Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **507,** 4149–4164 (2021) https://doi.org/10.1093/mnras/stab2229
Advance Access publication 2021 August 5

DeepSZ: identification of Sunyaev–Zel'dovich galaxy clusters using deep learning

Z. Lin,[1★] N. Huang,[2] C. Avestruz,[3,4★] W. L. K. Wu,[5,6] S. Trivedi,[7] J. Caldeira[8] and B. Nord[5,8,9]

[1] *Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*
[2] *Department of Physics, University of California, Berkeley, CA 94720, USA*
[3] *Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA*
[4] *Leinweber Center for Theoretical Physics, University of Michigan, Ann Arbor, MI 48109, USA*
[5] *Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, USA*
[6] *SLAC National Accelerator Laboratory & KIPAC, 2575 Sand Hill Road, Menlo Park, CA 94025, USA*
[7] *CSAIL, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA*
[8] *Fermi National Accelerator Laboratory, PO Box 500, Batavia, IL 60510, USA*
[9] *Department of Astronomy and Astrophysics, University of Chicago, Chicago, IL 60637, USA*

Accepted 2021 July 30. Received 2021 July 25; in original form 2021 March 19

ABSTRACT

Galaxy clusters identified via the Sunyaev–Zel'dovich (SZ) effect are a key ingredient in multiwavelength cluster cosmology. We present and compare three methods of cluster identification: the standard matched filter (MF) method in SZ cluster finding, a convolutional neural networks (CNN), and a 'combined' identifier. We apply the methods to simulated millimeter maps for several observing frequencies for a survey similar to SPT-3G, the third-generation camera for the South Pole Telescope. The MF requires image pre-processing to remove point sources and a model for the noise, while the CNN requires very little pre-processing of images. Additionally, the CNN requires tuning of hyperparameters in the model and takes cut-out images of the sky as input, identifying the cut-out as cluster-containing or not. We compare differences in purity and completeness. The MF signal-to-noise ratio depends on both mass and redshift. Our CNN, trained for a given mass threshold, captures a different set of clusters than the MF, some with signal-to-noise-ratio below the MF detection threshold. However, the CNN tends to mis-classify cut-out whose clusters are located near the edge of the cut-out, which can be mitigated with staggered cut-out. We leverage the complementarity of the two methods, combining the scores from each method for identification. The purity and completeness are both 0.61 for MF, and 0.59 and 0.61 for CNN. The combined method yields 0.60 and 0.77, a significant increase for completeness with a modest decrease in purity. We advocate for combined methods that increase the confidence of many low signal-to-noise clusters.

Key words: methods: data analysis – methods: statistical – galaxies: clusters: general.

1 INTRODUCTION

The abundance of galaxy clusters is sensitive to cosmological parameters (Allen, Evrard & Mantz 2011). Galaxy clusters have provided cosmological constraints with data from multiple wavelengths including X-ray (Vikhlinin et al. 2009; Mantz et al. 2010, 2015; Mehrtens et al. 2012), microwave (Sehgal et al. 2011; Planck Collaboration XXIV 2016a; Bocquet et al. 2019; Hilton et al. 2021), and optical (Rozo et al. 2010; Costanzi et al. 2019; To et al. 2021). One potential systematic uncertainty in cluster-based cosmology is the selection function of observed clusters. Clusters observed in millimeter maps have one of the better understood selection functions, providing a sample selected based on SZ signal significance, which is highly correlated with mass.

Galaxy clusters are collections of galaxies ensconced in a halo of dark matter, which provides most of the gravitational potential. Amidst the galaxies in a cluster, there exists a hot intracluster medium that emits in the X-rays (via bremsstrahlung), and which makes them observable in the millimeter through the Sunyaev–Zel'dovich (SZ) effect. The SZ effect is an upscattering of cosmic microwave background (CMB) photons that shifts the CMB blackbody spectrum along the line of sight of a galaxy cluster (Carlstrom, Holder & Reese 2002). The SZ effect is independent of redshift and dependent only on the pressure of the intracluster medium, a quantity strongly correlated with cluster mass. An SZ-selected galaxy cluster sample therefore provides what is close to a mass-limited selection function, which is straightforward to incorporate in cosmological analyses. SZ-selected clusters have resulted in a number of astrophysical studies as well. Follow-up observations probe the physics of the intracluster medium and cluster galaxies from data in other wavelengths. SZ cluster follow-up include *Chandra* (McDonald et al. 2017) or *XMM–Newton* (Bulbul et al. 2019) in the X-ray and the *Hubble Space Telescope* (Strazzullo et al. 2019) in the optical and *Spitzer* (Strazzullo et al. 2019) or *Herschel* (Zohren et al. 2019) in the infrared.

★ E-mail: zhenlin4@illinois.edu (ZL); cavestru@umich.edu (CA)

The traditional method of identifying SZ clusters is to deploy a matched filter (MF) based method on the maps (Melin, Bartlett & Delabrouille 2006; Melin et al. 2012), which identifies regions in the maps that maximize the signal-to-noise ratio (SNR) corresponding to the filter shape. The method has successfully identified cosmological samples in maps constructed with survey data from the South Pole Telescope (Bleem et al. 2015; Huang et al. 2020), *Planck* (Planck Collaboration XXVII 2016b), and the Atacama Cosmology Telescope (Hilton et al. 2018). The SPT-3G camera, deployed in 2017, dramatically increases mapping speed over previous cameras. It is expected that there will be 5000 cluster detections at 97 per cent purity in the 1500 deg^2 survey area (Benson et al. 2014). Next-generation experiments, like CMB-S4, will have lower noise and will likely be able to see even more objects (Abazajian et al. 2016).

Convolutional neural networks (CNNs) are quickly becoming an essential tool for cosmology and astrophysics (Ntampaka et al. 2019a). CNNs have already been used for both CMB analyses, e.g. by Caldeira et al. (2019), Krachmalnicoff & Tomasi (2019), and Hortúa et al. (2020), and analyses related to galaxy clusters. Recent applications of CNNs to galaxy clusters include mass estimations from mock X-ray images (Ntampaka et al. 2019b) and velocity dispersion distributions (Ho et al. 2019). Complementary to mass estimation analyses, Green et al. (2019) used machine learning methods to identify physically relevant features in X-ray images that correspond to a galaxy cluster dynamical state. However, applications of machine learning to galaxy cluster observables in the CMB are still emerging.

Examples of machine learning to galaxy clusters in the CMB include Hurier, Aghanim & Douspis (2017), where neural networks produced filtered and cleaned maps to improve resulting cluster catalogues with lower mass thresholds and therefore higher redshifts, a proof-of-concept deep learning application to identify SZ clusters in *Planck* survey data (Bonjean 2020), and to cluster mass estimates from CMB lensing (Gupta & Reichardt 2020). We emphasize that our work is the first to explicitly use deep neural networks for the identification of SZ clusters in their image space from millimeter maps in the absence of additional map filtering or cleaning steps. In particular, we use CNNs to identify cluster-containing cut-out of the simulated CMB sky. We compare our results with the standard cluster-finding method in the CMB, which has complementary performance, and explore the benefits of combining cluster-finding methods with an example implementation of such combinations and the resulting comparisons.

In this work, the CNN classification is binary, with the CNN output corresponding to a rank-ordered likelihood for a cut-out of the sky to contain a cluster above a mass threshold versus not. However, the standard MF cluster-finding method assigns detection significance as a function of SNR, which increases with cluster mass. As such, an apple-to-apples comparison between the two methods is admittedly artificial. We do devise a consistent way of comparing the two methods that highlights their respective strengths. But, we note that, for a future work, a CNN regression method that predicts the mass of a cluster would more naturally lend itself to an apples-to-apples comparison with the standard MF output.

We highlight a few innovative areas of this work. We have devised a new and effective training approach for extremely unbalanced samples. We introduce a metric widely used in the machine learning literature, the F1 score, that assesses the combined completeness and purity of the cluster sample. The F1 score efficiently summarizes the effectiveness of the cluster finder, enabling a comparison between the CNN and MF methods. We also use the F1 score to evaluate a combined method that incorporate both outputs from the CNN and the MF. One can apply this approach to combine other machine learning methods with a standard method of cluster finding.

The paper is organized as follows. Section 2 describes the data set we use to train and test the network. Section 3 describes both the traditional MF method used to detect SZ clusters and our neural network and training for our deep learning model. Section 4 describes our results for the network alone and the 'combined-classifier' that incorporates results from the MF method. We discuss the implications of the results and summarize our paper in Section 5. The codes we used are published on https://github.com/zlin7/deepsz.

2 DATA

In this section, we discuss the origin and preparation of the data.

2.1 Simulations

We take simulations of the microwave sky from Sehgal et al. (2010), which are built on top of an N-body simulation. Briefly, the N-body simulation used for the sky simulation has box size $L = 1000\,h^{-1}$ Mpc, with 1024^3 particles with particle mass $6.82 \times 10^{10}\,h^{-1}\,\mathrm{M}_\odot$ and softening length $\epsilon = 16.276\,h^{-1}$ kpc. These simulations provide a full-sky realization of the lensed CMB, galactic dust, point sources, and the SZ effect (both kinematic and thermal) for observing frequencies 27, 30, 39, 44, 70, 93, 100, 143, 145, 219, 225, 280, and 353 GHz.

