Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **535,** 3324–3341 (2024) https://doi.org/10.1093/mnras/stae2560
Advance Access publication 2024 November 14

Predictions for the abundance and clustering of H α emitting galaxies

Makun S. Madar,★ Carlton M. Baugh ★ and Difu. Shi
Institute for Computational Cosmology, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK

Accepted 2024 November 7. Received 2024 October 14; in original form 2024 May 7

ABSTRACT

We predict the surface density and clustering bias of H α emitting galaxies for the *Euclid* and *Nancy Grace Roman Space Telescope* redshift surveys using a new calibration of the GALFORM galaxy formation model. We generate 3000 GALFORM models to train an ensemble of deep learning algorithms to create an emulator. We then use this emulator in a Markov Chain Monte Carlo (MCMC) parameter search of an eleven-dimensional parameter space, to find a best-fitting model to a calibration data set that includes local luminosity function data, and, for the first time, higher redshift data, namely the number counts of H α emitters. We discover tensions when exploring fits for the observational data when applying a heuristic weighting scheme in the MCMC framework. We find improved fits to the H α number counts while maintaining appropriate predictions for the local universe luminosity function. For a flux limited *Euclid*-like survey to a depth of 2×10^{-16} erg^{-1} s^{-1} cm^{-2} for sources in the redshift range $0.9 < z < 1.8$, we estimate 2962–4331 H α emission-line sources deg^{-2}. For a *Nancy Grace Roman* survey, with a flux limit of 1×10^{-16} erg^{-1} s^{-1} cm^{-2} and a redshift range $1.0 < z < 2.0$, we predict 6786–10 322 H α emission-line sources deg^{-2}.

Key words: methods: numerical – methods: statistical – galaxies: formation.

1 INTRODUCTION

Forecasting the performance of cosmological surveys plays a central role in planning the survey strategy and evaluating how trade-offs in depth and solid angle will affect the science goals. The wide field redshift surveys planned with *Euclid* (Laureijs et al. 2011; Euclid Collaboration 2024) and the *Nancy Grace Roman Space Telescope* (Spergel et al. 2015; Wang et al. 2022) will mainly sample H α emitters to map the cosmic large-scale structure. The figure-of-merit of cosmological probes that use galaxy clustering is dependent upon the number density and clustering strength of the galaxies being targeted (Albrecht et al. 2006). This is still relevant for *Euclid* post-launch, as the performance of the various detectors is assessed *in situ* and changes may be required to the survey strategy (Euclid Collaboration 2022). *Roman* is due for launch in 2027 May.

There are two routes to making this characterization of the redshift survey galaxies: exploiting existing studies of the target galaxy population to fit empirical models or using physically motivated models to predict the properties of the sample. Pozzetti et al. (2016) attempted to describe the H α luminosity function (LF) estimates available at the time using empirical models. Three empirical models were fit to the H α LFs measured using the *Hubble Space Telescope* (*HST*) Wide Field Camera 3 (WFC3) Infrared Spectroscopic Parallels (WISP; Colbert et al. 2013), Hi-Z Emission Line Survey (HiZELS; Geach et al. 2008; Sobral et al. 2009, 2012, 2013); and the *HST* Near Infrared Camera and Multi-Object Spectrometer (NICMOS; Shim et al. 2009).

The resulting simple functional forms for the H α LF can be integrated to obtain the number counts. The uncertainties were considerable, with the predicted surface density of H α emitters barely being constrained to within a factor of two.

Recently, with the addition of further space data, the situation has improved somewhat, and there have been renewed efforts to estimate the number of H α emitters that *Euclid* and *Roman* are likely to observe (e.g. Colbert et al. 2013; Mehta et al. 2015; Valentino et al. 2017; Merson et al. 2018; Zhai et al. 2019, 2021; Wang et al. 2022). Bagley et al. (2020) constructed a new data sample of line emitters from several *HST* surveys and forecast the properties of H α (and [O III]) emission-line galaxies for future surveys. The results from Bagley et al. (2020) show a clear preference for the so-called pessimistic model 3 from Pozzetti et al. (2016), which predicted the lowest surface density of emission-line galaxies (ELGs).

With a physical model, it is possible to predict the clustering of the galaxies as well as their abundance (see for example Orsi et al. 2010; Merson et al. 2019; Knebe et al. 2022; Reyes-Peraza et al. 2024). Merson et al. (2018) used the `Galacticus` semi-analytical model of galaxy formation (Benson 2012) to forecast the number density of H α emitters using a variety of dust attenuation models. Merson et al. (2018) predict 3900–4800 emitters deg^{-2} for the *Euclid* selection. However, in this case, `Galacticus` was calibrated to reproduce a variety of observational constraints with particular emphasis on the local Universe, without any explicit reference to ELGs. This situation was rectified in Zhai et al. (2019), in which `Galacticus` was recalibrated using a new *N*-body simulation simulation, the UNIT run (Chuang et al. 2019) and different calibration data, which included the H α luminosity function from HiZELS (Geach et al. 2008; Sobral et al. 2009, 2013).

★ E-mail: c.m.baugh@durham.ac.uk (CMB); makun.s.madar@durham.ac.uk (MSM)

Efficient calibration and exploration of galaxy formation models have been investigated in several papers, typically in two forms: a direct exploration of the model parameter space, running the full simulation for each set of parameters, and emulation or interpolation, in which the full calculation is mimicked by a cheaper process. Despite semi-analytical models (SAMs) being vastly cheaper to run than hydrodynamic simulations, direct exploration of their parameter space is still computationally expensive due to the large number of model evaluations required for an extensive search.

Direct exploration examples include Kampakoglou, Trotta & Silk (2008), who used Markov Chain Monte Carlo (MCMC) to calibrate a SAM to multiple data sets. MCMC was used again in Henriques et al. (2009) to calibrate their SAM, where they found that the choice of calibration data set changed the values of the best-fitting parameters, pointing to deficiencies in their model. Lu et al. (2011, 2012) constrained the parameter space for their SAM using Bayesian inference to achieve acceptable fits to the K-band LF; this was expanded to include the H I mass function in Lu et al. (2014) (see also Martindale et al. 2017). Ruiz et al. (2015) employed a stochastic technique called particle swarm optimization (Kennedy & Eberhart 1995) to calibrate the SAG SAM (Springel et al. 2001; Cora 2006; Lagos, Cora & Padilla 2008; Padilla et al. 2014; Gargiulo et al. 2015) to the K-band LF.

The second class of calibration involves building a statistical emulator of the SAM which can be evaluated much faster than running the full model, with the drawback of this being approximate by nature. Bower et al. (2010) and Vernon, Goldstein & Bower (2010) constructed a Bayesian approximation technique (described in Goldstein & Wooff 2007) to the GALFORM model that can be rapidly evaluated at any point in parameter space to provide reasonable fits to the K- and b_J-band LFs. This work was extended in Benson & Bower (2010) to explore how adaptable this reduced parameter space was to fit further observational data sets, and in Rodrigues, Vernon & Bower (2017) to calibrate GALFORM to the local galaxy stellar mass function. Elliott, Baugh & Lacey (2021) used a deep learning algorithm to emulate GALFORM across a range of output statistics. Elliott et al. were able to run many simple MCMC chains to explore the parameter space and investigate how calibration to different data sets constrained the model parameters. The emulation method can cope with a high-dimension parameter space.

Building on Elliott et al. (2021), we extend the calibration of GALFORM to forecast the number counts of Hα emitters and their clustering bias. We emulate GALFORM in the PLanck Millennium N-body simulation (Baugh et al. 2019; hereafter PMILL). We use deep learning to build an emulator: this allows us to build flexible function approximators that can reveal non-linear relations within data without needing a pre-defined model. There have been many successful uses of deep learning in astronomy (e.g. Ravanbakhsh et al. 2016; Schmit & Pritchard 2018; Cranmer et al. 2019; He et al. 2019; Ntampaka et al. 2019; Perraudin et al. 2019; Zhang et al. 2019; de Oliveira et al. 2020). We demonstrate the accuracy that can be achieved with deep learning when emulating GALFORM for the Hα number counts. We can use a moderate number of training runs to achieve good accuracy when compared to other calibration methods outlined above. As a deep learning emulator can be evaluated much more rapidly than running GALFORM, we can run many MCMC chains to explore the parameter space and identify the range of parameters that fit the calibration data sets. We achieve this by minimizing the absolute error between the emulator output and the observational data sets, employing a heuristic weighting scheme to the various observational data sets. This automation of the model calibration allows us to exhaustively search the parameter space.

The layout of this paper is as follows: We present the theoretical background in Section 2 and present the data sets relevant to this work. In Section 3, we present our results. In Section 3.1, we review the generation of the training and testing data, in Section 3.2, we illustrate the predictive performance of the emulator, and in Section 3.3, we show the results of the model exploration and calibration and the results for the Hα number counts and galaxy bias predictions. Finally, in Section 4, we review the merits of our methods and outline potential future avenues. We assume a Λcold dark matter cosmology with $\Omega_{\rm M} = 0.307$, $\Omega_\Lambda = 0.693$, and $H_0 = 67.77\ {\rm km\,s^{-1}\,Mpc^{-1}}$.

2 GALAXY FORMATION MODEL AND CALIBRATION DATA

We give an overview of GALFORM (Section 2.1), then in Section 2.2 we give a brief review of deep learning and describe the emulator design, and in Section 2.3 we discuss how we find best-fitting parameters using MCMC. In Section 2.4, we outline the generation of training and testing data for the emulator and describe the observations used in the calibration.

2.1 GALFORM

GALFORM is a physically motivated semi-analytical galaxy formation model (Cole et al. 2000; Bower et al. 2006; Lacey et al. 2016). GALFORM populates the DM haloes at the earliest branches of the halo merger tree with hot baryonic gas and models the main physical processes behind the formation and evolution of galaxies using a set of coupled differential equations, including (i) the collapse and merging of DM haloes, (ii) the shock-heating and radiative cooling of gas inside DM haloes, leading to the formation of galactic discs, (iii) quiescent star formation in galactic discs, (iv) feedback from SNe, active galactic nuclei (AGN), and photoionization of the intergalactic medium, (v) chemical enrichment of stars and gas, and (vi) dynamical friction driven by mergers of galaxies within DM haloes, forming spheroids and triggering starbursts. Note starbursts can also be driven by dynamically unstable discs. Full descriptions of these physical processes are given in Lacey et al. (2016) (see also the reviews by Baugh 2006 and Benson 2010).

GALFORM distinguishes between central and satellite galaxies within their host dark matter halo, with some of the physical processes being affected by this designation. Central galaxies are placed at the centre of the most massive subhalo and are the focus of all the gas that is undergoing cooling. Halo merger events choose the central galaxy of the main (most massive) progenitor halo as the central galaxy of the descendant halo with other galaxies becoming satellites. In the default gas cooling model (see Font et al. 2008 for an alternative model), satellite galaxies are stripped of their hot gas as soon as they become satellites, hence quenching any further cooling and stopping any long-term star formation. A hybrid scheme is used to predict when galaxy mergers occur (Simha & Cole 2017). Initially, the satellite galaxy's dark matter subhalo can be identified and tracked through the main halo. Once sufficient mass-loss has occurred such that the subhalo can no longer be resolved, an analytic estimate is made of the time required for the satellite to merger, as set out in Simha & Cole (2017).

Here, we give an overview of the processes in GALFORM that are explored. The model parameters varied are listed in Table 1.

Table 1. The GALFORM parameter space investigated assuming a uniform range for each parameter. See Section 2.1 for an explanation of how each process is modelled and the equations which involve each parameter. The first column gives the parameter name (and units if relevant), the second column gives the range over which the parameter is allowed to vary, and the third column lists the process to which the parameter relates.

Parameter	Range	Process
$\nu_{\rm SF}$ (Gyr^{-1})	0.1–4.0	Quiescent star formation
$V_{\rm SN,\,disc}$ (kms^{-1})	10–800	SN feedback
$V_{\rm SN,\,burst}$ (kms^{-1})	10–800	SN feedback
$\gamma_{\rm SN}$	1.0–4.0	SN feedback
$\alpha_{\rm ret}$	0.2–3.0	SN feedback
$F_{\rm stab}$	0.5–1.2	Disc instability
$f_{\rm ellip}$	0.2–0.5	Galaxy mergers
$f_{\rm burst}$	0.01–0.3	Galaxy mergers
$\tau_{*\,\rm burst,min}$ (Gyr)	0.01–0.2	Starbursts
$f_{\rm SMBH}$	0.001–0.05	SMBH growth
$\alpha_{\rm cool}$	0.0–4.0	AGN feedback

2.1.1 *Quiescent star formation in discs*

The quiescent mode of star formation takes place in the disc following the accretion of cooled gas from the hot halo. The star formation rate (SFR) in the disc is calculated using the empirical law inferred from observations by Blitz & Rosolowsky (2006) (as implemented in GALFORM by Lagos et al. 2011; see also Fu et al. 2010; Popping, Somerville & Trager 2014 for the incorporation of similar schemes into other SAMS) which is based on observations of nearby star-forming disc galaxies. The SFR is assumed to be proportional to the mass of the molecular component of the gas in the disc $M_{\rm mol,disc}$

$$\psi_{\rm disc} = \nu_{\rm SF} M_{\rm mol,disc}, \tag{1}$$

where $\nu_{\rm SF}$ is the value of the SFR coefficient, which controls the rate of conversion of the molecular gas into stars in quiescent galaxy discs. This is an adjustable parameter set within the range inferred from observations by Bigiel et al. (2011). The mass of molecular gas depends on the gas pressure in the mid-plane of the disc.

2.1.2 *Supernova feedback*

Supernovae (SNe; mainly Type II) eject gas from galaxies and their host dark matter haloes. The model, therefore, assumes the rate of gas ejection due to supernova feedback is proportional to the instantaneous SFR ψ, with a mass loading factor that is dependent on the galaxy circular velocity, $V_{\rm c}$, as a power law:

$$\dot{M}_{\rm eject} = \left(\frac{V_{\rm c}}{V_{\rm SN}}\right)^{-\gamma_{\rm SN}} \psi, \tag{2}$$

where $\gamma_{\rm SN}$ and $V_{\rm SN}$ are adjustable parameters. We can further split the $V_{\rm SN}$ term into $V_{\rm SN,\,disc}$ and $V_{\rm SN,\,burst}$ to distinguish the feedback contributions in quiescent star formation in discs from star formation in bursts. Most studies have assumed that these velocity normalization parameters are equal (e.g. Gonzalez-Perez et al. 2014 and Lacey et al. 2016). However, recent versions of the model have relaxed this restriction (e.g. Benson & Bower 2010; Elliott et al. 2021).

