snapshots.

2.2.2 Data extraction

In order to ensure that the subhalos I consider are well resolved, I require that they have a total mass (dark matter plus baryons) that is above 10^9 $M_\odot$ at $z = 0$. This mass matches with the minimum halo mass resolved in most ~Gpc N-body simulations. I disregard subhalos whose stellar or gas mass is zero, as these are not my targets of interest. This leaves a total of 350,000 objects, roughly 8% of the initial subhalo catalogue.

In order to check the performance of my model I split the data into a train set and a test set. For this work I assume a train volume which is 70% of the simulation volume. The effect of varying the size of the training set is examined at a later point. I randomly place a subbox that has a volume of 70% of the full volume within the full box. All halos within the box are included in my training data, all halos outside are the test data. This means that the number of halos in the training set varies depending on where the subbox is placed. When determining the hyperparameters for the models I further split the training set into a training and validation set. This is done randomly, so the validation set does not correspond to a contiguous volume.

Baseline model

I adopt a similar baseline model to Jo & Kim (2019). The input features to my baseline model are the following halo properties at $z = 0$: dark matter mass (the total mass of dark matter particles bound to halo, multiplied by a factor of 6/5 to account for the fact that N-body simulations do not contain any baryonic mass), velocity dispersion, maximum of spherically-averaged circular velocity, and magnitude of the spin vector. I use the ERT algorithm for my baseline model.

I do not consider any environmental properties of the halo as input features. For the full model the total number of input features is given by $n_{\rm snap}n_{\rm prop}$, where $n_{\rm snap}$ is the total number of snapshots which I use as input for the model (I settle on $n_{\rm snap} = 10$), and $n_{\rm prop}$ is the number of halo properties for a single snapshot

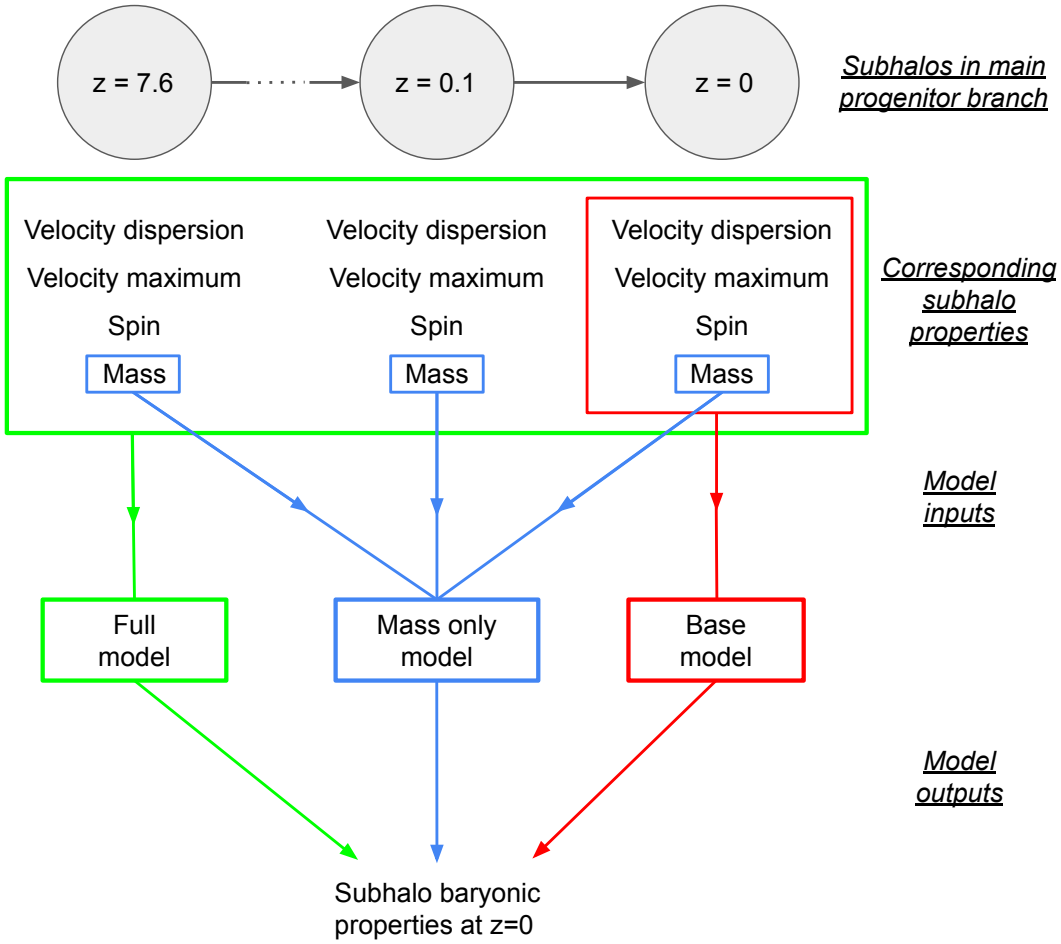

Figure 2.1 *Summary of the inputs to each of the three models discussed in this work. The input features for the base model are four dark matter subhalo properties (mass, velocity dispersion, maximum circular velocity, spin) at redshift zero. The mass only model takes in the dark matter mass of the subhalo over a range of snapshots. The full model takes in the four input features of the base model, but from a range of snapshots, not just redshift zero. The output for all models is the subhalo's baryonic properties at redshift zero. The ERT algorithm is used for all models.*

(for this work $n_{\text{prop}} = 4$). Therefore, increasing n_{prop} by including environmental properties would lead to the total number of input features in the full model being significantly larger, making the feature importance plots harder to interpret. I acknowledge that including them would improve the performance of the baseline model, but stress that their inclusion would increase the performance of the full model by a similar amount. As the purpose of this work is to demonstrate the value of taking in the full halo history, this decision is justified.

A major advantage of using decision tree based machine learning models is that they are invariant to the scaling of the input features. Therefore I do not scale the input features in any way, despite the fact that they values span multiple orders of magnitude.

Time series history

For each valid subhalo at $z = 0$ I track its properties back in time using its main progenitor at each snapshot. At each snapshot to be used as input for the model I store the same four properties that I use for my baseline model. All these features are passed as input features to my full model which predicts the subhalo's $z = 0$ baryonic properties. I also have a model which only takes in the subhalo mass at the different snapshots being considered. I refer to this as the mass only model. Figure 2.1 shows a summary of each of the three models.

To allow for subhalos that temporarily disappear, the LHaloTree algorithm may link a subhalo identified at snapshot n with one at snapshot $n-2$ if no progenitor can be found at snapshot $n-1$. Therefore a subhalo may be missing properties at a point in its merger tree. Whenever a subhalo is identified at snapshot $n-2$ and snapshot n, I set the subhalo properties at snapshot $n-1$ as equal to the values at snapshot n.

The TNG100 simulation has halo properties stored for 100 snapshots, with snapshot 99 corresponding to $z = 0$. Figure 2.2 shows the fraction of subhalos that can be tracked to at least the snapshot shown. Higher mass halos are easier to track further back. I therefore decide the lowest snapshot to consider is 9, which corresponds to $z = 7.6$. If a halo cannot be tracked back to a certain redshift the value of its input features for that snapshot are set to zero. In that case the performance of the model is worse than for the halos I can track back, but it is still better than my baseline model. As can be seen from Figure 2.2, a significant

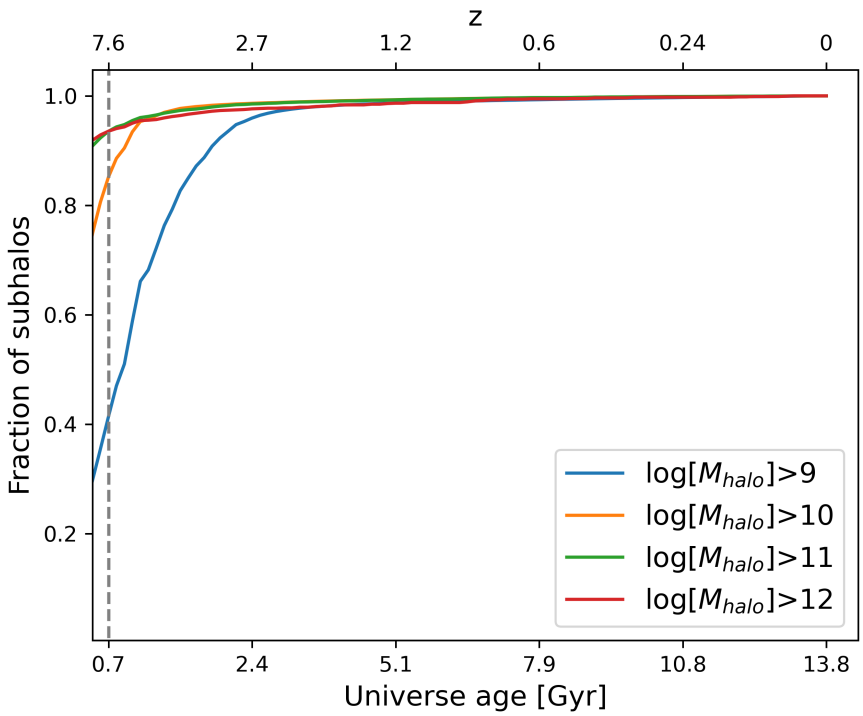

Figure 2.2 *Fraction of subhalos whose merger trees extend to a higher redshift than the value on the horizontal axis. The vertical dashed line shows the highest redshift halo properties I use in this work.*

fraction of subhalos can be tracked back to $z = 7.6$, so there is still a benefit to using the halo properties at this redshift as input to my models.

Output features

Although the model can be used to predict any baryonic property of a subhalo, in this work I focus on 8 properties: the gas mass, the total black hole mass, the stellar mass, the mass weighted metallicity of the star particles, the stellar half mass radius, the sum of the star formation rate of all gas cells, and the U and K band magnitudes. Following (Jo & Kim, 2019), I log all values except for the magnitudes. If I do not take the logarithm, during the training phase the data points from high mass halos are weighted much more highly than those from low mass halos. I scale each output feature using a MinMax scaler (subtract minimum value, divide by maximum - minimum value) to transform all values to be between 0 and 1. I use the mean squared error to evaluate the performance of my models.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

Here n is the number of data points, y_i is the true value of the output feature, and $\hat{y}_i$ is the value predicted by my models. Due to the normalization of the output features the value of the mean squared error does not have a physical significance. However this rescaling allows me to compare how difficult each output feature is to predict.

2.2.3 Machine Learning Methods

I exploit the results of fully hydrodynamic, high-resolution simulations to create a mapping between halo and galaxy properties. This type of problem is an example of supervised learning as I have a set of input data (dark matter only properties of a subhalo) and corresponding output data (baryonic properties of the galaxy hosted by that subhalo). Below I give an overview of the supervised machine learning algorithm that I use.

Extremely randomised tree ensembles

Decision trees are common supervised machine learning approach. A decision tree is made up of a number of nodes, each of which contains the value of an input feature on which is used to determine how to progress through the tree. The data point for which we want a prediction is begins at the top of the tree. If the value of the input feature for the data point is greater than the value of the node, we move down the right side of the tree. If the value is less we move down the left side. This process of moving down the tree based on the value of the input features continue until a leaf node at the bottom of the tree is reached. Each leaf node has a prediction value associated with it, so the prediction is given by the value of the node the data point ends up in.

An example of a decision tree used to predict the stellar mass of a halo is shown in Figure 2.3. Two different visualisations are shown, but both represent the same tree. This decision tree was trained on some simple mock data for visual clarity, it was not taken from the TNG simulation. Each node in the top visualisation

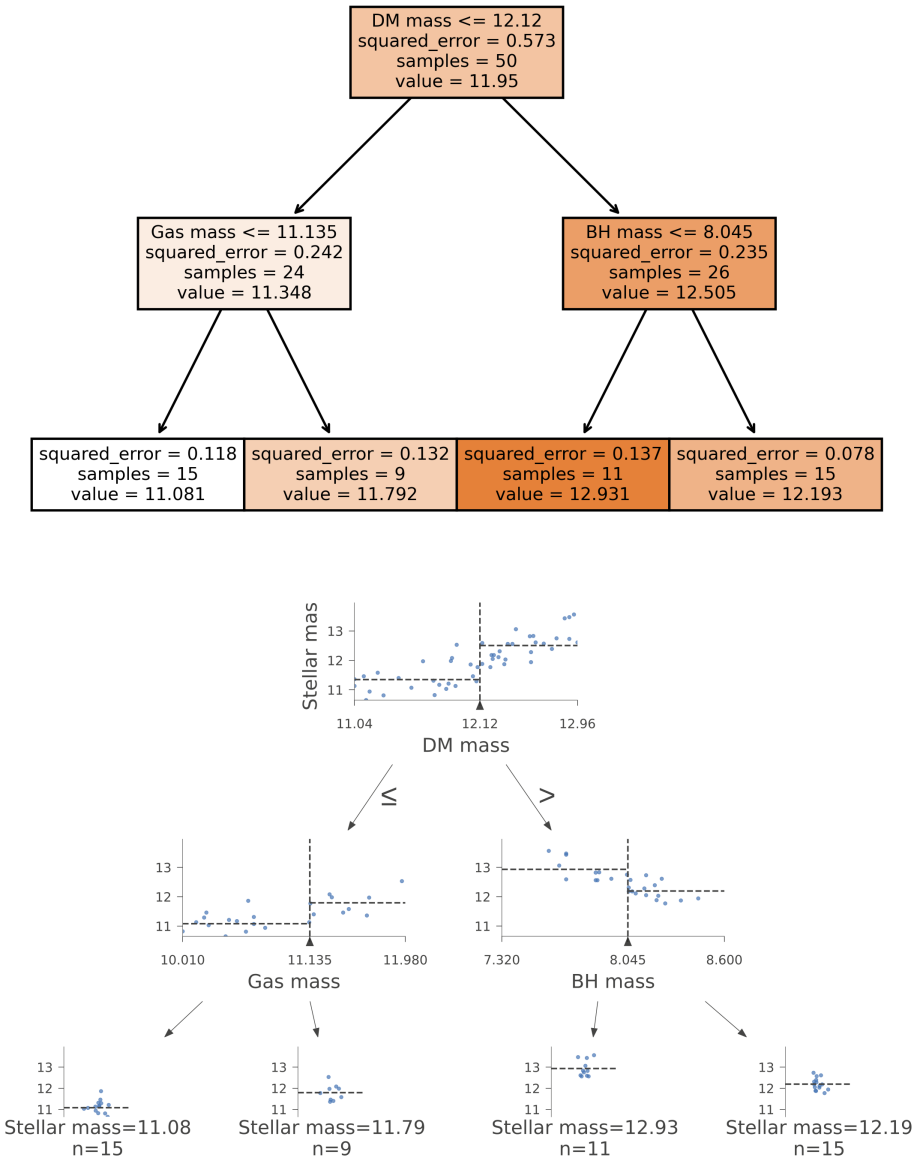

Figure 2.3 *Two visualisations of the same decision tree, which predicts stellar mass using DM mass, BH mass, and gas mass*

lists 4 pieces of information: the property and value used for the split, the mean squared error of all the data points in that node, the number of samples in the node, and the mean value of the stellar mass in that node. Imagine we wish to predict the stellar mass of a halo with $\log(M_{DM}) = 12$ and $\log(M_{Gas}) = 11.2$. The node at the top of the tree splits based on the dark matter mass, with a split value of 12.12. As the value of our halo is below this we move down the left side of the tree, coming to the node which splits based on the gas mass. As our halo has a higher value than 11.135 we move to the right at this split, ending in the leaf node with a stellar mass value of 11.79. Each node within the bottom visualisation plots the stellar mass against the feature used for the split. The vertical line shows the split value, with the horizontal dashed lines indicating the mean value for each of the child nodes.

We now know how to make a prediction when given a decision tree, but how is a decision tree generated? Each decision tree is constructed top-down from the root node. At each node a large number of splits of the training data is tried by varying the input feature used for the split, and the split value. The optimal split is chosen by minimizing the weighted average of the variance of the two bins (Breiman et al., 1984). This is done since calculating the variance is equivalent to calculating the MSE when the mean is used as the prediction. This partitioning of the data results in each leaf node at the bottom of the tree containing a small subset of the data, where almost all members of the subset have a similar output value. Predictions from decision trees are based on the assumption that test data points will have a similar output value to the other members of the leaf node it is placed into.

A random forest is made up of a number of decision trees (Tin Kam Ho, 1995; Breiman, 2001), known as an ensemble. There is a bootstrapping procedure such that each decision tree within the forest is trained on a randomly generated subset of the training data. Further randomness is added in that for each split only a subset of input features can be used. The prediction from a random forest is the average prediction of its component decision trees. A major advantage of random forests is that they are significantly less prone to overfitting data compared with a single decision tree. This results from the randomness added when training the individual decision trees.

For this work I use extremely randomised tree ensembles (Geurts, 2006). This is the algorithm used in previous work (Kamdar et al., 2016b; Agarwal et al., 2018; Jo & Kim, 2019), and I found it to slightly outperform the standard random

forest. It adds in additional randomization by computing a random split for each feature at each node, rather than the optimal split.

Feature importance

One major benefit to using ensembles based on decision trees is the ability to extract information on which input features are providing the information that is used to make the final predictions.

Consider a decision tree which contains n nodes, where the i_{th} node is given by N_i. Each node, except for the leaf nodes, has a left and right child, denoted as N_i^l and N_i^r respectively. The weight of a node, $W(N_i)$, is defined as the fraction of total number of data points which pass through that node. This is 1 for the root node by default. The other necessary property is the mean squared error of the data points in the node, shown as $MSE(N_i)$. The importance of a single node is given by

$$U(N_i) = W(N_i)MSE(N_i) - W(N_i^l)MSE(N_i^l) - W(N_i^r)MSE(N_i^r) \tag{2.2}$$

The intuition is that if the MSE in the parent node is large and the MSE in the child nodes is small, that means the split must have provided lots of information relevant to the feature being predicted, and so the feature used to make the split must be important. The importance of a feature X_j is then calculated by summing the importance of all the nodes which use that feature to make a split. The importance is normalised such that it sums to 1.

$$I(X_j) = \frac{\sum_{i=1}^{n} U(N_i)v(N_i, X_j)}{\sum_{i=1}^{n} U(N_i)} \tag{2.3}$$

where $v(N_i, X_j) = 1$ if node N_i splits on feature X_j, otherwise it equals zero.

For a random forest the importance is calculated for each individual decision tree. The final importance of a feature is then given as the mean value across all the trees in the forest. One must be aware that correlations between input features will affect their importance values, and can make the results more difficult to

interpret. Since the sum is normalised to one, when examining feature importance plots the differences in the relative importance of each input feature should be considered, rather than their absolute values.

After training a model to predict a single baryonic property, I look at the feature importance to establish which input features contribute most to determining the value of the output feature. As my input features span a range of redshifts, peaks in the feature importance will tell us which times in a galaxies evolution are most important for setting the final value of each baryonic property. It should be noted that a high feature importance value does not tell us if an input feature is positively or negatively correlated with the output feature.

Determining hyperparameters

Machine learning algorithms often have hyperparameters. These are parameters of the model itself, and the values do not change when the model is trained. They control properties of the model such as its complexity, or how fast it learns. For the ERT I consider different values for *n_estimators*, *max_depth*, *min_samples_leaf*, and *min_samples_split*. I retain the default values for the other hyperparameters. The value of *n_estimators* sets the number of decision trees within the ERT. With too few decision trees the model will have a tendency to overfit the data. Having too many trees should not decrease the performance of the model, but it will increase the time the model takes to run and make predictions. The value of the *max_depth* hyperparameter limits how many nodes there can be in each decision tree. This can constrain the maximum number of input features each decision tree can use, since each depth splits on only one input feature. If the value of *max_depth* is too high the model may be prone to overfitting. The *min_samples_leaf* and *min_samples_split* hyperparameters combine to specify the minimum number of data points a node must contain in order to split into further nodes. By increasing the value of these parameters, I can decrease the total number of splits. This limiting of the number of parameters in the model can further prevent overfitting.

Picking the values for the hyperparameters can be seen as a black box optimization problem, where the objective function to be minimized is the performance of the model on a test data set. Common methods for tuning hyperparameters include random search and grid search. For this work I use Bayesian optimization (Agnihotri & Batra, 2020). It works by evaluating the performance of the

model for a small set of randomly chosen hyperparameters, a prior distribution is calculated to capture beliefs about the behaviour of the objective function. From this an acquisition function is calculated that determines the next values of hyperparameters to try and evaluate.

Bayesian optimization is useful when finding the arguments x^* that minimize a function, $f(x)$, when f has the following properties: its gradients are not known, it is expensive to evaluate, and its evaluations are noisy. In my case f is given by the MSE of the predictions on the test set by a trained model, and x are the hyperparameters used to train the model. After evaluating f with different values of x, I use a Gaussian processes (Rasmussen & Williams, 2005) to approximate f. To decide the next value of x to evaluate, I pick the values that minimize the lower confidence bound,

$$LCB(x) = u_{GP}(x) - \kappa\sigma_{GP}(x) \tag{2.4}$$

where u_{GP} is the mean of the fitted Gaussian process, and σ_{GP} is its standard deviation. f is then evaluated with the values x that minimize the LCB, and the Gaussian process fit is updated with the new information. The value of κ sets the exploration-exploitation trade-off. If κ is small, then the values of x that minimize the acquisition function will be very close to the minimum of u_{GP}. If κ is large, the x will be taken from a region with high uncertainty, where σ_{GP} is large.

2.2.4 Which snapshots to include

I use the ERT model and vary the snapshots that I include to see what effect it has on the performance of my model. The results when predicting the stellar mass of subhalos are shown in Figure 2.4, where a lower MSE score indicates better performance. The start snapshot represents the highest redshift snapshot passed as input to the model, and the step between snapshots gives the spacing. For example a start snapshot of 72 with a step between snapshots of 9 would correspond to the halo properties at snapshots 72, 81, 90, 99 being fed as inputs features. The values for the MSE are the average of 10 different training/test splits. As the maximum snapshot from IllustrisTNG is 99, the

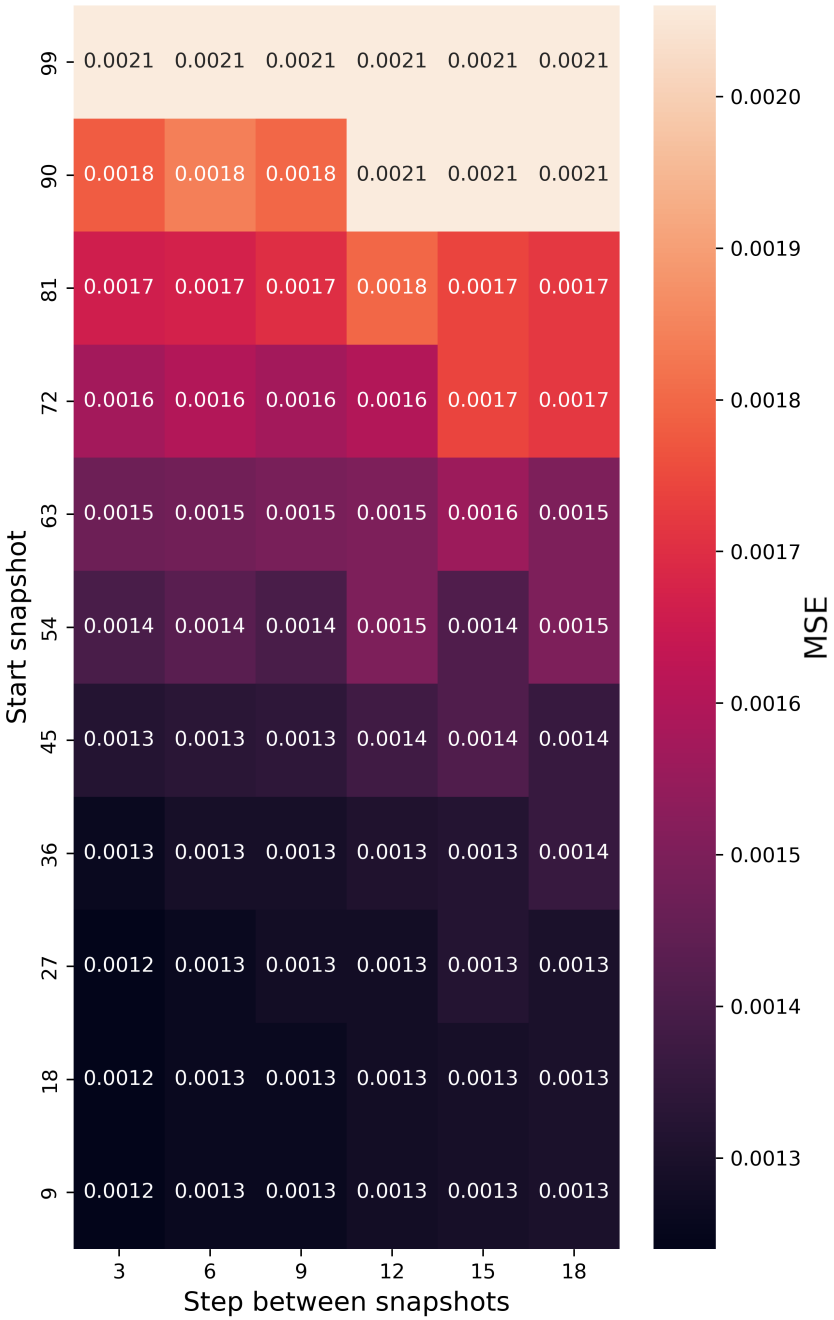

Figure 2.4 *Performance of the regressor for different snapshot ranges. A lower MSE score indicates more accurate predictions. The start snapshot values indicates the highest redshift halo properties passed to the machine learning model. The model performs more accurately as the start snapshot decreases, showing how including halo properties at high redshifts is beneficial.*

top row corresponds to the baseline model, where only the $z = 0$ properties are passed as input. It can been seen that starting at a lower snapshot improves the performance of the model. This improvement continues down to my lowest starting snapshot of 9, which corresponds to $z = 7.6$. Decreasing the jump between snapshots also causes the MSE to decrease, but the effect is much smaller than varying the starting snapshot. Setting the step size to 1 (not shown here) does not improve the model compared with a step size of 3. Similar figures showing the same trends are produced when predicting baryonic properties other than the stellar mass. From this figure I choose a starting snapshot of 9 with a step size of 10 as the full model for the rest of this work. I wish to start with the lowest possible snapshot to gain the most predictive power. I note that there is no disadvantage to starting with the lowest snapshot, even though not all subhalos can be tracked back to this point. I chose the larger step size as it is easier to interpret the feature importance plots when there are fewer input features, as discussed in Section 2.2.2.

2.2.5 Learning rate for different models

Figure 2.5 shows the learning rates for each model predicting baryonic properties. As expected, the performance of both the baseline and full model improves as the size of the training data increases. For most output features the difference between the performance of the two types of models is roughly constant as the training set gets larger, with both plateauing around the same point.

The shaded region represents one standard deviation in the values from the models being trained on 10 different training sets, and so shows the performance range that can be expected. When the training set is a region of high density the model performance is better. The shaded error decreases initially due to the variability in the size of training set, and increases again for larger training boxes as the size of the test set decreases.

The learning rates changes if the minimum mass of subhalos is increased. If the mass cut is high enough there is no difference in performance between the full and baseline models for small box sizes. This is a result of the small size of the training set which means there is not enough data for the machine learning model to pick up on information about formation histories.

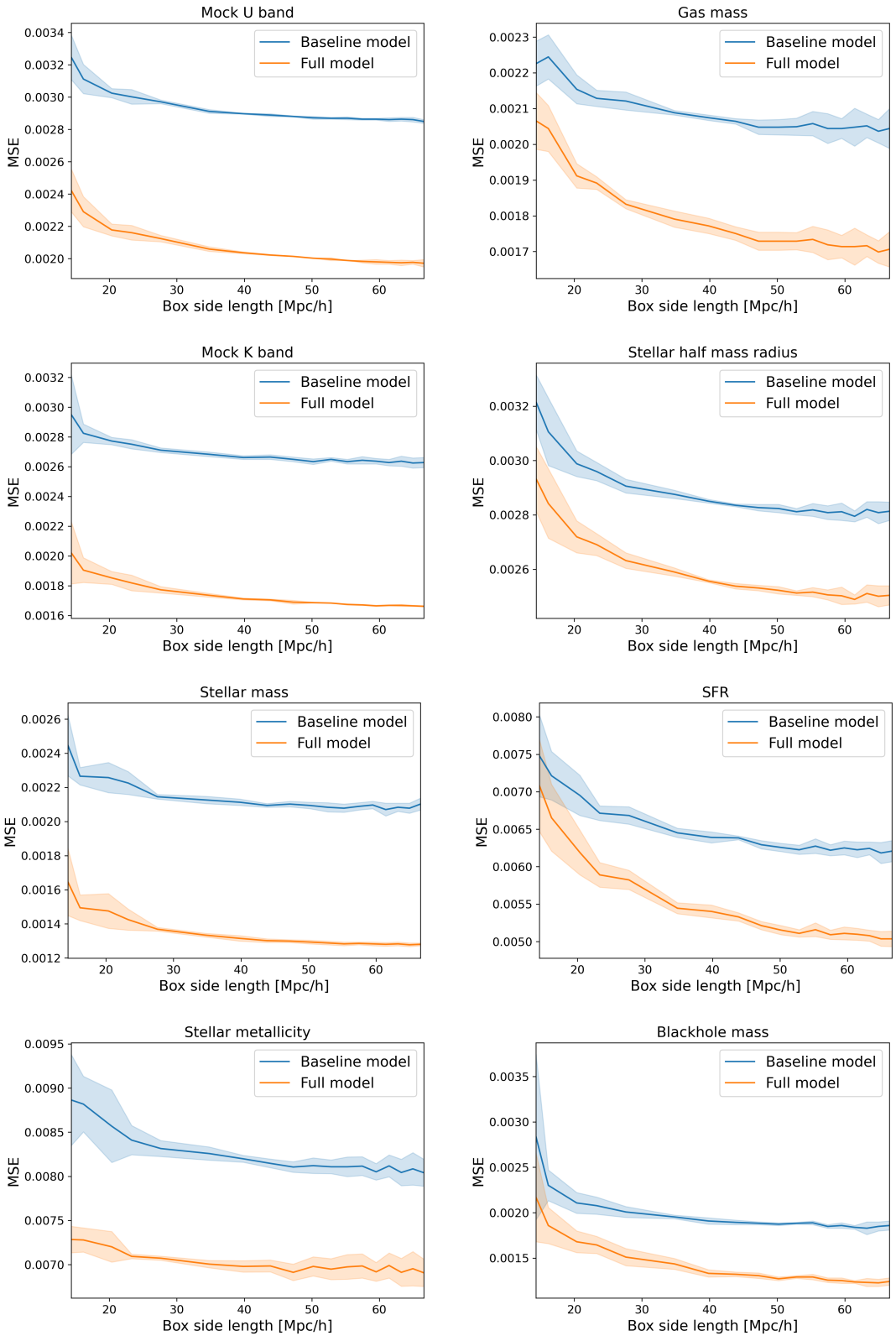

Figure 2.5 *Effect of the size of the training set box length on the performance of the models. Each plot represents the learning rate for one baryonic output feature. The shaded area represents one standard deviation in values for 10 different train/test splits.*

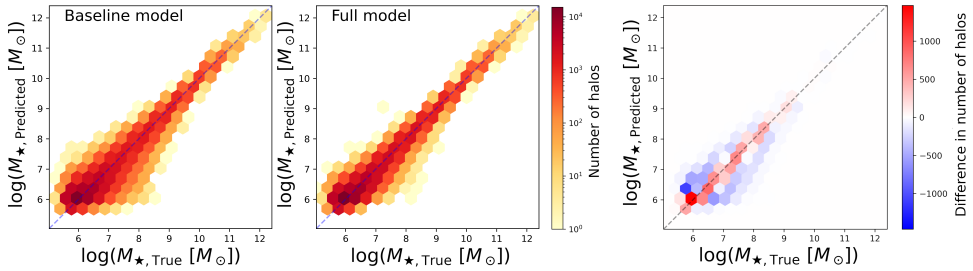

Figure 2.6 *(Left) A hexbin plot showing stellar mass values predicted by the baseline model compared with their true values. The blue dashed lines corresponds to a perfect prediction. (Middle) Same as left plot, except predictions are from the ERT model that takes in the halo properties from 10 snapshots, starting at redshift $z = 7.6$. Both plots are generated from the same train/test split. The scatter is reduced compared with the left plot, indicating an improvement in prediction accuracy. (Right) Difference in number of halos in each bin between the left and middle panels. Positive value indicate that the improved model has more halos in that bin than the baseline model.*

2.2.6 Comparison of models

The results of my different models are shown in Table 2.1. A lower MSE score indicates better model performance. The values given are the average of 10 different training/test splits. The standard error on the mean from the 10 runs is of the order 10^{-5} and so is not shown. This shows that in all cases the full ERT model outperforms the baseline model. I indicate this visually in Figure 2.6. In the left and middle panel I plot the stellar mass values predicted by the baseline and full models respectively, compared with their true values. It is clear to see that the scatter in values has decreased for the full model. To highlight this in the right panel I plot the residual of the left and middle panels. This shows that not only is there reduced scatter, but the full model has a larger number of predictions lying on the diagonal, indicating a correct prediction.

Since I have normalized the output features it is possible to get an idea of how difficult each feature is to predict by directly comparing MSE scores. I see that SFR and stellar metallicity are the most difficult to predict, and this is in agreement with previous work (e.g. Kamdar et al., 2016b). The reason that SFR is difficult to predict is due to it's stochasticity. Stellar mass, gas mass, and black hole mass are all integrated quantities which build up over time as gas falls onto the subhalo and is processed. This explains why they are

Table 2.1 *The mean squared error, Eq. 2.1, quantifying the performance of different models at predicting baryonic properties of subhalos. All scores are for predictions on the test set, aside from the final row.*

	Mass only model	Baseline model	Full model	Full model (Train)
BH mass	0.0017	0.0019	0.0012	0.0010
Gas mass	0.0019	0.0021	0.0017	0.0017
Half mass radius	0.0027	0.0028	0.0025	0.0024
U band	0.0024	0.0029	0.0019	0.0018
K band	0.0018	0.0026	0.0016	0.0015
SFR	0.0064	0.0061	0.0049	0.0045
Stellar Mass	0.0015	0.0021	0.0012	0.0012
Stellar Metallicity	0.0073	0.0081	0.0069	0.0068

easier to predict, as the stochastic processes involved in their rate of change are smoothed out by considering a large range of snapshots. The stellar metallicity is dependent on a number of complex factors, such as when the bulk of star formation took place, how much recycling of metals produced by previous star formation there was, and how much unpolluted fuel is available to the galaxy. The interplay of these numerous processes explains why the MSE is higher for stellar metallicity predictions than for any other output feature. The U band and K band magnitudes are strongly linked to the number of stars giving out light, i.e. the stellar mass. The MSE score for the U band is higher as it is linked to young stars and falls off quickly over time, and so it is more closely associated with the current SFR of the galaxy than the K band.

The improvement in score between the baseline model and the full model gives an indication of how important a subhalo's history is in determining the value of a certain property. I expect there to be a larger improvement for properties that are more dependent on the exact growth and merger history of the halo. For example the baseline model gives the same MSE score for both stellar mass and gas mass. Using the full model gives a much greater improvement to the stellar mass prediction than the gas mass prediction. This is to be expected as for most subhalos a significant fraction of their stellar mass was created at high redshifts, whereas the gas found in a subhalo at early times may be used up in star formation or blown out by feedback. The MSE is always lower for the full model than for the mass only model. This shows how it is important to include other properties of the subhalo at higher redshifts than just its mass history. When comparing the results of the mass only model with the baseline model I see that for most baryonic property predictions the mass only model is better. However for predictions of the SFR the baseline model is better. This is because SFR is

the only property I predict that is instantaneous, rather than being built up over time (e.g. stellar mass). In this case the machine learning mode finds it preferable to have as much information as possible about the $z = 0$ subhalo properties when predicting an instantaneous property. These results give the first indication that my models have shown nurture to be more important than nature. The full model can be linked to nurture, as it takes into account the evolution of the halo. The baseline model can be linked to nature, as it takes information about the halo at a single point in time. If nature was more important than nurture in determining a halo's properties, the model would be able to approximate the link between the halo properties at $z = 0$, and halo properties at the snapshot that defines the halo properties. However, as I see the full model always significantly outperforms the baseline model, then this cannot be the case.

In the final row of Table 2.1 I show the MSE of the full model on the training set. For most for the output features the MSE of the test set is slightly larger than the training set, but the difference is small enough that it shows my model is not overfitting.

While normalising the output features offers the advantage of enabling a direct comparison of the halo history's impact on a given property, it comes with the drawback that the score is difficult to interpret physically. This is a common issue as other papers in this area employ metrics, such as the Pearson correlation coefficient (Agarwal et al., 2018) or mean binned error (Jo & Kim, 2019), which also lack a direct physical meaning. One physically motivated error metric would be the mean absolute error (MAE), which is the average physical difference between predicted and true values. By considering the improvement in MSE between the baseline and the full models for predicting stellar mass, it can be estimated that this corresponds to a reduction of ~25% in the MAE.