The sky simulations include the SZ signal from galaxy clusters, with halo virial masses and locations identified in the N-body simulation using a friends-of-friends halofinder, with a linking length 0.2 of the mean interparticle spacing. The data products we use include separate all-sky maps for each component, as well as catalogues for the locations of N-body haloes, and point sources. The haloes and point sources are only unique on one octant of the sky (the other octants use various reflections of the catalogues). We therefore restrict our search to only one octant of the sky. We further restrict our search to the 90, 148, and 219 GHz channels, motivated by typical ground-based CMB telescopes. In addition to the simulated sky signals, we create white noise realizations to imitate the effect of instrumental noise. The instrument noise levels are 2.8, 2.6, and 6.6 μK-arcmin for 90, 148, and 219 GHz maps, respectively – consistent with projected performance for the SPT-3G camera (Benson et al. 2014).

For the purposes of our search, everything except the SZ signal from high-mass haloes is a noise term. On large angular scales ($\gtrsim$10 arcmin), the maps are dominated by the CMB (because our maps are in units relative to the CMB temperature, the unlensed CMB map for each band is identical). On arcminute scales, the maps are dominated by point sources. Thermal sources (dusty star-forming galaxies and galactic dust) are brighter at higher frequencies, while radio sources (radio-loud galaxies, typically AGN) are brighter at lower frequencies. The SZ signal from galaxy clusters occupies the space between point sources and the CMB, with angular scales typically between 1 and 10 arcmin. In the three bands used in this work, the SZ signal is negative, and most significant at 90 GHz. The 219 GHz channel is aligned with the null of the SZ spectrum, and has very little thermal SZ signal. These simulations contain thermal SZ contributions from a large number of low-mass clusters, which are well below the detection threshold. We must take these into account as an additional noise term. The kinematic SZ signal is much smaller than any of the other noise terms.

To make these simulations more realistic, we include a noise term based on the predicted instrumental noise from the full SPT-3G survey (as noted above). However, there are additional instrumental effects that we have ignored, which must be accounted for in real data. First, we have used the simulations at their native resolution (the highest resolution being 0.43 arcmin corresponding to HEALPix $N_{\rm side} = 8192$), without including the instrument beam. The beams from the SPT are well represented by Gaussians with a full width at half-maximum of $\sim 1\overset{\prime}{.}0$. Secondly, the map-making process used by the SPT collaboration includes time-domain filtering to remove low frequency noise due to both the instrument and the atmosphere. In the map domain, the filtering is approximately represented by an anistropic filter that preferentially removes long wavelength modes in the RA direction. Finally, the remaining noise is not white, but increases at larger scales. None of these effects are included in our simulations.

2.2 Data preparation

Data preparation for the CNN model is simpler than for the MF. Model training for the network does *not* require any special pre-processing of the maps such as normalization, or point source removal. For example, instead of affecting manual normalization, the usual steps in training CNNs such as convolution, batch normalization, etc. implicitly process the images in a manner suitable for the prediction. On the other hand, the MF method does in fact require data pre-processing, such as point source removal, as described in Section 3.1.

To construct maps for the CNN, we take components and simply add them together. First, we start with maps with just the CMB and tSZ. We then add instrument noise to the maps. We then progressively add infrared galaxies, radio sources, and galactic dust emissions to the cut-out cumulatively to facilitate investigation into which of these components have the largest effect on cluster identification. The maps used for the MF are described in Section 3.1.

Our data set consists of overlapping patches (cut-out) of the mock sky. There is no natural bijection between HEALPix pixels and the 2D image pixels usually used in image classification. Because of this and the fact that different components in the simulation have different native resolutions, we choose a resampling that both allows for cut-out that fully contain clusters and minimizes necessary modifications to the CNN base architecture (more in Section 3.2.1). We therefore prepare our data set by cutting the sky into 8 × 8 arcmin^2, with the resolution set to 32 pixels on a side. Cut-out are staggered with a stride of 6 arcmin. In other words, neighbouring maps share 2 arcmin on the edge with each other.

Any given cut-out likely contains a bound object somewhere along the line of sight. For the purposes of SZ cluster identification, we label cut-out as 'positive' under the following conditions. We first consider cut-out containing clusters whose footprint in the sky is sufficiently small compared with the size of the cut-out, by setting a redshift threshold. From this catalogue and cut-out, our selection consists of clusters at redshift above $z \geq 0.25$ and with virial mass above $\mathrm{M}_{\rm vir} \geq 2 \times 10^{14}\,\mathrm{M}_{\odot}$. Furthermore, we condition a cut-out as 'positive' only if the cluster position is located within the 6 × 6 arcmin region at the centre of the cut-out.

Our conditions for a positive cut-out ensure that most of the cluster's on-sky footprint is contained in the cut-out image, thereby reducing potential edge effects. Since the distance between the centers of our cut-out is 6 arcmin, each cluster in our mass and redshift range is contained in exactly one 'positive' cut-out.

Given our specific choice of mass and redshift threshold, only around 1 per cent of the cluster-containing cut-out contained more than one cluster that fit our threshold. A potential direction one could take is to iteratively train models where the thresholds for cluster-containing cut-out change. However, this would result in multiple clusters per cut-out, complicating comparisons with the MF results. We therefore choose the mass and redshift thresholds specified for the label 'cluster-containing' to simplify the performance comparison with the MF. To enable a straightforward treatment of purity, completeness, and F1 score, we assume a bijective correspondence between a true positive prediction in the data set and an actual halo from the original simulations.

With this procedure, our data set contains 808 201 total cut-out. Of these, 14 989 are labelled as positive. Note that the positive samples form a small percentage of the full data set (less than 2 per cent). In machine learning, this is referred to as an *unbalanced* data set since the class of 'Negative' labels has many more objects than the class of 'Positive' labels. Unbalanced data sets pose an additional challenge when training neural networks. We discuss this challenge and our approach in a later section (Section 3.3), where we have identified a metric that is not sensitive to the class imbalance.

The next stage of data preparation is to split our data set into training, test, and validation sets. We group cut-out by their position in the sky. The *training* set is comprised of cut-out with centre right ascension (RA) coordinate above 0.2 × 90 deg, the *test* set comprised of cut-out with centre RA coordinate below 0.13 × 90 deg, and the remaining cut-out grouped into the *validation* set. To avoid training a network on the periphery of a cluster, in the training set we remove the negative cut-out that are adjacent to a positive one. We do not do this on the validation or test set. The final numbers of cut-out in the training, validation and test sets are 601 400, 105 183, and 56 637, respectively.

Fig. 1 shows two sets of example maps at 90, 148, and 219 GHz. The top half of this figure shows cut-out maps with a galaxy cluster, where the true position of the cluster is marked with a red 'X'. The lower panel of this figure shows example cut-out maps that do not contain a galaxy cluster. From left to right, we add additional sources of noise. The addition of IR and radio galaxies increases the visual inhomogeneity of the image.

We next discuss the impact of high flux sources. Fig. 2 shows an example cluster-containing cut-out from our sample that happens to also host a high flux radio galaxy. The column on the left is the image with only the CMB + tSZ components simulated, similar to the leftmost column of Fig. 1. The column on the right has all other components added, similar to the rightmost column of Fig. 1. In the absence of the high flux radio galaxy (left-hand column), the tSZ signal from the galaxy cluster is visually identifiable in the 90 and 148 GHz frequencies and the colourmap for these images spans a few hundred μK. In contrast, the images that include the high flux radio galaxy (right-hand column) span the thousands of μK, and the tSZ signal from the galaxy cluster is barely identifiable in the only the 90 GHz frequency with the same linear scaling. The high-flux galaxy is the main visible feature in these images, seen as two very bright pixels near the centre of the cluster.

