

Efficient exploration and calibration of a semi-analytical model of galaxy formation with deep learning

Edward J. Elliott,^{1,2}★ Carlton M. Baugh^{1,2} and Cedric G. Lacey¹

¹*Institute for Computational Cosmology, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK*

²*Institute for Data Science, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK*

Accepted 2021 June 22. Received 2021 June 21; in original form 2021 March 1

ABSTRACT

We implement a sample-efficient method for rapid and accurate emulation of semi-analytical galaxy formation models over a wide range of model outputs. We use ensembled deep learning algorithms to produce a fast emulator of an updated version of the GALFORM model from a small number of training examples. We use the emulator to explore the model’s parameter space, and apply sensitivity analysis techniques to better understand the relative importance of the model parameters. We uncover key tensions between observational data sets by applying a heuristic weighting scheme in a Markov chain Monte Carlo framework and exploring the effects of requiring improved fits to certain data sets relative to others. Furthermore, we demonstrate that this method can be used to successfully calibrate the model parameters to a comprehensive list of observational constraints. In doing so, we re-discover previous GALFORM fits in an automatic and transparent way, and discover an improved fit by applying a heavier weighting to the fit to the metallicities of early-type galaxies. The deep learning emulator requires a fraction of the model evaluations needed in similar emulation approaches, achieving an out-of-sample mean absolute error at the knee of the *K*-band luminosity function of 0.06 dex with less than 1000 model evaluations. We demonstrate that this is an extremely efficient, inexpensive, and transparent way to explore multidimensional parameter spaces, and can be applied more widely beyond semi-analytical galaxy formation models.

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

1 INTRODUCTION

Galaxy formation is a complex and non-linear process, involving the interplay of gravitational, radiative, thermal, and fluid processes. Semi-analytical modelling is an approach used to improve our understanding of this problem by reducing it to its key ingredients using simplified mathematical relations motivated by physical and geometric arguments (e.g. Baugh 2006; Benson 2010). These relations take the form of coupled differential equations and simple algebraic relations describing processes such as star formation, gas cooling, and bar instabilities in galactic discs. Semi-analytical models (SAMs) provide a comprehensive theoretical framework with which to understand and develop intuition about galaxy formation, and have produced a number of insights (e.g. White & Frenk 1991; Benson et al. 2003; Bower et al. 2006; Croton et al. 2006; Lacey et al. 2016).

However, the semi-analytical approach has sometimes attracted scepticism for a number of reasons. The mathematical relations that describe the physical processes in the model often contain adjustable parameters, and a model is defined by a particular choice for the parameter values (analogous to the parametrized subgrid models employed in hydrodynamic simulations, e.g. Crain et al. 2015; Somerville & Davé 2015). These parameters are sometimes set by theoretical or observational considerations, but in many cases they

are less well specified (e.g. in the case of the parameters governing the strength of feedback due to supernovae – SNe).

There is a perception – which we believe to be misplaced – that these ‘free’ parameters allow SAMs to fit any arbitrary combination of data sets, therefore eliminating their predictive and explanatory power. We hope to dispel this view by demonstrating that the majority of the variance in the model output is contributed by just a few parameters that have clear physical interpretations [such as the strength of feedback due to SNe or active galactic nucleus (AGN)], and that these dominant parameters preclude arbitrary fitting.

Another major source of the scepticism towards SAMs arises from the seemingly opaque procedures that have commonly been used to calibrate the model parameters. This process often follows a ‘chi-by-eye’ methodology, in which the operator adjusts the parameters by hand, interprets the effect on the model output, and adjusts the parameters again to improve the match of the model output to an observable. This requires a high level of expertise and familiarity with the SAM, and the operator often makes trade-offs between fits to different constraining data sets in a way that is poorly defined; model predictions are often judged to be good fits when formally they would be rejected. This makes the process of setting the model parameters hard to reproduce. There is also no guarantee that the by-eye approach will produce the best fit to the calibration data sets; the model parameters may interact in a non-linear way, which can be difficult to conceptualize. This, coupled with the large parameter space, makes it unlikely that such a search will find the best-fitting parameters. We aim to side-step these issues by developing

* E-mail: edward.j.elliott@durham.ac.uk

a method to rapidly and robustly perform an exhaustive search of the parameter space, calibrate the SAM in an automatic way without the need for significant human intervention, and quantify the relative importance of the parameters. In this way, we hope to make the model calibration process transparent and reproducible, especially by researchers with less experience of running SAMs. Although the cosmological parameters are now well constrained, SAMs must still be re-tuned for simulations with different resolutions and cosmologies, such as $f(R)$ gravity simulations, or when a new implementation of a process is included. The question of how to set the model parameters therefore remains a relevant one.

The calibration and exploration of SAMs is not a new problem, and has been investigated in a number of previous works. This effort has generally taken two forms: direct exploration of the model parameter space and emulation. Although SAMs are orders of magnitude cheaper than hydrodynamic simulations, direct exploration of their parameter space is computationally expensive due to the sheer number of model runs required for a formal search; often this will take a prohibitive length of time except for the case of tuning the parameters to a small number of data sets. This approach has been investigated in a number of papers. Kampakoglou, Trotta & Silk (2008) implemented Markov chain Monte Carlo (MCMC) techniques to calibrate a SAM to multiple data sets. Henriques et al. (2009) again used MCMC to calibrate the L-GALAXIES SAM to a number of data sets, finding that the choice of data sets altered the values of the best-fitting parameters, pointing to deficiencies in their model. Martindale et al. (2017) expanded on this to include the HI mass function as a constraint, leading to a change in the best-fitting parameters. Lu et al. (2011, 2012) constrained the parameter space that gave acceptable fits to the K -band luminosity function (LF), and expanded this to include the HI mass function in Lu et al. (2014). Ruiz et al. (2015) used particle swarm optimization to calibrate a SAM to the K -band LF. The second class of methods involves constructing a statistical emulator of the SAM that can be evaluated orders of magnitude more quickly than running the SAM itself, but at the cost of being approximate by nature. Bower et al. (2010) and Vernon, Goldstein & Bower (2010) employed a Bayesian emulation technique (as developed by Goldstein & Wooff 2007) to constrain the parameter space that can provide reasonable fits to the K - and b_J -band LFs, and extended this in Benson & Bower (2010) to explore this ability of this reduced parameter space to fit to further observational data sets. This approach has also been applied by Rodrigues, Vernon & Bower (2017) to calibrate the GALFORM SAM to the galaxy stellar mass function in the local Universe, and recently by Van Der Velden et al. (2021) to calibrate the MERAXES SAM at high redshift.

Here, we aim to emulate an updated version of the GALFORM code implemented in the Planck Millennium N -body simulation (Baugh et al. 2019), which uses an improved galaxy merger scheme (devised by Simha & Cole 2017 and first implemented GALFORM by Campbell et al. 2015), but which also includes an improved model for gas cooling in haloes (introduced by Hou, Lacey & Frenk 2018).

We focus specifically on using deep learning to build our emulator (for an introductory review, see e.g. Emmert-Streib et al. 2020). This subfield of machine learning uses stacked neural layers (hence *deep*) to build flexible function approximators that are able to uncover non-linear relations in data without the need for a strongly pre-defined model, and have proven to be highly successful in astronomical applications (e.g. Ravanbakhsh et al. 2016; Schmit & Pritchard 2018; Cranmer et al. 2019; He et al. 2019; Perraudin et al. 2019; Zhang et al. 2019; De Oliveira et al. 2020; Ntampaka et al. 2019). We demonstrate that deep learning algorithms can be applied to

accurately emulate SAMs over the full range of model outputs, and require a relatively small number of training examples to achieve good accuracy when compared to other methods. Since the deep learning emulator can be evaluated orders of magnitude faster than the time taken to run GALFORM, we are able to run many simple MCMC chains to explore the parameter space, and investigate how calibration to different data sets constrains the model parameters. We achieve this by minimizing the absolute error between the emulator output and the data, and employing a heuristic weighting scheme to the different observational data sets to mimic the process employed by model practitioners. In this way, we hope to elucidate and automate the calibration process, as well as exhaustively search the parameter space of the model. A similar approach has been explored in Forbes, Krumholz & Speagle (2019), applied to a SAM of galactic discs, though our sampling scheme and MCMC implementation are different.

This approach has a number of advantages over previous work. Non-emulation approaches such as MCMC and particle swarm optimization offer a powerful way to quantify parameter uncertainty and fit the model to a particular observable, but are limited in terms of exploring and understanding the full parameter space, and come at significant computational expense. Previous emulation approaches, though informative, also do not fully address our aims; they have focused on reducing the parameter space based on measures of implausibility (a measure that incorporates information about the emulator prediction and target data, and their variances, to rule out regions of parameter space). By iteratively refining more approximate emulators over a number of waves of model runs, these methods hone in on a region of parameter space that could plausibly contain good fits to a pre-defined set of just a few observables. Here, we focus on producing an emulator of the GALFORM model that is accurate across the entire parameter space. This allows us to explore the full parameter space of the model and fit to a wide range of observables, and to consider more diverse combinations of observables than has been attempted in previous work. We also aim to reduce the requirement for a large number of GALFORM evaluations. Rodrigues et al. (2017), for example, used seven waves of 5000 runs each to hone in on the region of parameter space that gave acceptable fits to the local galaxy stellar mass function; here, we limit ourselves to <1000 full GALFORM runs. In doing so, we intend to develop a general method for investigating, understanding, and calibrating SAMs in an inexpensive, flexible, and reproducible way.

We also apply sensitivity analysis techniques to the model parameters, as recently applied to GALFORM by Oleskiewicz & Baugh (2019). This allows us to judge the importance of different parameters by quantifying the proportion of the model variance due to a given parameter through sensitivity indices. We are also able to gauge the degree of interaction between parameters, giving us important insight into the model.

The layout of the paper is as follows. In Section 2, we review the theoretical background. We describe the key processes of GALFORM that are relevant to this work in Section 2.1. In Section 2.2, we give a brief review of the deep learning approach and our emulator design, and in Section 2.3 we give a description of the sensitivity analysis method. In Section 2.4, we describe the observational constraints under consideration, and in Section 2.5 we discuss how we find best-fitting parameters using MCMC. In Section 3, we present our results. In Section 3.1, we review the predictive performance of the emulator, in Section 3.2 we show the results of our sensitivity analysis and model exploration, and in Section 3.3 we present our model calibration results. In Section 4, we discuss the merits of our methods and outline potential future work, and conclude in Section 5.

2 THEORETICAL BACKGROUND

Here, we briefly describe aspects of GALFORM pertinent to this work (Section 2.1) and describe the process of building a deep learning emulator and motivate our specific choice of model (Section 2.2). We then briefly describe sensitivity analysis (Section 2.3), the observational data sets considered (Section 2.4), and our calibration scheme (Section 2.5).

2.1 GALFORM

GALFORM is a state-of-the-art *ab initio* physically motivated SAM of galaxy formation. The model tracks the merger histories of dark matter haloes, the cooling of gas to form galactic discs, quiescent star formation in the disc, bursts of star formation associated with mergers or disc instabilities, the resultant feedback and gas ejection driven by SNe, the role of heating by AGN in inhibiting gas cooling, and the chemical enrichment of stars and gas (for a full description of GALFORM, see Cole et al. 2000; Lacey et al. 2016). Here, we review some aspects of the code relevant to this work and the following discussion.

2.1.1 Quiescent star formation in discs

The model uses an empirical star formation law formulated by Blitz & Rosolowsky (2006) (and implemented in GALFORM in Lagos et al. 2011) based on observations of nearby star-forming disc galaxies. The star formation rate in the disc is

$$\psi_{\text{disc}} = \nu_{\text{SF}} M_{\text{mol, disc}}, \quad (1)$$

where $M_{\text{mol, disc}}$ is the mass of molecular gas in the disc and ν_{SF} is a constant that we treat as an adjustable parameter within a reasonable range (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

SN feedback causes gas to be ejected from galaxies. The model assumes that this mass ejection is proportional to the instantaneous star formation rate, ψ , with a mass loading factor dependent on the circular velocity of the galaxy, V_c :

$$\dot{M}_{\text{eject}} = \left(\frac{V_c}{V_{\text{SN}}} \right)^{-\gamma_{\text{SN}}} \psi, \quad (2)$$

where both V_{SN} and γ_{SN} are model parameters. We can further distinguish V_{SN} into $V_{\text{SN, disc}}$ and $V_{\text{SN, burst}}$, allowing for different values for feedback in quiescent star formation and bursts, although these parameters have generally been assumed to be equal in most previous versions of the model. Gas ejected from the halo is assumed to gradually return from a reservoir beyond the halo's virial radius to the hot gas reservoir at a rate given by

$$\dot{M}_{\text{return}} = \alpha_{\text{ret}} \frac{M_{\text{res}}}{\tau_{\text{dyn, halo}}}, \quad (3)$$

where $\tau_{\text{dyn, halo}}$ is the dynamical time of the halo, M_{res} is the mass in the reservoir beyond the virial radius, and α_{ret} is a free parameter.

2.1.3 Galaxy mergers

In the model, galaxy mergers can trigger bursts of star formation and destroy galactic discs. We define two different thresholds, f_{ellip} and f_{burst} . When a satellite galaxy with baryonic mass $M_{\text{b, sat}}$ merges

Table 1. The GALFORM parameters under investigation. See Section 2.1 for the equations that define the symbols in the first column.

Parameter	Range	Process
f_{stab}	0.61–1.1	Disc instability
α_{cool}	0.2–1.2	AGN feedback
α_{ret}	0.2–1.2	SN feedback
γ_{SN}	1.0–4.0	SN feedback
$V_{\text{SN, disc}}$ (km s ⁻¹)	100–550	SN feedback
$V_{\text{SN, burst}}$ (kms ⁻¹)	100–550	SN feedback
f_{burst}	0.01–0.3	Mergers
f_{ellip}	0.2–0.5	Mergers
ν_{SF} (Gyr ⁻¹)	0.2–1.7	Quiescent star formation
f_{SMBH}	0.001–0.05	SMBH accretion

with a central galaxy with baryonic mass $M_{\text{b, cen}}$ two types of mergers may occur. First, if $M_{\text{b, sat}}/M_{\text{b, cen}} \geq f_{\text{ellip}}$ the merger is classified as a *major* merger, and the disc component of the galaxy is destroyed and forms a spheroid. The cold gas in the disc is assumed to be consumed in a burst of star formation. Secondly, if $M_{\text{b, sat}}/M_{\text{b, cen}} < f_{\text{ellip}}$, the merger is classified as *minor*, and the disc survives the merger. In this case, the cold gas is consumed in a starburst if a second condition is met, $M_{\text{b, sat}}/M_{\text{b, cen}} \geq f_{\text{burst}}$. Both f_{burst} and f_{ellip} are treated as free parameters. In the improved galaxy merger model of Simha & Cole (2017), once a subhalo can no longer be resolved, an analytic calculation of the merger time is made based on dynamical friction arguments.

