Mass Estimation of Galaxy Clusters with Deep Learning. I. Sunyaev–Zel’dovich Effect

N. Gupta and C. L. Reichardt

School of Physics, University of Melbourne, Parkville, VIC 3010, Australia; nikhel.gupta@unimelb.edu.au

Received 2020 March 13; revised 2020 June 22; accepted 2020 July 14; published 2020 September 7

Abstract

We present a new application of deep learning to infer the masses of galaxy clusters directly from images of the microwave sky. Effectively, this is a novel approach to determining the scaling relation between a cluster’s Sunyaev–Zel’dovich (SZ) effect signal and mass. The deep-learning algorithm used is mResUNet, which is a modified feed-forward deep-learning algorithm that broadly combines residual learning, convolution layers with different dilation rates, image regression activation, and a U-Net framework. We train and test the deep-learning model using simulated images of the microwave sky that include signals from the cosmic microwave background, dusty and radio galaxies, and instrumental noise as well as the cluster’s own SZ signal. The simulated cluster sample covers the mass range $1 \times 10^{14} M_\odot < M_{200c} < 8 \times 10^{15} M_\odot$ at $z = 0.7$. The trained model estimates the cluster masses with a 1σ uncertainty $\Delta M/M \lesssim 0.2$, consistent with the input scatter on the SZ signal of 20%. We verify that the model works for realistic SZ profiles even when trained on azimuthally symmetric SZ profiles by using the Magnetico hydrodynamical simulations.

Unified Astronomy Thesaurus concepts: Galaxy clusters (584); Cosmic microwave background radiation (322); Convolutional neural networks (1938); Sunyaev-Zeldovich effect (1654); Large-scale structure of the universe (902); Sky surveys (1464)

1. Introduction

Galaxy clusters reside in the most massive gravitationally bound halos in the cosmic web of large-scale structure (LSS) and can be observed across the electromagnetic spectrum. In recent years, the Sunyaev–Zel’dovich (SZ) effect (Sunyaev & Zel’dovich 1970, 1972), the inverse-Compton scattering of the cosmic microwave background (CMB) photons by the energetic electrons in the intracluster medium, has emerged as a powerful tool to detect galaxy clusters in the millimeter wavelength sky. Since Staniszewski et al. (2009) presented the first SZ-discovered clusters, the South Pole Telescope (SPT; Carlstrom et al. 2011), the Atacama Cosmology Telescope (ACT; Fowler et al. 2007), and the Planck satellite (The Planck Collaboration 2006) have released catalogs of hundreds to thousands of newly discovered clusters (e.g., Planck Collaboration et al. 2016; Hilton et al. 2018; Bleem et al. 2020; Huang et al. 2020). These cluster samples are significant because the abundance of galaxy clusters is one of the most promising avenues to constrain different cosmological models (e.g., Mantz et al. 2008; Vikhlinin et al. 2009; Hasselfield et al. 2013; de Haan et al. 2016; Planck Collaboration et al. 2016; Bocquet et al. 2019).

With ongoing (e.g., SPT-3G, AdvancedACT, Benson et al. 2014; Henderson et al. 2016) and upcoming (e.g., Simons Observatory, CMB-S4, Abazajian et al. 2019; Ade et al. 2019) CMB surveys, we expect to detect $>10^4$ galaxy clusters. These cluster samples could have a ground-breaking impact on our understanding of the expansion history and structure growth in the universe, but only if we can improve the calibration of cluster masses (see, e.g., Bocquet et al. 2015; Planck Collaboration et al. 2015).

Observationally, several techniques have been used to measure the masses of galaxy clusters, such as optical weak lensing (e.g., Johnston et al. 2007; Gruen et al. 2014; Hoekstra et al. 2015; McClintock et al. 2019; Stern et al. 2019), CMB lensing (e.g., Baxter et al. 2015; Madhavacheril et al. 2015; Planck Collaboration et al. 2016; Raghunathan et al. 2019), and dynamical mass measurements (e.g., Biviano et al. 2013; Sifón et al. 2016; Capasso et al. 2019). These techniques are typically used to calibrate the scaling relationship between mass and an easily measurable observable such as the richness or SZ signal (e.g., Sifón et al. 2013; Mantz et al. 2016; Stern et al. 2019). The latter is particularly interesting as numerical simulations have shown that the integrated SZ signal is tightly correlated with the mass of clusters (e.g., Gupta et al. 2017; le Brun et al. 2017).