The MF method requires a pre-processing step of point-source removal in order to identify the high signal-to-noise co-located pixels of the tSZ signal from galaxy clusters. Additionally, the MF requires an input of both the SZ frequency spectrum and an assumed cluster profile of a beta model. The CNN requires none of these. Many classical machine learning methods also require some sort of pre-processing, such as rescaling and normalization, to optimally perform. We emphasize that the CNN method we use does not require

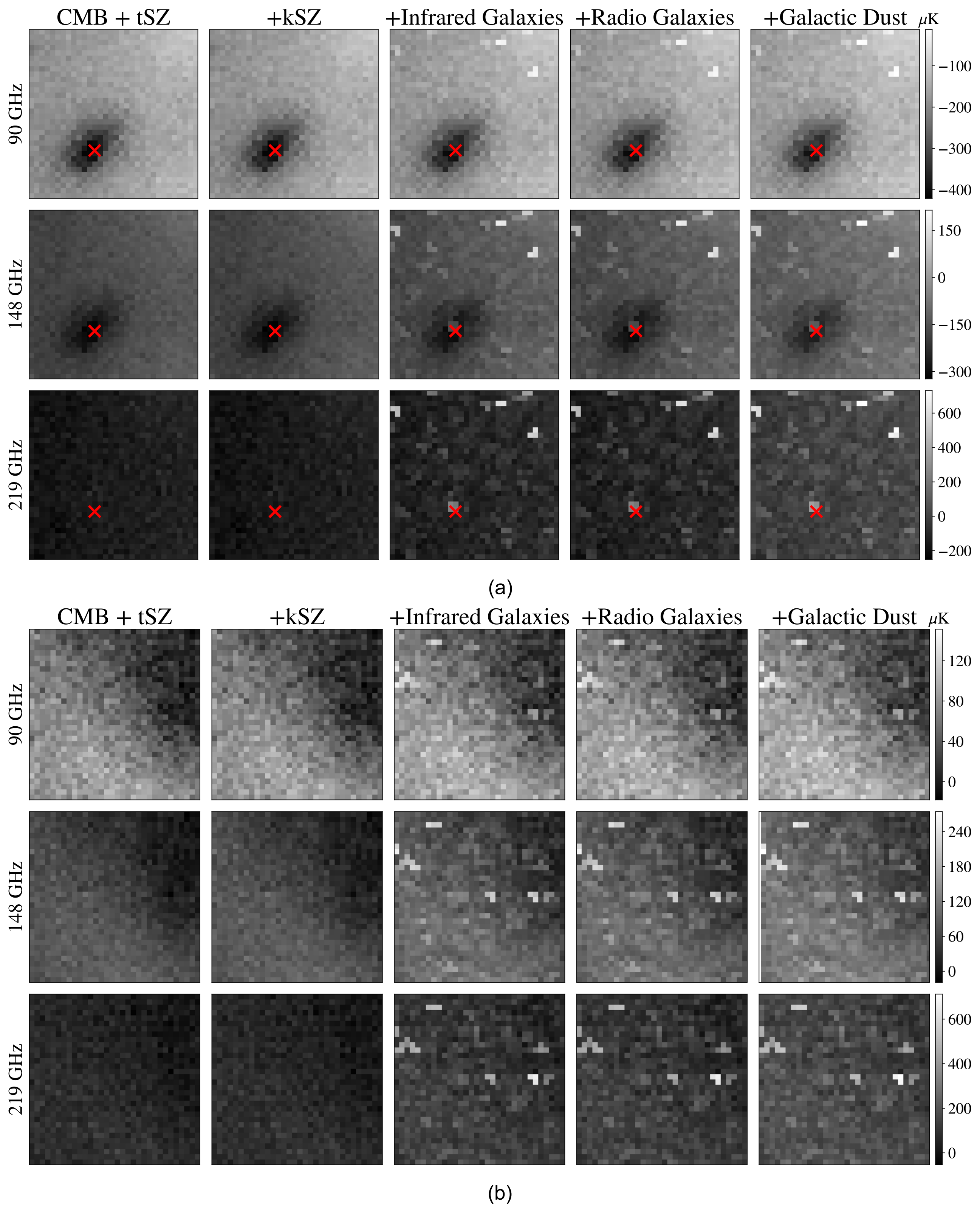


Figure 1. A positive (top) and negative (bottom) sample input to the neural network with three channels at 90GHz (first row in top/bottom), 148 GHz (second row), and 219 GHz maps (third row). We show the cut-out with different components (CMB, tSZ, kSZ, IR galaxies, radio galaxis, galactic dust) gradually added here. All cut-out contain instrument noise, which is 2.8, 2.6, and 6.6 μK-arcmin for 90, 148, and 219 GHz maps, respectively, consistent with projected performance for the upcoming CMB experiment SPT-3G. Each pixel is 0.25 × 0.25 arcmin. Each cut-out has 32 × 32 pixels. *x*-axis is RA, and *y*-axis is Dec. The position of the cluster is market with a red 'X'.

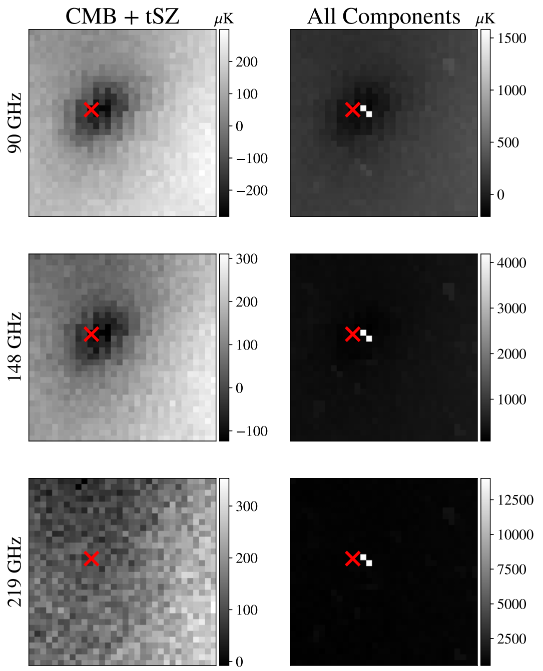


Figure 2. We provide an example cut-out that contains a cluster, and also happens to host a high-flux infrared galaxy. The column on the left is the image with only the CMB+tSZ components, and the column on the right is the image with all components adding on the kSZ, infrared and radio galaxies, and galactic dust. Similar to Fig. 1, top to bottom rows correspond to the 90, 148, and 219 GHz bands. The addition of a high-flux infrared galaxy increases the colourbar range by an order of magnitude, weakening the tSZ signal from the galaxy cluster relative to the rest of the cut-out. The CNN takes input images, such as those in the right-hand column, with no pre-processing.

any such manual pre-processing and takes as input, pixel data from images like the high-flux galaxy containing cluster.

3 METHODS IN SZ CLUSTER-FINDING

We next describe distinct methods of SZ galaxy cluster detection – first the canonical method, MF, and then an alternative method via Deep CNNs. We are developing the methods in simulations, and therefore know the ground truth of all the clusters, including, namely, the properties of their underlying dark matter haloes. To assess cluster-finding efficacy, we must match clusters to haloes, which we describe after the detection methods. We then note the differences and similarities of the methods in principle, in practical implementation, and compare their efficacies.

3.1 Matched filter

We applied a matched-filter-based method to the simulated microwave maps. The matched-filter is a standard method to identify galaxy clusters from their SZ signature, providing a baseline to compare our deep learning model. It uses the spectral and spatial characteristics of the SZ signal from galaxy clusters to differentiate them from noise. This method was developed in Melin et al. (2006). We apply the MF method as applied to data collected with the South Pole Telescope [e.g. Vanderlinde et al. 2010 (hereafter V10), Bleem et al. 2015, and Huang et al. 2020 (hereafter H19)]. In this section, we provide a descriptive overview of the method and leave more mathematical and detailed treatments to the above references.

An MF is a Fourier-domain filter that optimally filters for a given input profile, and a given set of noise power spectra. The input profile is weighted by the inverse of the noise power spectra. This weighting scheme provides an optimal filter under two conditions: the noise terms are Gaussian and stationary (that is, the noise power spectra do not vary over the map). In order to maximize signal to noise on SZ clusters, our MF also combines the maps across frequency bands. For each band, we model the following noise terms:

CMB: The CMB contributes noise primarily at large angular scales. We calculate the CMB power spectrum using the Code for Anisotropies in the Microwave Background (Lewis, Challinor & Lasenby 2000) with the best-fitting ΛCDM parameters from WMAP7 + SPT (Keisler et al. 2011; Komatsu et al. 2011). We also include a term for the kinematic SZ, based on Shirokoff et al. (2011). These terms have the same amplitude in each band. The simulations used in this work generate the CMB realization based on WMAP5 ΛCDM parameters.

Point Sources: The brightest point sources contribute noise on the smallest scales, but the background of unresolved sources contributes to a much broader range of scales. We model the combination of two source populations: one from dusty galaxies, which are brighter at higher observing frequencies; and one from radio-loud sources, which are dimmer at higher observing frequencies. We assume that the spatial power spectrum of the combined source population is flat in C_ℓ-space. Each frequency is normalized such that $D_\ell = \ell(\ell + 1)/(2\pi)C_\ell = (2.7, 8.8, 71.)\ \mu\mathrm{K_{CMB}}^2$ at (90, 148, 219) GHz and $\ell = 3000$. These values were chosen to match the amplitudes of the point source power spectra from the simulated maps used in this work.

SZ Background: There is a contribution to the noise from dim, undetected SZ sources. We model this as flat in D_ℓ-space, with $D_{3000} = 3.6\ \ \mu\mathrm{K_{CMB}}^2$ at 90 GHz, and the remaining bands scaled from this value using the non-relativistic form of the SZ frequency spectrum.

Instrumental Noise: Instrumental noise is also flat in C_ℓ-space, with amplitudes given in Section 2.

Point sources are the primary source of non-Gaussian non-local noise. In previous applications of this method, point sources with high SNR were masked to avoid the false detections they cause. Due to the low noise levels assumed, the threshold for point source masking is much lower, which would lead to masking a significant fraction of the map area. Instead, we have subtracted point sources that are brighter than 2 mJy before filtering the maps. This is consistent with the projections for finding clusters using the SPT-3G receiver.