2.2.7 Stellar mass function

The stellar mass function quantifies how many galaxies of a certain stellar mass are expected to be present in a random region of the universe. It is estimated using

$$\phi(M_*) = \frac{\mathrm{dn}}{\mathrm{dlog_{10}M_*}} \approx \frac{1}{V\,\Delta M}\, n_{sim}(M_* - \Delta M/2, M_* + \Delta M/2) \tag{2.5}$$

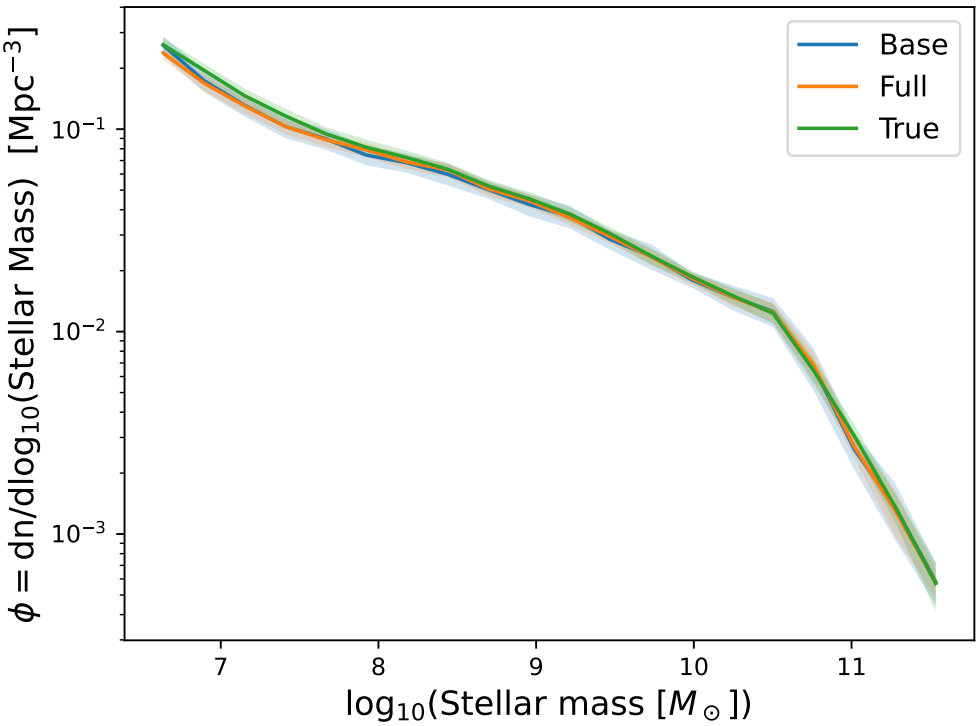

Figure 2.7 *The stellar mass function of 10 different test sets. The green line shows the true values, taken directly from the IllustrisTNG simulation, and the green shaded area represents one standard deviation in mass function values from the 10 test/train splits. The blue line shows the predicted stellar mass function from the baseline model, and the orange line shows the prediction from the ERT model that takes in the halo properties from 10 snapshots, starting at redshift* $z = 7.6$.

where V is the simulation volume, $n_{sim}(M_i, M_j)$ is the count of galaxies within the simulation that have a mass between M_i and M_j, and ΔM is the mass bin spacing. ΔM should be picked to be as small as possible, but will be limited by the simulation volume.

In Figure 2.7 I show the stellar mass function of the subhalos in the test set. I compare the mass function from the true values compared with the baseline model and the model that takes in the halos full history properties. The shaded area represents 1 standard deviation from 10 different random choices of the train-test split. Both the baseline and full models agree with the true values in that they are within one standard deviation over the full range of stellar masses considered. This indicates that if the only aim of populating a dark matter only simulation is to reproduce mass functions, then using the baseline model is sufficient. However, as is shown in Table 2.1, including halo history leads to improved performance for individual galaxies. Therefore, future work in this area must focus on further improving quantifiable metrics such as the MSE, as opposed to stopping when mass functions have been matched.

2.3 Insights from models

2.3.1 Feature importance from ERT models

In Figure 2.8 I show the plots of the feature importance values of each input feature fed into the full model. As I train a separate model for each baryonic property being predicted I end up with 8 different plots of feature importance. The feature importance is calculated as described in Section 2.2.3. The sum of the feature importance over all features in a model is normalised to be equal to one. Each point on the plot represents one input feature to the full model. The shaded region is the 1σ standard error in the mean from 10 different training/test splits. A test set is not needed to calculate the feature importance, but I train with 10 different splits to get an estimate of the variation. It is clear from the contrasting plots in Figure 2.8 that the feature importance varies significantly depending on the output feature which is being predicted. This shows my models are picking up on the different ways each baryonic property is built up. There are some common trends, mainly that the halo velocity dispersion and halo maximum velocity are determined to be the most important features. Both Kamdar et al.

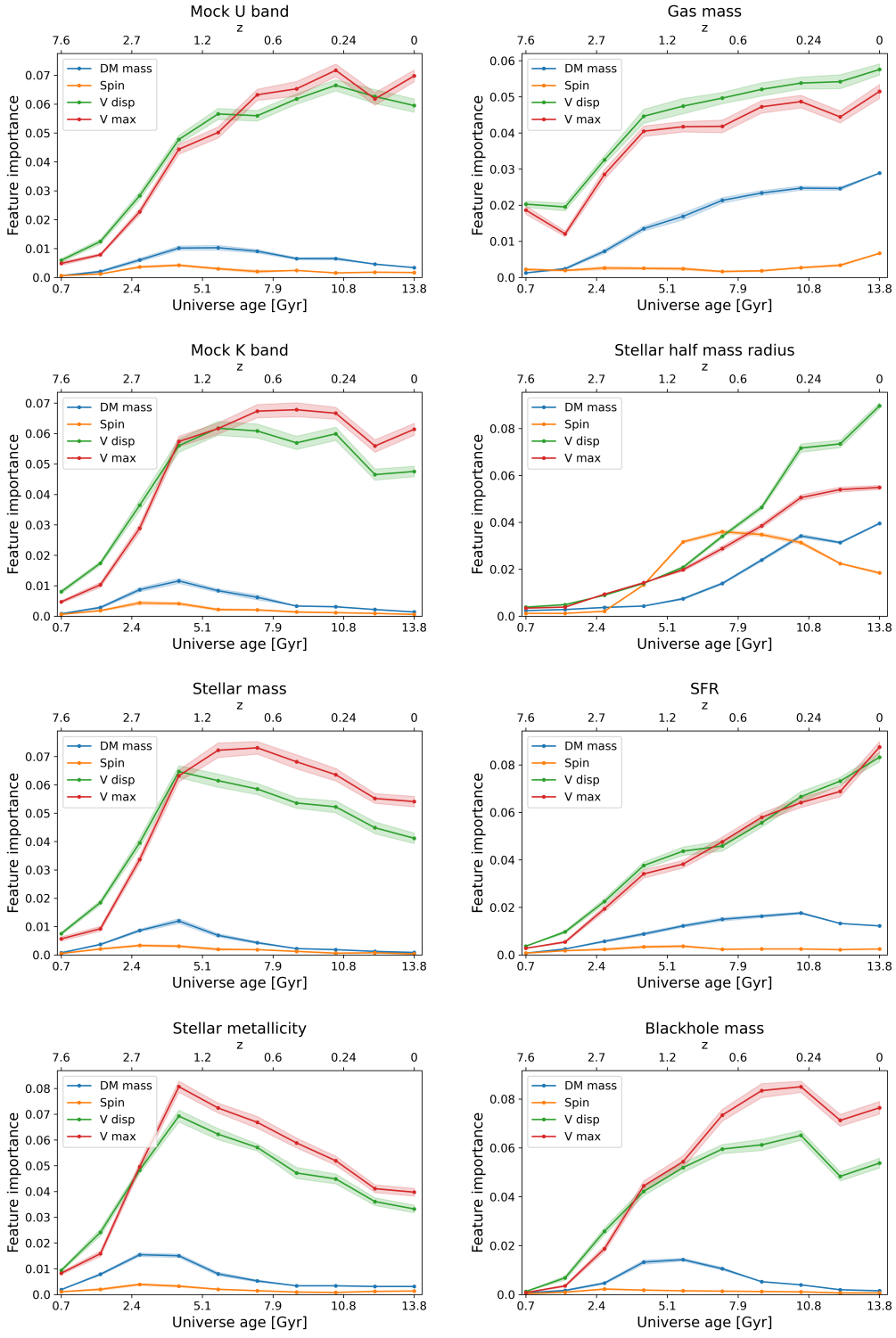

Figure 2.8 *Feature importance values from the ERT model that takes in the halo properties from 10 snapshots, starting at redshift $z = 7.6$. The shaded region represents one standard error in the mean based on training and evaluating the model 10 times. The variation results from a combination of the different training sets used each time, and from the inherent randomness in the ERT algorithm.*

(2016a) and Agarwal et al. (2018) looked at feature importance, but used similar models to my baseline model, i.e. they did not consider the full halo growth history. They also did not train a separate model for each output feature, so the feature importance values found were not associated with a single $z = 0$ galaxy property. In agreement with my overall trends, they also found velocity dispersion and maximum velocity to be good predictors. Also in concurrence with previous work I find that in general spin is the feature which provides the least information. The fact that for all plots there are times where the spin feature importance goes to zero is evidence that my models are not over-fitting.

The first major difference between the feature importance plots is the location of the peak. Comparing the peak of the feature importance plots gives an indication of what epoch is most important for the build up of that baryonic property. Unsurprisingly SFR peaks at $z = 0$. This agrees with the discussion in Section 2.2.6 about the score difference between the mass only and baseline model. Although feature importance plots such as Figure 2.8 can only be obtained from decision tree-based machine learning models, similar MSE scores to the ones shown in Table 2.1 are obtained if I use a different algorithm. The fact that my feature importance plots agree with the model-agnostic MSE results is a confirmation that the feature importance are able to determine physically interesting results.

The feature importance plots for stellar mass and stellar metallicity are similar. This is to be expected as metallicity of the stellar particles is strongly correlated with the time at which the stellar mass is formed. For the IllustrisTNG simulations the peak in cosmic star formation rate density occurs around $z = 2$. I might expect the stellar mass feature importance to peak at the same point, but it appears later, around $z = 1$. The stellar mass and K band magnitude plots are similar. K band magnitude is often used as a proxy for stellar mass (e.g. Cowie et al., 1994; Gavazzi et al., 1996; Kochanek et al., 2001; Cole et al., 2001), and my models are able to independently pick up on this connection. Comparing the feature importance of the K and U bands shows a peak at different times. The U band peaks at $z = 0$ which makes sense as UV light is emitted by young stars and so is correlated with SFR.

Looking at the plots for SFR and for gas mass, I see that in both cases halo velocity dispersion and halo maximum velocity are most important but the importance of dark matter mass differs. For SFR the peak in dark matter mass is prior to $z = 0$. As the dark matter mass determines the gravitational potential

of the subhalo it will be linked with the amount of infalling gas. However this gas can only be used for star formation once it has cooled, whereas the gas mass of the subhalo includes both hot and cold gas. This explains why the peak in dark matter mass feature importance does not occur at $z = 0$ for SFR, but does for gas mass.

Models of galaxy formation and evolution often assume a relation between the spin of a galaxy and its host halo (e.g. Fall & Efstathiou, 1980; Mo et al., 1998). This relation is used to set the size of galaxies in many semi-analytic models. However, recent work has suggested that this relation may not hold in cosmological simulations (e.g. Danovich et al., 2015; Jiang et al., 2019). From my feature importance plot for stellar half mass radius I see that within IllustrisTNG the spin of a halo is a predictor of galaxy size, in agreement with Yang et al. (2021). I find that the importance of the halo spin in determining the $z = 0$ galaxy size peaks around $z = 1$. This shows that the galaxy size at $z = 0$ is less correlated with the halo spin at $z = 0$ than the halo spin at $z = 1$. This suggests that at earlier times the halo spin was important in determining the angular momentum of the galaxy, and therefore the galaxy size. However, as the halo has continued to grow and evolve after the galaxy has formed the spin of the halo no longer effects the galaxy. This information could only be found because my method takes in such a wide range of redshifts. When considering the full range of redshifts I find that the other halo properties are more important than spin.

2.3.2 Nature vs Nurture

I here define the nature vs nurture problem as the question whether the properties of a galaxy can be determined if you know its state at a single point in time, or if one needs to consider its evolution through time. Feature importance is a useful approach to this question as it allows us to distinguish whether single points during the evolution of galaxies have a large impact on their present-day properties. By considering a wide range of snapshots as inputs my method provides insight into this question in a way previous approaches could not. My results suggest that nurture is more important. If nature was the most important I would expect the feature importance plots to peak at high redshifts which correspond to the initial conditions of the subhalo. However this is never the case, and for most output features the feature importance goes to zero at very high redshifts. It might be thought that this is because some subhalos cannot be

tracked back to $z = 7.6$ and so are skewing the peak of the feature importance plots to lower redshift, but if I calculate feature importance plots by training models using only subhalos that can be tracked to $z = 7.6$ I still do not find a peak at that point. Instead I find a peak at later points, which I discuss in Section 2.3.3. Even for properties whose feature importance peaks at early times, such as stellar mass, the feature importance is still high around $z = 0$. This shows that the evolution of the host halo at late times always plays a key role in determining the redshift zero galaxy properties.

To quantify nature vs nurture I consider the following integrals of the feature importance over time,

$$I_{\rm nat} = \int_{t_{peak}-\frac{\Delta t}{2}}^{t_{peak}+\frac{\Delta t}{2}} F(t)dt \tag{2.6}$$

$$I_{\rm nur} = \int_{t_0}^{t_{peak}-\frac{\Delta t}{2}} F(t)dt + \int_{t_{peak}+\frac{\Delta t}{2}}^{t_f} F(t)dt \tag{2.7}$$

where t_0 is the earliest time considered, t_f is the final time considered, t_{peak} is the time at which the feature importance peaks, Δt is the time around the peak to consider, and $F(t)$ is the feature importance over time.

Although the values of feature importance I obtain are at single points in time, I argue that feature importance can be treated as a continuous quantity, and therefore integration is a valid technique. This is because the physical properties that are used as input features are well-behaved, i.e. they evolve smoothly and do not exhibit any large discontinuities. I verify in the next section that if a smaller spacing of snapshots is considered then the trends shown in Figure 2.8 are unaffected.

To evaluate the integrals in equations 2.6 and 2.7 I need to choose a value for Δt. A natural choice would be the dynamical timescale, as any environmental effects associated with the galaxy evolving by nurture will take longer than this to affect the galaxy. I calculate the dynamical timescale for all subhalos in my sample with

$$t_{dyn} = \left(\frac{2R^3}{GM}\right)^{\frac{1}{2}} \tag{2.8}$$

Table 2.2 *The ratio of $I_{\rm nat}$ to $I_{\rm nur}$ for each of the output properties (first column) being predicted based on input properties (top row). Values larger than 0.5 suggest nature is more important for a given physical property of galaxies, while values smaller than 0.5 support nurture as the main driver. As can be seen in the table, all values for the galaxy properties listed here are below 0.5 and thus nurture is the dominant driver of galaxy properties.*

	DM mass	Spin	Vel disp	Vel max	All
BH mass	0.33	0.21	0.16	0.20	0.18
Gas mass	0.23	0.34	0.16	0.17	0.17
Half mass radius	0.30	0.23	0.32	0.26	0.22
U band	0.21	0.21	0.19	0.19	0.18
K band	0.31	0.32	0.17	0.17	0.16
SFR	0.24	0.21	0.25	0.25	0.23
Stellar Mass	0.37	0.22	0.17	0.16	0.17
Stellar Metallicity	0.32	0.32	0.18	0.19	0.18

where M is the subhalo dark matter mass and R is twice the radius of the rotation curve maximum. I set Δt = 1.5Gyr, as I find that 99% of subhalos have a dynamical time less than this. I calculate the ratio of $I_{\rm nat}$ to $I_{\rm nur}$ for each of the output properties, and show the results in Table 2.2.

For properties that peak at $z = 0$, I integrate between $t_f - \Delta t$ and t_f. For all output features being predicted $I_{\rm nur}$ is significantly larger than $I_{\rm nat}$. This shows that the majority of the predictive power of the model cannot come from a single snapshot. Therefore the physical processes that determine the value of the output property cannot occur at a single point in time. SFR has the highest fraction of its feature importance in its most important snapshot. This indicates that of the output features I consider it is the most set by nature rather than nurture.

Validating the integration of feature importance plots

For general applications feature importance can be a non-continuous function. However, I here assume that for physical properties which are continuous over time, as is the case for the subhalo properties, the feature importance for that property has to be continuous as well. In Figure 2.9 I show the feature importance values of a model trained using more finely spaced snapshots than Figure 2.8. The grey lines show the feature importance values obtained for the original snapshot spacing. As the sum of the feature importance is normalised to one, the grey lines have been rescaled. It should be noted that the standard errors are larger

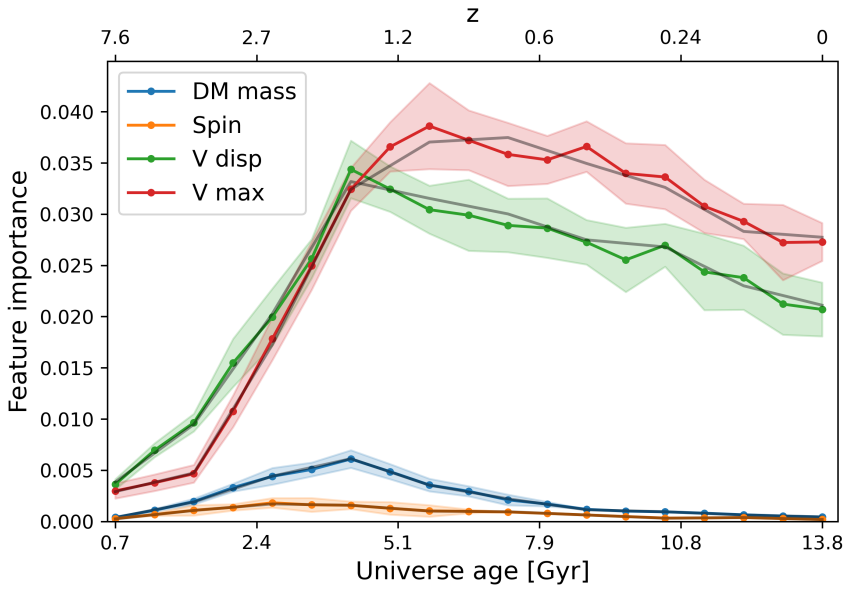

Figure 2.9 *Feature importance values from the ERT model predicting stellar mass that takes in the halo properties from 19 snapshots starting at redshift $z = 7.6$. The grey lines show the feature importance when only using 10 snapshots, as shown in Figure 2.8.*

for Figure 2.9 than Figure 2.8. This comparison of figures allows me to test whether discontinuities could be present in the feature importance that have been smoothed over due to the time step binning. I find that within the limits of the simulation data non such exist, as the grey line always lies within the standard error. This supports my assumption that for this application the feature importance of a single property evolves continuously.

To confirm whether the shapes of the feature importance values in Figure 2.8 are due to nature or nurture I consider a toy model. I generate 10000 mock input vectors. For each input vector I generate 2 sets of 21 numbers from $U_{[0,1]}$. For each one of these sets I sort the numbers. A set of 21 number corresponds to a single property sampled over 21 snapshots. This mimics the evolution of the dark matter properties of the halos I consider, which in general grow over time. I then create different output features corresponding to nature and nurture. I train ERT models to predict these output features, and plot their feature importance in Figure 2.10.

For my nature toy model the output feature being predicted is determined by the difference between a property at snapshot s and $s - 1$. In the top panel of Figure 2.10 the output feature is dependent on the difference between snapshot 10 and 9 from property 1 and snapshots 15 and 14 from property 2. I weight

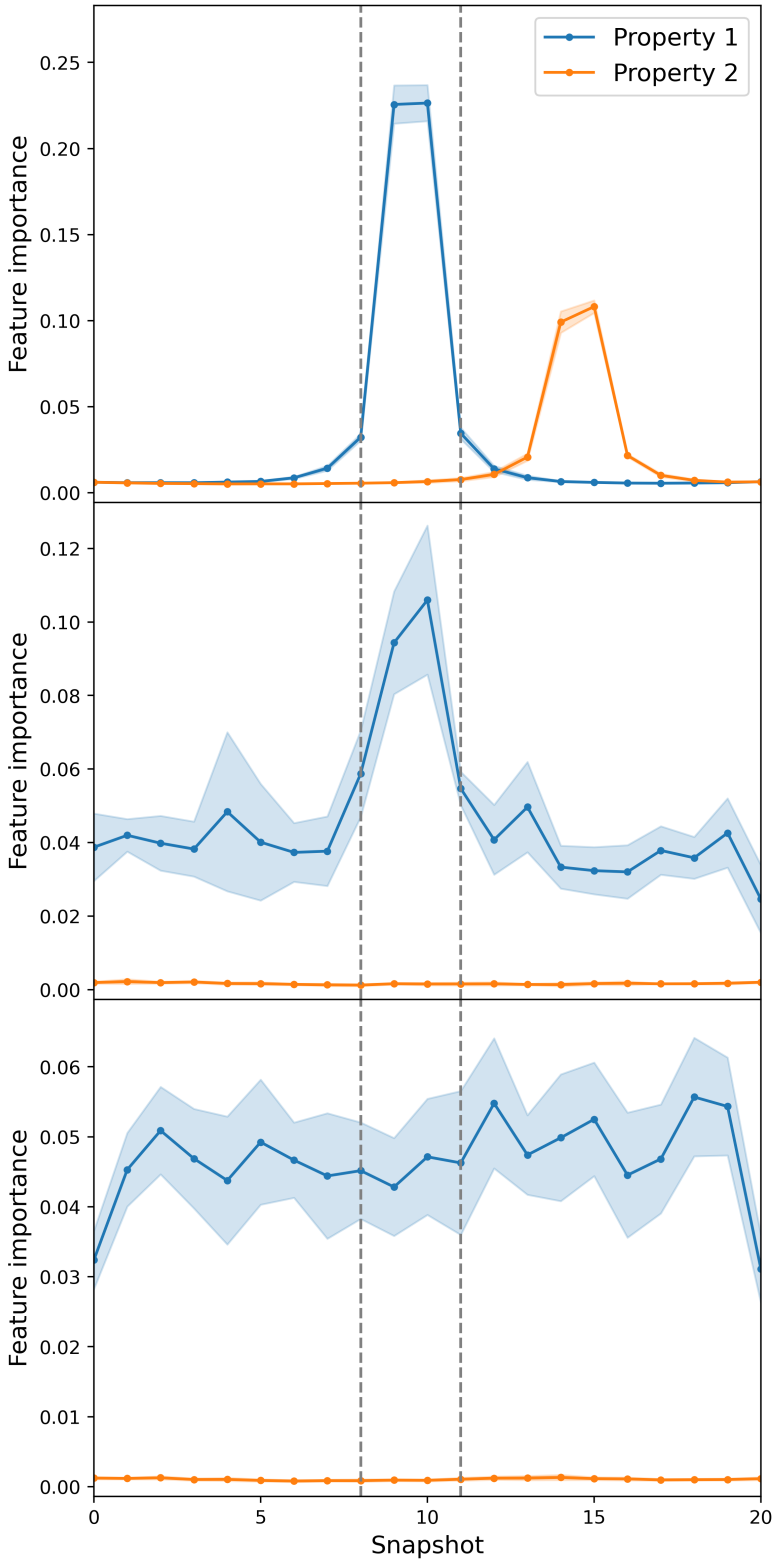

Figure 2.10 *Feature importance values from an ERT model trained to predict the output features of a toy model. (**Top**) Nature model (**Middle**) Mixed model (**Bottom**) Nurture model*

the contribution from property 1 as twice that from property 2. This results in a feature importance plot with distinct spikes at the snapshots which determine the output feature. Due to my weighting the spike for property 2 is lower than that for property 1. From this top panel I set the limits of my integral as indicated by the grey dashed line.

For my nurture model the output is given by the sum of the squares of the differences between all consecutive snapshots. In the bottom panel of Figure 2.10 I show the feature importance of the nurture model. The drop for snapshot 0 and snapshot 20 is because they are only used once in the calculation of the output feature, unlike all other snapshots which are used twice. Within the standard error the feature importance of this model is flat, as I would expect.

For my mixed model I combine the output features of the nurture model and a nature model that only depends on the difference between snapshots 10 and 9 from property 1. I weight the nurture model by a factor of 5. The resulting feature importance is shown in the middle panel of Figure 2.10. The most distinct feature is the spike around snapshot 10, which may initially suggest that nature is more important in the determination of the final output property. However the ratio of the integral within the grey lines to the integral outside the grey lines is equal to 0.35. This is less than the ratio of the weighting, but reflects the fact that the integral for between the grey lines for the pure nurture model is nonzero. Therefore considering the integral of the feature importance plots allows for a comparison of the effects of nature vs nurture.

2.3.3 Best snapshots to use for predictions

I wish to further verify that the trends shown in the feature importance plots are physical. To do this I train a number of models using halo properties from two different snapshots to predict the stellar mass at redshift zero. In Figure 2.11 I show the MSE scores that result from these different models. To generate these scores I used the ERT algorithm, but I have verified that the plot looks similar when other machine learning algorithms are used. The red lines have been added to highlight trends, and indicate contours of constant MSE.

The diagonal of Figure 2.11, where the first and second snapshot are equal, corresponds to a model trained on a single snapshot. Looking at the trend along this diagonal I can see that predictions are worst when using halo properties

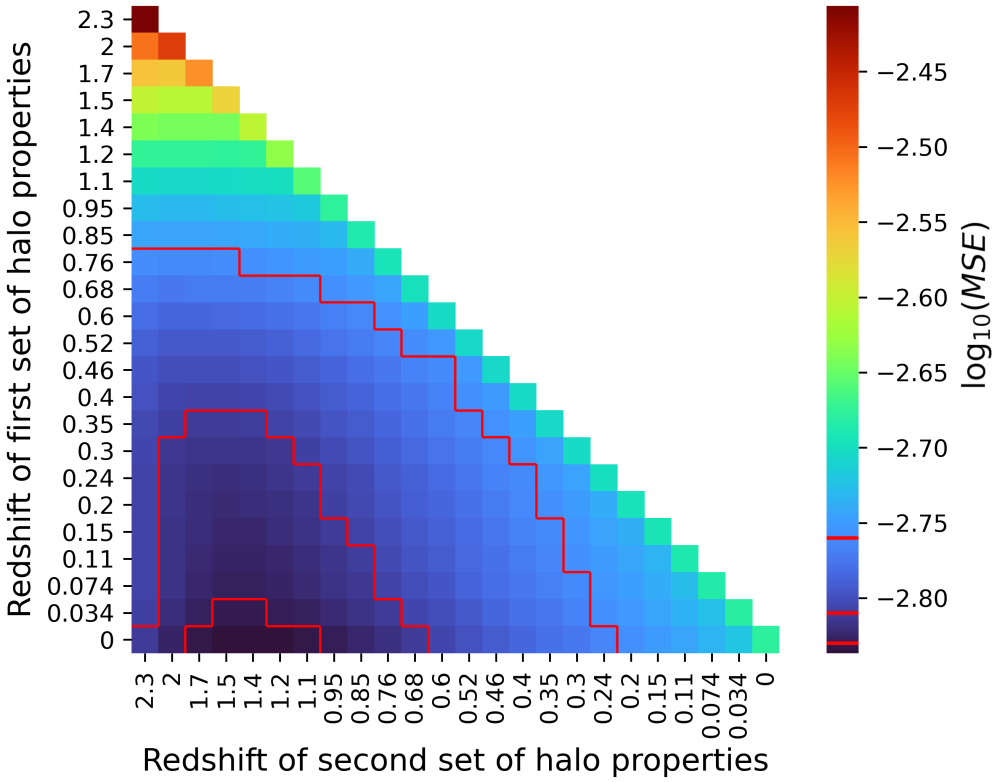

Figure 2.11 *Heatmap showing MSE scores when predicting subhalo stellar mass at $z = 0$ when halo properties from two snapshots are used as input to an ERT model. MSE scores have been logged to better highlight trends. Red lines indicate contours of constant MSE. The diagonal corresponds to input features from a single snapshot only, i.e. my baseline model. Adding a second snapshot gives significantly improved results, with the best performance coming when the first and second snapshot are well spaced.*

from a high redshift. As the redshift of the input properties decreases the model performance increases, but the increase plateaus around $z = 0.6$. When generating similar figures for predicting the other baryonic properties, this plateau happens at different redshifts. The plateau occurs latest for SFR and gas mass, agreeing with where their peak in feature importance appears.

When looking at the off-diagonal elements which correspond to models with input features taken from two snapshots, I see significant improvements compared with the single snapshot case. This highlights the fact that even adding one more snapshot already has a strong impact, and that limiting inputs to a single snapshot is hindering the ability of the model to make predictions. I would expect this behaviour for any system for which nurture is more important than nature. It reflects the underlying physics behind determining galaxy properties which is that it is key to consider as much information as possible about their history/evolution.

In Figure 2.12 I train models to predict redshift zero baryonic properties. I always pass the $z = 0$ dark matter properties as input, and I vary the redshift of the second set of halo properties fed into the model. Thus the central plot, which predicts the stellar mass of the entire subhalo population, is equivalent to the the bottom row of Figure 2.11. The red dashed line shows the MSE score of the baseline model. The grey dashed line shows the minimum point. In the bottom panel I show the prediction for the SFR rate. As the $z = 0$ properties are already provided as the first snapshot, the minimum of the second snapshot does not occur at redshift zero. It is significantly later than the minimum in the middle panel, again confirming that the different locations of the peaks in the feature importance plots are not just artefact of the ERT algorithm. In the top panel I predict the stellar mass of subhalos which have $M_{\rm DM} > 10^{11}$. The minimum occurs much later than in the middle panel. This shows evidence of hierarchical formation. For high mass subhalos a large fraction of their stellar mass comes from mergers with other subhalos. As I only consider the main progenitor branch, when considering a high redshift snapshot much information about the assembly history of the halo will not be included. For low mass subhalos which have evolved without many interactions, their growth history is smoother, and so taking an early snapshot does not lead to information being missed out. As the halo mass function is biased to low mass halos, the minimum for the MSE occurs at early times. If the hierarchical growth model was not correct I would expect the location of the minimums to be reversed, as large halos would form earlier than their smaller counterparts.

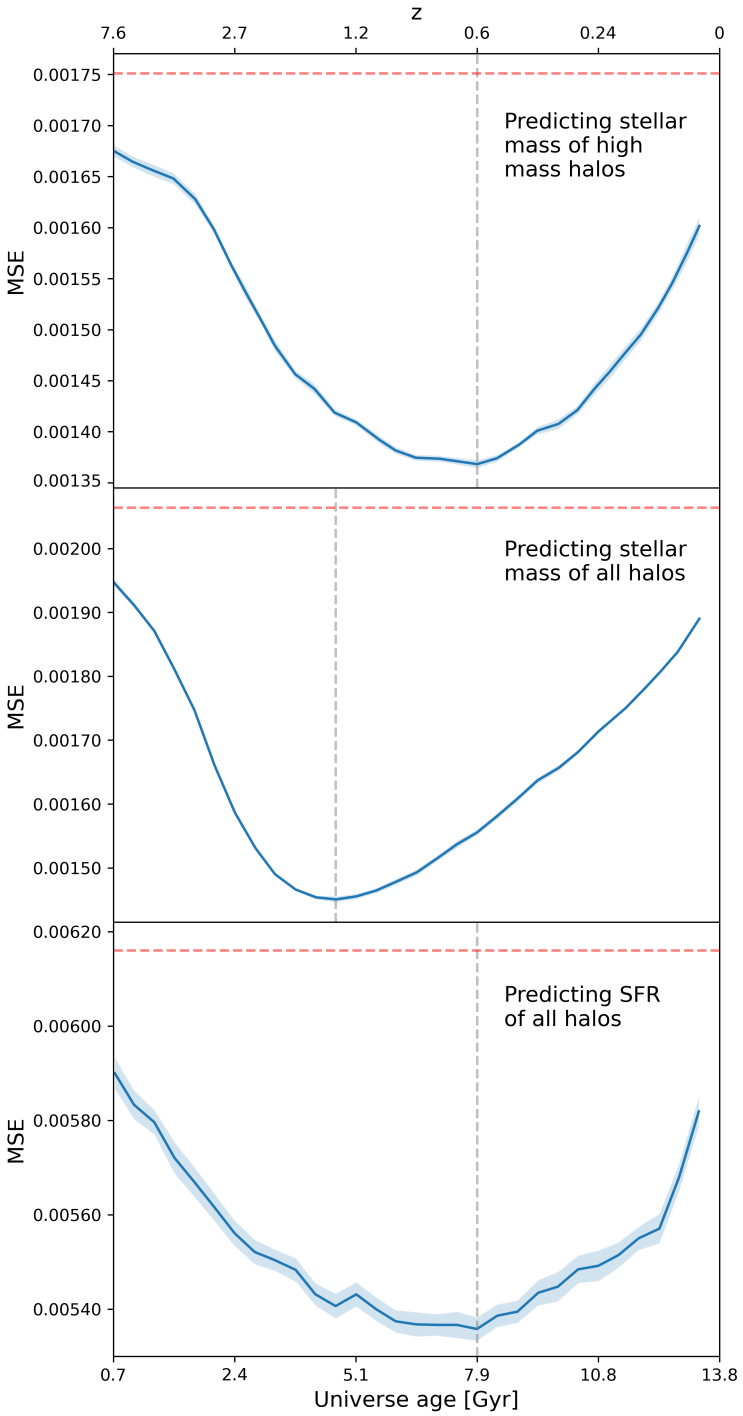

Figure 2.12 *The effect of varying the second snapshot of halo properties passed to my models. The first snapshot halo properties are fixed to be redshift 0. The shaded region represents one standard error from 10 train/test splits. The minimum MSE score is shown by the grey dashed line. The performance of the baseline model is indicated by the red dashed line. (**Top**) Prediction of stellar mass for halos with $M_{DM} > 10^{11}$ (**Middle**) Prediction of stellar mass for all halos (**Bottom**) Prediction of SFR for all halos*

Figure 2.12 also provides information about whether the resolution of objects close to the mass cut (10^9 $M_\odot$) impacts the feature importance plots. Due to the power law nature of the halo mass function the training set is dominated by objects with a low mass. For the central panel which shows all halos the minimum MSE scores occurs at the same point as the feature importance peak for stellar mass in Figure 2.8. From the top panel we would therefore expect that feature importance plots for a mass cut of 10^{11} $M_\odot$ would peak around $z = 0.6$. Jung et al. (2023) produce feature importance plots using the full merger history of subhalos from the TNG simulations (they do use different input features). They find a peak in the feature importance values prior to $z = 0$ when using a mass cut of 10^{10} $M_\odot$. As discussed in the previous paragraph this shift in the peak's location when changing the mass cut has a clear physical origin. Therefore we can conclude that the shape of the feature importance plots is not a result of resolution issues.

2.3.4 Discussion

Figure 2.12 shows that when looking at subsamples of the subhalo population the MSE scores will differ (as shown by the value of the red dashed line in the top vs middle panel), as will the most important snapshot (as shown by the grey dashed line). This suggests that if I look at the feature importance for models trained on different subsamples of the population I should get different feature importance plots. This shows how my model can pick out the fact that different populations of galaxies form in different ways. In general the relative importance of each halo property remains the same, but the peak moves, indicating that different times are most important for the formation of galaxy subpopulations. However, in some cases I find that the relative importance of the feature importance changes, indicating a different formation or evolution mechanism. I defer a detailed look into the feature importance plots of subsamples of galaxies until Chapter 3.

In Figure 2.13 I look at how the feature importance changes when predicting stellar mass at $z \neq 0$. For this plot I train a baseline model for each redshift. Note that at each time on the horizontal axis the feature importance will sum to one, whereas in Figure 2.8 the sum of feature importance over the whole plot is equal to one. This means there is no peak in feature importance in Figure 2.13. There are two major effects on the feature importance from using such a small number of input features. First the variation becomes greater. Secondly, the spin

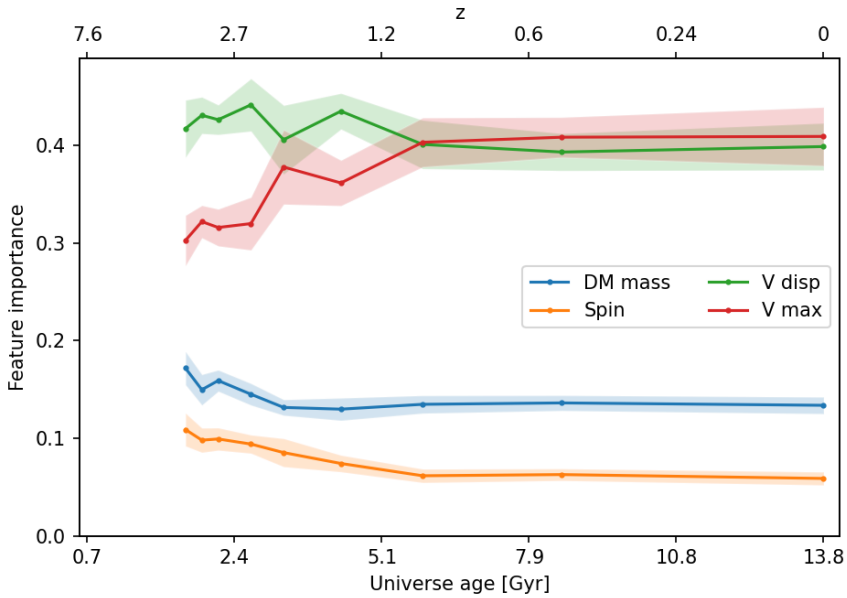

Figure 2.13 *The feature importance values from models trained to predict stellar mass at different redshifts. The baseline model is used, with the input being the subhalo dark matter properties from the redshift being predicted.*

feature importance is never zero. This is a result of the model overfitting as it does not have enough features to split on. This can be driven to zero by tuning the *max_depth* hyperparameter.

The relative importance of each property agrees between Figure 2.8 and Figure 2.13. The halo velocity dispersion is most important property at the highest redshifts with halo maximum velocity becoming the most important at later times. Spin is always determined to be the least important. Comparing the feature importance between the full model and single snapshot predictions yields similar results for the other baryonic properties. This is another reassuring test of the robustness of the multi-epoch approach. The halo properties that determined the stellar mass at $z = 1$ should remain the most important when looking at the relative importance of $z = 1$ input features of the full model. I expect this because most stars present in the galaxy at $z = 1$ remain in the galaxy at $z = 0$.

2.4 Properties of high redshift quasars

The previous sections of this chapter explored the method of using machine learning to predict galaxy properties within the context of simulations, and

what information that could provide about the simulations. In this section I apply the method to generate a mock catalog, and then compare it directly with observations. This facilitates the production of a large volume mock catalog, enabling comparisons of clustering which would not otherwise be possible. The findings from Section 2.3 guide the selection of properties which are essential to use as input into the machine learning models. The analysis from the previous sections give provides confidence that the catalogs produced will be consistent with the training simulation. To my knowledge this is the first direct comparison of a machine learning generated mock catalog with observational data.

