University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Gupta, N;Reichardt, CL

Title:
Mass Estimation of Galaxy Clusters with Deep Learning II. Cosmic Microwave Background Cluster Lensing

Date:
2021-12-01

Citation:
Gupta, N. & Reichardt, C. L. (2021). Mass Estimation of Galaxy Clusters with Deep Learning II. Cosmic Microwave Background Cluster Lensing. ASTROPHYSICAL JOURNAL, 923 (1), https://doi.org/10.3847/15384357/ac32d0.

Persistent Link:
https://hdl.handle.net/11343/310070

License:
CC BY

The Astrophysical Journal, 923:96 (8pp), 2021 December 10 https://doi.org/10.3847/1538-4357/ac32d0
© 2021. The Author(s). Published by the American Astronomical Society.
OPEN ACCESS

Mass Estimation of Galaxy Clusters with Deep Learning II. Cosmic Microwave Background Cluster Lensing

N. Gupta[1,2] and C. L. Reichardt[1]
[1] School of Physics, University of Melbourne, Parkville, VIC 3010, Australia; Nikhel.Gupta@csiro.au
[2] CSIRO Space and Astronomy Science, PO Box 1130, Bentley WA 6102, Australia
Received 2020 May 28; revised 2021 October 14; accepted 2021 October 21; published 2021 December 14

Abstract

We present a new application of deep learning to reconstruct the cosmic microwave background (CMB) temperature maps from images of the microwave sky and to use these reconstructed maps to estimate the masses of galaxy clusters. We use a feed-forward deep-learning network, mResUNet, for both steps of the analysis. The first deep-learning model, mResUNet-I, is trained to reconstruct foreground and noise-suppressed CMB maps from a set of simulated images of the microwave sky that include signals from the CMB, astrophysical foregrounds like dusty and radio galaxies, instrumental noise as well as the cluster's own thermal Sunyaev–Zel'dovich signal. The second deep-learning model, mResUNet-II, is trained to estimate cluster masses from the gravitational-lensing signature in the reconstructed foreground and noise-suppressed CMB maps. For SPTpol-like noise levels, the trained mResUNet-II model recovers the mass for 10^4 galaxy cluster samples with a 1σ uncertainty $\Delta M^{\rm est}_{\rm 200c}/M^{\rm est}_{\rm 200c}= 0.108$ and 0.016 for input cluster mass $M^{\rm true}_{\rm 200c} = 10^{14}\,M_\odot$ and $8\times10^{14}\,M_\odot$, respectively. We also test for potential bias on recovered masses, finding that for a set of 10^5 clusters the estimator recovers $M^{\rm est}_{\rm 200c} = 2.02\times10^{14}\,M_\odot$, consistent with the input at 1% level. The 2σ upper limit on potential bias is at 3.5% level.

Unified Astronomy Thesaurus concepts: Cosmic background radiation (317); Galaxy clusters (584); Large-scale structure of the universe (902); Cosmology (343)

1. Introduction

The number density of galaxy clusters is a promising approach to constrain cosmological models, especially those affecting late-time structure growth (e.g., Mantz et al. 2008; Vikhlinin et al. 2009; Hasselfield et al. 2013; de Haan et al. 2016; Planck Collaboration et al. 2016a; Bocquet et al. 2019; Costanzi et al. 2019). Current and upcoming experiments are expected to collectively detect more than 10^5 galaxy clusters in the next few years, two orders of magnitude larger than current cluster catalogs. Data are already being collected by some of these experiments (e.g., eROSITA, SPT-3G, AdvancedACT: Predehl et al. 2010; Benson et al. 2014; Henderson et al. 2016) and others plan to start operations in a near future (e.g., LSST, Euclid, Simons Observatory, CMB-S4: LSST Science Collaboration et al. 2009; Laureijs et al. 2011; Abazajian et al. 2019; Ade et al. 2019). The dramatically larger cluster catalogs have the potential to have a huge impact on our understanding of structure growth and the expansion history of the universe.

One major hurdle for extracting full cosmological information from these galaxy cluster catalogs is the calibration between the cluster mass and observables (see, e.g., DES Collaboration et al. 2020; Bocquet et al. 2015; Planck Collaboration et al. 2015). Several techniques have been used to estimate the mass of galaxy clusters. Among these, the weak gravitational lensing of background galaxies (e.g., Johnston et al. 2007; Gruen et al. 2014; Hoekstra et al. 2015; McClintock et al. 2019; Stern et al. 2019) and the lensing of the cosmic microwave background (CMB; e.g., Seljak & Zaldarriaga 2000; Dodelson 2004; Holder & Kosowsky 2004; Maturi et al. 2005; Lewis & Challinor 2006; Hu et al. 2007; Yoo & Zaldarriaga 2008; Baxter et al. 2015; Madhavacheril et al. 2015; Melin & Bartlett 2015; Planck Collaboration et al. 2016a; Geach & Peacock 2017; Baxter et al. 2018; Madhavacheril & Hill 2018; Raghunathan et al. 2019b) by galaxy clusters have demonstrated the potential for unbiased mass measurements.

Optical weak lensing becomes difficult for high-redshift galaxy clusters due to the decreasing density of background galaxies. In contrast, CMB lensing works well for at high redshift as CMB originates at $z\sim1100$. However, CMB lensing has its own limitations related to the raw signal-to-noise ratio (S/N) and the presence of foregrounds in CMB maps. A significant foreground for CMB lensing is the thermal Sunyaev–Zel'dovich (tSZ) effect (Sunyaev & Zel'dovich 1970, 1972) signal of the cluster itself that arises due to the inverse-Compton scattering of the CMB photons by energetic electrons in the intracluster medium. If not handled, the tSZ signal (and any other millimeter-wave signals sourced by the galaxy cluster) will bias the CMB lensing mass measurement of a galaxy cluster.

In the standard quadratic estimator (QE), the large-scale CMB gradient and the small-scale CMB anisotropy maps are used, and the lensing signal is extracted using the correlation between the different angular scales that are uncorrelated in the primordial CMB anisotropy map (Hu et al. 2007). Leveraging on the frequency dependence of the tSZ signal, the gradient CMB map can be freed from the tSZ that eliminates the induced correlation with the CMB anisotropy map (Madhavacheril & Hill 2018; Raghunathan et al. 2019b). While this approach eliminates the bias in gradient maps, the tSZ power in small-scale CMB anisotropy map adds extra variance that can become significant for high-mass clusters and low-noise surveys. A few other methods have been devised to reduce

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

the lensing bias and variance due to the tSZ signal from galaxy clusters. These methods include the inpainting of the gradient map based on the information from surrounding pixels (Raghunathan et al. 2019a) and tSZ template fitting (Patil et al. 2020).