Gas ejected from the galaxy due to SN feedback is assumed to gather in a reservoir beyond the virial radius of the host dark matter halo. The gas gradually returns to the hot gas reservoir within the virial radius at a rate of

$$\dot{M}_{\rm return} = \alpha_{\rm ret} \frac{M_{\rm res}}{\tau_{\rm dyn,\,halo}}, \tag{3}$$

where $\tau_{\rm dyn,\,halo}$ is the halo dynamical time, $M_{\rm res}$ is the mass of the reservoir beyond the virial radius, and $\alpha_{\rm ret}$ is a free parameter. Note that the hot gas reservoir in the halo is assumed to have an r^{-2} density profile with a core.

2.1.3 *Galaxy mergers*

It is assumed when galaxies merge there may be a burst of star formation and destruction of the galactic discs. To define the type of merger, we set two thresholds, $f_{\rm ellip}$ and $f_{\rm burst}$. These thresholds are compared to the baryonic masses of the central galaxy, $M_{\rm b,cen}$, and the merging satellite galaxy, $M_{\rm b,sat}$ through the ratio $M_{\rm b,sat}/M_{\rm b,cen}$. When $M_{\rm b,sat}/M_{\rm b,cen} \geq f_{\rm ellip}$, the merger is classified as a *major* merger. After a major merger, the disc component of the primary galaxy is destroyed and forms a spheroid. We assume the cold gas in the disc is used up in a burst of star formation which also adds stars to the spheroid. The case for which $M_{\rm b,sat}/M_{\rm b,cen} < f_{\rm ellip}$ is a *minor* merger. For the cold gas in the disc to be consumed in a starburst after a minor merger, we require $M_{\rm b,sat}/M_{\rm b,cen} \geq f_{\rm burst}$. The stars accreted from the satellite galaxy are added to the spheroid of the central for all mass ratios. Both $f_{\rm ellip}$ and $f_{\rm burst}$ are free parameters. We use the prescription of Simha & Cole (2017) to compute the time for a galaxy merger to take place.

2.1.4 *Disc instabilities*

Disc instabilities can trigger star formation. When a galaxy is dominated by rotational motion the disc is unstable to bar formation through sufficient self-gravitation. We assume that discs are dynamically unstable to bar formation if the following condition is met (Efstathiou, Lake & Negroponte 1982)

$$F_{\rm disc} \equiv \frac{V_{\rm c}(r_{\rm disc})}{(1.68 G M_{\rm disc}/r_{\rm disc})^{1/2}} < F_{\rm stab}, \tag{4}$$

where $M_{\rm disc}$ is the total disc mass and $r_{\rm disc}$ is the disc half-mass radius. $F_{\rm disc}$ describes the contribution of disc self-gravity to its circular velocity, with larger values equating to lower self-gravity and greater disc stability. Predictions of $F_{\rm disc}$ vary depending on the method; Efstathiou et al. (1982) found $F_{\rm disc} \approx 1.1$ for a suite of exponential stellar disc models, while Christodoulou, Shlosman & Tohline (1994) found $F_{\rm disc} \approx 0.9$ for gaseous discs. For a completely self-gravitating stellar disc, $F_{\rm disc} = 0.61$. $F_{\rm stab}$ is a model parameter.

If the disc instability condition $F_{\rm disc} < F_{\rm stab}$ is met we assume the disc forms a bar which evolves into a spheroid (Combes et al. 1990; Debattista et al. 2006). We assume that an unstable disc is disrupted by bar instabilities on a subresolution time-scale thus all the mass is instantly transferred to the spheroid and any gas present is used in a burst of star formation.

2.1.5 *Starbursts*

We assume the rate at which bursts of star formation form stars in a spheroid is

$$\psi_{\rm burst} = \nu_{\rm SF,burst} M_{\rm cold,burst} = \frac{M_{\rm cold,burst}}{\tau_{*\,\rm burst}}, \tag{5}$$

where the time-scale $\tau_{*\,\rm burst}$ is

$$\tau_{*\,\rm burst} = \max[f_{\rm dyn}\tau_{\rm dyn,bulge},\, \tau_{*\,\rm burst,min}]. \tag{6}$$

The bulge dynamical time is defined in terms of the half-mass radius and circular velocity of the bulge, $\tau_{\rm dyn,bulge} = r_{\rm bulge}/V_{\rm c}(r_{\rm bulge})$. We treat $\tau_{*\,\rm burst,min}$ as a parameter, but fix $f_{\rm dyn} = 20$ (Lacey et al. 2016).

2.1.6 Stellar initial mass function and stellar population synthesis

We assume that quiescent star formation in galactic discs produced stars with a solar neighbourhood stellar initial mass function (IMF). Bursts of star formation (triggered by mergers or dynamically unstable discs) produce stars with a top-heavy IMF, with a power-law slope of $x = 1$ (see Lacey et al. 2016). We use the stellar population synthesis models of Maraston (2005).

2.1.7 SMBH growth and AGN feedback

Supermassive black holes (SMBH) can inject energy into the halo gas disrupting gas cooling. Multiple instances can lead to black hole growth; hot halo accretion, BH–BH mergers, and starbursts (Bower et al. 2006; Fanidakis et al. 2011; Griffin et al. 2019). In a starburst, mass accreted on to the SMBH is a constant fraction of the mass of stars formed, $f_{\rm SMBH}$, where $f_{\rm SMBH}$ is a parameter. AGN heating of the hot gas halo is assumed to occur if two conditions are met: (1) the gas halo is in quasi-hydrostatic equilibrium, i.e.

$$\tau_{\rm cool}/\tau_{\rm ff} > 1/\alpha_{\rm cool}, \tag{7}$$

where $\tau_{\rm cool}$ is the gas cooling time, $\tau_{\rm ff}$ is the free-fall time, and $\alpha_{\rm cool}$ is a parameter and (2) the AGN power required to balance the radiative cooling luminosity is less than $f_{\rm Edd}$ times the SMBH Eddington luminosity.

2.1.8 Emission lines

The star formation histories predicted by GALFORM are convolved with a simple stellar population model, which gives the light emitted as a function of age by a population of stars produced with a given stellar initial mass function and metallicity, building up a composite spectral energy distribution for each galaxy (see the review by Conroy 2013). This information is used to compute the number of ionizing photons, which, along with the metallicity of the cold gas in the interstellar medium is combined with an H II region model to compute the luminosity of the emission lines (see Cole et al. 2000; Baugh et al. 2022 for more extensive descriptions of the emission line models). For some predictions we combine the Hα and N[III] line luminosities, as these lines are close together in wavelength and will not be fully resolved by the surveys we consider.

Dust is assumed to be mixed with the stars in two forms: in clouds and a diffuse component (Granato et al. 2000). Dust properties are assumed and combined with the predicted scalelengths of the disc and bulge allowing the optical depth and attenuation of the starlight to be calculated as a function of wavelength. The emission lines are assumed to have the same attenuation due to dust as the stellar continuum at the same wavelength.

2.2 Deep learning emulator

We now describe the construction of an efficient emulator of GALFORM using `tensorflow` (Abadi et al. 2016). This is a supervised learning problem (also known as associative learning) in which the emulator is trained by providing it with inputs and matching outputs. We define the input vector x to represent a set of GALFORM model parameters and predict an output vector y, which consists of the binned statistical properties of the resulting synthetic galaxy population, for example, the galaxy luminosity function. The emulator aims to map the input vector x to the output vector y via an unknown function $\hat{f}(.)$ which replaces running the full GALFORM model at a fraction of the computational cost. The emulator allows us to thoroughly search a multidimensional model parameter space. The problem is one of regression where the outputs are binned floats rather than the probabilities that might be found in classification problems.

We use an artificial neural network to emulate GALFORM. The first layer of the multilayer network is the input layer with a size equal to the number of entries or components in x. In our case, this is the number of GALFORM input parameters used to make the predictions, with one neuron per feature. Note that these input parameters are the subset that is being varied; the full parameter space of the model is larger than the 11 parameters that we vary here, but the other parameters are held fixed (for the full list of parameters see Table 1 in Lacey et al. 2016). The final layer is the output layer with one neuron per prediction value. Here, the number of output neurons is the total number of bins across all of the chosen statistics. The middle layers of the network are known as hidden layers. The neurons in these layers extract features for mapping an input to an output and the network is trained by evaluating the hidden layer neurons using labelled examples, i.e. with the output from runs of GALFORM. Networks with multiple hidden layers are known as deep learning networks. The connections between each neuron have an associated weight, w, and each neuron has a bias, θ. A network learns by adjusting these weights and biases from exposure to the training examples according to some learning rule. Each neuron is a simple mathematical function taking a vector of inputs and calculating an output. The i-th neuron in the j-th layer contains a vector of adjustable weights $\mathbf{w}_{ij}$ and an adjustable bias θ_{ij}. The vector $\mathbf{w}_{ij}$ contains all the weights linking a neuron i to each neuron in the previous layer, $j-1$. The data flow from the input to output neurons is strictly passed forward and every neuron in each layer is connected to every neuron in the following layer in what is known as a fully connected network. Note there are no connections *within* a layer. The total input of neuron i in layer j is a function of the outputs from each neuron in layer $j-1$, y_{j-1}, the neuron vector weights $\mathbf{w}_{ij}$, and the bias of the neuron θ_{ij}. An activation function $F(.)$ takes the total input of the neuron to produce an output,

$$y_{ij} = F(\mathbf{w}_{ij} \cdot y_{j-1} + \theta_{ij}). \tag{8}$$

The activation function is often a non-decreasing function of the total input of the neuron, introducing non-linearity to the network and allowing for complex representations and functions to develop, which is not possible with a simple linear input–output model. The activation function transforms the output value of the neuron to within certain limits, modified based on the application of the model. If unrestricted by the activation function, the outputs of neurons can explode in magnitude in deeper networks. Generally, some sort of non-linear threshold function is used, such as a sigmoid or hyperbolic tangent function. The outputs of the neurons, y_{ij}, are passed to the following layers of neurons, and so on, until the final layer is reached. The output from the final layer is the network prediction y from input x. An activation function is still applied to the final layer but this is usually a linear function in the case of regression.

To adjust the weights assigned to hidden neurons, we use the back-propagation learning rule (Rumelhart, Hinton & Williams 1986). During training the predictions from the output layer are compared to the true values and the error between these two are back-propagated from the output layer to the hidden layers and their weights are adjusted accordingly to minimize an error function. Following Elliott et al. (2021), we minimize the mean absolute error function (MAE) between the emulator predictions of the GALFORM outputs and the

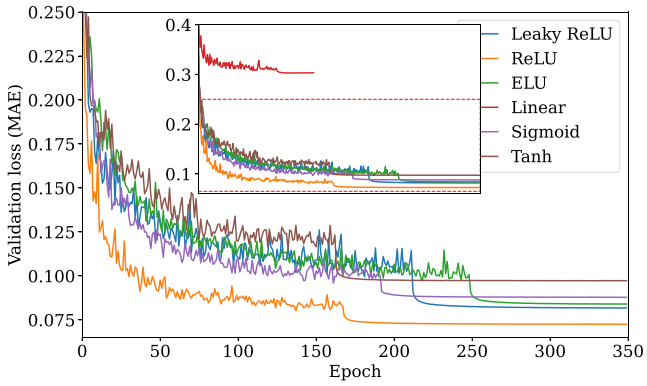

Figure 1. Testing the choice of activation function in the network. The MAE loss on the validation data set is plotted against the training epoch. A different colour is used for each choice of activation unit, as indicated by the key. Each network has the same architecture of two hidden layers, with 512 nodes and a linear output activation function. We display a zoomed-out inset to show the poor loss attained with a linear activation function. The sudden drop in loss value exhibited in all cases, when the curves also appear to become smoother, is due to the fine-tuning stage of training (see text for further details).

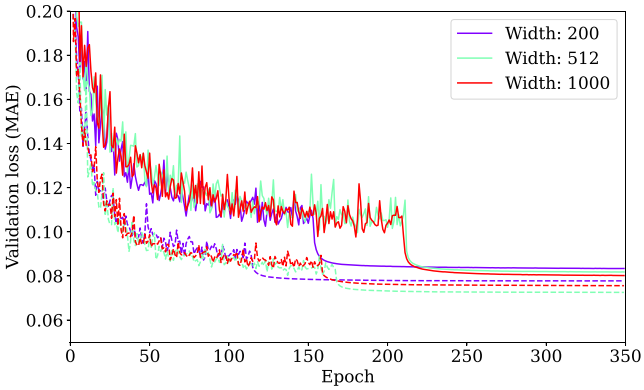

Figure 2. Measuring the MAE loss on the validation data set during training, when altering the hidden layer widths of our network for two activation functions, ReLU (dashed) and LReLU (solid). Each network has two hidden layers and a linear output activation function. There are no significant benefits to increasing the width of our network beyond 512 neurons per layer.

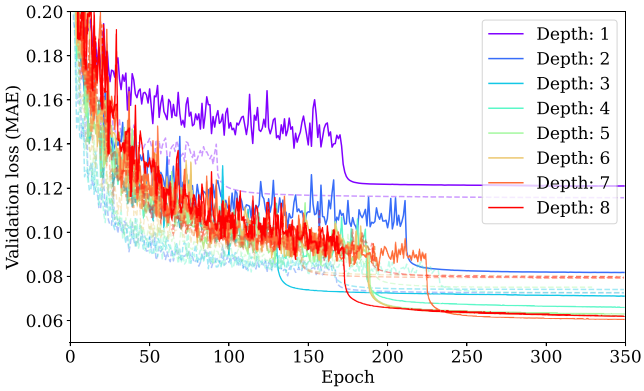

Figure 3. MAE validation loss when modifying the number of hidden layers in the network, with different colours indicating different numbers of layers, as shown by the key. We keep the width of the network fixed at 512 and show results for two activation functions, LReLU (solid) and ReLU (dashed). The LReLU function has greater potential for improvement than the ReLU networks. An increase in depth improves the performance of our network up to a depth of five or six layers. Beyond this, there is only a modest improvement in the MAE at the expense of an increase in the computational cost.

true outputs

$$\mathrm{MAE} = \frac{1}{n}\sum_{k=1}^{n} |\hat{y}_k - y_k|, \tag{9}$$

where $\hat{y}_k$ is the emulator prediction for the k-th sample out of n and y_k are the values computed by GALFORM for the same parameter values. The MAE is also known as the loss function and reveals how badly (or how well) the network is performing.