In recent years, deep learning has emerged as a powerful technique in computer vision. In this work, we demonstrate the first use of a deep-learning network to estimate the mass of galaxy clusters from a millimeter wavelength image of the cluster. We employ a modified version of a feed-forward deep-learning algorithm, mResUNet that combines residual learning (He et al. 2015) and U-Net framework (Ronneberger et al. 2015). We train the deep-learning algorithm with a set of simulations that include the cluster’s SZ signal added to Gaussian random realizations of the CMB, astrophysical foregrounds, and instrumental noise. We use the trained mResUNet model to infer the mass from a test data set, which is not used in the training process. We also test the accuracy of the trained model using hydrodynamical simulations of galaxy clusters, which again are not used in the training process.

The paper is structured as follows. In Section 2, we describe the deep-learning reconstruction model and the microwave sky simulation data. In Section 3, we describe the optimization process and the relevant hyperparameters of the deep-learning model. In Section 4, we present mass estimations using the images from test data sets as well as the images from the external hydrodynamical simulations of SZ clusters. Finally, in Section 5, we summarize our findings and discuss future prospects.

Throughout this paper, $M_{200c}$ is defined as the mass of the cluster within the region where the average mass density is 200 times the critical density of the universe. The central mass and the 1σ uncertainty is calculated as the median and half of the difference between the 16th and 84th percentile masses, respectively.
2. Methods

In this section, we first describe the deep-learning algorithm, and then present the microwave sky simulations that are used to train and test the deep-learning model.

2.1. Deep-learning Model

In recent years, deep-learning algorithms have been extensively used in a range of astrophysical and cosmological problems (e.g., George & Huerta 2018; Mathuriya et al. 2018; Alexander et al. 2020; Allen et al. 2019; Bottrell et al. 2019; Fluri et al. 2019). Recent studies have applied deep-learning (Ho et al. 2019; Ntampaka et al. 2019) and machine-learning (e.g., Armitage et al. 2019; Green et al. 2019; Ntampaka et al. 2015) algorithms to estimate galaxy cluster masses using mock X-ray and velocity dispersion observations. These studies found that these techniques produce more accurate X-ray and dynamical mass estimates than conventional methods.

In this work, we apply the mResUNet algorithm to extract the SZ profiles and the cluster masses from the simulated microwave sky maps. ResUNet is a feed-forward deep-learning algorithm that was first introduced for the segmentation of medical images (Kayalibay et al. 2017) and to extract roads from maps (Zhang et al. 2018), and it was later applied to a number of problems. The original algorithm was modified by Caldeira et al. (2019) to perform image-to-image regression, i.e., to get an output image that is a continuous function of the input image. We implement further modifications to the network to extract small- and large-scale features in the map. This modified ResUNet, or mResUNet, algorithm is well suited to astrophysical problems, such as the current use case of estimating the SZ signal from an image of the sky.

The mResUNet is a convolutional neural network and its basic building block is a convolution layer that performs discrete convolutions (see Gu et al. 2015 for a recent review). The aim of the convolution layer is to learn features of an input map. Convolutional neural networks assume that nearby pixels are more strongly correlated than the distant ones. The features of nearby pixels are extracted using filters that are applied to a set of neighboring pixels. This set of neighboring pixels is also called the receptive field. The filter applied to a set of pixels is typically a $k \times k$ array with $k = 1, 3, 5, ...$, and the size of the filter $k$ is denoted as the kernel size. A filter with a given kernel size is moved across the image from top left to bottom right and at each point in the image a convolution operation is performed to generate an output. Several such filters are used in a convolution layer to extract information about different aspects of the input image. For instance, one filter can be associated with the central region of the galaxy cluster and the rest of the filters could extract information from the other parts of the cluster. The filters can extract information across different length scales by using different dilation rates instead of increasing the kernel size. A dilation rate of $N$ stretches the receptive field by $k + (k - 1) (N - 1)$, thus doubling the dilation rate will increase the receptive field to $5 \times 5$ for $k = 3$. These dilated convolutions systematically aggregate multiscale contextual information without losing resolution (Yu & Koltun 2015).

The total receptive field increases for each pixel of the input image as we stack several convolution layers in the network. An activation function is applied after each convolution layer, which is desirable to detect nonlinear features and results in a highly nonlinear reconstruction of the input image (see Nwankpa et al. 2018 for a recent review). Each convolution layer produces a feature map for a given input image. The feature map $f_l$ for a convolution layer $(l)$ is obtained by convolving the input from a previous layer $(x_{l-1})$ with a learned kernel, such that, the feature value at location $(i, j)$ is written as

$$f_l^{ij} = w_l^T x_{l-1}^{ij} + b_l,$$

where $w_l$ is the weight vector and $b_l$ is the bias term. The weights are optimized using gradient descent (e.g., Ruder 2016) that involves back-propagation from the final output, back to each layer in reverse order to update the weights.

The mResUNet architecture used in this work has the following main components.