The maximum likelihood estimator (MLE) on the other hand extracts an optimal lensing signal from CMB temperature maps (e.g., Dodelson 2004; Baxter et al. 2015; Raghunathan et al. 2017). The approach is based on fitting the lensed CMB templates to observed CMB maps and performs better than the standard QE by a factor of 2 at very low noise levels in the absence of a tSZ signal and astrophysical foregrounds. However, current MLEs depend on the map being clean of foregrounds and are biased when applied to maps with residual foregrounds (Raghunathan et al. 2017).

In this work, we demonstrate the first use of a deep-learning network to estimate the mass of galaxy clusters from the CMB lensing signal. We employ a modified version of a feed-forward deep-learning algorithm, mResUNet (Gupta & Reichardt 2020, hereafter G20) that combines residual learning (He et al. 2015) and the U-Net framework (Ronneberger et al. 2015). Two separate mResUNet models are trained independently. In the first step, we reconstruct the CMB temperature maps from the simulated images of microwave sky maps. This is done by training the mResUNet-I network to learn CMB features and mitigate the foreground (tSZ and astrophysical) signals as well as instrumental noise. In the second step, we use the reconstructed CMB temperature maps and mResUNet-II network to estimate the underlying mass for individual galaxy clusters. The mResUNet-II network is trained to extract lensing features from CMB temperature maps. After training these models for CMB reconstruction and mass estimation, we test the robustness of the process by using external hydrodynamical simulations of galaxy clusters.

The paper is structured as follows. In Section 2, we describe the simulations of microwave sky maps, the deep-learning model, and the parameters for its optimization. In Section 3, we present mass predictions using the images from test data sets as well as images from the external hydrodynamical simulations of SZ clusters. In Section 4, 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. Unless specified, the term "foregrounds" refers to both the tSZ signal and astrophysical emission. The central mass and the 1σ uncertainty are calculated as the median and the standard deviation divided by $\sqrt{N}$ (standard error), where N is the number of clusters, respectively.

2. Methods

In this section, we describe the microwave sky simulations of the lensed CMB temperature anisotropy, tSZ effect, radio and dusty galaxies, and instrumental noise. We then discuss the deep-learning models used to extract foreground-cleaned CMB temperature maps and to estimate the masses of galaxy clusters.

2.1. Simulations of Lensed CMB Temperature Maps

We create sets of simulations at each of 15 cluster masses, ranging from $1–8 \times 10^{14}\,M_{\odot}$ in steps of $0.5 \times 10^{14}\,M_{\odot}$. All clusters are assumed to lie at $z = 0.7$. These simulations include the CMB (lensed by the cluster), astrophysical foregrounds (not lensed), the effects of the instrumental beam, and instrumental noise. The simulated maps are trimmed to a box size of $10' \times 10'$ with a pixel resolution of $0.'25$. A Hanning window is applied to the outer $2'$ of the box to prepare for the convolutions in the neural network.

The first ingredient in these simulated maps is the CMB. We generate Gaussian realizations of the best-fit lensed ΛCDM cosmology for the Planck 2016 cosmology results (Planck Collaboration et al. 2016b); the power spectrum is calculated using CAMB[3] (Lewis et al. 2000). As we do for all the fields, we generate these CMB realizations on a larger box ($60' \times 60'$) to avoid edge effects from the Fourier transforms and to be sure we capture the large-scale gradients across the final box. The CMB maps are then lensed by the galaxy cluster at their center. We use interpolation to handle the sub-pixel-scale deflection angles. The cluster-lensing convergence profile, $\kappa(M, z)$, is based on the projected Navarro–Frenk–White (NFW) profile given by Bartelmann (1996).

To these cluster-lensed CMB maps, we add both astrophysical foregrounds unassociated with the cluster and the cluster's own SZ signal. For the first, we create Gaussian realizations of the best-fit baseline foreground model in George et al. (2015) at 150 GHz. This model includes power from other SZ halos, the kinematic SZ effect, radio galaxies, and the cosmic infrared background. For the second, we follow G20 and add a tSZ template at the cluster position based on the GNFW profile. We scale the amplitude and size of this profile following the power-law relations between the tSZ signal, mass, and redshift given in Arnaud et al. (2010). We also add a 20% log-normal scatter in the peak amplitude of the tSZ profile. Neither of the foregrounds nor the cluster's own SZ signal is lensed by the cluster.

We then convolve both the cluster-lensed CMB and foreground maps by a Gaussian instrumental beam with FWHM $=1'$. We store three combinations of these maps for each realization. The first combination map is representative of what a telescope would see; it has the sum of the CMB, foregrounds, and a realization of 5 μK-arcmin white noise. We find that the network is not biased by modest changes in the map noise. Specifically, we find no bias in the recovered mass when providing maps that have 30% higher white noise to a network trained with baseline noise. The test was done for a sample of 10,000 clusters with $M_{200c} = 4 \times 10^{14}\,M_{\odot}$, with a 1σ uncertainty on the recovered mass of 3.8%. Note that these simulations ignore correlated instrumental noise, as well as non-Gaussianity in the astrophysical foregrounds. Future work should assess the impact of these on the deep-learning estimator. The second combination map is used as a "truth" set in training the first network; it has only the CMB. The third combination map is the input for training the second network; this map has the CMB and a realization of white noise. We randomly draw the white-noise level for this third map from a uniform distribution between 0 and 5 μK-arcmin. Varying noise levels are designed to diversify the training set to better handle the non-Gaussian statistics in the output map from stage I (see Figure 3). We do not use the maps from stage I to train the second stage in order to ensure that the network is not trained to recover masses using information besides lensing, for example, from some residual SZ signal. The κ maps are used as

