DRAFT VERSION DECEMBER 15, 2018
Typeset using LATEX **twocolumn** style in AASTeX61

EFFICIENT NEURAL NETWORK IMPLEMENTATION OF THE UNIVERSEMACHINE

ETHAN O. NADLER (05818244)[1, *] AND CHUN-HAO TO (06111342)[1, †]

[1] *Kavli Institute for Particle Astrophysics and Cosmology and Department of Physics, Stanford University, Stanford, CA 94305, USA*

ABSTRACT

The UNIVERSEMACHINE (UM; Behroozi et al. 2018) is a cosmological model that links dark matter (traced by *dark matter halos*) to luminous matter (traced by *galaxies*). This model is massively parallel and takes over 10^6 CPU hours to optimize based on the observed abundance and properties of galaxies derived from various surveys. Herein we explore an efficient implementation of the optimized UM model by using a random forest to identify key halo features and a deep convolutional residual network (ResNet) to learn the mapping from simulated halo distributions to observed galaxy distributions. Our ResNet accurately reproduces the abundance of galaxies as a function of stellar mass as predicted by UM and runs in a small fraction of the time, and our feature importance exploration highlights the most key aspects of the halo-to-galaxy mapping. We comment on several extensions of our deep learning model for future work. Our code is available at github.com/chto/umml.

Keywords: dark matter – galaxies: abundances – galaxies: halos – methods: numerical

1. INTRODUCTION AND RELATED WORK

One of the key challenges in cosmology is understanding the connection between luminous matter and dark matter over a wide range of lengthscales. This is a difficult problem on small scales due to our ignorance of the particle nature of dark matter, but the problem becomes simple on cosmological scales (i.e., lengthscales larger than about one megaparsec — 1 Mpc $\approx 10^{22}$ m). In this limit, luminous matter is traced by *galaxies*, which we observe according to their total stellar mass M_* and redshift (a proxy for their distance from us), while dark matter is traced by *halos*, which are also defined by their mass (M_h) and redshift. In standard cosmological models, all galaxies reside in dark matter halos; thus, understanding the relationship between luminous matter and dark matter boils down to understanding the connection between galaxies and halos, and specifically the M_*–M_h relation.

A simplified version of the standard method for constraining the M_*–M_h relation proceeds as follows:

1. Measure the number density of galaxies as a function of M_* (ϕ_*, referred to as the *stellar mass function* or SMF) in a galaxy survey;
2. Model the number of dark matter halos as a function of M_h using a cosmological simulation;
3. Assign galaxies to halos using a particular M_*–M_h relation, and fit this relation to the observed SMF.

Several authors (e.g., Behroozi et al. 2013) have constrained the galaxy–halo connection in this way, and Behroozi et al. (2018) recently presented the UNIVERSEMACHINE (UM), a forward model that predicts galaxy population statistics from simulated dark matter halo populations and constrains the galaxy–halo connection using a variety of observational data. Unfortunately, UM is computationally expensive; Behroozi et al. (2018) run 4 million Markov Chain Monte Carlo steps for 2.4 million CPU hours to optimize their model. In addition, UM assumes a specific form for the mapping between galaxies and halos, limiting its generality.

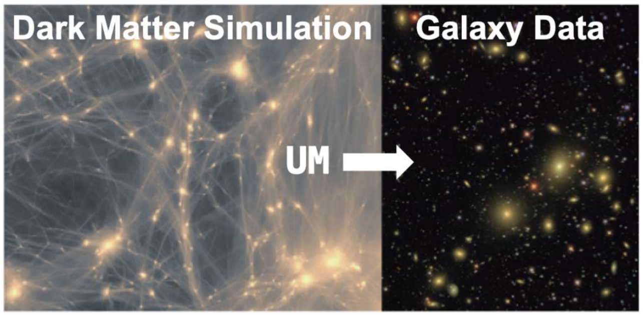

Figure 1. Visualization of the UNIVERSEMACHINE mapping captured by our deep learning model. Dark matter halos from a cosmological simulation are mapped to galaxies, and this mapping is optimized based on the observed abundance and properties of galaxy populations from various surveys.

Herein we explore the optimized UM model using a random forest (RF) to extract relevant halo features and a deep convolutional residual neural network (ResNet) to learn the halo-to-galaxy mapping predicted by UM. The advantages of applying deep learning to this problem are twofold: a ResNet trained to reproduce UM output provides a faster implementation of the model, allowing effi-

[*] enadler@stanford.edu
[†] chto@stanford.edu

cient querying; in addition, our method is not limited to a particular model for the galaxy–halo connection, meaning that our ResNet can potentially *learn* this mapping rather than determining the best-fit results for a given model.