1. We base our architecture on the encoder–decoder paradigm. This consists of a contracting path (encoder) to capture features, a symmetric expanding path (decoder) that enables precise localization and a bridge between these two. Figure 1 shows the full UNet framework, where the red and green dashed lines point to encoding and decoding frameworks, respectively.

2. Each gray colored box corresponds to a convolution block. We increase the filter size from 64 to 512 and use strides (e.g., Dumoulin & Visin 2016) to reduce the size of the feature map by half whenever the filter size is doubled (red arrows) during the encoding phase of the network. This process is known as down sampling by striding. For the decoding phase, we increase the size of the feature map by up sampling (green arrows). Each convolution block has four substages where convolution operations are applied with different dilation rates of $N = 1, 2, 3,$ and $4$, while keeping the stride length at unity, whenever the dilation rate is not 1. This improves the performance by identifying correlations between different locations in the image (e.g., Yu & Koltun 2015; Chen et al. 2016, 2017).

3. The feature maps from two substages (dilation rates $N = 2, 4$) of the first three encoding convolution blocks are cross-concatenated with the corresponding maps from decoding blocks using skip connections. These connections are useful to retrieve the spatial information lost due to striding operations (e.g., Drozdzal et al. 2016).

4. Each substage of encoding and decoding convolution blocks has a fixed number of layers. Among these the convolution, the activation, and the batch normalization layers are present in all substages. The batch normalization layer is helpful in improving the speed, stability, and performance of the network (Ioffe & Szegedy 2015). The input to these layers is always added to its output, as shown by the connection between input and addition layers. Such connections are called residual connections (He et al. 2015) and they are known to improve the performance of the network (e.g., Zhang et al. 2018; Caldeira et al. 2019).

5. A large feed-forward neural network when trained on a small set of data, typically performs poorly on the test data due to overfitting. This problem can be reduced by randomly omitting some of the features during the training phase by adding dropout layers to the network (Hinton et al. 2012). We add dropout layers to the decoding phase of the network.
In this section, we describe the microwave sky simulations of SZ clusters. We create 19 distinct sets of simulations for galaxy clusters with \( M_{200c} = (0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10) \times 10^{14} M_\odot \) at \( z = 0.7 \). For each mass, we create 800 simulated 10' \( \times \) 10' sky images, centered on the cluster with a pixel resolution of 0.25. While upcoming CMB surveys (see Section 1) will observe the microwave sky at multiple frequencies, we make the simplifying assumption in this work to focus on single-frequency maps at 150 GHz. The sky images include realizations of the CMB, white noise, SZ effect, cosmic infrared background (CIB) and radio galaxies. The CMB power spectrum is taken to be the lensed CMB power spectrum calculated by CAMB\(^1\) (Lewis et al. 2000) for the best-fit Planck \( \Lambda \)CDM parameters (Planck Collaboration et al. 2018). The foreground terms, the thermal and kinematic SZ effect from unrelated halos, CIB and radio galaxies, are taken from George et al. (2015). We assume the instrumental noise is white with a level of 5 \( \mu \)K arcmin, similar to what was achieved by the SPTpol survey (Henning et al. 2018). Note that these simulations neglect non-Gaussianity in the astrophysical foregrounds, as well as gravitational lensing of the CMB by large-scale structure besides the cluster itself. Future work should assess the impact of these sources of non-Gaussianity on the deep-learning estimator.

We assume the cluster’s own SZ signal follows the Generalized Navarro–Frenk–White (GNFW; Nagai et al. 2007) pressure profile, with parameters as a function of mass and redshift taken from the best-fit values in Arnaud et al. (2010). In addition, unless noted, we add a 20% log-normal scatter on the modeled amplitude of the SZ signal. This is slightly larger than the amount of scatter (0.16) found in the calibration of scaling relations using a light cone from large hydrodynamical simulations (e.g., Gupta et al. 2017), and thus conservative.

We convolve these maps with a 1' Gaussian beam, which is consistent with ground based SPT and ACT experiments at 150 GHz, and apply apodization. One of these cluster cutouts is shown in Figure 2 for \( M_{200c} = 5 \times 10^{14} M_\odot \) and a random CMB realization. In addition to these microwave sky SZ cluster maps, we save the corresponding SZ profiles and the mass of clusters that are used as labels in the training process. In order to recover masses from a framework designed to recover images, we set the central pixel value of the “mass map” to be proportional to the cluster mass. We then extract this central pixel value when reporting the recovered mass constraints.