[3] https://camb.info/

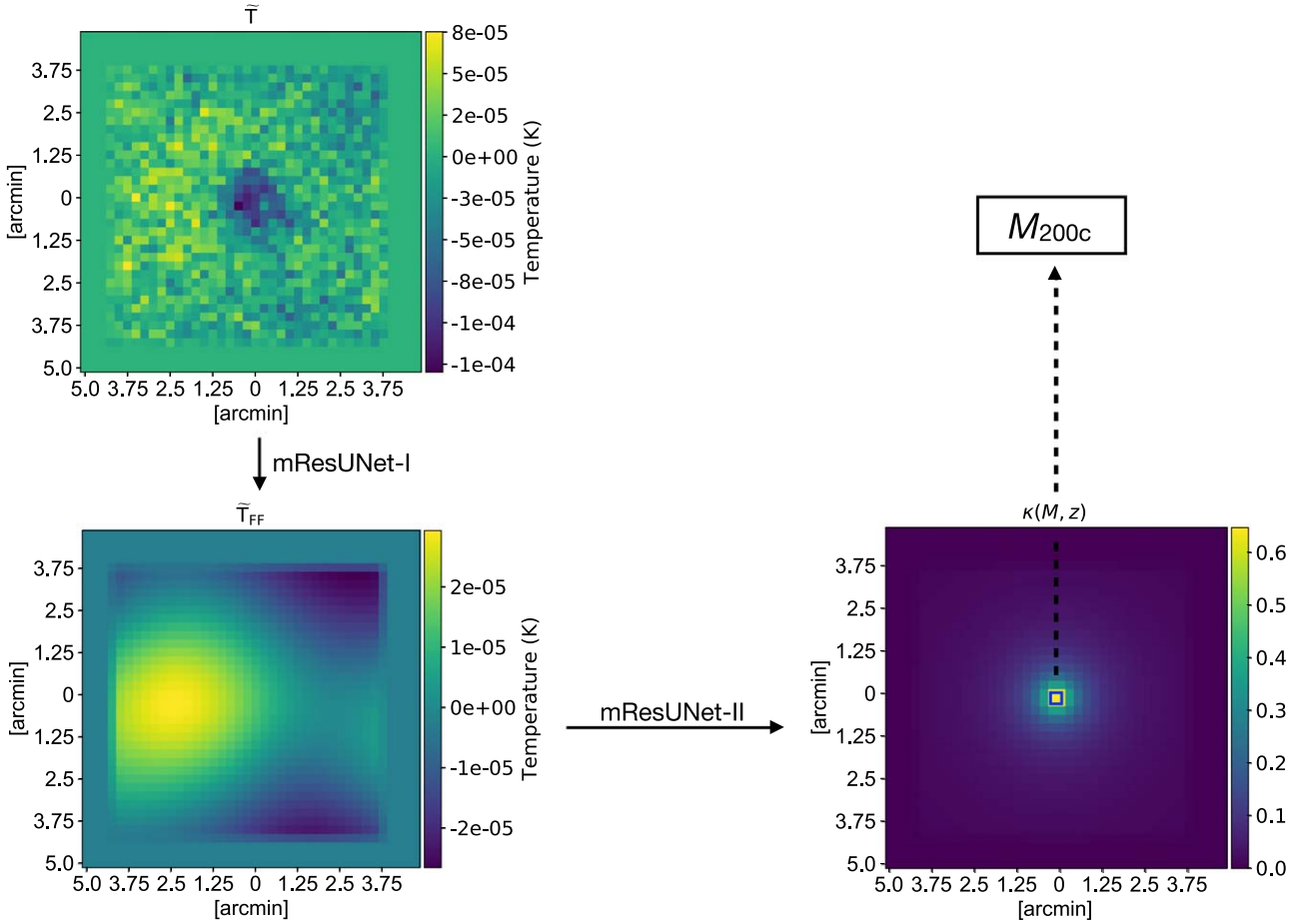


Figure 1. The workflow in the analysis. The top-left panel shows an example of the microwave sky CMB map ($\widetilde{T}$, where the tilde represents lensing) for a cluster with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$ at $z = 0.7$. This map includes cluster-lensed CMB, astrophysical foregrounds, cluster tSZ signal, and an instrumental white noise of 5 μK-arcmin. The map is convolved with a $1'$ telescope beam and apodization is applied. Several such maps for different cluster masses are used for training the neural network (mResUNet-I) to recover foreground and noise-suppressed maps ($\widetilde{T}_{FF}$) in the first stage of the analysis. In the second stage of the analysis, the cluster-lensed CMB maps with varying white-noise levels are used to train another network (mResUNet-II) to extract the lensing $\kappa(M, z)$ maps. The trained network is then applied to the foreground-cleaned CMB maps that are output by the first network. As described in Section 2.1, the mass of the cluster is then estimated from the central pixel of the extracted $\kappa(M, z)$ map.

the "truth" sets in training the second network. All maps are then trimmed to $10' \times 10'$ and apodized as described above. We have tested training the networks after trimming the maps to larger ($20' \times 20'$) sizes at a single cluster mass; we found a small improvement for $20' \times 20'$ maps. The uncertainty decreased by a factor of 0.93–0.98 for $20' \times 20'$ maps but consumed more computer resources. All results shown in this work use $10' \times 10'$ maps.

We end up with four maps per realization: the sky map with the CMB, foregrounds, and 5 μK-arcmin noise; the CMB-only map; the CMB map with a noise level uniformly drawn from the range 0–5 μK-arcmin; and the lensing convergence κ map. Because we are interested in the cluster mass only, we replace the central pixel of each $\kappa(M, z)$ map, such that the product of the central pixel value and the mean mass of the training sample equals M_{200c} of the cluster. The first network is trained on the sky maps to reconstruct the CMB-only maps. The second network is trained on the CMB maps with noise between 0–5 μK-arcmin level to reproduce the κ maps and recover the cluster mass. Figure 1 shows the examples of the sky map ($\widetilde{T}$), reconstructed CMB-only map ($\widetilde{T}_{FF}$), and lensing convergence $\kappa(M, z)$ map for a cluster with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$.

2.2. The mResUNet Model

We employ mResUNet (Caldeira et al. 2019), a feed-forward convolutional neural network, in the two legs of the analysis. The modified ResUNet, or mResUNet, algorithm was adapted by Caldeira et al. (2019) to do image-to-image regression, i.e., get an output image that is a continuous function of the input image, from the original ResUNet feed-forward deep-learning algorithm (Kayalibay et al. 2017; Zhang et al. 2018). The mResUNet algorithm is well suited to astrophysical problems, such as the current use case of estimating the lensed CMB signal from an image of the sky.

G20 tuned the mResUnet network by introducing dilation rates to extract small- and large-scale features in a CMB map, which we also use in this work. As shown in Figure 1 of G20, the framework is based on the encoder–decoder paradigm. This consists of a contracting path (encoder) to capture features, a symmetric expanding path (decoder) that enables precise localization. Each path has several convolution blocks, and each of these blocks has four substages. Every substage has a convolution layer, an activation function, and a batch normalization layer. The aim of the convolution layer is to learn features of an input map using filters that are applied to a receptive field of neighboring pixels. Each filter is typically a

$k \times k$ array with $k = 1, 3, 5, ...$, and the size of the filter ($k \times k$) is denoted as the kernel size. For each of the four substages, we apply dilations to the convolution layers with a rate of 1, 2, 3, and 4. A dilation rate of N stretches the receptive field by $k + (k-1)(N-1)$, thus doubling the dilation rate from 1 to 2 increases the receptive field from 3×3 to 5×5 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 in order to detect nonlinear features, leading to a highly nonlinear reconstruction of the input image (see Nwankpa et al. 2018 for a recent review). The batch normalization layer is helpful in improving the speed, stability, and performance of the network. The input to each convolution block is always added to its output using residual connections. To avoid overfitting, we add dropout layers to the decoding phase of the network. The weights of the network 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.