2.1.4 Disc instabilities

Galactic discs dominated by rotational motion can become unstable to bar formation if their degree of self-gravity is too large. The model follows the work of Efstathiou, Lake & Negroponte (1982), and assumes that discs become unstable if the criterion

$$\frac{V_c(r_{\text{disc}})}{(1.68GM_{\text{disc}}/r_{\text{disc}})^{1/2}} \leq f_{\text{stab}} \quad (4)$$

is met, where M_{disc} is the total disc mass and r_{disc} is the disc half-mass radius. Numerical simulations of a suite of exponential stellar discs by Efstathiou et al. (1982) found a value of $f_{\text{stab}} \approx 1.1$, while Christodoulou, Shlosman & Tohline (1995) found a value of 0.9 for gaseous discs. A value of 0.61 or below corresponds to universally stable discs, since this is the value of the left-hand side of equation (4) for a completely self-gravitating disc. We allow this parameter to vary within a reasonable range (see Table 1). We assume that unstable discs are disrupted by bar instabilities on a subresolution time-scale such that all the mass is instantaneously transferred to the spheroid and any gas present takes part in a burst of star formation.

2.1.5 SMBH growth and AGN feedback

Supermassive black holes (SMBH) can inject energy into the halo gas, disrupting gas cooling. Hot halo accretion, BH–BH mergers, as well as starbursts can increase the mass of the black hole (Bower et al. 2006; Griffin et al. 2019). In the case of starbursts, the mass accreted on to the SMBH is a fraction f_{SMBH} of the mass of stars formed, where f_{SMBH} is an adjustable parameter. AGN accretion is assumed to occur if both of the following conditions are met: (1) that the gas halo is in quasi-hydrostatic equilibrium, that is the condition

$$\tau_{\text{cool}}/\tau_{\text{ff}} > 1/\alpha_{\text{cool}}, \quad (5)$$

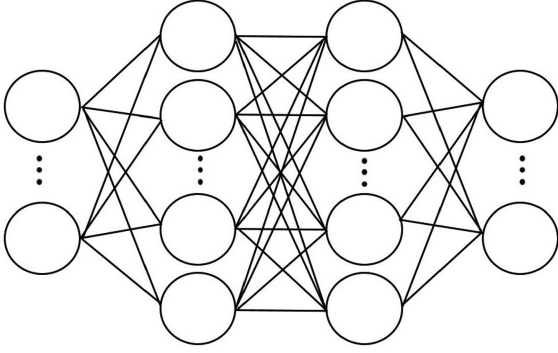


Figure 1. A schematic diagram showing a neural network with two hidden layers. The neurons on the left-hand side represent the input layer, the central two layers of neurons are the hidden layers, and the right-hand neurons comprise the output layer.

is met, where τ_{cool} is the cooling time of the gas, τ_{ff} the free-fall time, and α_{cool} is an adjustable parameter; (2) the AGN power required to balance the radiative cooling luminosity is below a fraction f_{Edd} of the Eddington luminosity of the SMBH.

2.2 Deep learning emulator

Before we consider observational data, we aim to construct a fast emulator of the GALFORM model using the TENSORFLOW deep learning framework (Abadi et al. 2015). We formulate the problem from the perspective of supervised learning. We treat the GALFORM model as an unknown function $\hat{f}(\cdot)$ that takes some input vector \mathbf{x} , representing a set of values for the model parameters, and produces an output vector \mathbf{y} , representing one or many binned statistical properties of the resulting synthetic galaxy population (e.g. the values of the K -band LF in given luminosity bins). Our goal is then to develop a fast and accurate approximation to the function $\hat{f}(\cdot)$ by training an emulator to predict \mathbf{y} given \mathbf{x} .

Since GALFORM is computationally expensive to run (at least in comparison to a potential deep learning emulator), we are limited in how many evaluations of the code we can perform, and so limited in the number of input–output pairs, $(\mathbf{x}_i, \mathbf{y}_i)$, we have to train our emulator. To sample the parameter space evenly and efficiently, we use Latin hypercube sampling (as described in e.g. Bower et al. 2010) to generate the model input parameters. This method aims to fill the target parameter space evenly given a fixed number of samples. After evaluating GALFORM at these points, we are therefore left with the pairs of vectors $(\mathbf{x}_i, \mathbf{y}_i)$, corresponding to the input and output of the model. We separate the samples randomly into three sets: the training set, the validation set, and the holdout set. The training and validation sets will be used to train the emulator, and the holdout set will be kept separate so it can be used for evaluating the emulator’s performance on out-of-sample data. The different roles of these sets are discussed further below.

The task of emulating GALFORM is therefore reduced to a regression problem. The deep learning emulator is comprised of stacked neural layers as shown in Fig. 1. Here we see a neural network with an input layer on the left, two *hidden* layers, and an output layer on the right. Note that the output from each neuron is passed to every neuron in the following layer. The network is defined by a set of weights and biases, W ; the i -th neuron in the j -th layer contains an adjustable weight vector \mathbf{w}_{ij} and an adjustable bias term b_{ij} . When we propagate inputs through our network to produce a prediction, the input layer first passes the inputs to every neuron in the first hidden

layer. Each neuron i in each subsequent layer j , starting with the first hidden layer, takes in the outputs from the previous layer and calculates its own output z_{ij} by performing the computation

$$z_{ij} = \hat{\sigma}(\mathbf{z}_{j-1} \cdot \mathbf{w}_{ij} + b_{ij}), \quad (6)$$

where we have taken the dot product between the vector of all the neuron outputs in the previous layer \mathbf{z}_{j-1} and the i -th neuron’s weights \mathbf{w}_{ij} , and added the bias term b_{ij} . An activation function $\hat{\sigma}(\cdot)$ is then applied. This is generally a non-linear function such as the sigmoid or hyperbolic tangent function. The neuron outputs z_{ij} are then passed to the next layer and the process is repeated until we reach the final layer. The output from the final layer is then the prediction of the network for these inputs. Usually, the neurons in the final layer only apply a linear activation function. Therefore, since the network outputs are linear sums of non-linear functions of the input parameters, we can think of this method as estimating non-linear basis functions from training data.

The weights and biases associated with each neuron are adjusted during training by seeking to minimize an error function between the emulator predictions and the true values. In our case, given a set of input parameters, we want to minimize the error between our emulator’s prediction of the GALFORM output and the actual GALFORM output. We choose to use the mean absolute error function (hereafter MAE)

$$\text{MAE} = \frac{1}{n} \sum_{k=1}^n |\hat{\mathbf{y}}_k - \mathbf{y}_k|, \quad (7)$$

where $\hat{\mathbf{y}}_k$ is the model emulator prediction for the k -th of n samples and \mathbf{y}_k is the true value. Since both $\hat{\mathbf{y}}_k$ and \mathbf{y}_k are vector quantities, the modulus signs represent the L1 norm (i.e. the sum of absolute errors of the vector components); we choose this metric as it gives less weight to outliers than the more commonly used L2 norm (i.e. the sum of squared errors of the vector components). If we denote the function represented by the neural network as f , parametrized by weights and biases W , we therefore attempt to find a function f_* such that

$$f_* = \arg \min_W \{\text{MAE}(f(\mathbf{x}), \mathbf{y})\}. \quad (8)$$

The training is performed iteratively in steps known as *epochs*. During each epoch, the model weights and biases, W , are adjusted by an optimizer to minimize the MAE of the network’s predictions for the training set. The optimizer is an algorithm that calculates how best to adjust the model weights by seeking minima on the error surface, usually by some form of gradient descent. We use the AMSGRAD variation of the Adam optimizer (Kingma & Ba 2015; Reddi, Kale & Kumar 2018). Adam is a momentum-based optimizer and AMSGRAD aims to improve the performance of Adam around minima on the error surface. At the end of each epoch, the adjusted model is evaluated on the validation set, to ensure the model generalizes to unseen data. If the performance on the validation set has improved (as measured by the MAE), we save the model weights and continue training. If the performance does not improve, we do not save the weights and continue training. This process is repeated until the performance on the validation set has not improved for 30 epochs at which point we halt the training. We then perform a final fine-tuning step using the RMSprop optimizer (Tieleman & Hinton 2012); this optimizer uses stochastic gradient descent and treats the error surface as a quadratic bowl. For this step, we use a very low learning rate of 10^{-5} , allowing us to take small gradient steps towards the minima of the error surface. We find this works well in boosting the performance of our emulator. We then evaluate

the model on the holdout set to ensure its performance generalizes to entirely unseen data (since we selected model weights which perform best on the validation set, the validation set itself is not a good test of out-of-sample performance).

2.2.1 Inputs and outputs

The aim of our emulator is to map an input vector \mathbf{x} , the GALFORM parameters, on to an output vector \mathbf{y} , the statistical galaxy properties that we wish to predict. Our choice of input parameters is informed by previous analyses (e.g. Lacey et al. 2016; Oleskiewicz & Baugh 2019), and we aim to emulate the effects of the parameters associated with the key processes outlined in Section 2.1. These parameters and their ranges are shown in Table 1. We train our emulator to predict a wide range of statistical galaxy properties calculated from the output of GALFORM. These are the K - and r -band LFs at $z = 0$, the early- and late-type galaxy sizes, the HI mass function, the early-type fraction with r -band magnitude, the I -band Tully–Fisher relation, the bulge–black hole mass relation, and the metallicities of early-type galaxies.

2.2.2 Model architecture

We find that a simple architecture is sufficient to accurately emulate GALFORM. We use a densely connected neural network, meaning that every neuron is connected to every neuron in the previous layer. We use two hidden layers, each with 512 neurons and sigmoid activation functions, and linear activations on the output layer. We investigated a number of other architectures, such as stacking long short term memory (LSTM; Hochreiter & Schmidhuber 1997) and 1D convolutional layers to try to exploit features of the data, but found limited improvement at the cost of slower evaluation speed.

2.2.3 Ensembling

Training a neural network is a stochastic process. The network weights are often initialized according to some distribution (e.g. Glorot & Bengio 2010), and the optimizer traverses the weight space using gradient steps calculated on mini-batches of the full data set (i.e. a small subset of the whole training set at a time), and so is inherently stochastic. This means that training a single network is suboptimal. Since the error surface is likely to contain many local minima we are unlikely to find the best possible network weights with one network alone, and each network will develop its own idiosyncrasies in how it fits the data. Neural networks also contain a vast number of parameters, and are therefore prone to overfitting. One way to address these problems is ensembling. This involves training a handful of networks with different weight initializations and combining the individual predictions. We can also shuffle the validation and training sets for each model in the ensemble, so that each model is exposed to a different distribution of input–output pairs. In general, this allows for a more robust prediction. Individual models may over- or underfit to different features of the data, and combining predictions averages over these individual behaviours.

We therefore train 10 models as described above, each with the same model architecture. Our emulator is then the simple average of the predictions of this ensemble of models. We must note however that this is a rich avenue for exploration in future work (for a review of popular ensembling methods, see Opitz & Maclin 1999). For example, it may be possible to ensemble different machine learning algorithms and combine the individual model predictions with a weighting scheme, or even another machine learning algorithm.

2.3 Sensitivity analysis

Once we have trained a deep learning emulator of GALFORM, we can apply sensitivity analysis techniques (e.g. Saltelli et al. 2010; Saltelli 2017) to understand the contribution of the different parameters to the bin-wise variance in the emulator outputs. For a full description of calculating sensitivity indices see Oleskiewicz & Baugh (2019), who first applied this type of analysis to a model of the entire galaxy population. Here, we provide a brief overview of the sensitivity indices and what they describe.

Since GALFORM is deterministic, all the variance in the output Y will be due to the effects of the input parameters X . Assuming the input parameters are independent, we can calculate the first-order variance due to parameter X_i by integrating the variance over the i -th dimension. Furthermore, we can calculate the variance due to interactions between parameters X_i and X_j by integrating the variance across the i -th and j -th dimensions, and subtracting the corresponding first-order effects of parameters X_i and X_j . This can be continued to account for the interactions between many parameters. The total variance of the model output Y can therefore be decomposed as

$$\sum_{i=1}^d \text{Var}_i + \sum_{i<j}^d \text{Var}_{i,j} + \dots + \text{Var}_{1,2,\dots,d} = \text{Var}(Y), \quad (9)$$

where Var_i represents the variance due to the i -th of the d parameters, the sum over $\text{Var}_{i,j}$ represents the variance due to interactions between the parameters X_i and X_j , and $\text{Var}(Y)$ is the total variance in the model output Y . This can be normalized to give the sensitivity indices of all orders

$$\sum_{i=1}^d S_i + \sum_{i<j}^d S_{i,j} + \dots + S_{1,2,\dots,d} = 1. \quad (10)$$

This can be separated into S_1 , the first-order sensitivity index, which describes the proportion of the variance due to the i -th parameter, and S_T , which encapsulates the proportion of variance due to the i -th parameter and all higher order interactions between the i -th parameter and all other parameters.

Given the low computational cost of our emulator, we can evaluate it at a large number of points in the parameter space following Saltelli sampling. This sampling method aims to both evenly sample the space and minimize the model discrepancy (a concept whose full explanation is beyond the scope of this work, but is described in Saltelli et al. 2010), allowing for sample-efficient calculation of the sensitivity indices. For this analysis, we use the SALIB python package (Herman & Usher 2017).

2.4 Calibration and comparison data sets

We will use our emulator to calibrate GALFORM using a number of data sets. For the most part, we adopt the data sets used for model calibration in Lacey et al. (2016), but with a focus on low-redshift observations. The key change we make is to the choice of LF data. We use the K - and r -band LFs from the GAMA survey (Driver et al. 2012); we choose these data sets as they correspond to the same survey volume and the same analysis methods are used for each band, with consistent k -corrections to $z = 0$ bands. The measured LFs should therefore be as consistent as possible, allowing our model to fit both. We apply a number of selection criteria to the GALFORM output to replicate the observational samples of the calibration data sets.

The full list of calibration and comparison data sets and their respective selection criteria are as follows:

(i) For the K -band LF, we calibrate to data from Driver et al. (2012) and also compare to data from Kochanek et al. (2001).

(ii) For the r -band LF, we calibrate to Driver et al. (2012).

(iii) For the early- and late-type sizes, we calibrate to data from Shen et al. (2003). We define early types in the model as galaxies with bulge-to-total r -band luminosities of $(B/T)_r > 0.5$ and late types as $(B/T)_r < 0.5$. Since the half-light radii of late-type galaxies are measured in circular apertures projected on the sky, the late-type galaxy sizes are corrected to face-on values by multiplying the median sizes by a factor of 1.34 (as in Lacey et al. 2016).

(iv) For the HI mass function, we calibrate to data from Zwaan et al. (2005) and compare to the estimate from Martin et al. (2010).

(v) For the early-type fraction, we calibrate to data $(B/T)_r$ derived from Moffett et al. (2016) (Moffett, private communication). Here, the $(B/T)_r$ ratio was calculated from GAMA using the disc/bulge decomposition method outlined in Lange et al. (2016). We also compare to data from González et al. (2009), which uses concentration indexes calculated from SDSS data (York et al. 2000). Again, early types are defined to have $(B/T)_r > 0.5$.

(vi) For the I -band Tully–Fisher relation, we compare to a subsample of Sb–Sd galaxies from the Mathewson, Ford & Buchhorn (1992) catalogue, as selected by De Jong & Lacey (2000). Model galaxies are selected with $(B/T)_B < 0.2$ and gas fractions $M_{\text{cold}}/M_* > 0.1$, where M_{cold} is the cold gas mass and M_* is the stellar mass.