2.3. Uncertainties in the SZ–Mass Scaling Relation

The deep-learning model in this work is trained on a specific SZ–mass scaling relation, here chosen to be the Arnaud model. Of course, we have imperfect knowledge of the relationship between a typical cluster’s SZ flux and mass. Recent measurements of the SZ–mass scaling relation are uncertain at the O(20%) level (Bocquet et al. 2019; Dietrich et al. 2019). This uncertainty is a fundamental limit to how well methods like this one that estimate cluster masses from the SZ signal can perform. However, this uncertainty can be reduced by calibrating the relationship on

\(^1\)https://camb.info/
samples of clusters using weak gravitational lensing (e.g., Corless & King 2009; Becker & Kravtsov 2011). Several programs employing gravitational lensing that are currently underway (e.g., Dark Energy Survey, Hyper Suprime-Cam Survey McClintock et al. 2019; Murata et al. 2019) or expected to start observing in the near future (e.g., LSST, Euclid LSST Science Collaboration et al. 2009; Laureijs et al. 2011), will lead to much tighter constraints on the SZ–mass scaling relation. In this paper, we test the deep-learning model on the simulated sky maps with SZ profiles taken from the Arnaud scaling relation and from the hydrodynamical simulations with a different intrinsic SZ–mass scaling relation.

3. Training and Optimization

The mResUNet model described in Section 2.1 and Figure 1 takes images as input and outputs the same sized images after passing through several convolutional blocks. This process is repeated for a number of epochs, where one epoch is when all of the training data are passed through the neural network once. The data are divided into three parts: training, validation, and test sets.

The training data set includes images of the microwave sky simulations of SZ clusters, the corresponding true SZ profiles and the true mass of clusters. As described in Section 2.2, both CMB maps and SZ profiles have a characteristic 20% log-normal SZ–mass scatter and all CMB maps have Gaussian random realizations of CMB. To make these simulations more realistic, we add foregrounds, 5 μK arcmin white noise, and 1′ beam smoothing to these maps. We normalize all maps, so that, the minimum and maximum pixel value is between −1 and 1, respectively, to improve the performance of the network. This is done by dividing the image pixels by a constant factor across all cluster masses. Our training data has 400 maps for each cluster mass and the true mass of clusters as labels to train the model. A different set of simulations is created for testing the model and the trained model is then used to predict the SZ profiles and the mass of clusters directly from the CMB maps of the testing set.

The work flow from simulations to mass estimations. Left panel shows an example of the microwave sky CMB map with the SZ imprint of a cluster with $M_{200c} = 5 \times 10^{14} M_\odot$, at $z = 0.7$. This map includes 5 μK arcmin white noise, foreground power estimates from George et al. (2015) and is smoothed by a 1′ beam. Several such maps for different cluster masses are used for training and validation of the neural network. Right panel shows SZ profile computed using the best-fit GNFW profile and the mass-observable scaling relation in Arnaud et al. (2010). In addition to microwave sky maps, the training set includes the true SZ profiles and the true mass of clusters as labels to train the model. A different set of simulations is created for testing the model and the trained model is then used to predict the SZ profiles and the mass of clusters directly from the CMB maps of the testing set.

The mResUNet model described in Section 2.1 and Figure 1 takes images as input and outputs the same sized images after passing through several convolutional blocks. This process is repeated for a number of epochs, where one epoch is when all of the training data are passed through the neural network once. The data are divided into three parts: training, validation, and test sets.

The training data set includes images of the microwave sky simulations of SZ clusters, the corresponding true SZ profiles and the true mass of clusters. As described in Section 2.2, both CMB maps and SZ profiles have a characteristic 20% log-normal SZ–mass scatter and all CMB maps have Gaussian random realizations of CMB. To make these simulations more realistic, we add foregrounds, 5 μK arcmin white noise, and 1′ beam smoothing to these maps. We normalize all maps, so that, the minimum and maximum pixel value is between −1 and 1, respectively, to improve the performance of the network. This is done by dividing the image pixels by a constant factor across all cluster masses. Our training data has 400 maps for each cluster mass and the true mass of clusters. For training, we only take cluster simulations with $M_{200c} = (1, 2, 3, 4, 5, 6, 7, 8) \times 10^{14} M_\odot$ and leave others for testing the model.

The validation set has same properties as the training set and is also used in the training phase to validate the model after each epoch. This is helpful as a nonlinear model is more likely to get high accuracy and over-fit when trained with training data only. Such a model gives poor performance with the test data. The validation of the model after every epoch ensures that the network is not overfitting and is useful to tune the model weights. We use 200 maps for each cluster mass and corresponding labels as our validation data.

The test data sets are never used in the training phase and are kept separately to analyze the trained model. We keep 200 CMB temperature maps and corresponding labels for testing. In addition to the cluster $M_{200c}$, used in training, we test our model for cluster masses that were not part of the training or validation process, that is, clusters with $M_{200c} = (0.5, 0.75, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 9, 10) \times 10^{14} M_\odot$.

