Learning Galaxy Properties from Merger Trees

Christian Kragh Jespersen [*1] **Miles Cranmer** [1] **Peter Melchior** [1 2] **Shirley Ho** [3 1 4] **Rachel S. Somerville** [3]
Austen Gabrielpillai [5 6 7]

Abstract

Efficiently mapping baryonic properties onto dark matter is a major challenge in astrophysics. Although semi-analytic models (SAMs) and hydrodynamical simulations have made impressive advances in reproducing galaxy observables across large cosmological volumes, these methods still require significant computation times, representing a barrier to many applications. However, with Machine Learning, simulations and SAMs can now be emulated in seconds. Graph Neural Networks (GNNs) are a powerful class of learning algorithms which can naturally incorporate the very structure of data, and have been shown to perform extremely well on physical modeling, and among the most inherently graph-like structures found in astrophysics are the dark matter merger trees used by SAMs. In this paper we show that several baryonic targets—as predicted by a SAM—can be emulated to unprecedented accuracy using a trained GNN, four orders of magnitude faster than the SAM. The GNN accurately predicts stellar masses for a range of redshifts, and interpolates successfully at redshifts where it was not trained. We compare our results to the current state of the art in the field, and show improvements in mean-squared error of up to a factor of four.

[*]Equal contribution [1]Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA [2]Center for Statistics and Machine Learning, Princeton University, Princeton, NJ 08544, USA [3]Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA [4]Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15217, USA [5]Institute for Astrophysics and Computational Sciences, Catholic University of America, 620 Michigan Ave., DC 20064, USA [6]Astrophysics Science Division, NASA/GSFC, 8800 Greenbelt Rd, Greenbelt, MD 20771, USA [7]Center for Research and Exploration in Space Science and Technology, NASA/GSFC, 8800 Greenbelt Rd, Greenbelt, MD 20771, USA. Correspondence to: Christian Kragh Jespersen <ckragh@princeton.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Machine Learning for Astrophysics Workshop, Baltimore, Maryland, USA, PMLR 162, 2022.

1. Introduction

In the hierarchical paradigm of ΛCDM cosmology, dark matter is a crucial constituent of galaxy formation. While modeling the evolution of universes with only dark matter can be done both analytically (Sheth et al., 2001) or through numerical N-body simulations (Aarseth et al., 1979; Efstathiou et al., 1985; Maksimova et al., 2021), co-evolving dark matter and baryons still represents a major challenge, as no simple, direct mapping between the two exists (Contreras et al., 2015; de Santi et al., 2022). Instead we turn to simulations for modeling these complex interactions. One widely accepted framework for doing so is semi-analytic models (SAMs), which in the last two decades have made it possible to populate cosmologically significant volumes with galaxies (Somerville et al., 2008; Somerville & Davé, 2015; Naab & Ostriker, 2017). Although SAMs achieve much greater computational efficiency than hydrodynamic simulations by combining dark matter *merger trees* with a suite of physically motivated recipes for evolving the baryonic components of galaxies, they still require several hundreds of CPU hours to fill a $(75Mpc/h)^3$ simulation box (White & Frenk, 1991; Somerville & Primack, 1999; Benson, 2012; Lacey et al., 2016; Lagos et al., 2018).

Kamdar et al. (2016); Agarwal et al. (2018); Jo & Kim (2019); Lovell et al. (2022); de Santi et al. (2022) each attempt to map between dark matter and galactic baryonic properties using simple machine learning (ML) algorithms, like Extremely Randomized Trees, Random Forests, Multi-Layer Perceptrons or a combination of the above. These methods all use only features from the final halos at $z = 0$, or summary statistics believed to encode the merger history along with the features of the $z = 0$ halo to learn this map. Even in cases where these methods were able to predict the *median* values of a quantity with relatively low error, they typically underestimate the *dispersion* in the baryonic property at a given halo mass (Agarwal et al., 2018).

In this work, we present a new method for learning this non-trivial mapping, using the natural choice for learning on merger trees, a Graph Neural Network (GNN). The GNN outperforms all previous models in the literature. This indicates that exploiting the inherent structure of the merger tree indeed is the stronger choice for mapping directly between