The focus of this section is the characteristics of the black hole mass function at high redshifts. We do not yet have a comprehensive understanding of how black holes grow. This is reflected in the fact that simulations deviate significantly in their predictions at this epoch (Habouzit et al., 2022). To determine which simulations are reproducing the correct trends we need to be able to compare them directly with observations. This can help to highlight issues in the current generation of subgrid models. In addition, studying the growth of black holes in simulations can help inform observational strategies about what measurements are needed to be taken, and also about the best regions to observe to get those measurements.

However, for high redshifts only the brightest objects are detected, and this means large areas need to be covered in order to build up a statistically significant sample. This is difficult to reconcile with hydrodynamical simulations, which are too computationally expensive to run for such volumes. In this section I apply a machine learning modelled trained on the IllustrisTNG simulations to a large volume N-body simulation. The mock catalog is then compared with observations of quasars at the same redshift.

2.4.1 Observational data

The primary source of the observational data is NED [1], which contains sources from several optical surveys, mainly the SDSS (Ahumada et al., 2020). All known quasars at $z \geq 3$ and their associated properties were collected. For each extracted object any papers that studied them were found, and the accompanying VizieR [2] catalogues (Ochsenbein et al., 2000) were used to collect derived properties. This

[1] NASA/IPAC Extragalactic Database
[2] vizier.u-strasbg.fr

provided physical properties, such as redshifts, luminosities, masses, line widths, and Eddington ratios. The data is publicly available on the Kaggle data platform [3].

The quasar luminosity function (QLF) is one of the key inputs to current conceptual models of BH growth (Volonteri, 2012; Natarajan, 2014). Observationally determined QLFs can be compared with those predicted by theoretical models of BH growth. After calibrating them with data the models can then be extrapolated down to fainter luminosities than current sensitivities and out to larger redshifts than current detections. Some of the most recent predicted QLFs extrapolated out to $z = 9$ derived from a combination of observational data and modeling can be found in Ricarte & Natarajan (2018). The QLF provides a census of the number of sources at a given redshift and absolute magnitude $\boldsymbol{M}$, but is not easy to determine. Calculating it depends on an accurate estimate of the volume which the survey is covering. This means that the specifications of observational surveys need to be taken into account to determine the QLF. An estimate for the survey sample completeness is needed, to determine how many sources are present in the volume but have been missed. The redshift bin size also needs to be known. The minimum and maximum redshift depend on the specifications of the survey, but also on the individual sources, as certain objects will remain detectable out to higher redshifts.

The fundamental properties of a black hole are its mass and spin. While mass estimates are available for several thousand sources at the present time (e.g. Kelly & Shen, 2013; Peterson, 2014; Vestergaard, 2019), spin measurements are available only for a handful of sources (e.g. Reynolds, 2020; Nandra et al., 2006). As observed correlations are between SMBH mass and host galaxy properties I do not consider black hole spin here, although there is no reason why it could not be predicted using machine learning, assuming the value is present in the training simulation. Many independent methods have been used to derive BH masses - including mapping of the orbits of individual stars within the Milky Way (Genzel et al., 1997; Ghez et al., 1998), modeling the orbits of bulge stars from imaging and spectroscopy as performed for nearby galaxies (Tremaine et al., 1994), and using measurements of the speed of rotating gas using water mega-masers as tracers of the mass (Miyoshi et al., 1995).

One widely adopted method for SMBH mass determination assumes that the

[3] QUOTAS database

BLR is virialized and that the motion of the emitting clouds therefore reflects the gravitational potential of the central BH (Blandford & McKee, 1982; Peterson, 1993). Under this assumption, the black hole mass $M_{\rm BH}$ can be estimated as

$$M_{\rm BH} = f\frac{{V_{\rm vir}}^2 R_{\rm BLR}}{G}, \tag{2.9}$$

where $V_{\rm vir}$ is the virial velocity, $R_{\rm BLR}$ is the size of the BLR, and f is the virial coefficient that accounts for the geometry and kinematics of the material around the BLR (Shen, 2013). The virial velocity can be estimated using the velocity dispersion derived from the width of observed BLR emission lines. $R_{\rm BLR}$ can be estimated using a technique known as reverberation mapping (Peterson & Horne, 2004) in which the time-lagged broad-line response to variations in the continuum flux enable the measurement of the light travel time from the central ionizing source to the broad line regions. However, acquiring these time lags from reverberation data is challenging, as it requires a long observational baseline, monitoring an accreting BH for six months to a year (Peterson et al., 2004; Grier et al., 2017).

An alternative method for SMBH mass measurements, that is not predicated on the assumption of virialization of the BLR, is one in which luminosities from the X-ray, ultraviolet, infrared, and optical wavelengths can be used to estimate the BLR size. Reverberation mapping has revealed a tight correlation between the size of BLR and the continuum luminosity (Kaspi et al., 2000, 2005; Bentz et al., 2009). Therefore by combining the continuum luminosity with the widths of broad emission lines, an empirical scaling relationship can be used to derive the black hole mass of quasars. The relationship is given as

$$\log M_{\rm BH} = a + b\log L + c\log(\Delta v) \tag{2.10}$$

where L is the continuum luminosity and Δv is the width of the broad emission lines. The values of prefactors a and b depend on the choice of the luminosity and velocity dispersion estimators, and if a virialized BLR is assumed, then the coefficient of the line width is taken to be $c = 2$ (Shen & Liu, 2012). Various cross-calibrations for this relation have been proposed (Shen & Liu, 2012; Vestergaard & Peterson, 2006; Vestergaard & Osmer, 2009; Trakhtenbrot & Netzer, 2012). Additional BH mass estimates are often made using individual lines, such as Carbon-IV (Vestergaard & Peterson, 2006).

The mass estimates in the data used for the comparison presented in this thesis

are based on a combination of the methods mentioned above, depending on the observations available for the source being considered.

2.4.2 Populating the Legacy N-body simulations

The Legacy project is a set of N-body simulations run at The University of Edinburgh using the Gadget-4 code (Springel et al., 2021). The halo-finding algorithm ROCKSTAR (Behroozi et al., 2013a) is used to generate halo catalogs. The main suite consists of a 1600 Mpc/h box with 2048^3 particles of mass $5.4 \times 10^{10} M_\odot$ which was run down to $z = 0$. There is a zoom simulation of a higher resolution box of size 700 Mpc/h and particle mass $6.8 \times 10^9 M_\odot$, named the Expanse. There are also a set of small 83 Mpc/h boxes which sample a range of density environments. For this work I use the Expanse as the basis for generating the mock catalog.

I employ a machine learning algorithm to populate the dark matter only Legacy simulation volume with accreting BHs. As I am predicting properties at a high redshift, and so do not have a large number of prior snapshots, I adopt the base model from Section 2.2. I train the model using the IllustrisTNG300 simulation volume, with the BH mass and accretion rate as the target variables. I then apply the trained model to the dark matter only Legacy $(1\text{Gpc})^3$ Expanse simulation. This gives me a catalog with approximately 40x the number of black holes that are present in the largest IllustrisTNG simulation. The luminosity of the black holes is calculated based on their accretion rate with the relation

$$L_{\rm BH} = \epsilon \dot{M}_{\rm BH} c^2 \tag{2.11}$$

where I set $\epsilon = 0.1$.

The robustness of this procedure must be verified before comparing to observations. This is necessary due to several differences between the training simulation and Legacy box I apply the model to. The halos in IllusrisTNG are identified via SubFind, while they are located using Rockstar in Legacy. As discussed in Section 1.2.1 this should not present an issue, especially as I am only considering the most massive halos in the simulation boxes, where agreement of halo finders is excellent. Another motivation for using the base model is that it means variations of the merger tree algorithms do not need to be considered. Another difference is the mass resolution of the two simulations. Again I argue that because I am

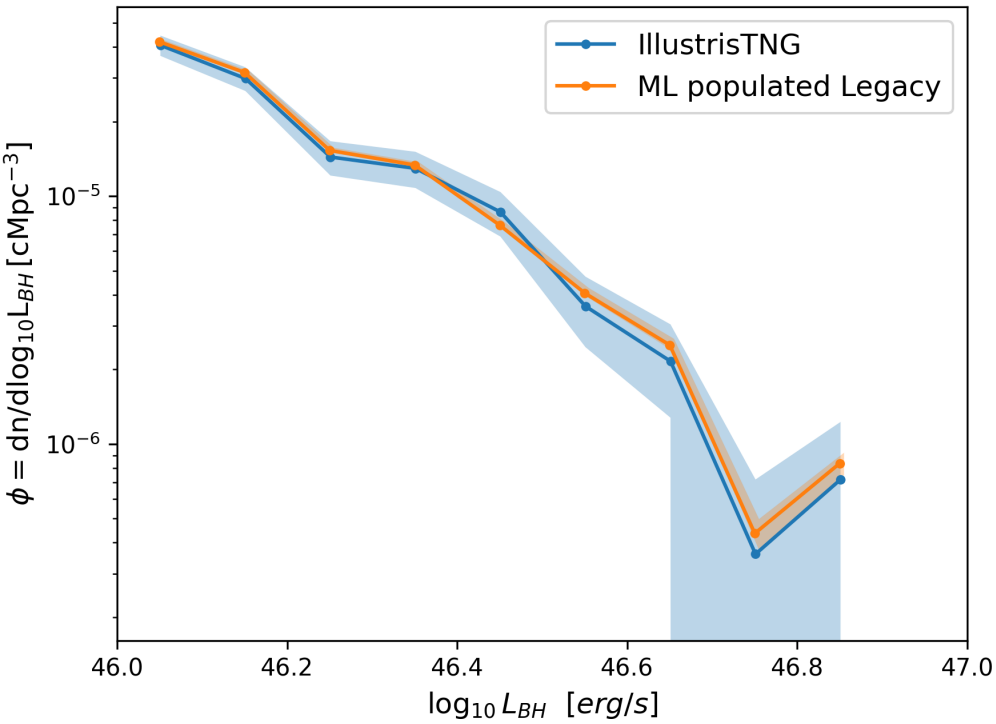

Figure 2.14 *The QLF derived for sources with bolometric luminosity $L > 10^{46}$ ergs^{-1} from the Illustris-TNG300 is shown in blue. The QLF of Legacy 1 Gpc Expanse populated using a model trained on Illustris-TNG300 is shown in orange. Both snapshots are at $z = 3.25$. There is excellent agreement showing that the method deployed here accurately reproduces the number statistics of SMBHs in the larger simulation box.*

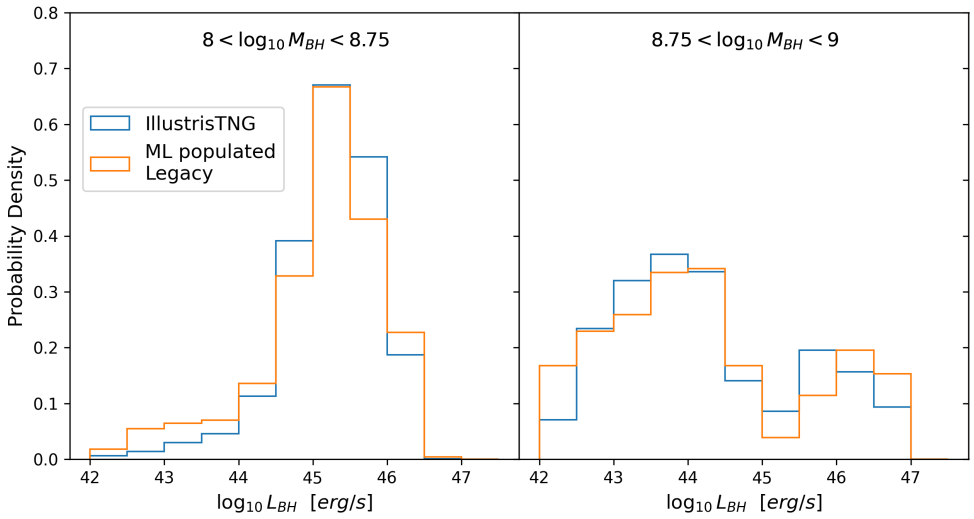

Figure 2.15 *Comparison of quasar population between the Illustris-TNG300 and the larger ML-populated Legacy 1 Gpc Expanse box. Sources with bolometric luminosity* $L > 10^{42}$ *erg*s^{-1} *from the* $z = 3.25$ *snapshot are shown in 2 mass bins.*

only considering the most massive halos this should not present an issue, as they are well resolved for both boxes. The final difference is a minor change to the cosmological parameters of the simulations since IllustrisTNG uses the results from Planck, whereas Legacy assumes the WMAP results.

In Figure 2.14, I plot the QLF derived from the training set, the Illustris-TNG300 box, and compare it with data from the populated Legacy $(1\mathrm{Gpc})^3$ Expanse box at $z = 3.25$. The blue and orange shaded region are estimates of the variation in the QLF based on Poisson statistics. Note that the QLF is in excellent agreement for the catalogs from the two simulations demonstrating that the deployed machine learning algorithm successfully enables me to expand simulation volumes while accurately capturing the occupation fraction of accreting BHs.

Further, in Figure 2.15, I split the accreting SMBH population in these two simulations into two bins based on their black hole mass. Again there is good agreement between the two datasets. Notable in these histograms is the bi-modal luminosity distribution that emerges at higher BH masses (as seen in the right panel of the plot).

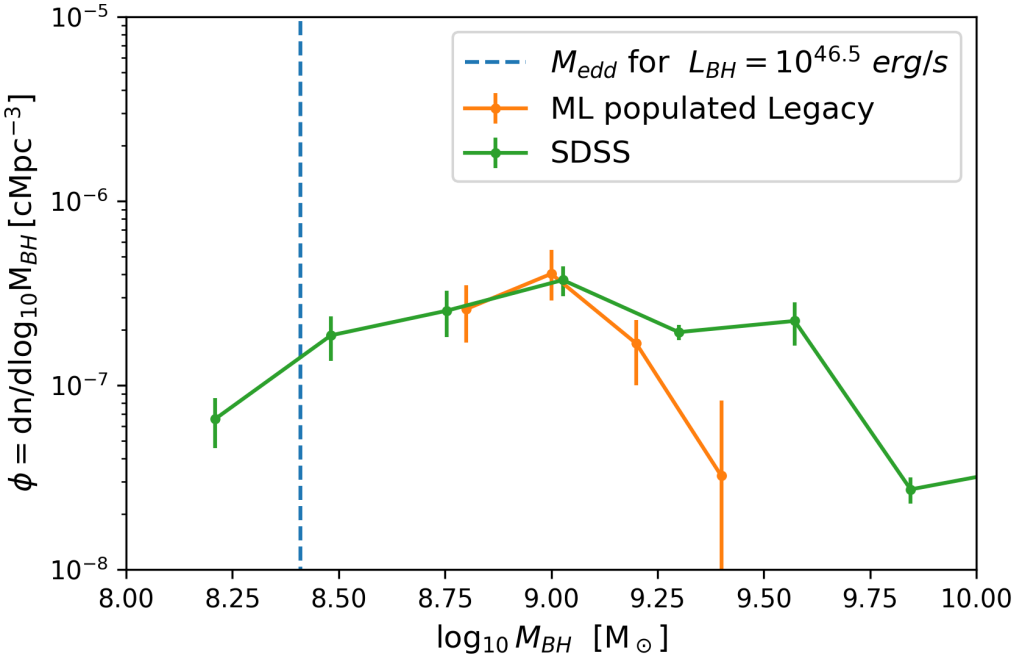

Figure 2.16 *Comparison of the BHMF from $z = 3 - 3.5$ SDSS quasars and the ML-populated Legacy 1 Gpc box ($z = 3.25$). The blue dashed line marks the BH mass that corresponds to the imposed luminosity cut of $L_{\rm bol} > 10^{46.5}\,{\rm erg\,s^{-1}}$. This cut implies that all BHs included in the census of the BHMF are accreting at sub-Eddington luminosities.*

2.4.3 Comparing mass functions

Confident that the machine learning model has populated the Legacy $(1{\rm Gpc})^3$ Expanse box in agreement with the Illustris-TNG300 simulation, now I compare the Legacy results to the observations. However, before doing so, I need to pay attention to the fact that the optically bright SDSS quasars collated in the observational data are a subset (Type I's) of the full population of AGN and therefore the accreting black hole population in the simulation needs to be scaled appropriately. Detailed multi-wavelength studies of AGN find that at $z \sim 3$, close to 90% of the sources are obscured Type II AGNs (quasars which are not optically bright), and their fraction decreases slightly and monotonically as function of X-ray luminosity in the range $L_X \sim 10^{42} - 10^{46}$ erg s^{-1}. Adopting an empirically derived fraction from Ananna et al. (2019), I scale the number of accreting BHs populated in the Legacy $(1{\rm Gpc})^3$ Expanse box to mimic the observed SDSS quasars at this epoch.

In Figure 2.16, I compare the black hole mass function (BHMF) of SDSS quasars

and the equivalent simulated population. The mass functions match extremely well at the turnover point, which corresponds to a bolometric luminosity cut of $L = 10^{46.5}$ erg s^{-1}. The dashed vertical line marks the location of BHs that would be accreting at the Eddington limit. Therefore it is clear that every accreting BH with mass $M_{\rm BH} > 10^{8.4} M_{\odot}$ in the simulation is accreting at sub-Eddington rates. At BH masses greater than $10^9\, M_{\odot}$, neither the amplitude nor the slope match well showing the BH population is underestimated in simulations. A surprising result is that at lower masses, $M_{\rm BH} < 10^{8.5} M_{\odot}$, the BHs that are detected by the SDSS $z \sim 3$ are completely missing from the simulations.

At the high mass end of the BHMF, this deficit of accreting SMBHs in the simulations could be partially attributed to a mismatch in sampling volumes between the SDSS and the simulation box, or as potentially arising as a limitation of the training set. When compared with the excellent agreement at the turnover point however, the discrepancy indicates that the simulation is underproducing the rarest, most luminous black holes.

The mismatch at lower masses, namely the lack of accreting BHs in mass range $M_{\rm BH} < 10^{8.5} M_{\odot}$ is glaring and strongly suggests that the adopted sub-grid accretion and feedback prescriptions in simulations are suspect. Note that accretion in the training set Illustris-TNG300 simulations is capped at the Eddington limit, so unsurprisingly no super-Eddington sources are predicted in these simulations and therefore none are found the Legacy $(1\text{Gpc})^3$ Expanse box either. This comparison clearly reveals that lower mass black holes in the simulation box are not luminous enough, as they are not accreting at high enough rates to survive the luminosity cut. This suggests that the sub-grid model for accretion adopted in Illustris-TNG300 does not accurately capture the accretion in observed quasars. Even though the Legacy $(1\text{Gpc})^3$ Expanse box encapsulates a larger range and diversity of formation and assembly histories for SMBHs, it replicates the issue with accretion rates found in the Illustris-TNG300. This work suggests that the sub-grid, multi-mode BH feedback ("quasar" mode at high accretion states, "wind" mode at low accretion states) adopted in Illustris-TNG300 is over-efficient at this epoch and appears to choke accretion onto BHs prematurely.

Note that current state-of-the-art simulations are capable of successfully reproducing observed properties at $z = 0$ such as the BHMF and the black hole-stellar mass relation. Therefore, the subgrid recipes for feedback from BHs, the modelling of gas accretion onto them, and prescriptions for star formation

reproduce integrated quantities well. However, as has been shown above, this application of machine learning permits a unique diagnosis of the mass assembly history over time of the BH population by focusing on a slice at $z \sim 3$. The mismatch found between SDSS quasars and their simulated counter-parts points to the fact that current sub-grid models of accretion and feedback do not reproduce the mass build-up over time accurately. More specifically, the probability distribution of accretion rates onto black holes with masses between 10^8 and $10^{8.75}$ $M_\odot$ does not match observations, and that this is not just a cosmic variance issue.

Weinberger et al. (2018) compared the TNG black hole population with observed QLFs at a range of redshifts. Unlike the approach presented here, they did not utilize machine learning methods, they instead directly employed data from the simulation box. They concluded that the simulation overpredicted the QLF, which appears to contradict the results presented here. However, several differences in the analyses should be noted. The primary distinction is that I consider a much higher luminosity cut. This is made possible by the large volume which means there are a large number of extremely bright quasars. For objects with $L = 10^{46.5}$ Weinberger et al. (2018) does see reasonable agreement with observations. It is only at lower luminosities that they find simulation predictions diverging from observations. There are also some differences in modelling. When calculating luminosities I use $\epsilon = 0.1$, where as Weinberger et al. (2018) use a variable model of radiative efficiency at low Eddington rates, and $\epsilon = 0.2$ for high accretion rates. Discrepancies are also likely to be introduced when converting observed black hole luminosities to masses. At the high mass end Weinberger et al. (2018) sees a steep drop in the number of objects in the simulation when compared with the shape of the observed QLF. This is in agreement with what is seen in Figure 2.16 for the highest mass black holes.

Habouzit et al. (2022) analyse and compare results on BH growth and assembly across redshift in several large-scale independent cosmological simulations like the Illustris-TNG100, Illustris-TNG300, Horizon-AGN, EAGLE, and SIMBA suites. They show that while all of them predict a similar BHMF and relation between BH mass and host galaxy stellar mass M_* at $z = 0$ in agreement with local observations, their predictions disagree at higher redshifts. For instance, there is much disagreement on whether BHs at $z = 6$ are overmassive or undermassive at fixed host galaxy stellar mass with respect to the $z = 0$ $M_{\rm BH} - M_*$ relation. This supports the argument that the AGN feedback and BH growth prescriptions

adopted in simulations do not get the growth build up over time correct and that a lot of development is still required for the theoretical subgrid models, specifically, the models adopted for gas accretion and feedback. The results presented above suggest this more strongly than Habouzit et al. (2022), given that I am considering redshifts closer to $z = 0$, the point at which the simulations are calibrated, and that I am are comparing directly with observations rather than just between simulations. On the observational side there is a need for the section of the QLF populated by lower luminosity quasar population to be filled in to help fully understand BH growth and evolution.

2.4.4 Comparing correlation functions

In order to compare the spatial distribution of quasars a statistical tool is needed that quantifies the probability that galaxies are close to each other. For this I use the correlation function $\xi(r)$. Given a random galaxy in a location, the correlation function describes the probability that another galaxy will be found within a given distance (Peebles, 1980). For a volume dV a distance r from a randomly picked galaxy, the probability there will be a galaxy in that volume is given by

$$dP = n[1 + \xi(r)]dV \tag{2.12}$$

where n is the mean number density of galaxies over the whole volume. To calculate the correlation function first calculate how many galaxies are separated by a distance less than r, and name this quantity $DD(r)$. The same number of points are then placed randomly in the same volume, and the number at a distance of less than r is also counted. This value is $RR(r)$. The correlation function is then given by

$$\xi(r) = \frac{DD(r)}{RR(r)} - 1 \tag{2.13}$$

In practice the Landy-Szalay estimator Landy & Szalay (1993) is used. It helps to account for the effects of survey geometry and non-periodic boundaries. It is

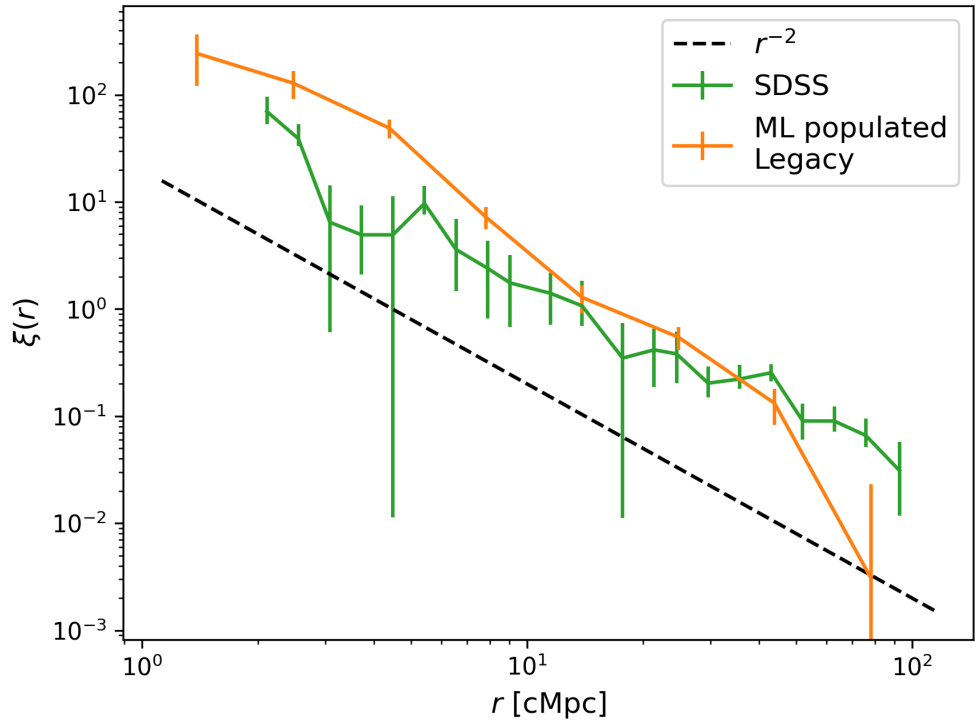

Figure 2.17 *Comparison of the clustering of quasars from SDSS and the ML-populated Legacy 1 Gpc Expanse box at $z \sim 3$. The SDSS quasars plotted here are also part of the BOSS survey, with clustering measurements reported in Eftekharzadeh et al. (2015)*

given by

$$\xi_{LS}(r) = \left(\frac{N_R}{N_D}\right)^2 \frac{DD(r)}{RR(r)} - 2\frac{N_R}{N_D}\frac{DR(r)}{RR(r)} + 1 \tag{2.14}$$

where $DR(r)$ is the number of galaxies and random pairs separated by distance r, N_D is the number of galaxies and N_R is the number of randomly distributed data points in the same volume.

One of the main advantages of the large box mock galaxy catalog is that it allows for statistics such as the correlation function to be calculated which would not be possible with the smaller IllustrisTNG volume. This offers the ability to test whether the learned mapping of SMBHs to host halos is able to reproduce the observed clustering properties of the quasar population.

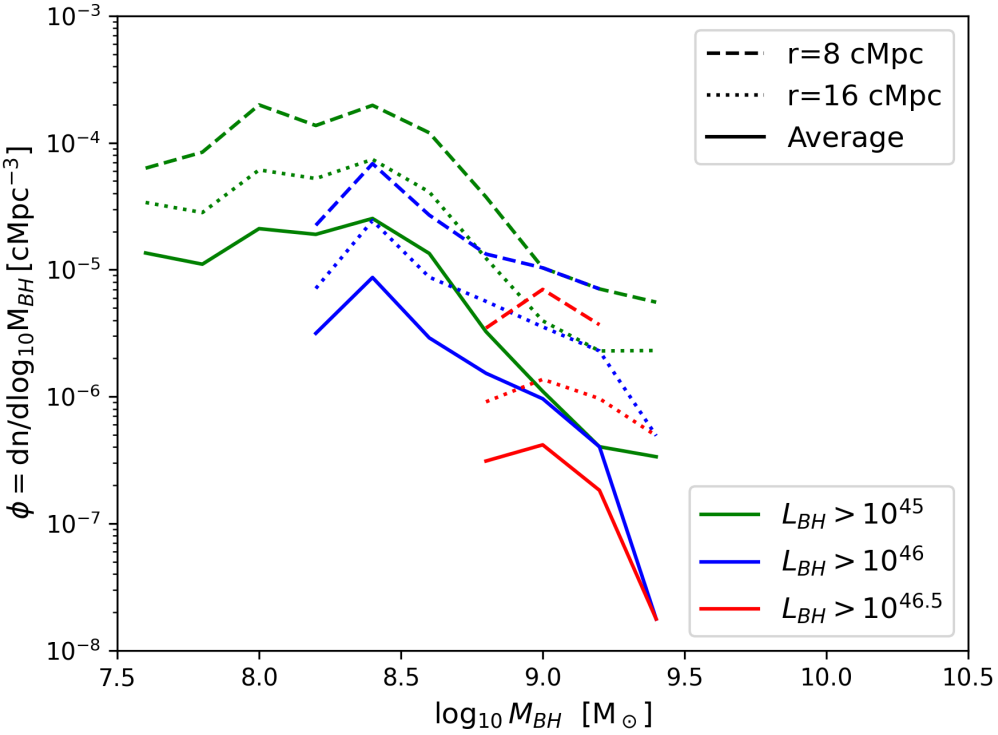

Figure 2.18 *New neighborhood statistic at $z \sim 3$ - comparison of the BHMF of the accreting SMBH population that lies within 8, 16 and 40 comoving Mpc radius annuli around $10^9\,M_\odot$ quasars in the Legacy 1 Gpc Expanse box.*

One of the fundamental science questions that large volume machine learning generated mock catalogs helps to address is how best to devise future survey strategies that will uncover the lower luminosity, more characteristic, accreting BH population in the modest and high-redshift Universe. In Figure 2.17, I show that the clustering properties of optical SDSS quasars in the redshift range of $3 < z < 3.5$, are generally reproduced by the accreting SMBH population at the corresponding redshift snapshot ($z = 3.25$) in the Legacy $(1\text{Gpc})^3$ Expanse box. This suggests that simulations capture the statistics of the association of SMBHs with their parent dark matter halos inferred from clustering studies of SDSS quasars. Therefore, the data is well suited to infer an observational survey strategy that will help with uncovering the fainter quasar population. To do this, I construct a novel neighborhood statistic and evaluate its robustness using the simulation data. The proposed statistic is derived by enumerating the BHMF found in the neighborhood of a bright quasar powered by a $10^9\,M_\odot$ SMBH in spheres of varying radius, 8 and 16 comoving Mpc respectively. As the radial region under consideration changes, the mass function of BHs within that radius changes. In Figure 2.18, I plot this neighborhood occupation statistic for quasars from the ML-populated Legacy $(1\text{Gpc})^3$ Expanse box. Included in this plot is the mean BHMF, shown with a solid line, derived from averaging over the entire box. It is clear to see that there is a preferential excess of lower luminosity sources well above the mean value of the neighborhood statistic in the vicinity of a bright quasar. A strong excess of sources is detected even when considering sources with a range of luminosity cuts. This suggests that an optimal observational strategy for detecting the hitherto undetected population of lower luminosity quasars would be to survey regions around the most luminous sources currently detected in surveys. Deeper pointings in regions of around the rare, brightest quasars in found in SDSS will permit filling into the lower luminosity end of the observed QLF. This filling in of the QLF will finally provide even tighter constraints on theoretical models.

2.5 Conclusions and future work

In this chapter I have introduced a new method of multi-epoch machine learning for generating mock galaxy catalogs. I have then used such a catalog to compare with observations. My conclusions related to the multi-epoch technique can be summarized as follows.

- I introduce a novel method of predicting the baryonic properties of subhalos from dark matter only simulations using machine learning. My model takes subhalo properties from a wide range of redshifts as input, and can be trained on any simulation with merger trees available.

- When compared with a baseline model that only uses $z = 0$ input features, the new model yields significantly more accurate predictions. It also outperforms a model which only uses the mass history of subhalos. Therefore future work which predicts baryonic properties should include a variety of subhalo properties taken over a range of redshifts as their input.

- I use a normalized version of mean squared error as my loss function, which allows me to determine which output properties are most difficult to predict.

- Using decision tree based algorithms allows me to determine the relative importance of each input feature. Figure 2.8 shows how the feature importance varies depending on the output feature being predicted. This allows me to infer information about how the different baryonic properties of a subhalo are determined, especially the redshift which is most important.

- My feature importance plots and ratios of $I_{\rm nur}/I_{\rm nat}$ show that for the IllustrisTNG simulations nurture is more important than nature in determining the properties of a galaxy

- I confirm my feature importance are not an artefact of the ERT algorithm used in the work by examining how the MSE varies depending on what snapshots are passed to the model. These results hold for machine learning algorithms not based on decision trees.

Since the work presented in this chapter was carried out there have been a number of further publications focusing on machine learning the galaxy halo connection. These include Hausen et al. (2022) who employed explainable boosting machines and found that environmental features only provided significant improvement in the predictions for a small fraction of the total galaxy population. de Andres et al. (2022) considered all the possible $z = 0$ halo catalog values as input features for their model. Rodrigues et al. (2023) predicted joint probability distributions directly rather than combining predictions for individual properties. Two works have since also included merger tree history as an input to their models. Chittenden & Tojeiro (2023) choose recurrent neural networks as their algorithm as they are well suited to time series data. They predict the SFR

for every snapshot in the simulation. Jespersen et al. (2022) apply graph neural networks to the full merger history of halos. This continually developing body of work highlights how efficiently mapping baryonic properties onto dark matter still remains a major challenge, and that there is likely to be considerable more interest in the area over the coming years.

An interesting possible avenue where machine learning could be applied is to train a model on a physical process within a high resolution simulation. The trained model could then be applied as a subgrid model within a larger box low resolution simulation. A proof of concept of this approach has been demonstrated in a number of works. Wells & Norman (2021) used 3D convolutional neural networks to predict what regions would host primordial star formation. Grassi et al. (2022) employed autoencoders to reduce the dimensionality of a chemical network and gained an increase in speed by a factor of 65. Most recently Hirashima et al. (2023) used a deep learning model to predict what particles in an SPH simulation would require small timesteps. An area where this approach could be profitable would be in predicting the accretion rate of SMBHs.

The results in this Chapter from comparing a mock catalog with observations can be summarized as follows.

- I train a machine learning model using the IllustrisTNG300 simulations to predict the mass and accretion rate of SMBHs based on their host halo properties. I apply this model to the Legacy N-body simulations, which results in a mock catalog with a volume of $(1\text{Gpc})^3$.

- There is good agreement in both the mass and luminosity distribution between the data from IllustrisTNG and the ML-populated Legacy, indicating that the model has successfully learned an accurate mapping of the SMBH-halo connection.

- I compare the BHMF from the Legacy simulation with observed data at $z \sim 3$. The mass functions match extremely well at the turnover point, but above and below this point the simulated data is considerably lower. This indicates a lack of accretion onto SMBHs at this epoch.

- Using the two-point correlation function I compare the spatial distribution of the simulated and observed data. Good agreement is found. Given this success I plot the number of faint black holes that can be expected

to be found close to the brightest quasars, which is useful for informing observational strategies.

This is the first instance of a direct comparison between machine learning generated mock catalogs and observational data. In addition to permitting the analysis of large, complex data-sets, I have shown how machine learning techniques can help interrogate key theoretical model assumptions.

The next generation of observations means that it will be possible to carry out these kind of comparisons at higher redshifts. Indeed, there is currently observational data available at $z = 4-5$ that could have been used for comparison. However, due to limitations in the training data, I was unable to generate mock catalogs for $z \geq 4$. To solve this issue models could be trained on zoom simulations, as presented in Lovell et al. (2022) and de Andres et al. (2022).

Chapter 3

Identifying physical drivers of galaxy evolution

The material in this chapter was originally published in McGibbon & Khochfar (2023). The code used to produce the results is available on GitHub.

3.1 Introduction

As discussed in Chapters 1 and 2, hydrodynamical cosmological simulations have become key tools for helping to understand both cosmology and galaxy formation. Different simulations make use of various methods for modelling gas, including particle based methods (e.g. Springel, 2005), and grid based approaches, using both structured and unstructured meshes (e.g. Springel, 2010; Morton et al., 2023), with some codes utilising adaptive mesh refinement (e.g. Bryan et al., 2014). Each comes with its own strengths and weakness, and numerical effects from the implementations of the various methods can affect the hydrodynamics of the gas in different ways. More variation in simulations comes from the fact that many of the relevant physical processes that need to be modelled occur below the typical resolution limits of cosmological simulations. Therefore various "sub-grid physics" prescriptions are employed to model them. There are many different implementations of different subgrid models, and each model tends to have a number of tunable parameters. In this chapter I focus on implementations of stellar and black hole feedback, as discussed in Section 1.2.2.

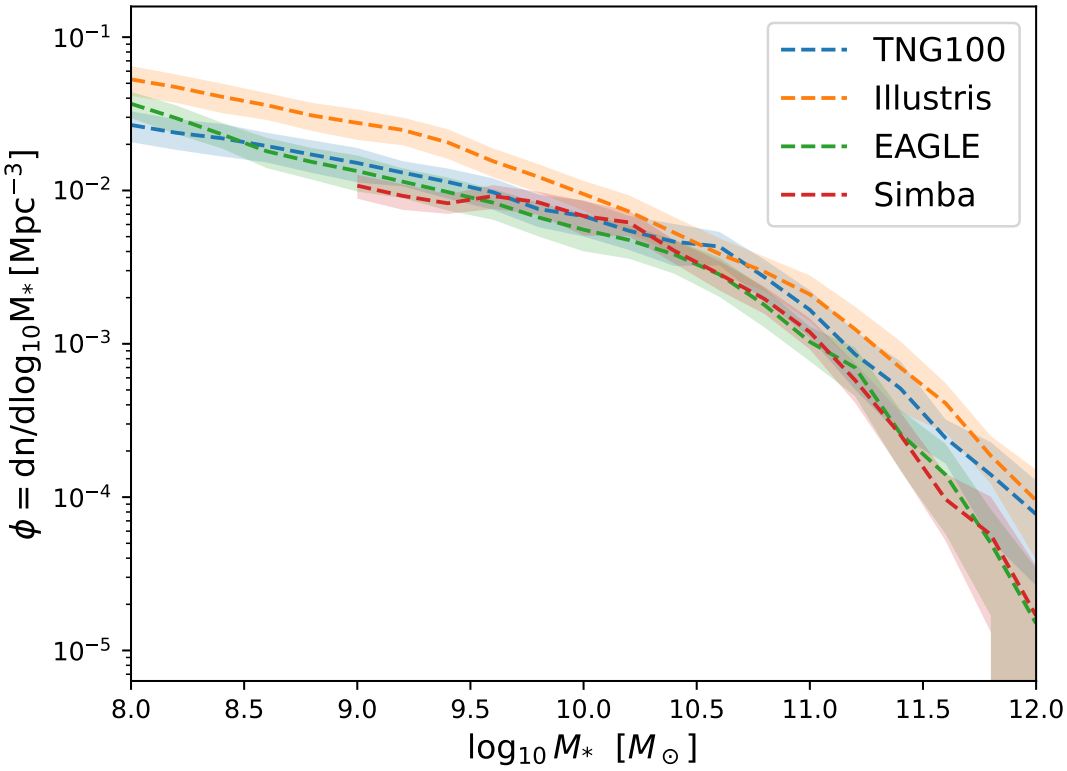

Figure 3.1 *Stellar mass functions from the Illustris, TNG100, EAGLE, and Simba simulations. The shaded area corresponds to the spread computed from jackknife re-sampling over eight simulation sub-octants.*

Even with these variations in modelling, subgrid models from different simulations can be tuned to reproduce observables. An example of this is given in Figure 3.1, which shows the $z = 0$ stellar mass function from a number of simulations. The results from Simba and EAGLE agree over the full mass range of galaxies they resolve. IllustrisTNG is also consistent with EAGLE, apart from at the highest masses. The original Illustris run does not show as good agreement, but displays the same general behaviour as the other simulations. Despite the similar stellar populations at $z = 0$, the way in which galaxies build up their stellar mass can differ greatly between simulations, which is what the technique presented in this chapter is able to distinguish.