Figure 1 shows the overview of our analysis. In the first stage of analysis, we train the mResUNet-I network to reconstruct the CMB-only maps from the input sky maps that have CMB, noise, and foregrounds. In the second stage, we train the mResUNet-II network to estimate the lensing convergence map, and specifically the mass of galaxy clusters, from the recovered CMB-only maps.

2.3. Training and Optimization

The mResUNet-I and the mResUNet-II models take images as input and output same-sized images after passing through several convolutional blocks. The mResUNet-I network is trained to reconstruct $\widetilde{T}_{FF}$ maps, and the mResUNet-II network is trained to extract the $\kappa(M, z)$ maps. The central pixels of the extracted $\kappa(M, z)$ maps are used to estimate the mass of clusters (see Section 2.1). Note that a different image-to-float regression method can be used in the second step instead of the image-to-image regression model mResUNet-II. For simplicity, we chose the same network structure in both steps. A future study should compare different deep-learning networks to identify the most suitable model.

To train these networks, we normalize all input maps, so that the minimum and the maximum pixel values are between –1 and 1, respectively. This is done by dividing the image pixels by a constant factor across all cluster masses. The data for both networks are divided into three parts: training, validation, and test sets. The training and the validation data sets are used in the model-learning process. In both stages of the analysis, the training data have 400 maps and corresponding truth sets for each cluster. For the first model, we take cluster simulations with $M_{200c} = (1, 2, 3, 4, 5, 6, 7, 8) \times 10^{14}\, M_\odot$ for training. In second stage, the model is trained with $M_{200c} = \pm(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20...90, 100, 200...500) \times 10^{14}\, M_\odot$. We find that $M_{200}c = \pm 5 \times 10^{16}\, M_\odot$ is the optimized mass range for training the second model and for a narrower range, the model underestimates the mass uncertainties for clusters in the mass range of the first model.

The validation set has the same properties as the training set and is used to validate the model after each epoch, where one epoch is complete when entire training data are passed through the neural network once. We use 200 maps for each cluster mass and the corresponding truth sets 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 10^4 sky maps (for each cluster mass) for testing in the first stage of analysis and use recovered CMB-only maps from these test data to estimate the mass of the individual clusters in the second stage of analysis.

We use CMB maps with random realizations of white noise drawn from a uniform distribution between the 0 and 5 μK-arcmin level to train (400 maps) and validate (200 maps) the network in the second stage of analysis. In addition to the mass set used in training, we test our mResUNet-I and mResUNet-II models for cluster masses that were not the part of the training and validation process, that is, clusters with $M_{200c} = (1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5) \times 10^{14}\, M_\odot$.

The maps in the training set are passed through the neural networks with a batch size of 4 (16) for the first (second) model, and a training loss is computed as the 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 gradient descent and back-propagation. 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 mResUNet-I network, which is reduced to 0.2 for the mResUNet-II network. We consider the network to be trained and stop the training process if the validation loss does not improve for 15 epochs.

As described in Section 2.2, each substage of a convolution block in both models has a convolution layer, an activation layer, and a batch normalization layer. We set the kernel size of each convolution layer to 3×3 and change the stride length from 1 to 2, whenever the filter size is doubled. All activation layers in these networks have Scale Exponential Linear Unit (SELU; Klambauer et al. 2017) activation functions, which induce self-normalizing properties such that activations close to zero mean and unit variance converge toward zero mean and unit variance when propagated through many network layers, even under the presence of noise and perturbations. Only for the final layer in both models is a linear (or identity) activation function used to get same-sized output images as inputs. Each network has approximately 16 million parameters and is separately trained on a single GPU using Keras with a TensorFlow backend. For the first and second networks, training took roughly 100 and 240 s per epoch, respectively, for a total of 10 hours. The trained networks take a total of about 30 minutes to run over all 150,000 realizations in the test set including the time to load the network and simulations from disk.

3. Results

In this section, we look at the performance of our trained neural network models. We show the reconstructed CMB-only maps with foreground and noise suppressed ($\widetilde{T}_{FF}$) using the mResUNet-I network. We demonstrate that the mass of galaxy clusters can be estimated directly from the reconstructed $\widetilde{T}_{FF}$

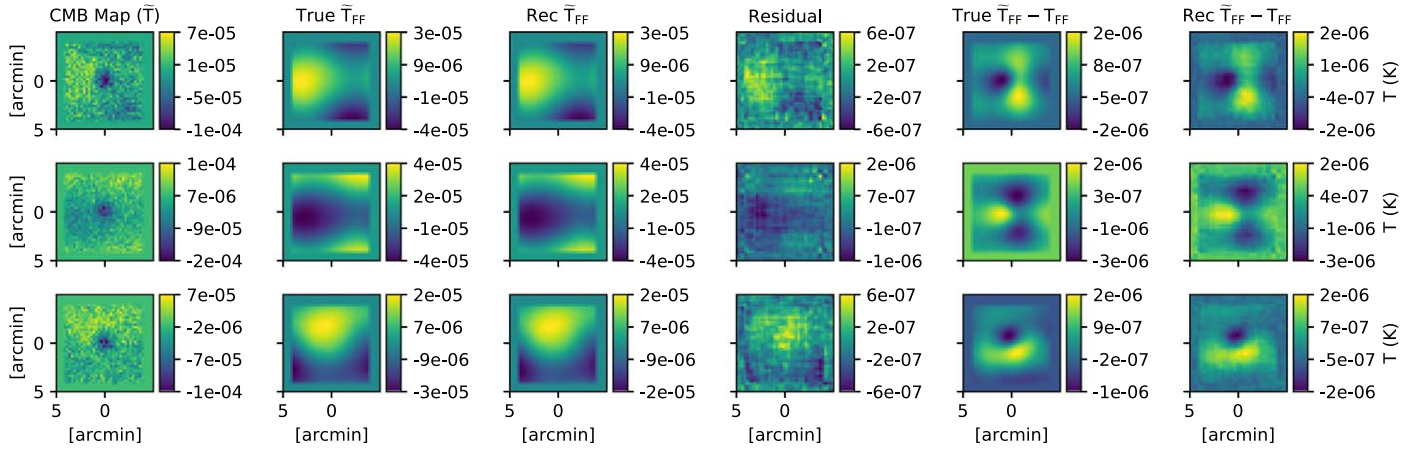


Figure 2. This figure illustrates the performance of the method for a random set of three galaxy clusters, with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$. Each row focuses on a single galaxy cluster, with the columns showing maps from different stages in the analysis. The first column shows the microwave sky ($\widetilde{T}$), including the cluster-lensed CMB, foregrounds, and instrumental noise. The second and third columns show the true and recovered CMB-only maps ($\widetilde{T}_{FF}$). The difference between these two is shown in the fourth column, labeled "Residual." The last two columns demonstrate the recovery of the desired CMB-cluster-lensing signal by showing first the true lensed–unlensed CMB map in column 5, and second the recovered lensed CMB map–true unlensed CMB map in column 6. The trained mResUNet-I model successfully reconstructs the CMB map at high accuracy including the critical CMB-cluster-lensing signal.

maps using the mResUNet-II network. We find per-cluster $\Delta M^{\rm est}_{200c}/M^{\rm est}_{200c}=$ 10.8 and 1.6 for input cluster mass $M^{\rm true}_{200c} = 10^{14}\, M_{\odot}$ and $8 \times 10^{14}\, M_{\odot}$, respectively. We test the performance of our trained deep-learning models by applying them to the sky maps with a more realistic tSZ signal from the external hydrodynamical simulations. Finally, we compare the S/N of the mass estimations from the current analysis to that estimated with the MLE.