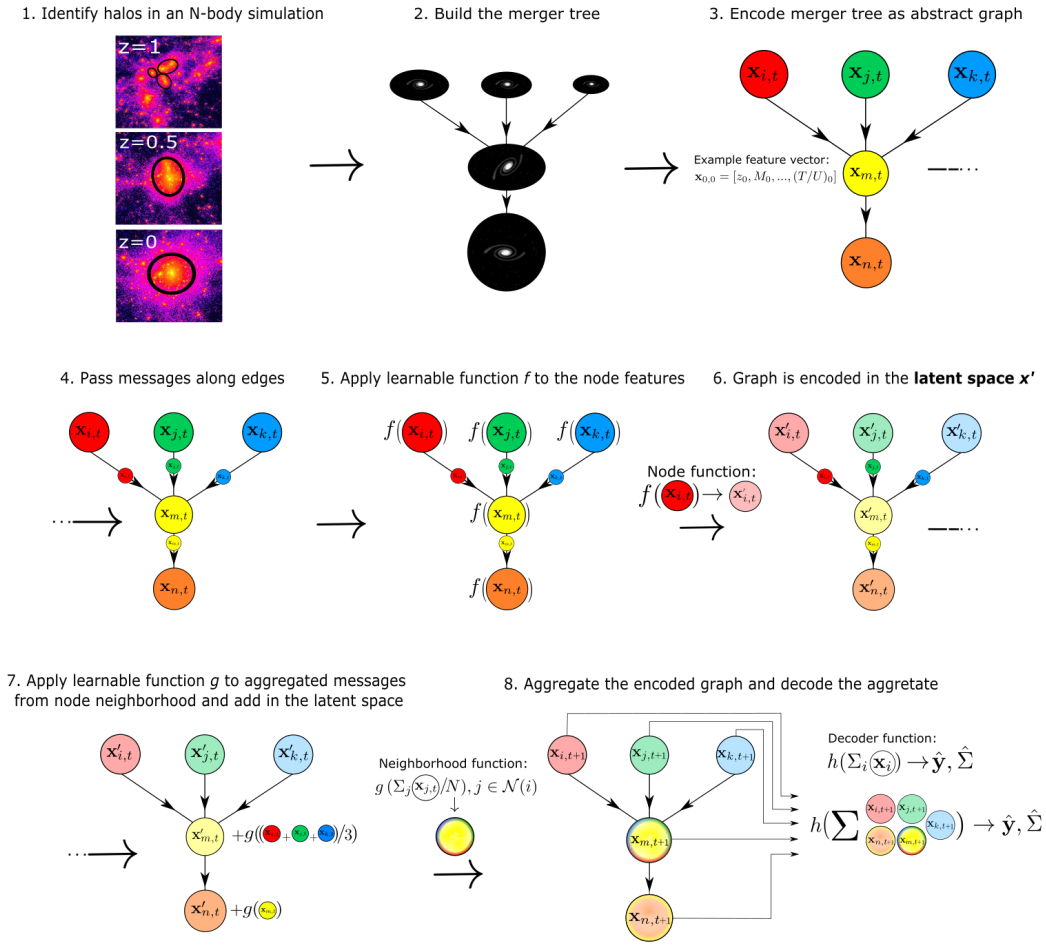

Figure 1. An illustration of our technique. Merger trees are encoded as graphs, which are then passed trough a GNN. Messages are passed forward in time only, since merger trees are directed in time. Node states are updated by applying a learnable function f to the current node states, applying a learnable function g to the mean of the node states of the neighboring nodes and adding these two as described in §3. Latent space is marked by shaded colors. Adding neighborhood information is marked by mixing of colors. After five of these message-passing steps, the graph nodes are summed over, and this sum is then decoded by another learnable function, h, which gives the predictions and the Gaussian covariance matrix. All learnable functions are MLPs.

dark matter and baryonic properties.

2. Simulations and Data

We use the dark matter only version of the IllustrisTNG simulation, TNG-100-1-Dark. This simulation contains $(1820)^3$ particles within a box of 75 h^{-1} on a side. This implies a dark matter particle mass of 6×10^6 $h^{-1} M_\odot$. The halo finding code ROCKSTAR (Behroozi et al., 2013a) has been run on 99 snapshots from this simulation, and the CONSISTENTTREES (Behroozi et al., 2013b) code is used to construct merger trees from these halo catalogues. See Gabrielpillai et al. (2021, hereafter G21) for more details on the halo finding and merger tree algorithms. See the appendix for details on our treatment of the merger trees. We then run the well-established Santa Cruz semi-analytic model (Somerville & Primack, 1999; Somerville et al., 2008; 2015) on the merger trees described above. The current version of the SC-SAM is documented in G21.

SAMs output a large range of baryonic galactic properties, but for exploring the possibility of emulating them with a GNN, we pick a few quantities of interest.

The main target of interest is **stellar mass** ($\log(M_*/M_\odot)$, hereafter M_*). This is the central quantity for both creating mock catalogues and for simulators to successfully reproduce, and is therefore also the main focus of this project. To explore the possibility of emulating other baryonic proper-

ties, we also include a range of other targets. We include cold gas (ISM) mass ($\log(\mathrm{M_{cold}}/\mathrm{M_\odot})$, hereafter M_{cold}), black hole mass ($\log(\mathrm{M_{BH}}/\mathrm{M_\odot})$, hereafter M_{BH}), cold gas (ISM) metallicity ($\log(\mathrm{M_{Z_{gas}}}/\mathrm{M_{cold}})$), hereafter Z_{gas}), instantaneous Star Formation Rate ($\log(\mathrm{SFR}/\mathrm{M_\odot}/\mathrm{yr})$, hereafter SFR), and Star Formation Rate (SFR) averaged over 100 Myr ($\log(\mathrm{SFR_{100}}/\mathrm{M_\odot}/\mathrm{yr})$, hereafter SFR_{100}).

3. Graph Neural Networks

In machine learning, the most successful models are the ones which embed well-motivated inductive biases into the model that one wishes to fit—such as convolutional neural networks for images or recurrent neural networks for sequences. Since halo and galaxy evolution are naturally encoded in merger trees, which are graphs, a Graph Neural Network (GNN) is a natural choice of architecture when regressing quantities from a merger history, such as mapping galactic baryonic physics onto dark matter. GNNs are neural networks, but for graph-like data. They are usually implemented as a sequential series of message-passing/graph convolutional layers (Kipf & Welling, 2017; Battaglia et al., 2018) which pass information from the **nodes** along the **edges** of the graph, followed by a differentiable pooling function and a decoder function, which is usually a Multi-Layer Perceptron (MLP) (Rumelhart et al., 1986).

In this project we use GraphSAGE convolutional layer (Hamilton et al., 2017). With each application of this layer, each node is embedded from the input state $\mathbf{x}_i$ into a *hidden state* $\mathbf{x}'_i$, through:

$$\mathbf{x}'_i = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2 \cdot \mathrm{mean}_{j\in\mathcal{N}(i)}\mathbf{x}_j \tag{1}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices and $\mathcal{N}(i)$ denotes the neighborhood of node i. Thus, $\mathbf{W}_1$ operates on information from the node itself, and $\mathbf{W}_2$ on the mean of the node states of the neighborhood nodes. An *activation function*, is applied to the output of this mapping, allowing expression of nonlinear functions. In this work we use the ReLU activation function between these GraphSAGE layers. We train the model by maximizing a Gaussian likelihood.

A description of the architecture of the model, as well as training details, core concepts about graphs and information about the loss function can be found in Appendix §B - E.

4. Results

In this section, we introduce the metrics used to characterize the performance of the GNN, and present our results. We compare our results to results from two other frameworks.

- Our M_* prediction will be compared to the more widely used method for connecting halo masses and galactic stellar masses, Abundance Matching (AM) (Vale & Ostriker, 2004).
- In order to faithfully compare to the SOTA, we train a MLP on the $z = 0$ halos (final halos) of our dataset. This is comparable to the methods in the literature, and avoids dataset biases.

Table 1. Metrics for the methods discussed in this paper. GNN denotes the results of our model using the full merger history and all halo parameters. Final halo denotes the results of our model using all halo parameters for the $z = 0$ halo, i.e., the final halo. This is the current state-of-the-art (SOTA) method. We bold the best performance for each metric for each target variable.