As cosmological simulations output large volumes of data, machine learning can be a useful technique for gaining understanding of the processes occurring within the simulation. These kinds of methods can be used for obtaining insights into both dark-matter-only and hydrodynamical simulations. I highlight a number of recent examples below.

Using N-body simulations Lucie-Smith et al. (2019) trained a decision-tree based model to predict the final mass of halos by using the simulation initial conditions as input, and examined their model to determine the importance of the tidal shear field in establishing halo mass. In Lucie-Smith et al. (2022) the authors predicted the density profiles of halos, and included the mass accretion history of the halo as model input. They showed how the feature importance from the model could be interpreted in terms of physical timescales.

By training neural networks on hydrodynamical cosmological simulations, and then applying symbolic regression, Shao et al. (2022) were able to recover a version of the virial theorem. Shi et al. (2022) used random forests to learn which galaxy properties are most useful for determining the fraction of accreted stellar mass, and compared the feature importance values for high mass and low mass galaxy samples. Eisert et al. (2022) showed that by using invertible neural networks it is possible to gain information about the properties that are most predictive of the time of a galaxies' last major merger. Wadekar et al. (2020) used saliency maps from a convolutional neural network to show how the HI content of halos exhibits a strong dependence on their local environment. Villanueva-Domingo et al. (2021a) trained a graph neural network to predict the mass of a halo based on its host galaxies, and examined which galaxy properties were the most predictive.

These types of machine learning methods are not only restricted for use with simulations, but can also be applied directly to observations to advance our understanding of galaxy formation processes. Bluck et al. (2022) and Piotrowska et al. (2022) both used the feature importance from trained random forest models to examine the most important parameters for predicting whether a galaxy is quenched. Curti et al. (2022) investigated which physical properties are most predictive for determining the position of galaxies on BPT diagrams by using neural networks and random forests, and Holwerda et al. (2022) used self-organizing maps to help analyse galaxy bimodalities.

In this chapter I expand on the method introduced in Chapter 2. I now use baryonic features as input to the model, rather than just halo properties, and study the resulting feature importance plots. The purpose of this is to gain insight into the different formation processes leading to distinct populations within a given simulation and also that occur within different simulations. Using baryonic features as inputs allows me to more easily examine the impact of feedback than if I only consider halo properties.

The remainder of this chapter is organized as follows. In Section 3.2 I give an overview of the simulations used in this work, focusing on the differences in their feedback subgrid model implementations. Section 3.3 contains tests of the robustness of my method and shows how it can highlight differences in galaxy populations when applied to the IllustrisTNG simulation suite. In Section 3.4 I look at the insights that can be gained when comparing different simulations. I apply my method to the CAMELS simulation suite in Section 3.5, and show how the feature importance changes when subgrid model parameters are varied. In Section 3.6 I discuss how my results relate to existing literature. I summarize my findings and consider possible future work in Section 3.7.

3.2 Methods

3.2.1 Simulations

In this subsection I summarise the hydrodynamical cosmological simulations used in this work. Each simulation includes all significant physical processes required to track the evolution of dark matter, cosmic gas, luminous stars, and supermassive blackholes (SMBHs) from high redshifts ($z \sim 100$) to the present day $z = 0$. I focus on the different subgrid implementations of supernova and black hole feedback. I quote the dark matter particle mass, $m_{\rm DM}$, as a measure of the resolution of the simulation rather than attempting to directly compare the mass resolution of different baryonic elements. For all simulations the halos are first located using the FOF algorithm (Davis et al., 1985), then substructure is identified using the SubFind subhalo finder (Springel et al., 2001). To avoid poorly resolved objects I only consider subhalos with $10^{8.5} < M_* < 10^{12}$.

Illustris

Illustris[1] (Vogelsberger et al., 2014a,b; Genel et al., 2014; Sijacki et al., 2015) is run with the moving mesh code Arepo (Springel, 2010). The simulation adopts a WMAP-9 (Bennett et al., 2013) consistent cosmology, and merger trees are constructed using the SubLink algorithm (Rodriguez-Gomez et al., 2015). The box size is $(75\ h^{-1}\mathrm{Mpc})^3$, with $m_{\rm DM} = 6.3 \times 10^6 M_\odot$.

[1] illustris-project.org/

Feedback associated with star formation is assumed to drive galactic scale outflows. The generated winds have a velocity scaled to the local dark matter velocity dispersion. The mass loading factor of the wind is calculated using the desired wind speed and available supernova energy. The direction of the wind is determined by the parent gas cell in such a way that wind particles are ejected preferentially along the rotation axis of spinning objects. SMBH feedback occurs in two different modes. If the Eddington ratio is below 0.05, a radio-mode model injects highly bursty thermal energy into large, ~50 kpc 'bubbles' which are displaced away from the central galaxy. Above this accretion rate, a quasar-mode model injects thermal energy into the immediately surrounding gas.

IllustrisTNG

The IllustrisTNG suite[2] (Springel et al., 2018; Pillepich et al., 2018b; Naiman et al., 2018; Nelson et al., 2018; Marinacci et al., 2018) is an update to Illustris simulation. It is also run using the Arepo code (Springel, 2010), but a notable addition is the inclusion of magnetic fields. Cosmological parameters are set to the Planck 2015 values (Planck Collaboration et al., 2016). There are 3 different box sizes, each with its own resolutions. Comparable with the original Illustris simulation, TNG100 has a box size of $(75\ h^{-1}\mathrm{Mpc})^3$ and dark matter particles have $m_{\mathrm{DM}} = 7.5 \times 10^6 M_\odot$. TNG100 uses the same initial conditions as Illustris, although they have been adjusted for the updated cosmology. TNG300 has a box size of $(205\ h^{-1}\mathrm{Mpc})^3$ and $m_{\mathrm{DM}} = 5.9 \times 10^7 M_\odot$, while TNG50 has a box size of $(35\ h^{-1}\mathrm{Mpc})^3$ and $m_{\mathrm{DM}} = 4.5 \times 10^5 M_\odot$. Two sets of merger trees are available: one generated using Sublink (Rodriguez-Gomez et al., 2015), the second created by LHaloTree (Springel et al., 2005).

The TNG model for stellar feedback is based on the Illustris model, with some modifications. Winds are now ejected isotropically, although will still naturally propagate along the direction of least resistance. The wind velocity is now redshift-dependent, and a wind velocity floor is also introduced. The result of these changes is that stellar feedback in the TNG is more effective at suppressing star formation. The TNG also features two modes of SMBH feedback, with the mode being dependent on whether the Eddington ratio is above a critical value. For the TNG this critical value increases with the mass of the black hole. The high accretion thermal mode is the same as Illustris, but for low accretion

[2] tng-project.org/

rates a kinetic mode is used which adds momentum to neighbouring gas cells. For more details regarding the specific implementation of the IllustisTNG simulations, including all relevant subgrid models, I refer the reader to Weinberger et al. (2017) and Pillepich et al. (2018a).

EAGLE

EAGLE[3] (Schaye et al., 2015; McAlpine et al., 2016) is a suite of cosmological simulations run with the smoothed particle hydrodynamics code GADGET-3 (Springel, 2005) using the Anarchy scheme. It adopts a Planck 2013 cosmology (Planck Collaboration et al., 2014a). Merger trees are built using the D-Trees algorithm (Jiang et al., 2014). The fiducial EAGLE simulation, named Ref-L100N1504, has a box size of (100 Mpc)^3, and dark matter particles have $m_{\rm DM} = 9.7 \times 10^6 M_\odot$.

Supernova feedback in EAGLE is implemented by injecting thermal energy in a stochastic manner to nearby particles. The energy injected per unit stellar mass varies based on the metallicity and density of the interstellar medium (ISM). SMBH feedback in EAGLE is achieved using a single mode of feedback that operates at any Eddington ratio, in contrast to the dual modes of Illustris, TNG, and Simba. Feedback energy is stored until it is sufficient to heat the surrounding particles by $\Delta T = 10^{7.5}$K and then is stochastically injected as thermal energy. This 'pulsed' nature of the thermal feedback prevents the energy being immediately radiated away and offsets cooling. This makes it more efficient at quenching the galaxy than the corresponding thermal mode in Illustris and TNG.

Alongside the fiducial EAGLE simulation, I consider some variants run with the same resolution, but differing subgrid models. I utilise the FBconst and FBZ simulations, which are described in Crain et al. (2015). Both of these runs have been calibrated to match the observed stellar mass function. The FBconst model injects into the ISM a fixed amount of energy per unit stellar mass formed. For the FBZ simulation the energy associated with supernova feedback depends on the ISM metallicity, but unlike the fiducial model the energy is independent of gas density. I also examine the NoAGN run, which has the same subgrid models as the reference EAGLE simulation, but does not include black holes.

[3] icc.dur.ac.uk/Eagle; eagle.strw.leidenuniv.nl

Table 3.1 *The parameters that are varied for each run of the CAMELS simulations.* ***Min*** *and* ***Max*** *give the minimum and maximum values that the parameters take on.* ***log scale*** *indicates whether the values are sampled linearly, or if they are varied with a logarithmic scale.* ***IllustrisTNG effect*** *(****Simba effect****) gives information about how the value of the parameter changes the feedback models within IllustrisTNG (Simba). For more details about the parameters see Villaescusa-Navarro et al. (2022).*

Parameter	**Min**	**Max**	**log scale**	**IllustrisTNG effect**	**Simba effect**
Ω_m	0.1	0.5	No	Initial conditions	Initial conditions
σ_8	0.6	1	No	Initial conditions	Initial conditions
A_{SN1}	0.25	4	Yes	Energy output per unit star formation	Wind mass outflow rate per unit star formation
A_{AGN1}	0.25	4	Yes	Prefactor for power injected in kinetic mode	Prefactor for momentum flux of outflows
A_{SN2}	0.5	2	Yes	Speed of galactic winds	Speed of galactic winds
A_{AGN2}	0.5	2	Yes	Burstiness and temperature	Speed of jets

CAMELS simulations

The CAMELS project[4] (Villaescusa-Navarro et al., 2021, 2022) contains two different suites of state-of-the-art hydrodynamic simulations.

The simulations in the first suite have been run with the Arepo code (Springel, 2010) and employ the same subgrid physics model as the IllustrisTNG simulations. See Section 3.2.1 for more details on the subgrid models for this suite.

The simulations in the second suite have been run with the GIZMO code (Hopkins, 2015) and employ the same subgrid physics model as the Simba simulation (Davé et al., 2019), which built on its precursor MUFASA (Davé et al., 2016) with the addition of supermassive black hole growth and feedback (Anglés-Alcázar et al., 2017). Star formation in the Simba model drives winds similar to those found in the TNG. The mass loading factor is scaled by redshift, and is constant for low mass galaxies. The Simba model has two SMBH kinetic feedback modes. At high Eddington ratios SMBHs drive multi-phase winds at velocities of $\sim 10^3$ km s^{-1}. At low Eddington ratios gas is heated to the virial temperature of the halo and ejected at velocities of $\sim 10^4$ km s^{-1}. For both of these modes the ejection is bipolar and parallel to the angular momentum vector of the SMBH accretion disc. X-ray feedback from SMBHs is also implemented, but it has a minimal effect on the galaxy mass function.

All simulations from both suites follow the evolution of 2×256^3 dark matter plus fluid elements in a periodic comoving volume of $(25\ h^{-1}\mathrm{Mpc})^3$. All simulations share the value of the following cosmological parameters: $\Omega_\mathrm{b} = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $\sum m_\nu = 0.0$ eV, $w = -1$. However, each simulation has a different value of Ω_m and σ_8. The simulations also vary the values of four astrophysical parameters that control the efficiency of supernova and SMBH feedback: A_SN1, A_SN2, A_AGN1, and A_AGN2. Details about the effect of these parameters and the range of values they can take is given in Table 3.1.

The simulations with the different parameters are arranged into 4 sets. In the LH set, which contains 1000 simulations for each code, values are arranged on a latin-hypercube, and each simulation has a different random seed for initial conditions. I note that the latin-hypercubes of the IllustrisTNG and Simba simulations are different, i.e. there is no correspondence between simulations among the two sets. The 1P set contains simulations in which a single parameter varies, and all other

[4] camels.readthedocs.io/

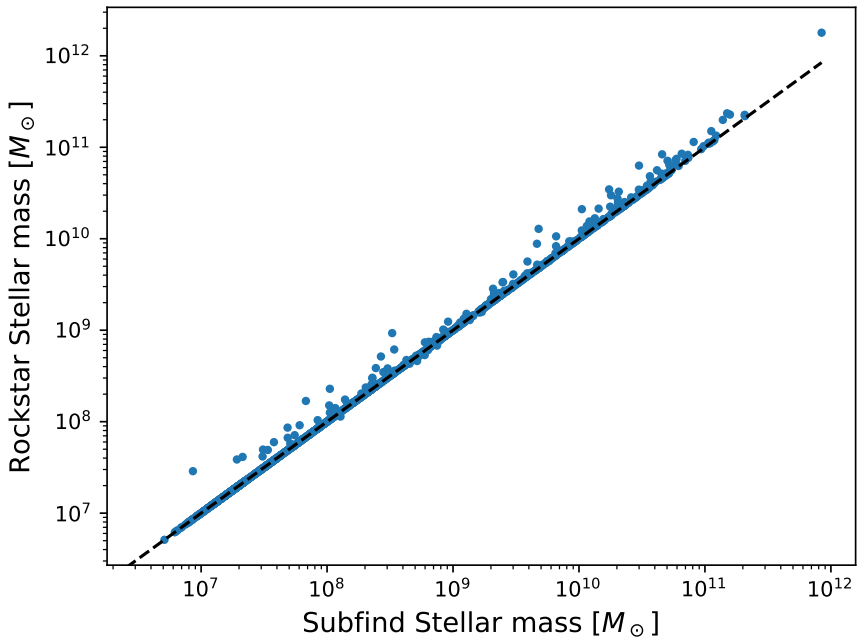

Figure 3.2 *Stellar mass from* ROCKSTAR *and* SUBFIND *catalogues of all the* $z = 0$ *matched subhalos from the IllustrisTNG LH0 simulation.*

parameters are kept fixed at their fiducial value. There are 11 simulations for each parameter, meaning there are 61 simulations for each code. The same initial conditions random seed is used for all the 1P simulations. The CAMELS project also contains a cosmic variance and extreme set, but I do not use either of these in my work.

Matching Rockstar and SubFind catalogs

The merger trees available in the CAMELS project are created using the CONSISTENTTREES code (Behroozi et al., 2013b) which is built on top of halos located using the ROCKSTAR algorithm (Behroozi et al., 2013a). However the ROCKSTAR halo catalogs do not contain all the galaxy properties I require for this work, such as star formation rate (SFR). Therefore I match the ROCKSTAR and SUBFIND halos so that I have access to the data that I require. I use the method described in Gómez et al. (2022), but allow halos to be within 3x half mass radius to increase my sample size.

I take the positions of subhalos from both the ROCKSTAR and SUBFIND catalogues. For each of the SUBFIND halos I locate any ROCKSTAR halos that satisfy the following criteria:

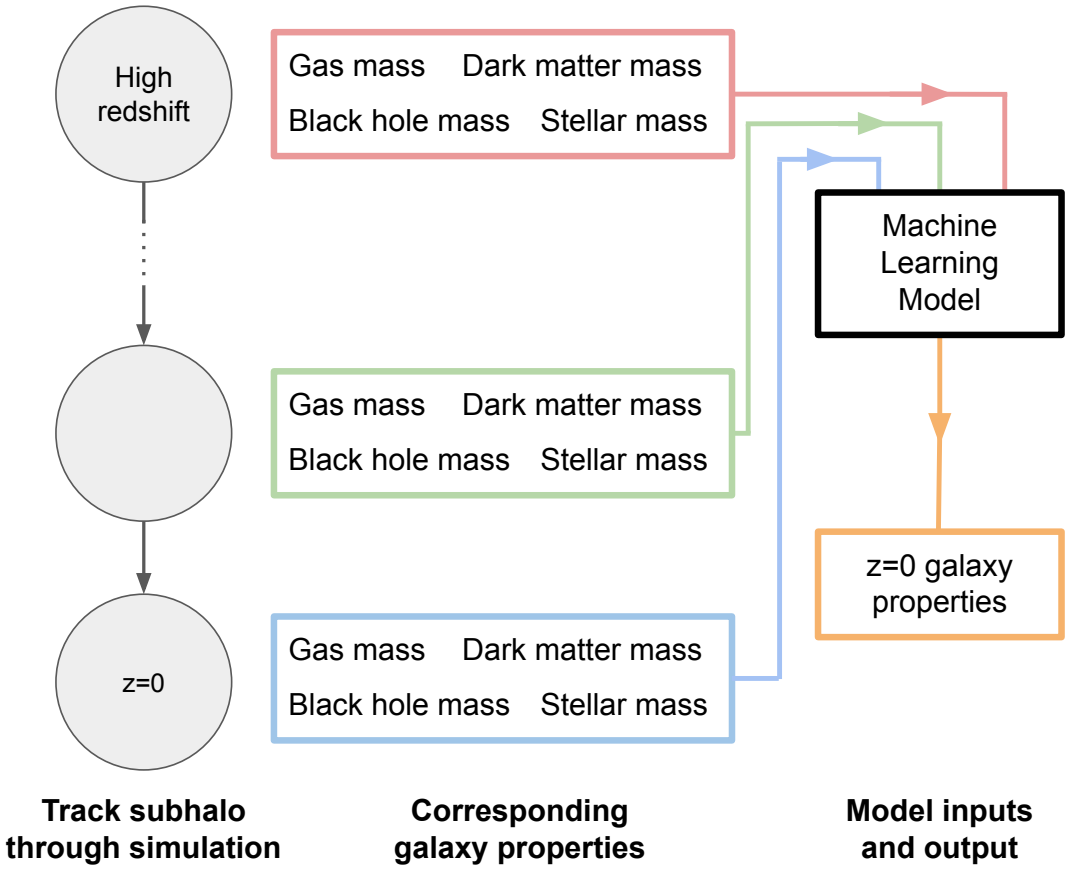

Figure 3.3 *Summary of the method used in this work. The machine learning model takes in the four input features of the base model, but from a range of snapshots, not just redshift zero. The output for all models is the subhalo's baryonic properties at redshift zero.*

- Position within 3x the half mass radius of the SubFind subhalo.

- Mass is within a factor 3 of the SubFind subhalo.

If there are multiple Rockstar halos which fulfil these criteria I pick the closest one. I repeat this process for every snapshot in the simulation. In Figure 3.2 I show the Rockstar and SubFind stellar mass of matched halos. Despite the larger minimum matching distance than Gómez et al. (2022), which allows me to gain a larger sample size, the scatter in the stellar mass as shown in the Figure does not increase significantly.

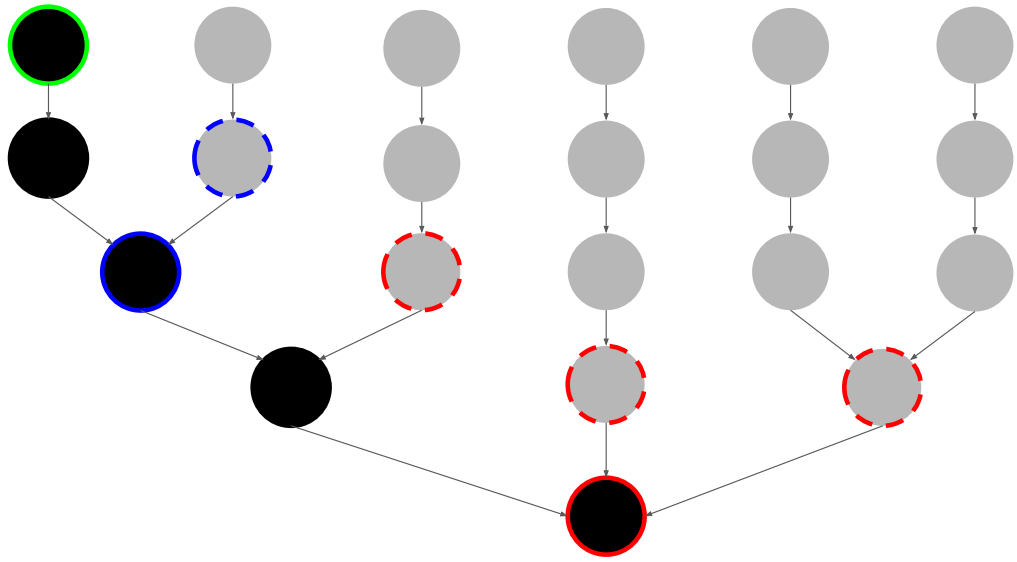

Figure 3.4 *How subhalos that have merged are passed as input to the model. The black circles indicate the main progenitor branch of the merger tree, the grey circles indicates subhalos that merge with the main subhalo. Three snapshots are considered as input here, the green circle, blue circle, and red circle. The merger input feature is defined as the sum of the properties of the subhalos that have merged since the last snapshot that was used as input. This means that for the blue snapshot the single subhalo with the blue dashed line is used as the merger feature. For the red snapshot the merger feature is equal to the sum of the masses of the three subhalos with a red dashed outline.*

3.2.2 Input and output features

As in Chapter 2, the inputs to my model are properties of a subhalo taken from a wide range of redshifts. As decision trees are invariant to the scaling of the input features, therefore I do not scale the input features in any way, despite the fact that their values span multiple orders of magnitude. As discussed in Chapter 2 I log the output features to prevent high mass galaxies being given a significantly higher weight than low mass ones.

I do not consider every snapshot from the simulations as model input since this results in too large correlations in my input features for the feature importance to work effectively. Therefore I use properties from every d^{th} snapshot as model input. I choose d for each simulation such that I get 10 snapshots approximately evenly spaced in time. For each of these input snapshots I use the value of a number of the galaxy properties as an input, where i denotes the property (gas

mass, dm mass, bh mass, or stellar mass). A summary of this part of my method is shown in Figure 3.3.

In Chapter 2 I only considered the main progenitor branch as input to my model. In this work I consider the impact of other branches that merge. I define the merger feature at the s^{th} snapshot for the i^{th} property as

$$M_s^i = \sum_{t=s-d}^{t=s} m_t^i \tag{3.1}$$

where m_t^i is the amount of mass of property i that merged into the main progenitor branch at snapshot t. Thus I am summing the mass of all the subhalos that merge into the main progenitor branch between the input snapshots. This method is not able to distinguish between a large number of minor mergers and a single major merger, but it still captures information about the importance of discrete vs smooth accretion for the halo's evolution. A schematic of this process is shown in Figure 3.4.

In this work I show the results from predicting four different output features, although this method could be used to gain information about any galaxy property. I predict the $z = 0$ stellar and gas mass, which are given by the total mass of all stellar/gas particles/cells identified by SubFind as bound to the subhalo. The galaxy SFR is defined as the sum of the individual star formation rates of all gas elements in the subhalo. The stellar metallicity is given by the mass-weighted average metallicity of the star particles.

3.2.3 Machine learning methods

For this chapter I continue to use extremely randomised tree ensembles as the algorithm to build the regressor models. A full description is given in Section 2.2.3. However, I now normalize the feature importance such that the maximum value has a value equal to one, rather than all values summing to one. This eases comparison between simulations.

Principal Component Analysis (PCA)

In order to help visualize the results of applying my method to the CAMELS simulations, I make use of PCA (e.g. Shlens, 2014; Jolliffe & Cadima, 2016) to

reduce the dimensionality of the data. PCA is an unsupervised statistical learning algorithm. It takes in a data set which has linearly correlated variables and returns a new set of uncorrelated variables known as principal components. The principal components returned have the property that the first principal component has the largest variance.

Consider a set of n data vectors, $\{\mathbf{x}^i\}$, where each vector contains m features. The basis that the data is collected in is known as the naive basis. To carry out PCA first form the data matrix X, which is the $n \times m$ matrix created by setting each of the n data vectors as columns. X must then be row-centered such that the mean of each row has been shifted to be zero. The aim of PCA is to find a new basis, which is a linear combination of the naive basis, that will better represent the data. Let Y be the data matrix of the new representation, and say it is related to X by a linear transformation P, such that

$$Y = PX \tag{3.2}$$

The aim of PCA is to find the matrix representing the linear transformation P. The rows of this matrix are the principal components of the data set. We require that the principal components are orthonormal. Thus geometrically, applying P simply corresponds to a rotation of the basis vectors. The $m \times m$ covariance matrix corresponding to the data matrix X is defined as

$$C_X = \frac{1}{n-1} XX^T \tag{3.3}$$

Defining the covariance matrix in this way means that the i, j^{th} component of C_X corresponds to the dot product of the i^{th} and j^{th} feature vectors of X. But since X has been row-centered, the mean of each feature vector is now zero, so the dot product of two feature vectors is equivalent to the covariance between the features. Therefore the i, j^{th} component of C_X gives the covariance of the i^{th} and j^{th} features. The diagonal elements of C_X give the variance of each of the different features.

For the new data we wish to minimize redundancy between features, so that there are no correlations between components. This means the off diagonal

elements of the new covariance matrix C_Y should be zero, as the covariance between each pair of the new features should be zero. We also require the first feature to have the greatest variance, the second feature to have the second largest variance, and so on. Therefore the diagonal elements of C_Y should be ordered by magnitude.Therefore we want to find the matrix P such that the matrix

$$C_Y = \frac{1}{n-1} YY^T = \frac{1}{n-1} PXX^T P^T = PC_X P^T \tag{3.4}$$

is diagonal and in rank order. By definition C_X is a symmetric matrix. Therefore it is diagonalizable and its eigenvectors are orthogonal. Let the decomposition of C_X be given by

$$C_X = EDE^T \tag{3.5}$$

where E is the orthogonal matrix with the eigenvectors of C_X as its columns, and D is the diagonal matrix of the corresponding eigenvalues. Place the eigenvalues into D in rank order. Then by setting $P = E^T$, the new covariance matrix becomes

$$C_Y = P(EDE^T)P = (E^T E)D(E^T E) = D \ . \tag{3.6}$$

Thus choosing P in this way ensures that C_Y has the properties we desire. Therefore the principal components of a data set are given by the eigenvectors of its covariance matrix. The variance along each of the principal components is given by the corresponding eigenvalue.

I choose to use PCA over other available non-linear dimensionality reduction methods as it allows me to easily extract information about the reduced components that the algorithm finds. I verify that the dimensionality reduction is not significantly different when using the UMAP algorithm (McInnes et al., 2018) instead.

Table 3.2 *The MSE, quantifying the performance of different models at predicting baryonic properties of subhalos. All scores are for predictions on the test set. Values were calculated from averaging 10 train/test splits.*

Prediction	Figure	MSE ($\times 10^{-3}$)
TNG100	Fig. 3.6	0.82
TNG50	Fig. 3.7	0.78
TNG100: Sublink merger trees	Fig. 3.7	0.75
TNG300	Fig. 3.7	1.13
TNG100: Low density environment	Fig. 3.7	0.90
TNG100: Medium density environment	Fig. 3.7	1.08
TNG100: High density environment	Fig. 3.7	1.25
TNG100: Stellar metallicity	Fig. 3.7	1.16
TNG100: SFR	Fig. 3.7	3.47
TNG100: Gas mass	Fig. 3.7	1.37
TNG100: $z = 2$	Fig. 3.8	0.51
TNG100: $z = 1$	Fig. 3.8	0.82
Illustris	Fig. 3.9	0.64
EAGLE	Fig. 3.9	0.76
EAGLE: FBconst variation	Fig. 3.9	0.77
EAGLE: FBZ variation	Fig. 3.9	1.01
EAGLE: NoAGN variation	Fig. 3.9	0.05

3.3 Applying to subsamples from IllustrisTNG

3.3.1 Is the model learning relationships?

In this work I show how the feature importance changes for different simulations and different subsamples of galaxies. In order for the feature importance to be meaningful, I need to ensure that the model has been able to successfully learn a relationship between the input and output features. In the left panel of Figure 3.5 I show the true vs predicted stellar mass value for 1000 randomly sampled galaxies from TNG100. A model that made perfect predictions would correspond to all points lying on the diagonal. The small scatter in the figure shows that the ERT model has successfully learnt a function mapping the input features to stellar mass, so the feature importance values can be trusted.

The MSE scores from each model are shown in Table 3.2. As in Chapter 3 I scale the output values to the range $[0, 1]$ when calculating the MSE. This allows MSE scores from models trained on different data sets to be easily compared. As the MSE score for the SFR prediction is significantly worse than for any other model,

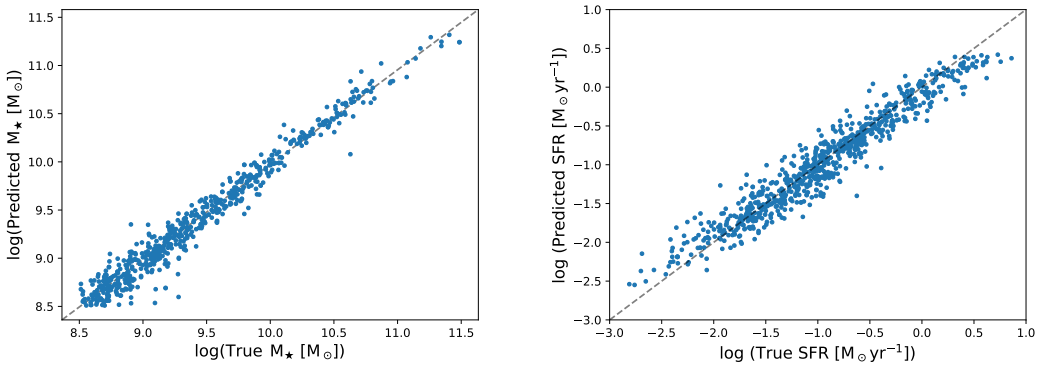

Figure 3.5 *Model performance on 1000 randomly sampled galaxies from the TNG100 test set. The x axis shows the true value from the simulation, and the y axis shows the value predicted by the machine learning model.* ***Left*** *Stellar mass prediction.* ***Right*** *SFR prediction. This shows that the model has learned a relationship between the input and output features.*

I show the true vs predicted SFR in the right panel of Figure 3.5.

3.3.2 Reading feature importance plots

Figure 3.6 show the feature importance obtained from a model trained to predict $z = 0$ stellar mass of galaxies in the TNG100 simulation. Merger trees were extracted using the LHaloTree algorithm. Each point on the plot corresponds to the importance of an input property at a certain time in the simulation. I include all input properties other than the one I am predicting, e.g. I do not use stellar mass as an input feature when predicting stellar mass. The maximum value of the feature importance is normalized to one for all models, so I only consider the relative importance of the input properties at different times rather than focusing on the absolute values. I highlight the fact that a large feature importance value does not necessarily mean that the input feature has a large value at that point. For example, in Figure 3.6 the feature importance for the dark matter mass peaks at early times then drops off. This does not mean that the halos within IllustrisTNG are decreasing in dark matter mass, instead it indicates that the dark matter mass at early times is more informative for predicting the stellar mass at $z = 0$ than the dark matter mass at late times. I also note that a high feature importance value does not mean that the input feature has a positive correlation with the output feature being predicted. For example, I find that black hole mass is an important factor when predicting SFR, but for large

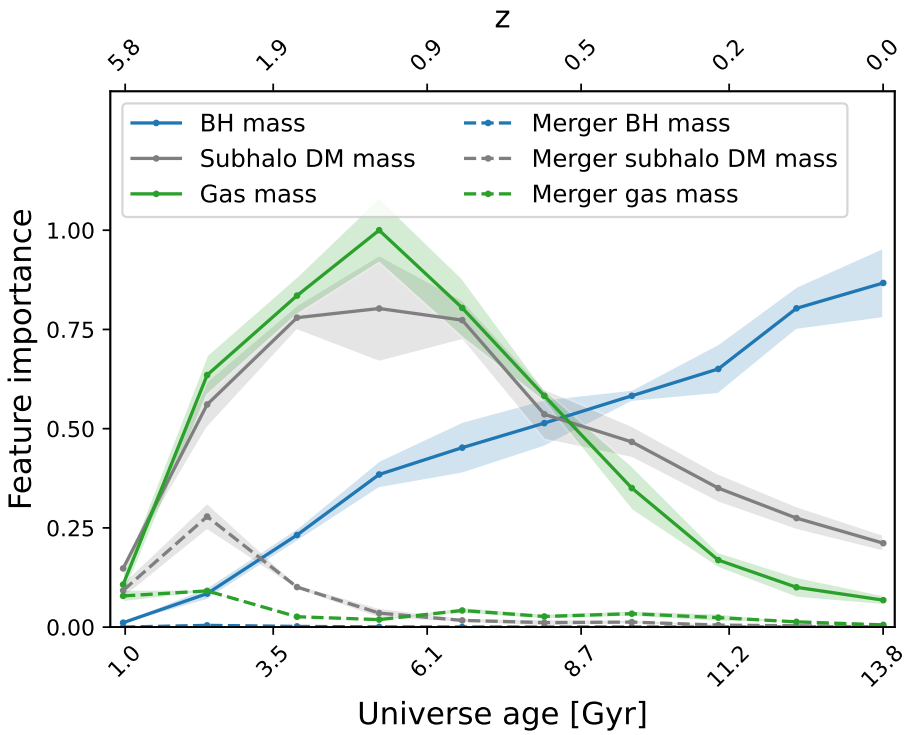

Figure 3.6 *Feature importance of an ERT model trained to predict stellar mass of galaxies in the TNG100-1 simulation. Merger trees were generated using the LHaloTree algorithm.*

galaxies black hole mass is negatively correlated with SFR. To give an error on the feature importance I train a model for 10 different train/test splits of the data. The shaded region in Figure 3.6 corresponds to the standard error taken on the feature importance values from the 10 different models.

I now interpret the feature importance plot shown in Figure 3.6. I see that the dark matter and gas mass feature importance peak at early times, then decrease. Physically this corresponds to the initial period of formation when the universe SFRD is highest and most stars are being formed. The black hole feature importance is similar in magnitude to that from gas and dark matter mass, but peaks at late times. Relative to the importance of the main progenitor branch, the feature importance of mergers is very small, and peaks earlier than the main progenitor feature importance. Black holes from mergers are deemed to be completely unimportant. This is unsurprising as most halos that merge will be too small to host a black hole. However, the fact that the feature importance for black hole mergers is zero provides evidence that my model is not overfitting, since an overfitted model would end up using uninformative features to make splits.

3.3.3 Comparing feature importance plots

Figure 3.7 contains a number of feature importance plots from models trained on the IllustrisTNG simulation suite. In the top row I show the effect of varying resolution. The top centre plot gives the feature importance of a model trained to predict the stellar mass of TNG100 galaxies. This is the same as Figure 3.6, except the merger trees were generated using the SubLink algorithm. I see the same trends in the two plots, with all progenitor input properties peaking at the same point, and having the same relative importance. There is a minor difference in the importance of the merger features, with them being deemed less important for the SubLink merger trees. However, the overall agreement provides evidence of the robustness of this method to the choice of merger tree algorithm. For the remaining IllustrisTNG plots I use the LHaloTree merger trees.

The top left and top right panels show a model trained using galaxies from TNG50 and TNG300 respectively. The general trends are very similar to the model trained on TNG100, but some differences appear. For all three resolutions the BH mass peak has a similar value as the gas mass peak. For TNG50 the peak of the gas mass feature importance has a similar magnitude to the peak of the dark matter mass, whereas for TNG300 the peaks are further apart, and for TNG100 the distance between peaks is intermediate. Thus there is a clear trend in decreasing dark matter feature importance with decreasing resolution. This trend also occurs in the feature importance plots from models trained on the lower resolution Illustris simulations. In low resolution simulations the deep potentials at the centre of halos cannot be fully resolved. This means they have less ability to hold on to baryons since stellar feedback is capable of driving gas further from the ISM, significantly impacting star formation. Thus this method allows for the impact of resolution to be quantified as it can be clearly seen at what resolution the feature importance plots start to diverge. There are no significant differences in the black hole feature importance. The peak of star formation appears to occur at the same point for all three resolutions. In Ludlow et al. (2020) a suite of simulations was run using the EAGLE model with fixed particle mass, but the force softening scale was varied. They found that the cosmic star formation history becomes increasingly biased toward high-redshift, but that the effect was small for the range of values used by the different TNG simulations.