2. DATASET AND FEATURES

We use dark matter halo catalogs from the Bolshoi-Planck simulation (Klypin et al. 2016), which tracks the evolution of $\approx 10^{10}$ dark matter particles, each of mass 10^8 $\mathrm{M}_\odot$ (solar masses), in a cubic box of side length 250 h^{-1} Mpc that represents the expanding universe.[1] These catalogs contain the 3D position (x,y,z) and a variety of internal properties for each halo (e.g., mass, size, spin, etc.) as well as the relationships between halos at different snapshots. The halo-to-galaxy mapping underlying UM only explicitly depends on the *peak circular velocity* V_{peak} of dark matter halos, which is a numerically stable mass proxy, although other halo features enter the mapping implicitly. We explore optimal feature selection in Section 3.1.

We find $\sim 10^7$ halos at the final snapshot of the Bolshoi-Planck simulation. We discard all halos with $V_{\mathrm{peak}} < 150$ km s^{-1} to ensure that we study well-resolved objects and for computational efficiency while prototyping our network. This cut results in $\sim 2\times 10^6$ halos in our fiducial dataset.

The UM data we employ are provided by Behroozi et al. (2018) at peterbehroozi.com/data.html. In particular, we use the UM prediction for the galaxies that inhabit the simulated dark matter halos described above. The abundance and properties of these galaxies match measurements from the PRIMUS survey (Moustakas et al. 2013); for example, the SMF $\phi_*(M_*)$ represents the number density of predicted galaxies in units of h^3 Mpc^{-3} dex^{-1}, and it is tuned to match the observed SMF in 0.1 dex logarithmic stellar mass bins from $M_* = 10^9\ h^{-2}\ \mathrm{M}_\odot$ to $M_* = 10^{12}\ h^{-2}\ \mathrm{M}_\odot$. We incorporate the covariance of these binned SMF predictions, which is estimated from the observational data, into the training of our ResNet model.

3. METHODS

3.1. *Random Forest Model*

To determine the most relevant halo feature(s) for learning the UM model and to benchmark our ResNet results against a simple algorithm, we train a RF to learn the mapping from the internal features of dark matter halos to the stellar masses of the galaxies that reside within these halos. In particular, we train a RF on 80% of the halos described above using the eight halo features listed in Figure 3 and the galaxy stellar mass labels predicted by UM. We use the Scikit-Learn package to implement our RF, and we use the GridSearchCV module to optimize over RF hyperparameters and select the ones that yield the highest out-of-bag (OOB) classification score averaged over 5 cross-validation folds of the training data. These hyperparameters include the number of decision trees, the depth of each tree, and the maximum number of features used at each decision tree split. Our qualitative results are not sensitive to the values of these hyperparameters.

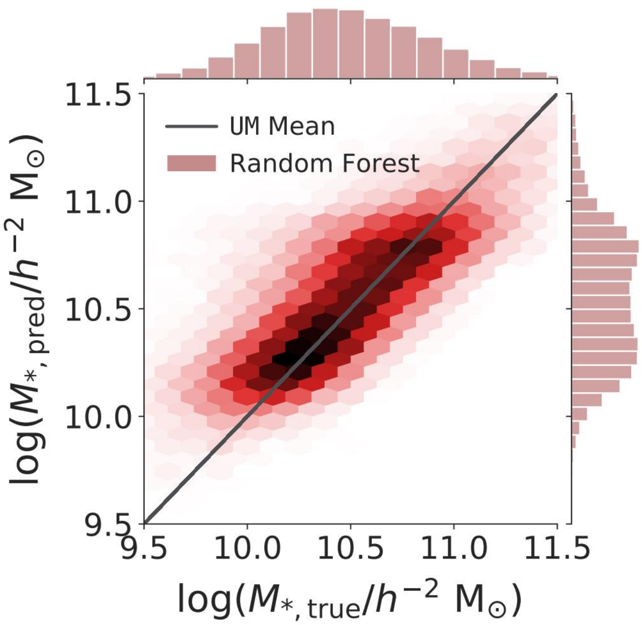

Figure 2. Galaxy masses predicted by our random forest versus the true galaxy masses predicted by UNIVERSEMACHINE for dark matter halos in our test set. The predicted values are indicated by shaded hexagons, with darker colors corresponding to higher number densities. Our RF accurately captures the mean UM halo-to-galaxy mass relation and the expected ~ 0.2 dex of scatter in $M_{*,\mathrm{pred}}$ at fixed $M_{*,\mathrm{true}}$.