Target	Method	σ [dex]	Bias [dex]	$\rho_{Pearson}$
M_*	GNN	**0.070**	0.002	**0.997**
	Final halo	0.132	0.003	0.990
	AM	0.311	**0.000**	0.945
M_{cold}	GNN	**0.161**	-0.009	**0.954**
	Final halo	0.182	**0.001**	0.941
M_{BH}	GNN	**0.127**	**-0.001**	**0.975**
	Final halo	0.175	-0.013	0.951
Z_{gas}	GNN	**0.123**	**-0.005**	**0.974**
	Final halo	0.151	-0.007	0.960
SFR	GNN	**0.353**	-0.025	**0.936**
	Final halo	0.392	**0.002**	0.921
SFR_{100}	GNN	**0.347**	0.022	**0.938**
	Final halo	0.388	**0.003**	0.922

Results from using the GNN are compared to results from these two methods in Table 1. The GNN using the full merger history vastly outperforms the current SOTA and traditional methods across variables, except for negligible differences in the bias. Note that models regressing more targets generally performed better due to weight smoothing. Results cited are for models trained to predict all targets unless otherwise stated.

4.1. Metrics

We will primarily compare estimators using the **scatter** (RMSE) of their predictions with respect to the truth, defined as:

$$\sigma(\Delta y) = \sqrt{\frac{1}{N_{test}}\sum^{N_{test}}(\Delta y - \overline{\Delta y})^2} \tag{2}$$

where $\Delta y \equiv y - \hat{y}$ is the residual of a single prediction and $\overline{\Delta y}$ is the mean of the residuals. Since this metric does not measure any systematic offset in the residuals we introduce the **bias** as an auxiliary metric (which we will not explicitly optimize for), i.e.:

$$\text{bias}(\Delta y) = \sum^{N_{test}} \Delta y / N_{test} \quad (3)$$

The bias measures systematic offset. Since the scatter is susceptible to outliers, we also include the Pearson correlation coefficient (ρ), i.e., the linear correlation between the target and the model prediction: $\rho \equiv \text{cov}\,(y, \hat{y})/\sigma_y \sigma_{\hat{y}}$.

4.2. Stellar Mass at Other Redshifts

Whether a given ML model generalizes outside of the distribution of data it has seen during training is crucial to evaluate whether its scientific potential. In astrophysics, generalization to redshifts not seen during training is essential. We investigated this by:

1. Training and testing models at individual $z \geq 0$. This generally works very well, and the model has good accuracy as can be seen in Figure 2.
2. Training a general model by pooling training sets at $z \in \{0, 0.5, 1, 2\}$, and testing at $z \in \{0, 0.5, 1, 2\}$. Compared to training and testing at individual redshifts, we obtain similar precision at all redshifts.
3. Pooling training sets at $z \in \{0, 0.5, 1, 2\}$, and testing at $z \in \{0.25, 0.75, 1.5, 1.75\}$ where the model was *not* trained. Surprisingly, we obtain comparable bias and scatter to where the model was trained.

Although not explicitly shown, we also observed that if one extrapolates instead of interpolating between redshifts seen during training, the results are both biased and have high scatters, although they still have lower scatter than the SOTA methods.

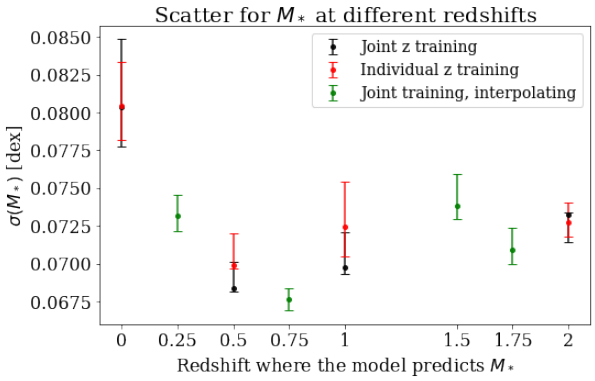

Figure 2. Median, 16th and 84th percentile of scatters from 10 models trained to predict only M_* at a series of redshifts in three different ways. Red, black, and green correspond to experiment 1, 2, and 3 in §4.2. The model successfully predicts M_* at unseen redshifts.

Features used	σ [dex]	$N_{features}$
All	0.0776	37
Only redshift	0.1704	1
None/Empty tree	0.2574	0
Only mass	0.1436	1
Only NFW[1] profile	0.1082	2
Only V_{max}	0.1194	1
Redshift and NFW profile	0.0993	3

Table 2. Results for feature ablation for predicting only M_*. Training on a smaller subset of features renders information about the importance of each subset. Interestingly, the empty tree regresses significantly better than abundance matching, demonstrating that there is significant information in the pure structure.

5. Feature Importance

Training and evaluating a GNN on our considered datasets can act as a metric for what features are most important in determining the properties of a galaxy. To explore this, we perform experiments where we only include certain sets of features during both training and testing of the GNN. Besides a series of physically motivated sets of quantities (see Table 2) from the literature (Rodríguez-Puebla et al., 2016), we also attempt to regress M_* from an **empty tree**, i.e., a tree with no features. The merger tree then contains no information but that encoded in the geometric structure itself. This approach gives less precise predictions than using the final halo only, but outperforms abundance matching.

6. Discussion

Although this paper shows that highly accurate mappings between dark matter merger trees and galactic properties exist, there is still significant scatter between the GNN and SAM M_{cold} and SFRs. It should, however, be noted that these quantities have high uncertainties from the SAM. To test if the GNN learns physically meaningful relationships, we analyze the interdependence of the target residuals. We find that the residuals between the two SFR targets and M_{cold} are strongly correlated (see Figure 5 in the appendix) i.e., if the GNN predicts a too high M_{cold}, it also predicts a too high SFR, analogous to the Kennicutt-Schmidt relation (Kennicutt, 1998). The improvement in these three quantities when exploiting the full merger history is smaller than expected, since they are thought to be strongly connected to the merger history of the galaxy (Somerville & Davé, 2015).

For M_* and M_{BH}, we observe a highly significant improvement when including merger history, as the reconstruction scatter is almost halved when including the merger tree, compared to using only the final halo.

Testing the median and dispersion relations between the

halo mass and target quantity, we observe that the GNN predictions not only reproduce the *median* relation, but also the *dispersion* (see Figure 6 in the appendix).