In the middle row of Figure 3.7 I show the feature importance from models trained on galaxies taken from different density environments. All galaxies are taken

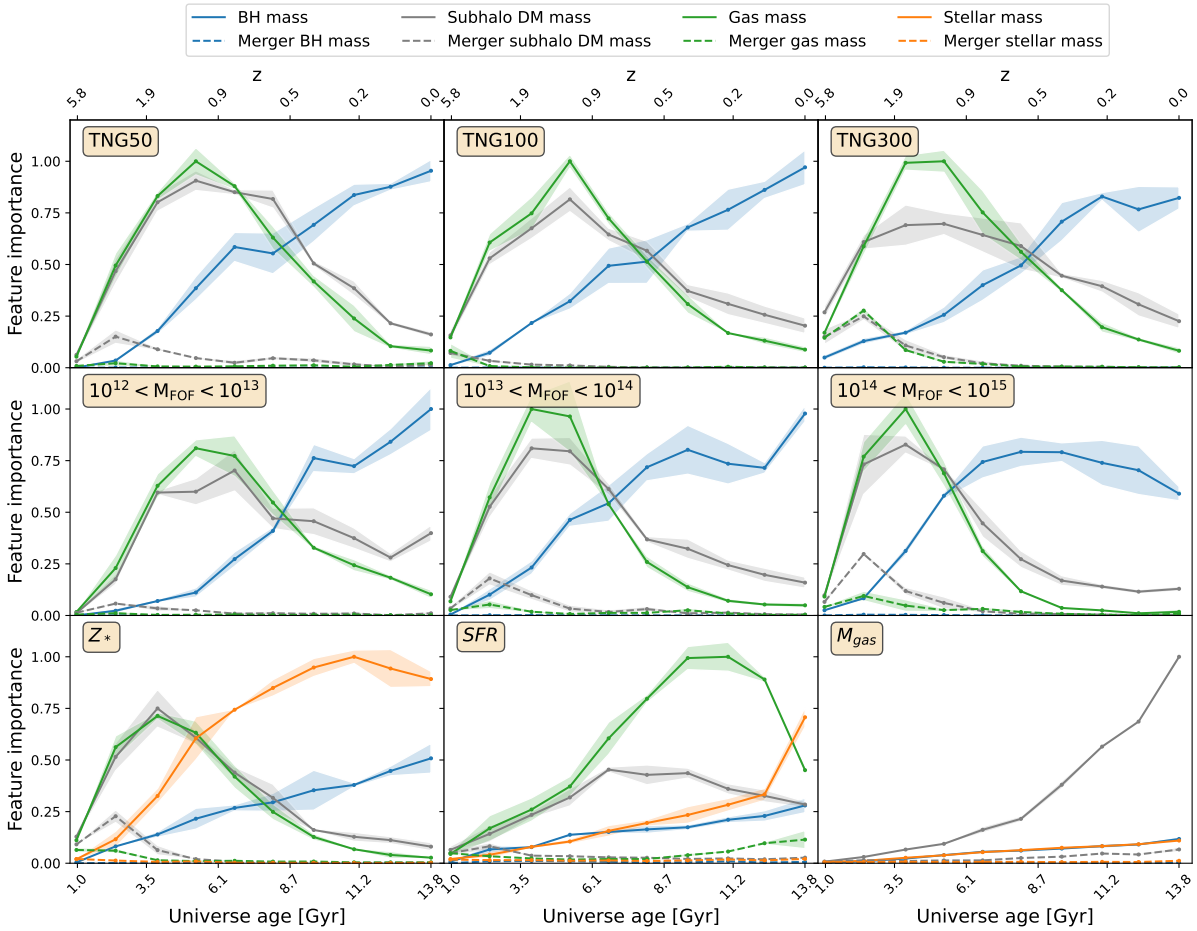

Figure 3.7 *Feature importance plots from models trained on IllustrisTNG, so all panels have the same subgrid models. Models are trained to predict $z = 0$ stellar mass unless otherwise indicated. The legend at the top of the figure is shared across all panels.* ***Top left*** *TNG50 - higher resolution simulation,* ***Top centre*** *TNG100 - SubLink merger trees,* ***Top right*** *TNG300 - lower resolution simulation,* ***Middle left*** $10^{12} < M_{FOF} < 10^{13}$ *- Low density environment,* ***Middle centre*** $10^{13} < M_{FOF} < 10^{14}$ *- Medium density environment,* ***Middle right*** $10^{14} < M_{FOF} < 10^{15}$ *- High density environment,* ***Bottom left*** *Predicting $z = 0$ stellar metallicity,* ***Bottom centre*** *Predicting $z = 0$ SFR,* ***Bottom right*** *Predicting $z = 0$ gas mass*

from TNG100 and are split into three samples based on the mass of their FOF halo. The bin edges are given by $\log \mathrm{M_{FOF}} = 12, 13, 14, 15$. The time at which the peak in gas and dark matter importance occurs shifts as the environment is varied. For galaxies in low density areas the peak occurs at later times, indicating delayed galaxy formation. This agrees with the findings of Jeon et al. (2022) who examined the star formation history (SFH) of a range of galaxies from the Horizon-AGN simulation (Dubois et al., 2014), showing that IllustrisTNG and Horizon-AGN are in agreement in this area. These results are also consistent with observational studies which determine SFHs using stellar population modelling (Thomas et al., 2005; Guglielmo et al., 2015). For the galaxies in high density regions the majority of star formation occurs at earlier times. This also causes the black hole feature importance to peak prior to $z = 0$. I find that the $z = 0$ black hole feature importance decreases with increasing density, in agreement with the observational results of Ceccarelli et al. (2022). As I increase density the importance of merger features relative to the progenitor features increases. This is to be expected as galaxies within groups and clusters will experience a large number of mergers, albeit at early times.

In the final row of Figure 3.7 I show the feature importance from models trained to predict other properties of TNG100 galaxies at $z = 0$. The left, centre, and right panels correspond to output features of stellar metallicity, SFR, and gas mass respectively. The feature importance of dark matter and gas, and black hole mass in the stellar metallicity plot is similar to that from stellar mass plots.

The feature importance plot for SFR is significantly different to those for stellar mass, with features peaking close to or at $z = 0$. This is because SFR is an instantaneous property unlike stellar mass which builds up over time. As the dark matter mass gives the gravitational potential, which indicates how much gas will fall onto the halo. However, gas must cool before it can form stars, which explains why the dark matter feature importance peaks at earlier times than any other input property, as any gas which was recently accreted is unlikely to contribute to star formation. Of the merger features only gas mass is important, but it does indicate that mergers have a minor effect on the overall galaxy population SFR at $z = 0$.

For the prediction of gas mass the dark matter mass dominates. This is because I am predicting the gas mass of the halo, of which the majority is hot gas, rather than the mass of the ISM. Stellar mass and black hole mass do have some importance, but this plot shows that in general the feedback in the IllustrisTNG

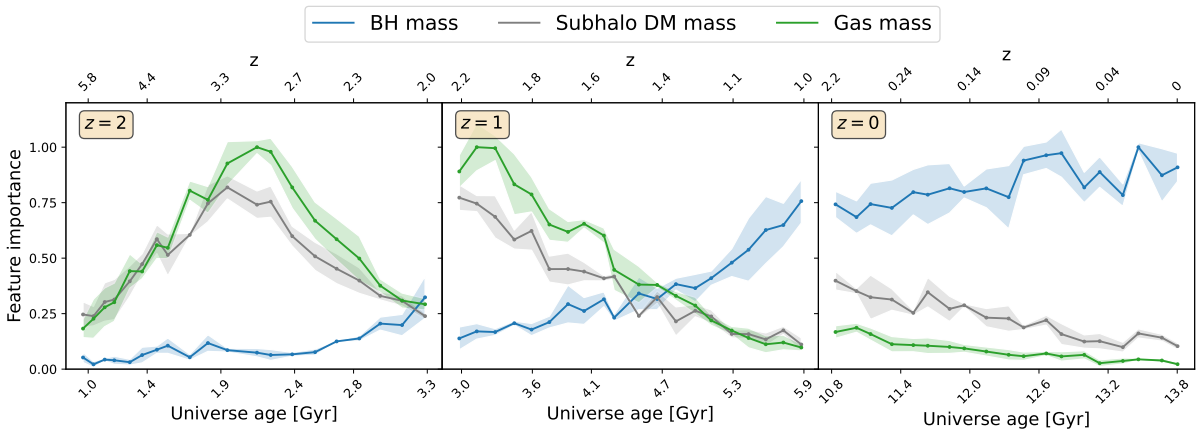

Figure 3.8 *Feature importance plots for predicting the stellar mass of TNG100 galaxies at different redshifts.* ***Left*** $z = 2$, ***Centre*** $z = 1$, ***Right*** $z = 0$. *The legend at the top of the figure is shared across all panels.*

model is insufficient to eject gas from halos.

3.3.4 Predictions at different redshifts

Figure 3.8 shows models trained to predict stellar mass for galaxies from TNG100 at different redshifts. I wish to consider how the relative importance of the various input properties changes when considering predictions of stellar mass at different times. The left, centre, and right panels show predictions for $z = 2, 1, 0$ galaxies respectively. Due to the age of the universe at $z = 2$ I can only use galaxy properties from the past 3Gyr as input features. I therefore also restrict the inputs to the $z = 1$ and $z = 0$ models to the past 3Gyr to allow for a direct comparison.

The leftmost panel shows a clear peak in gas and dark matter mass around $z = 3$. This is despite the fact that there is still a large amount of star formation ongoing at $z = 2$. However, the gas needs time to collapse into the gravitational well of the halo, as well as radiate away energy, before it is able to form stars. This explains why the peak is located about 1Gyr prior to $z = 2$. In the $z = 1$ panel I see gas and dark matter mass continuously dropping, indicating that 3Gyr before $z = 1$ is already past the peak of star formation. However, the black hole feature importance continues to increase, indicating it's coupling to the stellar mass of galaxies. For the $z = 0$ plot the feature importance is nearly flat across time. This is a reflection of the lack of star formation in the majority of galaxies in the present epoch. A trend across the three panels is the increasing importance of

black holes. This is because it takes time for black holes to build up and become effective, at $z = 2$ they are not that massive for most galaxies. The right panel of Figure 3.8 can be compared with the last 3 Gyr of Figure 3.6. When looking at the feature importance in this range of Figure 3.6 it can be seen that BH mass is the most important feature, and its importance is increasing with time. It can be seen that the importance of gas mass and DM mass is decreasing with time. These trends are in agreement with Figure 3.8.

3.4 Applying to different simulations

In Figure 3.9 I compare the feature importance from models trained on simulations with different subgrid model implementations. As these simulations have slightly differing cosmologies, the horizontal axes are not exactly aligned, but they are close enough for comparisons to still be valid. Figure 3.1 contains the stellar mass function for TNG, Illustris, and EAGLE. Within the stellar mass range of the training data there is reasonable agreement between the simulations. Restricting the mass range further still yields different feature importance plots between simulations. The top left panel shows the results from TNG100, as discussed in Section 3.3. The top centre panel shows a model trained on the original Illustris simulation. There are three differences when compared with the TNG. Firstly the gas mass importance relative to dark matter is significantly increased for Illustris. Secondly the peaks in gas and dark matter mass are less pronounced for Illustris, and they occur at later times compared to the TNG. Both of these differences are reflections of the changes to the supernova feedback implementations. The feedback in TNG is more effective, which means a deep gravitational potential is needed to hold on to gas in star forming halos. This explains why dark matter importance increases for the TNG. The reduced efficiency of stellar feedback in Illustris means that star formation can continue for longer, which is reflected in the shape and location of the peaks. The third difference is the relative importance of black holes, which are much more prominent in the TNG. This is a result of the new black hole feedback mode introduced in the TNG which boosts black hole feedback at low accretion rates. The merger trees for the Illustris simulation are created using the SubLink algorithm, which explains the differences in the merger feature importance.

The top right panel shows the results of the model trained on the fiducial EAGLE simulation. The gas mass and dark matter mass are close to those from the TNG,

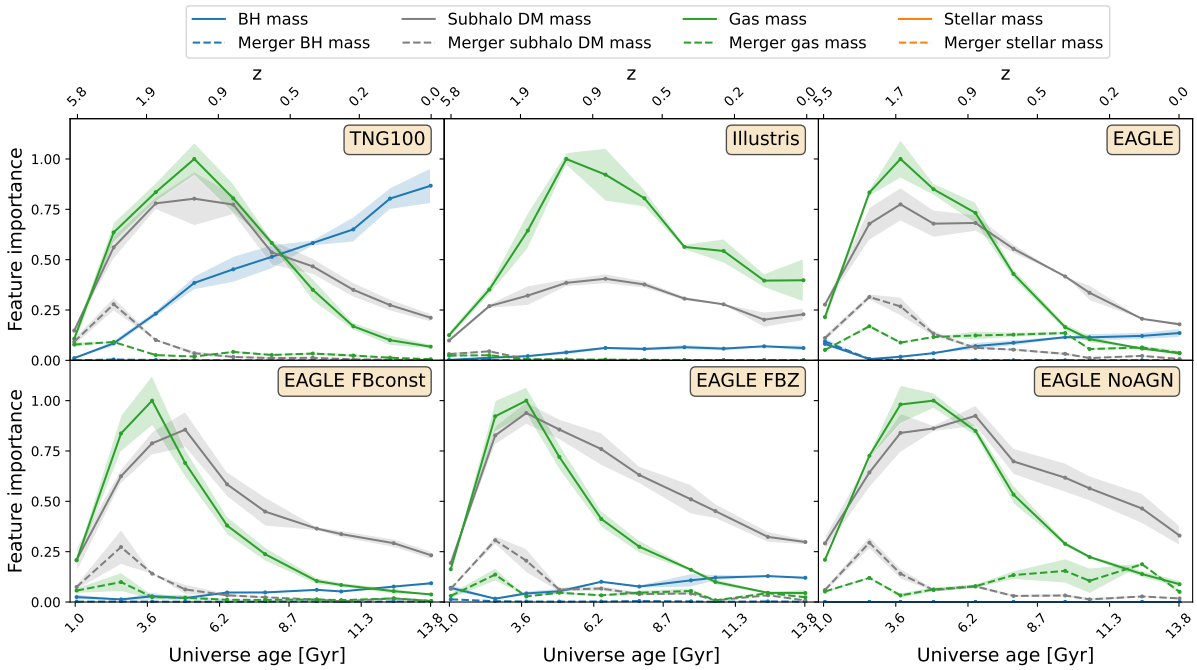

Figure 3.9 *Feature importance plots for predicting stellar mass of simulations with different subgrid models. The legend at the top of the figure is shared across all panels.* ***Top left*** *TNG100,* ***Top centre*** *Illustris,* ***Top right*** *EAGLE fiducial,* ***Bottom left*** *EAGLE FBconst,* ***Bottom centre*** *EAGLE FBZ,* ***Bottom right*** *EAGLE NoAGN. As the simulations have slightly different cosmologies, the horizontal axis are not exactly aligned.*

but the black hole importance is similar to Illustris. This is interesting as EAGLE is run with an SPH code, but both TNG and Illustris use a moving mesh to model the gas hydrodynamics. The fact that the EAGLE gas mass and dark matter mass importances are much closer to the TNG than the TNG is to Illustris shows that it is the subgrid models that are to first order key to determining the correct build up of galaxy properties, rather than the hydrodynamics solver which is used. This agrees with the results of Scannapieco et al. (2012) who simulated an individual Milky Way-like galaxy using multiple cosmological hydrodynamical codes. The black hole importance shows that AGN feedback in EAGLE has similar efficiency to that in Illustris, despite the significantly different implementations.

The bottom right panel shows the EAGLE run without any black holes. Compared with the fiducial EAGLE run the gas and dark matter mass is more important at late times in the NoAGN run. This confirms that AGN do have any effect in shutting off some star formation in EAGLE.

The bottom left panel shows the EAGLE run where supernova feedback is independent of environment. The ISM dependence in the fiducial run makes feedback more efficient for low mass galaxies. Thus when the feedback is constant more star formation can occur in low mass galaxies, which means more stellar mass will build up at early times. This is reflected in the fact that the gas and dark matter peaks move to the left for the FBconst plot. There is also a decrease in the black hole importance.

The bottom centre panel displays the EAGLE run where supernova feedback depends only on metallicity, unlike the fiducial run where it also depends on density. The peak occurs at a similar time to the FBconst run, showing that it is the density rather than the metallicity which is the factor determining the location of the peak of star formation density. However, the dark matter feature importance is more similar to the fiducial run, indicating the metallicity dependence is responsible for this feature.

3.5 Applying to CAMELS suite

In this section I apply the method to the CAMELS suite. I first focus on the effect of varying the A_{SN2} parameter in the IllustrisTNG simulations from CAMELS, then compare the other parameters and the Simba simulations.

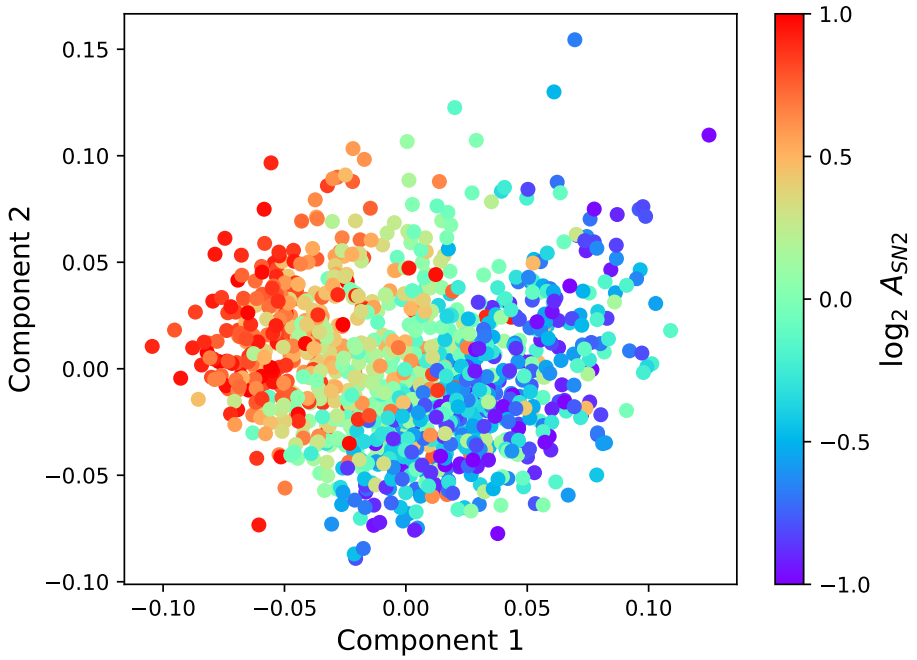

Figure 3.10 *PCA plot of stellar mass feature importance applied to the IllustrisTNG CAMELS simulations. Each point represents a single simulation. Points are colored by the speed of the supernova winds within the simulation.*

3.5.1 Correlations between supernova feedback and PCA components

For each of the $N = 1061$ IllustrisTNG simulations in the CAMELS suite I train an ERT model to predict the stellar mass at $z = 0$. From each of these models I extract the feature importances, and concatenate them into a vector. The feature importance vector from each model has a length of $M = 30$, which corresponds to 3 input properties (black hole mass, dark matter mass, gas mass) at 10 different snapshots. Combining the feature importances from all the simulations gives a matrix of size $N \times M$. I apply PCA to this matrix, and show the results in Figure 3.10. In this plot each point corresponds to a single simulation. The horizontal axis corresponds to the first PCA component, and the vertical axis corresponds to the second PCA component. I colour each of the points by the value of the A_{SN2} parameter of the simulation. Since A_{SN2} is sampled uniformly in log space for the LH and 1P sets, the colorbar is also logged.

It can be seen that there is a clear trend with supernova feedback in the value of the PCA components, with a large A_{SN2} value corresponding to a low component

1 coefficient. There is a slight trend in component 2, but this is minor compared to the variation in component 1. Rather than showing scatter plots for all the PCA components, I summarise the information in Figure 3.10 in the bottom left panel of Figure 3.12. Here I plot the mean value of the coefficient of the i^{th} component for different A_{SN2} bins. This shows the negative correlation between the value of A_{SN2} and the first PCA component, along with the minor trend in the second component. This plot also allows for the other PCA components to be examined. It can be seen that there is some correlation with A_{SN2} and the third component. I have not plotted the fourth component to avoid overcrowding, but it does not show any correlation.

3.5.2 Physical interpretation of PCA components

Now that I have established the relationship between each of the PCA components and A_{SN2}, I wish to determine what each component corresponds to physically. To do this I plot the feature importance of the PCA mean and each of the first three components in Figure 3.11. It should be expected that the feature importance of the mean will be similar to the one obtained from TNG100 (shown in Figure 3.6). I find the relative importance of each of the input properties to be similar, but the peak occurs later. This is a result of three factors which reduce my ability to track the halos through the simulation. Firstly the mass resolution is lower for CAMELS than for TNG100, which means resolution effects are more likely for small halos, and halos which exist at early times in TNG100 might not have enough particles to be identified in CAMELS. Another factor is that the spacing between snapshots is larger for CAMELS, meaning that it is more difficult to track halos between snapshots as they may have moved further, and gained/lost more mass. Finally, while I am able to match a high fraction of Rockstar halos with their SubFind counterparts, some halos are matched incorrectly or no match can be found.

Unlike the previous feature importance plots, in Figure 3.11 each of the components must sum to zero by definition. This means it is possible to have negative importance values. This can be seen in the plot of component 1, where dark matter mass always has a negative value. Consider two simulations, A and B, where simulation A has a larger component 1 value than simulation B, and the coefficients for all the other components are identical. In this case the relative importance of gas mass to dark matter mass would be greater for simulation

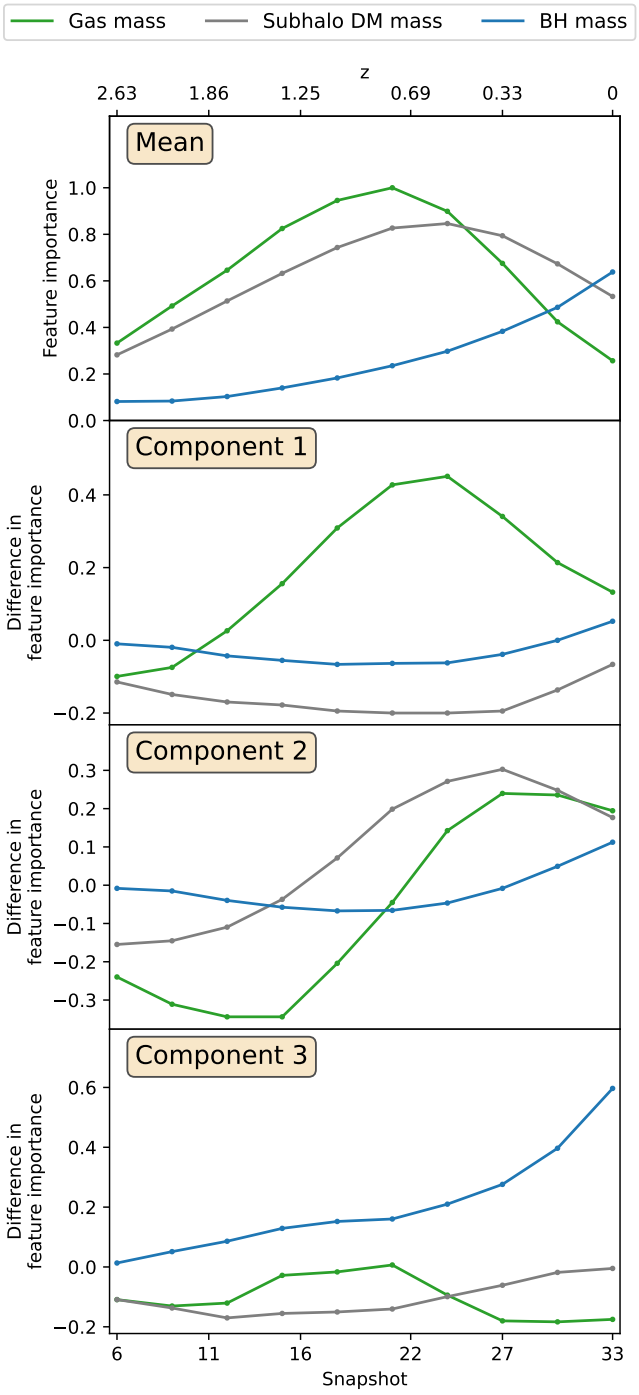

Figure 3.11 *Plots of the value of each component resulting from PCA applied to the feature importance vectors of models trained to predict the stellar mass of galaxies from the CAMELS simulations. The legend at the top of the figure is shared across all panels.*

A than for simulation B. This means that increasing component 1 corresponds to a decreasing importance of dark matter mass, which gives the gravitational potential of the halo. When the speed of supernova winds are increased, they are more likely to blow gas out of the halo. This is why the dark matter mass is more important for simulations with a large A_{SN2} value, and explains the bifurcation shown in Figure 3.10.

When plotting component 2, it can be seen that there are negative values of gas and dark matter mass at early times, and positive values at late times. Therefore increasing the component 2 value means that the peak in gas and dark matter mass will occur at a later point in time, corresponding to galaxies which form later. Increasing A_{SN2} causes galaxies to form later as the gas is ejected further from the galaxy, and so takes longer to cool and return before it can form stars.

For component 3 the difference in gas and dark matter mass is always negative, but is relatively flat. Thus component 3 corresponds to an increasing importance of the black hole.

3.5.3 Comparing parameters

I now consider the effect of the varying the other simulation parameters. From Figure 3.12 the effect of modifying the simulation parameters on the feature importance of the CAMELS TNG simulations can be seen. The vertical axis is set to the same scale for all plots to allow for comparison.

The top left panel shows the effect of changing Ω_m. For low Ω_m values galaxies form earlier. It is interesting to compare this with the feature importance plots from the different density environments as shown in Figure 3.7. In that case I found that galaxies in low density regions tended to form later. Low density regions can be regarded as a separate universe with a low Ω_m value, so this agrees with my findings from the PCA components. However, for the different density regions there was no difference in the gas and dark matter mass relative importance, but in CAMELS I do find a correlation with Ω_m.

Changing the value of σ_8 also effects the time at which galaxies form. Since for low σ_8 values the density peaks are smaller, it takes longer for the halos to collapse and allow galaxies to form.

Varying A_{SN1}, which increases the energy per unit star formation, causes similar

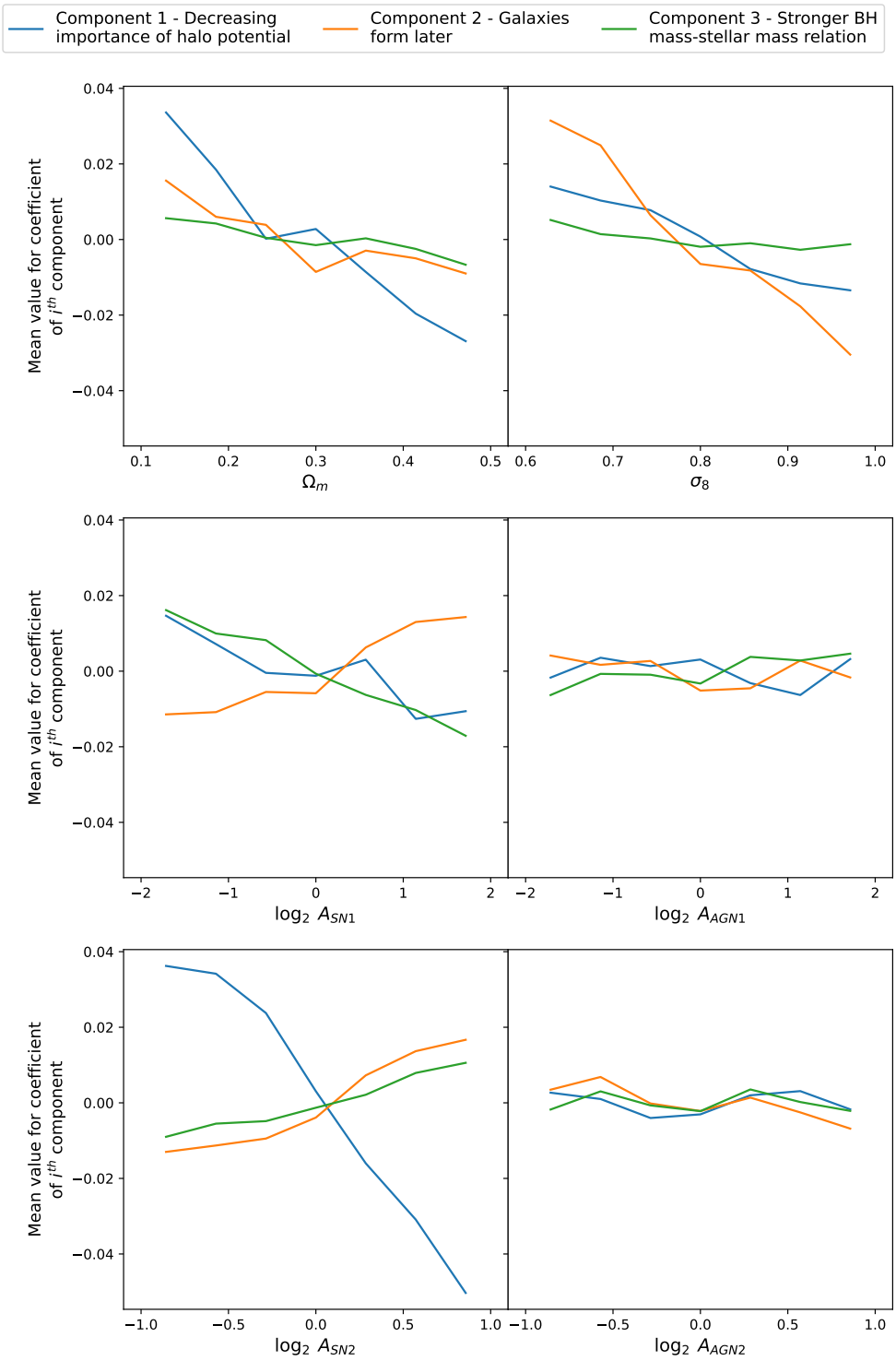

Figure 3.12 *The effect of varying the simulation parameters on the PCA components. Models are trained on the IllustrisTNG simulations from CAMELS.* ***Top left*** *Omega matter,* ***Top right*** *Sigma 8,* ***Middle left*** A_{SN1}*,* ***Middle right*** A_{AGN1}*,* ***Bottom left*** A_{SN2}*,* ***Bottom right*** A_{AGN2}*. For a description of each parameter see Table 3.1.*

but less pronounced effects to A_{SN2}. However, it has the opposite effect on the black hole mass relation.

Changing the two parameters associated with the AGN feedback strength has no effect on the mean PCA coefficients. The reason for this could be because the number of galaxies without supermassive black holes is significantly larger than the number of those with one. Ideally I would consider only galaxies above a certain mass cut to exclude those without any black hole activity, but the $25(Mpc/h)^3$ box size of the CAMELS simulation is not large enough for me to do this. However, it also suggests that the black holes in the most massive galaxies are not having any significant effect on the evolution of neighbouring galaxies.

3.5.4 Comparing IllustrisTNG and Simba

I now apply the same analysis to the Simba simulations from the CAMELS suite. I decompose the feature importance vectors from Simba using the PCA components I obtained from the TNG galaxies. This allows for a more direct comparison between the two codes. I remind the reader that due to the different subgrid model implementations the subgrid model parameters (e.g. A_{SN1}) have different meaning for TNG and Simba. See Table 3.1 for a description of each parameter.

In Figure 3.13 I show the results of this analysis. In general the same trends as in the TNG simulations are found, but there is often a difference with component 3, which is mainly linked to the black hole importance. This is the case for Ω_m, σ_8, and A_{SN2}.

When increasing the value A_{SN1} for Simba I find that galaxies form later, which also occurs for the TNG, but that the halo potential importance decreases, opposite to the TNG. Tuning the A_{SN2} parameter in the Simba runs produces a larger impact on how late galaxies form than it does for the IllustrisTNG runs.

Unlike for the TNG, changing the black hole feedback parameters does have an impact on the feature importance. For the bottom right panel it can be seen that decreasing A_{AGN2} means that galaxies form later, due to the fact that their star formation is not being shut down. However, A_{AGN1} still shows no clear trend with any of the PCA components.

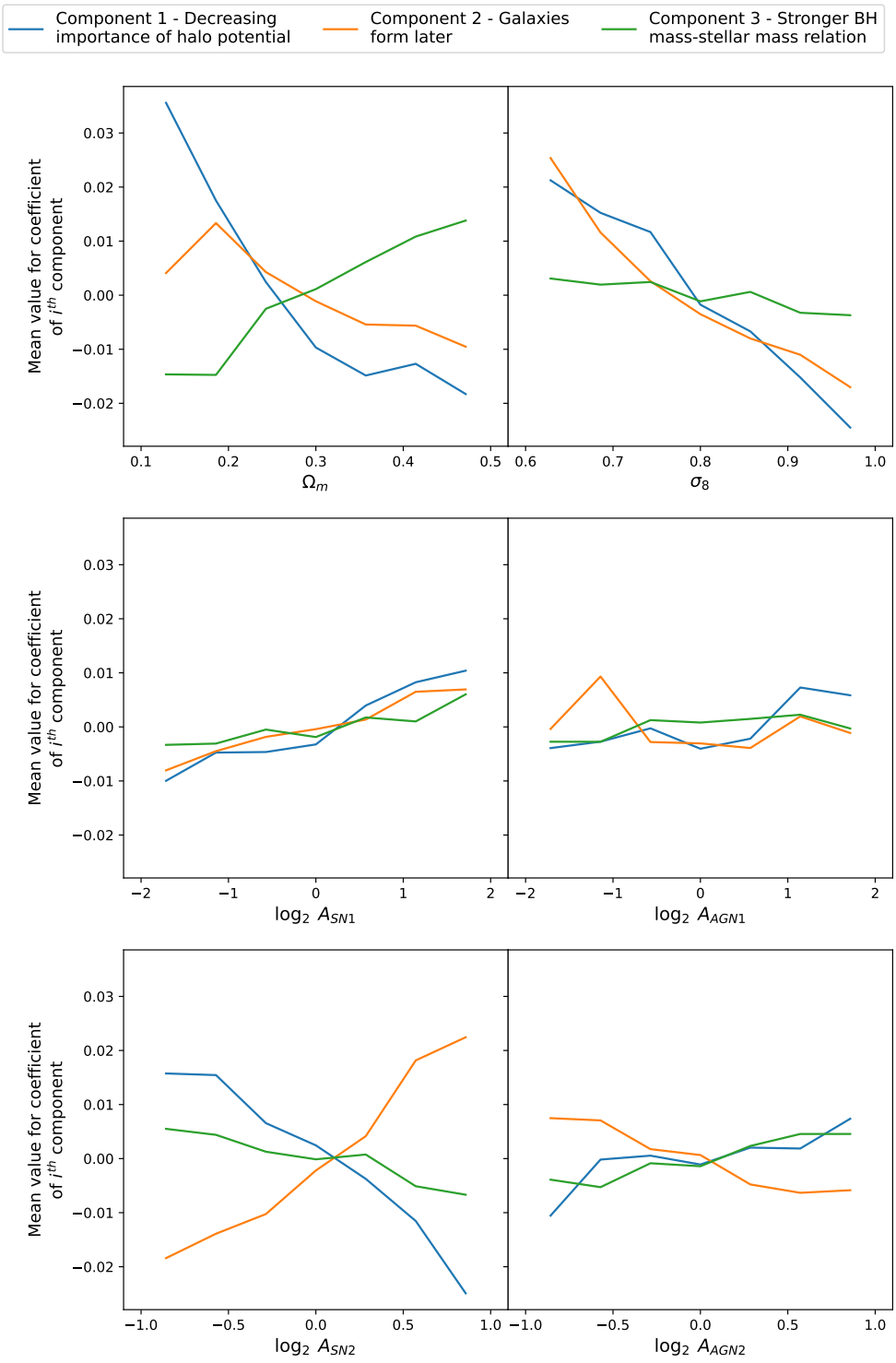

Figure 3.13 *Same as Figure 3.12, but using the Simba simulation from the CAMELS suite. Components are determined from PCA applied to the IllustrisTNG simulations only*

3.6 Discussion

In this section I add context to my results by discussing how they relate to the existing literature. However, it must be stressed that it is difficult to find direct comparisons with my work. The majority of work comparing different simulations, or examining how well simulations agree with observations, only considers galaxy properties at a single point in time (e.g. Ayromlou et al., 2022; Ma et al., 2022; Yang et al., 2022). The method presented in this work is novel in that it makes it possible to determine differences in how properties build up in different simulations, both in terms of the relative importance of physical processes, and the time at which they occur.

When comparing the results of Sections 3.3 and 3.5 I can sometimes see similar changes to the feature importance, e.g. decreasing the resolution of IllustrisTNG results in a decreased halo mass importance, which is also the result of increasing the supernova strength in CAMELS. This highlights the number of degeneracies in simulation outputs that can occur from the modelling choices made for a simulation (cosmology, hydrodynamics solver, subgrid models, resolution). It emphasizes the importance of developing methods that can break these degeneracies. The method shown in this chapter can identify what observations can be used to distinguish simulations. For example, Figure 3.13 shows that varying galactic winds speeds has a larger effect on galaxy formation times than varying the total energy per unit star formation, and so comparisons with observations of galaxy SFHs could be used to calibrate supernova feedback subgrid models.

The standard model of cosmology has proved incredibly effective at providing a good description of a wide range of astrophysical and cosmological data, but there remain observational tensions in the values of cosmological parameters (e.g. Di Valentino et al., 2021; Abdalla et al., 2022). Simulations must take these uncertainties into account. However, it is not clear how much effect the ΛCDM parameters will have on the evolution of galaxies within a simulation. The results shown in this work (which had not before been possible without large data set sizes and data analysis methods) by training a multi-epoch model on the CAMELS simulations show that varying the cosmological parameters has a large effect on the feature importance and therefore on the build up of galaxy properties. I find that tuning the cosmological parameters has a similar effect size to modifying the subgrid model parameters. As the majority of recent simulations are not

run for a range of cosmological parameters, this needs to be considered before comparing them to observations. A large number of high-z simulations (e.g. FLARES (Lovell et al., 2021), Forever22 (Yajima et al., 2022), SERRA (Pallottini et al., 2022), THESAN (Kannan et al., 2022)) have been introduced recently in order to compare with results from JWST. My results are especially relevant in this area since I find a significant effect on how early galaxies form.

Despite being a well researched topic, the impact of resolution on the formation of galaxies is an especially pertinent question currently for two reasons. One is that simulations which explore a large parameter space, such as the CAMELS simulations, cannot be run at high resolution. Another is that upcoming surveys such as Euclid (Racca et al., 2016) and LSST (Ivezić et al., 2019) are going to cover $\sim$Gpc^3 volumes. Large volume simulations such as the MilleniumTNG (Pakmor et al., 2022) and FLAMINGO (Schaye et al., 2023) simulations have been run to compare with these observed volumes, but due to the box size their resolution is low. The results of this work, as shown in Figure 3.7, suggest that decreasing resolution by 2 dex has only a minor effect on the formation history of stellar mass. The times at which star formation occurs is also unchanged. However, this analysis would need to be repeated for a range of resolutions in other simulations than IllustrisTNG.