The neural network is trained iteratively over many epochs. One epoch is equivalent to the network cycling through every sample in the training set once; the number of training epochs is a user choice. An optimizer algorithm is used to change the weights and biases of the neural network by seeking minima on the error surface, often via a form of gradient descent. The optimizer also specifies the size of steps taken during the gradient descent towards the local minima, known as the learning rate. At the end of each epoch, the adjusted model is tested on a validation sample, which is a subset of the data that has not been used during the training to ensure the model is generalizable to completely unseen data. The number of training epochs is fixed by plotting the MAE against the epoch; this curve flattens off after some number of training epochs so that the precise choice of the number of epochs used is not important once this flat part of the MAE curve has been reached (see e.g. Fig. 1).

The final network is tested on a hold-out set of samples to carry out a performance analysis on completely unseen data (Section 2.4.1).

2.2.1 *Inputs and outputs*

We aim to develop an emulator to map an input vector x, which is the subset of GALFORM parameters that are allowed to vary, on to an output vector y, corresponding to the statistical galaxy properties we wish to predict. Our choice of the input parameters that are allowed to vary is made through physical intuition and guidance from previous analyses (see Section 2.4). These parameters and their ranges are shown in Table 1. We tune the emulator to predict three statistical properties calculated from the output of GALFORM to calibrate a model to make accurate predictions for *Euclid* and *Roman*: these are the redshift distribution of Hα emitters between $0.69 \leq z \leq 2$, and the local luminosity functions in the r and K bands (see Section 2.4.2 for more information about these data sets). Each data set is weighted equally in the metric when the emulator is being constructed.

2.2.2 *Network architecture*

The neural network architecture was determined by testing individual hyperparameter configurations. Taking inspiration from Elliott et al. (2021), we start with an architecture with two hidden layers, each containing 512 neurons with the sigmoid activation function on hidden layers, and linear activations on the output layer. Here, we test modifying the choice of activation function on hidden layers, the width of the network (the number of neurons per layer), and the depth of the network (the number of hidden layers). For the output layer, the linear activation function is consistently used, which is suitable given that the emulator is essentially a regression model. All networks are trained with the same data set. We track the MAE against the validation data set at each epoch during training and show the results in Figs 1, 2, and 3. We note there is a caveat with these tests due to the stochastic nature of training a neural network; an

identical network architecture trained on identical training data can display a small variability in its final validation score, so we take this into account when deciding on the final network.

Starting from the architecture used in Elliott et al. (2021), we modify the activation functions, testing a linear function, Logistic Sigmoid, Tanh, rectified linear unit (ReLU; Nair & Hinton 2010; Sun, Wang & Tang 2015), leaky ReLU (LReLU; Maas et al. 2013; Xu et al. 2015), and exponential linear unit (Clevert, Unterthiner & Hochreiter 2015), with the results displayed in Fig. 1 (for a full review of the many activation functions available see Dubey, Singh & Chaudhuri 2022). We found that modifying the activation function to a type of rectifier unit was the best option.

Next, we test both the ReLU and LReLU activation functions while modifying the width of our network but keeping the number of hidden layers at two. We consider 200, 512, and 1000 neurons per hidden layer. We want to see if there is a positive trend in terms of a reduction in the MAE when increasing the number of neurons per layer. In Fig. 2, we plot the results from both the LReLU (solid line) and the ReLU (dotted) network activation functions. There are training speed benefits to using a thinner network: the percentage increase in training speed for the network to reach epoch 350, between the thinnest network (width 200) and the widest network (width 1000) is $\sim$ 190 per cent for either activation function. We see that for both cases the 200-width network does not perform as well as the wider networks. However, there is no clear gain in performance to support increasing the width beyond 512 neurons. Therefore, we will use hidden layer widths of 512 to optimize the performance and training speed.

Finally, we test the depth of the network, that is, the number of hidden layers our network contains. Once again we train two identical networks, one with an LReLU activation function, and the other with the ReLU activation function, shown in Fig. 3 as solid and dashed lines, respectively. An interesting observation is the improvement seen with the LReLU network when more layers are included. In Fig. 1, we saw the ReLU activation function performs best when two hidden layers were used, but as the number of hidden layers increases the performance increases with the LReLU network putting it ahead of all of the ReLU. Furthermore, we do see performance gains when increasing the number of hidden layers up to a certain number when they start to converge on a minimum MAE loss. We find, that for both activation functions, once there are five hidden layers, there are no further significant gains in network performance when more layers are used. Computational speed again is a factor here, with the training time needed for a network with eight hidden layers being 217 per cent longer than for one with a single hidden layer. Our final network architecture, based on the results presented here, is six hidden layers, each with 512 neurons, and LReLU activation functions.

Having made this choice, we explain in more detail the difference between an ReLU and a Leaky ReLU. A Leaky ReLU builds from the original ReLU by modifying the handling of negative input values. The ReLU returns an output of zero for a negative input,

$$F(s_{ij}) = \max(0, s_{ij}), \tag{10}$$

whereas a Leaky ReLU assigns a non-zero slope on the negative end,

$$F(s_{ij}) = \max(\alpha s_{ij}, s_{ij}). \tag{11}$$

In equation (11), α is a hyperparameter generally set to 0.01, and s_{ij} is the total input to neuron i in the jth layer. The Leaky ReLU solves the 'dying' ReLU problem Lu (2020), where a standard ReLU can become inactive and output zero for any input value. In this case, it can never recover and can lead to network regions becoming 'inactive'. We find using a Leaky ReLU instead of ReLU improves the MAE performance during training on the validation set, reducing the average MAE loss by $\sim$ 29 per cent.

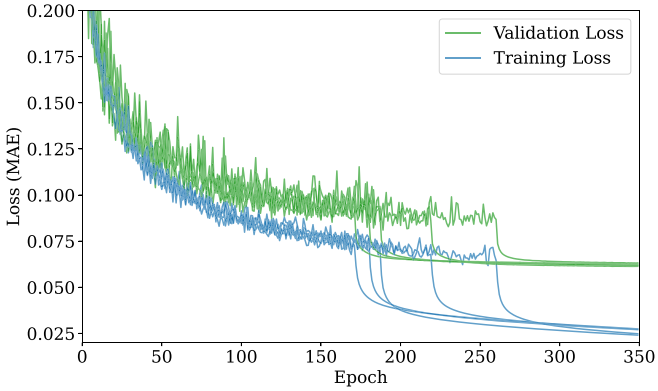

Figure 4. Comparing the training (blue) and validation (green) loss curves for each of the five models that make up the final emulator model architecture (as described in Section 2.2.2). We do not observe any overfitting (i.e. the validation loss is greater than the training loss and the two curves have similar shapes). However, the gap between the validation loss and the training loss curves could be reduced and suggests some underfitting.

We use the adaptive momentum estimation (Adam) optimizer which is a popular momentum-based gradient descent optimization algorithm (Kingma & Ba 2014; Reddi, Kale & Kumar 2019) and set the learning rate to 0.005. We add the AMSGRAD variation (Tran et al. 2019) which aims to improve the performance of Adam around the minima on the error surface using a stochastic method, which evaluates the weights after every minibatch iteration (minibatches are small subsets of the whole training set). At the end of each epoch, we save the model weights if the performance on the validation set has improved (as measured by equation 9) and continue training until there is no improvement for 30 epochs. Then the learning rate is reduced to 10^{-5} for a fine-tuning stage with the RMSprop optimizer (Tieleman & Hinton 2012), allowing us to take small steps towards the minimum of the error surface. RMSprop uses stochastic gradient descent and assumes the error surface is a quadratic bowl. This method boosts the performance of the emulator as we can descend into fine local minima, and we measure improvements to our network by tracking the MAE of the validation samples throughout training. We see evidence of this in Figs 1, 2, and 3 where the MAE rapidly drops when the network transitions into fine-tuning training mode.

The training of our final model architecture is displayed in Fig. 4. We show the training loss (blue line) together with the validation loss (green line) for each of the five individual networks that make up our ensemble model. The loss can be calculated at each epoch in the training process using the validation data to look for signs of overfitting. We do not see any overfitting of the model on to the training set from the comparison of the two loss curves (i.e. the loss curve for the validation set exceeds the loss curve for the training set and the curves have similar shapes). In other words, the model has not learnt the training data set so well as to include the statistical noise or random fluctuations that are present, and the loss stops decreasing after some number of training epochs. If the model was overfitting to the training data set, the training loss would continue to decrease as more training epochs passed. An overfitted network would be more specialized to the training data and less able to generalize to new data. In this case, the loss curve for the training set would continue to decline with decreasing epoch, whereas the loss curve for the validation data may start to increase. The increase in

generalization error can be measured by the performance of the model on the validation data set. In Fig. 4, we see the validation loss curve does not degrade (i.e. begin to increase with epoch), symbolizing a lack of overfitting. There is, however, evidence of underfitting to the validation data set coming from the size of the gap between the training loss curve and the validation loss curve. This indicates that the model is capable of further learning and possible further improvements on using a larger representation from the training data set. A good fit is identified by a training and validation loss that decreases to a point of stability with a minimal gap between the two final loss values.

2.2.3 *Ensembling*

Before training, the weights of a network are initialized according to some distribution, often random. We use an initializer described in Glorot & Bengio (2010). Due to the stochastic nature of the training process training a single network is insufficient since the error surface is likely to contain many local minima and one network is unlikely to traverse enough of the weight space to find the best possible mapping. Overfitting is also a potential problem due to the large number of parameters especially as more layers are added. One solution to these issues is ensembling multiple network predictions (Opitz & Maclin 1999; Sagi & Rokach 2018; Ganaie et al. 2022). This involves training several identical networks with different weight initializations and shuffling the validation and training sets for each model in the ensemble so the models are distributed from input to output. This should allow for a more robust final prediction. We average over the predictions of each model to negate any over- or underfitting to different features of the data.

Using this method, we train five separate networks, each with the same model architecture. The final emulator prediction is the average of the predictions from the ensemble of models. There is scope in the future to improve on this method via a method called stacking (Wolpert 1992) where the ensemble networks themselves are the inputs to a single network with generalizes the outputs for improved results. However, this works best where the ensemble networks are varied and provide different information, such as different architectures or combining different types of machine learning algorithms.

2.3 Parameter fitting

We use the emulator for inference on target data sets; that is, fitting the model to given data sets. We employ an MCMC sampler to compare the generated models against the observed data sets with the goal of sampling from a set of parameters that produces the models that best fit the observables. The Metropolis–Hastings algorithm (Robert et al. 2004) is a common and simple method of executing an MCMC, generating serially correlated draws from a sequence of probability distributions, eventually converging to a given target distribution. The means of convergence comes from the minimization of the absolute error between the emulator output and the observational constraints. We note that as we are minimizing the absolute error between multiple data sets i.e. the discrepancy between the model predictions and the data, we do not take into account their associated (measurement)[1] errors. We wish to weight certain data sets over others to allow us to investigate the effect of requiring better fits to some data sets and how this affects the reproduction of other data sets, as well as seeing how the optimal parameter choices change as a result. We therefore introduce a modified version of the MAE (introduced in equation 9) which includes a vector of heuristic weights, W, to vary the contribution of the residuals from constraint i to the total error,

$$\mathrm{MAE}^{\mathrm{obs}}(y,\hat{y}) = \frac{1}{n^{\mathrm{obs}}}\sum_{i=1}^{n^{\mathrm{obs}}}\frac{W_i}{n_i^{\mathrm{obs}}}\frac{|y_i-\hat{y}_i|}{\boldsymbol{\sigma}_i}, \tag{12}$$

where $\hat{\mathbf{y}}_i$ is the predicted value of the i-th observable constraint, and y_i is the corresponding observable value across n_i^{obs} data points. Due to $\hat{\mathbf{y}}_i$ and y_i being vector quantities, the modulus represents the L1 norm. $\boldsymbol{\sigma}_i$ is a vector of errors corresponding to y_i. We sum over the n^{obs} observable constraints. The different observational data sets contain different numbers of data points, therefore we divide the weighted absolute error of the i-th data set by the number of data points, n_i^{obs}, for equal contribution to the mean error result. In later sections, we refer to equation (12) as the MAE.

The Metropolis–Hastings procedure for updating a Markov Chain compares the likelihoods from the current parameter location or state to a proposed (new) state. Assuming uniform priors throughout, each chain is initialized on a random point in the parameter space which is assigned as the *current* state, x. Then, we sample a proposed state, x', from independent Laplacian proposal distributions about x, $L(x'|\boldsymbol{\mu}, b) = (1/2b)\exp(-|x'-\boldsymbol{\mu}|/b)$ where $\boldsymbol{\mu} = x$ and the scale parameter vector b is set as 1/20th of the parameter ranges given in Table 1. The proposed state must satisfy the condition that the proposal lies within the defined parameter bounds given in Table 1. We decide whether the proposed state is accepted or not by measuring the likelihood improvement of emulator predictions to the observational data from the current to the proposed state using a Laplacian likelihood with scale parameter $b_{\mathrm{obs}} = 0.005$. Taking the ratio of likelihoods at states x' and x gives the *acceptance ratio*, α,

$$\alpha = \frac{L(f_*(x')|\boldsymbol{y}, b_{\mathrm{obs}})}{L(f_*(x)|y, b_{\mathrm{obs}})}, \tag{13}$$

where y is the vector of observables and $f_*(\cdot)$ is the emulator. We could use a ratio of errors as an acceptance ratio in our MCMC, however, doing so may not align with the principles of Bayesian inference and so could have implications for the accuracy and efficiency of our algorithm. The likelihood is often used in Bayesian inference due to its probabilistic interpretation, as it provides a measure of how well the model explains the observed data given a set of parameters. The acceptance ratio is compared to an acceptance criterion, u, which is a random uniform number $u \in [0, 1]$; a proposed state is *accepted* if $\alpha \geq u$, in which case $x = x'$ and the next sample is drawn from a Laplacian centred on the new state, or a proposed state is *rejected* if $\alpha < u$ for which case we sample again from the original Laplacian centred on x. Using this method, if the error between the emulator predictions and the observables reduces when moving from state x to x' the sample is always accepted, else we accept the proposed state x' with a probability α. We expect the density of accepted samples to trace the regions in the parameter space which give the best fits to the data. At the start of the chain, there will be a burn-in phase as the accepted samples tend towards local maxima in the parameter space so we discard the first half

[1]Here, we mean the errors made when estimating the statistic. For example, for the luminosity function, this could be the Poisson error derived from the number of galaxies in a luminosity bin or a more advanced estimate which includes sample variance, inferred by using independent mock catalogues. As different people make different assumptions regarding these errors, it is hard to compare them across very different data sets.

of accepted samples. Testing multiple chain lengths, we find chains converge to local MAE minima (given by equation 12) within the first 5000 samples and so we choose the chain length as 7500 (after discarding the burn-in phase).