3.1. Reconstruction of $\widetilde{T}_{FF}$ Maps with mResUNet-I

We use the test data set of 10^4 sky maps for each of the clusters to recover the CMB-only maps using the trained mResUNet-I model. These test maps are not used for training and validation purposes 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 tSZ signal (see Section 2.1). The first column in Figure 2 shows the examples of the input sky maps for three random realizations of a cluster with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$. The second and the third columns show the true and the reconstructed CMB-only maps, respectively. The fourth column shows residual signals, that is, the difference between the true and the recovered CMB-only maps. The fifth column shows the difference between the true lensed and the unlensed CMB-only maps, and the last column shows the same for the recovered CMB-only maps. This demonstrates that the trained mResUNet-I model successfully reconstructs the CMB map at high accuracy, including the critical CMB-cluster-lensing signal.

Figure 3 shows the distributions of the residual temperature calculated as the difference between the true and the recovered CMB-only maps. The green solid and black dotted contours show the distribution of all image pixels and central pixels, respectively, for 800 realizations of the cluster with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$. This indicates that the noise in the recovered CMB-only maps is non-Gaussian and resembles a Lorentz distribution with $\gamma = 0.5$.

3.2. Mass Estimation with mResUNet-II

We estimate the mass of galaxy clusters with the trained mResUNet-II network and use the 10^4 reconstructed CMB-only maps from the first stage of analysis. As the second network is trained on the cluster-lensed CMB maps with random noise realizations drawn from the uniform distribution between the 0 and 5 μK-arcmin level, it extracts the κ maps using only the cluster-lensing features in the reconstructed CMB-only maps. The mass of galaxy clusters is then estimated from the 10^4 extracted κ maps. As described in Section 2.1, this is done by multiplying the central pixel of the extracted κ maps by the mean mass of the training sample. The top panel in Figure 4 shows the estimated mass of single clusters as a function of the true mass. The black data points show estimations for cluster masses that are trained in the network, and the green data points describe the estimations for the interpolated mass set. All estimations are in the range of trained mass sample ($1 \times 10^{14}\, M_{\odot} < M_{200c} < 8 \times 10^{14} M_{\odot}$).

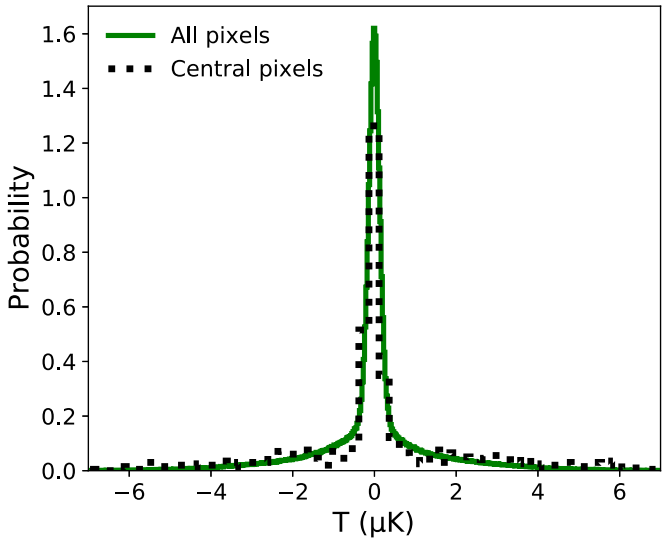


Figure 3. The distribution of the residual temperature, which is the difference between the true and the recovered CMB-only maps ($\widetilde{T}_{FF}$) as shown in the fourth column of Figure 2. The green solid and the black dotted contours show the distribution of all image pixels and central image pixels for 800 realizations of a cluster with $M_{200c} = 4 \times 10^{14}\, M_{\odot}$, respectively. The distribution is clearly non-Gaussian, resembling a Lorentz distribution with $\gamma = 0.5$. This indicates that the noise in the recovered $\widetilde{T}_{FF}$ maps is non-Gaussian.

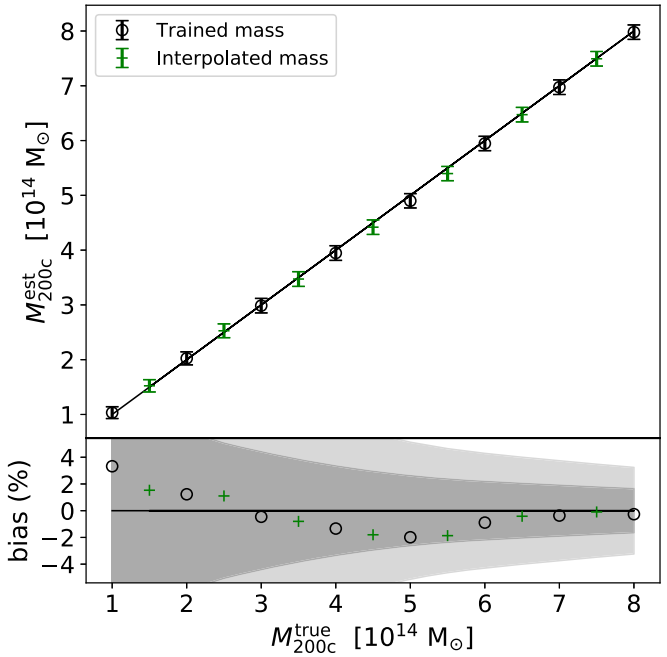


Figure 4. The mass estimates from the trained mResUNet-II model. The top panel shows the estimated and the true mass of clusters using a test data set of 10^4 CMB-only maps (with foreground and noise suppressed) per cluster mass recovered from the microwave sky maps in the first stage of analysis. The uncertainties are for 10^4 clusters of a given mass. The black data points show results for cluster masses equal to one of the training sets. The green data points (interpolation) describe results for clusters with masses between the trained masses. The bottom panel shows the percentage bias between the median estimated mass and true mass for a set of 10^4 clusters. The dark and light shaded regions denote the 1σ and 2σ standard error. This shows that the mass estimations from the deep-learning method are unbiased. For instance, the bias for cluster $M_{200c} = 4 \times 10^{14}\ M_\odot$ is <1.5% with a 1σ limit at the 3.5% level.