(vii) For the bulge–BH mass relation, we compare to data from Häring & Rix (2004). To match the bias towards early types in the sample, we choose model galaxies with $(B/T)_r > 0.3$.

(viii) For the early-type metallicity, we compare to data from Smith, Lucey & Hudson (2009). We select model galaxies which reside in dark matter haloes with $M_{\text{halo}} > 10^{14} h^{-1} M_{\odot}$ and define early types as before. The observed metallicities are corrected for metallicity gradients as described in Lacey et al. (2016).

(ix) Finally, we explore the model predictions for data in a very different redshift range to our calibration data sets. We test the calibrated model predictions against observational estimates of the star formation rate density (SFRD) with redshift. We compare to data from Burgarella et al. (2013), Cucciati et al. (2012), Oesch et al. (2013), Sobral et al. (2013), and Gunawardhana et al. (2013). Since the observationally derived SFR values depend on an assumed initial mass function, and our model assumes a mildly top-heavy initial mass function in starbursts, we account for this in the observational comparison by applying an approximate correction in which we weight the starburst SFR by a factor of 1.9 (see Lacey et al. (2016) for further details).

2.5 Parameter fitting

Once we have trained our emulator, we use MCMC to explore the effect of calibration to different data sets with a simple implementation of the Metropolis–Hastings algorithm (e.g. Robert 2004). The complication here is that the observational errors on the data sets cannot be combined straightforwardly. For example, if we aimed to minimize χ^2 , and the error bar on a particular data point in the constraining observational data set was very small, this point would dominate the total error measure. Our MCMC chain would simply be trying to find the best fit to this one data point, without fitting to the others. We therefore aim to minimize the absolute error between the emulator output and the observational constraints, without considering the observational errors. This allows us to combine and fit to multiple data sets, without having to worry about the robustness of the associated observational error bars, and hence to avoid the complications described above.

We also wish to have the flexibility to give more consideration to a selected observational constraint over the others. This will allow us to investigate the effect of requiring better fits to some data sets, and to see how this affects the fit to other data sets, as well as how the optimal parameter choices change as a result. We therefore include a vector of heuristic weights, \mathbf{W} , which can be varied to increase the contribution of the residuals from one constraint to the total error,

$$\text{MAE}_{\text{obs}} = \frac{1}{n} \sum_{i=1}^n \frac{1}{n_i^{\text{obs}}} W_i |\hat{y}_i - y_i^{\text{obs}}|, \quad (11)$$

where the sum is over the n observational constraints, W_i represents the weighting of the contribution to the total error of the i -th constraint, \hat{y}_i represents the emulator prediction for a set of model parameters, and y_i^{obs} is the observational data for the i -th constraint with n_i^{obs} data points. Since \hat{y}_i and y_i^{obs} are vector quantities, the modulus signs represent the L1 norm. As the constraining data sets have a variety of values, we scale each one by a constant factor and apply a constant offset so that the range of each y_i^{obs} is $[0, 1]$. We apply the same scaling to the emulator predictions \hat{y}_i before calculating equations (7) and (11). Note that since different data sets contain different numbers of data points, we divide the i -th data set's error by the number of data points n_i^{obs} so that each contributes equivalently to the mean error. In later sections, when considering observational data, we shall refer to equation (11) as just the mean absolute error (MAE). We have checked that using the more common L2 norm instead of L1 moves attention to outliers and degrades the overall performance of the emulator.

We implement the Metropolis–Hastings algorithm as follows: we initialize each chain at a (different) random point in the parameter space, \mathbf{x} . We then draw the next sample in the chain, \mathbf{x}' , from independent Laplacian distributions, $\mathcal{L}(x'_i | \mu_i, b_i) = \frac{1}{2b_i} \exp(-|x'_i - \mu_i|/b_i)$ with $\mu_i = x_i$ and the scale parameter for the i -th model parameter, b_i , taken to be 1/20th of the parameter ranges given in Table 1. We then calculate the *acceptance ratio*, α , by taking the likelihood ratio of the emulator predictions to the observational data for the parameter sets \mathbf{x} and \mathbf{x}' under a Laplacian likelihood with scale parameter $b_{\text{obs}} = 1/20$ (i.e. the ratio $\mathcal{L}(f_*(\mathbf{x}') | \boldsymbol{\mu}, b_{\text{obs}}) / \mathcal{L}(f_*(\mathbf{x}) | \boldsymbol{\mu}, b_{\text{obs}})$, where $\boldsymbol{\mu}$ represents the values of the observational data and $f_*(\cdot)$ the emulator, and recalling we are using the modified absolute difference given in equation 11). We next generate a uniform random number $u \in [0, 1]$; samples are *accepted* if $u \leq \alpha$, in which case we draw the next sample from Laplacians centred on \mathbf{x}' , or *rejected* if $u > \alpha$, in which case we draw the next sample from Laplacians centred on the original point \mathbf{x} . Therefore, if the error between the emulator predictions for the parameter set \mathbf{x}' and the observational data is less than or equal to the error for the predictions for \mathbf{x} , we accept the sample. If the error for \mathbf{x}' is not an improvement over the previous sample, we accept it with probability α . The density of accepted samples should then trace the regions in the parameter space which give the best fits to the observational data. We discard the first 50 per cent of accepted samples to allow for burn-in. We test a number of values of the sampling Laplacian widths b_i in the range 0.05–0.2, in conjunction with the likelihood width b_{obs} , and find that these parameters have little effect on the convergence of the chains, and larger b_{obs} simply increases the proportion of accepted samples. We ran longer chains up to 100 000 samples and found that they quickly converged to their minimum MAE (as given by equation 11) within the first 10 000 samples, and so choose this as our chain length.

3 RESULTS

Here, we present our main results, starting with a demonstration of the accuracy of the emulator (Section 3.1), a sensitivity analysis of the model parameters (Section 3.2), and closing with a discussion of the calibration of the model parameters and the tensions that arise when using different combinations of data sets (Section 3.3).

3.1 Emulator performance

Having trained our emulator as described in Section 2.2, we evaluate its ability to predict the output of GALFORM at unseen points in the parameter space. We use a set of 930 GALFORM runs. The emulator was trained as described in Section 2.2 with 80 per cent of the runs used as the training set (i.e. 744 combinations of parameter values), a 93 sample validation set, and a 93 sample holdout set. For each model in the emulator ensemble (i.e. each version of the neural network), the training and validation sets were shuffled. Fig. 2 shows the emulator prediction versus the true GALFORM output for the holdout set. Generally, the emulator follows a tight relation on the $y = x$ line, indicating that the emulator is accurately predicting the GALFORM output for the parameters sets in the holdout set. The HI mass function, Tully–Fisher relation, and bulge–BH mass relations are accurately predicted, as well as the faint end of the LFs and late-type galaxy sizes. The uncertainty is greater for the predictions of the bright-end of the LFs, and for the early-type sizes, fraction of early-type galaxies with luminosity, and the early-type metallicity. The lower panel of Fig. 2 sheds some light on the source of inaccuracies in the early-type predictions, notably the early-type sizes, which exhibit noisy behaviour for some choices of parameters, and for a few cases (e.g. the purple line) the lower luminosity sizes are not well predicted. For the early-type fraction, while the error bars look large, inspection of the lower panels shows that such errors are generally in the brighter bins. We are nevertheless able to discriminate between parameter sets at the fainter magnitudes as the overall shape is well captured.

We see that the emulator is able to characterize a wide range of behaviour in the LFs, with the majority tightly predicted. In the bottom row of Fig. 2, the orange curves in the K -band panel show a substantial discrepancy between the true and predicted outputs; this usually indicates that the training data did not contain sufficient examples of this behaviour. The emulator constructs the function $f_*(\cdot)$ by fitting to the training examples, and in doing so should build a function that can interpolate between points in the parameter space. However, in sparsely sampled regions of the space, such as at the edges of our parameter bounds, the interpolation is less reliable. Indeed, if a point in the holdout set is an extrapolation with respect to the training set, performance can be affected. This is why we aim to fill the parameter space as evenly as possible using the Latin hypercube sampling method. We expect that such disagreements will decrease on increasing the number of training samples.

We can also judge from the distribution of predictions for the K - and r -band LFs in Fig. 2 that the emulator slightly overpredicts the bright end of the LF. This is a consequence of the emulator training; in the interest of computing speed, we run GALFORM on only a subregion accounting for 1 per cent of the full volume of the P-MILL simulation. This leads to sampling effects at low galaxy number densities, and for different choices of parameters the LF is cut-off at different luminosities. Since the output of our emulator must be fixed-length, during training we mask any points beyond this luminosity cut-off when computing the loss. This means that in the brighter luminosity bins the emulator is only fitting to a small

number of runs which are biased towards having higher values of ϕ in these brighter bins. There is therefore a tendency to overpredict at these luminosities. This should only be a minor problem in terms of our fitting routine, since the Driver et al. (2012) data we are fitting to does not sample ϕ to very low number densities. We also see a quantization effect in the brighter LF bins, again due to the discrete sampling of galaxies. These problems could be removed by evaluating GALFORM on a larger fraction of the P-Mill simulation volume, though this would be more computationally expensive.

3.1.1 Scaling with training set size

We train three emulators with 250, 500, and 750 samples of parameters respectively (split into training and validation sets with 10 per cent of the samples being used for validation) to investigate the scaling of the emulator performance with the number of full GALFORM calculations carried out. The emulators consist of an ensemble of 10 networks each trained on the same (shuffled) training and validation data and the same holdout set of 93 samples. Fig. 3 shows the scaling of the emulator performance on the holdout set (as measured by the MAE) with the number of training samples N . The dashed line shows average performance of the individual networks, and the solid line shows the performance of the ensemble. The model scales well with increasing training samples, and ensembling affords an almost constant improvement in performance (we find that at ~ 10 models, the performance increase from adding more models to the ensemble saturates).

We test the ability of the emulator to generalize to unseen data by evaluating the version of the emulator trained with 500 samples in Fig. 3 on the remaining 430 unseen samples. We find very little variation in the accuracy of the model between the two holdout sets. The MAE on the 93 sample holdout set was 0.034, and on the full 430 available holdout samples was 0.032. Further, we perform a 10-fold cross-validation with the training, validation, and holdout sets as described in Section 3.1. We find a mean MAE of 0.030, with a range between 0.027 and 0.034. This gives us confidence that the model is able to learn a function which provides a very good approximation to GALFORM across the full parameter space.

The impressive scaling of the emulator error with number of training samples is encouraging. SAMs are used to build mock catalogues for upcoming surveys, and some of these have stringent requirements on fits to certain data sets, such as the redshift distribution of galaxies. We can envisage using this technique to produce high-accuracy parameter estimates for fits to such data sets by increasing the number of training samples, or using ‘zoom-in’ training samples as in previous work (e.g. Bower et al. 2010) to focus in on a particular region of parameter space which is deemed to give acceptable fits to the constraining data sets. Nevertheless, we find that our current emulator is sufficiently accurate to facilitate calibration and model exploration.

3.2 Sensitivity analysis

We apply the techniques described in Section 2.3 to calculate the contribution of each parameter to the variance in each bin of the nine constraints. The results are shown in Fig. 4. The open circles indicate the first-order sensitivity index, S_1 , which quantifies the proportion of the variance due to just one parameter. The total order sensitivity, S_T , is shown as solid lines, and indicates the proportion of the variance contributed by one parameter and its interactions with the other parameters. We can interpret the difference between the

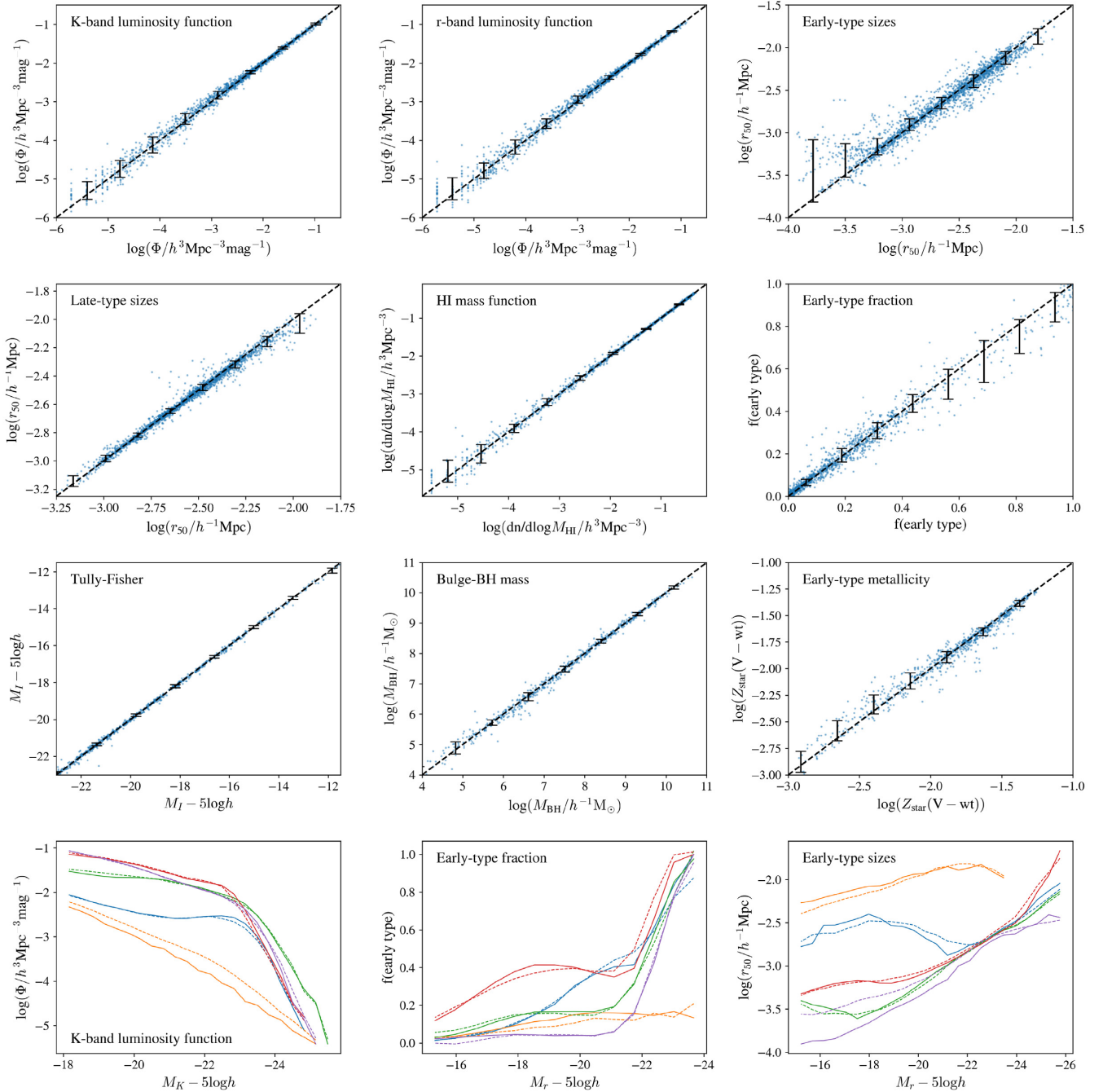


Figure 2. Emulator performance across nine statistics computed from the model output for out-of-sample parameter sets. These statistics are either number densities or median values in luminosity or mass bins, and are the same ones used for the observational comparisons. The first three rows show the emulator output (y-axis) versus the GALFORM output (x-axis). Black error bars indicate the 10–90th percentile range of the residuals. The bottom row shows a random draw of emulator outputs (*dotted*) and true GALFORM outputs (*solid*) for the *K*-band LF, early-type fraction, and early-type sizes, reading from left to right. In these panels, different colours denote different parameter sets.

first order and total order sensitivity as a measure of the strength of the interaction between a given parameter and the other parameters. For clarity, we exclude parameters which never contribute more than 10 per cent of the variance to any bin. Both f_{burst} and f_{ellip} meet this condition, and so do not appear in the plots.