7. Conclusion

Using the full merger history, we greatly improve upon the current SOTA for learning baryonic physics and mapping it onto dark matter only simulations. The improvement from using the merger history is consistent across all features, although varying in strength. The model works at a range of redshifts and can reliably interpolate between redshifts, even if not trained on galaxies at a given redshift. The trained model is 4 orders of magnitude faster than the SC-SAM. Our code is publicly available at https://github.com/astrockragh/GraphMerge

References

Aarseth, S. J., Gott, J. R., I., and Turner, E. L. N-body simulations of galaxy clustering. I. Initial conditions and galaxy collapse times. ApJ, 228:664–683, March 1979. doi: 10.1086/156892.

Agarwal, S., Davé, R., and Bassett, B. A. Painting galaxies into dark matter haloes using machine learning. MNRAS, 478(3):3410–3422, August 2018. doi: 10.1093/mnras/sty1169.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *arXiv e-prints*, art. arXiv:1806.01261, June 2018.

Behroozi, P. S., Wechsler, R. H., and Wu, H.-Y. The ROCKSTAR Phase-space Temporal Halo Finder and the Velocity Offsets of Cluster Cores. ApJ, 762(2):109, January 2013a. doi: 10.1088/0004-637X/762/2/109.

Behroozi, P. S., Wechsler, R. H., Wu, H.-Y., Busha, M. T., Klypin, A. A., and Primack, J. R. Gravitationally Consistent Halo Catalogs and Merger Trees for Precision Cosmology. ApJ, 763(1):18, Jan 2013b. doi: 10.1088/0004-637X/763/1/18.

Benson, A. J. Galacticus: A semi-analytic model of galaxy formation. *New Astronomy*, 17(2):175–197, 2012. ISSN 1384-1076. doi: https://doi.org/10.1016/j.newast.2011.07.004. URL https://www.sciencedirect.com/science/article/pii/S1384107611000807.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, July 2017. doi: 10.1109/MSP.2017.2693418.

Contreras, S., Baugh, C. M., Norberg, P., and Padilla, N. The galaxy-dark matter halo connection: which galaxy properties are correlated with the host halo mass? MNRAS, 452(2):1861–1876, September 2015. doi: 10.1093/mnras/stv1438.

Cranmer, M. D., Xu, R., Battaglia, P., and Ho, S. Learning Symbolic Physics with Graph Networks. *arXiv e-prints*, art. arXiv:1909.05862, September 2019.

de Santi, N. S. M., Rodrigues, N. V. N., Montero-Dorta, A. D., Abramo, L. R., Tucci, B., and Artale, M. C. Mimicking the halo-galaxy connection using machine learning. *arXiv e-prints*, art. arXiv:2201.06054, January 2022.

Efstathiou, G., Davis, M., White, S. D. M., and Frenk, C. S. Numerical techniques for large cosmological N-body simulations. ApJS, 57:241–260, February 1985. doi: 10.1086/191003.

Foreman-Mackey, D. corner.py: Scatterplot matrices in python. *The Journal of Open Source Software*, 1(2): 24, jun 2016. doi: 10.21105/joss.00024. URL https://doi.org/10.21105/joss.00024.

Gabrielpillai, A., Somerville, R. S., Genel, S., Rodriguez-Gomez, V., Pandya, V., Yung, L. Y. A., and Hernquist, L. Galaxy Formation in the Santa Cruz semi-analytic model compared with IllustrisTNG – I. Galaxy scaling relations, dispersions, and residuals at z=0. *arXiv e-prints*, art. arXiv:2111.03077, November 2021.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv e-prints*, art. arXiv:1706.02216, June 2017.

Hearin, A. P., Zentner, A. R., van den Bosch, F. C., Campbell, D., and Tollerud, E. Introducing decorated HODs: modelling assembly bias in the galaxy-halo connection. MNRAS, 460(3):2552–2570, August 2016. doi: 10.1093/mnras/stw840.

Jo, Y. and Kim, J.-h. Machine-assisted semi-simulation model (MSSM): estimating galactic baryonic properties from their dark matter using a machine trained on hydrodynamic simulations. MNRAS, 489(3):3565–3581, November 2019. doi: 10.1093/mnras/stz2304.

Kamdar, H. M., Turk, M. J., and Brunner, R. J. Machine learning and cosmological simulations - I. Semi-analytical models. MNRAS, 455(1):642–658, January 2016. doi: 10.1093/mnras/stv2310.

Kennicutt, Robert C., J. The Global Schmidt Law in Star-forming Galaxies. ApJ, 498(2):541–552, May 1998. doi: 10.1086/305588.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks, 2017.

Kuhn, M. and Johnson, K. *Applied predictive modeling*, volume 26 of *statistics*. Springer, 2013.

Lacey, C. G., Baugh, C. M., Frenk, C. S., Benson, A. J., Bower, R. G., Cole, S., Gonzalez-Perez, V., Helly, J. C., Lagos, C. D. P., and Mitchell, P. D. A unified multiwavelength model of galaxy formation. MNRAS, 462(4):3854–3911, November 2016. doi: 10.1093/mnras/stw1888.

Lagos, C. d. P., Tobar, R. J., Robotham, A. S. G., Obreschkow, D., Mitchell, P. D., Power, C., and Elahi, P. J. Shark: introducing an open source, free, and flexible semi-analytic model of galaxy formation. MNRAS, 481 (3):3573–3603, December 2018. doi: 10.1093/mnras/sty2440.

Lovell, C. C., Wilkins, S. M., Thomas, P. A., Schaller, M., Baugh, C. M., Fabbian, G., and Bahé, Y. A machine learning approach to mapping baryons on to dark matter haloes using the EAGLE and C-EAGLE simulations. MNRAS, 509(4):5046–5061, February 2022. doi: 10.1093/mnras/stab3221.

Maksimova, N. A., Garrison, L. H., Eisenstein, D. J., Hadzhiyska, B., Bose, S., and Satterthwaite, T. P. ABACUSSUMMIT: a massive set of high-accuracy, high-resolution N-body simulations. MNRAS, 508(3):4017–4037, December 2021. doi: 10.1093/mnras/stab2484.

Naab, T. and Ostriker, J. P. Theoretical Challenges in Galaxy Formation. ARA&A, 55(1):59–109, August 2017. doi: 10.1146/annurev-astro-081913-040019.