Differences in hydrodynamical solvers introduce further uncertainty in the galaxy formation. New solvers continue to be introduced (e.g. Alonso Asensio et al., 2023; Morton et al., 2023), and tools have been developed to make it easier to compare different methods (e.g. Schaller et al., 2023). Recent work in Braspenning et al. (2023) has shown that different hydrodynamics solvers do not agree even for standard test cases, and thus it is important to consider how they will affect the galaxies produced in cosmological simulations. Studies (e.g. Schaller et al., 2015; Huang et al., 2019) examining the impact of galaxy properties at a single point in time find most are not significantly affected by the details of the hydrodynamics solver. Both Hayward et al. (2014) and Hopkins et al. (2018) considered the effect of hydrodynamics on the evolution of galaxies, but they ran simulations of isolated objects rather than comparing full cosmological volumes. My results consider the build up of properties in different cosmological simulations, as shown in Figure 3.9. I have demonstrated that there are more differences in physical drivers of galaxy evolution between IllustrisTNG and Illustris than between IllustrisTNG and EAGLE, showing that the impact of hydrodynamics method is considerably less important than the choice of subgrid models. Similarly the comparison between

TNG300 and TNG100 highlights the ability of subgrid models to address the limitations of resolution. These results emphasizes the importance of continuing to develop and tune subgrid prescriptions, including well established models such as those used for supernova feedback.

3.7 Conclusions and future work

My conclusions can be summarized as follows:

- I have introduced a novel method for extracting information about galaxy formation from simulations by extending the technique from Chapter 2. A summary of my method is shown in Figure 3.3. By considering the feature importance of baryonic properties it is possible to gain insights into the relative importance of different processes and the time at which they occur. I provide a guide for interpreting the resulting plots in Section 3.3.2.

- In the top row of Figure 3.7 I examine the impact of resolution. Decreasing the resolution has a clear effect on the feature importance, showing this novel method can be applied as a check for simulation convergence.

- From the central row of Figure 3.7 it can be seen that galaxies in higher density environments in IllustrisTNG produce stars at earlier times than in low density regions, but the impact of black holes is decreased. I also show that the properties of void galaxies in IllustrisTNG are in agreement with observations and other simulations

- By directly analysing cosmological simulations I show that differences due to subgrid models are considerably more significant than those introduced by modelling the gas using Arepo compared with the Anarchy SPH scheme. This can be seen by comparing the feature importance for EAGLE, Illustris, and IllustrisTNG in Figure 3.9.

- I use PCA to determine the effects of varying the subgrid model parameters within the CAMELS simulations. In Figure 3.11 I show the feature importance values corresponding to each of the principal components. I find the first component corresponds to the importance of the halo gravitational potential, and the second component relates to the time when galaxy formation takes place.

- In Figures 3.12 and 3.13 it can be seen that the Simba black hole feedback model has a larger effect on galaxy formation than the IllustrisTNG model, but stellar feedback remains the main driver in both.

- Through my analysis of the CAMELS simulations, I discover a substantial dependence between σ_8 and the time of galaxy formation. Given the current observational tensions in cosmological parameters, it is crucial for high-redshift simulations to consider this aspect when comparing their results with JWST.

This work is an example of how machine learning can help inform strategies about the best way to run future simulations. Other work in this emerging area includes Oh et al. (2022), who used a neural density estimator to tune the star formation model in simulations of a single MW-like halo, better aligning it with observational data. Kugel et al. (2023) and Jo et al. (2023) both employed emulators to connect subgrid model parameters to the observables resulting from a cosmological simulation. This approach provides valuable insights for determining appropriate parameter values. The method presented in this chapter holds promise for various applications, such as extending its usage to other simulations like the recently completed ASTRID and Magneticum runs of the CAMELS simulations (Ni et al., 2023). It would be interesting to apply to SAMs, especially to evaluate how well the model presented in Chapter 4 matches the simulation it is derived from.

Chapter 4

From simulations to SAMs

4.1 Introduction

As has been discussed in the previous chapters, modelling of galaxy formation is a complex non-linear process. In recent decades hydro-simulations and SAMs have been two of the main methods for modelling the evolution of galaxies. Consequently, there have been numerous papers which compare the galaxy populations produced by SAMs with those generated by cosmological simulations. (e.g Benson et al., 2001; Hirschmann et al., 2012; Mitchell et al., 2018; Ayromlou et al., 2021; Gabrielpillai et al., 2022) These studies allow for areas that SAMs are modelling incorrectly to be identified, as well as investigating the effect of physics missing from simulations.

An interesting approach is to try and directly tune a SAM to match the galaxies that result from a hydro-simulation. This strategy is significantly more viable than attempting to calibrate subgrid models to match a SAM, thanks to the high iterative speed achievable when testing new versions of a SAM. Stringer et al. (2010) modified a SAM to match a simulation code, and compared the results when modelling a single disk galaxy. They found good agreement between the two realizations of the galaxy. Neistein et al. (2012) took this a step further, by attempting to construct a SAM which would be capable of reproducing the galaxy population found within a simulation. To do this they started with a simple SAM which consisted of a set of coupled differential equations for describing the evolution of three phases (hot gas, cold gas, stars). They then extracted

the coefficients used in the differential equations from a simulation, e.g. for each galaxy at each snapshot they calculated the rate of conversion of gas into stars. Mitchell & Schaye (2021) extended the method of Neistein et al. (2012) by including further tracking of accretion and outflows. They applied this method to the EAGLE simulation, and used it to examine the stellar-halo mass relation by fixing the coefficients for different halo sizes and examining the resulting galaxy population. This allowed them to isolate the relative effects of star formation, ejection via outflow, and wind recycling, without having to run a large number of costly simulations.

In this chapter I alter the method used in the previous studies. All the preceding work in this area has been applied to SPH simulations. For such simulations it is easy to examine gas flow into and out of a halo by directly tracking what phase individual particles are in. However, mesh-based simulation codes are now often used for simulations, so I begin by changing the method to allow for the analysis to be carried out on both Lagrangian and Eulerian simulations. As has been discussed in Chapter 3, the past decade has shown the importance of black hole feedback within galaxy simulations, and therefore I include a black hole in my SAM. The resulting equations can be examined to learn about the results of the non-linear combination of the subgrid models, and the relative importance of each process that occurs in the simulations. The derived SAMs could also be applied to an N-body simulation to generate a large volume version of the galaxy population found in the reference simulation (as done with random forests in Chapter 2).

In order to derive equations from the simulation data, I use a machine learning method known as symbolic regression. It works by evolving a population of equations which are continuously applied to the data to assess their performance. I describe the details of the method in Section 4.4.1. A review of the use of symbolic regression within the physical sciences can be found in Angelis et al. (2023), and Cranmer (2023) describes a recent open-source symbolic regression library. Symbolic regression has only recently begun to be used within astrophysics, but has been applied to a wide range of problems including the prediction of solar activity (Shepherd et al., 2014), the modelling of exoplanet atmospheres (Matchev et al., 2022), and the classification of AGN (Russeil et al., 2022). Given the available data from cosmological simulations, it has been applied here in various works. Wadekar et al. (2020) connect the amount of neutral hydrogen within a halo to its environment. Shao et al. (2022) recover an

approximation of the virial theorem when attempting to predict the mass of a subhalo from its other properties. Delgado et al. (2022) use symbolic regression to explore what additional parameters can be useful for a HOD model. Wadekar et al. (2023) and Shao et al. (2023) use the CAMELS dataset to discover scaling relations which are robust to the feedback model used.

In the following chapter I begin with a description of how the simulation outputs are processed to extract the information needed to train the machine learning model. I examine the accuracy of a SAM where the coefficients of the differential equations are binned by halo mass, initially focusing on the IllustrisTNG simulation. I introduce and summarise the machine learning methods used in order to derive equations describing the evolution of galaxies. Results are then shown concerning the accuracy of the machine learning models, and I discuss what can be learned from the equations. I then compare the results found for IllustrisTNG with a similar analysis applied to other simulations.

4.2 Methods

4.2.1 One phase model

In this section I introduce the single-phase model, and describe how the data is extracted from the hydrodynamical simulations. The previous works (Neistein et al. (2012), Mitchell & Schaye (2021)) considered two phases of gas - cold gas which met the simulation's star formation threshold, and hot gas, which was all other gas bound to the subhalo. As a galaxy evolves hot gas will cool and fall to the centre of the halo, forming part of the ISM. At the same time gas within the ISM may be heated by feedback processes and become hot. For SPH codes it is possible to track individual particles to see what phase they were in at the previous snapshot, and so directly determine the rate of hot $\rightarrow$ cold and cold $\rightarrow$ hot gas. However for mesh-based codes this approach is not possible, unless tracer particles were to be included (e.g. Nelson et al., 2013). I therefore use a one-phase model, as applied in many other works (e.g. Cole, 1991; Khochfar & Silk, 2011; Krumholz & Dekel, 2012; Mitra et al., 2017), to allow the method to be applied to any simulation. Neistein et al. (2012) test a two-phase and a one-phase model, and find that both are capable of reproducing the galaxy population from the hydro-simulation.

Each galaxy is described by three values: the total gas mass, $M_{\rm gas}$, the stellar mass, M_*, and the mass of it's central black hole, $M_{\rm bh}$. The dark matter mass of its host halo, $M_{\rm h}$, is also tracked. The supply of gas is assumed to be provided by the accretion of gas as the dark matter halo grows. The gas supply is depleted due to stars forming and accretion onto the black hole. Each of these terms has an efficiency associated with it: $f_{\rm a}$ for the amount of gas brought into the halo alongside the dark matter, $f_{\rm s}$ for the fraction of gas that is converted into stars per unit time, and $f_{\rm b}$ for the fraction of gas which accretes onto the black hole per unit time. This leads to the following set of differential equations which describe the evolution of the one-phase model.

$$\begin{aligned} \dot{M}_{\rm gas} &= f_{\rm a}\,\dot{M}_{\rm h} \;-\; f_{\rm s}\,M_{\rm gas} \;-\; f_{\rm b}\,M_{\rm gas} \\ \dot{M}_{*} &= f_{\rm s}\,M_{\rm gas} \\ \dot{M}_{\rm bh} &= f_{\rm b}\,M_{\rm gas} \end{aligned} \tag{4.1}$$

where $\dot{M}_{\rm h}$ is the rate of dark matter accretion onto the halo.

4.2.2 Extracting data from simulation

For this chapter the results come from using the Illustris and TNG simulations, which are described in Sections 2.2.1, 3.2.1, & 3.2.1. For the simulation considered I select any central subhalos with more than 300 stellar particles at $z = 0$. These halos are then tracked back using their merger trees to the snapshot when they were first identified. Any halos which cannot be tracked to at least $z = 5$ are discarded. For each halo I extract the three properties of the one-phase model, plus the dark matter mass.

Figure 4.1 shows the properties used by the one-phase model for 4 representative galaxies from TNG100-1. The solid coloured lines show the raw data taken directly from the simulation. To this data I apply a smoothing procedure. There are two reasons for this. One is that a significant number of halos at some point in their history are misidentified by the halo finder. This can be seen as a spike in the dark matter and gas mass, and sometimes also in stellar mass. An example of this can be seen in the bottom left panel of Figure 4.1. The second reason is that at late times some halos exhibit a drop in gas mass, but this is due

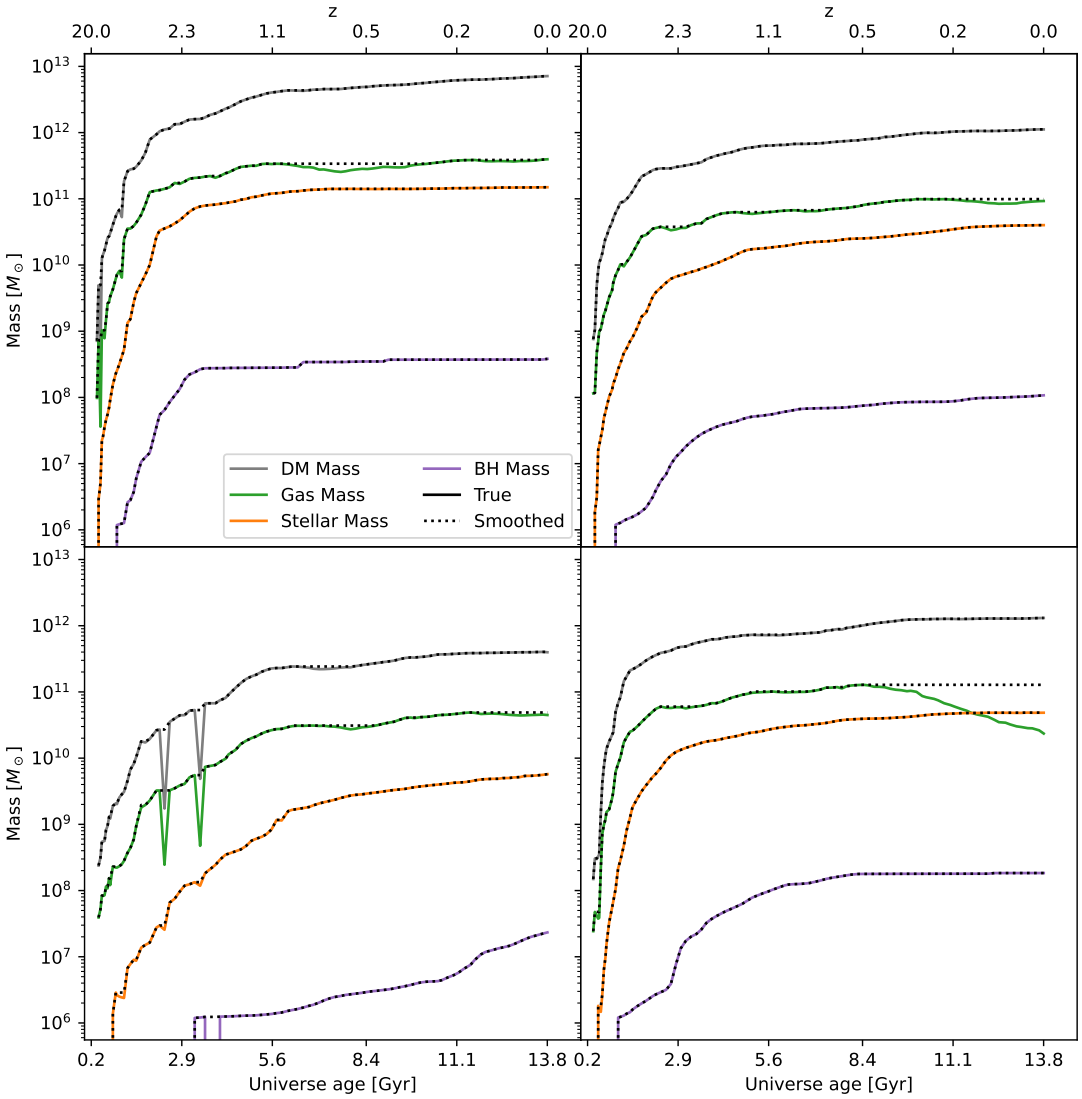

Figure 4.1 *Example of four halo mass histories from TNG100-1 before and after smoothing procedure. Solid coloured lines show the raw data from the simulation, dotted lines show the history after applying smoothing procedure.* ***Top row*** *Smooth halo histories,* ***Bottom left*** *Discontinuities due to halo finder misidentification,* ***Bottom right*** *Drop in gas mass at late times*

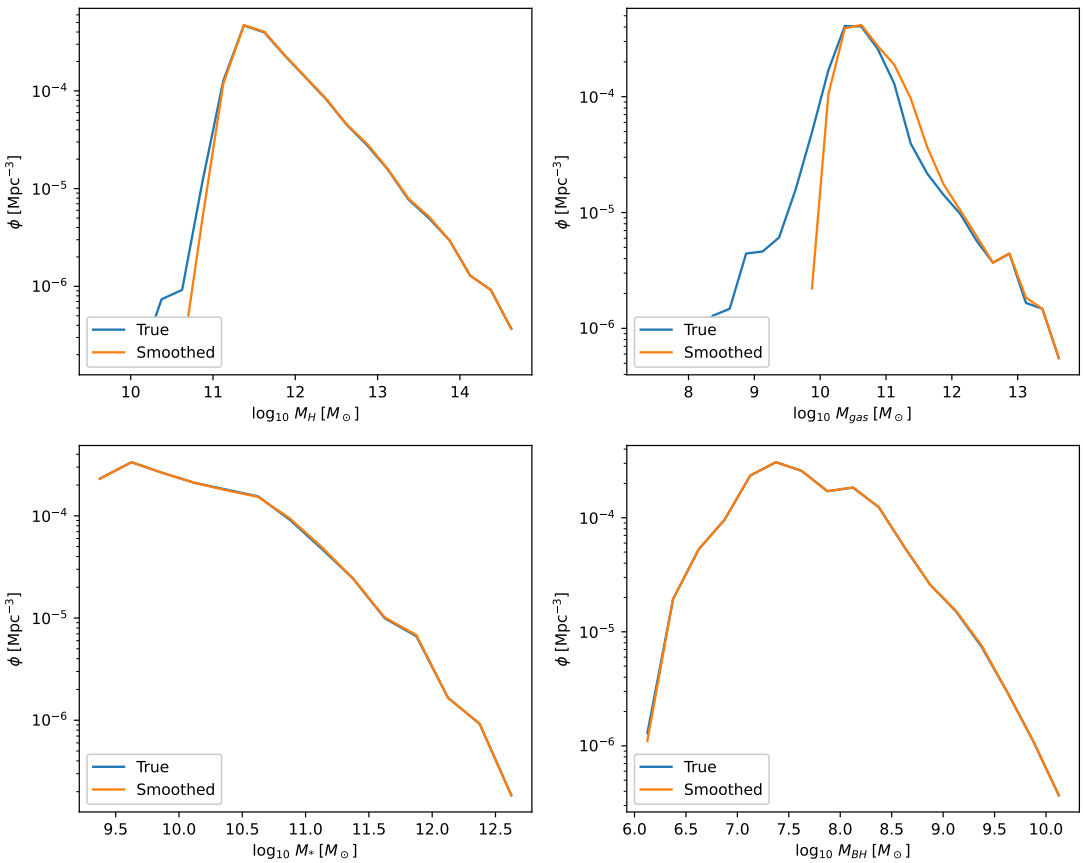

Figure 4.2 *Comparison of $z = 0$ true galaxy population from TNG100-1 simulation with smoothed data.* ***Top left*** *Dark matter mass,* ***Top right*** *Gas mass,* ***Bottom left*** *Stellar mass,* ***Bottom right*** *Black hole mass*

to outflows rather than star formation or black hole accretion. The one-phase model used here does not have a term for this kind of mass loss. Therefore the main part of the smoothing procedure is to require a monotonic increase in the properties. I loop through the snapshots, where snapshot i occurs at time t_i, and set $M(t_{i+1}) = \max(M(t_{i+1}), M(t_i))$. If between two snapshots the halo mass decreases, but the gas mass increases then I shift the gas mass increase to the nearest snapshot where the halo mass also increased. However, this is not required for many snapshots. The smoothed data is shown in Figure 4.1 with the dotted lines.

In Figure 4.2 I show comparisons of the mass functions before and after smoothing. It can be seen that the impact on the dark matter, stellar, and black hole mass is negligible. However, the gas mass does not line up. By looking at mass functions at other redshifts, and by visually inspecting the history of a large number of halos (as in Figure 4.1), this drop in gas mass does not occur for

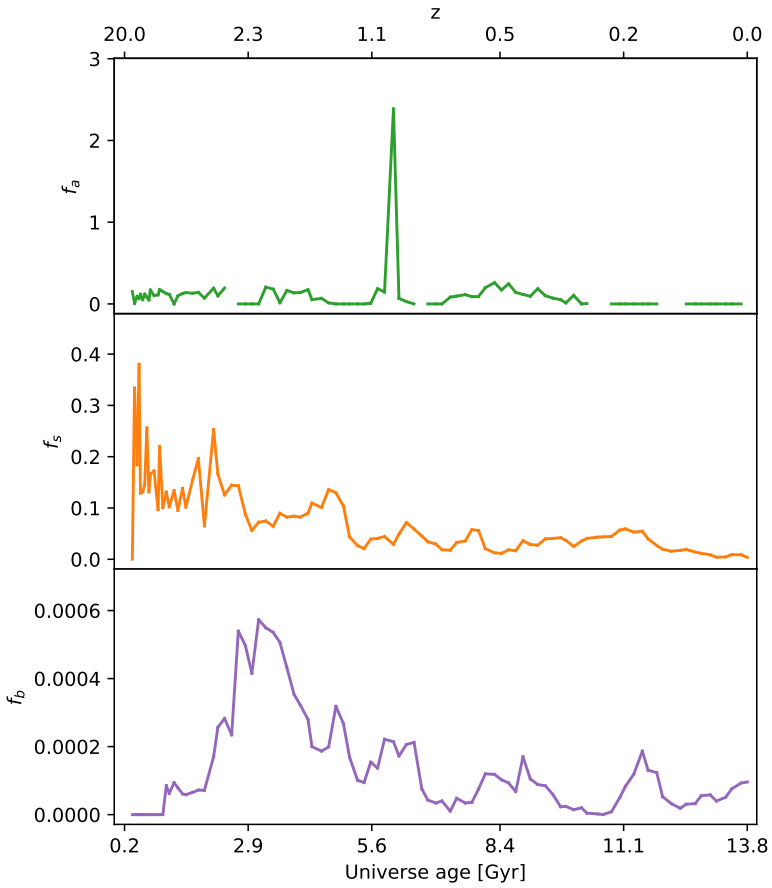

Figure 4.3 *Efficiencies of the galaxy in the top right panel of Figure 4.1*

a significant fraction of halos until after $z = 1$. The impact of this on the symbolic regression equations is discussed in Section 4.4.2.

Once the smoothing procedure has been applied to all galaxies, I calculate the efficiencies. For snapshots i and $i + 1$ which have corresponding times t_i and t_{i+1}, the efficiencies are calculated as

$$f_{\rm s} = \frac{M_*(t_{i+1}) - M_*(t_i)}{\Delta t\ M_{\rm gas}(t_i)} \tag{4.2}$$

$$f_{\rm b} = \frac{M_{\rm bh}(t_{i+1}) - M_{\rm bh}(t_i)}{\Delta t\ M_{\rm gas}(t_i)} \tag{4.3}$$

where $\Delta t = t_{i+1} - t_i$. For the calculation of $f_{\rm a}$ the gas that has been converted to

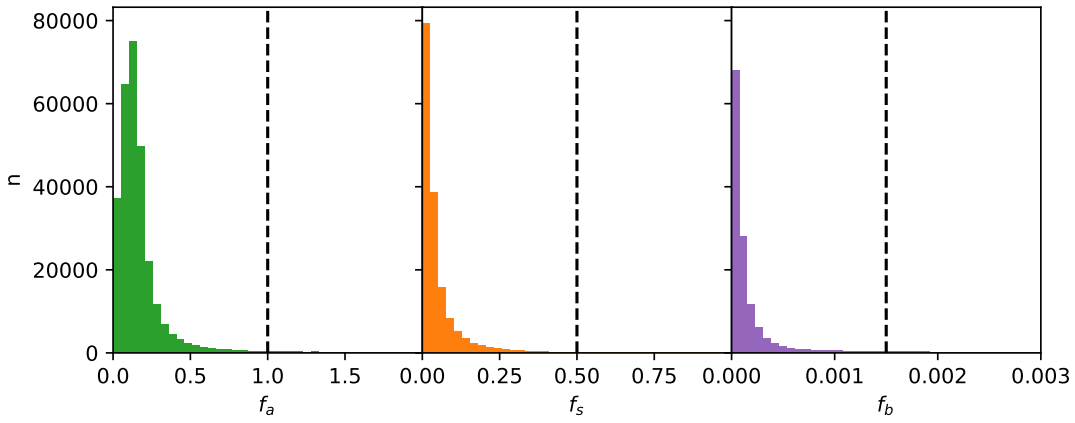

Figure 4.4 *Distribution of the efficiency values extracted from TNG100-1. The black dashed lines indicated the limits above which values are no longer used for training the machine learning models.*

stars or accreted to the black hole must be taken into account

$$f_{\rm a} = \frac{M_{\rm gas}(t_{i+1}) + (f_{\rm s} + f_{\rm b})\,\Delta t\,M_{\rm gas}(t_i) - M_{\rm gas}(t_i)}{M_{\rm h}(t_{i+1}) - M_{\rm h}(t_i)} \tag{4.4}$$

Figure 4.3 shows the efficiencies from the galaxy in the top right panel of Figure 4.1. Some values for efficiencies that are obtained are a results of numerical artifacts and are unphysical. An example of this is the spike in $f_{\rm a}$ which occurs around $z = 1$. It is not obvious from looking at the halo history, which is smooth, that such a spike should occur. These spikes result from snapshots where there is minimal accretion of dark matter mass. Prior to training my machine learning models I therefore remove any efficiency values with a value above $f_{\rm a} = 1$, $f_{\rm s} = 0.5$, and $f_{\rm b} = 0.0015$. The limits were determined by plotting the distribution of each efficiency and picking a natural truncation point within it, as shown in Figure 4.4.

4.2.3 Binning efficiencies

To get a baseline for how well the SAM is expected to match the hydro-simulation, I calculate the values of the efficiencies by binning them based on halo mass. To do this I split the data into a train and a test set, with 70% of the data used for training. For each snapshot I bin the galaxies based on their halo mass, and take the mean values of the efficiencies. I then model the evolution of galaxies in

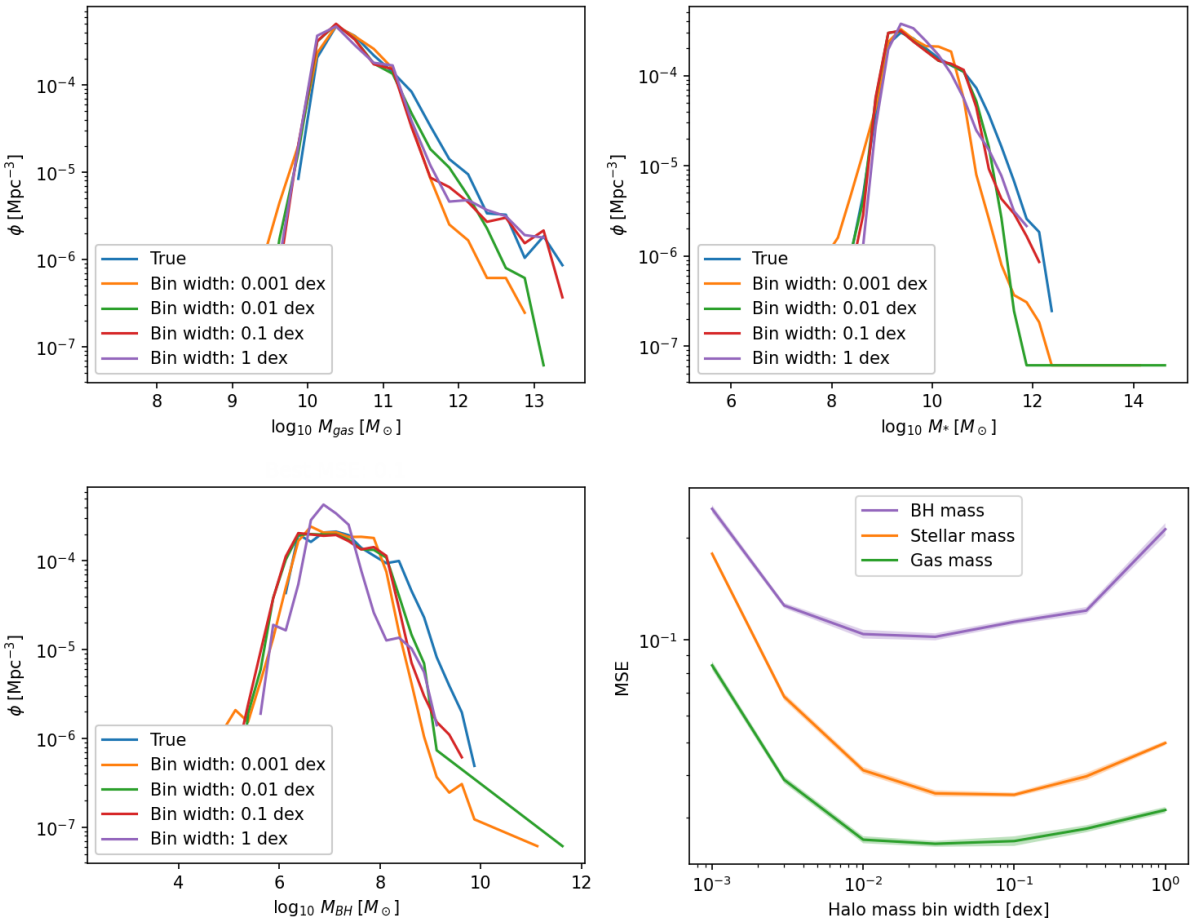

Figure 4.5 *Comparison of $z = 0$ galaxy population from TNG simulations with those calculated using efficiencies binned with different halo masses.* ***Top left*** *Gas mass,* ***Top right*** *Stellar mass,* ***Bottom left*** *BH mass. The text above the panels gives the best MSE achieved among the mass bin sizes used.* ***Bottom right*** *The MSE as a function of the bin size for each property*

the test set using the efficiencies calculated from the training set. I repeat this process for 10 train/test splits.

Figure 4.5 shows the resulting test set mass functions at $z = 0$, and also the MSE of each property. For TNG the most massive halos have the largest $f_{\rm a}$ values, so the binning procedure does not reproduce the the gas mass function at the high mass end. This means that stellar and black hole mass are also underestimated, as within this model their growth rate depends on the gas mass. The bottom right panel shows the effect of using different sized halo mass bins. The accuracy is poor for the largest halo mass bins. This is because averaging the efficiencies over a wide mass range fails to capture the different processes occurring within halos of varying sizes. The performance plateaus at around 0.1 dex, suggesting this is the point where galaxy-scale stochastic processes begin to dominate. Applying a different mass cut to the data does not significantly change the optimal bin size, suggesting this point is common across halo mass scales. For the smallest halo mass bin there is overfitting to the training set, since if no corresponding training set halo is found for a halo in the test set then I default to the median value of the efficiency over the entire box. This explains the decrease in performance. When I generate the plot shown in the bottom right panel for the training data then the MSE continues to decrease as bin size decreases.

4.3 Neural network

4.3.1 Method

For the machine learning predictions for this work I use a fully connected neural network. Neural networks consists of interconnected units called neurons, as shown in Figure 4.6. The neurons are arranged in layers, with each neuron in layer i being connected to every neuron in layer $i+1$. The inputs are connected directly to the first layer. Every connection between neurons has a weight associated with it, and each neuron has another parameter known as the bias. The output of a neuron is calculated by taking a linear combination of all its inputs, and adding the bias term. The output value is then passed to a nonlinear activation function such as tanh or ReLU. The parameters are tuned to improve the predictions as the model is trained. For each training example the network makes a prediction, and a loss function (such as MSE) is used to quantify the accuracy of the prediction.

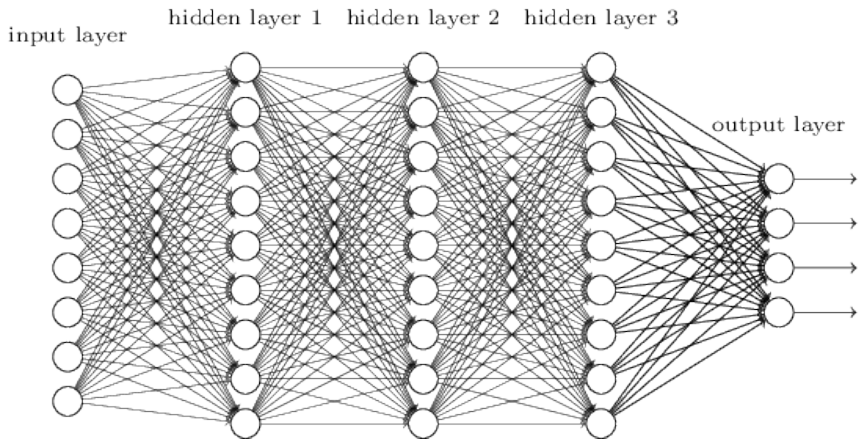

Figure 4.6 *Visualisation of a fully connected neural network, taken from Nielsen (2015).*

The partial differential of the loss function with respect to each of the input parameters is calculated by backpropagation, which consists of applying the chain rule multiple times. Once the gradient for a neuron is known then its value can be updated using a gradient descent based optimizer. Data is typically passed through the network in batches, and one complete pass through all training examples is known as an epoch. I use the PyTorch library (Paszke et al., 2019) to implement the neural network.

As with other machine learning methods, neural networks have a number of hyperparameters whose values need to be set before training. For the network architecture the hyperparameters to consider were the number of layers, and the number of neurons in each layer. For the training I varied the values of the optimizer used, the number of epochs to train for, and the batch size. The hyperparameters were selected using Bayesian optimization as discussed in Section 2.2.3.

Unlike tree-based algorithms, where the importance of an input feature can be determined by how often it is used to make a split, another method is needed for finding the importance of different input features for a neural network model. For this work I use saliency maps (Simonyan et al., 2013). This technique was introduced as a way to see what parts of images were causing convolutional neural networks to make their decisions. Within astrophysics it has been used for helping to understand classifiers of AGN (Peruzzi et al., 2021) and galaxy morphology (Bhambra et al., 2022). However, it can also be used for a multi-layer perceptron. The importance of an input feature for a single training example is given by the gradient of the output with respect to the input feature. This value is calculated

and averaged over the whole data set. The values are normalised such that the sum of all input features is equal to one.

4.3.2 Accuracy compared with binning data

As in the previous section when I was binning the efficiencies, I split data into a train and test set. I use this data to train a separate neural network for each of the three types of efficiencies.

I consider three sets of input properties to the model. For the dark matter only model I use the same properties as Chapter 2 (mass, velocity dispersion, maximum circular velocity), but not spin since it does not provide much predictive power. For the second model type I also include a number of baryonic properties (gas metallicity, stellar mass, gas mass, black hole mass). As this model takes in features not available in the one phase model, it is not possible to apply it to a dark matter only simulation. The purpose of this model is to get an idea of how accurate the predictions can be. The final model type only takes in halo mass and black hole mass as inputs, and is used for fitting symbolic regression. Halo mass is chosen as a subgrid-independent fundamental property of halos. It also allows for a comparison of how much improvement adding a second input feature provides compared with the binned efficiencies model. For the second input feature I choose black hole mass, since in the feature importance plots below it appears as a significant feature for all three efficiencies. Another reason, as discussed in Chapter 3, is that there are major differences in the subgrid modelling of black holes in various simulations, and the method in this chapter provides opportunities to gain understanding into the impact of these differences.

For each set of input properties I use two different approaches to provide information about redshift. For the first a single model is trained for all snapshots. This model takes redshift as an input feature. By examining the saliency map of this model it allows for the relative importance of time to be determined between each efficiency. The second approach is to consider four redshift bins $0 \leq z < 0.5$, $0.5 \leq z < 1.5$, $1.5 \leq z < 2.5$, and $2.5 \leq z$. The data is split and a separate model is trained for each bin. Redshift is not used as an input for these models. These models allow us to see how the importance of each property evolves over time.

Figure 4.7 shows the mass functions that result from the model which is trained using data from all snapshots and takes redshift and all input features as an

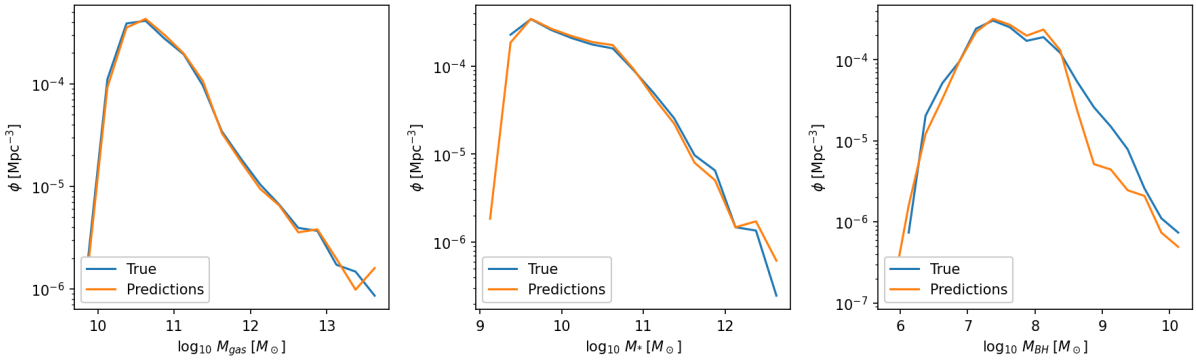

Figure 4.7 *True and predicted $z = 0$ mass functions of test data. Predictions of efficiencies were made using the model with the largest number of input features.*

Table 4.1 *$z = 0$ MSE scores calculated using the average of 10 test sets. The scores shown for the binned efficiencies correspond to optimal bin width. † indicates a separate model was trained for each of four redshift bins, otherwise a single model was trained with z as an input feature.*

	Gas Mass	Stellar Mass	BH Mass
Binned efficiencies	0.025	0.035	0.11
ML - Dark matter only	0.0074	0.012	0.046
ML - All inputs	0.0055	0.0082	0.030
ML - DM mass & BH mass	0.0073	0.015	0.035
ML† - DM mass & BH mass	0.0075	0.015	0.036

input. It is clear to see better agreement in the mass functions than those shown in Figure 4.5 which came from binned efficiencies. The neural network model is able to accurately model the mass function across the entire mass range apart from the largest black holes. This is likely due to the fact that these objects are growing through mergers, which are not always captured by the efficiency values.