The maps from the training set are passed through the neural networks with a batch size of 4 and a training loss is computed as mean-squared-error (MSE) between the predicted and the true labels after each batch. Batch after batch, the weights of the network are updated using the gradient descent and the back-propagation (see Section 2.1). In this work, we use the Adam optimizer (an algorithm for first-order gradient-based optimization, see Kingma & Ba (2014)) with an initial learning rate of 0.001. After each epoch, the validation loss (or validation MSE) is calculated and we change the learning rate by implementing callbacks during the training, such that, the learning rate is reduced to half if the validation loss does not improve for five consecutive epochs. In addition, to avoid overfitting, we set a dropout rate of 0.3 in the encoding phase of the network. We consider the network to be trained and stop the training process, if the validation loss does not improve for 15 epochs.

Every convolution block in the encoding, bridging, and decoding phase has a convolution layer, an activation layer and a batch normalization layer. The kernel size of each convolution layer is set to 3 × 3 and we change stride length from 1 to 2, whenever filter size is doubled. All activation layers in the network have Scale Exponential Linear Unit (Klambauer et al. 2017) activation functions that induce self-normalizing properties, such that, activations close to zero mean and unit variance converge towards zero mean and unit variance, when propagated through many network layers, even under the presence of noise and perturbations. Only for the final layer, a linear (or identity) activation function is used to get the same sized output images as inputs. The network has approximately 16 million parameters.
parameters and is trained on a single GPU using Keras with a TensorFlow backend.

4. Results

We now look at the performance of the trained deep-learning model on the test data. We test the performance of the trained model in three regimes: (i) cluster masses within the trained mass range (i.e., interpolation); (ii) cluster masses outside the trained mass range (i.e., extrapolation); and (iii) more realistic SZ clusters drawn from a large hydrodynamical simulation, the Magneticum Pathfinder Simulation\(^2\) (MPS). We find the model performs well in the first and third cases, but fails in the extrapolation case.

4.1. Predictions with Trained Cluster Mass

We use the test data with 200 CMB maps for each of the clusters with \(M_{200c} = (1, 2, 3, 4, 5, 6, 7, 8) \times 10^{14} M_\odot\). This testing mass is the same as that used in the training of our mResUNet model. These test maps are not used in training and validation phases and are distinct due to the Gaussian random realizations of the CMB and foregrounds as well as the 20% log-normal scatter in the estimation of the SZ signal. The trained model predicts SZ profiles as well as the mass of clusters from the CMB maps. The first column in Figure 3 shows examples of the input CMB temperature maps for clusters with \(M_{200c} = (2, 4, 6) \times 10^{14} M_\odot\), from top to bottom. The second and third columns show true and predicted mean SZ profiles, respectively, for 200 test maps. The last column shows residual signals, that is, the difference between the true and the predicted mean SZ profiles. This demonstrates that the deep-learning model reconstructs SZ profiles with a high accuracy, such that, the residual signal is at least two orders of magnitude smaller than the true SZ signal.

We simultaneously estimate the mass of galaxy clusters using the trained model. As described in Section 2.2, this is done by multiplying the central pixel of the predicted normalized NFW profiles by the mean mass of the training sample. The top panel in Figure 4 shows the estimated mass of clusters as a function of their true mass (green data points). This demonstrates that our trained mResUNet model can estimate cluster masses with high accuracy. For instance, we find \(M_{200c}^{\text{est}} = (1.99 \pm 0.40) \times 10^{14} M_\odot\) for a cluster with \(M_{200c}^{\text{true}} = 2 \times 10^{14} M_\odot\) and \(\Delta M/M \leq 0.2\) for all cluster masses. The bottom panel shows the ratio of the difference between the estimated and the true mass of clusters to the estimated uncertainty. This indicates that the mass estimations with our trained neural network model are consistent with the input mass at the 1\(\sigma\) level.

4.2. Predictions with Interpolated and Extrapolated Cluster Mass

In this section, we present the mass estimations using the test maps for clusters with untrained masses. We divide these samples into two types, that is, interpolated and extrapolated cluster masses. The first type of clusters lie within the mass range of the trained cluster sample with \(M_{200c}^{\text{true}} = (1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 9, 10) \times 10^{14} M_\odot\), and the second type of clusters are out of the training mass range with \(M_{200c}^{\text{true}} = (0.5, 0.75, 9, 10) \times 10^{14} M_\odot\). As before, white noise and 20% log-normal scatter is added to the SZ signal, and these maps are smoothed by a 1’ beam as well.