We see that the dominant parameters for the majority of the model outputs are, perhaps unsurprisingly, the SN feedback parameters. $V_{\text{SN,disk}}$ and γ_{SN} account for the majority of the variance at the faint end of the *K*- and *r*-band LFs. At the bright end, α_{cool} , the

parameter governing the strength of AGN feedback, contributes the largest proportion of the variance. The majority of the variance in the late- and early-type sizes, the Tully-Fisher relation, as well as the HI mass function is also contributed largely by the same two or three parameters.

The early-type fraction is dominated by the threshold for disc instability, f_{stab} , up until $M_r - 5\log h \approx -21$. At brighter magnitudes, disc instabilities become unimportant as mergers takes over as the main channel for building spheroidal components (see Husko et al., in

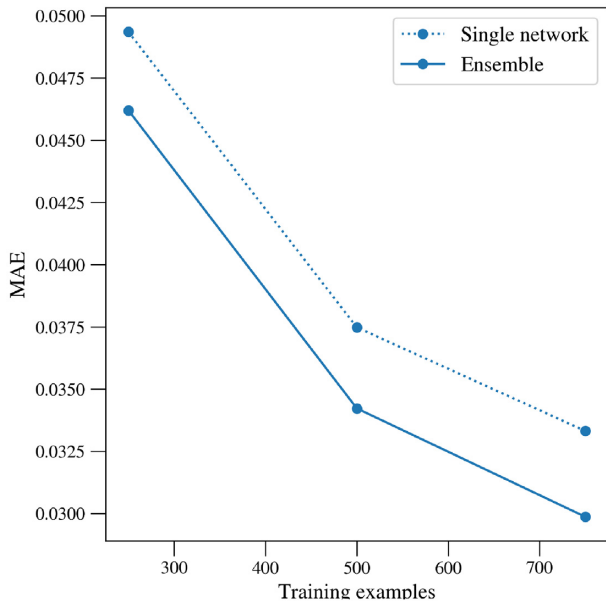


Figure 3. Emulator mean absolute error with the number of training examples of full GALFORM runs for the ensemble (solid line) and single (dotted line) networks. The emulators were trained on 250, 500, and 750 samples and performance evaluated on the same holdout set of 93 samples. Recall that the emulator outputs are scaled as described in Section 2.5.

preparation, for an exploration of the relative importance of different channels for the growth of galaxy stellar mass).

The sensitivity analysis hence dispels one of the myths surrounding SAMs as it shows that the model cannot be made to fit to any arbitrary combination of data sets. To match the faint end of the K -band LF, we are strongly constrained in our choice of SN feedback parameters, which contribute the vast majority of the variance to these bins. Our predictions of early- and late-type galaxy sizes, the HI mass function, the Tully–Fisher relation, and the bright end early-type fraction are also then largely constrained, since the SN feedback parameters dominate these outputs too. This is in line with the analysis performed by Bower et al. (2010), which reached similar conclusions.

The parameters also have clear physical interpretations, and are analogous to the parameters used in the subgrid physics models in hydrodynamic simulations (e.g. Crain et al. 2015; Weinberger et al. 2016; Pillepich et al. 2017). The parametric model for SN feedback can indeed be tuned to give a good match to the late-type galaxy sizes, but in doing so we are strongly constraining our fits to other data sets; the model does not include arbitrary parameters that allow for fine-tuning to an individual data set without physical motivation or consequences for the fits to other data sets.

3.3 Calibration and data set tensions

We now apply the methods described in Section 2.5 to calibrate the model to the data sets described in Section 2.4, focusing on uncovering any tensions that exist between data sets. First, we aim to replicate a known tension in the model discussed in Bower et al. (2010) and Lacey et al. (2016). This is the tension between reproducing late-type galaxy sizes and the galaxy LFs; these data sets have been found to prefer different values for the SN feedback parameters. We can investigate this by adjusting the weightings applied to the residuals between our emulator prediction and each data set (as in

equation 11), and then performing an MCMC parameter search to see how the best-fitting parameter choices respond.

In Fig. 5, we show the emulator predictions for three sets of best-fitting parameters. In the first case, shown by the blue line, we weight only the residuals for the K -band LF. For the orange line, we weight only the size–luminosity relation for late-type galaxies, and the green line shows the results when weighting both data sets equally (i.e. both data sets have equal influence over the best-fitting parameter values). The shaded region is shown only around the fit to the K -band LF for clarity, and represents the 10–90th percentile error of the emulator when predicting similar values in the holdout set (this gives a rough idea of the uncertainty of the emulator, but is certainly not an exact measure). We can clearly see the tension between these two data sets uncovered in an automatic and objective way; matching the sizes of faint late-type galaxies leads to an overprediction of the LF at all luminosities by up to an order of magnitude. When matching both the K -band LF and the late-type galaxy sizes, we see an overprediction in the faint-end of the LFs, and the sizes of faint late-types are overpredicted by a factor of ~ 2 . The early-type sizes and Tully–Fisher relation are also shown in Fig. 5. Although no weighting was applied to these data sets in this exercise, we can see improved matches emerge naturally when we fit to the late-type galaxy sizes. We can gain some intuition for this behaviour from Fig. 4. As discussed, the Tully–Fisher relation, early- and late-type galaxies sizes, and the faint-end of the galaxy LF are highly sensitive to the choice of SN feedback parameters, γ_{SN} and $V_{\text{SN,disc}}$ (which together account for ~ 90 per cent of the variance in the faint-end LFs and the sizes of faint late-type galaxies). Therefore, we might expect that some tension would arise in trying to fit to a number of the above data sets at the same time.

It is also informative to investigate how the acceptable regions of parameter space change as we introduce weightings to other data sets. We demonstrate this for the tension between the LF/late-type sizes in Fig. 6. The shaded regions represent accepted samples from our 20 MCMC chains, each 10 000 steps in length, with the first 50 per cent of each chain discarded to allow for burn-in. The red region corresponds to a fit to the K -band LF, and the blue region to fits to both the K -band LF and late-type galaxy sizes. The shading gives a sense of the density of accepted samples, i.e. the darker colours correspond to the more favoured parts of parameter space in this projection. The darkest regions correspond to the 25th percentile, and the lighter regions to the 50th and 75th percentiles. Also shown in Fig. 6 are 1D histograms of the density of accepted samples. We find that, as in previous analyses, a reasonably large range of parameter values result in acceptable fits to a given constraint. This can be best understood (as explained in Bower et al. 2010) as the effect of the high dimensionality of the parameter space; though when plotted in projection down to one or two dimensions the space appears widely sampled, the higher dimensional acceptable region is reduced significantly. Also, some of the parameters produce degenerate effects (see for example Fig. A1 in Appendix A, where we show the degenerate effects of the f_{stab} and $V_{\text{SN,burst}}$ parameters). Nevertheless, we see that the K -band LF fit prefers somewhat higher values of $\gamma_{\text{SN}} \approx 3.6$ and lower values of $V_{\text{SN,disc}} \approx 200 \text{ km s}^{-1}$, in contrast to the fit to both the K -band LF and late-type sizes, where we find a preferred value of $\gamma_{\text{SN}} \approx 2.3$ and $V_{\text{SN,disc}}$ at the top of the explored range at $\sim 550 \text{ km s}^{-1}$. Interestingly, there seems also to be a preference for lower values of ν_{SF} to match the late-type galaxy sizes. We can understand this crudely by investigating the first-order effect associated with the ν_{SF} parameter. Inspecting Fig. A2 in Appendix A, we see that the ν_{SF} parameter has a some effect on the bright-end of the K -band LF. This counteracts the enhancement from the higher

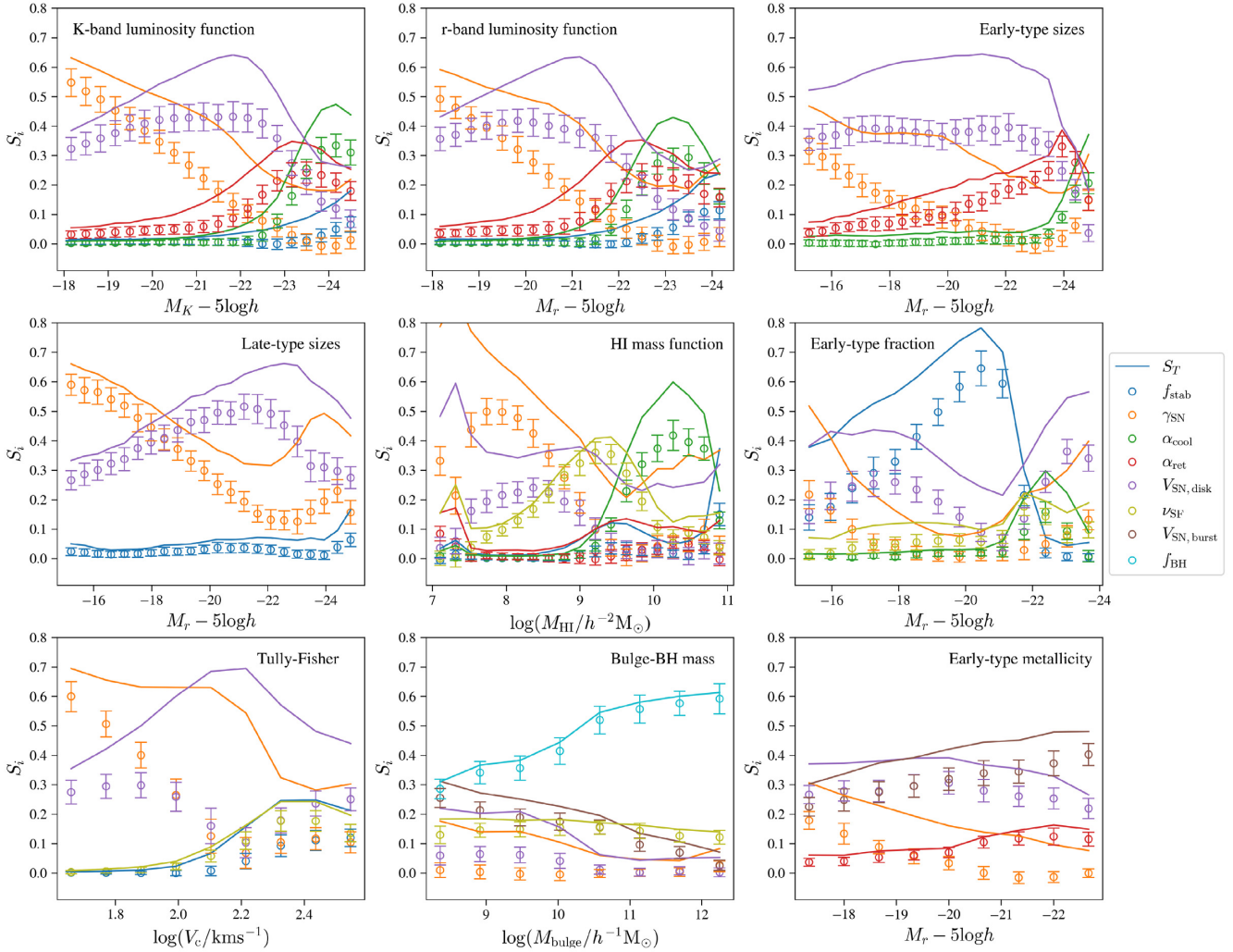


Figure 4. The emulator sensitivity to different parameters for each of the observables considered in this work; each panel shows a different observable, as labelled. Open circles indicate S_1 as described in the text, and solid lines represent S_T . For clarity, error estimates are shown for the S_1 calculation but not for S_T , although they are similar. Sensitivities for parameters that never exceed more than 10 per cent of the variance in any bin are not plotted.

value of $V_{\text{SN,disk}}$, and also marginally improves the fit to the late-type galaxy sizes.

Another tension arises between the HI mass function and the bright end of the K - and r -band LFs. This is shown in Fig. 7. As before, the blue line corresponds to the fit to the K -band LF alone, the orange line to the fit using the HI mass function alone, and the green to a fit to both data sets. We can again propose (from our plot of the sensitivity indices, Fig. 4) that the main cause of this discrepancy is a tension in the choices for the AGN feedback parameter, α_{cool} , and the SN feedback parameters. Indeed, when fitting the observational constraints individually, the fit to the K -band LF prefers a higher value for the AGN feedback parameter, with $\alpha_{\text{cool}} \approx 0.8$, whereas the fit to the HI mass function prefers $\alpha_{\text{cool}} \approx 0.5$. We can also investigate how calibrating to both data sets shifts the parameter values. We do this as before with an MCMC exploration of the parameter space (see Fig. A3 in Appendix A). Fitting to both the K -band LF and the HI mass function (as compared with a fit just to the K -band LF) causes a shift in the preferred $V_{\text{SN,disk}}$ to higher values. ν_{SF} , the parameter that controls the rate of quiescent star formation, shifts to the lowest values in the explored range, and the parameter α_{ret} , which is involved in gas return to haloes following SN feedback, becomes

more strongly peaked, with the peak shifted to slightly higher values.