2.4 Data sets

Our decision as to which GALFORM input parameters to vary comes from a combination of physical motivation and choices informed by previous analyses (mainly Elliott et al. 2021). We differ from their parameter choices by focusing more on the contribution from quiescent galaxies and less on galaxies experiencing a starburst. For a typical Hα galaxy survey, we find burst galaxies only affect the extremely bright end of the luminosity function and have little impact on the overall number counts (see for example the predictions from Lacey et al. 2011 for the ultraviolet LF, which is also sensitive to recent star formation). Close to the *Euclid* and *Roman* flux limits, quiescent galaxies are dominant. Burst galaxies do, however, dominate the high-flux tail of the Hα emitter counts, but this is not important for the overall clustering measurement.

2.4.1 *Training and testing data*

We use a supervised machine-learning method to emulate running a computationally expensive model, GALFORM. Training the emulator requires running the full model. Generally, the more samples used during training, the better the predictions should be.

Whereas Elliott et al. (2021) ran GALFORM at a single output redshift to make predictions for their calibration data, the computational cost per model is much higher in our case as we need to compute the redshift distribution of Hα emitters. This is due to the structure of the GALFORM code; running for N output redshifts effectively increases the run time by a factor of N. One option to produce predictions for the redshift distribution of Hα emitters would be to generate a lightcone catalogue (e.g. Merson et al. 2013). For the PMILL simulation, for the range of redshifts of interest for *Euclid* and *Roman*, this would require running 135 output redshifts. Instead, we can run GALFORM for a much smaller, select number of output redshifts, taken from the target range. For each output, we construct the LF of Hα emitters. We then use this information to compute the redshift distribution of Hα emitters, interpolating between the luminosity functions computed at the output redshifts. Another computational saving that can be made is to run GALFORM for a fraction of the available dark matter halo merger histories. We experimented with using different numbers of output redshifts and different fractions of the merger histories to compute the redshift distribution of Hα emitters, comparing the answers to a full lightcone calculation. The calculation converges with five PMILL redshift snapshots between redshifts $0.69 \leq z \leq 2.00$ using ~ 0.5 per cent of the available halo merger histories. We also produce output at $z = 0$ to compare to the local calibration data.

Model parameters were generated via Latin hypercube sampling for efficient and smooth coverage (as described in Loh 1996; Bower et al. 2010). The parameter ranges are given in Table 1. The Latin hypercube sampler generated 3000 sets of the 11 parameters, resulting in 3000 GALFORM outputs, each with an associated Hα redshift distribution and $z = 0$ K- and r-band LFs. The GALFORM inputs and outputs formed the input–output vector pairs, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ for the deep learning emulator, where $\boldsymbol{x}_i$ is the ith set of model parameters and $\boldsymbol{y}_i$ is the corresponding output vector of the redshift distribution and LFs. We separate the samples randomly into three sets: training, validation, and holdout sets. The training and validation sets are used during the training of the emulator, and the hold-out set is kept separate to evaluate performance on unseen data. The ratio of training samples to hold-out test samples was 29:1 and for each network trained, 20 per cent of the training samples were randomly chosen as the validation data.

2.4.2 *Calibration and comparison data sets*

Traditionally, GALFORM has been calibrated mostly using local data, as these have been the measurements with the smallest errors (see Lacey et al. 2016). We continue this trend by using the r and $K-$band LFs measured from the galaxy and mass assembly (GAMA) survey (Driver et al. 2012); here, these data replace the older b_J and K-band LFs used to calibrate GALFORM. This choice has the advantage that the same team has done the data reduction and made the assumptions about the k- and evolutionary corrections. We also use the Hα redshift distribution measured by Bagley et al. (2020). Using calibration data sets at different redshifts greatly reduces the volume of the viable parameter space.

The full list of calibration and comparison data sets is as follows:

(i) For the Hα redshift distribution, we calibrate the emulator to the redshift distribution from Bagley et al. (2020). They used measurements from two slitless spectroscopic WFC3-infrared data sets, 3D-HST + A Grism H-Alpha SpecTroscopic (AGHAST) survey and the WISP survey (Atek et al. 2010) to construct a *Euclid*-like sample. They detect the combined Hα + [N II]-emission from galaxies in the redshift range $0.9 \leq z \leq 1.6$ with total line flux brighter than $\geq 2 \times 10^{-16}$erg s^{-1}cm^{-2}.

(ii) For the $z = 0$ K-band and $r-$band LFs, we take data from Driver et al. (2012) who used the GAMA data set to construct the low-redshift ($z < 0.1$) galaxy luminosity functions in multiple bands.

We also compare our best-fitting models to the previous local LF calibration data (the $K-$band LF measured by Cole et al. 2001 and the b_J measured by Norberg et al. 2002). This is to see if models using the new local calibration data sets still give good fits to the old calibration data; this is an indirect way of seeing (through a model) if these observational LFs are consistent with one another.

3 RESULTS

3.1 GALFORM runs for training and testing

We start with the Lacey et al. (2016) model and replace the parameters highlighted in Table 1, using the 3000 combinations generated by the Latin hypercube sampler. For each model, we run GALFORM at six redshift snapshots $z = 0, 0.69, 0.90, 1.14, 1.60, 2$. These were selected to cover the redshift range probed by the Bagley et al. (2020) Hα redshift distribution and the local LFs.

3.2 Emulator performance

The development of the architecture for the emulator is described in Section 2.2. During the architecture training phase, we only ran our networks up to 350 training epochs. For the final network, we chose not to limit the number of epochs but instead included an early stopping clause which stopped and saved the network at its lowest training MAE validation loss score. On average the networks gave their lowest loss score between 500 and 700 epochs. We also follow Elliott et al. (2021) by ensembling networks of the same

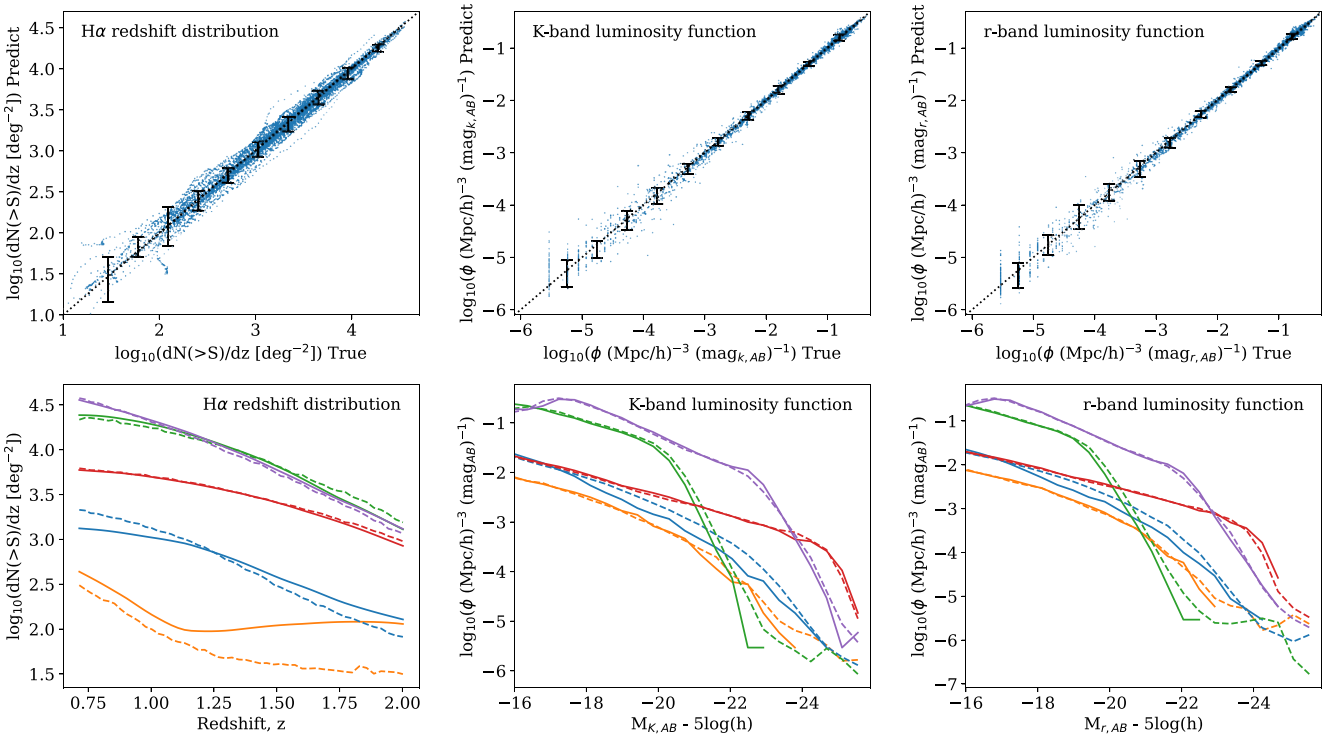

Figure 5. Emulator performance across the three calibration statistics computed with the holdout parameter sets. The top row shows the emulator output (y-axis) against the true GALFORM output (x-axis). Black error bars indicate the 10th–90th percentile range of the residuals. The bottom row shows a draw of emulator outputs (dashed lines) and true GALFORM outputs (solid lines) for selected parameter sets. In these panels, different colours denote different parameter sets.

architecture, averaging over the outputs to produce a final result. We tested ensembles of five and ten networks and found little to no improvement in emulator performance against the hold-out set. Note that going from one network to an ensemble of five gives roughly a ten per cent reduction in the MAE. Furthermore, the more networks to be averaged over, the greater the computational time which becomes important as we run an MCMC across a substantial number of walkers each with around 15 000 steps. Therefore, the emulator consists of five equal architecture networks (described in Section 2.2). We want to evaluate the ability of the emulator to output accurate GALFORM predictions at new points in the parameter space. The set of 3000 GALFORM outputs was split up with 96.67 per cent of the outputs used for training our emulator as described in Section 2.2 (equating to 2900 parameter combinations) and the remaining 3.33 per cent (100 parameter combinations) being used as unseen outputs for testing purposes (hold-out set). This split maximizes the number of training samples and provides an appropriate range of unseen test samples to evaluate the network. When training each network, we randomly split the 2900 parameter output combinations into a training set and a validation set with 20 per cent going towards validation (580 parameter combinations). For each network trained in the emulator ensemble, the training and validation sets were shuffled.

In the upper panels of Fig. 5, we show the emulator predictions against the hold-out set outputs from the corresponding full GALFORM runs. A perfect emulator would follow the $y = x$ line (dotted) with no scatter. In general, we see the emulator following a tight relation to the diagonal across the three statistics, indicating that the emulator is accurately predicting GALFORM output for the holdout set parameters, without any significant biases and a reasonably small scatter. Out of the three statistics, the redshift distribution predictions appear to have a greater uncertainty than the K and r-band luminosity functions. However, this is largely an artefact of the redshift distribution predictions spanning a smaller dynamic range than the other statistics, so this scatter plot is 'zoomed-in' compared to the others (covering just over 4.5 decades in scale as opposed to six decades in the other panels). In the lower panels of Fig. 5, we show the performance of the emulator across the three statistics on a sample of the holdout set parameters, plotting the emulator outputs as dashed lines and the true GALFORM outputs as solid lines. The parameter samples drawn from the holdout set were chosen to reflect the range of emulator performances, including parameters that the emulator most struggled with for each statistic. Each colour across the three panels is the same combination of parameters from the holdout set. The luminosity function plots display the ability of the emulator to predict beyond the resolution of GALFORM when the true model was generated with a finite sample of merger histories from the simulation, which can result in some luminosity bins being empty at the bright end. The lower panel of Fig. 5 reveals some sources of inaccuracies in the predictions, particularly the Hα redshift distribution, which is more prone to exhibiting noisy behaviour for some choices of parameters, for example, the low-redshift distribution (orange line) is poorly predicted. The error bars for the redshift distribution predictions are fairly even across the redshift bins and this is reflected in the lower panel plots where the majority of predictions follow the shape of the true GALFORM output but with varying degrees of offset. The main source of errors for the luminosity function predictions is seen at low values of ϕ. We do see that at the bright end of the luminosity function plots the predictions can become noisy but the overall shape is well captured.

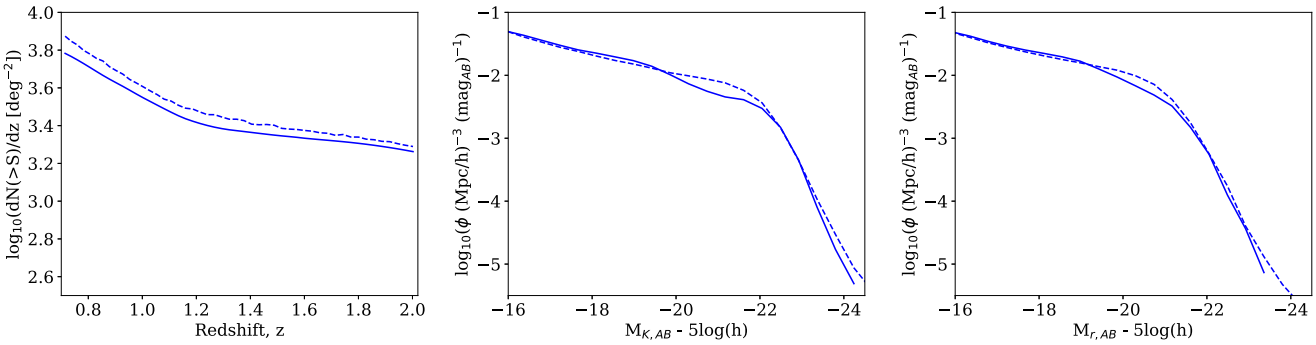

Figure 6. Emulator predictions (dashed lines) using the Lacey et al. (2016) GALFORM parameters compared with the true GALFORM outputs (solid lines). We predict the Hα redshift distribution (left), and the $z = 0K$- (middle) and r-band (right) luminosity functions.

The majority of emulator predictions for the redshift distribution are reasonably close to the GALFORM predictions, but we do come across cases with substantial discrepancies between the true and predicted outputs (as exhibited by the orange line in the bottom row of Fig. 5). We see far fewer cases like this within the holdout set of poor predictions of the true GALFORM outputs when it comes to both luminosity functions, with the largest discrepancy seen in the blue parameter set. These poor predictions are usually indications that the training data did not contain sufficient examples of this behaviour as these examples appear to be extreme cases of the output and so are less common. The emulator constructs a function $f_*(\cdot)$ by fitting it to the training examples, where $f_*(\cdot)$ can interpolate between the points in the parameter space. However, the interpolation is less reliable in the sparser regions of the space, such as at the extremities of our parameter bounds.