Navarro, J. F., Frenk, C. S., and White, S. D. M. A Universal Density Profile from Hierarchical Clustering. ApJ, 490 (2):493–508, December 1997. doi: 10.1086/304888.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rodríguez-Puebla, A., Behroozi, P., Primack, J., Klypin, A., Lee, C., and Hellinger, D. Halo and subhalo demographics with Planck cosmological parameters: Bolshoi-Planck and MultiDark-Planck simulations. MNRAS, 462(1):893–916, October 2016. doi: 10.1093/mnras/stw1705.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. Nature, 323(6088):533–536, October 1986. doi: 10.1038/323533a0.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Sheth, R. K., Mo, H. J., and Tormen, G. Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes. MNRAS, 323(1):1–12, May 2001. doi: 10.1046/j.1365-8711.2001.04006.x.

Smith, L. N. and Topin, N. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

Somerville, R. S. and Davé, R. Physical Models of Galaxy Formation in a Cosmological Framework. ARA&A, 53:51–113, August 2015. doi: 10.1146/annurev-astro-082812-140951.

Somerville, R. S. and Primack, J. R. Semi-analytic modelling of galaxy formation: the local Universe. MNRAS, 310(4):1087–1110, December 1999. doi: 10.1046/j.1365-8711.1999.03032.x.

Somerville, R. S., Hopkins, P. F., Cox, T. J., Robertson, B. E., and Hernquist, L. A semi-analytic model for the co-evolution of galaxies, black holes and active galactic nuclei. MNRAS, 391(2):481–506, December 2008. doi: 10.1111/j.1365-2966.2008.13805.x.

Somerville, R. S., Popping, G., and Trager, S. C. Star formation in semi-analytic galaxy formation models with multiphase gas. MNRAS, 453(4):4337–4367, November 2015. doi: 10.1093/mnras/stv1877.

Vale, A. and Ostriker, J. P. Linking halo mass to galaxy luminosity. MNRAS, 353(1):189–200, September 2004. doi: 10.1111/j.1365-2966.2004.08059.x.

White, S. D. M. and Frenk, C. S. Galaxy Formation through Hierarchical Clustering. ApJ, 379:52, September 1991. doi: 10.1086/170483.

A. Further Notes on Data

To ensure that the galaxies in the dataset have reliable features and targets, we employ a set of selection criteria. First, only merger trees where the final halo has a mass of $10^{10} M_{\odot}$ or above are included. This choice is made as the mass of the final halo indicates both the reliability with which the dark matter properties can be measured as well as the reliability of the SAM baryonic properties derived from said dark matter properties. Secondly, only central galaxies are included, since central and satellite galaxies are believed to have different relationships with their host halos (Hearin et al., 2016).

In any given merger tree, there can be upwards of millions of nodes, some reductions are made. Since we are mainly interested in probing the merger history, we preserve nodes/halos that are either:

- A progenitor node, i.e., the first time a halo was detected in the simulation
- Pre-merger nodes, i.e., halos the snapshot before they merge
- Post-merger nodes, i.e., halos that are the direct result of a merger
- The final node, i.e., the final halo

This reduces the number of nodes by a factor of $\sim$10-50, depending on the merger tree in question. This, of course, produces a strong inductive bias, since smooth accretion modes are not included. We also limit the total number of nodes to be $< 2 \cdot 10^4$, which results in the exclusion of 107 merger trees. Since we regress logarithmic targets, only galaxies with non-zero target quantities are included (this excludes 470 trees). In total, the $z = 0$ dataset consists of 108,338 merger trees.

In the basic dataset, we include all dark matter features that are not IDs, x,y,z positions, or x,y,z velocities, even features not explicitly used by the SAM.

As outlined in Kuhn & Johnson (2013), it is important that the final model evaluation is made on data that is not used in either the training nor for optimizing hyperparameters. Therefore we here split our data in three groups, a training set, a validation set used for evaluating performance during hyperparameter tuning, and a test set for independently evaluating the performance of the final GNN. The test set is never used during training or hyperparameter optimization.

A 70/10/20 split is used. After optimizing the hyperparameters via the validation set, it is absorbed into the training set for the final training of the models before testing.

Since all hyper-parameter tuning is done at z = 0, only a training and testing set are constructed for predicting at $z > 0$. For training and testing at $z > 0$, it is important to keep in mind that most galaxies at any $z = z_1$ will be a progenitor of a galaxy at $z_2 < z_1$. Thus, if one were to naively train a model on baryonic quantities at both z_1 and z_2 with randomly chosen training and testing sets, there would be significant information leakage from the training to the test set.

Therefore, we first construct the $z = 0$ dataset according to the above prescription. Next, for a dataset at any $z_n > 0$, for every merger tree, we test if it contains any part of any merger tree in any dataset at a redshift lower than z_n. If it does, we assign it to the set which the descendant galaxy is part of. All merger trees not assigned to either set are then split such that the overall dataset at z_n has an 80/20 split between training and testing.

B. Model Architecture

Here we wish to describe the architecture a bit more in depth for the purpose of reproducibility. The architecture is visualized in Figure 3. The merger tree is passed through a 2-layer Multi-Layer Perceptron (MLP) to encode the node state before any graph convolutional layers. Then the encoded merger tree is passed through 5 GraphSAGE layers, each with a ReLU activation layer between. The encoded merger tree is then summed over with a global sum pooling. Using a global max pooling renders similar performance. Each of the targets then has its own 3-layer MLP decoder "head." "Heads" refers to different branches of the model that all take the same input, which allows each head to predict more independently of the others. Thus, each target/covariance component will have its own independent decoding function that is not influenced by the backpropagation from other predicted quantities, whereas the graph convolutional layers will all be influenced by all targets. If the uncorrelated Gaussian loss is used, no off-diagonal components of the covariance matrix are predicted, and $\hat{\Sigma}$ is diagonal and corresponding to just having the usual Gaussian uncertainties. The layer normalization description can be found in Ba et al. (2016). After the sequence of convolutional layers, a differentiable global pooling operator is applied

across all nodes in order to standardize the output size. The dimensionality of the latent space (known as the number of hidden states) was 128.