To test whether our RF accurately captures the UM halo-to-galaxy mapping, we plot the predicted galaxy mass corresponding to each halo as a function of the UM-predicted galaxy mass in Figure 2. The RF predictions are centered around the mean UM-predicted masses, and the scatter in the RF predictions is consistent with the expected ~ 0.2 dex scatter in the underlying galaxy–halo connection for these systems. Thus, our RF provides a relatively accurate version of the UM model that we will use to benchmark our ResNet results.

Figure 3 shows the relative feature importances obtained from our RF (i.e., the feature importances normalized by the highest feature importance score). As expected, V_{peak} dominates the feature importances, since this is the only halo feature that explicitly enters the UM model; however, the fact that the remaining feature importances are nonzero shows that these secondary halo properties correlate with the galaxy properties predicted by UM. Because V_{peak} dominates the galaxy-to-halo mapping, we base our ResNet on this feature as described in the next section.

3.2. *Deep Learning Model*

We now describe our deep convolutional ResNet, which is based on the ResNet18 model.[2] To process the halo data into a suitable form for our ResNet, we slice the (250 h^{-1} Mpc)^3 simulation box

[1] h is a cosmological parameter which is fixed at 0.678 to match the simulations used in this work.

[2] download.pytorch.org/models/resnet18-5c106cde.pth

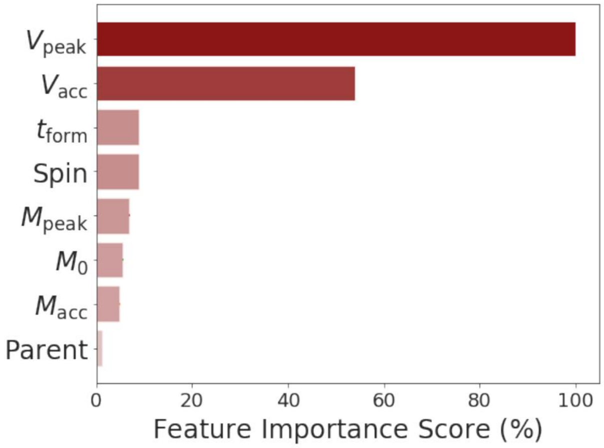

Figure 3. Feature importances normalized by the highest score among all features for our random forest trained on the halo-to-galaxy mapping predicted by UM. In descending order, the features represent dark matter halo peak circular velocity evaluated at two different times, formation time, spin, mass evaluated at three different times, and whether the halo lives inside of a larger halo. $V_{\rm peak}$ is the most important halo property in the halo-to-galaxy mapping.

into 10^3 sub-boxes of side length 25 h^{-1} Mpc, and we segment each sub-box into 25 contiguous sub-volumes; this procedure typically yields at least one halo per sub-volume. We then assign each of these sub-volumes a value corresponding to the mean $V_{\rm peak}$ of the halos in that region.

We illustrate our fiducial architecture in Figure 4. We use several 3×3 filter-size 3D convolutional layers with stride $s=2$, each of which is followed by a batch normalization operation and a fully-connected layer that uses a ReLu activation function. After preprocessing the data in this way and max-pooling the results, we implement three layers, each of which consists of two convolution-batch normalization-ReLu operations with a residual connection. The two subsequent fully-connected layers use ReLu activations with dropout, and the final layer uses a linear activation function since we are predicting a continuous-valued quantity.

Before detailing our optimization procedure, we outline our general strategy:

1. For a given sub-box, feed the segmented $V_{\rm peak}$ values into a ResNet that outputs $\phi_*(M_*)$ in 14 discrete M_* bins;
2. Train on 80% of the sub-boxes from Bolshoi-Planck using a χ^2 likelihood with a mean set by the UM-predicted SMF in that sub-box and a covariance set by the Moustakas et al. (2013) measurements in each M_* bin;
3. Validate using the remaining sub-boxes and by comparing the results to the RF and UM-predicted SMFs.