The top panel in Figure 4 shows the estimated and the true mass for interpolated (blue) and extrapolated (red) test data sets. The bottom panel shows the ratio of the difference between estimated and true mass of clusters to the estimated uncertainty. The 1\(\sigma\) error in the mass estimation for interpolated clusters is consistent with the true input mass. The uncertainties are similar to those from the trained sample (Section 4.1), for instance, the \(M_{200c}^{\text{est}} = (3.52 \pm 0.61) \times 10^{14} M_\odot\) for a cluster with \(M_{200c}^{\text{true}} = 3.5 \times 10^{14} M_\odot\). The \(\Delta M/M \leq 0.21\) for all cluster masses, except

---

\(^{2}\) http://www.magneticum.org/
The cluster masses are distributed across the training range 

\( \Delta M \) \( \pm 0.013 \) for simulations with 20\% intrinsic SZ scatter, and 0.100 \pm 0.012 for the no-scatter simulations. The apparent small reduction in scatter in the first case is consistent with a statistical fluctuation at 1.5\( \sigma \). These results clearly demonstrate that the deep-learning method to estimate cluster masses from the SZ signal has reached the theoretical lower limit set by the intrinsic SZ scatter.

A secondary implication of this result is that although upcoming CMB surveys with multiple observing frequencies and lower noise levels will yield higher fidelity measurements of the cluster SZ signal, this improvement may not translate to better mass estimates. Nevertheless, we plan to consider the impact of multiple frequency maps on the deep-learning analysis in future work.

4.4. Testing Model with External Hydrodynamical Simulations

In this section, we present our trained mResUNet model predictions for test images from the MPS, a large hydrodynamical simulation carried out as a counterpart to ongoing, multiwavelength surveys. The details about the simulations are discussed elsewhere (e.g., Dolag et al. 2016; Gupta et al. 2017; Soergel et al. 2018), and here we briefly summarize the most relevant features used in this work.

We use the two-dimensional Compton-\( \gamma \) map created by applying the so-called “gather approximation” with the SPH kernel (Monaghan & Lattanzio 1985; Dolag et al. 2005), where all gas particles that project into the target pixel contribute to the total \( \gamma \). The projection effects due to the uncorrelated line-of-sight structures are added by constructing four light cones from random selected slices without rotating the simulation box. Each light cone is a stack of 27 slices extracted from the simulation box at different redshifts. We use these light cones to extract cutouts of 95 galaxy clusters at \( z = 0.67 \) and \( z = 0.73 \) with 2 \( \times 10^{14} \) \( M_\odot \) \( < M_{200} < 7 \times 10^{14} \) \( M_\odot \). These cutouts have a resolution of \( \sim 0.2 \)
per pixel and we increase it to 0.25 to match with the pixel size of our training sample. The cluster catalog for these light cones has masses defined as $M_{500c}$, that is, the mass within the region where the average mass density is 500 times the critical density of the universe. We change this to $M_{200c}$ using a model of concentration-mass relation given by Diemer & Kravtsov (2015). We change the Compton-$y$ maps to temperature maps at 150 GHz and add them to the random realizations of CMB as well as foregrounds as described in Section 2.2. Similar to training and validation samples, we add 5 $\mu$K arcmin white noise and convolve these maps with a 1′ telescope beam.

Since the SZ–mass scaling relation used in training the deep-learning model is different than that found in the MPS simulation (Gupta et al. 2017), we should not expect the deep-learning model to recover unbiased masses for the MPS simulation. As discussed in Section 2.3, uncertainty in the SZ–mass scaling relation poses a fundamental limit to how accurately masses can be recovered from the SZ flux. This limit will improve as future lensing surveys improve our knowledge of the relationship. The interesting question to test with the MPS simulations is not whether the method is sensitive to the SZ–mass scaling relation (it is), but whether the deep-learning technique can recover masses from more realistic SZ signals when trained on the simple Arnaud profile.

Thus, we rescale the estimated masses based on the scaling relation differences. Specifically, we scale the mass of each cluster by the factor, $r$

$$r = \frac{F_{\text{Ar}}(G_{\text{MPS}}(M, z), z = 0.7)}{M},$$

where $Y = G_{\text{MPS}}(M, z)$ is the function describing the expected $Y$ for a cluster of a given mass and redshift in the MPS simulation, and $M = F_{\text{Ar}}(Y, z)$ the inverse function for the Arnaud scaling relation used in training the model. The redshift is fixed to $z = 0.7$ as in the training set. Recall that the redshift in MPS is restricted to the narrow range $z \in [0.67, 0.73]$. The SZ scaling relation in the MPS is taken from Table 4 in Gupta et al. (2017). The reported uncertainties on the scaling relation parameters in that work are small and only lead to a small 1.7% scatter in this factor (which we neglect). A caveat is that, since that work only reports the $Y_{500c} - M_{500c}$ scaling relation, we are adjusting the $M_{200c}$ results in this work by the expected $M_{500c}$ mass ratios. We scale the masses estimated by the deep-learning model by this factor $r$ to get rescaled mass estimates:

$$M_{200c}^{\text{est, scal}} = r M_{200c}^{\text{est}},$$

The mean $r$ over the set of MPS clusters used is 1.287.