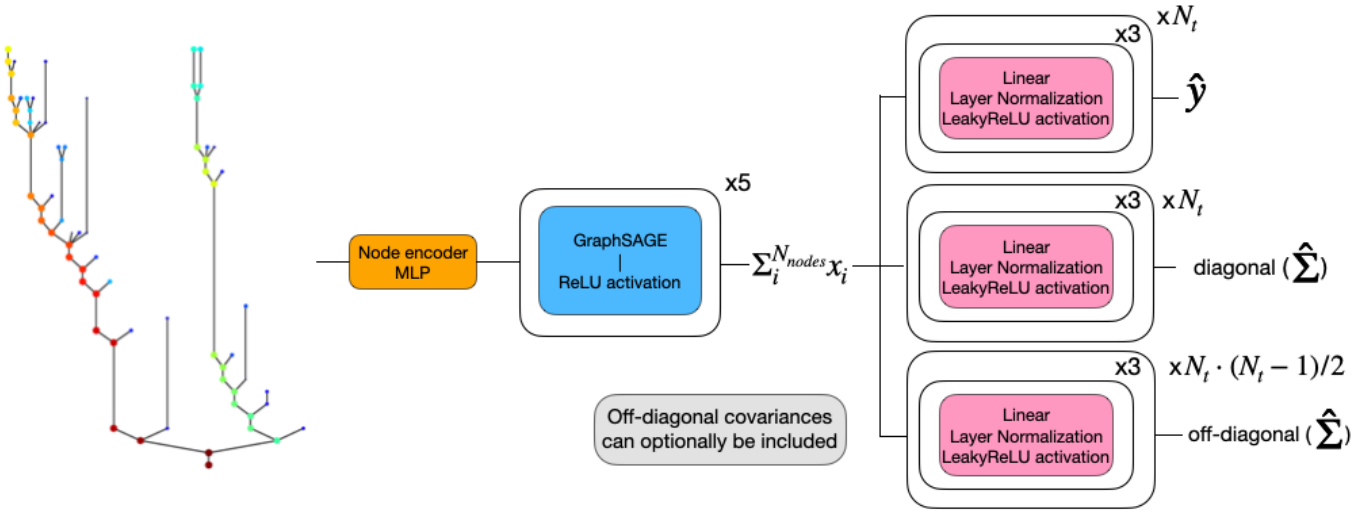

Figure 3. A diagram of the model for predicting values and the full-covariance matrix. N_t is the number of targets one wishes to regress. The number of times a given block is repeated is written by the upper right corner of the block. A linear layer is the same as a 1-layer MLP. The flow is from left to right, but inside each box the flow is from top to bottom. Each layer operates with 128 hidden states.

C. Training the Model

We train the models using the `Pytorch` OneCycleLR learning rate scheduler (Smith & Topin, 2018; Paszke et al., 2019), using a max learning rate of 10^{-2} and a batch size of 256 using the Adam optimizer (Kingma & Ba, 2017). The models were trained for 1000 epochs when optimized for all targets, and 500 for 2 targets or less, as this was determined during hyperparameter[2] optimization to be above the average number of epochs required for a model to converge. A Gaussian quantile transform [3], which maps each parameter to a Gaussian distribution defined by the quantiles of the parameter in question, was fit on the training set and applied to all input data before training, except for categorical data such as the number of progenitor halos or whether the halo had recently undergone a major merger, which is encoded as a boolean in the data. This makes training more stable at the risk of destroying some information. We also attempted using a standard scaler, which scales data to have zero mean and unity variance. This resulted in slightly higher scatters by about 3-5%.

We employ a max learning rate of 10^{-2}, a 15% start percentage and a final division factor of 10^3.

D. Core Concepts of Graphs

GNNs are a species of neural network which operates on graph-structured data (Scarselli et al., 2008; Bronstein et al., 2017; Battaglia et al., 2018). For our purpose, the graphs G on which GNNs operate is usually defined as 2-tuples, $G = (V, E)$,[4] where $V = \{\mathbf{v}_i\}_{i=1:N^v}$, where N^v is the total number of nodes, $\mathbf{v}_i \in \mathbb{R}^{D^v}$ is a set of node attribute vectors of dimensionality D^v and $E = \{(\mathbf{e}_k, r_k, s_k)\}$ is a set of edge attribute vectors $\mathbf{e}_k \in \mathbb{R}^{D^e}$ of dimensionality D^e, and indices $r_k, s_k \in \{1 : N^v\}$ of the "receiver" and "sender" nodes connected by the k-th edge.

In this work, only node attributes and edge indices are used. Note that our graphs are **directed**, since merger trees naturally

[2]The hyperparameters of the model and training scheme are defined as parameters not of the model itself, but about the model or training scheme. Examples include the dimensionality of the latent space, the number of layers and the learning rate.

[3]sklearn source code

[4]We adhere closely to the notation used in (Battaglia et al., 2018; Cranmer et al., 2019) for formal definitions.

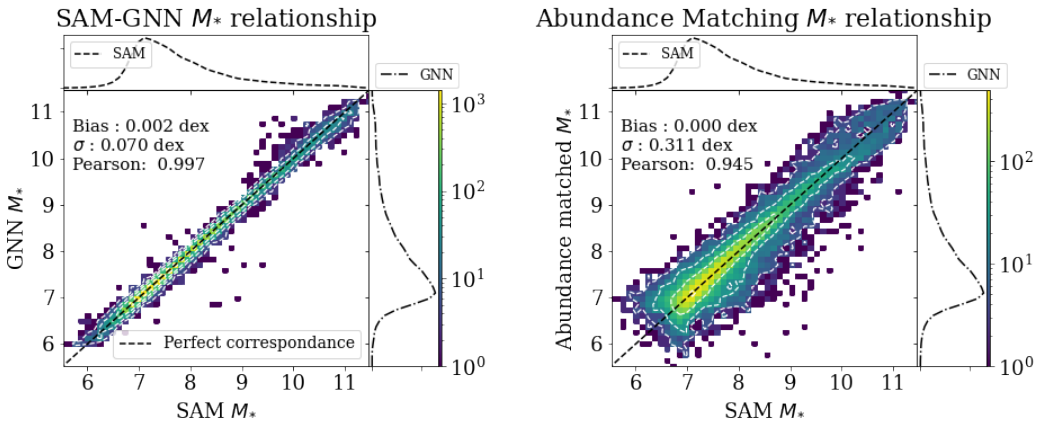

Figure 4. Histogram of the SAM M_* versus the model-predicted M_* with logarithmically colored bin heights. The left panel shows the target-prediction relation of the method presented in this paper, and the right panel shows target-prediction relation of the common abundance matching approach. As can be seen in Table 1, our results improve by a factor of two over the SOTA and is comparable to lower information limit of the SAM outputs.

are directed in time. A directed graph means that information can only be passed one way on a given edge, which for our purpose follows the flow of time since propagating information backwards in time would break causality.