We can see that at the bright ends of the K- and r-band LFs in Fig. 5, the emulator tends to slightly overpredict the GALFORM output. This is a consequence of using a small fraction of the available merger histories (0.6 per cent of the total), which leads to noisy predictions at low-galaxy number densities, and, as seen in Fig. 5, cutoffs at different luminosities for different choices of parameters. The emulator outputs a fixed number of bins, therefore during training, we omit any luminosity bins which contain zero galaxies when computing the loss. This leads to the emulator having fewer brighter luminosity bins to fit which are biased towards having higher values of ϕ in these brighter bins. This causes more cases of overprediction at these luminosities. This problem is minor since the Driver et al. (2012) luminosity function data does not sample ϕ to very low-number densities. These issues could be resolved by evaluating GALFORM using a larger fraction of the available merger histories, although this would be more expensive computationally with little gain.

We also evaluate the performance of the emulator against the Lacey et al. (2016) GALFORM model in Fig. 6. We see an overall good fit to the true model, with the emulator redshift distribution overpredicting the true GALFORM model by a small amount. This matches our findings of the emulator performance on the holdout set above. For the redshift distribution, the emulator can still accurately identify the shape of the true model. The emulator does well at matching the true GALFORM model for the local LFs, with the only deviation seen around the break at magnitudes ~ -22 for the K band and ~ -21 for the r band. The emulator is unable to recreate the dipped features around these magnitudes which indicates a deficiency of these types of parameters within our training set. The possible changes we could make to the training set of the emulator that we highlighted before would improve our predictions against the Lacey et al. (2016) parameter set.

3.2.1 *Performance and training set size*

To find how the emulator performance depends on the number of full GALFORM calculations, we train the emulator with 900, 1900, and 2900 samples of parameters (in each case split with 20 per cent of the samples going towards validation). The emulators consist of an ensemble of five identical networks each trained on the same (shuffled) training and validation sets. Performance is evaluated on the same 100 holdout parameter samples. The emulator shows a clear reduction in the MAE with an increasing number of training samples. Using an ensemble of networks results in a near-constant improvement in performance of almost 12 per cent compared to using a single network: however, this effect saturates after five networks.

3.3 Parameter fitting on the calibration data – model optimization

We apply the methods described in Section 2.3 to calibrate the model to the data sets introduced in Section 2.4.2. We begin by investigating the tensions between the three statistics by adjusting the weights applied to the residuals between our emulator prediction and each data set (given by equation 12) and then performing an MCMC parameter search to see how the best-fitting parameter choices respond. In Fig. 7, we show the emulator predictions for the best-fitting parameters found from five MCMC chains using different weighting schemes. To make accurate predictions for *Euclid* and *Roman*, we need to fit the Hα redshift distribution data from Bagley et al. (2020). However, to reduce the overall model parameter space, it is important to constrain the model to reproduce the local luminosity functions. Hence, we need to find a balance of fits between the two. When the weighting to the Hα redshift distribution data is low, for example, a weighting of one or two (blue and orange lines in Fig. 7, respectively), we see a poor accuracy reproduction of the Hα redshift distribution data and strong performance regarding the luminosity functions, particularly around the break. As the redshift distribution weighting increases, we notice increasing deviation at the bright- and faint ends of the luminosity functions, but an improved fit to the redshift distribution data, with the predicted distributions being within the error bounds of the observations. Applying a weight of four to the redshift distribution (green line) still allows us to recover the LF break at L^* and stays just as close to the high-redshift data points in the redshift distribution as a weight of six (purple line).

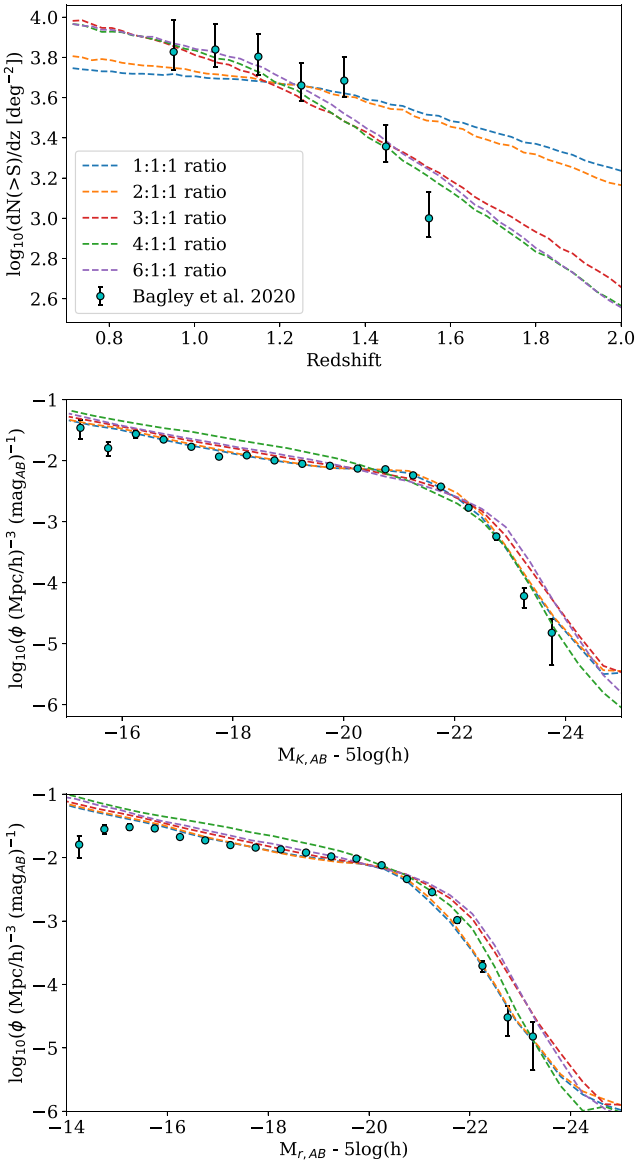

Figure 7. Best MCMC fits for five different weighting schemes (as indicated by the key in the top panel), on increasing the weight applied to the Hα redshift distribution (first number in the line label) to display the tensions between the constraints (the other two numbers show the weights applied to the local LFs). We show a redshift distribution weight value W of one (blue), two (orange), three (red), four (green), and six (purple), plotted with the Bagley et al. (2020) Hα redshift distribution (top) and Driver et al. (2012) $z = 0$ K- (middle) and r-band luminosity functions (bottom).

The spread across the LFs for the different weightings is surprisingly low given that the spread in the redshift distribution fits is large in comparison. This could indicate that there are multiple regions in the parameter space that can fit these models, according to the emulator. This likely arises from the error of the emulator outputs, particularly for the redshift distribution predictions. It is worth noting that these parameter fits come from a small number of MCMC chains: we expect to see improvements in the best-fitting parameters when we evaluate 100 MCMC chains.

We set $W_i = 4$ for the redshift distribution constraint, and apply unit weight to both K- and r-band LF constraints. With the weighting scheme for the three statistics fixed, we recalibrate GALFORM across the three constraints to estimate the best-fitting parameters. We run 100 MCMC chains with our emulator, each with 7500 steps after the burn-in phase (which itself is 7500 steps). The residual of each sample is computed using the emulator and the weighted MAE function. The minimum MAE obtained for each chain lies in the range $\sim 0.25 - 0.28$. As we have seen in Section 3.2, our emulator outputs have an associated error, so we cannot confidently discern which parameter sets give the best fit to the observational data with the emulator alone. Hence, we evaluate the parameters that gave the lowest MAE value from each of the 100 MCMC chains with GALFORM.

In Fig. 8, we illustrate the regions in the parameter space sampled by the MCMC chains. The shaded regions show the accepted samples from our chains, each 7500 steps long after discarding the burn-in. The shading indicates the density of the accepted samples, with the darker regions corresponding to the more favoured regions of the parameter space. Also shown in Fig. 8 are 1D histograms of the density of accepted samples. For some parameters, a reasonably large range of parameter values results in acceptable fits to the constraints. However, when plotted in one or two dimensions the space appears widely sampled, on moving to a higher dimension the acceptable regions are reduced significantly. This is the effect of the high dimensionality of the parameter space, as described in Bower et al. (2010). We see that to fit the three statistics using the weighting scheme described, the fits prefer high values of $\gamma_{\rm SN} \sim 4$ possibly beyond the sampling parameter boundary. We have the option to extend our parameter space, but doing so will probe parameters beyond the space used to train the emulator. This could result in more uncertain predictions. Furthermore, we do not want to extend our parameter ranges to unphysical choices for the processes being modelled. We also observe a bimodal distribution for the $V_{\rm SN,\,burst}$ parameter which tends towards the lower and upper boundaries of our parameter range at ~ 10 and $\sim 800\,{\rm km\,s^{-1}}$, respectively. In contrast, the parameters $f_{\rm ellip}$, $f_{\rm burst}$, and $\tau_{*\rm burst,min}$ are weakly constrained showing almost uniform sampling, whereas the parameters that contribute to the SN and AGN feedback are more tightly bound.

Out of the lowest MAE parameters from the 100 MCMC chains, we plot the output from the 50 best sets of parameters evaluated using GALFORM in Fig. 9. These runs cover a range of weighted MAE, from $0.25 - 0.31$, with the remaining runs extending to a weighted MAE of 0.64. The 50 best-fitting runs characterize the constraint data sets well and confirm the effectiveness of our MCMC optimization and emulator while also indicating the level of uncertainty present in our method. We show the run with the lowest MAE in Fig. 10, along with the emulator prediction for the same set of parameters (red dashed line), along with the output of the model presented in Lacey et al. (2016) as the solid grey line. We see that there is a spread of possible parameters. Therefore, the best-fitting parameters presented are just one realization of many possible choices due to the effects of calibrating to multiple data sets with tensions between them and the degeneracies between the parameters.

The spread across the 50 best MCMC chains as evaluated by GALFORM is tight across the K- and r-band LFs; there is somewhat more variance in the redshift distribution outputs particularly at higher redshifts. The redshift distribution predicted using our overall best-fitting set of parameters is within the error bars of most of the Bagley et al. (2020) data points. Due to the tension between the Hα redshift distribution and the local LFs, the general trend of fits to the LFs is to overpredict the bright end. This is particularly evident for the $z = 0$ r-band LF, although the selected parameters do well at replicating the break. There is greater uncertainty in the fits to the $z = 0$ K-band LF. The lowest weighted MAE parameter set predicts a far weaker break compared to the data from Driver et al. (2012). In Table 2, we show the set of parameters with the lowest weighted

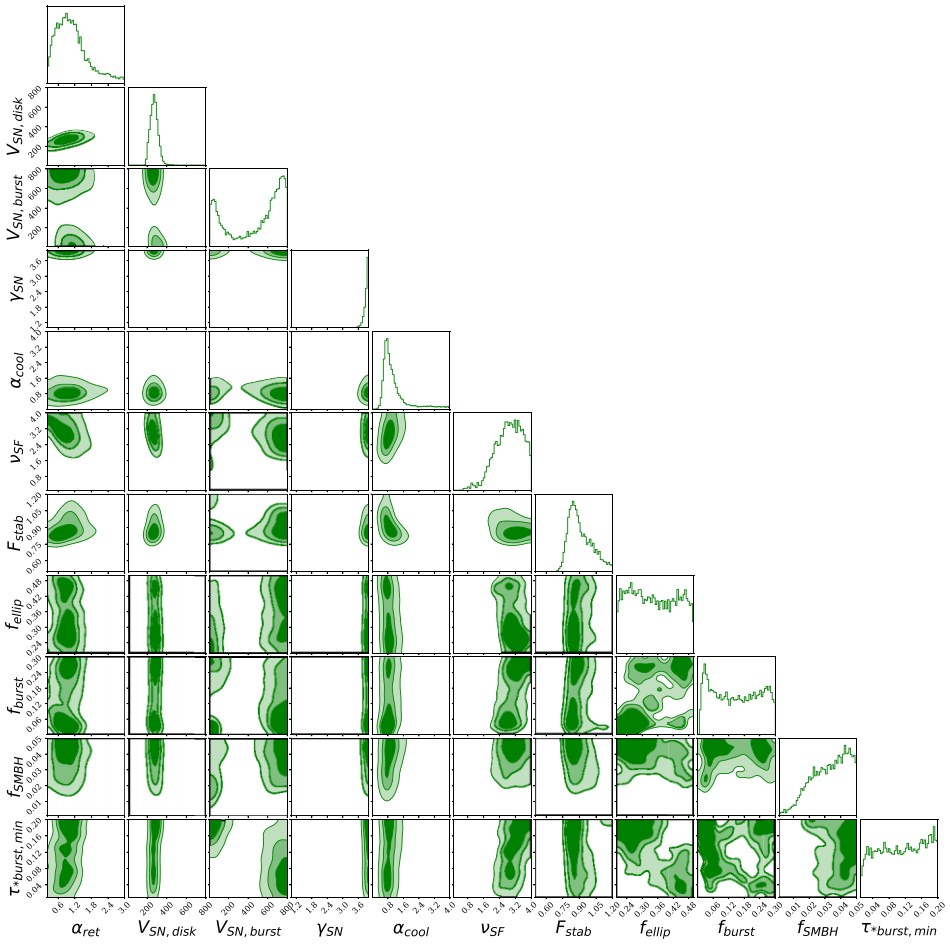

Figure 8. Accepted samples from 100 MCMC chains for fits to the H α redshift distribution, K- and r-band LFs. The first 50 per cent of samples were discarded to allow for burn-in. The histograms show the marginalised distribution of the parameters. The ranges on each axis are the same as those quoted in Table 1. The shading corresponds to the density of chain steps, with darker colours corresponding to more densely sampled regions. The darkest regions correspond to the 25th percentile and the lighter regions to the 50th and 75th percentiles.

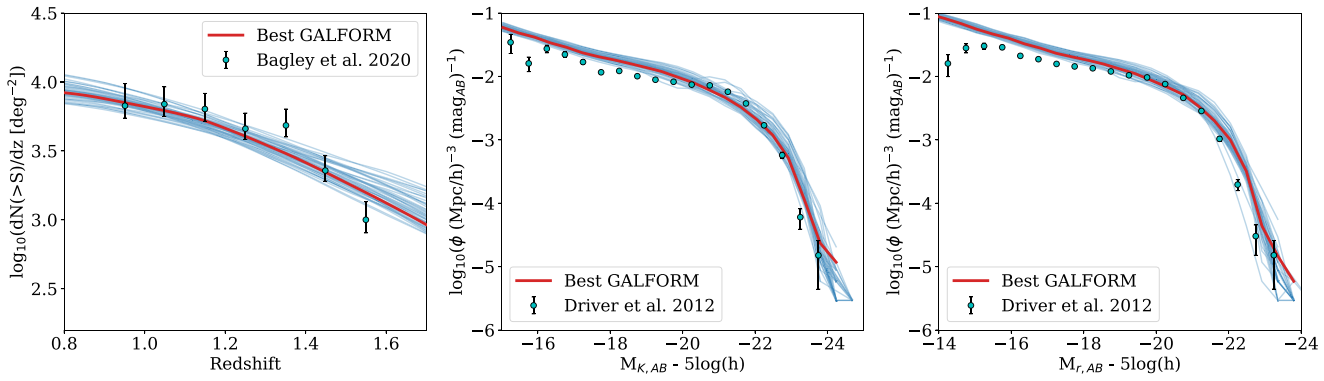

Figure 9. The GALFORM evaluations of the best-fitting parameters found with 100 MCMC chains, each 7500 samples in length, using the constraint weightings described in the text. Here, we plot a sample of the best 50 runs, as measured by weighted MAE (equation 12). The red line indicates the parameter set with the lowest weighted MAE. The remaining 49 runs are plotted in blue. The data described in Section 2.4.2 is shown in cyan.