To find clusters, we filter the maps using a projected β-model (Cavaliere & Fusco-Femiano 1976) with $\beta = 1$. V10 explored more complex models, but found no increase in the efficacy of the MF. The β-model has one free parameter, which sets the angular scale of the profile. We create 12 different filters to account for clusters of different angular sizes.

Our filtering procedure produces a set of 12 maps (one for each profile) in signal-to-noise units. We run a peak-finding algorithm that groups connected pixels above a given SNR threshold. Each group's position is calculated by taking the SNR-weighted mean of the pixel positions. Finally, for groups with detections in multiple output maps, we take the group with the highest SNR. These detections make up the final cluster candidate list. Each candidate has a location, SNR (which is used as a proxy for mass; see e.g. H19), and the profile that maximized its SNR.

3.2 Deep convolutional neural networks

The machine learning model we have chosen for this work is a deep CNN, which is known to perform well in image classification tasks. Compared with the MF method, neural-network-based methods require less image pre-processing, and are very flexible in that the same architecture can be used in different tasks with little modification. For example, model structures developed in classification problems are often used for regression problems. In fact, even the weights trained on one task can often help in the training of a different task. With this in mind, we will use a well-known architecture as a starting point for our network structure.

3.2.1 *Network architecture: ResNet-50*

Our network design is based on ResNet50 (He et al. 2016), which is a powerful and popular deep CNN architecture (LeCun et al. 1998).

A typical *neural network* takes in a multidimensional array (e.g. a red giant branch image as a 3D array), and outputs a scalar or array, depending on the use case. In our binary classification case, the output is a real number denoting the 'score', which should be a rank-preserving function of the probability that a cut-out contains a target. A *deep neural network* usually consists of several layers, and in the simplest case, each layer has several *neurons*, each of which computes a linear combination of its inputs, and then passes the value through a non-linear function. This output then forms the input of the next layer.

A deep convolutional neural network (DCNN or CNN) is characterized by the presence of convolutional layers, possibly among others. A convolutional layer takes in an image or a batch of images as the input. A layer contains small (usually several pixels by several pixels) learnable filters, and convolves each filter with the input using a sliding window, usually with a stride, to get a feature map. This feature map output is then passed through an element-wise non-linear function. Beyond the first layer, the input are feature maps generated by previous layers instead of images. A convolutional layer has a few notable distinctions compared with a fully connected layer: It reduces the number of parameters (parameter sharing), and, partly because of this, its output is less sensitive to translation in the input (*translation invariant*), which proves to be a very desirable property in applications like image classification. After a convolutional layer, a typical CNN will have a pooling layer that takes the maximum activation in a small region (usually 2 × 2 pixels). Strides in the convolutional layer and pooling layer are two ways to down-sample the feature maps, and by repeating these steps, the neurons in the deeper layers can 'see' a larger and larger region of the original image.

If layers are just stacked to build a very deep neural network with no other modifications, we often see a degradation in accuracy, since the parameters become very difficult to optimize as the network becomes deeper. Note that while the number of parameters grows with the number of layers, the phenomenon referred to here is manifested in a higher *training* error, so this is not related to overfitting. A *Residual Network* (ResNet) is a DCNN that aims to mitigate this issue. An ResNet has several residual blocks, each consisting of a few convolutional layers and a shortcut connection from the first layer to the last within the block. The idea is that the shortcut connection behaves like an identity function, and the layers inside a residual block will only need to add what has not been learned by layers prior to the block: the 'residual' of this identity map. Shortcut connections then allow us to make models deeper without degrading performance. This is because simply adding identity layers to a model (that is, if the additional layers that are skipped add nothing new) will not increase the *training* error.

ResNet50, proposed in the original ResNet paper, is a popular ResNet model used in many different image classification tasks. It has 50 convolutional or fully connected layers in total, grouped into several residual blocks. The depth of ResNet50 was optimized to train on relatively big cut-out. However, to cater to the small image sizes in our data set, we effect the following changes:

(1) We remove the first convolution layer (kernel size 7, stride 2, padding 3), and replace it with 2 smaller convolution layers with kernel size 3 and padding 1. The first convolution layer has a stride of 1, whereas the secondly has a stride of 2. Like the original ResNet50, we have batch normalization and ReLU after each convolution layer, and we also have a max pooling layer of stride 2 after the secondly convolution layer.

(2) Instead of having a fifth stage, we put two fully connected layers of size 256 on top of stage 4's output, and put a prediction and softmax for two classes afterwards.

Fig. 3 shows the structure of the network visually.

3.2.2 *Training details*

As mentioned earlier, our data is highly unbalanced. The ratio of negative cut-out to positive cut-out is over a factor of 50. To address this challenge, we experiment with several training strategies. We first manually assign different weights to positive and negative samples to give comparable importance to the positive and negative subsamples. Similarly, we oversample the positive samples to establish an effective negative-to-positive ratio closer to 1. However, these strategies result in a trade-off. If we set the effective ratio low (closer to a balanced sample), then the network carries a bias unrepresentative of the real sample. On the other hand, if the effective ratio is too high, then the network is very difficult to train. This difficulty comes from the fact that a blind guess of all negatives can already produce a good accuracy and loss. Moreover, iterating the process over different ratios is very time consuming. To solve this problem, we devise the following strategy:

(i) Let us denote r as the effective ratio of the number of negative cut-out to the number of positive cut-out. S_r is a sample where the positive cut-out are oversampled such that the negative-to-positive effective ratio is r. One can think of this as a data loader of the same training set that gives positive cut-out a higher probability of being sampled.

(ii) We start by training the model on S_1, a sample where there is an equal probability of drawing a negative or a positive cut-out. We train the model with a batch size of 128, using cross-entropy loss. We evaluate the loss on the oversampled validation set every 100 batches and continue training until the loss on the validation set converges (i.e. does not decrease for 1000 batches). We consider this as an epoch. Note that each *epoch* can have *different sizes.* As an example, the first epoch took 2400 batches to complete.

(iii) In each epoch, we use stochastic gradient descent for training with an L2 weight decay rate set to 0.003. We set the learning rate to 5×10^{-3}, with a linear warm-up schedule in the first 500 batches in each epoch (linearly increasing from $5/3 \times 10^{-3}$ to 5×10^{-3}). After 1000 batches, we decrease the learning rate to 5×10^{-4}.

(iv) At the end of each epoch, we save the best model weight so far, feed the original validation set (no oversampling involved) to this model and save the results. We then increase r by one and continue training starting with this model weight.