The **neighborhood** of node i consists of all nodes that are connected to node i by an edge. Note that for a directed graph, this only includes the set of nodes for which $r_k = i$. Some prefer to instead define two separate notions of neighborhoods for directed graphs, an **incoming** neighborhood and an **outgoing** neighborhood. Our definition would be the same as the incoming neighborhood. We denote the neighborhood of node i by $\mathcal{N}(i)$.

E. Loss Function

The loss function $\mathcal{L}$ is central to the optimization and performance of the GNN, as the parameter set θ which make up the GNN is optimized to satisfy $min(\mathcal{L}_\theta(\{G\}_{train}))$. In this work we employ a generalized Gaussian Negative Log-Likelihood (NLL).

For a single input, the general Gaussian NLL is defined as:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}, \hat{\Sigma}) \equiv \frac{ln(|\hat{\Sigma}|)}{2} + \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^T \hat{\Sigma}^{-1} (\mathbf{y} - \hat{\mathbf{y}}) \tag{4}$$

where, $\mathbf{y}$ is the true target vector, $\hat{\mathbf{y}}$ is the network prediction vector, $\hat{\Sigma}$ is the predicted covariance matrix, and $|\hat{\Sigma}|$ denotes its determinant. These are then extended to batch form by summing over all inputs in a batch.

In this paper, the quoted results are obtained via a purely diagonal covariance matrix.

F. Stellar Mass Results

As the central quantity of interest, the stellar mass received the most attention. The test set results were a scatter of **0.070 dex**, with **0.002 dex** bias. This is shown in Figure 4, along with a comparison to the usual abundance matching approach.[5] Abundance matching (Vale & Ostriker, 2004), simply rank-orders all galaxies and halos by mass and assumes a monotonic matching relation exists between the two. We include this comparison as a baseline due to its simplicity and widespread use.

Figure 4 shows the relation between target value and predicted value, along with distributions on the respective axes. The

[5]Other metrics can be found in Table 1

figure shows the (target, prediction)-relation as a 2D histogram with logarithmic bin heights. If this relation follows the diagonal, that would indicate perfect predictions. The tighter the relation follows the diagonal, the better.

A few comparisons are beneficial to keep in mind:

- Training a GNN to predict only M_* yields a scatter of 0.078 dex, significantly worse than the performance when training a GNN to predict all quantities simultaneously.
- The performance of the GNN worsens to **0.132 dex** when using only the parameters of the final halo, indicating a strong dependence on assembly history.
- The scatter of the GNN M_* predictions is comparable to the SAM probabilistic limit as defined above, which renders a scatter of **0.043 dex** (see Table 1).

G. Further interpretation plots

Residual-residual plots are very useful for investigating the interdependence between predictions. Here we provide a plot to illustrate these interdependencies. Figure 5 shows residual-residual relations for the GNN relative to the SAM targets, along with the slope (a) and intercept (b) of a line fitted using least squares (not using the σ predicted by the model).
From this plot we clearly observe a strong interdependence between SFR- and SFR_{100}-residuals (as expected), positive correlations between M_*- and SFR / SFR_{100} / Z_{gas}-residuals, positive correlations between M_{cold} and SFR / SFR_{100}-residuals (analogous to a Kennicutt-Schmidt relation) and a negative correlation between M_{cold}- and Z_{gas}-residuals.

Halo mass-variable relations are also very useful for identifying the regions where the model fails to reproduce the SAM. This general comparison can be found in Figure 6. Here we quickly identify one of the reasons for the GNNs poor performance on SFR and SFR_{100}, namely that it does not successfully capture the two diverging branches of SFR around $M_{halo} \approx 11.7$, regressing only the lower branch accurately. We also observe that M_*, M_{cold}, Z_{gas} and M_{BH} generally follow both the median relation as well as reproducing the scatter. The scatter is not reproduced for the two SFR targets. This is a problem discussed in Agarwal et al. (2018), which our method also improves significantly upon.