The top panel of Figure 6 shows the scaled mass estimate plotted against the true mass of the 95 MPS galaxy clusters. The error bars are estimated by looking at the scatter across 100 realizations of the CMB and foregrounds that are added to the SZ signal of each cluster. The bottom panel of Figure 6 shows the distribution of the logarithm of the ratio of the scaled mass estimate to the true mass (solid green line). As in the previous section, we fit a Gaussian function to this distribution. We find the mean is $0.013 \pm 0.011$, consistent with zero, i.e., no mass bias. This argues that the method can accurately recover the mass from realistic SZ profiles even when the deep-learning model is trained on simpler axisymmetric profiles.

In Section 4.3, we showed that the uncertainty in the recovered mass was dominated by the intrinsic scatter in the SZ–mass scaling relation. We now check if this is still true for the more realistic SZ profiles in the MPS simulations. As in Section 4.3, we would like to compare the log-normal scatter in the scaled mass estimate to the intrinsic scatter in the MPS simulation. For the former, the Gaussian fit to the bottom panel of Figure 6 has a width $\sigma = 0.232 \pm 0.018$. For the latter, Gupta et al. (2017) found an intrinsic log-normal scatter of

---

3 We note that Gupta et al. (2017) refers to “cyl” by “lc.”
0.159 ± 0.002 in the $Y_{500c}^{\text{cyl}}$–$M_{500c}$ scaling relation. Unfortunately, that work did not look at the scaling between $Y_{200c}^{\text{cyl}}$ and $M_{200c}$. However, they did report that the scatter within $R_{500c}$ is a factor of 1.22 times larger than the scatter within $R_{500c}$ for the spherical $Y$ quantities (Table 3 in Gupta et al. 2017). Assuming that the same factor is valid for the cylindrical quantities, at $0.232 ± 0.018$, the scatter in the estimated mass is only slightly larger than the intrinsic scatter of $0.194 ± 0.002$ in the simulation, with the shift marginally detected at $2.1\sigma$ level. The performance of the deep-learning method appears limited by the intrinsic scatter in the SZ flux.

### 5. Conclusions

We estimate masses of galaxy clusters directly from simulated images of the microwave sky for the first time, using the mResUnet deep-learning algorithm. The mResUnet model is a feed-forward neural network designed for image-to-image regression. The trained mResUnet model simultaneously predicts a cluster’s SZ profile and mass, directly from an image of the microwave sky at the cluster location.

We train the model using Arnaud profiles for the SZ signal added to Gaussian realization of the CMB and astrophysical foregrounds. We include a 20% log-normal scatter in the predicted SZ signal as a function of cluster mass. We train the model with 200 simulated images at each of eight cluster masses, with $M_{200c} = (1, 2, 3, 4, 5, 6, 7, 8) \times 10^{14} M_\odot$.

We verify the trained model using different simulated images. We find that the trained model accurately recovers the cluster masses when the masses are within the trained range. For instance, we find $M_{200c} = (1.99 ± 0.40) \times 10^{14} M_\odot$ for an input mass of $M_{200c}^{\text{true}} = 2 \times 10^{14} M_\odot$. The combined intrinsic and observational scatter is consistent with the modeled 20% intrinsic log-normal $SZ$–mass scatter. We test this by comparing the scatter in the recovered masses for a set of 1000 clusters with masses randomly drawn from the mass range $2 \times 10^{14} M_\odot < M_{200c} < 7 \times 10^{14} M_\odot$. The fractional mass error across this set of 1000 clusters drops from 0.180 ± 0.013 to 0.100 ± 0.012 when the log-normal SZ scatter is set to zero, proving that the SZ scatter is the main source of uncertainty.

The model does not recover the mass of clusters outside the trained mass range. Unsurprisingly, for lower (higher) masses, it returns the lowest (highest) trained mass instead of the true mass.

While the model is trained on simplified SZ profiles (spherically symmetric Arnaud profiles), the trained model performs well when provided images with more realistic SZ profiles. We demonstrate this by taking 95 galaxy cluster cutouts included both more complex images, is analogous to taking the integrated Compton-$y$ within a radius. This test demonstrates that the deep-learning method should work on actual SZ images of galaxy clusters, even if the training set does not capture the full complexity of the real SZ signal.