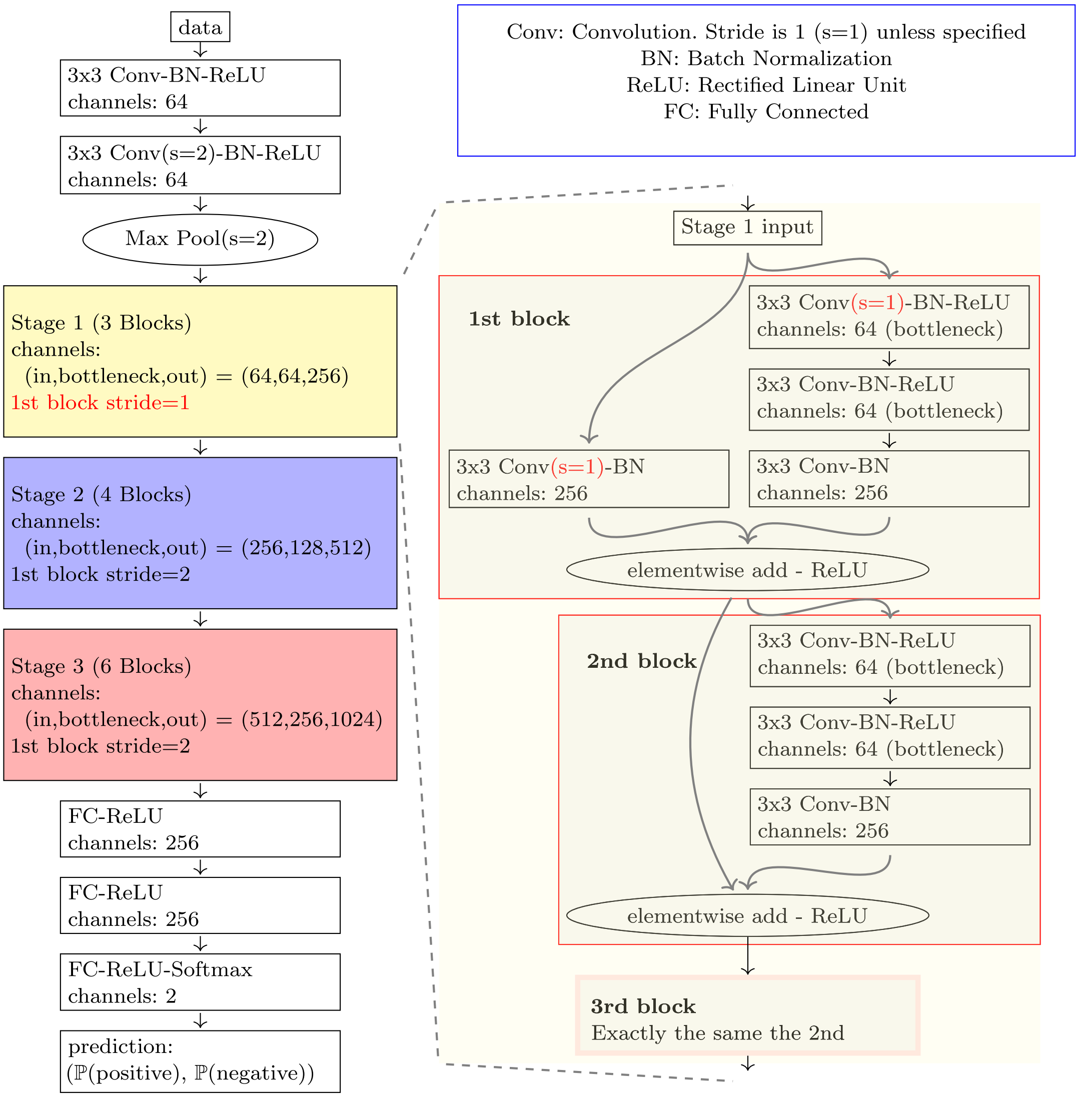


Figure 3. Visualization of the modified ResNet. There are three stages with residual blocks, mostly like the original design, and we only show the details of the first one for simplicity. In the first block within each stage, we have two Conv-BN-ReLU layers with a small number of channels (bottleneck) for better learning, and another Conv-BN layer with more channels, whose output is taken a sum with the skip connection. In the following blocks, the skip connection does not perform any operation.

(v) After training on S_{20}, we go back and pick the model weights that best perform on the entire validation set. In this experiment, the best-performing model that we select is from S_{16}.

We can consider this to be a dynamic stratified sampling. The entire training process, including evaluation of the test and validation sets at the end of each ratio, took approximately 8 h on a GTX 1060 GPU. We first show some of the training and validation cross-entropy loss statistics during the training process in the top panel of Fig. 4. We also overlay the ratio at each iteration in the plot. In the bottom panel of Fig. 4, we show the training and validation accuracy adjusted for the ratio, along with the ratio at each iteration. Unless specified, all numbers reported are for cut-out with all noise components. We perform a breakdown of these results in Section 4. Finally, we perform the classical Platt Scaling calibration method for binary classification, (Platt 1999), on the final output, so that we can interpret the output as probabilities. This is equivalent to running a logistic regression using output on the validation set. All CNN outputs in this paper are the calibrated numbers.

3.3 Matching cluster detections to haloes

The MF assigns detections to specific coordinates, while the CNN detects whether or not each cut-out contains a cluster. To compare two methods, we need to match MF detections to cut-out.

MF-identified clusters whose centers lie near the edge of a cut-out may have an identified centre just inside the cut-out, but a true centre outside the cut-out region. In these cases, the cut-out containing the MF identified cluster position would not be a ‘cluster-containing’ cut-out, despite the actual correspondence. These cases would lead to a comparison that understates the MF performance. To ensure correspondence and to make a fair comparison between methods, we perform the following:

(i) We match MF detections to the largest cluster within a 1-arcmin radius of the detection. The radius was chosen to maximize MF’s performance.

(ii) If the detection corresponds to a real cluster, we match the detection to the cut-out corresponding to the true cluster centre. If

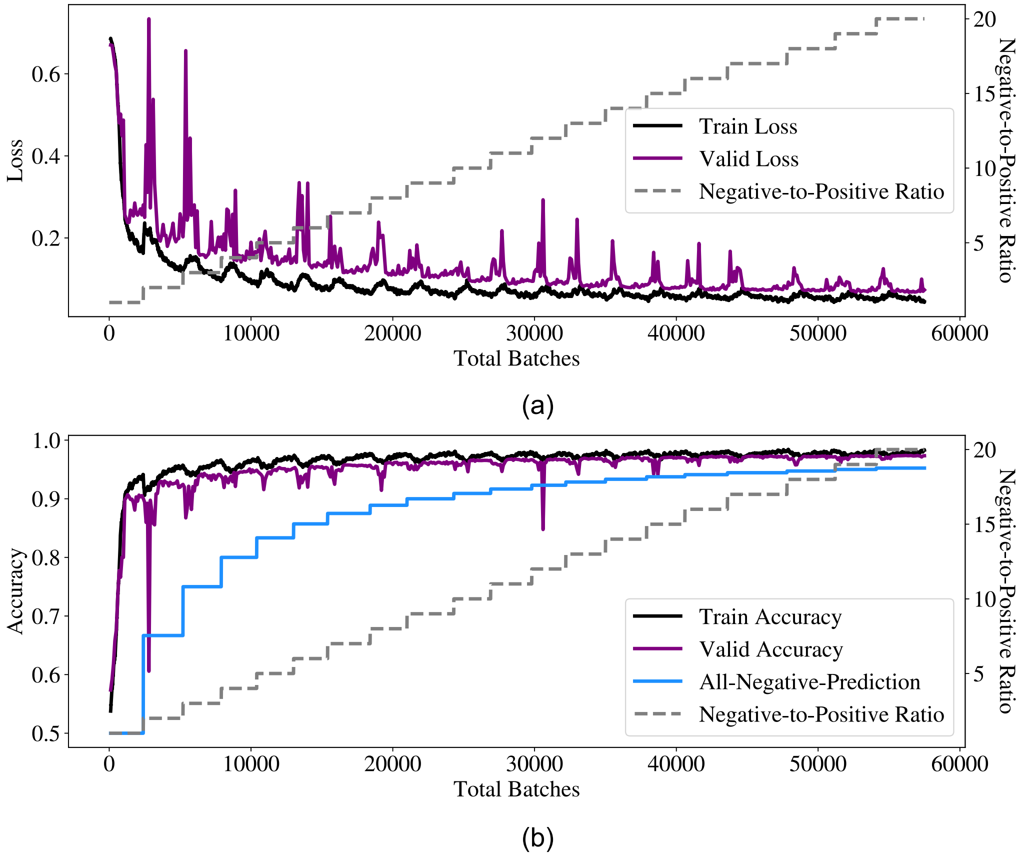


Figure 4. In this figure, (a) shows the loss history, and (b) shows the accuracy history as a function of the number of batches (iterations).

Table 1. A summary of all the confusion matrices. All confusion matrices are given by selecting the threshold for each method on the validation set to maximize F1 score, and then apply the thresholds on the test set.

Method	Prediction	Has cluster (Truth)	No cluster (Truth)	Precision (Purity)	Recall (Completeness)	F1
MF	Has cluster	1133	729	0.61	0.61	0.61
	No cluster	712	102609			
CNN	Has cluster	1118	780	0.59	0.61	0.60
	No cluster	727	102558			
MF + CNN Ensemble AND	Has cluster	1283	625	0.67	0.70	0.68
	No cluster	561	102713			
MF + CNN Ensemble rankproduct	Has cluster	1429	946	0.60	0.77	0.68
	No cluster	416	102392			

the detection does not correspond to a real cluster, we match the detection to the cut-out containing the MF identified centre.

(iii) We assign the corresponding SNR of the MF detection to the cut-out to which the detection is assigned in the step above. If multiple detections map to the same cut-out, the largest SNR is assigned to that cut-out.

4 RESULTS

Below, we present results of both methods applied to the mock data set with all noise components. Table 1 summarizes all the confusion matrices after appropriate threshold selection. These results use thresholds that maximize the F1 score of the methods applied to the validation set (see Section 4.2 for the detailed procedures). Solely focusing on purity or completeness as a metric can be misleading; the F1 score offers a fairer platform for comparison. We see that the F1 score performance of MF and CNN are comparable. Additionally, the Ensemble methods can significantly improve the performance measured by any of the metrics. The Ensemble methods either simultaneously improve both purity and completeness (AND ensemble), or greatly improve one without sacrificing the other (PROD ensemble).

4.1 Ambiguity in definition of true positives

In this subsection, we describe one possible systematic in our comparison of the two methods. To assign a performance metric to a cluster detection method, we need a mass threshold for a ‘true cluster’. There are otherwise multiple haloes in any given patch of the simulated sky, some of which sit above that threshold and some of which sit below. Due to hierarchical structure formation, there are more objects below a mass cut than above. An artefact of a cut-off is that there will likely be some confusion near the selected threshold.

In particular, the MF does not assume a clean cut-off in mass when identifying a cluster, but rather identifies a cluster according to the SNR. The SNR produced by the MF scales with cluster mass (and weakly with redshift, see Bleem et al. 2015, and H19) with a ~20 per cent lognormal scatter in the scaling relation (see e.g. Bocquet et al. 2019). To briefly describe the relationship, at fixed mass, the temperature of the cluster gas increases with redshift therefore increasing the tSZ as well. And, at larger angular sizes, the cluster size becomes comparable to the smallest CMB fluctuations. The MF downweights such modes, ultimately contributing to a redshift dependence in the MF performance. We included a redshift threshold of 0.25 (see Section 2.2) to somewhat address such issues.