To understand this further, we investigate the first-order effects of the parameters ($V_{\text{SN,disk}}$, ν_{SF} , and α_{ret}), perturbed around the fit to the K -band LF. We show the results in Fig. 8. We vary the parameters individually (‘one-at-a-time’) across their explored range, with lighter colours corresponding to lower parameter values. We can begin to understand the changes in the preferred parameter choices in terms of these transformations. When fitting both the HI mass function and the K -band LF, we find that there is a slight overprediction of the bright-end of the LF. From these one-at-a-time plots, we can see that the increase in $V_{\text{SN,disk}}$ causes an overprediction at the bright-end of the LF, and a reduction in amplitude at the faint-end, but more accurately matches the high-mass end of the HI mass function. The HI mass function can be better matched at intermediate masses by a decrease in ν_{SF} . In GALFORM, reducing ν_{SF} has the effect of decreasing the rate of quiescent star formation in discs. As a result, lower values of this parameter provide a better fit to intermediate masses of the HI mass function, while simultaneously reducing the number density of the most luminous galaxies in the K -band LF, and so counteracting the enhancement due to the increase in the $V_{\text{SN,disk}}$

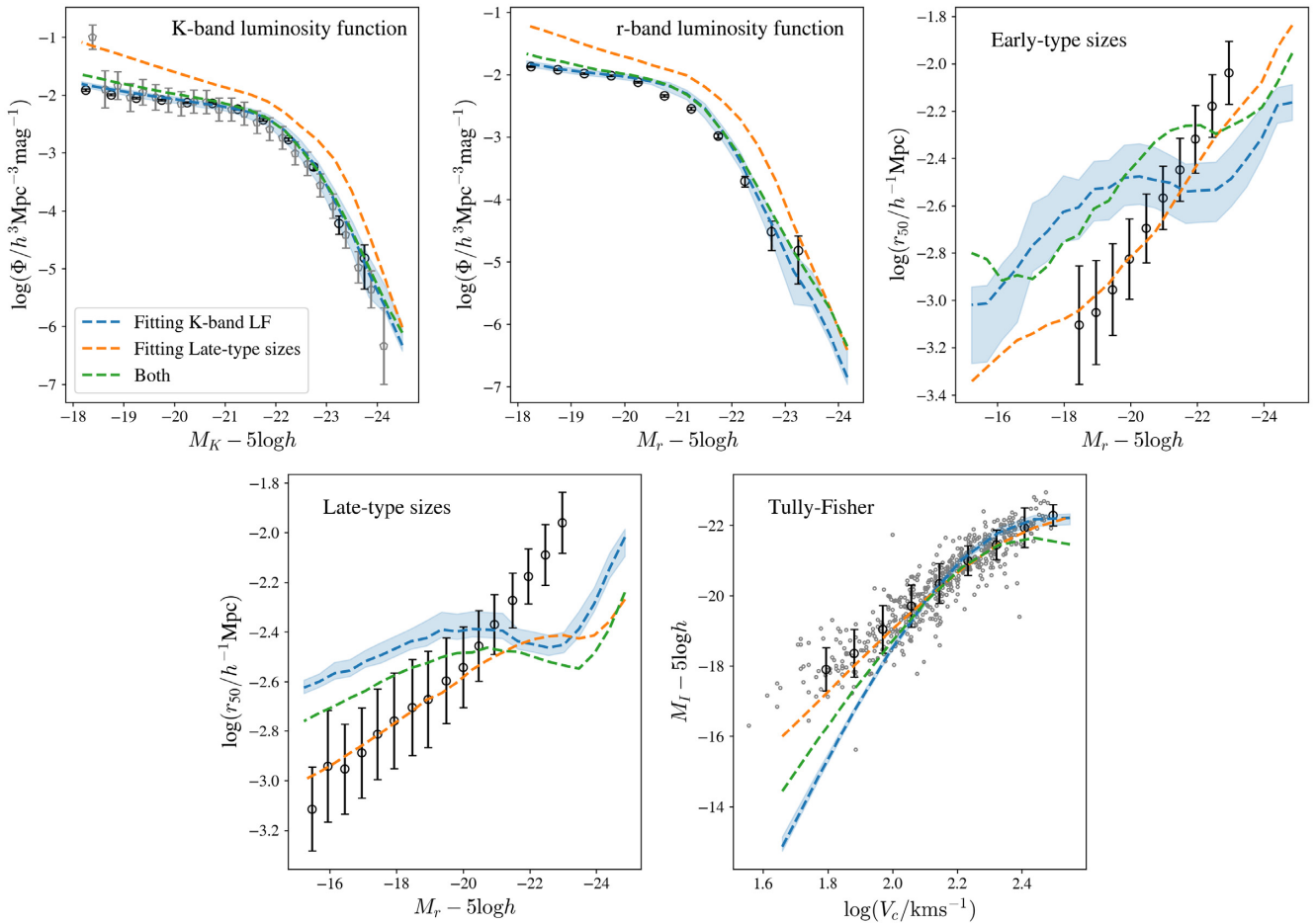


Figure 5. A comparison of the emulator predictions for fits to the K -band LFs, the late sizes, and a combination of the two (represented by different colour dashed lines). We fit to the data from Driver et al. (2012) (black) for the K -band LF, and Shen et al. (2003) for the late-type sizes. The emulator predictions correspond to the best fit found from 20 MCMC chains, each 10 000 steps in length. The blue shading represents the 10–90th percentile errors when predicting a similar value in the holdout set. The black and grey data points represent the calibration data described in Section 2.4. For the K -band LF, we also compare to data from Kochanek et al. (2001) (grey). For the r -band LF, we compare to data from Driver et al. (2012). For the early-type sizes, we compare to data from Shen et al. (2003), and for the Tully–Fisher relation we compare to data from De Jong & Lacey (2000).

parameter. We can further improve the match of the prediction for the LF to the observational data by increasing α_{ret} , which has little impact on the H I mass function but reverses some of the ‘flattening’ of the LF caused by the increase in $V_{\text{SN,disc}}$. In previous galaxy formation models, using the *WMAP-7* cosmological parameters, this tension has not been so apparent, but can also be seen between the b_J -band LF and the H I mass function in Baugh et al. (2019).

Our approach also allows us to uncover a significant tension between the bright end of the LFs, the early-type fraction, the H I mass function, and the early-type metallicity. We demonstrate this in Fig. 9, where we compare a fit found by calibrating to the K -band LF, H I mass function, and the early-type fraction with and without including the early-type metallicity constraint (note that we do not fit to data sets shown in grey). Including the early-type metallicity has a significant effect on the best-fitting parameter values; it generally improves the fits to the galaxy sizes, and degrades the fit to the early-type fraction [at least when considering the Moffett et al. (2016) data] and the H I mass function. We investigate the impact on the acceptable region of parameter space in Fig. 10, where we show the key changes induced by including the early-type metallicity constraint. The red region shows the fit to the K -band LF, H I mass

function, and early-type fraction, and the blue region also includes the early-type metallicity. We find that there is a reconfiguration of the SN feedback parameters, γ_{SN} , and $V_{\text{SN,burst}}$ to match the early-type metallicity. This reconfiguration provides better fits to the galaxy sizes, while degrading the fit to the H I mass function, which is also very sensitive to the choice of γ_{SN} . The fits found when we choose not to include the early-type metallicity constraint are very similar to those found in Lacey et al. (2016) and Baugh et al. (2019), with overpredictions for the sizes of faint early-type galaxies, good fits to the H I mass function, and an underprediction of the metallicity of faint early-type galaxies. Including the early-type metallicity constraint, however, moves us to a different region of parameter space for this updated version of the GALFORM code.

Another key shift is in the preferred value of f_{stab} ; the preference for lower values of f_{stab} leads to a suppression of the early-type fraction at intermediate luminosities. At these luminosities, disc instabilities are the main channel for building up spheroid components and decreasing f_{stab} limits the number of disc instabilities (see Husko et al., in preparation). Although f_{stab} does not appear in the early-type metallicity sensitivity analysis (as shown in Fig. 4), this is because the sensitivity indices are dominated by the strong effects of the

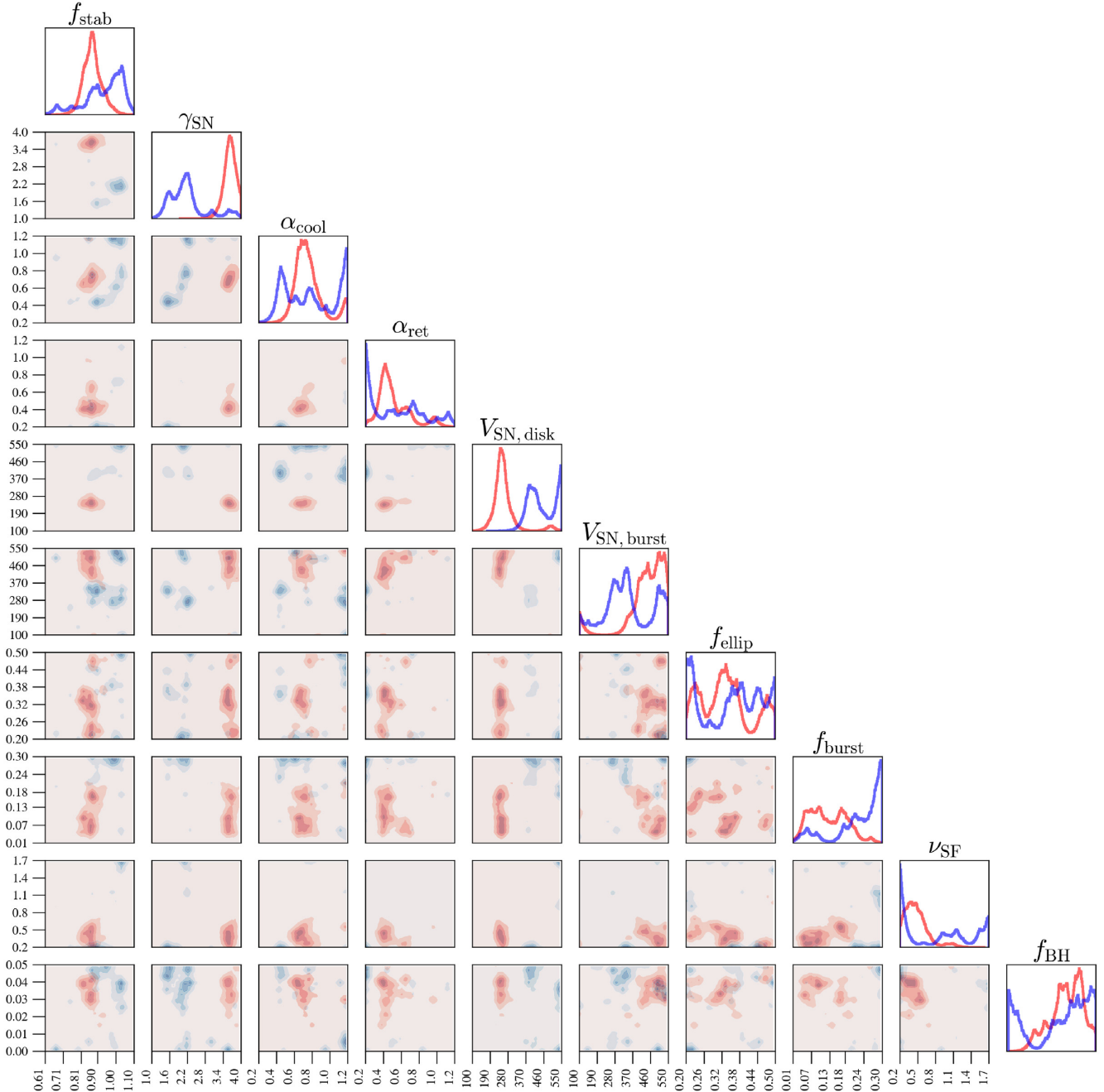


Figure 6. Accepted samples from 20 MCMC chains for fits to the K -band LF (red), and both the K -band LF and the late-type galaxy sizes (blue). The first 50 per cent of samples were discarded to allow for burn-in. The histograms show the marginalized distribution of the parameters. The ranges on each axis are the same as those quoted in Table 1. The shading gives a sense of the density, 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.

SN feedback parameters. A lower f_{stab} does increase the early-type metallicity but to a far lesser extent than the SN feedback parameters, and so gives a more exact match to the observational data.

In our analysis so far, we are perhaps making the mistake of attempting to understand a non-linear model in terms of just first order, one-at-a-time changes to the parameters. Indeed, this is one of the key weaknesses of traditional ‘chi-by-eye’ parameter fitting. However, as shown in Fig. 4, we can justify this mode of investigation; the majority of the variance due to a given parameter is

generally due to just its first-order effect. ν_{SF} , α_{ret} , α_{cool} , and f_{SMBH} only have weak higher order variance contributions. In the cases where this assumption is less valid, for example in the case of the parameter γ_{SN} and $V_{\text{SN,disk}}$, this can be understood straightforwardly with reference to equation (2); these parameters directly interact in the implementation of SN feedback. It is striking how much of the variance is due to the parameters’ first-order effects. The outlier is f_{stab} , which has strong higher order interactions and is not directly coupled to the other parameters in any equation.

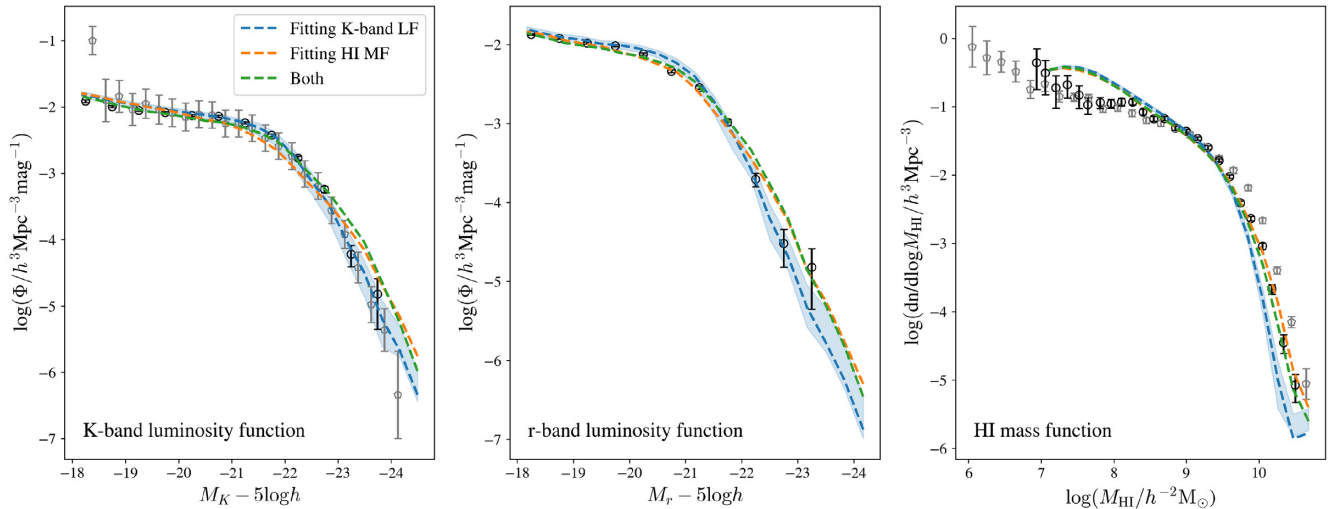


Figure 7. A comparison of the emulator predictions for fits to the K -band LFs, the HI mass function, and a combination of the two (represented by the different colour dashed lines). The black and grey data points represent the calibration data described in Section 2.4. For the HI mass function, we fit to data from Zwaan et al. (2005) (black) and include data from Martin et al. (2010) (grey) for comparison.

3.3.1 Best-fitting model

We can now re-calibrate the GALFORM model across all constraints to produce an estimate of the best-fitting parameters. As we have seen, there is no single choice of parameters that can reproduce all of the constraints, and we have to decide during the calibration which data sets we would like to give more or less weighting. The ideal of automatically calibrating a semi-analytical model is therefore a difficult one to realize; we will always have to make trade-offs in how we fit to the various data sets. As described in Section 2.4, we can do this in a semi-automatic way using the heuristic weighting scheme.