MAE to the observational data (corresponding to the red line in Figs 9 and 10), and compare with the parameters presented in Lacey et al. (2016) (hereafter named Lacey16). We also show the parameter set, which, out of the 50 best-fitting models, is the closest in parameter space to the Lacey16 model. Looking at this model and the parameter values from the 50 best MCMC chains in general, we find that certain parameters, such as $V_{\rm SN,disc}$ and $\gamma_{\rm SN}$ are constrained to a tight range of values, whereas parameters such as $V_{\rm SN,burst}$, $f_{\rm burst}$, and $\tau_{*\rm burst,min}$ can be drawn from a large proportion of the explored range. Although we do see some parameters that have a large proportion of their

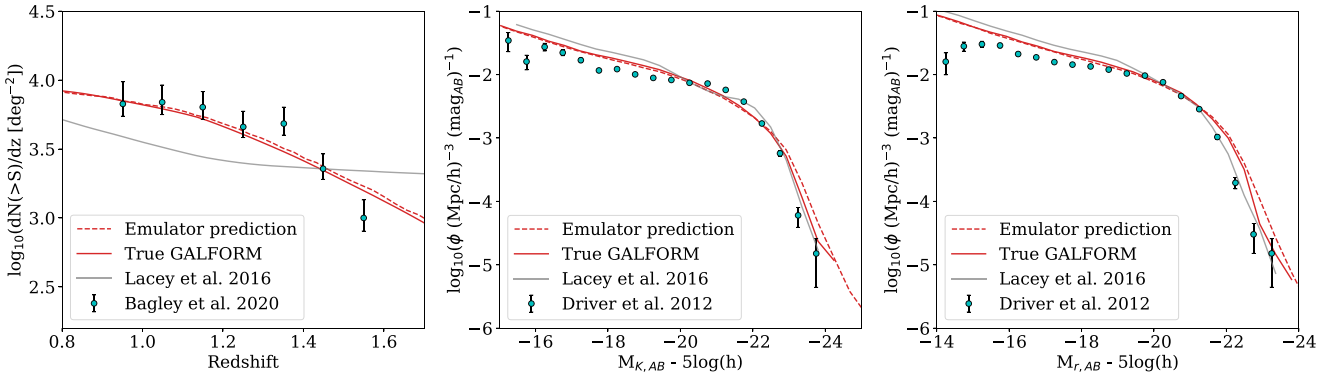

Figure 10. Predictions for the calibration data from the lowest MAE parameter set as evaluated by GALFORM (solid red) compared with the equivalent parameters evaluated by our emulator (red dashed) with the calibration data described in Section 2.4.2. The grey line shows the predictions of the Lacey et al. (2016) model.

Table 2. Results from the 50 best-fitting MCMC chain parameters (as measured by the weighted MAE in equation 12) found using the emulator. The first column gives the parameter name. In the second column, we present the parameters for the best-fit seen in Fig. 10. The Lacey16 model parameters are given in the third column. The final column gives the parameters of the model drawn from the 50 best-fitting models whose parameters are closest to those in Lacey et al.

Parameter	This work	Lacey16	Lacey16-like example
$\nu_{\rm SF}$	3.97	0.74	2.81
$V_{\rm SN,\,disc}$ (kms^{-1})	201.30	320	248.21
$V_{\rm SN,\,burst}$ (kms^{-1})	785.64	320	765.76
$\gamma_{\rm SN}$	3.98	3.40	3.98
$\alpha_{\rm ret}$	0.27	1.00	1.08
$F_{\rm stab}$	0.85	0.90	0.85
$f_{\rm ellip}$	0.22	0.30	0.04
$f_{\rm burst}$	0.083	0.05	0.05
$\tau_{*\,\rm burst,min}$ (Gyr)	0.032	0.10	0.11
$f_{\rm SMBH}$	0.039	0.005	0.05
$\alpha_{\rm cool}$	0.79	0.80	0.99

parameter spaces sampled, the distribution is not always uniform. The $\nu_{\rm SF}$ parameter appears to cover a very large range. However, when looking at the corner plot of Fig. 8 we see that the majority of the sampling occurs at the high values of $\nu_{\rm SF}$ but there is a small subregion sampled at ~ 1.0. The parameter $f_{\rm SMBH}$ sampling distribution is skewed left which extends the accepted parameter range.

We compare the weighted MAE of our best-fitting model with the Lacey16 model, using the procedure described in Section 3.3, that is using the same weighting scheme we have been using up to this point. Using this metric, as expected, the new model is a better overall fit to the calibration data, with a weighted MAE of 0.25, compared with 0.50 for Lacey16. The MAE for Lacey16 is outside the range of the minimum MAE reached by our 50 best MCMC chains but within the range of lowest MAE values from the 100 MCMC chains. The improved MAE of the new best-fitting model (and indeed the majority of our MCMC-found models) is mainly due to the large improvement in the fits to the Hα redshift distribution, while the fits of the new models to the K- and r-band LFs are similar to those of the Lacey16 model. The new model fit is closer to the faint end of the observed LFs, whereas the Lacey16 model is closer to the Driver et al. (2012) data points at the bright end, particularly in the r band. The main source of error for the Lacey16 model is the poor fit to the Hα redshift distribution, whereas our model more accurately describes the drop off in number counts beyond $z \sim 1.4$. This can be quantified by considering the contribution to the MAE from each statistic: the best-fitting model has an MAE of 0.09 for the Hα redshift distribution whereas the Lacey16 model is worse with an MAE of 0.26. Although by eye the fits to the K-band luminosity functions are similar between the new model and Lacey16, the MAE values indicate that the new model fits better to the Driver et al. (2012) data than the Lacey16 model does: the new model has an MAE of 0.17, whereas the Lacey16 model achieves 0.20. This is likely to be due to closer fits at the faint end contributing to a higher proportion of the MAE score. The fit to the bright end is very similar but the Lacey16 model has a much sharper break in its luminosity function. We can break down the K-band LF MAE calculation further by focusing on the bright half of the observed data points. As mentioned above, the Lacey16 model is a closer fit to the data points at the bright end as measured by eye. This is confirmed as the MAE of the bright part for the Lacey16 model is 0.11 and for our new model is 0.14. Finally, our model performs slightly better than the Lacey16 model when predicting the r-band LF, scoring an MAE of 0.23 versus 0.25 for the Lacey16 model. This is likely to be for similar reasons as the K band, where our model is closer to the Driver et al. (2012) data at the faint end. It is clear that the Lacey16 model is slightly better at predicting the luminosity function from the exponential break to the bright end as our model overpredicts the bright end. If we focus only on the bright half of the luminosity function, the Lacey16 model is a closer fit to the observed data than our model, with an MAE of 0.10 versus 0.16.

Due to the tensions between the calibration data, better fits to the Hα redshift distribution data come at the expense of more severe overpredictions of the bright end of the LFs as previously discussed, and as shown by the lines when increasing the weighting in Fig. 7. Similarly, if we try to improve the fits to the LFs, this leads to an overestimation of the number of Hα emitters at higher redshifts.

3.3.1 *Number count predictions for the* Euclid *redshift survey*

Galaxies detected through their emission in the unresolved Hα(+ N[II]) lines are the main target for the *Euclid* and *Roman* redshift surveys. Satisfied with the best-fitting parameters from the MCMC search using the emulator as evaluated using GALFORM, we can use these models to predict the number of galaxies that will be seen by the upcoming surveys. The cumulative number counts are shown in Fig. 11, along with the recent WISP + 3D-HST data from Bagley et al. (2020) for galaxies in the redshift range $0.9 \leq z \leq 1.6$. The corrected number counts from Bagley et al. (2020) for the

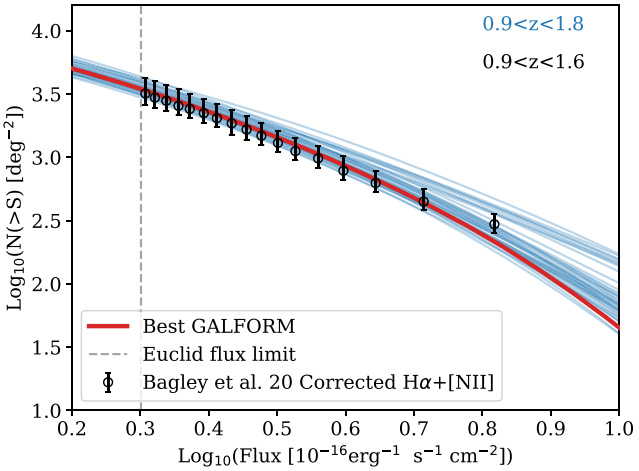

Figure 11. Number counts predictions from our 50 best MCMC parameters for galaxies with $0.9 < z < 1.8$ (blue lines), with the best set of parameters as evaluated by GALFORM in red. We plot this against the Bagley et al. (2020) $0.9 < z < 1.6$ number counts (black points). The Euclid flux limit 2×10^{-16} erg^{-1} s^{-1} cm^{-2} is marked by the vertical dashed grey line.

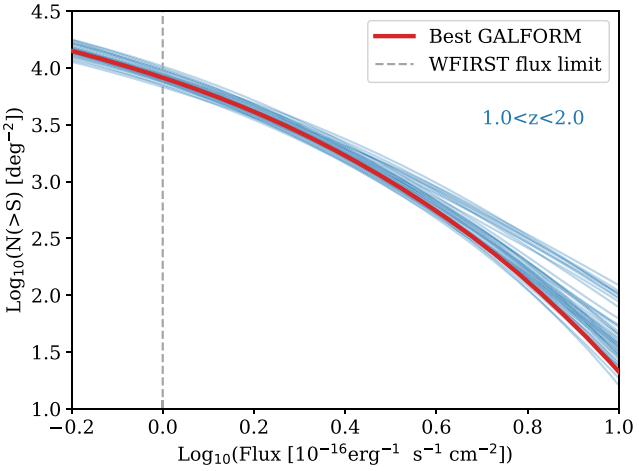

Figure 12. *Roman* number counts predictions from our 50 best MCMC parameters for galaxies between $1.0 < z < 2.0$ (blue lines), with the best set of parameters as evaluated by GALFORM in red. The *Roman* flux limit 1×10^{-16} erg^{-1} s^{-1} cm^{-2} is shown by the vertical dashed grey line.

WISP + 3D-HST data at the *Euclid* flux limit is $3266^{+157.7}_{-174.8}$ Hα + N[II] emitters deg^{-2}. Our models predict the galaxy density in the slightly broader redshift range $0.9 < z < 1.8$, which matches that of the *Euclid* redshift survey. From the 50 best models, the spread in emission-line number counts estimates for galaxies with a flux greater than the *Euclid* limit ($f \geq 2 \times 10^{-16}$ erg^{-1} s^{-1} cm^{-2}) is 2962–4331 deg^{-2}, with our best-fitting model to the calibration data outputting a number count of 3462.5 deg^{-2}, corresponding to $\sim$ 46.7 million sources over 13 500 deg^2. Our best-fitting model comfortably lies within the range of the Bagley et al. (2020) Hα + N[II] number counts. The distribution of predicted number counts can also be quantified using the 10–90 percentile range of the 50 best models, which gives the narrower spread of 3158–3952 deg^{-2}. We compare our number count predictions with those of Pozzetti et al. (2016) who empirically fit luminosity functions to earlier surveys, HiZELS, WISP, and *HST* + NICMOS. Covering the redshift range $0.9 < z < 1.8$ to a flux limit of 2×10^{-16} erg^{-1} s^{-1} cm^{-2}, Pozzetti et al. predicted 2000–4800 Hα emitters deg^{-2}. It is worth noting that the Pozzetti et al. (2016) predictions are in terms of observed Hα flux, i.e. they are corrected for [N II] contamination. In contrast, our results blend Hα + N[II] to match the results of Bagley et al. (2020). At the spectral resolution of *Euclid*, these two lines will be partially blended.

Fig. 7 of Bagley et al. (2020) shows the observed cumulative number counts along with fits from various models including the three empirical models from Pozzetti et al. (2016). For the purposes of this comparison, Bagley et al. (2020) converted the Hα counts from the Pozzetti et al. (2016) models to Hα + [N II] counts using a fixed [N II]/Hα line ratio: Hα = 0.71 (Hα + [N II]). Out of their three models, the only one that fits the $0.9 \leq z \leq 1.6$ observations well is Model 3 which shows a similar fit to our best-fitting model in Fig. 11. Fig. 7 from Bagley et al. (2020) also shows the redshift distribution predictions from Pozzetti et al. (2016), where once again Model 3 is the best-fit to the observed redshift distribution. However, the fits are only good for the first five data points before the drop off in counts observed for $z \sim 1.4$. As seen in Fig. 10, our best-fitting model is a better fit to the observed redshift distribution data points as we represent more closely the trends beyond $z \sim 1.4$.

We also calculate the number counts for a *Euclid*-like survey with a magnitude limit of $H = 24$ using our best model, keeping the line flux limit and redshift range fixed. The number of galaxies counted in this case is 3444.5 deg^{-2}; including the H-band cut reduces this by 0.5 per cent, which is somewhat smaller than the 3 per cent reduction reported by Zhai et al. (2021).

3.3.2 *Number count predictions for* Roman

The High Latitude Spectroscopic Survey onboard NASA's *Nancy Grace Roman Space Telescope* will cover 2000 deg^2 and will use Hα(+ N [II]) galaxy redshifts to map large-scale structure at $1 < z < 2$ (Spergel et al. 2015) to a flux limit of 1×10^{-16} erg^{-1} s^{-1} cm^{-2}. We use the same 50 best-fitting parameters described in Section 3.3 to evaluate GALFORM to predict the number of galaxies that will be seen by a *Roman*-like survey. The cumulative number counts are shown in Fig. 12. From the 50 best models, the spread in number counts estimates for Hα sources seen by *Roman* is 6786–10 322 deg^{-1}, with the same best model as described in Section 3.3 outputting a number count of 8212.5 deg^{-1}. This corresponds to $\sim$16.4 million sources over the 2000 deg^2 survey. Our best-fitting model agrees with the number counts predicted by Zhai et al. (2019) who used GALACTICUS. The 10–90 percentile range of the counts is 7536–9470 deg^{-1}.