We can conjecture that the CNN will also get confused by an object whose mass is near the threshold. We assume a mass threshold when labelling a cut-out as ‘cluster-containing’. But some haloes below the mass threshold may have a higher MF SNR or CNN score than haloes above the detection threshold. This ambiguity affects the precision of both methods. None the less, we fix the threshold of each method in our comparison in an effort to fairly assess each.

4.2 F1 score

In Section 4.1, we discussed how the ambiguity in defining true positives in the sample might affect the metrics for model performance. Another aspect of performance metric comparisons is assessing the trade-off that occurs when we vary the classification threshold; we can increase the number of true positives, improving the completeness of a sample, at the cost of increasing the number of false positives, worsening the purity of a sample. The cost–benefit comparison becomes more complex in an imbalanced data set, where there is a large difference between the number of true positives and the number of true negatives. This is the case for the number of cluster containing cut-out; galaxy clusters are relatively rare objects in the overall footprint of the sky.

The F1 score is a common choice of performance metric in problems with imbalanced samples. If a classifier blindly predicted everything to be positive (or negative) and there were many more true positives (negatives) in a sample, we would see a misleadingly high accuracy. The F1 score accounts for the imbalance, defined as the harmonic mean of precision and recall:

$$\mathrm{F1} = \frac{2}{\mathrm{precision}^{-1} + \mathrm{recall}^{-1}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}, \quad (1)$$

where ‘TP’ is true positive rate, ‘FN’ is false negative rate, and ‘FP’ is false positive rate. In our particular use case, a classifier could identify all cut-out as non-cluster-containing and achieve a high accuracy. But the F1 score would equal zero, indicating the lack of information provided by this classification scheme. A high F1 score indicates a classifier with high precision while detecting as many clusters as possible despite the imbalance between the two classes in our data set. We therefore emphasize the F1 score in subsequent discussions of model performance.

Note we advocate for F1 as a useful metric, regardless of the method classifier. We can use the F1 score to determine an optimal threshold for positive predictions for a given classifier. We do that by looking at how F1 varies as a function of the threshold for positive identification of that classifier. The maximum F1 score corresponds to a threshold that maximizes positively identified clusters while minimizing false positives, useful for any cluster analysis that requires further follow-up observations. The peak F1 score also provides a single metric to compare any classifier.

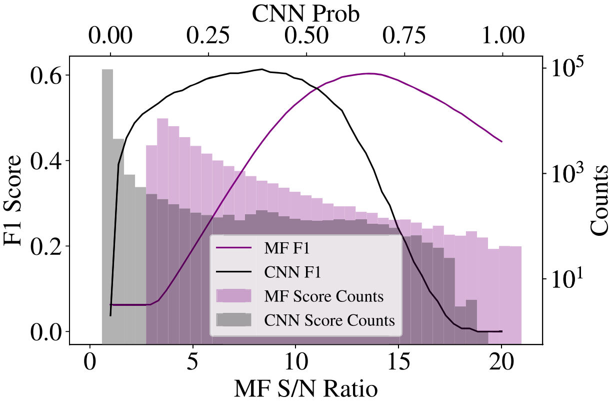
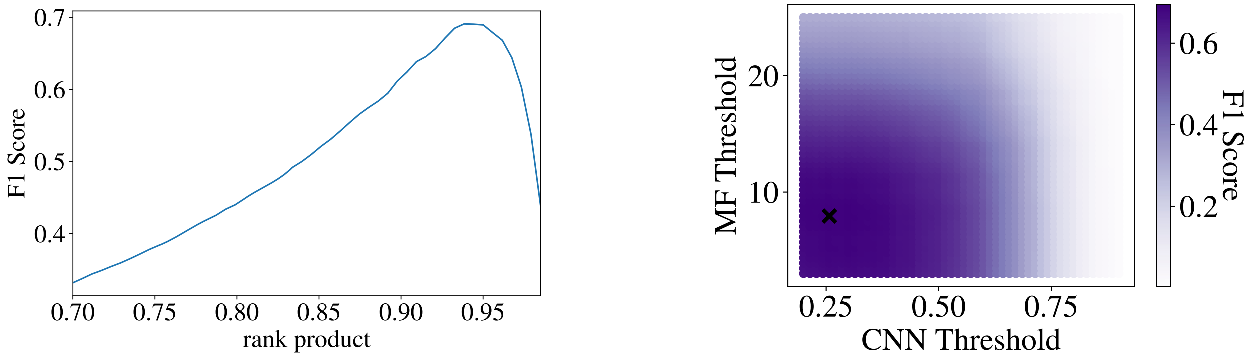


Figure 5. F1 score as a function of the threshold for S/N ratio from MF and probability from CNN (on the validation set). We pick the thresholds for each method here and later evaluate them on the test sets. The distribution of the scores (S/N ratio for MF, and probability for CNN) are shown in the background. We also did not include any MF SNRs below 3.0 because there are too many of them.

Fig. 5 shows the F1 score on the validation set of MF and CNN as a function of the threshold for positive identification. The MF has a threshold at some SNR, and the corresponding F1 score is shown with the purple line. The CNN has thresholds of probability output with the corresponding F1 score shown in the black line. The shaded purple and black histograms, respectively, show the number of objects within a given SNR bin and probability bin.

From this figure, we see that the peak F1 score of MF applied to the validation set is 0.60, which corresponds to a SNR threshold of 13.4, the 98.2 per cent rank percentile of the MF SNR of all cut-out. That same SNR threshold results in an F1 score of 0.62 when applied to the MF SNR of the test set. We can also determine that the peak F1 score of CNN applied to the validation set is 0.60, corresponding to a probability threshold of 0.367, the 98.0 per cent rank percentile of the CNN scores of all cut-out. That same probability threshold results in an F1 score of 0.60 on the test set. Using the F1 score to compare the two methods, we see that they do comparably well. We summarize the metrics in Table 1.

Note, if we consider true-positives with lower cluster mass thresholds, the F1 score of the MF as a function of SNR would shift. At fixed SNR, as we decrease the mass threshold, purity (precision) will increase, but completeness (recall) will decrease. As a result, the direction of the shift of F1 at each SNR value will be determined jointly by these two effects. But, the peak of the F1 score curve, i.e. the optimal SNR threshold, will systematically shift to lower SNR values since true clusters include lower mass objects. However, our analysis does not include these results, since we want to compare the MF results with a given CNN model that was trained at a given mass and redshift threshold. It would be computationally expensive to iteratively train more CNN models to compare with the MF performance at different mass thresholds. Furthermore, we will have multiple clusters per cut-out, complicating our comparison metrics (see discussion in Section 2.2).



(a) F1 score on the validation set if we form a single score by multiplying the rank percentile of CNN score and MF S/N ratio. The validation set selects the rank product threshold to be 93.8%.

(b) F1 score on the validation set if we require both MF and CNN's score to pass a certain threshold. The optimal point is when CNN probability is greater than 0.26 and MF S/N ratio greater than 7.9 (marked with a black cross).

Figure 6. Here, we show the performance of two Ensemble methods: EnsemblePROD (6a) and EnsembleAND (6b). Their performances are very similar: Both achieve an F1 score of 0.69 on the validation set, and **0.68** on the test set. Both methods, however, noticeably outperform CNN or MF standalone.

The F1 score also provides a means to combining the two methods for cluster identification. We evaluated two ways to combine the predictions, an **EnsemblePROD** method and an **EnsembleAND** method described below.

(i) **EnsemblePROD** method: Here, we form one single score by multiplying the rank- percentile of CNN and MF scores, similar to an interaction term in regression tasks. Suppose we have N cut-out, and one CNN score p_i^{CNN} and one MF SNR s_i^{MF} for each cut-out $i \in [N]$. The CNN rank percentile is defined as $q_i^{CNN} := \frac{1}{N}\sum_{j\in[N]} \mathbb{1}[p_j^{CNN} \leq p_i^{CNN}]$, and the MF rank percentile is defined similarly – just the percentile its MF SNR falls in the sample. The combined rank percentile is simply their product: $q_i^{Ensemble} := q_i^{CNN} q_i^{MF}$.

(ii) **EnsembleAND** method: Here, we classify a cut-out basing on a logical AND condition basing on CNN and MF scores. Unlike previous methods, this method requires two thresholds (one for CNN and one for MF), and classify a cut-out as positive if both thresholds are met.

We evaluate the metrics for the combined classifiers using the same metrics and procedures as we did for the individual methods. We choose the threshold on validation set and evaluate the performance on the test set.

Fig. 6(a) shows the results of the F1 score curve for **EnsemblePROD** as a function of the rank product, defined above. The peak of the curve corresponds to the rank product that results in the maximum F1 score. We summarize the metrics in Table 1. Recall that the thresholds picked on the validation set for the individual methods are 98.2 per cent for MF and 98.0 per cent for CNN. The new rank-product threshold, 93.8 per cent, is at the 97.7 per cent – percentile of all rank-products, which is very close to the two stand-alone threshold ranks. In other words, the threshold of **EnsemblePROD** is at a similar percentile (compared with MF and CNN), which, due to the size of the sample, translates to a few hundreds more positive predictions. The overall F1 score, as shown in Table 1, is greatly improved. This suggests that **EnsemblePROD** successfully incorporates information from both methods. We could also qualitatively see such an example in Fig. 7, where **EnsemblePROD** identifies some cut-out missed by either MF or CNN.