In a future work, we will implement this deep-learning approach to estimate the mass of galaxy clusters using the real observations of microwave sky. Deep-learning-based mass estimation could provide an efficient way to estimate cluster masses for the sample of $>10^4$ galaxy clusters expected from ongoing (e.g., SPT-3G, AdvancedACT, Benson et al. 2014; Henderson et al. 2016) and upcoming (e.g., Simons Observatory, CMB-S4, Abazajian et al. 2019; Ade et al. 2019) CMB surveys. While requiring much larger training and validation data sets with wider dynamic range of mass and redshift of clusters, deep-learning networks can provide accurate mass measurements of galaxy clusters for current and future SZ surveys.

We acknowledge support from the Australian Research Council’s Discovery Projects scheme (DP150103208). We thank Raffaella Capasso, Sebastian Grandis, Brian Nord, João Caldeira, Sanjay Patil, and Federico Bianchini for their helpful feedback.

### ORCID iDs

N. Gupta @ https://orcid.org/0000-0001-7652-9451
C. L. Reichardt @ https://orcid.org/0000-0003-2226-9169

### References

Abazajian, K., Addison, G., Ashdahd, P., et al. 2019, arXiv:1907.04473
Ade, P., Aguirre, J., Ahmed, Z., et al. 2019, JCAP, 2019, 056
Alexander, S., Gleyzer, S., McDonough, E., Toomey, M. W., & Usai, E. 2020, ApJ, 893, 15
Allen, G., Andreoni, I., Bachelet, E., et al. 2019, arXiv:1902.00522
Armitage, T. J., Kay, S. T., & Barnes, D. J. 2019, MNRAS, 484, 1526
Arnaud, M., Pratt, G. W., Piffaretti, R., et al. 2010, A&A, 517, A92
Baxter, E. J., Keisler, R., Dodelson, S., et al. 2015, ApJ, 806, 247
Becker, M. R., & Kravtsov, A. V. 2011, ApJ, 740, 25
Benson, B. A., Ade, P. A. R., Ahmed, Z., et al. 2014, Proc. SPIE, 9153, 9153P
Biviano, A., Rosati, P., Balestra, I., et al. 2013, A&A, 558, A1
Bleem, L. E., Bocquet, S., Stalder, B., et al. 2020, ApJS, 247, 25
Bocquet, S., Dietrich, J. P., Schrabback, T., et al. 2019, ApJ, 878, 55
Bocquet, S., Saro, A., Mohr, J. J., et al. 2015, ApJ, 799, 214
Bottrell, C., Hani, M. H., Teimoorinia, H., et al. 2019, ApJ, 878, 55
Caldeira, J., Wu, W. L. K., Nord, B., et al. 2019, A&C, 28, 100307
Capazzo, R., Saro, A., Mohr, J. J., et al. 2019, MNRAS, 482, 1043
Carlstrom, J. E., Ade, P. A. R., Aird, K. A., et al. 2011, PASP, 123, 568
Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. 2016, arXiv:1606.00915
Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. 2017, arXiv:1706.05587
Corless, V. L., & King, L. J. 2009, MNRAS, 396, 315
de Haan, T., Benson, B. A., Bleem, L. E., et al. 2016, ApJ, 832, 95
Diemer, B., & Kravtsov, A. V. 2015, ApJ, 799, 108
Dietrich, J. P., Bocquet, S., Schrabback, T., et al. 2019, MNRAS, 483, 2871
Dolag, K., Komatsu, E., & Sunyaev, R. 2016, MNRAS, 461, 1797
Dolag, K., Vazza, F., Brunetti, G., & Tormen, G. 2005, MNRAS, 364, 753
Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., & Pal, C. 2016, arXiv:1608.04117
Dumoulin, V., & Visin, F. 2016, arXiv:1603.07285
Flurî, J., Kacprzak, T., Lucevi, A., et al. 2019, PhRvD, 100, 063514
Fowler, J. W., Nimmack, M. D., Dicker, S. R., et al. 2007, ApOpt, 46, 3444
Green, D., & Huerta, E. A. 2018, PhRvD, 97, 044039
George, E. M., Reichardt, C. L., Aird, K. A., et al. 2015, ApJ, 799, 177
Green, S. B., Ntampaka, M., Nagai, D., et al. 2019, ApJ, 884, 33
Gruen, D., Seitz, S., Brinckmann, F., et al. 2014, MNRAS, 442, 1507
Gu, J., Wang, Z., Kuen, J., et al. 2015, arXiv:1512.07108
Gupta, N., Saro, A., Mohr, J. J., Dolag, K., & Liu, J. 2017, MNRAS, 469, 3069
Hasselfield, M., Hilton, M., Marriage, T. A., et al. 2013, JCAP, 07, 008
He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1512.03385
Henderson, S. W., Allison, R., Austermann, J., et al. 2016, ILTP, 184, 772
Henning, J. W., Sayre, J. T., Reichardt, C. L., et al. 2018, APh, 852, 97
Hilton, M., Hasselfield, M., Sifón, C., et al. 2018, APh, 235, 20