We implement our ResNet using PyTorch (Paszke et al. 2017). To train the network, we define the χ^2 loss function

$$\mathcal{L}=\frac{1}{N_{\rm boxes}}\frac{1}{N_{\rm bins}}\sum_{\text{boxes } i}(\log\phi^{\rm obs}_{*,i}-\log\phi^{\rm pred}_{*,i})^T\Sigma^{-1}(\log\phi^{\rm obs}_{*,i}-\log\phi^{\rm pred}_{*,i}),\qquad(1)$$

where $N_{\rm bins}=14$ is the number of M_* bins (evenly spaced logarithmically), $\phi^{\rm obs}_{*,i}$ ($\phi^{\rm pred}_{*,i}$) are 14×1 vectors that represent the "observed" (i.e., UM-predicted) and ResNet-predicted SMF in sub-box i, and Σ^{-1} is the covariance matrix of the observed SMF, which we take to be diagonal with entries given by the Gaussian errors reported in Moustakas et al. (2013). Note that this loss function is simply a χ^2 test for the likelihood of our predicted SMF values given the null hypothesis of a normally distributed SMF about the UM-predicted mean in each sub-box. We train on $N_{\rm boxes}=800$ sub-boxes, reserving 200 sub-boxes for our validation set.[3]

We train our model for 15 epochs using the PyTorch implementation of the Adam optimizer (Kingma & Ba 2014) with the following hyperparameters: a minibatch size of 32 sub-boxes and a hand-tuned learning rate of 10^{-1} for the first 10 epochs and 10^{-3} for the remaining 5 epochs. These hyperparameters were chosen manually by inspecting the behavior of the loss function.

In addition, we optimize several aspects of our architecture; for example, the network used in our milestone implemented 2D convolutions, which resulted in biased predictions using our updated loss function. We find that 3D convolutions result in less biased predictions and smaller characteristic values of $\mathcal{L}$ on both the training and test sets, which is reassuring because the simulated dark matter halo distribution is inherently three-dimensional. The number of convolutional channels is a free parameter in our framework, and we scanned over values from $N_{\rm channels}=20$ to $N_{\rm channels}=200$ by calculating the average loss on the validation set; as expected, we find that increasing $N_{\rm channels}$ results in a smaller loss, and we use $N_{\rm channels}=200$ for the results presented below.[4]

4. RESULTS AND DISCUSSION

In Figure 5, we plot the training and cross-validation loss as a function of training epoch. Although the magnitude of our validation loss is somewhat large when interpreted as a χ^2 statistic, the training and validation loss steadily decrease until training epoch 5, after which they plateau. However, we note that these loss values are significantly smaller than those reported in the milestone due to the increased complexity of our network. Note that we decreased the learning rate after 10 training epochs because we observed that the predicted SMF simply fluctuated about its mean value when training past 10 epochs with a constant learning rate.

To assess our predicted SMF, we compare our ResNet output to the UM result and to our RF prediction. Table 1 lists the values of the χ^2 loss averaged over validation sub-boxes for our ResNet and RF models; the limiting SMF values obtained by our ResNet are clearly significantly closer to the mean UM values. Figure 6 illustrates the SMF for these validation sub-boxes; the errorbars on the RF and ResNet predictions represent 1σ scatter about the mean predictions. We observe that the ResNet-predicted SMF matches the UM prediction at all galaxy masses and outperforms the RF both in

[3] We do not distinguish between validation and test sets because of the limited number of available sub-boxes. We hope to treat this issue more carefully in future work based on a larger number of simulations.

[4] The average begins to plateau around $N_{\rm channels}=100$; see Table 1.