We have seen that there are a number of trade-offs or tensions to consider when aiming to find a best-fitting model. Fitting to the late-type galaxy sizes, the Tully–Fisher relation, or the HI mass function generally degrades the fit to the K - and r -band LFs. We have also seen that trying to reproduce the early-type metallicities worsens the fit to the Moffett et al. (2016) data for the early-type fraction, and worsens the fit to the high-mass end of the HI mass function. On the other hand, other observational constraints are more easily fitted; since the bulge–BH mass relation is largely dependent solely on the f_{SMBH} parameter, and this has very little influence on other observables, fitting this constraint is trivial.

With these considerations in mind, we choose heuristic weights such that the r - and K -band LFs are strongly weighted. We know from our previous analysis that there will be trade-offs between both the bright- and faint-ends of the LFs, but we require good fits to both. Therefore, we doubly weight both of these constraints when calculating the MAE given in equation (11) (i.e. by setting $W_i = 2$ for each observable). Since the late-type sizes, early-type sizes, and the Tully–Fisher relation are important constraints, but lead to compromised LF fits, we apply single weighting to all these constraints (i.e. $W_i = 1$). We also give a single weighting to the early-type metallicity since this trades-off against the bright end of the LF and the high mass end of the HI mass function. Since the HI mass function is an important constraint, but as we are aware that it generally degrades the fit to the bright end of the LF, we give this constraint a triple weighting. This is to ensure that more total weight is applied to the K - and r -band LFs in combination. We apply a single

weighting to the early-type fraction; we have seen that this fit is in strong tension with the early-type metallicities and sizes.

We run 100 MCMC chains with our emulator, each 10 000 steps in length. We find that the minimum MAEs (as computed using the emulator) obtained with each chain lie in the range ~ 0.15 – 0.20 ; since this range is similar to the out-of-sample accuracy of the emulator, and so in principle we cannot discern which parameter sets give the best fit to the observational data with the emulator alone, we evaluate these 100 minimum MAE parameter sets with the GALFORM code.

The best fits are shown in Fig. 11. Here, we plot the best 50 sets of parameters from the 100 MCMC chains, as evaluated with the GALFORM code. These runs have very similar MAEs, covering the range 0.16–0.18, while the runs not shown cover the range 0.18–0.22, which is slightly wider than the range predicted by the emulator, but within the expected emulator error (0.04 in this weighting scheme). The solid red line indicates the run with the lowest MAE, and the blue lines show the remaining 49 runs. The shading on these lines indicates the size of the residuals between the model and the HI mass function, with darker lines indicating smaller residuals, and demonstrates that the parameter choices that provide the best fits to the HI mass function overpredict the bright-end of the LFs. The black dashed line shows the statistical galaxy properties of the model presented in Baugh et al. (2019) (hereafter Baugh19). In Table 2, we show the set of parameters with the lowest MAE to the observational data (corresponding to the red line in Fig. 11), the parameter range of the best 50 parameter sets, and compare with the parameters adopted in Baugh19 for an older version of the model. We reiterate, however, that the best-fitting parameters are just one realization out of many possible choices due to the degeneracies between the parameters, and the effect of calibrating to multiple data sets. Also, the ranges shown in Table 2 do not indicate that any choice of parameters within these ranges will yield an equivalent fit; the value of one parameter will constrain the choices for the other parameters, hence the reason for giving an example of a best-fitting set of parameters. We find that some parameters, such as ν_{SF} and γ_{SN} are constrained to a tight range of values, whereas others, such as f_{stab} can be drawn from a large fraction of the explored range.

Calculating the mean absolute error of the best-fitting model, and the Baugh19 model, using the same procedure as described

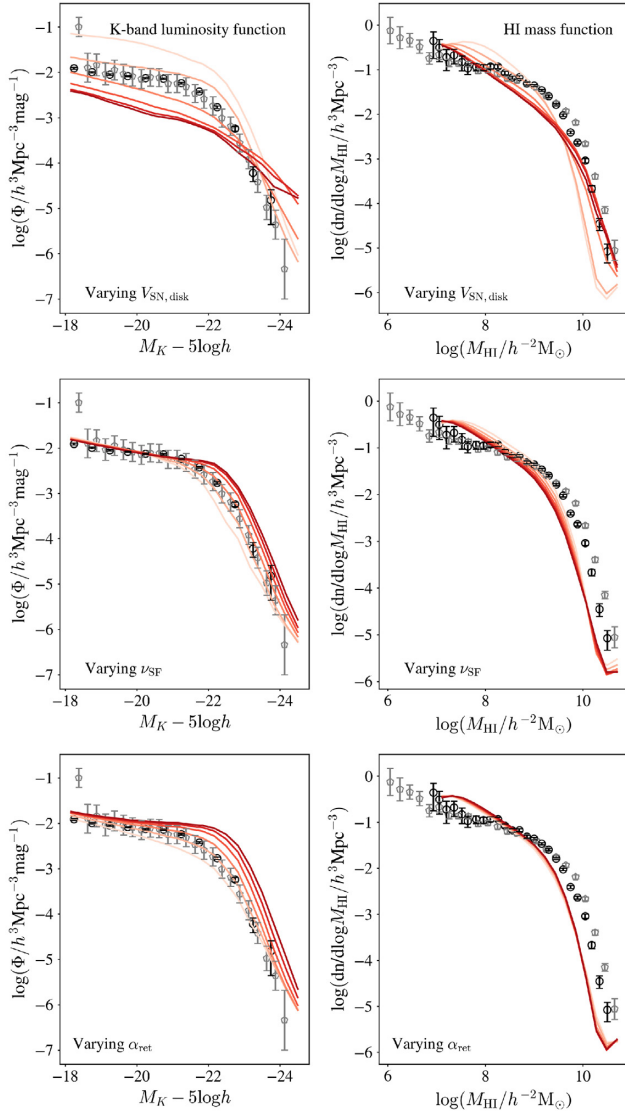


Figure 8. Emulator predictions for perturbing three key parameters around a fit to the K -band LF. The top row shows the result of varying the parameter $V_{\text{SN,disk}}$ between 100 and 550 km s^{-1} , the middle row varies ν_{SF} between 0.2 and 1.7 Gyr^{-1} , and the bottom row varies α_{ret} between 0.2 and 1.2. Darker colours correspond to higher values of the varied parameter.

in Section 2.5 (and recalling that we scale each output so that the data lie in the range $[0, 1]$), we find that at least under this metric the new model is a better fit to the data. Over all the data sets, the new best fit found in this work gives an MAE of 0.16 versus an MAE of 0.20 for the [Baugh19](#) model. We note that the MAE for the model used in [Baugh19](#) is within the range of the minimum MAE reached by the 100 MCMC chains. The reduced MAE of the new best-fitting model compared to the [Baugh19](#) model is mainly due to large improvements in the fits to the early-type galaxy sizes and their metallicities, while the fits of the new model to the early-type fraction and Tully–Fisher relation are slightly worse.

As shown in Fig. 11, we find that our model provides a slightly better fit to the K - and r -band LFs than the [Baugh19](#) model.¹ For

¹Baugh et al. (2019) concentrated on reproducing the b_j -band LF, and the HI mass function, and did not consider the r -band LF.

the updated model presented in this work, we find an MAE of 0.05 versus 0.08 for the [Baugh19](#) model in the K band and 0.04 versus 0.06 in the r band. The galaxy sizes are an improvement over previous iterations of the GALFORM model, particularly the early types, which are now more qualitatively similar to the observational data in that they are monotonically increasing with luminosity (at least in the range of the data), whereas the [Baugh19](#) model features a marked dip at intermediate magnitudes and significant overprediction at fainter magnitudes, differing from the observed sizes by a factor of ~ 3 . The MAEs in this case are also significantly lower for the new model: for the late-type galaxies, we find an MAE of 0.14 in this work versus 0.21 for the [Baugh19](#) model, and 0.09 versus 0.39 for the early-type sizes. This difference is largely due to the different choices for the γ_{SN} parameter. Here, we find a preference for much lower values of γ_{SN} , in the range 2.05–2.72, versus 3.40 for the [Baugh19](#) model. Reducing this parameter significantly weakens the effect of SN feedback in low-mass galaxies, leading to smaller sizes (see fig. C10 of Lacey et al. 2016). Interestingly, the preferred γ_{SN} we recover is much closer to the value expected from energy conservation arguments, $\gamma_{\text{SN}} = 2$ (Larson 1974; Lagos, Lacey & Baugh 2013).

The fit to the HI mass function is slightly worse than the fit found in the [Baugh19](#) version of the model (with MAEs of 0.09 versus 0.08); a better fit would come at the expense of a more severe overprediction of the bright-end of the LF as previously discussed, and as shown by the shading of the blue lines in Fig. 11. As we have seen in Fig. 9, we are able to produce better matches to the HI mass function and the LFs if we exclude the early-type metallicity and galaxy sizes constraints (the fits found in this case are much more similar to the [Baugh19](#) model, with similarly high γ_{SN} in the range ~ 3.2 – 3.8 , as shown in Fig. 10). Our fit to the early-type metallicities is an improvement over the prediction of the [Baugh19](#) version of the model, where the MAE of our model is 0.15 versus 0.55 for the [Baugh19](#) model. However, our early-type metallicities fit comes at the cost of slightly degrading the fit to the early-type fraction (0.13 versus 0.10). Our fit to the Tully–Fisher relation is worse than in the [Baugh19](#) model, with an MAE of 0.28 versus 0.17, though we have demonstrated that we can retrieve a fit more similar to [Baugh19](#) by giving less weight to the early-type metallicity constraint (again as shown in Fig. 9).

3.3.2 Predictions for cosmic star formation history

We have calibrated GALFORM to low-redshift constraints and now investigate the predictions for the evolution of the SFRD with redshift. To do this, we evaluate the SFRD with redshift for the sets of parameters corresponding to the GALFORM runs shown in Fig. 11. Fig. 12 shows the SFRD predictions for these parameter choices. Since GALFORM assumes a mildly top-heavy initial mass function (IMF) for stars formed in starbursts, we apply an approximate correction to give the SFR that would be inferred assuming a Kennicutt IMF (Kennicutt 1983) by weighting the starburst SFR by a factor of 1.9 (as in Lacey et al. 2016). The curves therefore represent an apparent SFRD that can be compared with observational estimates that assume a solar neighbourhood IMF. Interestingly, we see that the spread of the model predictions only increases slightly as we move out to larger redshifts. This suggests that the low-redshift calibration data sets actually constrain the redshift evolution of the model reasonably well.

4 DISCUSSION

We have presented a method for efficiently calibrating and exploring a SAM of galaxy formation across a wide range of outputs. In doing

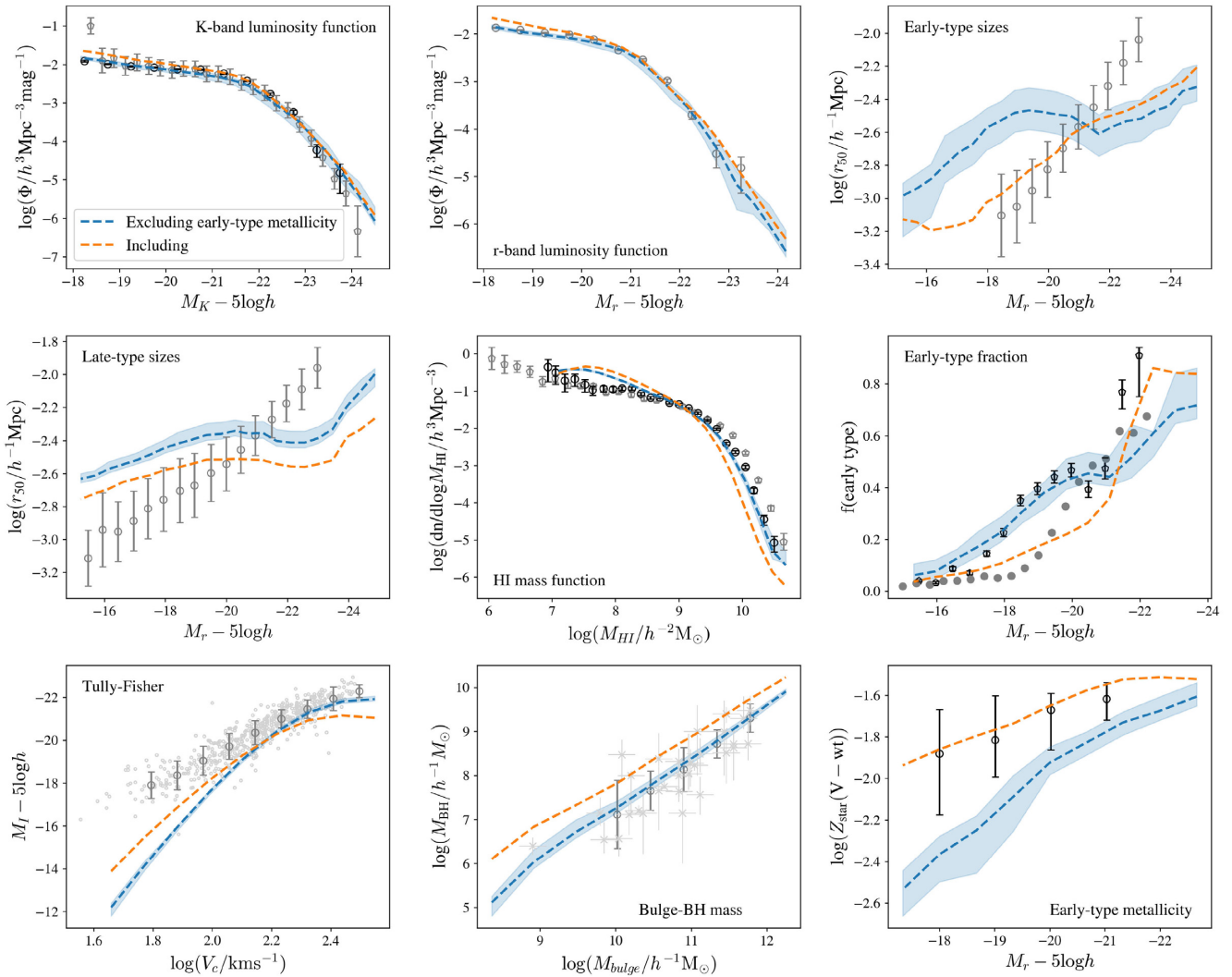


Figure 9. A comparison of the emulator predictions for fits to the *K*-band LF, the HI mass function, and the early-type fraction with and without including the early-type metallicity constraint (represented by different colour dashed lines, as labelled in the top left panel). The emulator predictions correspond to the best fit found from 20 MCMC chains, each 10 000 steps in length. In both cases, all included constraints were equally weighted. The data described in Section 2.4 are shown in black and grey. For the bulge–BH mass relation, we compare to data from Häring & Rix (2004), for the early-type fraction we fit to data from Moffett et al. (2016) and compare to data from González et al. (2009), and for the early-type metallicity we compare to data from Smith et al. (2009). Black data points indicate that the data was used for fitting, grey data points are included for comparison.

so, we have uncovered a number of tensions between data sets: for example, in Fig. 9, we found that on relaxing the requirement for a good fit to the early-type metallicities, we recovered a fit very similar to those found in Baugh et al. (2019) and Lacey et al. (2016). By increasing the weight given to the early-type metallicity constraint, we moved to a new region of parameter space, changing our fit to the early-type fraction and early-type sizes. Tensions such as this point to either deficiencies in the model or a discrepancy between the observational data sets. For example, again in Fig. 9, we see that the early-type fraction fit to the Moffett et al. (2016) data (shown in black) degrades when we include the early-type metallicity constraint. However, in this case, the fit is then in better agreement with the González et al. (2009) data (shown in grey). Similarly, for the HI mass function, the Zwaan et al. (2005) and Martin et al. (2010) data sets do not agree with one another, differing by up to a factor of 5 in abundance at high masses.