This shows that our trained mResUNet-II model can estimate the mass with good precision. For instance, we find $M^{est}_{200c} = 3.94 \pm 0.13 \times 10^{14}\ M_\odot$ for 10^4 clusters with $M^{true}_{200c} = 4 \times 10^{14}\ M_\odot$. While we have not looked at extrapolated masses in this work, we expect the network to show the same behavior as in G20. As shown in Figure 4 of that work, the network returned the highest (lowest) trained mass for clusters that were above (below) that mass. The training sample should encompass the entire cluster mass range of interest.

We estimate the bias for each mass as

$$\text{bias} = \frac{M^{est}_{200c}}{M^{true}_{200c}} - 1 \quad (1)$$

and find that the bias in the median mass estimation from this deep-learning method is $\leqslant$3% and within the 1σ level of the standard error for all masses (see bottom panel of Figure 4). For instance, the bias for the cluster mass $4 \times 10^{14}\ M_\odot$ is <1.5% with a 1σ limit for 10^4 clusters at the 3.5% level. This shows no evidence of bias from the estimator. While reassuring, we also confirm the nondetection for a lower-mass cluster with $M^{true}_{200c} = 2 \times 10^{14}\ M_\odot$ using 10^5 realizations, finding a 1% bias with a 2σ upper limit at the 3.5% level. This is much smaller than the 6.3% bias estimated by Raghunathan et al. (2017; see Table 1 and Section 4.6.2 in their paper) using MLE with a 1% residual tSZ signal in maps, indicating that the first step in our deep-learning method possibly leaves a <1% residual tSZ signal in the reconstructed maps.

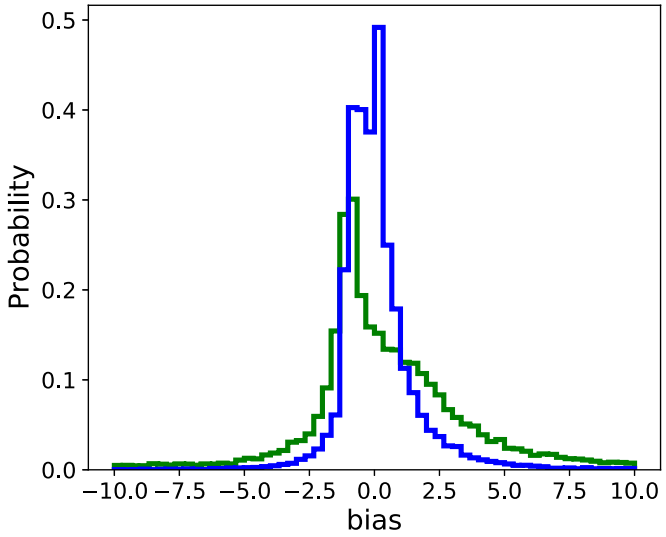


Figure 5. Test with external simulations: the deep-learning models precisely recover cluster masses for the sky maps with a more realistic tSZ signal from the independent Magneticum hydrodynamical simulation. Shown is the histogram of the bias for 30,000 CMB realizations using 140 tSZ profiles in the simulation (green contour). The median bias of the distribution is –0.04, and the standard error for 30,000 clusters is 0.04. The blue contour shows the distribution when Arnaud-based tSZ cluster profiles are used. This shows that trained neural networks recover unbiased mass estimates at the 4% level for a more realistic tSZ signal, even though the training is done with simplistic Arnaud profiles.

3.3. Testing Model with External Hydrodynamical Simulations

While we have shown that machine learning works well on the symmetric Arnaud profile used for training, a reasonable question is how it will perform with more realistic (i.e., complex) cluster profiles. We explore this question by running the trained network on images drawn from clusters in the Magneticum Pathfinder Simulation[4] (MPS; Dolag et al. 2016; Gupta et al. 2017; Soergel et al. 2018).

The MPS is a large hydrodynamical simulation carried out as a counterpart to ongoing, multiwavelength surveys and includes both the kSZ and tSZ effects. We take cutouts of the kSZ and tSZ maps provided with the MPS simulations at the locations of 140 galaxy clusters at $z = 0.67$ and $z = 0.73$ with $2 \times 10^{14}\ M_\odot < M_{200c} < 7 \times 10^{14}\ M_\odot$. The median mass of this sample is $M_{200c} = 2.5 \times 10^{14}\ M_\odot$. The MPS cluster catalog lists masses in terms of the overdensity of 500 times the critical density of universe; we convert this to M_{200c} using a model of the concentration–mass relation (Diemer & Kravtsov 2015). As the MPS maps have a pixel size of 0.′19, we interpolate these maps to the match the 0.′25 pixelation used in this work. We also convolve the SZ maps by the assumed $1'$ beam. The other signals in the millimeter-wave sky (the lensed CMB, instrumental noise, radio, and dusty galaxies) are generated as in Section 2.1 and added to the SZ maps. To increase the statistical power of the sample, we create 30,000 realizations of lensed CMB, white noise, and foregrounds for this sample of 140 SZ cluster profiles.

We pass the resulting maps to the two trained neural networks. Note that these networks have been trained with the symmetric Arnaud-based tSZ cluster profiles, not the MPS images. We plot the histogram of bias between the estimated and true masses in Figure 5. The median bias and standard error

[4] http://www.magneticum.org/

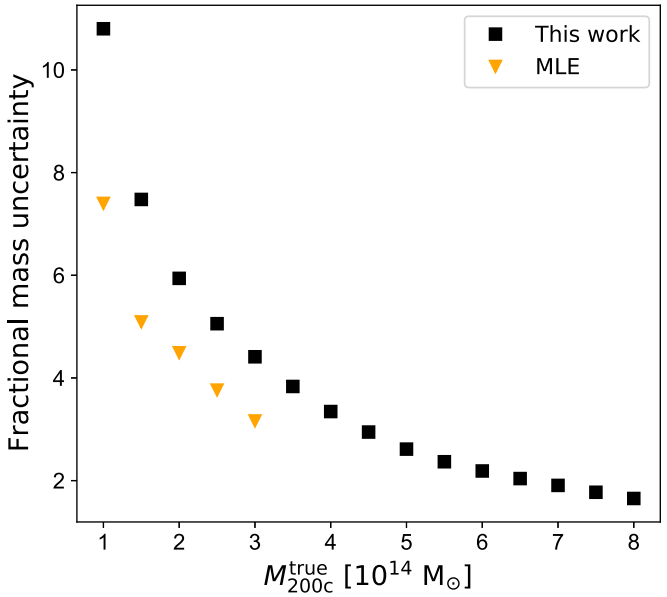


Figure 6. Comparison with MLE: the fractional mass uncertainty on a single cluster as a function of input mass. The black squares show the $\Delta M^{\rm est}_{\rm 200c}/M^{\rm est}_{\rm 200c}$ from the current work. The orange triangles show the $\Delta M^{\rm MLE}_{\rm 200c}/M^{\rm MLE}_{\rm 200c}$ from the MLE using the noisy CMB maps with astrophysical backgrounds. This shows that the fractional uncertainties from our deep-learning method are factors of 1.3–1.45 times higher than those from the MLE. For instance, the fractional mass uncertainty for the $2\times10^{14}\,M_\odot$ cluster is a factor 1.3 higher than the MLE limit.