3.3.3 *Predictions for the evolution of galaxy bias*

As our model is physically motivated and connects galaxies to dark matter haloes, we can also use GALFORM to predict the effective clustering bias as a function of redshift. The bias is a direct input into the calculation of the signal-to-noise of the clustering measurements. We calculate the asymptotic effective bias in real space using the COLOSSUS package (Diemer 2018), choosing the numerically calibrated bias—halo mass model from Tinker et al. (2010).

In Fig. 13, we plot the results for the effective linear bias ($b_{\rm eff}$) in real-space as a function of redshift for a *Euclid*-like survey (left panel) and for a *Roman*-like survey (right panel). We compute the effective bias for haloes containing an Hα emitter that is brighter than the flux limit of the corresponding survey and in the expected redshift range. We find similar results for the linear real-space

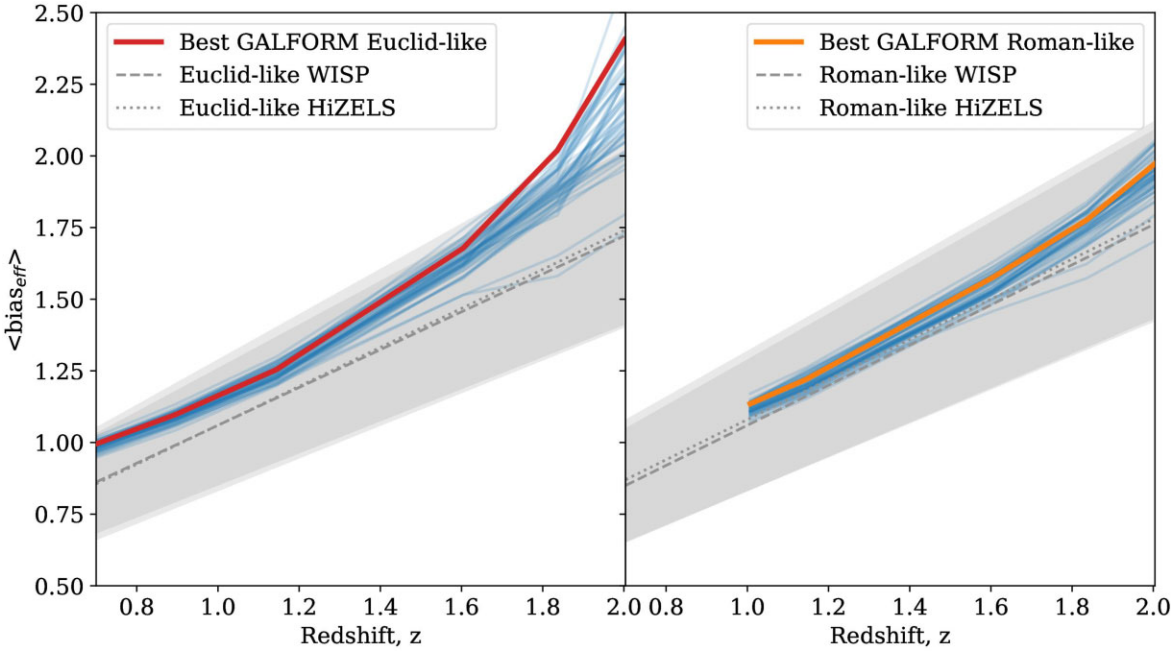

Figure 13. The effective clustering bias for a *Euclid* limited survey (left) and *Roman* (right). In both cases, the blue curves show the 50 best models as a function of redshift, evaluated using GALFORM and using the Colossus routines for computing bias as a function of host halo mass. We have highlighted our best-fitting model to the calibration data set as red (left) or oragne(right) lines. We also plot the fits to the bias predictions from Merson et al. (2019) when adopting a WISP-calibrated lightcone (grey dashed line) and a HiZELS-calibrated lightcone (grey dotted line) and their uncertainty (shading).

bias for the *Euclid* and *Roman* selections. At lower redshifts, the predicted bias has a linear dependence on redshift. This steepens at the highest redshifts shown. The dashed and dotted grey lines show the Merson et al. (2019) models of the linear bias evolution with a WISP- and HiZELS-calibrated models, respectively. Their results show a linear dependence of the effective bias on redshift. Merson et al. calibrate their dust extinction specifically to reproduce the observed Hα luminosity functions. Finally, differences in the choice of cosmologies and bias-halo mass relations will cause slight discrepancies between our predictions and those of Merson et al.

3.3.4 *Comparison to older calibration data sets*

Our best-fitting model is calibrated to the local K- and r-band LFs from Driver et al. (2012). Therefore, we have expanded the comparison data sets to include an older K-band LF from Cole et al. (2001) which was used in the calibration of many previous GALFORM variants. In Fig. 14, we plot our best-fitting GALFORM model $z = 0$ K-band LF, found using our emulator-based MCMC calibrated to the Driver et al. (2012) LF, and compare this with the Cole et al. (2001) K-band LF data. We also plot the Driver et al. (2012) K-band LF for comparison. We see that the Cole et al. (2001) and Driver et al. (2012) data agree reasonably well, particularly for bright galaxies. The consistency between the new local calibration data and the old calibration data indicates that the two observational LFs agree with one another. Therefore, our GALFORM prediction agrees as well with the Cole et al. (2001) data as it does for the Driver et al. (2012) data, up to faint galaxies where the Cole et al. (2001) data is noisier. The Cole et al. (2001) LF estimate overlaps mainly with the brighter Driver et al. (2012) data (as expected given the greater depth of the GAMA survey compared with the 2-degree Field Galaxy Redshift Srvey (2dFGRS) and 2-micron All Sky Survey (2MASS) data used by Cole et al.), and as we have seen in our previous analyses, the

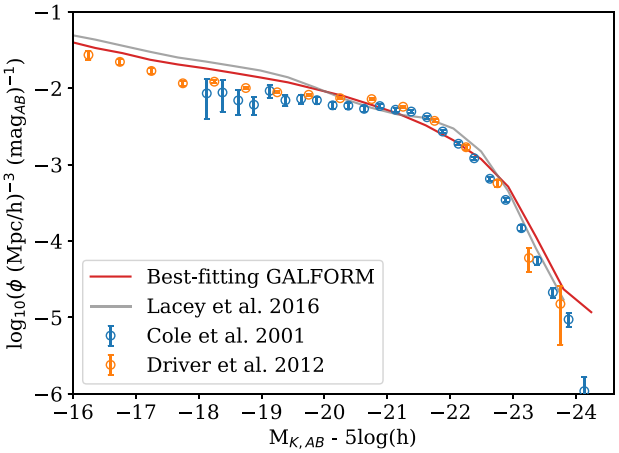

Figure 14. The prediction of the new GALFORM variant (red line) for the $z = 0$ K-band LF compared with the Driver et al. (2012) (orange) and Cole et al. (2001) (blue) data sets. We also plot the GALFORM model by Lacey et al. (2016) (grey line), which was calibrated by hand to several data sets including the Cole et al. LF. Here, we calibrate to the Driver et al. (2012) data.

weighting scheme compromises our new model at the bright end. Therefore, the new model achieves poorer fits at the bright end when compared to the Lacey16 model for the Cole et al. (2001) calibration data also. Our model across all data points scored an MAE of 0.27 compared to the Cole et al. (2001) data; this is worse than the Lacey16 model which achieves an MAE of 0.23. Our model scores worse for similar reasons as previous analysis on the Driver et al. (2012) data; overprediction at the very bright end, and a much shallower turn-over. Nevertheless, our model is still a good approximation to the Cole et al. (2001) data.

4 DISCUSSION AND CONCLUSIONS

We have presented a method for efficiently exploring and calibrating the GALFORM semi-analytical galaxy formation model across a wide range of outputs, building on Elliott et al. (2021). Whereas Elliott et al. focused on using different local data sets in their model calibration, we have also used data at intermediate redshifts, specifically to find models that reproduce current data on the redshift distribution of H α emitters. We calibrated the model over an eleven-dimensional subset of the full model parameter space. We used a deep learning method to mimic running the full GALFORM model. Training the emulator required of the order of 1000 full model runs. With the emulator, we explored the parameter space using MCMC walkers.

We calibrated the model to three sets of observational data: the $z = 0$ galaxy LFs in the r and K bands from Driver et al. (2012) and the redshift distribution of H α emitters at intermediate redshifts from Bagley et al. (2020). However, we did not consider the observational error bars during the model exploration. Instead, we used an absolute error metric (MAE) to quantify the distance between the emulator output and the full model calculations. Hence, it is difficult to provide meaningful uncertainties on the best-fitting parameters. We give an illustration of the uncertainty on the model predictions by plotting the results from the best-fitting model for each MCMC walker, as judged by the model that returned the smallest MAE. We have discovered tensions between the calibration data sets and the model predictions as we could not find equally good fits when all data sets are weighted equally in the MCMC search. The weight given to the H α redshift distribution constraint was increased, moving to a different region of parameter space which modified the fits to the K- and r-band LFs, leading to overprediction at the bright end.

Similarly, we have not considered the uncertainties associated with the emulator. There are two types of uncertainty to account for when emulating model outputs: the uncertainty due to the emulator parameters (that is the weights of the neural network), and the uncertainty inherent in the data generation process (for example, the sampling noise on the GALFORM outputs, such as the bright end of the LF where there are few galaxies). The network hyperparameter space was explored using a trial-and-error process to justify the choice of network architecture. We further reduce uncertainties relating to the weights of the emulator by ensembling individual network estimates.

The majority of variance in the output of our model is due to a few key parameters, which leads to tensions when trying to calibrate to multiple observational data sets. The tensions between the observed data sets were explored, using our MCMC algorithm to fit the emulator output to the constraints, eventually finding the weighting scheme for a global fit to the observations. With this, we find a set of parameters which provides an improved fit to the redshift distribution data as compared with an earlier version of a GALFORM model presented in Lacey16. We go further by producing number count predictions for a *Euclid*-like survey using our best model, improving on previous empirical models by Pozzetti et al. (2016) by using more recent and complete data sets from Bagley et al. (2020). For a flux limit of 2×10^{-16} erg^{-1} s^{-1} cm^{-2} between the redshift range $0.9 < z < 1.8$, our 50 best models predict 2962–4331 H α emission-line sources deg^{-2}, with 3158–3952 sources deg^{-2} between the 10th and 90th percentile. Our best-fitting model estimates 3462.5 sources deg^{-2}, which is comparable to the Bagley et al. (2020) observation. The predictions we produce for the number of galaxies estimated to be seen from the *Euclid* wide field are more constrained than previous models and are better in line with the recent observed number counts. Adding a requirement that the sources are also brighter than $H = 24$ removes only 0.5 per cent of the emitters.

As we are using a physical model that connects galaxies to their host dark matter haloes, we can predict the clustering of H α emitters. Our bias predictions are similar to those of Merson et al. (2019), but with some differences in detail: Merson et al. found that their bias prediction has a linear dependence on redshift, whereas we find that the bias evolves somewhat more rapidly at higher redshift. Similar results like to ours, but without the extensive parameter search and emulation of the semi-analytic model have been presented by Zhai et al. (2019, 2021) and Wang et al. (2022).

We have shown that the method used by Elliott et al. (2021) to automate the calibration of GALFORM can be applied to calibration data that include intermediate redshift observations. Elliot et al. (in preparation) address a similar data calibration challenge using an even more efficient method, Bayesian optimization.

ACKNOWLEDGEMENTS

We thank the anonymous referee for providing a report that helped to improve the paper. We acknowledge the comments from Cedric Lacey. MSM was supported by a CASE PhD Studentship from the Science and Technology Facilities Council (STFCST/S005617/1). CMB and DS acknowledge support from STFC (ST/T000244/1, ST/X001075/1). We used the DiRAC@Durham facility managed by the Institute for Computational Cosmology on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The equipment was funded by BEIS capital funding via STFC grants ST/K00042X/1, ST/P002293/1, ST/R002371/1, and ST/S002502/1, Durham University, and STFC operations grant ST/R000832/1. DiRAC is part of the National e-Infrastructure.

DATA AVAILABILITY

The GALFORM outputs, parameter values, and best-fitting parameters may be shared on reasonable request to the corresponding author.

REFERENCES

Abadi M. et al., 2016, preprint (arXiv:1603.04467)
Albrecht A. et al., 2006, preprint (arXiv:astro–ph/0609591)
Atek H. et al., 2010, ApJ, 723, 104
Bagley M. B. et al., 2020, ApJ, 897, 98
Baugh C. M., 2006, Rep. Prog. Phys., 69, 3101
Baugh C. M. et al., 2019, MNRAS, 483, 4922 (PMILL)
Baugh C. M., Lacey C. G., Gonzalez-Perez V., Manzoni G., 2022, MNRAS, 510, 1880
Benson A. J., 2010, Phys. Rep., 495, 33
Benson A. J., 2012, New Astron., 17, 175
Benson A. J., Bower R., 2010, MNRAS, 405, 1573
Bigiel F. et al., 2011, ApJ, 730, L13
Blitz L., Rosolowsky E., 2006, ApJ, 650, 933
Bower R. G., Benson A. J., Malbon R., Helly J. C., Frenk C. S., Baugh C. M., Cole S., Lacey C. G., 2006, MNRAS, 370, 645
Bower R. G., Vernon I., Goldstein M., Benson A. J., Lacey C. G., Baugh C. M., Cole S., Frenk C. S., 2010, MNRAS, 407, 2017
Christodoulou D. M., Shlosman I., Tohline J. E., 1994, preprint (arXiv:astro-ph/9411031)
Chuang C.-H. et al., 2019, MNRAS, 487, 48
Clevert D.-A., Unterthiner T., Hochreiter S., 2015, preprint (arXiv:1511.07289)
Colbert J. W. et al., 2013, ApJ, 779, 34
Cole S., Lacey C. G., Baugh C. M., Frenk C. S., 2000, MNRAS, 319, 168
Cole S. et al., 2001, MNRAS, 326, 255
Combes F., Debbasch F., Friedli D., Pfenniger D., 1990, A&A, 233, 82
Conroy C., 2013, ARA&A, 51, 393