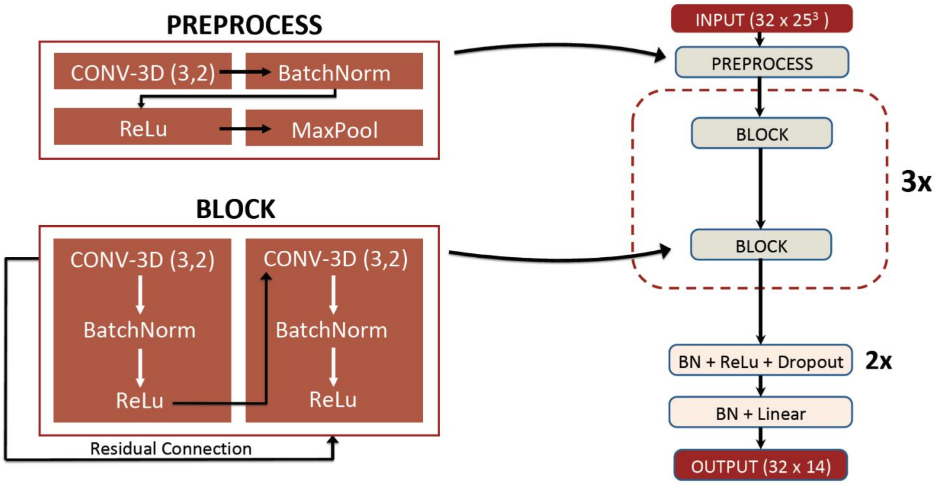

Figure 4. Illustration of the architecture for our convolutional ResNet model. The input distribution of dark matter halos corresponds to $N_{\rm minibatch} = 32$ sub-boxes, each of volume $(25\ h^{-1}\ {\rm Mpc})^3$; for each minibatch, the output corresponds to a galaxy population described by a 14×1 stellar mass function (i.e., the number density of galaxies in 14 logarithmically spaced bins of galaxy mass).

RF	ResNet ($N = 20$)	ResNet ($N = 100$)	ResNet ($N = 200$)
272	135	93	84

Table 1. Average χ^2 loss (Equation 1) over validation sub-boxes for our RF and our ResNet with $N = 20$, 100, and 200 channels.

terms of smaller mean errors and smaller uncertainties. This is an encouraging result, since our ResNet trains and makes predictions in a small fraction of the time it takes UM to perform these tasks.

The physical implications of this result are also interesting: our ResNet is trained using the mean $V_{\rm peak}$ values of the halos in each sub-volume, while our RF utilizes *all* available internal halo properties. However, the 3D convolutions implemented by our ResNet capture spatial correlations among the $V_{\rm peak}$ values of the simulated halos, which in turn encode information about the galaxies that reside in these halos, leading to a more accurate prediction. Finally, we reiterate that the UM prediction is tuned to match the *observed* number density of galaxies in the range plotted in Figure 6, meaning that our ResNet has learned how to map simulated dark matter halo populations to galaxy populations that are consistent with those found *in actual survey data*.

5. FUTURE WORK

We plan to generalize our model by using *all* internal halo properties (rather than just $V_{\rm peak}$) as input features; although our feature selection exploration in Section 3.1 suggests that this will only yield a modest improvement, it is possible that our ResNet will capture additional correlations among internal halo features that lead to more accurate galaxy population predictions. We also plan to generalize our network's output as follows: i) predict and jointly fit to both the SMF and the remainder of the observables predicted by UM (namely, star formation rates and the fraction of non star-forming galaxies in a given population) at a fixed simulation snapshot; ii) predict observables as a function of time using multiple simulation snapshots.

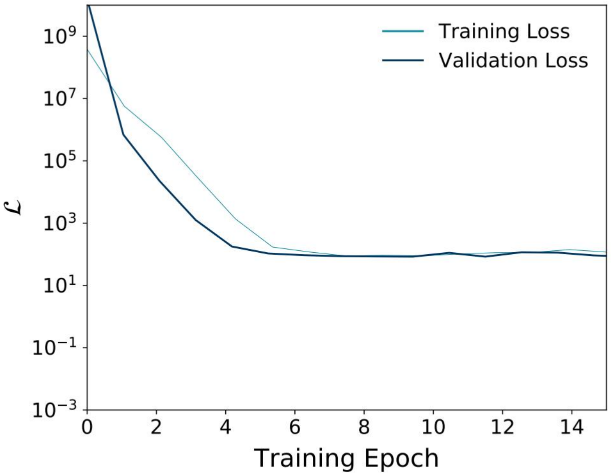

Figure 5. χ^2 loss function (Equation 1) averaged over 800 training sub-boxes (light blue) and 200 validation sub-boxes (dark blue) versus training epoch. Note that we decreased the learning rate after 10 training epochs for stability. Both quantities decrease during the first ~ 5 training epochs and then plateau, indicating that our ResNet is learning the halo-to-galaxy mapping.