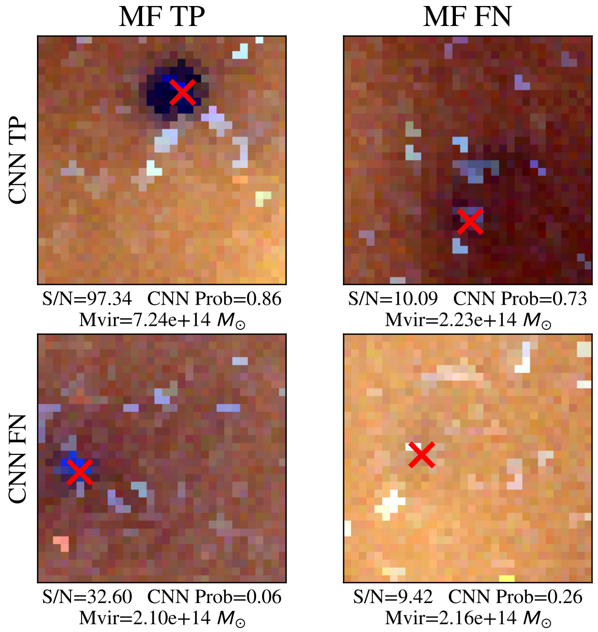
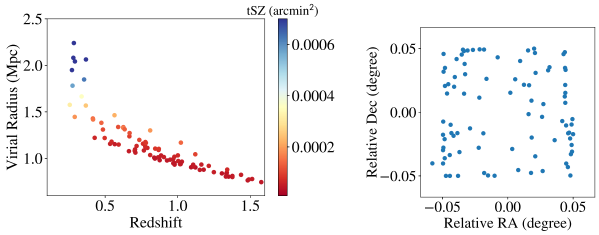


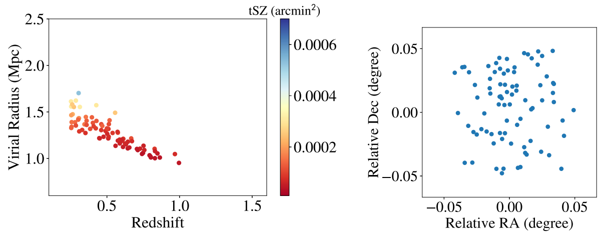
Figure 7. Examples of cluster-containing cut-out correctly classified by the combined method EnsemblePROD. The pictures are generated by treating the three channels (90/148/219 GHz) of input as RGB channels. For each channel, the pixel value is re-scaled to [0, 255], with the same channel (e.g. red channel) of all cut-out sharing the same colour scale. The top (bottom) row has high (low) probabilities assigned by the CNN classifier, which we identify as a CNN 'true positive'/TP (CNN 'false negative'/FN). The left (right) column has high (low) signal to noise as measured by the MF algorithm, which we identify as a MF 'true positive', MF TP (MF 'false negative', MF FN). While the bottom right-hand panel might be identified as both a CNN FN and an MF FN, the EnsemblePROD correctly identifies this cut-out because of its relatively high ranking in the CNN and MF scores. Note that although the score given by CNN in the bottom left is only 0.06, we can see that it is a high percentile already because CNN gives most cut-out a score of essentially 0 (see the histogram in Fig. 5).



(a) The virial radius as a function of redshift colorcoded by their tSZ, for clusters with a high MF SNR and low CNN score.

(b) Position of the clusters in the cutout.

Figure 8. A total of 89 clusters with a high MF SNR, but low score from the CNN. These clusters span the mass and redshift range, but tend to sit close to the edges of the cut-out that the CNN evaluated.



(a) The virial radius as a function of redshift colorcoded by their tSZ, for clusters with a high CNN score and low MF SNR.

(b) Position of the clusters in the cutout.

Figure 9. A total of 83 clusters with a low MF SNR, but high score from the CNN. There is no cluster with redshift above 1.0 with low MF S/N in this case, because for the same mass, S/N tends to be higher for higher redshift objects. This is discussed in 3.1 and can be verified in Fig. 14.

Fig. 6(b) shows the results of the F1 score for **EnsembleAND**, colour-coding the F1 score in the space of the MF and CNN thresholds for positive identification. We summarize the metrics in Table 1. As expected, the two thresholds are significantly lower than the counterparts in the stand-alone MF or CNN methods, which is a direct result of the AND logic we apply here. This, however, suggests that CNN predicts certain low-MF-SNR cut-out with higher probabilities and the MF measured a higher SNR for certain low CNN probability cut-out; the two methods complement one another.

We now compare the performance of the ensemble methods with the standalone on the test set. The F1 score allows for an overall comparison along a single dimension. Both ensemble methods, with F1 scores of 0.68, lead to significant improvement over stand-alone MF or CNN method, with F1 scores of 0.61 and 0.60, respectively. This illustrates the strength in combining methods, particularly with the use of F1 scores. In Section 4.3, we investigate how the two methods complement one another.

4.3 Comparison of performance

We compare the CNN and MF performance using thresholds that maximize the respective F1 score of each method. With this, we arrive at the precision and recall values provided in Table 1. The first two rows of the table show the stand-alone performance of each the MF and CNN. The precision of the CNN is slightly worse than the

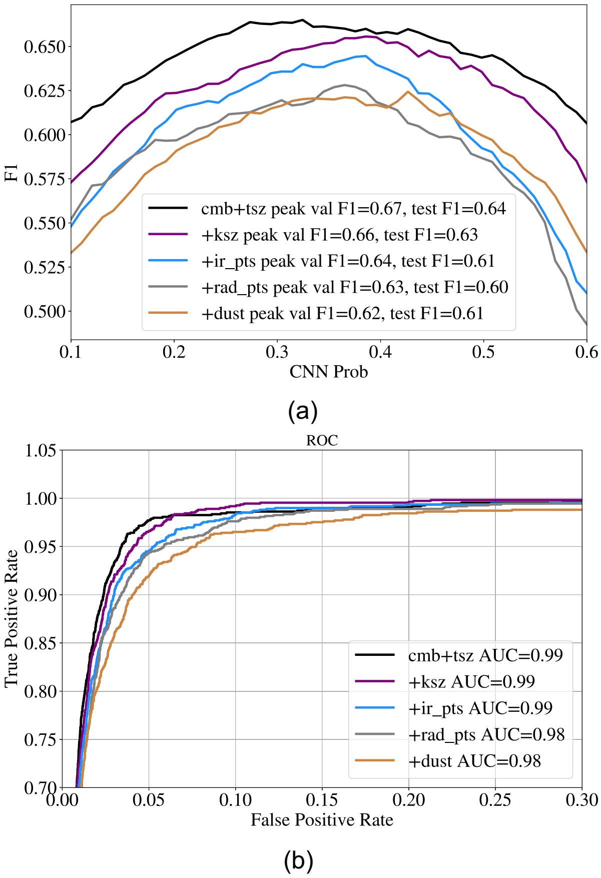


Figure 10. Metrics: (8a) shows the curves of F1 score as a function of the threshold on **CNN probability** for different levels of noises. (8b) shows the ROCs of CNN's predictions on different levels of noises. Both are on the validation set.

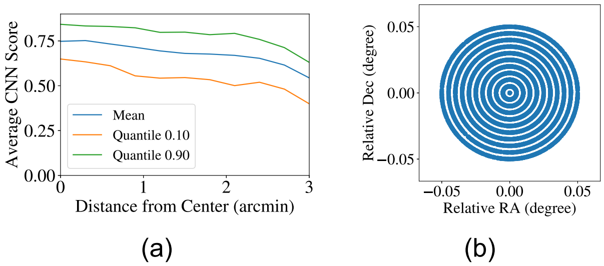


Figure 11. To investigate whether an edge effect is indeed the main driver of the low prediction probability on these clusters, we randomly regenerated cut-out with these clusters located at varying distances from the centre of the cut-out. On average, the prediction value given by CNN is a decreasing function of the distance between the centre of the cluster and the centre of the cut-out. We first divide the clusters in Fig. 8 into 11 groups. For each group, we randomly regenerate cut-out for these clusters such that the cluster has different distance from and angle relative to the cut-out centre (Middle), and then compute the curve of prediction score as a function of distance from centre for these randomly generated cut-out. Finally, we plot the mean, 10th percentile and 90th percentile (over the 11 groups) of these curves in Fig. 11(a).

MF, and the recall slightly improved. We want to emphasize again that only looking at either precision or recall here would be misleading, as our evaluation metric is mainly F1 score. For example, it is true that the **EnsemblePROD** method does not improve precision, but that is because it trades (on the validation set) precision for a much higher recall to improve the F1 score.