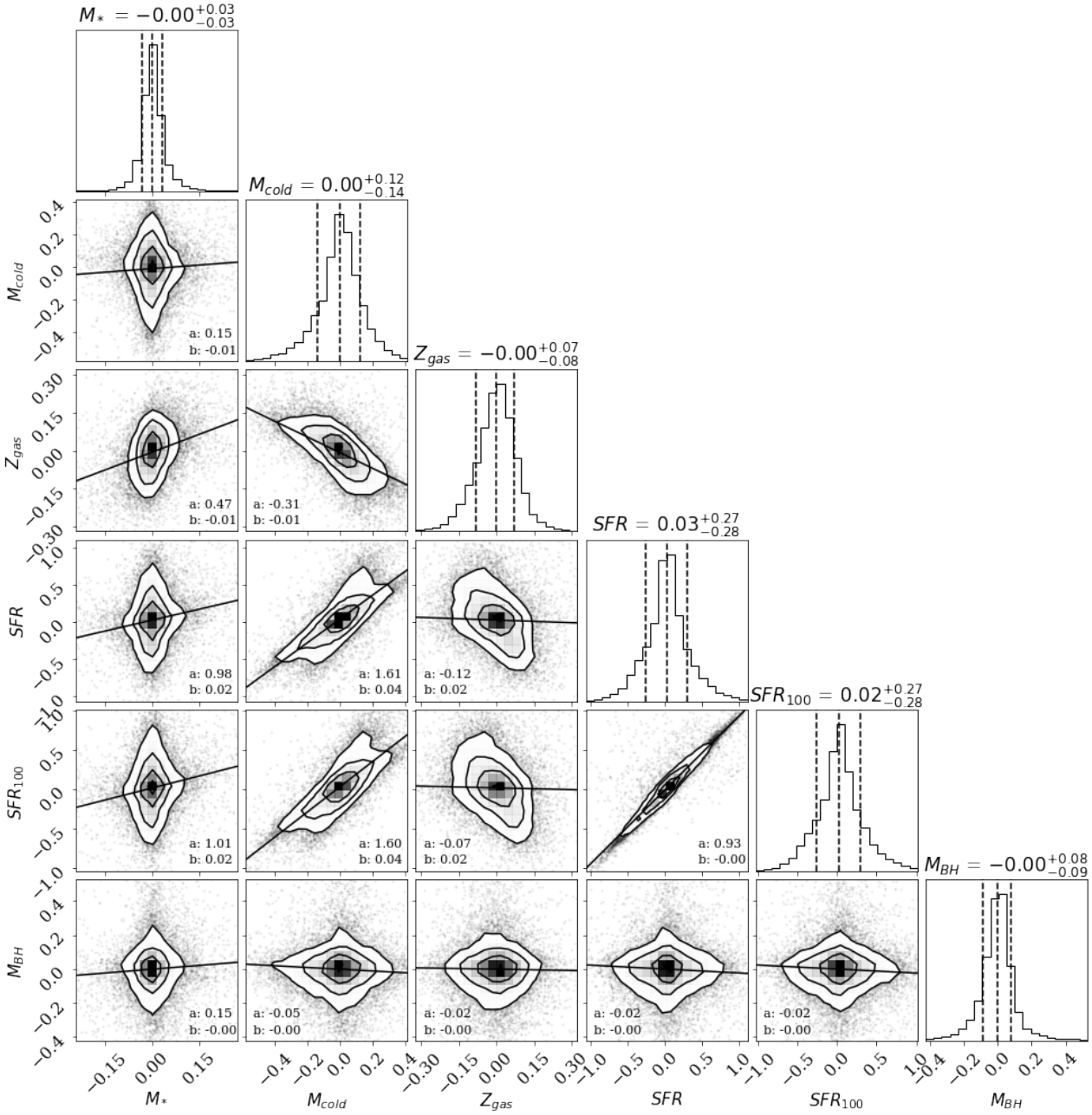

Figure 5. Residual for all targets, along with linear (a*x+b) fits. Each window is annotated with the slope (a) and the intercept (b) of the residual-residual relation in question. The plot is made with the `corner` package (Foreman-Mackey, 2016).

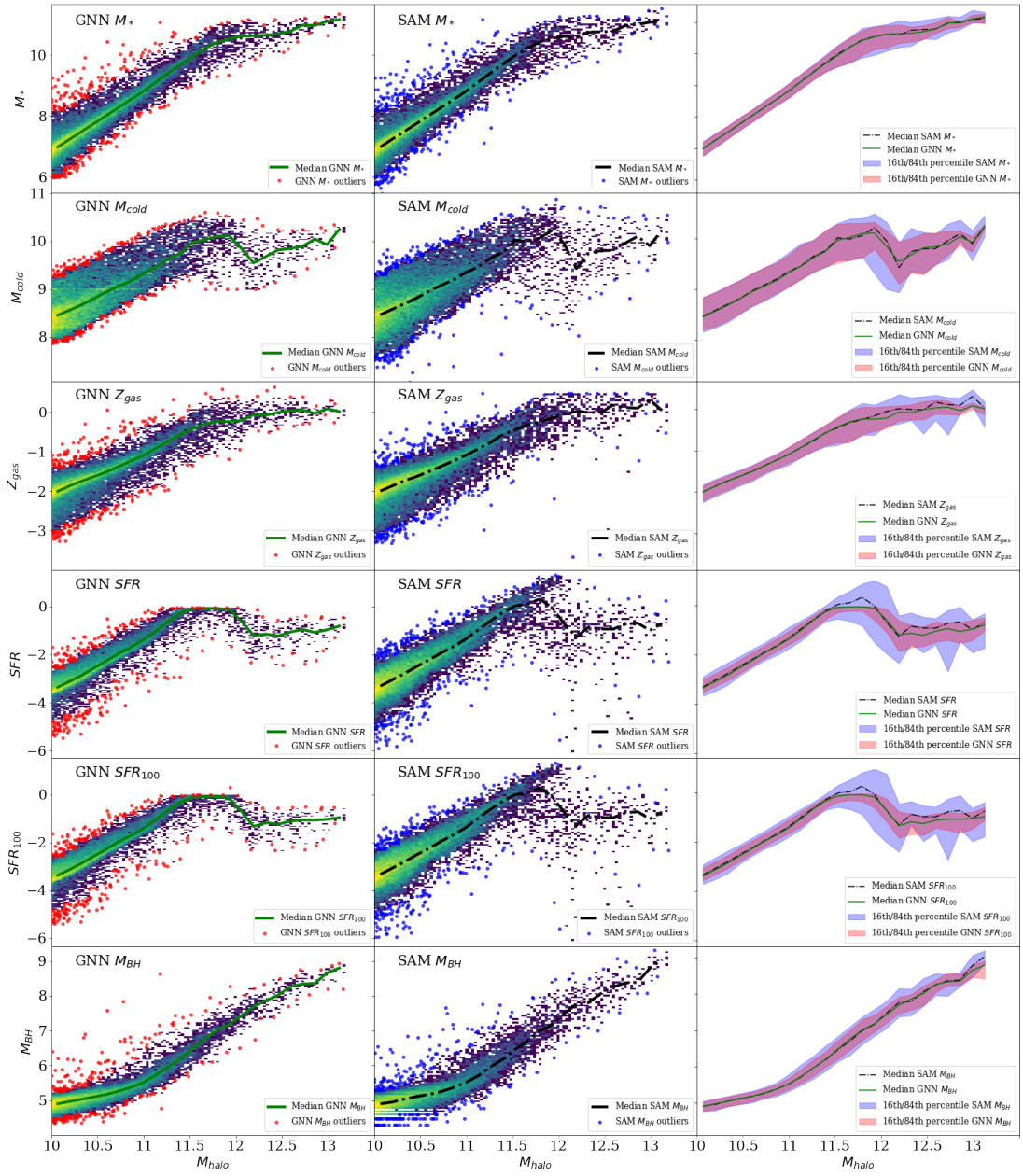

Figure 6. Relation between halo masses and target parameter for all targets for both the SAM and the GNN predictions, with outliers clearly marked. We furthermore show general trends in the right column, where the dashed and solid lines show the medians and the shaded areas show the 16 and 84th percentiles for the parameter in question for both the SAM and GNN. Here we immediately see the source of some of the errors, as for example, the inability of the GNN to accurately capture the two diverging branches in SFR.