In other cases, we can see a clearer deficiency in the GALFORM predictions. For example, in Fig. 5, we show the effect of fitting to the *K*-band LF or the late-type galaxy sizes, or both together. We see that even when we fit only to the late-type size constraint, we are still not able to recover the observed monotonic increase in radius with increasing luminosity. Clearly, this suggests that the treatment of the galaxy disc sizes in GALFORM needs to be improved.

The emulation method presented here contrasts with previous work; most emulators have focused on reducing the parameter space by using more approximate emulators, but with robust uncertainty measures, to iteratively reduce the volume of parameter space that could plausibly produce good fits to the data. Van Der Velden et al. (2021), for example, used a total of 3000 runs over three waves to calibrate the MERAXES SAM to the stellar mass function. We have focused instead on maximizing the accuracy of our emulator of GALFORM across the whole parameter space. Our aim is to build an emulator that allows us to explore a wide range of calibration

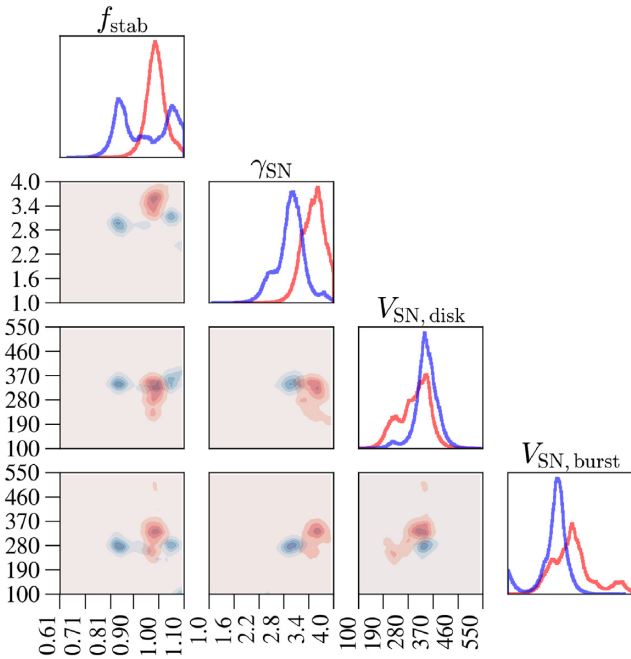


Figure 10. Accepted samples from 20 MCMC chains for fits to the K band, the H I mass function, and the early-type fraction with (blue) and without (red) including the early-type metallicity constraint for a few key parameters. The shading gives a sense of the density of the samples, and the histograms show the distribution of each parameter in 1D projection. The darkest regions correspond to the 25th percentile, and the lighter regions to the 50th and 75th percentiles.

data sets, and different combinations of these data sets. As shown in Fig. 2, our emulator performs well: most of the key constraints are tightly predicted.

In this vein, we have discounted the observational error bars to facilitate model exploration. In Section 3.3.1, we calibrated our model to the full set of observational data sets under consideration. However, since we did not include observational errors and used an absolute error metric, it is difficult to give meaningful error bars around our estimates of the best-fitting parameters quoted in Table 2. As previously mentioned, SAM calibration involves making trade-offs between certain observational constraints; often the best-fitting model is calibrated in a way that is poorly defined. We have attempted to reproduce and elucidate this process in an automatic way through a heuristic weighting scheme. We aim to investigate a more robust calibration analysis in the future with an improved treatment of the observational errors.

Similarly, our approach could be extended to include a more robust measure of the emulator’s uncertainty in reproducing GALFORM outputs. When emulating a set of model outputs, we should ideally account for *epistemic* and *aleatoric* uncertainty. *Epistemic* uncertainty refers to the uncertainty associated with the emulator’s parameters (in this case, the weights of the neural network), and *aleatoric* uncertainty refers to uncertainty inherent in the data generating process (e.g. the sampling noise on the GALFORM outputs). Our approach does not currently model the epistemic uncertainty on the emulator’s weights, but instead acts to reduce it by averaging over a number of individual estimates provided by the neural networks in our ensemble. It is possible therefore that we are discarding regions of the parameter space that potentially contain reasonable fits to the data. However, we are somewhat protected from this scenario in that the

regions that are most difficult for our emulator to model are regions that produce ‘unusual’ or ‘undesirable’ outputs (e.g. LFs without a clear exponential break), which are unlikely to be good matches to the observations anyway. Nevertheless, ideally we would like our emulator to return an estimate of its uncertainty (both the uncertainty in the emulator’s weights and uncertainty inherent in the data-generating process). GALFORM is a deterministic code, but we are still limited by the noise associated with sampling from a relatively small population of galaxies at high masses. Bayesian neural networks (Neal 1994; Bishop 1997) are a class of models that seek to incorporate epistemic and aleatoric uncertainty into the deep learning framework; these networks often apply independent Gaussian prior distributions over model weights, and model the outputs themselves as distributions. We believe this may be a promising line of inquiry to combine the power of the neural network’s adaptive basis functions with the uncertainty quantification of a full Bayesian analysis.

Another appealing method is the deep kernel learning approach (Wilson et al. 2016). Here, a deep neural network is employed to transform the inputs to the kernel of a Gaussian process regression, and it has been shown to outperform both the plain Gaussian process model and the plain deep neural network in a number of cases (e.g. Wilson et al. 2016; Patacchiola et al. 2020) while also providing robust uncertainty estimates. Here, the deep neural network can be thought of as a feature extractor that reduces the number of features input into the Gaussian process kernel and so allowing it to better generalize to higher dimensional inputs.

In Fig. 3, we demonstrated that we could improve the performance of our emulator as much as 10 per cent by averaging over 10 neural networks, rather than using just one. It may be interesting to investigate this avenue further. Our method used a simple average, but if a selection of machine learning algorithms are able to give errors that are not strongly correlated (i.e. some fit better to certain examples than others), it may be possible to use a more sophisticated approach to incorporate the respective advantages of a number of different algorithms (see e.g. Opitz & Maclin 1999).

We have proposed a number of ways to investigate the GALFORM model with our emulator. We can use sensitivity analysis techniques to evaluate the effect of different parameters, and since the emulator is extremely fast, we can manually explore the outputs in detail. It may also be possible to use symbolic regression such as the proprietary software EUREKA (as described in Dubčáková 2011) or sparse regression-based methods (see e.g. Rudy et al. 2019) to generate closed-form expressions of the neural network outputs if desired (i.e. an estimate of the functional form of the outputs). Cranmer et al. (2019), for example, applied symbolic regression techniques in conjunction with graph neural networks to extract equations from cosmological simulations.

5 CONCLUSIONS

We have implemented a deep learning approach to emulate the GALFORM SAM. We trained an ensemble of deep learning algorithms to approximate the full model using just 930 evaluations of GALFORM. We used this to explore the parameter space of GALFORM, and to calibrate the model parameters to a wide array of observations. Typically, the exploration of a model parameter space and the determination of a best-fitting set of parameter requires many more than 930 explicit full calculations. Our emulator is remarkably accurate, particularly in regions of the parameter space for which the model gives outputs that are close to matching the observed Universe.

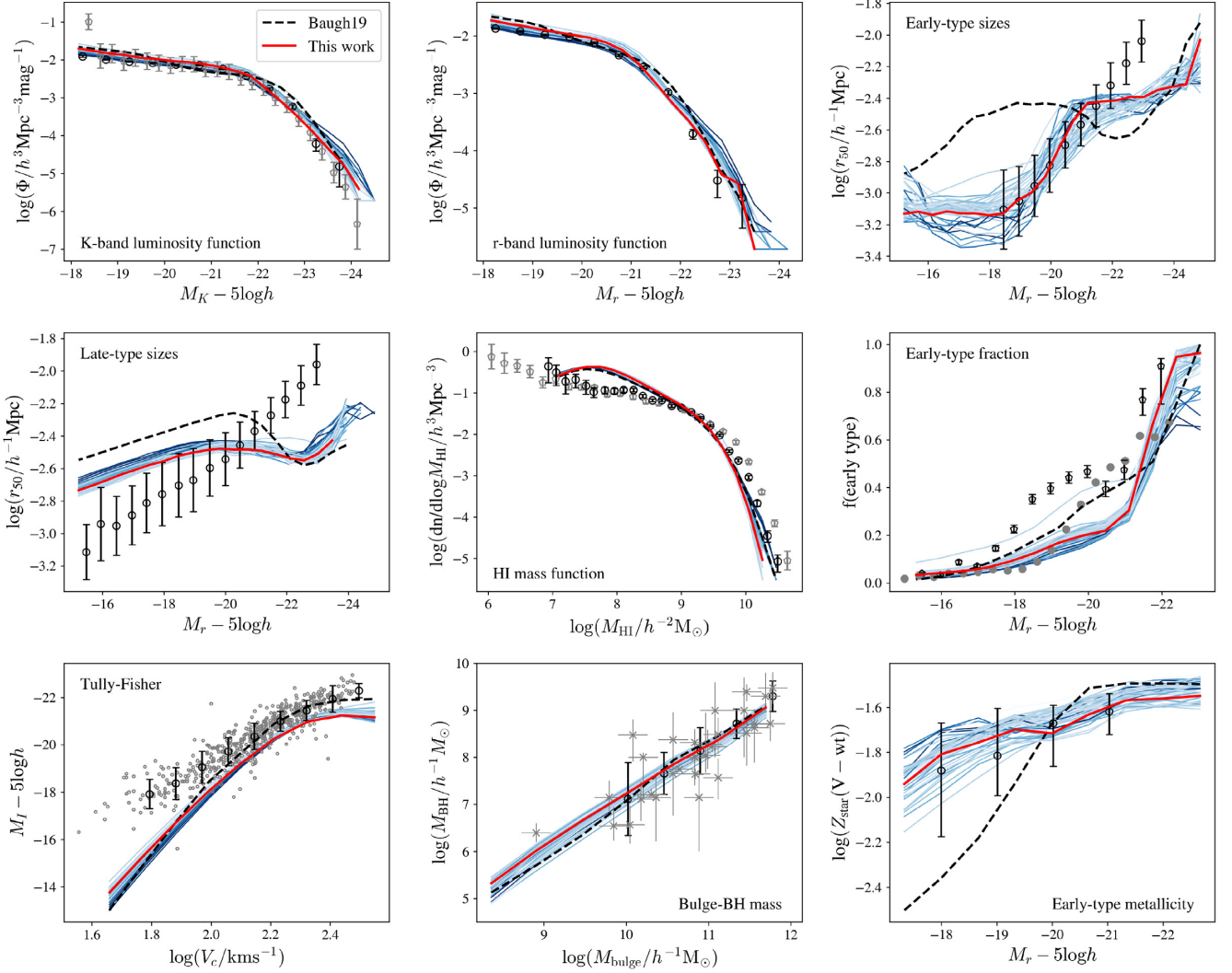


Figure 11. The GALFORM evaluations of the best-fitting parameters found with 100 MCMC chains, each 10 000 samples in length, using the constraint weightings described in the text. Here, we plot a sample of the best 50 runs, as measured by MAE. The red line indicates the parameter set with the lowest MAE. The remaining 49 runs are plotted in blue, with darker shades indicating small residuals to the H I mass function. Note therefore that runs with the smallest residuals to the H I mass function overpredict the bright-end of the K - and r -band LFs. The black dashed line shows the Baugh19 model. The data described in Section 2.4 are shown in black and grey. We calibrate to the data shown in black.

Table 2. The best-fitting parameters (as measured by MAE, equation 11) found by using MCMC combined with our emulator. For reference the last column gives the parameter values used by Baugh et al. (2019). The first column indicates the set of parameters with the lowest MAE, and the second column indicates the parameter ranges of the 50 best runs of the 100 MCMC chains, again selected by MAE as described in the text.

Parameter	This work	Range	Baugh19
f_{stab}	0.79	0.73–1.00	0.90
α_{cool}	0.84	0.66–1.16	0.80
α_{ret}	0.59	0.32–0.86	1.00
γ_{SN}	2.24	2.05–2.72	3.40
$V_{\text{SN,disc}} (\text{km s}^{-1})$	489	368–541	320
$V_{\text{SN,burst}} (\text{km s}^{-1})$	284	230–292	320
f_{burst}	0.25	0.12–0.30	0.05
f_{ellip}	0.20	0.20–0.39	0.30
$\nu_{\text{SF}} (\text{Gyr}^{-1})$	0.20	0.20–0.33	0.74
f_{SMBH}	0.003	0.001–0.004	0.005

We used sensitivity analysis to quantify the influence of different parameters on the model outputs, to better understand which parameters are of greatest importance in fitting to different observations (see Oleskiewicz & Baugh 2019). Here, as shown in Fig. 4, we found that the majority of the variance is due to just a few key parameters, which leads to tension when trying to calibrate to multiple observational data sets.

We explored the tensions between the use of different observational data sets further, using MCMC to fit the emulator output to observational data with a heuristic weighting scheme. This allowed us to reproduce the known tension between the faint-end galaxy LFs in the K and r bands and the late-type galaxy sizes, and to uncover a number of others. Furthermore, we used the same technique to find a global fit to the observational data sets, finding a set of parameters that provide an improved fit to the early-type galaxy sizes and metallicities as compared with an earlier version of the GALFORM code presented in Baugh et al. (2019).

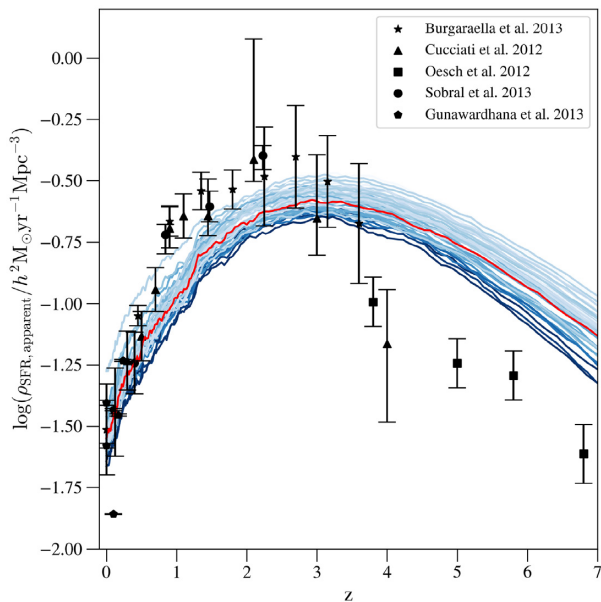


Figure 12. The apparent SFRD predictions for the GALFORM model evaluations shown in Fig. 11. The red line indicates the predictions for the best-fitting parameters (as calculated by MAE), while the blue lines indicate the remaining 49 runs. These lines are shaded according to the model’s residuals to the HI mass function, with darker shades indicating smaller residuals. We compare to observational data from Burgarella et al. (2013), Cucciati et al. (2012), Oesch et al. (2013), Sobral et al. (2013), and Gunawardhana et al. (2013). Note that these data were not used in the fitting. A correction has been applied to the predicted SFRD in bursts to give an apparent SFRD, as described in the text.