The $z = 0$ MSE scores for the different models considered are shown in Table 4.1. Using a random forest as the algorithm yields similar MSE scores to the neural network. The last two rows of Table 4.1 show the performance of models which take halo mass and black hole mass as their input features. In one case I train a single model with z as an input, in the second I train a model for each redshift bin. The performance of the redshift binned models is slightly worse, but there is still reasonable agreement. For this reason when I fit the symbolic regression model I derive an equation for each redshift bin, since passing z as an input feature significantly complicates the equations that result. This decision is justified by the similarity in scores, as any accuracy advantage gained from the lower neural network MSE would be negated by the increased complexity of the equations. All

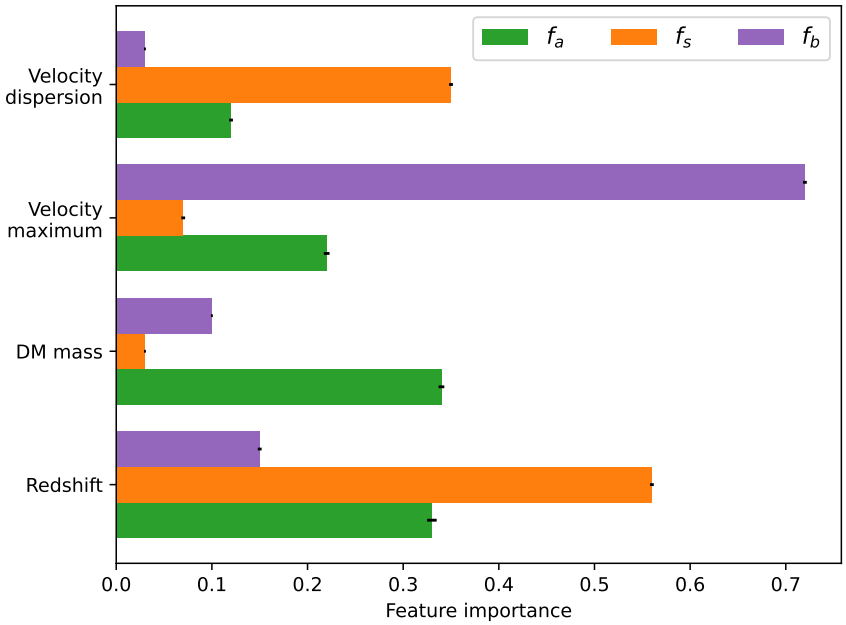

Figure 4.8 *Feature importance of model trained to predict f_a (**green**), f_s (**orange**), and f_b (**purple**). All models have redshift and dark matter only properties as input features.*

the machine learning models are considerably better than binning the efficiencies. For gas mass the predictions of the two feature and dark matter only models are similar, with the all input model outperforming both. The differences in scores are similar for the stellar mass MSE, but in this case the dark matter only model slightly outperforms the two feature model. Predictions of black hole mass show the opposite trend with the two feature model outperforming the dark matter only model.

4.3.3 Feature importance

Figure 4.8 shows the feature importance of a model trained over the full range of redshifts. The largest value for each input feature corresponds to different efficiency, indicating the models are picking up on different physical mechanisms. For f_a dark matter mass is the highest, but the importance is spread relatively evenly across all the input features. For velocity dispersion f_s is highest. f_s will be proportional to the amount of star forming gas in the halo, and so this result shows velocity dispersion is the best proxy for the amount of cold gas. For f_b velocity maximum in highest, possibly because this feature gives an indication of how concentrated the halo is. For the redshift we see the highest dependence for

f_s. This is likely a reflection of the importance of cold mode accretion which can occur at high redshifts. f_b has a low dependence on z, which suggests that the black hole growth is not limited by the available fuel, the growth rate depends on the galaxy properties. In general properties that are driven by large scale structure will have a large redshift dependence, properties that determined by internal processes will find halo-scale properties to be more important.

Figure 4.9 contains three panels, one for each efficiency being predicted. Within each panel there are four models for the redshift bins, using a range of input properties. These plots are useful for explaining the trends in MSE shown in Table 4.1. f_a shows gas mass as the most important feature, explaining why the all input model outperforms the others. There is a drop in gas mass importance over time. This is because if there is a lot of gas at $z = 3$ then f_a must be high, but that does not hold at later times. At later times an opposite effect might be at play, where a large amount of gas stops more being accreted due to pressure. The black hole mass importance does increase over time, but it's never dominant compared with other inputs.

In the middle panel we see stellar mass peaking at $z = 3$, as with gas mass for f_a, but it sharply drops after that point. The importance of dark matter mass decreases, as at early times all that matters is the size of the gravitational potential, whereas at later times the processes ongoing in the halo are more important. Within the exception of $z = 2$, which displays an unusual peak in velocity maximum, velocity dispersion is the most important feature. This explains why the dark matter only model makes better predictions for stellar mass than the two feature model. However, black hole mass is also an important feature, which increases with time. There is more variation in f_s for different redshifts (e.g. for $z = 3$ stellar and DM mass are important, but not at later times), agreeing with Figure 4.8.

For f_b there is high black hole importance at early times which then drops, similar to gas mass for f_a. This explains why the two feature model outperforms the dark matter only model when predicting black hole mass. Gas metallicity shows an increase with time, reflecting its importance in helping gas to collapse to the very centre of the galaxy.

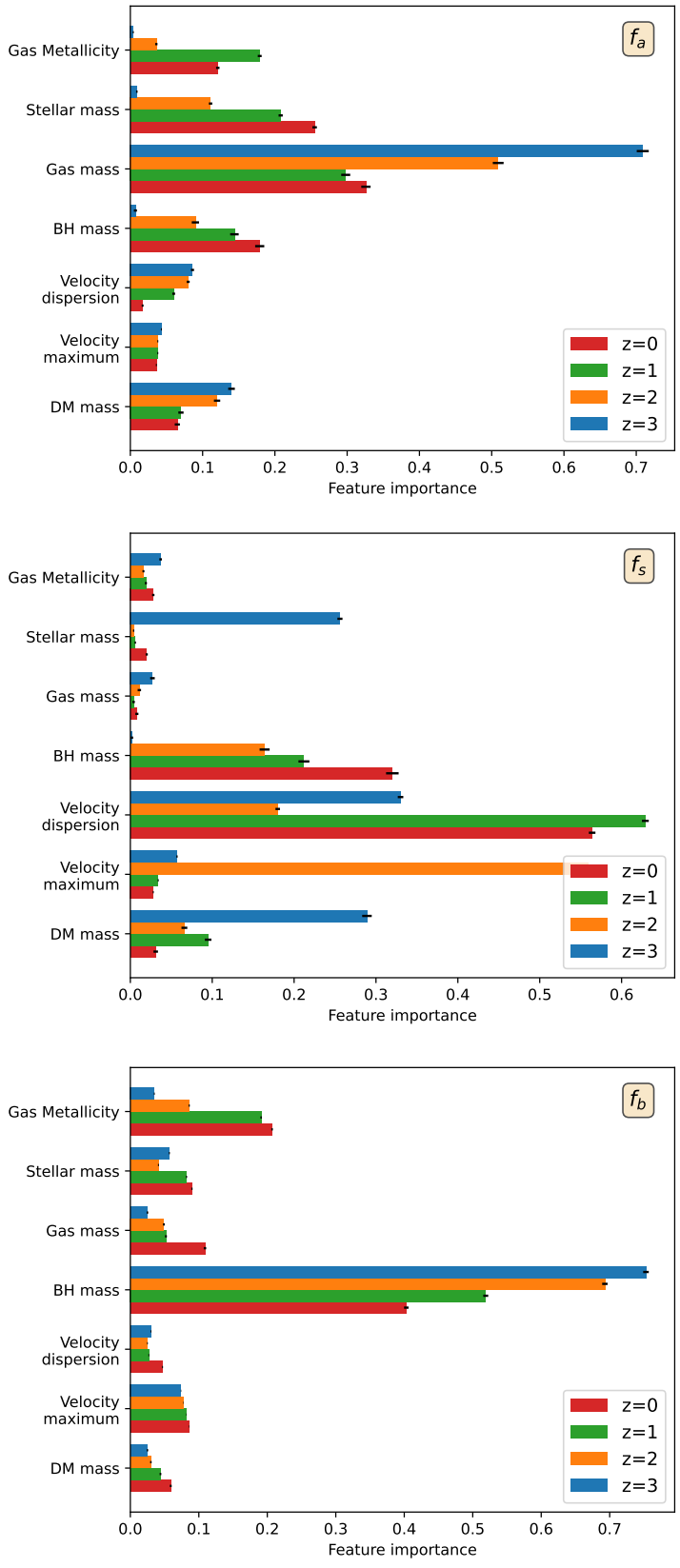

Figure 4.9 *Feature importance values of models trained at different redshifts.* ***Top*** *f_a prediction,* ***Middle*** *f_s prediction,* ***Bottom*** *f_b prediction*

4.4 Symbolic regression

4.4.1 Method

Symbolic regression is a machine learning approach which aims to discover the equation which best describes the relationship between a set of input variables and their corresponding output. It is an example of a solution to a supervised learning problem. The process begins by randomly generating a population of equations. Each formula is used to predict the data in the training set, and is assigned an accuracy score based on its performance. A selection step then takes place, where the best performing programs are given a higher likelihood of being picked. Those programs which are picked undergo an evolution step, where the equations are combine or mutated in order to produce new expressions. In this way symbolic regression is a stochastic optimization algorithm, which over time will tend to equations which minimize the loss function of the data being fitted. In this work I use the GPLEARN package (Stephens, 2015), and refer to hyperparameters using the same names as those employed in this library. Two initial hyperparameters are the *population_size*, which gives the number of equations within each generation, and the *n_generations*, the number of generations to run for. The process can also be stopped when an equation with a certain level of performance has been found.

The equations are represented as a syntax tree, as shown in Figure 4.10. Leaf nodes correspond to variables and constants, and all other nodes represent functions. This tree structure allows for equations to be easily combined, as any subtree can be replaced with another to produce a new valid equation. The functions available to act as interior nodes must be chosen before starting the fitting procedure. For this work I use the following functions - addition, subtraction, multiplication, division, sine, cosine, maximum, and minimum. Division is a protected operation, such that division by a number very close to zero will yield a value of 1.

The performance of the equations are given by evaluation of a standard metric, in this case the MSE. In order to prevent bloat, where equations in the population become larger over time, a second term is added to the fitness function to penalise

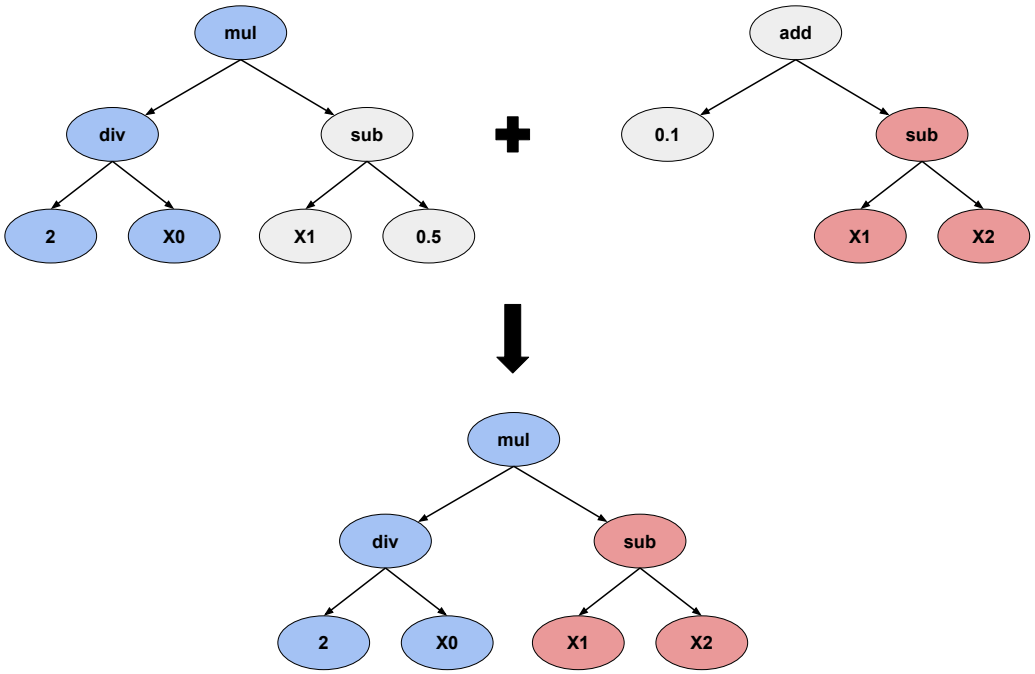

Figure 4.10 *Visualisation of crossover operation combining two equations. A single equation is represented by a tree structure, e.g. the top left equation corresponds to* $\frac{2}{x_0}(x_1 - 0.5)$

equations which are too long. The fitness of a function ϕ is therefore given by

$$\mathcal{F}(\phi) = MSE(\phi) + \sigma L(\phi) \tag{4.5}$$

where $L(\phi)$ is the number of nodes in the tree representation of ϕ, and σ is the hyperparameter known as the *parsimony_coefficient*. After the fitness of all equations has been determined a process of selection and evolution steps takes place until a new population has been generated. For the selection step a random subset of equations is chosen by uniformly sampling the population. The size of this subset is controlled by the *tournament_size* hyperparameter. The fittest individual from the subset is then selected for the evolution step. Setting *tournament_size* to a large value means poorly performing programs will be very rarely selected, and the population will converge in a short time. A small value will keep diversity in the population.

There are a number of operations to produce new trees. The operation to be used is chosen at random, although each operation is given a different weight. The most common operation is the crossover, as visualised in Figure 4.10. Two parent trees are required. A subtree from the first parent is randomly selected, and replaced

with a randomly selected subtree from the second parent. The mutation operation is similar to crossover, except the subtree inserted is randomly generated rather than being taken from an equation that exists in the population. The hoist mutation operation is used to trim the size of equations, and works by replacing a subtree with one of its leaf nodes. The final operation is point mutation, where a number of nodes are randomly replaced by another valid constant or function from the available set. There is also a small chance that reproduction will occur, where the selected equation is directly added to the next generation without any alteration.

Due to the nature of this method, a number of considerations need to be taken before beginning the fitting procedure. The constants that are initialised are set to have order ~ 1. Therefore, in order for the constants to be effective in improving the accuracy of the equations, the data most also have order ~ 1. I therefore apply standard scaling (subtract mean, divide by standard deviation) to the input and output data. As the space of equations is vast, symbolic regression is only viable for a small number of input features. As a result I do not attempt to use all the possible halo and galaxy properties as input, focusing instead on a subset.

Symbolic regression finds it difficult to deal with noise within a dataset. The common approach is to first train a neural network on the data, and then apply symbolic regression to approximate the neural network itself. I adopt this approach to help combat the inherent randomness in the efficiencies being predicted. Symbolic regression also finds discontinuities difficult to fit, which means tree-based methods such as random forests are not suitable for the initial fitting due to their discrete nature. Neural networks are better suited since their output is continuous. I use a ensemble of neural networks as the model to fit the symbolic regression to.

Symbolic regression can still perform well when fitting to a small number of observations (Wilstrup & Kasak, 2021), thus I do not fit the full set of training data (~6000 galaxies $\times$ 100 snapshots). I initially attempted to fit a random sample of the training data. However, the resulting equations provided a good match for low mass objects, but were inaccurate at the high mass end. Instead I construct a uniform 2D grid over the log parameter space with 900 points. I discard any points which are not nearby (within $\sqrt{2}$ dex) any data points in the training set. This procedure yields round ~ 300 observations which are then used to fit the symbolic regression. Decreasing the spacing of the grid does not lead to

an improvement in accuracy, but it increases the computational cost of the fitting procedure.

4.4.2 Equations resulting from SR

In this section I present the results obtained from the symbolic regression for TNG100-1. The equations are shown in the scaled feature space, so to get the full expression the standard scaling must be reapplied (multiply by standard deviation, add mean). Given a small enough value of the *parsimony_coefficient*, the resulting equations will always match the performance of the neural network. The value was varied to try and find a balance between accuracy and equation length, and so a constant value was not applied for all efficiencies and redshifts. For each set of equations presented here ten runs of symbolic regression were carried out. Not all runs converged on the same equations, as will be discussed further below. Degeneracy in sin and cos has been manually removed, for example in the case of f_{b}.

The sin and cos terms arise from the concentration of data points primarily at the centre of the range, with fewer points being fit at the edges. Consequently, the presence of these terms does not imply periodicity, it instead signifies a peak in the specific property. However, as discussed later, these peaks do have physical meanings, e.g. the peaks for the f_{b} prediction. As a general principle in machine learning, it is important to exercise caution when extrapolating beyond the confines of the training set.

Table 4.2 *$z = 0$ MSE scores from predictions of the TNG100-1 galaxy population calculated using the average of 10 test sets.*

	Gas Mass	Stellar Mass	BH Mass
Neural network	0.0075	0.015	0.036
Symbolic regression	0.0075	0.019	0.038

The result of applying the derived equations to the test data set is shown in Figure 4.11. The MSE scores from symbolic regression are shown in Table 4.2. For gas mass predictions the equations match the performance of the neural network. The scores of each of the two methods are similar for stellar mass and black hole mass, but slightly worse for symbolic regression. Thus the equations are able to capture the evolution of galaxies nearly as well as the neural network, despite having far less parameters.

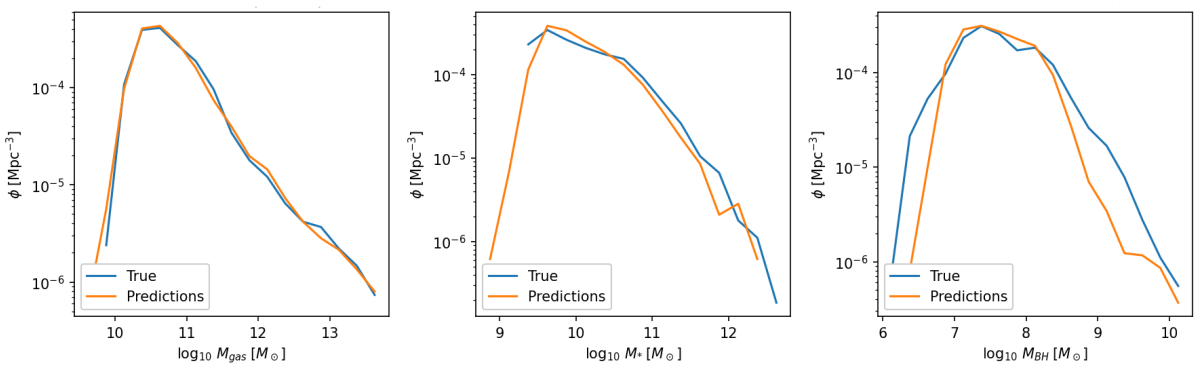

Figure 4.11 *True and predicted $z = 0$ mass functions on 10 different test data splits. Predictions of efficiencies were made using the equations resulting from symbolic regression.*

$$f_{\mathrm{a}} = \begin{cases} 0.67M_{\mathrm{h}} - M_{\mathrm{bh}} & \text{if} \quad z \geq 2.5 \\ M_{\mathrm{h}} - M_{\mathrm{bh}} & \text{if} \quad 2.5 > z \geq 1.5 \\ 1.5(M_{\mathrm{h}} - M_{\mathrm{bh}}) & \text{if} \quad 1.5 > z \geq 0.5 \\ M_{\mathrm{h}} - M_{\mathrm{bh}}(1 + \cos(2M_{\mathrm{h}})) & \text{if} \quad 0.5 > z \end{cases} \tag{4.6}$$

For f_{a} we see a clear structure in the equations, with black hole mass being subtracted from the halo mass, with some coefficients. As the halo mass increases the value of f_{a} increases. This is a result of an increased gravitational potential overcoming preventative feedback from star formation. As the black hole increases it has a clear negative effect on the ability of the halo to accrete. The coefficients in front of the two terms vary for the different redshift bins. However, one should be wary of assigning too much meaning to these coefficients given the standard scaling that takes place when the equations are applied. The $z = 0$ equation has an extra term compared with the other redshift bins. This could be a result of the smoothing procedure preventing the gas mass from dropping, as the $z = 0$ f_{b} equation also has an additional term compared with the other bins.

$$f_{\mathrm{s}} = \begin{cases} \sin(M_{\mathrm{h}} + 0.55) + \cos(M_{\mathrm{bh}}(M_{\mathrm{bh}} - 0.2)) - 0.85 & \text{if} \quad z \geq 2.5 \\ \min(1.5M_{\mathrm{h}} + 0.37,\ \cos(M_{\mathrm{bh}} \max(-0.9, M_{\mathrm{bh}}))) & \text{if} \quad 2.5 > z \geq 1.5 \\ M_{\mathrm{h}} - M_{\mathrm{bh}} \cos(M_{\mathrm{bh}} + \min(0.85, M_{\mathrm{bh}})) & \text{if} \quad 1.5 > z \geq 0.5 \\ \min(0.65, M_{\mathrm{h}}) - M_{\mathrm{bh}} & \text{if} \quad 0.5 > z \end{cases} \tag{4.7}$$

There are a number of differences between the equations for f_s compared with f_a. The first is length. Simpler equations were not capable of getting close to matching the accuracy of the neural network. Even with the extra fitting the performance is not equal the neural network, unlike the f_a predictions. The second difference is the lack of consistency between the equations at different redshifts. There are some commonalities, both the higher redshift bins feature a term which depends on the square of the black hole mass, suggesting it will help shut down star formation. The lower redshift equations both have a form similar to f_a with a M_{bh} term being subtracted from an M_h. However, it is difficult to obtain any direct insight into what is going on in the simulation. This is likely a reflection of Table 4.1, where the performance of the model with dark matter inputs is better than the one with halo mass and black hole mass, as discussed in Section 4.3.3. Unlike the f_a and f_b equations, where almost all runs came up with the same equations, perhaps with minor differences in the values of the coefficients, for f_s most of the equations appeared only two or three times. This again suggests that no suitable underlying description can be found given the current input parameters.

$$f_b = \begin{cases} \cos(2M_{bh} - 0.82) & \text{if} \quad z \geq 2.5 \\ \cos(2M_{bh} - 0.3) & \text{if} \quad 2.5 > z \geq 1.5 \\ \cos(2.3M_{bh}) & \text{if} \quad 1.5 > z \geq 0.5 \\ \cos(2.4\min(0.4, M_{bh}) + 1.6) - \min(1.2M_h, -0.5) & \text{if} \quad 0.5 > z \end{cases} \tag{4.8}$$

As with f_a, the equations for f_b are consistent up to constants apart from the final redshift bin. The equations do not have any dependence on M_h, hence within this model there is still a requirement for black holes to be seeded into halos once they reach a certain mass. The cosine in the equations shows that f_b grows up until it hits a certain mass, and after that point it drops off. The increase in accretion is due to the larger potential well as the black holes become more massive. At some point accretion begins to hinder growth as feedback effects prevent gas from infalling. For each redshift bin I can calculate the physical value of this turnover point by applying scaling, e.g. for the $z \geq 2.5$ bin I calculate the physical value which corresponds to a scaled value of 0.41. The resulting values are $10^{7.82}M_\odot, 10^{7.75}M_\odot, 10^{7.76}M_\odot$ for each of the first three bins. For the first

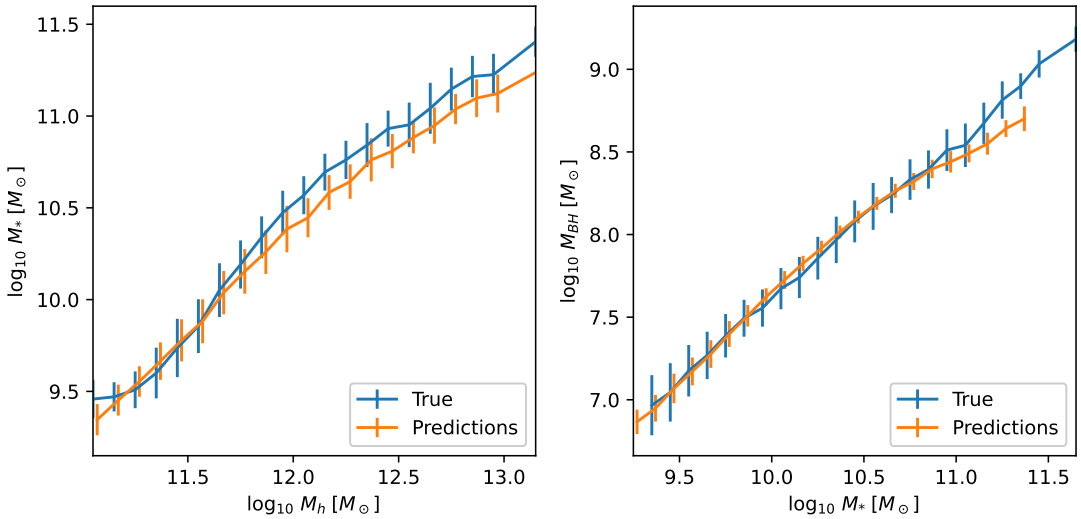

Figure 4.12 *True and predicted $z = 0$ relations. Errorbars indicate the standard deviation of each bin.* ***Left*** *Stellar-halo mass relation* ***Right*** *Black hole-stellar mass relation. Predictions of efficiencies were made on the test dataset using the equations resulting from symbolic regression*

term in the final redshift bin the transition mass corresponds to $10^{7.60} M_\odot$, but this value should not be taken as exact given the correction that appears from the second term. Thus the model has identified the redshift-independent point at which black hole accretion becomes efficient. In the TNG black hole feedback model (Weinberger et al., 2017) there is a transition from thermal to kinetic feedback if the accretion rate exceeds a fraction χ of the Eddington rate, with $\chi = \min(0.1, \chi_0 (M_{\rm bh}/10^8 M_\odot)^2)$. χ_0 is therefore degenerate with the pivot mass of $10^8 M_\odot$. My results suggests that kinetic feedback is activated in most SMBHs at $10^{7.8} M_\odot$, despite the fact that most accretion takes place in the thermal mode until a mass of $10^{8.5} M_\odot$ is reached (Weinberger et al., 2018).

In the left panel of Figure 4.12 I show the true and predicted stellar-halo mass relation. There is good agreement in the two relations at low masses, but galaxies are not growing large enough in the most massive halos. This can also be seen from the stellar mass function in Figure 4.11. At the lowest stellar masses the true and predicted curves diverge, but this is just an effect of the stellar mass cut which is applied when I select the halos from the simulation. A similar feature appears when considering the same plots for the lower resolution TNG run and for the original Illustris. The scatter in the relation is reproduced well by the symbolic regression method.

In the right panel the mean black hole-stellar mass relation matches well, but the

scatter is underestimated. Previous papers, which have applied a method similar to Chapter 2 to predict galaxy properties at a single redshift, have also found it difficult to reproduce the scatter in this relation (Kamdar et al., 2016b; Agarwal et al., 2018).

4.4.3 Effect of resolution

Table 4.3 *$z = 0$ MSE scores from predictions of the TNG100-2 galaxy population calculated using the average of 10 test sets. TNG100-2 (TNG100-2) equations shows the results from applying the symbolic regression expressions resulting from fitting the TNG100-2 (TNG100-1) data.*

	Gas Mass	Stellar Mass	BH Mass
Neural network	0.007	0.015	0.043
TNG100-2 equations	0.009	0.017	0.044
TNG100-1 equations	0.013	0.038	0.046

Table 4.3 shows the MSE scores when predicting the properties of galaxies in the TNG100-2 simulation, which has the same initial conditions and subgrid models as TNG100-1, but lower resolution. The first row shows the predictions from the neural network. The second row shows the performance of equations resulting from fitting the TNG100-2 data, and displays similar trends as seen in Table 4.2. The final row shows the MSE that results when applying the equations from the previous section (using the scaler from TNG100-1). The performance for black holes shows minimal variation, while for gas mass there exists a slightly higher difference, and the stellar mass shows a significant disparity.

$$f_{\rm a} = \begin{cases} (0.5M_{\rm h} - M_{\rm bh})\max(M_{\rm bh}, 0.65 - M_{\rm h}) & \text{if} \quad z \geq 2.5 \\ \min(M_{\rm h} - 2M_{\rm bh} + 0.5, \max(-M_{\rm h}, 0.4)) & \text{if} \quad 2.5 > z \geq 1.5 \\ (M_{\rm h} - M_{\rm bh})\max(0.6, 3M_{\rm bh}) & \text{if} \quad 1.5 > z \geq 0.5 \\ M_{\rm h} - M_{\rm bh} - 0.35 & \text{if} \quad 0.5 > z \end{cases} \tag{4.9}$$

The equations for $f_{\rm a}$ show the same general form as TNG100-1, with $M_{\rm h} - M_{\rm bh}$ appearing in all expressions. However, in this case in order to get close to the neural network performance some extra terms are required.

As with equations 4.7, the expressions resulting from fitting symbolic regression to the data for $f_{\rm s}$ do not converge, nor do they show consistency across redshift

bins. As a result I do not present them here, but the equations are similar in length to equations 4.7.

$$f_{\mathrm{b}} = \begin{cases} \cos(1.8M_{\mathrm{bh}} - 0.85) & \text{if} \quad z \geq 2.5 \\ \cos(2.2M_{\mathrm{bh}} - 0.2) & \text{if} \quad 2.5 > z \geq 1.5 \\ \cos(2.5M_{\mathrm{bh}} + 0.25) & \text{if} \quad 1.5 > z \geq 0.5 \\ \cos(2.4M_{\mathrm{bh}} + 1.3) + \max(0.8, -M_{\mathrm{h}}) & \text{if} \quad 0.5 > z \end{cases} \tag{4.10}$$

For the first three redshift bins the resulting equations for f_{b} have the same form as those from the higher resolution run. The TNG100-2 data has a higher average mass than TNG100-1 because in both cases the selection cut is based on the number of stellar particles. As a result the coefficients are different, but the resulting physical pivot mass is consistent for the two resolutions. This explains why using the TNG100-1 equations results in the same MSE. This shows that the IllustrisTNG subgrid model for black hole growth and feedback is robust to resolution, but this is not the case for the modelling of star formation.

4.4.4 Application to Illustris

Table 4.4 *$z = 0$ MSE scores from predictions of the Illustris galaxy population calculated using the average of 10 test sets.*

	Gas Mass	Stellar Mass	BH Mass
Neural network	0.0063	0.015	0.085
Symbolic regression	0.0071	0.019	0.091

Table 4.4 shows the performance when applying the method to the Illustris-1 galaxy population. A description of this simulation can be found in Section 3.2.1.

$$f_{\mathrm{a}} = \begin{cases} 0.3 - \sin(M_{\mathrm{h}}) + \max(0.1, -0.2M_{\mathrm{h}}) & \text{if} \quad z \geq 2.5 \\ \max(M_{\mathrm{h}}, -0.6) - \min(0.8, M_{\mathrm{h}}) - \max(-0.9, M_{\mathrm{bh}}) & \text{if} \quad 2.5 > z \geq 1.5 \\ M_{\mathrm{h}} - M_{\mathrm{bh}} - \min(0.6, M_{\mathrm{bh}} + 0.4) & \text{if} \quad 1.5 > z \geq 0.5 \\ \max(-0.6, 3.8(M_{\mathrm{h}} - M_{\mathrm{bh}})) & \text{if} \quad 0.5 > z \end{cases} \tag{4.11}$$

The first redshift bin shows a clear decrease in accretion rate with halo mass, with no effect from the black hole. The min and max functions in the second equation combine to give the same trend as the first equation, apart from for the largest objects. The final two redshifts bins depend on $M_{\rm h} - M_{\rm bh}$. Black holes are undersized in small halos for Illustris (Habouzit et al., 2022), so these bins also show a decrease in $f_{\rm a}$ with increasing mass.

$$f_{\rm b} = \begin{cases} \max(-1, 2M_{\rm bh} - M_{\rm h} - 0.2) & \text{if} \quad z \geq 2.5 \\ 0.25M_{\rm bh} + \max(-0.7, M_{\rm bh} - 0.25) & \text{if} \quad 2.5 > z \geq 1.5 \\ 1.7(M_{\rm bh} - M_{\rm h} + \min(1, M_{\rm h})) - 0.2 & \text{if} \quad 1.5 > z \geq 0.5 \\ M_{\rm bh} + (M_{\rm bh} + M_{\rm h} - 0.3)\cos(M_{\rm h}) - 0.3 & \text{if} \quad 0.5 > z \end{cases} \tag{4.12}$$

The $f_{\rm b}$ equations for Illustris and TNG are vastly different, despite the fact that both use Bondi accretion to determine the accretion rate. Therefore these differences must be due to the feedback implementation. Contrasting the TNG simulations which display a clear turnover point, for the first two redshifts bins of Illustris the accretion rate continues to grow over the entire mass range. For later times there is a reduction in accretion for the largest objects, but this is driven by the size of the halo rather than the black hole itself.

4.4.5 Discussion

Figure 4.13 shows the mean efficiency values as a function of halo mass, for both the Illustris and TNG simulations. When combined with the equations from the previous section, which provide information about the role of black holes, these plots yield insight into the baryon cycle in these simulations. In this section I discuss how these results compare with existing literature. It is worth noting that prior investigations in this area have predominantly concentrated on SPH simulations due to the difficulty of tracking gas in mesh-based simulations, as discussed earlier in this chapter.

The top row shows the gas accretion efficiency. Both TNG and the original Illustris display higher rates at high redshifts. For Illustris there is a linear decrease in $f_{\rm a}$ with halo mass. The low efficiency of supernova feedback in the Illustris simulations has been discussed in Chapter 3 with regards to its inability

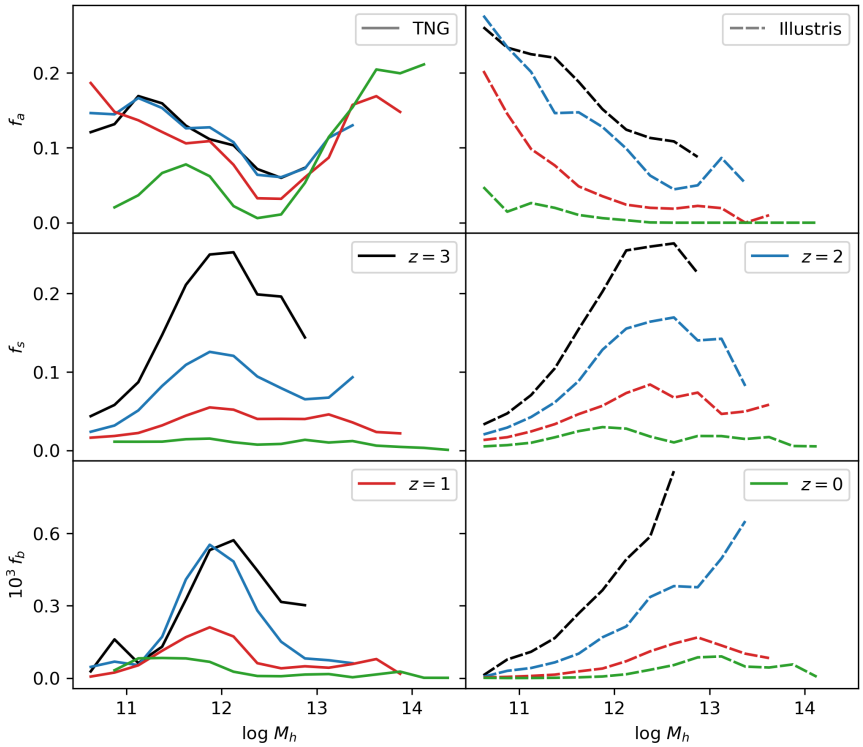

Figure 4.13 *Mean efficiency values against halo mass* ***Left column*** *TNG* ***Right column*** *Illustris*

to shut down star formation. This plot also shows it is not capable of preventing accretion. Kelly et al. (2022) carried out zoom simulations using the EAGLE and Auriga models. As with Illustris, they found that in Auriga gas accretion is almost unaffected by feedback. A similar result was also found by Pandya et al. (2020) for the Santa Cruz SAM. For the higher mass halos in Illustris the accretion efficiency does drop. This is because the presence of a massive hot gaseous coronae can exert pressure on the nearby gas and thus impede its accretion. The f_{a} data from the TNG shows a dip which begins around $10^{12}M_{\odot}$. From the equations in the previous section it is clear that this is a result of black hole feedback. This can also be seen be comparing with the f_{b} plot on the bottom row, which has a turnover point at $10^{12}M_{\odot}$. For the highest masses the value of f_{a} rises again, showing feedback is no longer effective at preventing accretion in this regime. This agrees with the results of Correa et al. (2018), who investigated the effect of black hole feedback on gas accretion in EAGLE and found a similar trend. The f_{a} values for the low mass halos are small for TNG when compared with Illustris. This is a result of preventative feedback delaying accretion, but its effect is to allow larger f_{a} values for the more massive halos (which are naturally the descendants of the low mass halos). This result also lines up with the EAGLE

simulations (Mitchell et al., 2020; Wright et al., 2020).

The trends seen for f_s are shaped by two factors: the cooling efficiency in different halo masses (as described in Section 1.1.2), and feedback. For low mass halos f_s appears similar for two simulations. This is surprising given that Illustris forms more stars in low mass halos, but we must take into account the amount of available gas in Illustris vs TNG due to f_a. This suggests that preventative feedback drives the decrease in star formation efficiency in TNG compared with Illustris, rather than processes going on in the ISM. There is inefficient radiative cooling at high masses, which explains why both simulations have a turnover point. However, the turnover appears at a lower mass for TNG, again lining up with the point where black hole feedback becomes efficient. This agrees with the results of Davies et al. (2020) who examined the $z = 0$ CGM mass fraction in TNG and EAGLE. Unlike supernovas, which appear to inhibit star formation primarily through preventative feedback, black hole feedback decreases star formation efficiency by slowing the collapse of gas from the CGM into the ISM.

For the TNG the derived equations for f_b are cosine terms with a dependence only on black hole mass. Due to the correlation between black hole and halo mass, it is possible to see this cosine shape in the bottom left panel of Figure 4.13. For Illustris the values of f_b also increase for low mass halos, but there is no turnover point. This is a result of the fact that the radio-mode feedback injects energy into the ICM (Sijacki et al., 2007; Vogelsberger et al., 2013), which means it does not prevent further accretion onto the SMBH.

Figure 4.13 shows a significant difference between simulations for all three efficiencies. This suggests that observations of gas flows within the CGM, such as the metallicity distribution, would be effective at breaking degeneracies within simulations. Another possibility would be to use the method presented in this paper with other combinations of input features other than halo mass and black hole mass. If the equations found were accurate at predicting the efficiency values, and were robust across simulations, they would be a valuable tool for helping to constrain galaxy simulations.