Cora S. A., 2006, MNRAS, 368, 1540
Cranmer M. D., Xu R., Battaglia P., Ho S., 2019, preprint (arXiv:1909.05862)
de Oliveira R. A., Li Y., Villaescusa-Navarro F., Ho S., Spergel D. N., 2020, preprint (arXiv:2012.00240)
Debattista V. P., Mayer L., Carollo C. M., Moore B., Wadsley J., Quinn T., 2006, ApJ, 645, 209
Diemer B., 2018, ApJS, 239, 35
Driver S. P. et al., 2012, MNRAS, 427, 3244
Dubey S. R., Singh S. K., Chaudhuri B. B., 2022, Neurocomputing, 503, 92
Efstathiou G., Lake G., Negroponte J., 1982, MNRAS, 199, 1069
Elliott E. J., Baugh C. M., Lacey C. G., 2021, MNRAS, 506, 4011
Euclid Collaboration, 2022, A&A, 662, A112
Euclid Collaboration, 2024, preprint (arXiv:2405.13491)
Fanidakis N., Baugh C., Benson A., Bower R., Cole S., Done C., Frenk C., 2011, MNRAS, 410, 53
Font A. S. et al., 2008, MNRAS, 389, 1619
Fu J., Guo Q., Kauffmann G., Krumholz M. R., 2010, MNRAS, 409, 515
Ganaie M. A., Hu M., Malik A., Tanveer M., Suganthan P., 2022, Eng. Appl. Artif. Intell., 115, 105151
Gargiulo I. D. et al., 2015, MNRAS, 446, 3820
Geach J. E., Smail I., Best P., Kurk J., Casali M., Ivison R., Coppin K., 2008, MNRAS, 388, 1473
Glorot X., Bengio Y., 2010, in Teh Y. W., Titterington M., eds, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9, PMLR, p. 249
Goldstein M., Wooff D., 2007, Bayes Linear Statistics: Theory and Methods. John Wiley and Sons, Chichester, England, Hoboken, New Jersey
Gonzalez-Perez V., Lacey C. G., Baugh C. M., Lagos C., Helly J., Campbell D., Mitchell P. D., 2014, MNRAS, 439, 264
Graham A. W., Driver S. P., Allen P. D., Liske J., 2007, MNRAS, 378, 198
Granato G. L., Lacey C., Silva L., Bressan A., Baugh C., Cole S., Frenk C., 2000, ApJ, 542, 710
Griffin A., Lacey C., Gonzalez-Perez V., Lagos C., Baugh C., Fanidakis N., 2019, MNRAS, 487, 198
Häring N., Rix H.-W., 2004, ApJ, 604, L89
He S., Li Y., Feng Y., Ho S., Ravanbakhsh S., Chen W., Póczos B., 2019, Proc. Natl. Acad. Sci. USA, 116, 13825
Henriques B. M., Thomas P. A., Oliver S., Roseboom I., 2009, MNRAS, 396, 535
Hopkins A. M., Beacom J. F., 2006, ApJ, 651, 142
Kampakoglou M., Trotta R., Silk J., 2008, MNRAS, 384, 1414
Kennedy J., Eberhart R., 1995, in Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, p.1942
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Knebe A. et al., 2022, MNRAS, 510, 5392
Lacey C. G., Baugh C. M., Frenk C. S., Benson A. J., 2011, MNRAS, 412, 1828
Lacey C. G. et al., 2016, MNRAS, 462, 3854 (Lacey16)
Lagos C. d. P., Cora S. A., Padilla N. D., 2008, MNRAS, 388, 587
Lagos C. d. P., Lacey C. G., Baugh C. M., Bower R. G., Benson A. J., 2011, MNRAS, 416, 1566
Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
Loh W.-L., 1996, Ann. Stat., 24, 2058
Lu L., 2020, Commun. Comput. Phys., 28, 1671
Lu Y., Mo H., Weinberg M. D., Katz N., 2011, MNRAS, 416, 1949
Lu Y., Mo H., Katz N., Weinberg M. D., 2012, MNRAS, 421, 1779
Lu Y., Mo H., Lu Z., Katz N., Weinberg M. D., 2014, MNRAS, 443, 1252
Maas A. L., Hannun A. Y., Ng A. Y. 2013, J. Mach. Learn. Res., 28, 3
Maraston C., 2005, MNRAS, 362, 799
Martin A. M., Papastergis E., Giovanelli R., Haynes M. P., Springob C. M., Stierwalt S., 2010, ApJ, 723, 1359
Martindale H., Thomas P. A., Henriques B. M., Loveday J., 2017, MNRAS, 472, 1981
Mehta V. et al., 2015, ApJ, 811, 141
Merson A. I. et al., 2013, MNRAS, 429, 556
Merson A., Wang Y., Benson A., Faisst A., Masters D., Kiessling A., Rhodes J., 2018, MNRAS, 474, 177
Merson A., Smith A., Benson A., Wang Y., Baugh C., 2019, MNRAS, 486, 5737
Nair V., Hinton G. E., 2010, Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10), Omnipress, p. 807
Norberg P. et al., 2002, MNRAS, 336, 907
Ntampaka M. et al., 2019, Bull. Am. Astron. Soc., 51, 14
Opitz D., Maclin R., 1999, J. Artif. Int. Res., 11, 169
Orsi A., Baugh C., Lacey C., Cimatti A., Wang Y., Zamorani G., 2010, MNRAS, 405, 1006
Padilla N. D., Salazar-Albornoz S., Contreras S., Cora S. A., Ruiz A. N., 2014, MNRAS, 443, 2801
Perraudin N., Srivastava A., Lucchi A., Kacprzak T., Hofmann T., Réfrégier A., 2019, Comput. Astrophys. Cosmol., 6, 1
Popping G., Somerville R. S., Trager S. C., 2014, MNRAS, 442, 2398
Pozzetti L. et al., 2016, A&A, 590, A3
Ravanbakhsh S., Oliva J., Fromenteau S., Price L., Ho S., Schneider J., Póczos B., 2016, International Conference on Machine Learning. PMLR, New York, USA, p. 2407
Reddi S. J., Kale S., Kumar S., 2019, preprint (arXiv:1904.09237)
Reyes-Peraza G., Avila S., Gonzalez-Perez V., Lopez-Cano D., Knebe A., Ramakrishnan S., Yepes G., 2024, MNRAS, 529, 3877
Robert C. P., Casella G., Robert C. P., Casella G., 2004, Monte Carlo Statistical Methods. Springer-Verlag, Berlin, p. 267
Rodrigues L. F. S., Vernon I., Bower R. G., 2017, MNRAS, 466, 2418
Ruiz A. N. et al., 2015, ApJ, 801, 139
Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533
Sagi O., Rokach L., 2018, WIREs Data Mining Knowl. Discov., 8, e1249
Schmit C. J., Pritchard J. R., 2018, MNRAS, 475, 1213
Shim H., Colbert J., Teplitz H., Henry A., Malkan M., McCarthy P., Yan L., 2009, ApJ, 696, 785
Simha V., Cole S., 2017, MNRAS, 472, 1392
Sobral D. et al., 2009, MNRAS, 398, 75
Sobral D., Best P. N., Matsuda Y., Smail I., Geach J. E., Cirasuolo M., 2012, MNRAS, 420, 1926
Sobral D., Smail I., Best P. N., Geach J. E., Matsuda Y., Stott J. P., Cirasuolo M., Kurk J., 2013, MNRAS, 428, 1128
Spergel D. et al., 2015, preprint (arXiv:1503.03757)
Springel V., White S. D., Tormen G., Kauffmann G., 2001, MNRAS, 328, 726
Sun Y., Wang X., Tang X., 2015, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, p. 2892
Tieleman T., Hinton G., 2012, COURSERA: Neural Netw. Mach. Learn., 4, 26
Tinker J. L., Robertson B. E., Kravtsov A. V., Klypin A., Warren M. S., Yepes G., Gottlöber S., 2010, ApJ, 724, 878
Tran P. T. et al., 2019, IEEE Access, 7, 61706
Valentino F. et al., 2017, MNRAS, 472, 4878
Vernon I., Goldstein M., Bower R. G., 2010, Bayesian Anal., 5, 697
Wang Y. et al., 2022, ApJ, 928, 1
Wolpert D. H., 1992, Neural Netw., 5, 241
Xu B., Wang N., Chen T., Li M., 2015, preprint (arXiv:1505.00853)
Zhai Z., Benson A., Wang Y., Yepes G., Chuang C.-H., 2019, MNRAS, 490, 3667
Zhai Z., Wang Y., Benson A., Colbert J., Bagley M., Henry A., Baronchelli I., 2021, preprint (arXiv:2109.12216)
Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, preprint (arXiv:1902.05965)
Zwaan M. A., Meyer M. J., Staveley-Smith L., Webster R. L., 2005, MNRAS, 359, L30

APPENDIX A: OTHER MODEL PREDICTIONS

Here, we present some further predictions of the new model. These data sets were not used to constrain the model parameters. Fig. A1 shows the local atomic hydrogen (H I) (top left). The new model (red)

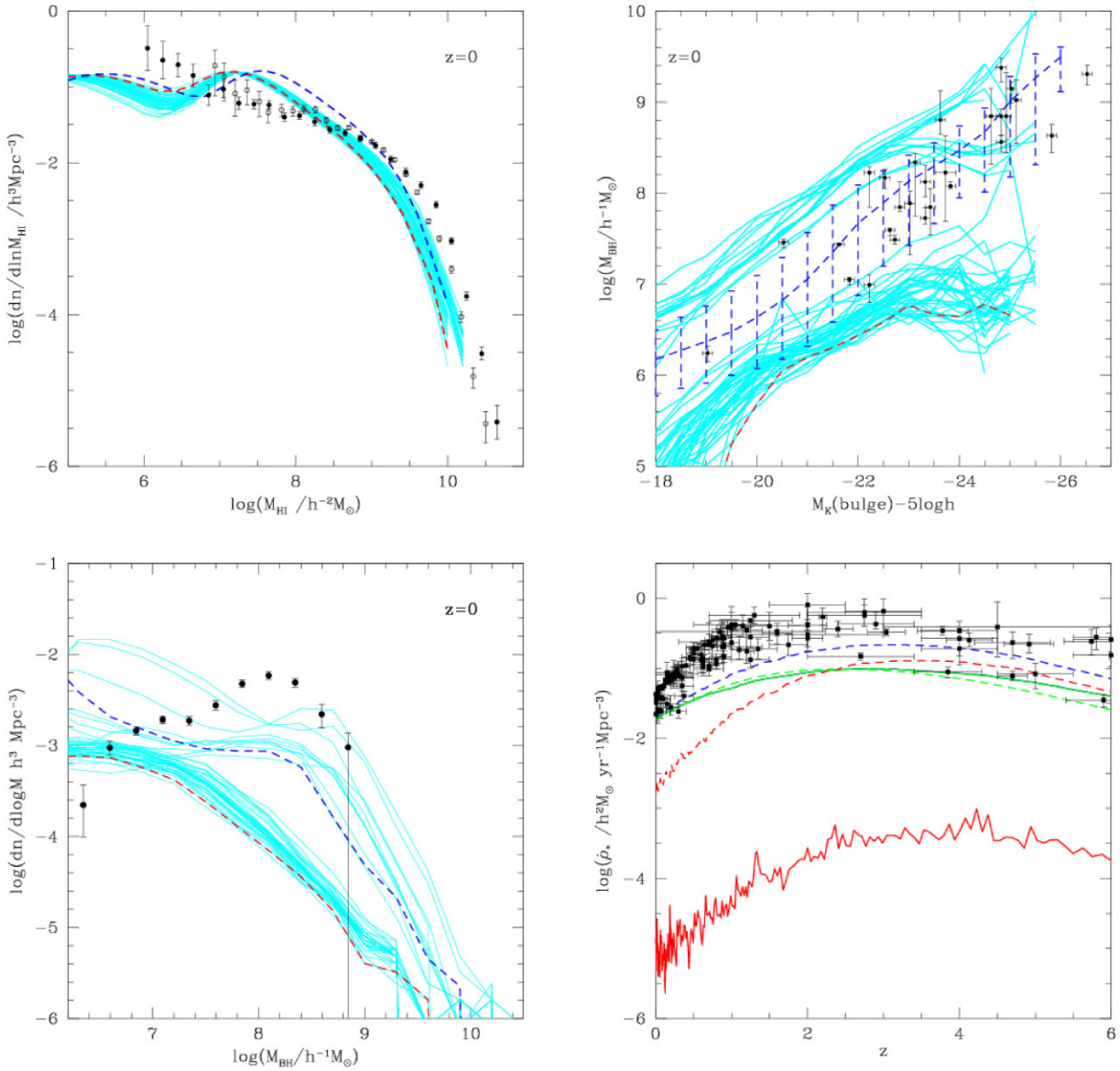

Figure A1. Top left: The prediction of the new GALFORM variant (red line) for the $z = 0$ H I mass function compared with the Zwaan et al. (2005) (open circles) and Martin et al. (2010) (filled circles) data sets. The cyan curves show the predictions of other best-fitting models from each MCMC chain. We also plot the GALFORM model by Lacey16 (dark blue dashed line). Top right: predictions for the SMBH mass versus bulge K-band magnitude. The red curve shows the best-fitting new model and cyan lines show the other MCMC chain best-fitting models. The Lacey et al. model is shown by the blue dashed line with errorbars that show the 10–90 percentile range of the model predictions. The black points show observational estimates from Häring & Rix (2004). Bottom left: SMBH mass function. The best-fitting model is red, the cyan curves show the best-fitting models from other MCMC chains and the blue dashed line shows Lacey et al. The black points with errorbars show an observational estimate from Graham et al. (2007); in this case the bars indicate the interquartile range. Bottom right: global star formation rate density (SFRD) versus redshift. The best-fitting model is shown by solid lines and Lacey et al. by dashed lines. Blue shows total SFRD, green quiescent SFRD, and red bursts. The black points show a compilation of observational estimates from Hopkins & Beacom (2006).

underpredicts the high-mass end and is lower than the Lacey et al. prediction. We also show the best-fitting model from each MCMC chain to gain an impression of the range of 'acceptable' predictions (cyan lines). These predictions extend beyond the Lacey et al. model at the highest H I masses.

The top left and bottom right panels of Fig. A1 show predictions for SMBH. The best-fitting model predicts somewhat lower SMBH for a given bulge K-band luminosity but still overlaps with the observations. The range of best-fitting models is quite large and seems to bifurcate into two regions. The Lacey et al. model is in good agreement with the observations. The bottom left panel presents the SMBH mass function. Again the new model gives a lower SMBH mass function than Lacey et al., but the range of best-fitting models is broad, splits into two groups and encloses the Lacey et al. prediction.

Finally, the bottom right panel of Fig. A1 shows the global star formation rate density (SFRD) as a function of redshift. The best-fitting model and Lacey et al. models have very similar quiescent SFRDs. The new model has a much smaller SFRD in starbursts than the Lacey et al. model; this is due to the much stronger SNe feedback in bursts in the new model.

This paper has been typeset from a TEX/LATEX file prepared by the author.