In addition, we plan to test how our model generalizes to independent simulations and galaxy populations. For example, is the halo-to-galaxy mapping learned by our ResNet general enough to predict realistic galaxy populations from simulations that differ in detail from Bolshoi-Planck (e.g., in simulation size)? Can its predictions be extrapolated beyond the $V_{\rm peak}$ values of the dark matter halos in the training set? Finally, we note that in a more sophisticated treatment it might be necessary to impose physical priors

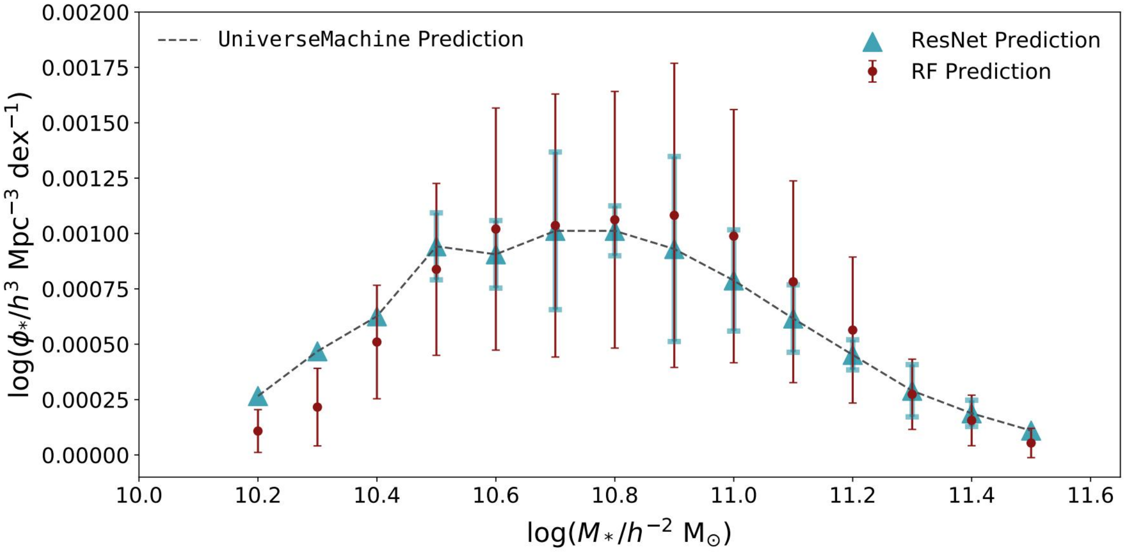

Figure 6. Stellar mass function predicted by the UNIVERSEMACHINE (gray dashed line), our ResNet (blue triangles and errorbars), and our RF (red circles and errorbars) for sub-boxes in our validation set. The errorbars on the ResNet and RF predictions indicate 1σ scatter about the mean prediction averaged over the validation set. Our ResNet prediction matches the mean UM SMF accurately and predicts galaxy populations with smaller scatter than our RF. Note that the SMF does not increase monotonically for smaller galaxies because of our halo resolution cut.

on the smoothness and absolute values of the ResNet output. For example, by modifying the loss function appropriately, we might require the bin-to-bin variations in the SMF to be bounded from above, since averaged galaxy number counts should not vary drastically as a function of stellar mass.

APPENDIX

A. CONTRIBUTIONS

The authors (EN and CH-T) collaborated on the work described above, with the following exceptions: EN located the simulation data and produced the figures, and CH-T ran and monitored the RF and ResNet training jobs on the Sherlock HPC cluster.

B. ACKNOWLEDGEMENTS AND SOFTWARE

Our ResNet implementation is based on 1) ResNet18 (download.pytorch.org/models/resnet18-5c106cde.pth) and 2) the starter code provided at github.com/cs230-stanford/cs230-code-examples. This research made use of computational resources at SLAC National Accelerator Laboratory, a U.S. Department of Energy Office. This research made use of the following community-developed or maintained software packages: IPython (Pérez & Granger 2007), Jupyter (jupyter.org), NumPy (van der Walt et al. 2011), PyTorch (Paszke et al. 2017), Scikit-Learn (Pedregosa et al. 2011), and TensorFlow (Abadi et al. 2015).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from tensorflow.org

Behroozi, P., Wechsler, R., Hearin, A., & Conroy, C. 2018, ArXiv e-prints, arXiv:1806.07893

Behroozi, P. S., Wechsler, R. H., & Conroy, C. 2013, ApJ, 770, 57

Kingma, D. P., & Ba, J. 2014, ArXiv e-prints, arXiv:1412.6980

Klypin, A., Yepes, G., Gottlöber, S., Prada, F., & Heß, S. 2016, MNRAS, 457, 4340

Moustakas, J., Coil, A. L., Aird, J., et al. 2013, ApJ, 767, 50

Paszke, A., Gross, S., Chintala, S., et al. 2017

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research

Pérez, F., & Granger, B. E. 2007, Computing in Science Engineering, 9, 21

van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in Science Engineering, 13, 22