for 30,000 clusters are –0.04 and 0.04, respectively (green contour). For comparison, we also plot the distribution for 30,000 sky maps that have Arnaud-based tSZ cluster profiles with $M_{\rm 200c} = (2,\ 2.5,\ 3)\times10^{14}M_\odot$, finding median bias and standard error of –0.01 and 0.025 (blue contour). This shows that despite the networks being trained on symmetric Arnaud SZ profiles, the neural networks still produce unbiased mass estimates on realistic SZ profiles with a 1σ limit for 30,000 clusters at the 4% level.

3.4. Comparison with Mass Estimations from MLE

The MLE uses pixel-to-pixel correlations to fit lensed CMB templates to the observed microwave sky CMB maps. The pixel-to-pixel covariance matrix is estimated using a set of simulated skies and is a function of cluster mass and redshift (e.g., Baxter et al. 2015). Figure 6 shows the uncertainties in our mass estimations as a function of input mass (black squares). The MLE uncertainties are estimated from the Gaussian realizations of CMB maps with 5 μK-arcmin white noise and astrophysical foregrounds. The tSZ signal is assumed to have been removed perfectly from these maps. For uncertainties estimated with deep learning, we use test maps with Gaussian realizations of CMB, 5 μK-arcmin white noise, astrophysical foregrounds, and tSZ signal. These maps are passed through the already-trained mResUNet-I model to reconstruct CMB-only maps with suppressed foregrounds and noise (see Section 3.1). The reconstructed CMB-only maps passed through the trained mResUNet-II model are used to estimate the uncertainties (see Section 3.2).

With deep learning, we find per cluster $\Delta M^{\rm est}_{\rm 200c}/M^{\rm est}_{\rm 200c}= 10.8$ and 1.6 for input cluster masses $M^{\rm true}_{\rm 200c} = 10^{14}\,M_\odot$ and $8\times10^{14}\,M_\odot$, respectively. We compare the deep-learning numbers to the MLE uncertainties for $10^{14}\,M_\odot - 3\times10^{14}\,M_\odot$ galaxy clusters with 5 μK-arcmin noise with foregrounds (orange triangles). Note that the MLE uncertainty is estimated for 10^5 clusters and the deep-learning errors are for 10^4 clusters (except for cluster mass $2\times10^{14}\,M_\odot$ for which 10^5 clusters are used to get deep-learning errors), we scale them for one cluster for this comparison in Figure 6. We find that the fractional mass uncertainties from the deep-learning method are factors of 1.3–1.45 higher than MLE estimates in the compared mass range. For instance, the $\Delta M^{\rm est}_{\rm 200c}/M^{\rm est}_{\rm 200c} = 5.9$ for $2\times10^{14}\,M_\odot$ which is a factor of 1.3 higher than the MLE limit.

The above comparisons show that the trained deep-learning models put competitive constraints on the cluster masses. Note that we do not use the reconstructed CMB-only maps with suppressed foregrounds and noise to derive the MLE mass uncertainties. This is mainly due to the non-Gaussian nature of the residual noise in these maps (see Figure 3). The application of MLE to the reconstructed CMB-only maps will require new developments, and we plan to modify the MLE in our future work.

4. Conclusions

We demonstrate for the first time a two-stage deep-learning algorithm that first extracts CMB maps with foreground and noise suppressed, and second estimates the mass of galaxy clusters from their gravitational-lensing imprint upon the CMB map.

The CMB sky maps include Gaussian realizations of CMB and astrophysical foregrounds, the cluster's own tSZ signal with a 20% intrinsic scatter (a foreground in this case), 5 μK-arcmin instrumental white noise, and 1′ beam smoothing. We train and validate the mResUNet-I network with 400 and 200 sky maps, respectively, for each of the eight cluster masses with $M_{\rm 200c} = (1, 2, 3, 4, 5, 6, 7, 8)\times10^{14}\,M_\odot$. The mResUNet-II network is trained (400) and validated (200) using CMB maps with $M_{\rm 200c} = \pm(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20...90, 100, 200...500)\times10^{14}\,M_\odot$, having white noise randomly drawn from a uniform distribution between the 0 to 5 μK-arcmin level. A test set of 10^4 sky maps is used to first reconstruct CMB-only maps with suppressed foregrounds and noise using the trained mResUNet-I model and then to estimate the underlying cluster mass with the trained mResUNet-II model. We find that the trained models recover the unbiased estimates of input mass. For instance, the bias for cluster mass $4\times10^{14}\,M_\odot$ is <1.5% with a 1σ limit for 10^4 clusters at the 3.5% level. As the uncertainty is larger for lower-mass clusters, we also confirm the nondetection for a lower-mass cluster with $M^{\rm true}_{\rm 200c} = 2\times10^{14}\,M_\odot$ using 10^5 realizations, finding a 1% bias with a 2σ upper limit at 3.5% level.

As the cluster's own SZ signal acts as the brightest foreground for the CMB-lensing signal, we test our trained models on more realistic SZ signals from the external hydrodynamical simulations. While the mResUNet-I model is trained on simplified tSZ profiles (spherically symmetric Arnaud profiles), the trained model performs well when provided CMB images with more realistic SZ profiles. We demonstrate this by taking 140 galaxy cluster SZ cutouts from the light cones of the Magneticum hydrodynamical simulation at $z = 0.67$ and $z = 0.73$ with $2\times10^{14}\,M_\odot < M_{\rm 200c} < 7\times10^{14}\,M_\odot$. These cutouts include more complex tSZ structure from the cluster itself, the kSZ signal from bulk motion of cluster, as well as the added tSZ contributions from other objects along nearby lines of sight. We randomly add 30,000

realizations of CMB, noise, and foregrounds to these SZ profiles and pass them through the trained deep-learning networks. Our trained mResUNet-I and mResUNet-II models recover the unbiased estimates of the true masses of the clusters with a 1σ upper limit at the 4% level.