4.5 Conclusions and future work

My conclusions can be summarized as follows:

- I introduced a model to track the growth of gas, stars, and SMBHs within halos. The evolution of each phase is described by a set of coupled differential equations, where dark matter accretion is the original source term. The coefficients for these equations can be directly obtained from a cosmological simulation.
- I bin the efficiencies based on halo mass to obtain a baseline for how well the model can reproduce the galaxy population from a simulation. The performance initially improves with decreasing bin size, but plateaus for a bin size of 0.1 dex, indicating that below this point internal processes begin to dominate.
- Using a neural network to predict the efficiency values yields significantly better performance than binning based on halo mass. Saliency maps reveal major differences in input feature importance for each efficiency.
- I use symbolic regression to derive equations which predicting the efficiency values for IllustrisTNG. For $f_{\rm a}$ and $f_{\rm b}$ the equations are capable of matching the performance of a neural network which has the same input features. However, this is not the case for $f_{\rm s}$.
- Symbolic regression identifies the transition point between SMBH feedback modes as $10^{7.8} M_{\odot}$. This result is consistent across a range of redshift bins. Analysing the lower resolution TNG run results in similar equations for $f_{\rm b}$, showing the subgrid model for black hole growth and feedback is robust to resolution.
- I discuss the effect of supernova and AGN feedback on gas inflow rates within Illustris and TNG, and how this relates to other simulations.

There are a number of ways that the work presented in this chapter could be extended. If SPH simulations were analysed, then a wider range of efficiencies, such as those considered in Mitchell & Schaye (2021), could be examined. This would allow for modelling of the gas loss which occurs in halos at late times. For non-SPH simulations the gas could be split into cold and hot phases. However,

given the near constant conversion of cold gas to stars found in other work, the expressions found for f_s are unlikely to be drastically different. As mentioned in previous chapters, it would be beneficial to consider novel input properties, such as those which give environmental dependence, to see what impact they have on the performance of the models. This would be especially relevant for the f_s predictors. To consider the suitability of this method for populating N-body simulations, a direct comparison could be carried out with the model presented in Chapter 2. Even if the performance is worse, this model allows for certain features to be recovered that the other method does not, such as the SFH of a galaxy. In order to account for the outflows which occur at $z < 1$, symbolic regression could be used to predict the rate of change of each property directly, rather than predicting the efficiency values.

For changes to the method itself a template fitting step could be used, where after symbolic regression has found the form of equations then the coefficients are adjusted to find the minimal MSE. Another approach would be to use an alternative loss function. Since symbolic regression is not gradient-based, then the loss function does not need to be differentiable, so a loss function that depends on the mass functions could be used. In Tenachi et al. (2023) a method was introduced for finding equations while also considering dimensional analysis to ensure consistency of units, which was not the case for the equations shown here. Another possibility would be to train the symbolic regressor in two stages, first based on halo mass, then based on black hole mass, to help remove any correlations between the two.

Chapter 5

Conclusions

In this thesis I have demonstrated how machine learning methods can be used to augment cosmological simulations, thereby helping us study galaxy formation. The main themes of this work are the ability of machine learning to produce large data sets for comparison with observations, and to aid understanding of processes ongoing in simulations. This chapter briefly summarises the main conclusions of this thesis, and highlights potential future work arising from the different projects. Further discussion can be found at the end of each respective chapter.

Within Chapter 2 I focus on how machine learning can be used to help generate mock galaxy catalogs with a much greater volume than would be possible from running a full physics hydrodynamical simulation. The development of this approach is primarily motivated by the need for catalogs to compare with upcoming surveys. The large number of publications in this area over recent years highlights the interest in, and challenges of, mapping baryons to dark matter. I introduce a novel method of predicting the baryonic properties of subhalos from N-body simulations using machine learning. My model takes subhalo properties from a wide range of redshifts as input, and can be trained on any simulation with merger trees available. When compared with a baseline model that only uses $z = 0$ input features, the new model yields significantly more accurate predictions. It also outperforms a model which only uses the mass history of subhalos. Therefore future work in this area should make sure to include a variety of subhalo properties taken over a range of redshifts as input features. I then investigate the predictive power of each input property, mainly by looking at feature importance scores resulting from tree-based algorithms, although I show my results hold when

examining the model-agnostic MSE scores. Generating feature importance plots for a variety of output features allows me to infer information about how the different baryonic properties of a subhalo are determined, especially the redshift which is most important. By integrating the feature importance plots I show that for the IllustrisTNG simulations nurture is more important than nature in determining the properties of a galaxy. I then train a machine learning model using the IllustrisTNG300 simulations to predict the mass and accretion rate of $z = 3$ SMBHs based on their host halo properties. I apply this model to the Legacy N-body simulations, which results in a mock catalog with a volume of $(1\text{Gpc})^3$. There is good agreement in both the mass and luminosity distribution between the data for IllustrisTNG and the populated Legacy halos, indicating that the model has successfully learned an accurate mapping of the SMBH-halo connection. I compare the BHMF from the Legacy simulation with observed data at $z \sim 3$. This was the first time that this baryon painting method has been used to compare with observations. Having established the viability of this approach, it should now emerge as a standard method for comparing the outputs of simulations with observations. The mass functions match extremely well at the turnover point, but above and below this point the simulated data is considerably lower, indicating that IllustrisTNG is not accurately capturing accretion onto SMBHs at this epoch. Using the two-point correlation function I compare the spatial distribution of the simulated and observed data, and good agreement is found. Given this success I plot the number of faint black holes that can be expected to be found close to the brightest quasars, which is useful for informing observational strategies. Future work using machine learning to produce larger volume catalogs will likely consist of the introduction of subgrid models based on machine learning. The models will be trained on high resolution simulations but will fall back on more computationally expensive methods if they encounter data from outside the training set.

In Chapter 3 I look into how machine learning can help inform about differences in galaxy populations within simulations. I have introduced a novel method for extracting information about galaxy formation by extending the technique from Chapter 2. By considering the feature importance of baryonic properties it is possible to gain insights into the relative importance of different processes and the time at which they occur. I examine the impact of resolution, and find that decreasing the resolution has a clear effect on the feature importance, showing this novel method can be applied as a check for simulation convergence. When looking at galaxies in different density environments in IllustrisTNG I find cluster galaxies

produce stars at earlier times than those in low density regions, but the impact of black holes is decreased, results which are in agreement with observations. I show that differences due to subgrid models are considerably more significant than those introduced by modelling the gas using a moving-mesh instead of SPH. This can be seen by comparing the feature importance plots for EAGLE, Illustris, and IllustrisTNG. I demonstrate how the use of PCA in combination with feature importance values is capable of identifying physically meaningfully components when predicting stellar mass, finding one component which corresponds to the importance of the halo gravitational potential, and another component relates to the time when galaxy formation takes place. I show how the Simba black hole feedback model has a larger effect on galaxy formation than the IllustrisTNG model, but that stellar feedback remains the main driver in both. Through my analysis of the CAMELS simulations, I discover a substantial dependence between σ_8 and the time of galaxy formation. Given the current observational tensions in cosmological parameters, it is crucial for high-redshift simulations to consider this when comparing their results to JWST. This work is an example of how machine learning can help inform strategies about the best way to run future simulations by highlighting where discrepancies lie. In the future these kinds of methods will help to determine the observations required to constrain and distinguish between the galaxy formation models used within different simulations.

The work presented in Chapter 4 is the development of a model to track the growth of gas, stars, and SMBHs within halos. The evolution of each phase is described by a set of coupled differential equations, where dark matter accretion is the original source term. To determine the coefficients in these equations I first bin the data from TNG based solely on halo mass. The performance initially improves with decreasing bin size, but then plateaus, indicating that below this bin size internal processes begin to dominate. I then apply symbolic regression to the data from the simulation to derive expressions for the coefficients. This method yields significantly better performance than binning based on halo mass. Symbolic regression identifies the transition point between SMBH feedback modes, a result which is consistent across a range of redshift bins. Analysing the lower resolution TNG run results in similar equations for black hole accretion, showing the subgrid model for black hole growth and feedback is robust to resolution. I then discuss what the expressions can tell us about the effect of feedback on gas inflow rates within Illustris and TNG, and how this relates to other simulations. This work is an example of how the field of machine learning is advancing towards the point where it may be able to discover new physics from data alone.

Bibliography

Abdalla E., et al., 2022, Journal of High Energy Astrophysics, 34, 49

Agarwal S., Abdalla F. B., Feldman H. A., Lahav O., Thomas S. A., 2014, MNRAS, 439, 2102

Agarwal S., Davé R., Bassett B. A., 2018, MNRAS, 478, 3410

Agertz O., Kravtsov A. V., Leitner S. N., Gnedin N. Y., 2013, ApJ, 770, 25

Agnihotri A., Batra N., 2020, Distill

Ahumada R., et al., 2020, ApJS, 249, 3

Alonso Asensio I., Dalla Vecchia C., Potter D., Stadel J., 2023, MNRAS, 519, 300

Alpher R. A., Bethe H., Gamow G., 1948, Physical Review, 73, 803

Ananna T. T., et al., 2019, ApJ, 871, 240

Angelis D., Sofos F., Karakasidis T. E., 2023, Archives of Computational Methods in Engineering

Anglés-Alcázar D., Davé R., Faucher-Giguère C.-A., Özel F., Hopkins P. F., 2017, MNRAS, 464, 2840

Arun K., Gudennavar S., Sivaram C., 2017, Advances in Space Research, 60, 166

Ayromlou M., Nelson D., Yates R. M., Kauffmann G., Renneby M., White S. D. M., 2021, MNRAS, 502, 1051

Ayromlou M., Nelson D., Pillepich A., 2022, arXiv e-prints, p. arXiv:2211.07659

Barnes J., Hut P., 1986, Nature, 324, 446

Baron D., 2019, arXiv e-prints, p. arXiv:1904.07248

Bastian N., Covey K. R., Meyer M. R., 2010, Annual Review of Astronomy and Astrophysics, 48, 339

Baugh C. M., Gaztanaga E., Efstathiou G., 1995, MNRAS, 274, 1049

Behroozi P. S., Wechsler R. H., Wu H.-Y., 2013a, ApJ, 762, 109

Behroozi P. S., Wechsler R. H., Wu H.-Y., Busha M. T., Klypin A. A., Primack J. R., 2013b, ApJ, 763, 18

Bennett C. L., et al., 2013, ApJS, 208, 20

Benson A. J., Pearce F. R., Frenk C. S., Baugh C. M., Jenkins A., 2001, MNRAS, 320, 261

Bentz M. C., Peterson B. M., Netzer H., Pogge R. W., Vestergaard M., 2009, ApJ, 697, 160

Berlind A. A., Weinberg D. H., 2002, ApJ, 575, 587

Bhambra P., Joachimi B., Lahav O., 2022, MNRAS, 511, 5032

Bigiel F., et al., 2011, ApJ, 730, L13

Bishop C. M., 2006, Pattern Recognition and Machine Learning, 1 edn. Springer-Verlag

Blandford R. D., McKee C. F., 1982, ApJ, 255, 419

Blandford R., Meier D., Readhead A., 2019, ARA&A, 57, 467

Bluck A. F. L., Maiolino R., Brownson S., Conselice C. J., Ellison S. L., Piotrowska J. M., Thorp M. D., 2022, A&A, 659, A160

Bondi H., 1952, MNRAS, 112, 195

Booth C. M., Schaye J., 2009, MNRAS, 398, 53

Bosman S. E. I., et al., 2022, MNRAS, 514, 55

Boylan-Kolchin M., Ma C.-P., Quataert E., 2008, MNRAS, 383, 93

Braspenning J., Schaye J., Borrow J., Schaller M., 2023, MNRAS, 523, 1280

Breiman L., 2001, Mach. Learn., 45, 5–32

Breiman L., Friedman J. H., Olshen R. A., Stone C. J., 1984, Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA

Bryan G. L., et al., 2014, ApJS, 211, 19

Burbidge E. M., Burbidge G. R., Fowler W. A., Hoyle F., 1957, Reviews of Modern Physics, 29, 547

Carlesi E., Hoffman Y., Libeskind N. I., 2022, MNRAS, 513, 2385

Ceccarelli L., Duplancic F., Garcia Lambas D., 2022, MNRAS, 509, 1805

Chitre A., Jog C. J., 2002, A&A, 388, 407

Chittenden H. G., Tojeiro R., 2023, MNRAS, 518, 5670

Cole S., 1991, ApJ, 367, 45

Cole S., et al., 2001, MNRAS, 326, 255

Cole S., et al., 2005, MNRAS, 362, 505

Colless M., et al., 2001, MNRAS, 328, 1039

Collister A. A., Lahav O., 2004, PASP, 116, 345

Correa C. A., Schaye J., van de Voort F., Duffy A. R., Wyithe J. S. B., 2018, MNRAS, 478, 255

Cowie L. L., Gardner J. P., Hu E. M., Songaila A., Hodapp K. W., Wainscoat R. J., 1994, ApJ, 434, 114

Crain R. A., et al., 2015, MNRAS, 450, 1937

Cranmer M., 2023, arXiv e-prints, p. arXiv:2305.01582

Cui Y., Xiang Y., Rong K., Feris R., Cao L., 2014, in IEEE Winter Conference on Applications of Computer Vision. pp 213–219, doi:10.1109/WACV.2014.6836098

Curti M., et al., 2022, MNRAS, 512, 4136

DESI Collaboration et al., 2016, arXiv e-prints, p. arXiv:1611.00036

Dalla Vecchia C., Schaye J., 2012, MNRAS, 426, 140

Danovich M., Dekel A., Hahn O., Ceverino D., Primack J., 2015, MNRAS, 449, 2087

Davé R., Thompson R., Hopkins P. F., 2016, MNRAS, 462, 3265

Davé R., Anglés-Alcázar D., Narayanan D., Li Q., Rafieferantsoa M. H., Appleby S., 2019, MNRAS, 486, 2827

Davies J. J., Crain R. A., Oppenheimer B. D., Schaye J., 2020, MNRAS, 491, 4462

Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, ApJ, 292, 371

Dawson K. S., et al., 2016, AJ, 151, 44

De Lucia G., Weinmann S., Poggianti B. M., Aragón-Salamanca A., Zaritsky D., 2012, MNRAS, 423, 1277

De Lucia G., Hirschmann M., Fontanot F., 2019, MNRAS, 482, 5041

Dehnen W., Read J. I., 2011, European Physical Journal Plus, 126, 55

Dekel A., et al., 2009, Nature, 457, 451

Delgado A. M., Wadekar D., Hadzhiyska B., Bose S., Hernquist L., Ho S., 2022, MNRAS, 515, 2733

Di Valentino E., et al., 2021, Classical and Quantum Gravity, 38, 153001

Djorgovski S. G., Mahabal A. A., Graham M. J., Polsterer K., Krone-Martins A., 2022, arXiv e-prints, p. arXiv:2212.01493

Dubois Y., et al., 2014, MNRAS, 444, 1453

Eftekharzadeh S., et al., 2015, MNRAS, 453, 2779

Eide M. B., Ciardi B., Feng Y., Di Matteo T., 2020, MNRAS, 499, 5978

Eisenstein D. J., et al., 2005, ApJ, 633, 560

Eisert L., Pillepich A., Nelson D., Klessen R. S., Huertas-Company M., Rodriguez-Gomez V., 2022, arXiv e-prints, p. arXiv:2202.06967

Erhan D., Bengio Y., Courville A., Vincent P., 2009, Technical Report, Univeristé de Montréal

Fall S. M., Efstathiou G., 1980, MNRAS, 193, 189

Ferrarese L., Merritt D., 2000, ApJ, 539, L9

Ferreira L., Conselice C. J., Kuchner U., Tohill C.-B., 2022, ApJ, 931, 34

Fluke C. J., Jacobs C., 2020, WIREs Data Mining and Knowledge Discovery, 10, e1349

Gabrielpillai A., Somerville R. S., Genel S., Rodriguez-Gomez V., Pandya V., Yung L. Y. A., Hernquist L., 2022, MNRAS, 517, 6091

Gavazzi G., Pierini D., Boselli A., 1996, A&A, 312, 397

Geha M., Blanton M. R., Yan R., Tinker J. L., 2012, ApJ, 757, 85

Genel S., et al., 2014, MNRAS, 445, 175

Genzel R., Eckart A., Ott T., Eisenhauer F., 1997, MNRAS, 291, 219

Geurts P., 2006, Mach. Learn., 63, 3–42

Ghez A. M., Klein B. L., Morris M., Becklin E. E., 1998, ApJ, 509, 678

Gingold R. A., Monaghan J. J., 1977, MNRAS, 181, 375

Gómez J. S., Padilla N. D., Helly J. C., Lacey C. G., Baugh C. M., Lagos C. D. P., 2022, MNRAS, 510, 5500

Goto H., et al., 2021, ApJ, 923, 229

Grassi T., Nauman F., Ramsey J. P., Bovino S., Picogna G., Ercolano B., 2022, A&A, 668, A139

Grier C. J., Pancoast A., Barth A. J., Fausnaugh M. M., Brewer B. J., Treu T., Peterson B. M., 2017, ApJ, 849, 146

Grylls P. J., Shankar F., Leja J., Menci N., Moster B., Behroozi P., Zanisi L., 2020, MNRAS, 491, 634

Guglielmo V., Poggianti B. M., Moretti A., Fritz J., Calvi R., Vulcani B., Fasano G., Paccagnella A., 2015, MNRAS, 450, 2749

Gunn J. E., Gott J. Richard I., 1972, ApJ, 176, 1

Haardt F., Madau P., 2012, ApJ, 746, 125

Habouzit M., et al., 2022, MNRAS, 511, 3751

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

Hausen R., Robertson B. E., Zhu H., Gnedin N. Y., Madau P., Schneider E. E., Villasenor B., Drakos N. E., 2022, arXiv e-prints, p. arXiv:2204.10332

Hayward C. C., Torrey P., Springel V., Hernquist L., Vogelsberger M., 2014, MNRAS, 442, 1992

Hillebrandt W., Niemeyer J. C., 2000, ARA&A, 38, 191

Hirashima K., Moriwaki K., Fujii M. S., Hirai Y., Saitoh T. R., Makino J., 2023, arXiv e-prints, p. arXiv:2302.00026

Hirschmann M., Naab T., Somerville R. S., Burkert A., Oser L., 2012, MNRAS, 419, 3200

Hockney R. W., Eastwood J. W., 1988, Computer simulation using particles. CRC Press

Holwerda B. W., et al., 2022, MNRAS, 513, 1972

Hopkins P. F., 2015, MNRAS, 450, 53

Hopkins P. F., Quataert E., 2011, MNRAS, 415, 1027

Hopkins P. F., et al., 2018, MNRAS, 480, 800

Huang S., et al., 2019, MNRAS, 484, 2021

Hubble E., 1929, Proceedings of the National Academy of Science, 15, 168

Icaza-Lizaola M., Bower R. G., Norberg P., Cole S., Schaller M., Egan S., 2021, MNRAS, 507, 4584

Icaza-Lizaola M., Bower R. G., Norberg P., Cole S., Schaller M., 2023, MNRAS, 518, 2903

Ivanov M. M., Simonović M., Zaldarriaga M., 2020, J. Cosmology Astropart. Phys., 2020, 042

Ivezić Ž., et al., 2019, ApJ, 873, 111

Jennings W. D., Watkinson C. A., Abdalla F. B., McEwen J. D., 2019, MNRAS, 483, 2907

Jeon S., et al., 2022, ApJ, 941, 5

Jespersen C. K., Cranmer M., Melchior P., Ho S., Somerville R. S., Gabrielpillai A., 2022, ApJ, 941, 7

Jiang L., Helly J. C., Cole S., Frenk C. S., 2014, MNRAS, 440, 2115

Jiang F., et al., 2019, MNRAS, 488, 4801

Jo Y., Kim J.-h., 2019, MNRAS, 489, 3565

Jo Y., et al., 2023, ApJ, 944, 67

Johnson H. L., 1966, ARA&A, 4, 193

Johnson J. L., Dalla Vecchia C., Khochfar S., 2013, MNRAS, 428, 1857

Jolliffe I. T., Cadima J., 2016, Philosophical Transactions of the Royal Society of London Series A, 374, 20150202

Jung M., Kim J.-h., Kiat Oh B., Hong S. E., Lee J., Kim J., 2023, arXiv e-prints, p. arXiv:2312.02466

Kamdar H. M., Turk M. J., Brunner R. J., 2016a, MNRAS, 455, 642

Kamdar H. M., Turk M. J., Brunner R. J., 2016b, MNRAS, 457, 1162

Kannan R., Garaldi E., Smith A., Pakmor R., Springel V., Vogelsberger M., Hernquist L., 2022, MNRAS, 511, 4005

Kasmanoff N., Villaescusa-Navarro F., Tinker J., Ho S., 2020, arXiv e-prints, p. arXiv:2012.00186

Kaspi S., Smith P. S., Netzer H., Maoz D., Jannuzi B. T., Giveon U., 2000, ApJ, 533, 631

Kaspi S., Maoz D., Netzer H., Peterson B. M., Vestergaard M., Jannuzi B. T., 2005, ApJ, 629, 61

Kelly B. C., Shen Y., 2013, ApJ, 764, 45

Kelly A. J., Jenkins A., Deason A., Fattahi A., Grand R. J. J., Pakmor R., Springel V., Frenk C. S., 2022, MNRAS, 514, 3113

Kennicutt Robert C. J., 1998, ApJ, 498, 541

Kereš D., Katz N., Weinberg D. H., Davé R., 2005, MNRAS, 363, 2

Khochfar S., Silk J., 2011, MNRAS, 410, L42

Knebe A., et al., 2011, MNRAS, 415, 2293

Knebe A., et al., 2015, MNRAS, 451, 4029

Kochanek C. S., et al., 2001, ApJ, 560, 566

Kormendy J., Ho L. C., 2013, ARA&A, 51, 511

Kruk S. J., et al., 2018, MNRAS, 473, 4731

Krumholz M. R., Dekel A., 2012, ApJ, 753, 16

Krumholz M. R., Dekel A., McKee C. F., 2012, ApJ, 745, 69

Kugel R., et al., 2023, arXiv e-prints, p. arXiv:2306.05492

Lahav O., Lilje P. B., Primack J. R., Rees M. J., 1991, MNRAS, 251, 128

Landy S. D., Szalay A. S., 1993, ApJ, 412, 64

Li Y., Ni Y., Croft R. A. C., Di Matteo T., Bird S., Feng Y., 2021, Proceedings of the National Academy of Science, 118, e2022038118

Liddle A., 2003, An Introduction to Modern Cosmology, Second Edition. Wiley

Lovell C. C., Vijayan A. P., Thomas P. A., Wilkins S. M., Barnes D. J., Irodotou D., Roper W., 2021, MNRAS, 500, 2127

Lovell C. C., Wilkins S. M., Thomas P. A., Schaller M., Baugh C. M., Fabbian G., Bahé Y., 2022, MNRAS, 509, 5046

Lucie-Smith L., Peiris H. V., Pontzen A., 2019, MNRAS, 490, 331

Lucie-Smith L., Adhikari S., Wechsler R. H., 2022, arXiv e-prints, p. arXiv:2205.04474

Ludlow A. D., Schaye J., Schaller M., Bower R., 2020, MNRAS, 493, 2926

Ma W., Liu K., Guo H., Cui W., Jones M. G., Wang J., Zhang L., Davé R., 2022, ApJ, 941, 205

Machado Poletti Valle L. F., Avestruz C., Barnes D. J., Farahi A., Lau E. T., Nagai D., 2021, MNRAS, 507, 1468

Magorrian J., et al., 1998, AJ, 115, 2285

Marinacci F., et al., 2018, MNRAS, 480, 5113

Martig M., et al., 2013, MNRAS, 432, 1914

Martin J., 2012, Comptes Rendus Physique, 13, 566

Martinsson T. P. K., Verheijen M. A. W., Westfall K. B., Bershady M. A., Schechtman-Rook A., Andersen D. R., Swaters R. A., 2013, A&A, 557, A130

Massey R., Kitching T., Richard J., 2010, Reports on Progress in Physics, 73, 086901

Matchev K. T., Matcheva K., Roman A., 2022, ApJ, 930, 33

McAlpine S., et al., 2016, Astronomy and Computing, 15, 72

McGibbon R. J., Khochfar S., 2022, MNRAS, 513, 5423

McGibbon R. J., Khochfar S., 2023, MNRAS, 523, 5583

McInnes L., Healy J., Saul N., Großberger L., 2018, Journal of Open Source Software, 3, 861

Mehta P., Bukov M., Wang C.-H., Day A. G. R., Richardson C., Fisher C. K., Schwab D. J., 2019, Phys. Rep., 810, 1

Mistani P. A., et al., 2016, MNRAS, 455, 2323

Mitchell P. D., Schaye J., 2021, arXiv e-prints, p. arXiv:2103.10966

Mitchell P. D., et al., 2018, MNRAS, 474, 492

Mitchell P. D., Schaye J., Bower R. G., 2020, MNRAS, 497, 4495

Mitra S., Davé R., Simha V., Finlator K., 2017, MNRAS, 464, 2766

Miyoshi M., Moran J., Herrnstein J., Greenhill L., Nakai N., Diamond P., Inoue M., 1995, Nature, 373, 127

Mo H. J., Mao S., White S. D. M., 1998, MNRAS, 295, 319

Mo H., van den Bosch F. C., White S., 2010, Galaxy Formation and Evolution. Cambridge University Press

Moews B., Davé R., Mitra S., Hassan S., Cui W., 2021, MNRAS, 504, 4024

Morton B., Khochfar S., Wu Z., 2023, MNRAS, 518, 4401

Moster B. P., Somerville R. S., Maulbetsch C., van den Bosch F. C., Macciò A. V., Naab T., Oser L., 2010, ApJ, 710, 903

Moster B. P., Naab T., Lindström M., O'Leary J. A., 2021, MNRAS, 507, 2115

Murphy K. P., 2012, Machine learning: a probabilistic perspective, 1 edn. MIT Press

Naiman J. P., et al., 2018, MNRAS, 477, 1206

Nandra K., O'Neill P. M., George I. M., Reeves J. N., Turner T. J., 2006, Astronomische Nachrichten, 327, 1039

Natarajan P., 2014, General Relativity and Gravitation, 46, 1702

Natarajan P., et al., 2023, ApJ, 952, 146

Neistein E., Li C., Khochfar S., Weinmann S. M., Shankar F., Boylan-Kolchin M., 2011, MNRAS, 416, 1486

Neistein E., Khochfar S., Dalla Vecchia C., Schaye J., 2012, MNRAS, 421, 3579

Nelson D., Vogelsberger M., Genel S., Sijacki D., Kereš D., Springel V., Hernquist L., 2013, MNRAS, 429, 3353

Nelson D., et al., 2018, MNRAS, 475, 624

Nelson D., et al., 2019, MNRAS, 490, 3234

Netzer H., 2015, ARA&A, 53, 365

Ni Y., Li Y., Lachance P., Croft R. A. C., Di Matteo T., Bird S., Feng Y., 2021, MNRAS, 507, 1021

Ni Y., et al., 2023, arXiv e-prints, p. arXiv:2304.02096

Nielsen M. A., 2015, Neural networks and deep learning. Determination press, San Francisco, CA, USA

Nishiyama K., Nakai N., 2001, PASJ, 53, 713

Nojiri S., Odintsov S. D., Oikonomou V. K., 2017, Phys. Rep., 692, 1

Ochsenbein F., Bauer P., Marcout J., 2000, A&AS, 143, 23

Odewahn S. C., Stockwell E. B., Pennington R. L., Humphreys R. M., Zumach W. A., 1992, AJ, 103, 318

Oh B. K., An H., Shin E.-j., Kim J.-h., Hong S. E., 2022, MNRAS, 515, 693

Oppenheimer B. D., Davé R., 2008, MNRAS, 387, 577

Pakmor R., et al., 2022, arXiv e-prints, p. arXiv:2210.10060

Pallottini A., et al., 2022, MNRAS, 513, 5621

Pandya V., et al., 2020, ApJ, 905, 4

Paranjape A., Kovač K., Hartley W. G., Pahwa I., 2015, MNRAS, 454, 3030

Parodi B. R., Binggeli B., 2003, A&A, 398, 501

Paszke A., et al., 2019, arXiv e-prints, p. arXiv:1912.01703

Peebles P. J. E., 1980, The large-scale structure of the universe. Princeton University Press

Penzias A. A., Wilson R. W., 1965, ApJ, 142, 419

Perlmutter S., et al., 1999, ApJ, 517, 565

Peruzzi T., Pasquato M., Ciroi S., Berton M., Marziani P., Nardini E., 2021, A&A, 652, A19

Peterson B. M., 1993, PASP, 105, 247

Peterson B. M., 2014, Space Sci. Rev., 183, 253

Peterson B. M., Horne K., 2004, Astronomische Nachrichten, 325, 248

Peterson B. M., et al., 2004, ApJ, 613, 682

Pillepich A., et al., 2018a, MNRAS, 473, 4077

Pillepich A., et al., 2018b, MNRAS, 475, 648

Piotrowska-Karpov J., 2022, PhD thesis, University of Cambridge, doi:10.17863/CAM.89393

Piotrowska J. M., Bluck A. F. L., Maiolino R., Peng Y., 2022, MNRAS, 512, 1052

Planck Collaboration et al., 2014a, A&A, 571, A1

Planck Collaboration et al., 2014b, A&A, 571, A23

Planck Collaboration et al., 2016, A&A, 594, A13

Pugliese R., Regondi S., Marini R., 2021, Data Science and Management, 4, 19

Racca G. D., et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N., Tong E. C., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. p. 99040O (arXiv:1610.05508), doi:10.1117/12.2230762

Rasmussen C. E., Williams C. K. I., 2005, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, https://dl.acm.org/doi/10.5555/1162254

Reynolds C. S., 2020, arXiv e-prints, p. arXiv:2011.08948

Ricarte A., Natarajan P., 2018, MNRAS, 481, 3278

Riess A. G., et al., 1998, AJ, 116, 1009

Riess A. G., Casertano S., Yuan W., Bowers J. B., Macri L., Zinn J. C., Scolnic D., 2021, ApJ, 908, L6

Robertson B. E., 2022, ARA&A, 60, 121

Robles S., Gómez J. S., Ramírez Rivera A., Padilla N. D., Dujovne D., 2022, MNRAS, 514, 3692

Rodrigues N. V. N., de Santi N. S. M., Montero-Dorta A. D., Abramo L. R., 2023, arXiv e-prints, p. arXiv:2301.06398

Rodriguez-Gomez V., et al., 2015, MNRAS, 449, 49

Rubin V. C., Ford W. K. J., Thonnard N., 1980, ApJ, 238, 471

Russeil E., Ishida E. E. O., Le Montagner R., Peloton J., Moller A., 2022, arXiv e-prints, p. arXiv:2211.10987

Sarah Guido A. M., 2016, Introduction to Machine Learning with Python: A Guide for Data Scientists, 1 edn. O'Reilly Media

Scannapieco C., et al., 2012, MNRAS, 423, 1726

Schaller M., Dalla Vecchia C., Schaye J., Bower R. G., Theuns T., Crain R. A., Furlong M., McCarthy I. G., 2015, MNRAS, 454, 2277

Schaller M., et al., 2023, arXiv e-prints, p. arXiv:2305.13380

Schaurecker D., Li Y., Tinker J., Ho S., Refregier A., 2021, arXiv e-prints, p. arXiv:2111.06393

Schaye J., et al., 2015, MNRAS, 446, 521

Schaye J., et al., 2023, arXiv e-prints, p. arXiv:2306.04024

Schmidt M., 1959, ApJ, 129, 243

Secrest N. J., von Hausegger S., Rameez M., Mohayaee R., Sarkar S., Colin J., 2021, ApJ, 908, L51

Seyfert C. K., 1943, ApJ, 97, 28

Shao H., et al., 2022, ApJ, 927, 85

Shao H., et al., 2023, arXiv e-prints, p. arXiv:2302.14591

Shen Y., 2013, Bulletin of the Astronomical Society of India, 41, 61

Shen Y., Liu X., 2012, ApJ, 753, 125

Shepherd S. J., Zharkov S. I., Zharkova V. V., 2014, ApJ, 795, 46

Shi R., Wang W., Li Z., Han J., Shi J., Rodriguez-Gomez V., Peng Y., Li Q., 2022, MNRAS, 515, 3938

Shlens J., 2014, arXiv e-prints, p. arXiv:1404.1100

Sijacki D., Springel V., Di Matteo T., Hernquist L., 2007, MNRAS, 380, 877

Sijacki D., Vogelsberger M., Genel S., Springel V., Torrey P., Snyder G. F., Nelson D., Hernquist L., 2015, MNRAS, 452, 575

Silver D. Schrittwieser J. S. K. e. a., 2017, Nature, 550, 354–359

Simonyan K., Vedaldi A., Zisserman A., 2013, arXiv e-prints, p. arXiv:1312.6034

Smartt S. J., 2009, ARA&A, 47, 63

Smith M. J., Geach J. E., 2023, Royal Society Open Science, 10, 221454

Smoot G. F., et al., 1992, ApJ, 396, L1

Somerville R. S., Davé R., 2015, ARA&A, 53, 51

Springel V., 2005, MNRAS, 364, 1105

Springel V., 2010, MNRAS, 401, 791

Springel V., Hernquist L., 2003, MNRAS, 339, 289

Springel V., White S. D. M., Tormen G., Kauffmann G., 2001, MNRAS, 328, 726

Springel V., et al., 2005, Nature, 435, 629

Springel V., et al., 2018, MNRAS, 475, 676

Springel V., Pakmor R., Zier O., Reinecke M., 2021, MNRAS, 506, 2871

Stephens T., 2015, gplearn: Genetic Programming in Python, `https://gplearn.readthedocs.io`

Stinson G., Seth A., Katz N., Wadsley J., Governato F., Quinn T., 2006, MNRAS, 373, 1074

Stringer M. J., Brooks A. M., Benson A. J., Governato F., 2010, MNRAS, 407, 632

T. J. Hastie R. T., Friedman J. H., 2005, The elements of statistical learning: Data mining,inference, and prediction, 2 edn. Springer

Tenachi W., Ibata R., Diakogiannis F. I., 2023, arXiv e-prints, p. arXiv:2303.03192

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

Thomas D., Maraston C., Bender R., Mendes de Oliveira C., 2005, ApJ, 621, 673

Tin Kam Ho 1995, in Proceedings of 3rd International Conference on Document Analysis and Recognition. pp 278–282 vol.1, doi:10.1109/ICDAR.1995.598994

Trakhtenbrot B., Netzer H., 2012, MNRAS, 427, 3081

Tremaine S., Richstone D. O., Byun Y.-I., Dressler A., Faber S. M., Grillmair C., Kormendy J., Lauer T. R., 1994, AJ, 107, 634

Tremmel M., Karcher M., Governato F., Volonteri M., Quinn T. R., Pontzen A., Anderson L., Bellovary J., 2017, MNRAS, 470, 1121

Vale A., Ostriker J. P., 2004, MNRAS, 353, 189

Veilleux S., Cecil G., Bland-Hawthorn J., 2005, ARA&A, 43, 769

Vestergaard M., 2019, Nature Astronomy, 3, 11

Vestergaard M., Osmer P. S., 2009, ApJ, 699, 800

Vestergaard M., Peterson B. M., 2006, ApJ, 641, 689

Villaescusa-Navarro F., et al., 2021, ApJ, 915, 71

Villaescusa-Navarro F., et al., 2022, arXiv e-prints, p. arXiv:2201.01300

Villanueva-Domingo P., et al., 2021a, arXiv e-prints, p. arXiv:2111.08683

Villanueva-Domingo P., et al., 2021b, arXiv e-prints, p. arXiv:2111.14874

Vogelsberger M., Genel S., Sijacki D., Torrey P., Springel V., Hernquist L., 2013, MNRAS, 436, 3031

Vogelsberger M., et al., 2014a, MNRAS, 444, 1518

Vogelsberger M., et al., 2014b, Nature, 509, 177

Vogelsberger M., Marinacci F., Torrey P., Puchwein E., 2020, Nature Reviews Physics, 2, 42

Volonteri M., 2012, Science, 337, 544

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

Wadekar D., et al., 2023, MNRAS, 522, 2628

Wang J., Bose S., Frenk C. S., Gao L., Jenkins A., Springel V., White S. D. M., 2020, Nature, 585, 39

Wang F., et al., 2021, ApJ, 907, L1

Weinberger R., et al., 2017, MNRAS, 465, 3291

Weinberger R., et al., 2018, MNRAS, 479, 4056

Wells A. I., Norman M. L., 2021, ApJS, 254, 41

Wiersma R. P. C., Schaye J., Theuns T., Dalla Vecchia C., Tornatore L., 2009, MNRAS, 399, 574

Wilstrup C., Kasak J., 2021, arXiv e-prints, p. arXiv:2103.15147

Winkel N., Pasquali A., Kraljic K., Smith R., Gallazzi A., Jackson T. M., 2021, MNRAS, 505, 4920

Winkler P. F., Gupta G., Long K. S., 2003, ApJ, 585, 324

Wright R. J., Lagos C. d. P., Power C., Mitchell P. D., 2020, MNRAS, 498, 1668

Xu X., Ho S., Trac H., Schneider J., Poczos B., Ntampaka M., 2013, ApJ, 772, 147

Xu X., Kumar S., Zehavi I., Contreras S., 2021, MNRAS, 507, 4879

Yajima H., Abe M., Fukushima H., Ono Y., Harikane Y., Ouchi M., Hashimoto T., Khochfar S., 2022, arXiv e-prints, p. arXiv:2211.12970

Yang H., Gao L., Frenk C. S., Grand R. J. J., Guo Q., Liao S., Shao S., 2021, arXiv e-prints, p. arXiv:2110.04434

Yang T., Cai Y.-C., Cui W., Davé R., Peacock J. A., Sorini D., 2022, MNRAS, 516, 4084

Yip J. H. T., et al., 2019, arXiv e-prints, p. arXiv:1910.07813

York D. G., et al., 2000, AJ, 120, 1579

Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, arXiv e-prints, p. arXiv:1902.05965

Zwicky F., 1937, ApJ, 86, 217

de Andres D., Yepes G., Sembolini F., Martínez-Muñoz G., Cui W., Robledo F., Chuang C.-H., Rasia E., 2022, arXiv e-prints, p. arXiv:2204.10751

de Jaeger T., Stahl B. E., Zheng W., Filippenko A. V., Riess A. G., Galbany L., 2020, MNRAS, 496, 3402

van de Voort F., Schaye J., Booth C. M., Haas M. R., Dalla Vecchia C., 2011, MNRAS, 414, 2458

van den Bergh S., 2009, ApJ, 702, 1502

von Hippel T., Storrie-Lombardi L. J., Storrie-Lombardi M. C., Irwin M. J., 1994, MNRAS, 269, 97