We intend to apply our emulation approach to calibrate GALFORM using the observed galaxy redshift distribution to generate mock galaxy catalogues for the DESI bright galaxy survey (Aghamousa et al. 2016). This requires model outputs over a large number of redshifts, which makes running GALFORM more computationally expensive. We are motivated therefore to reduce the required number of model evaluations as much as possible; calibrating the model across this redshift range would be prohibitively expensive for direct MCMC methods, and very difficult to achieve by eye. Our emulator is ideally suited to this task; we have demonstrated that we require very few runs to achieve good accuracy, and that we are able to emulate over a wide range of outputs.

We believe our approach to be an inexpensive, intuitive, and accurate alternative to other emulation techniques in the literature, and that this method will serve as an invaluable tool in quickly exploring and calibrating SAMs, and for the rapid assessment of the implications of changes to the underlying model.

ACKNOWLEDGEMENTS

EJE was supported by a PhD Studentship from the Durham Centre for Doctoral Training in Data Intensive Science, funded by the UK Science and Technology Facilities Council (STFC, ST/P006744/1) and Durham University. CMB and CGL acknowledge the support from STFC (ST/T000244/1). This work 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 capital 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 observational data sets, GALFORM outputs, parameter values, and best-fitting parameter values underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

- Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available at: <https://www.tensorflow.org/>
- Aghamousa A. et al., 2016, The DESI Experiment Part I: Science, Targeting, and Survey Design, preprint ([arXiv:1611.00036](https://arxiv.org/abs/1611.00036))
- Baugh C. M., 2006, *Rep. Prog. Phys.*, 69, 3101
- Baugh C. M. et al., 2019, *MNRAS*, 483, 4922 (Baugh19)
- Benson A. J., 2010, *Phys. Rep.*, 495, 33
- Benson A. J., Bower R., 2010, *MNRAS*, 405, 1573
- Benson A., Bower R., Frenk C., Lacey C., Baugh C., Cole S., 2003, *ApJ*, 599, 38
- Bigiel F. et al., 2011, *ApJ*, 730, 1
- Bishop C. M., 1997, *Journal of the Brazilian Computer Society*, 61, 4
- 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
- Burgarella D. et al., 2013, *A&A*, 554, 1
- Campbell D. J. et al., 2015, *MNRAS*, 452, 852
- Christodoulou D. M., Shlosman I., Tohline J. E., 1995, *ApJ*, 551, 443
- Cole S., Lacey C. G., Baugh C. M., Frenk C. S., 2000, *MNRAS*, 319, 168
- Crain R. A. et al., 2015, *MNRAS*, 450, 1937
- Cranmer M. D., Xu R., Battaglia P. W., Ho S., 2019, CoRR, abs/1909.05862
- Croton D. J. et al., 2006, *MNRAS*, 365, 11
- Cucciati O. et al., 2012, *A&A*, 539, A31
- De Jong R. S., Lacey C., 2000, *ApJ*, 545, 781
- De Oliveira R. A., Li Y., Villaescusa-Navarro F., Ho S., Spergel D. N., 2020, in Larochelle H., ed., Machine Learning and the Physical Sciences Workshop. NeurIPS 2020, Virtual Conference
- Driver S. P. et al., 2012, *MNRAS*, 427, 3244
- Dubčáková R., 2011, *Genet. Program. Evolvable Mach.*, 12, 173
- Efstathiou G., Lake G., Negroponte J., 1982, *MNRAS*, 199, 1069
- Emmert-Streib F., Yang Z., Feng H., Tripathi S., Dehmer M., 2020, *Front. Artif. Intell.*, 3, 1
- Forbes J. C., Krumholz M. R., Speagle J. S., 2019, *MNRAS*, 487, 3581
- Glorot X., Bengio Y., 2010, *J. Mach. Learn. Res.*, 9, 249
- Goldstein M., Wooff D., 2007, *Bayes Linear Statistics: Theory and Methods*, Wiley Series in Probability and Statistics. Wiley, Chichester
- González J. E., Lacey C. G., Baugh C. M., Frenk C. S., Benson A. J., 2009, *MNRAS*, 397, 1254
- Griffin A. J., Lacey C. G., Gonzalez-Perez V., Lagos C. d. P., Baugh C. M., Fanidakis N., 2019, *MNRAS*, 487, 198
- Gunawardhana M. L. et al., 2013, *MNRAS*, 433, 2764
- Håring 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
- Herman J., Usher W., 2017, *J. Open Source Softw.*, 2, 97
- Hochreiter S., Schmidhuber J., 1997, *Neural Comput.*, 9, 1735
- Hou J., Lacey C. G., Frenk C. S., 2018, *MNRAS*, 475, 543
- Kampakoglou M., Trotta R., Silk J., 2008, *MNRAS*, 384, 1414
- Kennicutt R. C. J., 1983, *ApJ*, 272, 54
- Kingma D. P., Ba J. L., 2015, in Bengio Y., LeCun Y., eds, 3rd International Conference on Learning Representations. ICLR 2015, San Diego, CA, USA

Kochanek C. S. et al., 2001, *ApJ*, 20, 566
 Lacey C. G. et al., 2016, *MNRAS*, 462, 3854
 Lagos C. d. P., Lacey C. G., Baugh C. M., Bower R. G., Benson A. J., 2011, *MNRAS*, 416, 1566
 Lagos C. d. P., Lacey C. G., Baugh C. M., 2013, *MNRAS*, 436, 1787
 Lange R. et al., 2016, *MNRAS*, 462, 1470
 Larson R. B., 1974, *MNRAS*, 169, 229
 Lu Y., Mo H. J., Weinberg M. D., Katz N., 2011, *MNRAS*, 416, 1949
 Lu Y., Mo H. J., Katz N., Weinberg M. D., 2012, *MNRAS*, 421, 1779
 Lu Y., Mo H. J., Lu Z., Katz N., Weinberg M. D., 2014, *MNRAS*, 443, 1252
 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
 Mathewson D. S., Ford V. L., Buchhorn M., 1992, *ApJS*, 81, 413
 Moffett A. J. et al., 2016, *MNRAS*, 457, 1308
 Neal R. M., 1994, *J. Am. Stat. Assoc.*, 92, 791
 Ntampaka M. et al., 2019, *Bulletin of the American Astronomical Society*, 51, 14
 Oesch P. A. et al., 2013, *ApJ*, 773, 75
 Oleskiewicz P., Baugh C. M., 2019, *MNRAS*, 493, 1827
 Opitz D., Maclin R., 1999, *J. Artif. Int. Res.*, 11, 169
 Patacchiola M., Turner J., Crowley E. J., O’Boyle M., Storkey A., 2020, *Advances in Neural Information Processing Systems*, 33, 16108
 Perraudin N., Srivastava A., Lucchi A., Kacprzak T., Hofmann T., Réfrégier A., 2019, *Comput. Astrophys. Cosmol.*, 6, 5
 Pillepich A. et al., 2017, *MNRAS*, 473, 4077
 Ravanbakhsh S., Oliva J., Fromenteau S., Price L. C., Ho S., Schneider J., Póczos B., Langford J., 2016, 33rd International Conference on Machine Learning, ICML 2016, New York, USA
 Reddi S. J., Kale S., Kumar S., 2018, ICLR, Vancouver, BC, Canada
 Robert C. P., 2004, *The Metropolis-Hastings Algorithm*, Monte Carlo Statistical Methods. Springer New York, New York, NY, p. 267
 Rodrigues L. F. S., Vernon I., Bower R. G., 2017, *MNRAS*, 466, 2418
 Rudy S., Alla A., Brunton S. L., Kutz J. N., 2019, *SIAM J. Appl. Dyn. Syst.*, 18, 643
 Ruiz A. N. et al., 2015, *ApJ*, 801, 139
 Saltelli A., 2017, *J. Chem. Inf. Model.*, 53, 1689
 Saltelli A., Annoni P., Azzini I., Campolongo F., Ratto M., Tarantola S., 2010, *Comput. Phys. Commun.*, 181, 259
 Schmit C. J., Pritchard J. R., 2018, *MNRAS*, 475, 1213
 Shen S., Mo H. J., White S. D., Blanton M. R., Kauffmann G., Voges W., Brinkmann J., Csabai I., 2003, *MNRAS*, 343, 978
 Simha V., Cole S., 2017, *MNRAS*, 472, 1392
 Smith R. J., Lucey J. R., Hudson M. J., 2009, *MNRAS*, 400, 1690
 Sobral D., Best P. N., Smail I., Mobasher B., Stott J., Nisbet D., 2013, *MNRAS*, 437, 3516
 Somerville R. S., Davé R., 2015, *ARA&A*, 53, 51
 Tieleman T., Hinton G., 2012, Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude, COURSERA: Neural Networks for Machine Learning, Vol. 4. p. 26
 Van Der Velden E., Duffy A. R., Croton D., Mutch S. J., 2021, *ApJS*, 253, 50
 Vernon I., Goldstein M., Bower R. G., 2010, *Bayesian Anal.*, 5, 619
 Weinberger R. et al., 2016, *MNRAS*, 465, 3291
 White S. D. M., Frenk C. S., 1991, *ApJ*, 379, 52
 Wilson A. G., Hu Z., Salakhutdinov R., Xing E. P., 2016, in Gretton A., Robert C., eds, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. AISTATS 2016, Cadiz, Spain, p. 370
 York D. G. et al., 2000, *AJ*, 120, 1579
 Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, in Wallach H., Larochelle H., Beygelzimer A., d’Alche-Buc F., Fox E., Garnett R., eds, *Thirty-third Conference on Neural Information Processing Systems*. NeurIPS 2019, Vancouver, Canada
 Zwaan M. A., Meyer M. J., Staveley-Smith L., Webster R. L., 2005, *MNRAS*, 359, L30

APPENDIX: SUPPLEMENTARY FIGURES

Here, we provide some additional figures to provide further illustration of points discussed in the main text.

Fig. A1 illustrates the one-at-a-time effect of varying the parameters f_{stab} and $V_{\text{SN, burst}}$, as a demonstration of their degenerate effects.

Fig. A2 shows the one-at-a-time effect of varying ν_{SF} on the K -band LF and late-type galaxy sizes. When fitting both the K -band LF and the late-type galaxy sizes, we see a decrease in the preferred value of ν_{SF} ; Fig. A2 demonstrates that this is because a lower ν_{SF} counteracts the enhancement in the bright-end of the K -band LF caused by the higher value of $V_{\text{SN, disc}}$ when including both constraints. We also see that reducing ν_{SF} marginally improves the fit to the late-type galaxy sizes.

Fig. A3 shows the accepted parameters of 20 MCMC chains when we fit the K -band LF (red), and when we fit to the K -band LF and the H I mass function (blue). Here, we see that including the H I mass function results in higher values of $V_{\text{SN, disc}}$ being preferred. ν_{SF} is also moved to the bottom end of the explored range, and α_{ret} becomes more sharply peaked and takes slightly higher values.

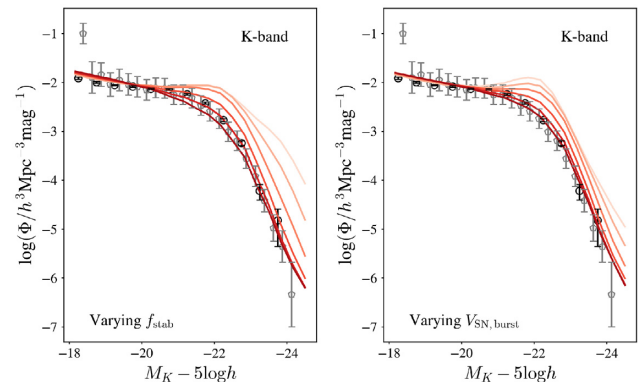


Figure A1. Emulator predictions for one-at-a-time perturbations of the parameters f_{stab} (left) and $V_{\text{SN, burst}}$ (right) around a fit to the K -band LF. We vary the parameters between the full range given in Table 1. Darker colours correspond to higher values. The data shown correspond to those described in Section 2.4.

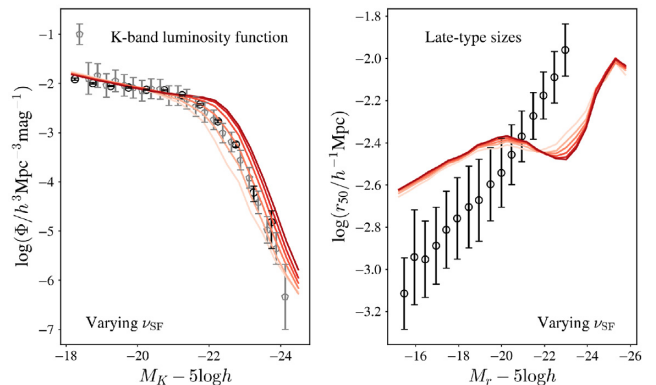


Figure A2. Emulator predictions for one-at-a-time perturbations of the parameter ν_{SF} for the K -band LF (left) and the late-type galaxy sizes (right) around a fit to the K -band LF. We vary the parameters between the full range given in Table 1. Darker colours correspond to higher values. The data shown correspond to those described in Section 2.4.

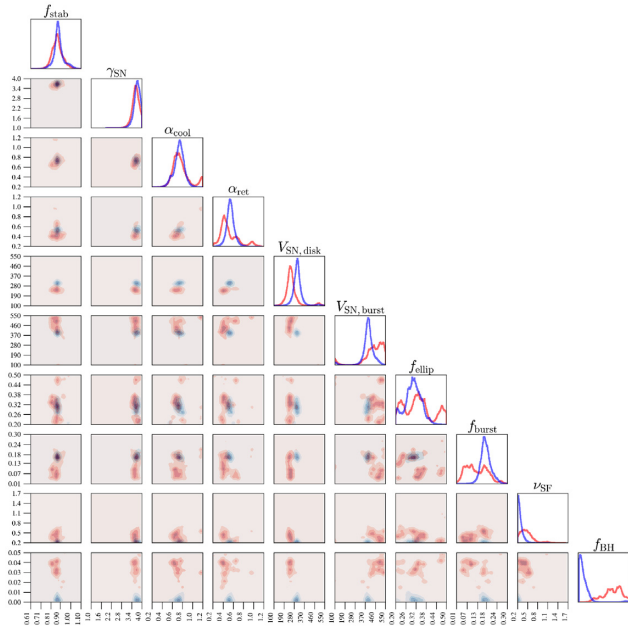


Figure A3. Accepted samples from 20 MCMC chains for fits to the K -band LF (red), and both the K -band LF and the H I mass function (blue). The first 50 per cent of samples were discarded to allow for burn-in. The histograms show the distribution of the parameters in 1D projection. The ranges on each axis are the same as those quoted in Table 1. The shading gives a sense of the density, 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.

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.