We compare the mass uncertainties from the deep-learning models with those estimated from the MLE. We find that the fractional mass uncertainties from the deep-learning method are factors of 1.3–1.45 higher than MLE estimates in the compared mass range. For instance, the per-cluster $\Delta M^{\rm est}_{200c}/M^{\rm est}_{200c}= 5.9$ for $2\times10^{14}\,M_{\odot}$ is a factor of 1.3 higher than the MLE limit.

In our future work, we plan to apply this two-stage deep-learning approach to estimate galaxy cluster masses in real CMB data. Presuming no insurmountable challenges appear, deep learning would be a valuable tool for determining the masses of high-redshift galaxy clusters in ongoing and upcoming CMB surveys (e.g., SPT-3G, AdvancedACT, Simons Observatory, CMB-S4; Benson et al. 2014; Henderson et al. 2016; Abazajian et al. 2019; Ade et al. 2019). Accurate mass estimates across the full range of redshifts will be essential to fully utilizing the large, $>10^5$ galaxy cluster samples expected from CMB, X-ray, and optical surveys (e.g., eROSITA, LSST, Euclid; LSST Science Collaboration et al. 2009; Predehl et al. 2010; Laureijs et al.2011).

We acknowledge support from the Australian Research-Council's Discovery Projects scheme (DP150103208). NG acknowledges support from CSIRO's Machine Learning and Artificial Intelligence Future Science Platform. This research uses resources of the National Energy Research Scientific Computing Center (NERSC). We thank Srinivasan Raghunathan, Sanjay Patil, Brian Nord, João Caldeira, 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., Adshead, P., et al. 2019, arXiv:1907.04473
Ade, P., Aguirre, J., Ahmed, Z., et al. 2019, JCAP, 2019, 056
Arnaud, M., Pratt, G. W., Piffaretti, R., et al. 2010, A&A, 517, A92
Bartelmann, M. 1996, A&A, 313, 697
Baxter, E. J., Keisler, R., Dodelson, S., et al. 2015, ApJ, 806, 247
Baxter, E. J., Raghunathan, S., Crawford, T. M., et al. 2018, MNRAS, 476, 2674
Benson, B. A., Ade, P. A. R., Ahmed, Z., et al. 2014, Proc. SPIE, 9153, 91531P
Bocquet, S., Saro, A., Mohr, J. J., et al. 2015, ApJ, 799, 214
Bocquet, S., Dietrich, J. P., Schrabback, T., et al. 2019, ApJ, 878, 55
Caldeira, J., Wu, W. L. K., Nord, B., et al. 2019, A&C, 28, 100307
Costanzi, M., Rozo, E., Simet, M., et al. 2019, MNRAS, 488, 4779
de Haan, T., Benson, B. A., Bleem, L. E., et al. 2016, ApJ, 832, 95
DES Collaboration, Abbott, T., Aguena, M., et al. 2020, PhRvD, 102, 023509
Diemer, B., & Kravtsov, A. V. 2015, ApJ, 799, 108
Dodelson, S. 2004, PhRvD, 70, 023009
Dolag, K., Komatsu, E., & Sunyaev, R. 2016, MNRAS, 463, 1797
Geach, J. E., & Peacock, J. A. 2017, NatAs, 1, 795
George, E. M., Reichardt, C. L., Aird, K. A., et al. 2015, ApJ, 799, 177
Gruen, D., Seitz, S., Brimioulle, F., et al. 2014, MNRAS, 442, 1507
Gupta, N., & Reichardt, C. L. 2020, ApJ, 900, 110
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, 7, 008
He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1512.03385
Henderson, S. W., Allison, R., Austermann, J., et al. 2016, JLTP, 184, 772
Hoekstra, H., Herbonnet, R., Muzzin, A., et al. 2015, MNRAS, 449, 685
Holder, G., & Kosowsky, A. 2004, ApJ, 616, 8
Hu, W., DeDeo, S., & Vale, C. 2007, NJPh, 9, 441
Johnston, D. E., Sheldon, E. S., Wechsler, R. H., et al. 2007, arXiv:0709.1159
Kayalibay, B., Jensen, G., & van der Smagt, P. 2017, arXiv:1701.03056
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. 2017, arXiv:1706.02515
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv:1110.3193
Lewis, A., & Challinor, A. 2006, PhR, 429, 1
Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473
LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv:0912.0201
Madhavacheril, M., Sehgal, N., Allison, R., et al. 2015, PhRvL, 114, 151302
Madhavacheril, M. S., & Hill, J. C. 2018, PhRvD, 98, 023534
Mantz, A., Allen, S. W., Ebeling, H., & Rapetti, D. 2008, MNRAS, 387, 1179
Maturi, M., Bartelmann, M., Meneghetti, M., & Moscardini, L. 2005, A&A, 436, 37
McClintock, T., Varga, T. N., Gruen, D., et al. 2019, MNRAS, 482, 1352
Melin, J.-B., & Bartlett, J. G. 2015, A&A, 578, A21
Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. 2018, arXiv:1811.03378
Patil, S., Raghunathan, S., & Reichardt, C. L. 2020, ApJ, 888, 9
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2015, A&A, 594, A24
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016a, A&A, 594, A24
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016b, A&A, 594, A13
Predehl, P., Andritschke, R., Böhringer, H., et al. 2010, Proc. SPIE, 7732, 77320U
Raghunathan, S., Holder, G. P., Bartlett, J. G., et al. 2019a, JCAP, 2019, 037
Raghunathan, S., Patil, S., Baxter, E. J., et al. 2017, JCAP, 2017, 030
Raghunathan, S., Patil, S., Baxter, E., et al. 2019b, ApJ, 872, 170
Ronneberger, O., Fischer, P., & Brox, T. 2015, arXiv:1505.04597
Ruder, S. 2016, arXiv:1609.04747
Seljak, U., & Zaldarriaga, M. 2000, ApJ, 538, 57
Soergel, B., Saro, A., Giannantonio, T., Efstathiou, G., & Dolag, K. 2018, MNRAS, 478, 5320
Stern, C., Dietrich, J. P., Bocquet, S., et al. 2019, MNRAS, 485, 69
Sunyaev, R. A., & Zel'dovich, Y. B. 1970, CoASP, 2, 66
Sunyaev, R. A., & Zel'dovich, Y. B. 1972, CoASP, 4, 173
Vikhlinin, A., Kravtsov, A. V., Burenin, R. A., et al. 2009, ApJ, 692, 1060
Yoo, J., & Zaldarriaga, M. 2008, PhRvD, 78, 083002
Yu, F., & Koltun, V. 2015, arXiv:1511.07122
Zhang, Z., Liu, Q., & Wang, Y. 2018, IGRSL, 15, 749