Given the differences inherent to each method, we further explore if the two methods are complementary to one another, preferentially selecting (or missing) clusters with particular attributes. To further investigate, we looked at two sets of clusters in the extremes of the CNN and MF performance:

(i) **Clusters with high MF SNRs but low CNN scores:** The left-hand panel of Fig. 8 shows $R_{\rm vir}$ as a function of cluster redshift, colour-coded by $t_{\rm SZ}$. We show the clusters whose CNN score is below 0.21 (0.5 before calibration) and MF SNR above 15.4. The right-hand panel shows the position of each cluster within the cut-out. These clusters are very close to the edge of the centre of the cut-out, which is a 6 × 6 arcmin region, appearing as 'almost' negative samples. To investigate this edge effect further, we randomly re-generated cut-out with clusters with different distances to the centre and evaluated the CNN on these new cut-out. We discuss this 'edge effect' in Section 4.4.2.

(ii) **Clusters with high CNN scores but low MF SNRs:** The left-hand panel of Fig. 9 shows $R_{\rm vir}$ as a function of cluster redshift, colour-coded by $t_{\rm SZ}$. We show the clusters whose CNN score is above 0.47 (0.8 before calibration) and <11.3 MF SNRs. The right-hand panel shows the position of each cluster within the cut-out. Compared with clusters with low CNN scores, these are closer to the cut-out centre on average.

Low-redshift clusters in Fig. 8(a) seem to have larger radius than in Fig. 9(a). Upon further checking, we found that the radius distribution of low-redshift clusters with low-CNN scores is similar to the population. On the other hand, MF SNR is lower for clusters with a particularly large on-sky radius (due to confusion with the primary CMB). This is likely driving such a difference, but the relationship with the virial radius is non-trivial.

4.4 CNN specific considerations

4.4.1 Effects of noise components on CNN performance

Given the slightly worse overall performance of the CNN, we examine noise components as a potential cause to the lower performance. Fig. 10 shows the performance of the stand-alone CNN with different levels of noise (different components), quantified with an receiver operating characteristic (ROC) curve, on the validation set. With each added noise component, the area under the corresponding ROC curve (AUC) decreases, as one might expect. However, the difference is not drastic varying only from AUC = 0.99 to 0.98. The resulting **test** F1s, using the thresholds achieving the peak validation F1s, are 0.64 for cmb+tsz, 0.63 for +ksz, 0.61 for +IR galaxies, 0.6 for +radio galaxies, and 0.61 for + galactic dust. In both cases, infrared galaxies and radio galaxies seem to affect the performance of CNN the most. We conclude that the CNN is relatively robust to the noise components in the microwave sky.

4.4.2 Effects of edge location on CNN performance

As described in Section 4.3, we found that many of the low-CNN-probability and high-MF-SNR cut-out have their cluster very close to the edge. We re-generated cut-out with the same cluster at varying distance from the centre, and apply the same trained network on these shifted cut-out. Fig. 11 shows the positions of these clusters in the new cut-out. Interestingly, we did find a very obvious negative correlation between the prediction score and the clusters' distance from the centre of the cut-out, as shown in Fig. 11. This suggests potential room of improvement with some different ways to apply the network. A potential algorithm to maximize the performance of the CNN could be to simply overlap cut-out by one fourth the width of a cut-out. This way, each cluster is within the middle two quarters of at least four cut-out. While several cluster containing cut-out would suffer from the edge effect, each cluster would be positively identified in some subset of the cut-out.

4.5 Completeness comparison

We now compare the cluster identification methods showing the *completeness* of each method, i.e. their performance as a function of cluster properties.

Fig. 12 shows the completeness, also known as recall, of each method as a function of cluster mass. This is the fraction of true positives identified by a given method out of the total number of cluster images evaluated. Each line colour corresponds to a different identification method: black, purple, and blue, respectively, correspond to CNN, MF, and EnsemblePROD. The solid lines show the completeness for the entire sample, and the dashed lines for clusters above $z > 0.25$. Recall, we impose both a mass cut-off $M_{\rm halo} \geq 2 \times 10^{14}\,{\rm M}_{\odot}$ and a redshift cut-off $z > 0.25$ for cut-out labelled as 'cluster-containing' in our training set for the CNN classifier. We mark this threshold with a red vertical line. For comparison, the blue histogram illustrates the underlying halo mass function of the simulated galaxy cluster sample that we used to produce the cut-out, plotted as 'Counts of Objects' (right y-axis labels) as a function of the virial mass bin. To guide the eye for comparison, the bottom panel shows the ratio of the recall of CNN and MF with the recall of EnsemblePROD with the same redshift selection.

In the range of $1.5 \lesssim \frac{M_{\rm halo}}{10^{14}\,{\rm M}_{\odot}} \lesssim 3.5$ the CNN has a higher recall. At smaller or larger masses, the MF has a higher recall. However, across the mass range, the EnsemblePROD has a systematically higher completeness compared with either the MF or the CNN. The overall improvement indicates the complementarity of the two methods. We note that the decrease of the CNN recall (the black curves) at the highest masses is due to the decreased sample size for training. Very few of the highest mass objects are at $z > 0.25$, meaning that many of the high-mass objects were also not labelled as 'cluster-containing'.

Fig. 13 shows the corresponding selection as a function of redshift. Here, we only show the completeness curves of each method for clusters that are above our mass cut for 'true', $M_{\rm halo} \geq 2 \times 10^{14}\,{\rm M}_{\odot}$. The red vertical line marks the redshift threshold for cut-out labelled 'cluster-containing'.

For clusters below $z \lesssim 0.9$, the EnsemblePROD outperforms both the CNN and the MF in completeness. At higher redshifts, the completeness of the sample selected by the MF is comparable to that of the sample selected by EnsemblePROD, both exceeding the completeness of the CNN, which remains relatively constant until $z \sim 1.5$. The EnsemblePROD has a larger F1 value with similar completeness at high redshifts as the MF, thereby illustrating that information from the CNN helps to maintain purity in cluster identification. In fact, for both lower mass and lower redshift galaxy clusters, the completeness of the sample improved when we combine the two methods. This improvement suggests that the MF and the CNN are picking up complementary features in our galaxy cluster sample.

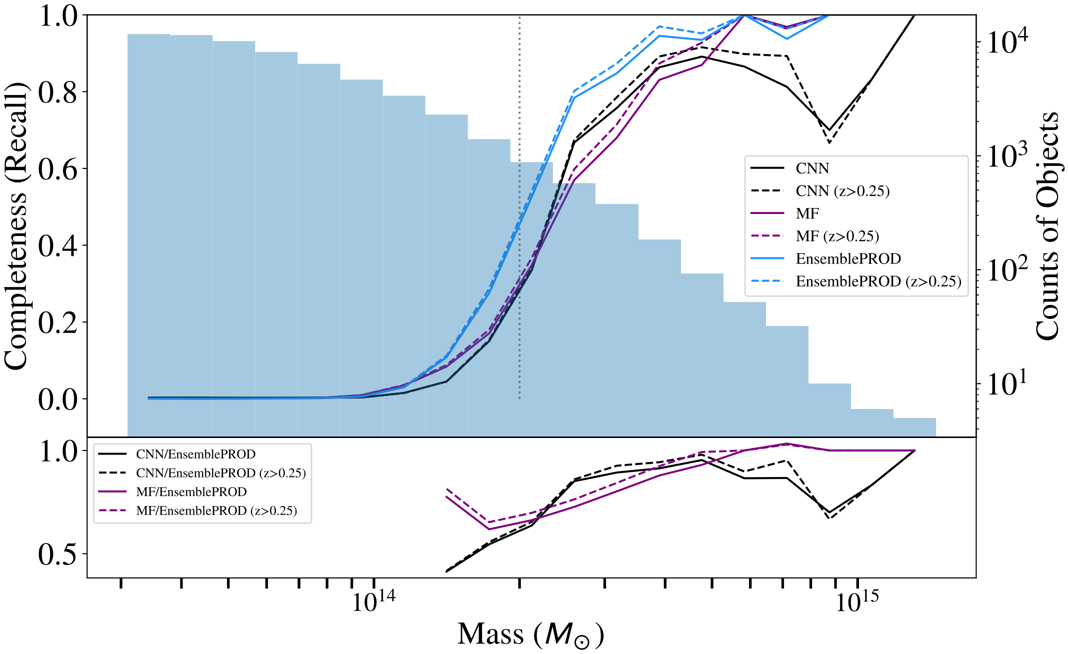


Figure 12. Top panel: The blue histogram shows underlying mass function of the simulated galaxy cluster sample in the cut-out (Counts of Objects versus Mass). The lines show the completeness curves for each method, with the dashed lines corresponding to the completeness curve for objects that satisfy our redshift threshold of $z > 0.25$ used for training. The purple dashed lines correspond to the MF, black dashed the CNN, and blue dashed the EnsemblePROD. We can see that the EnsemblePROD completeness curve sits above both methods until $\sim 4 \times 10^{14} \mathrm{M}_\odot$, where there are few objects. The decrease of the CNN curve at the highest masses could be due to the decreased sample size for training in the corresponding mass range. Bottom panel: Ratio of CNN (or MF) completeness curve to the corresponding EnsemblePROD completeness curve with the same line style and colour as the top panel. We also include the solid lines for the full sample without the redshift threshold, but since the impact of the redshift threshold is small, they are very similar.

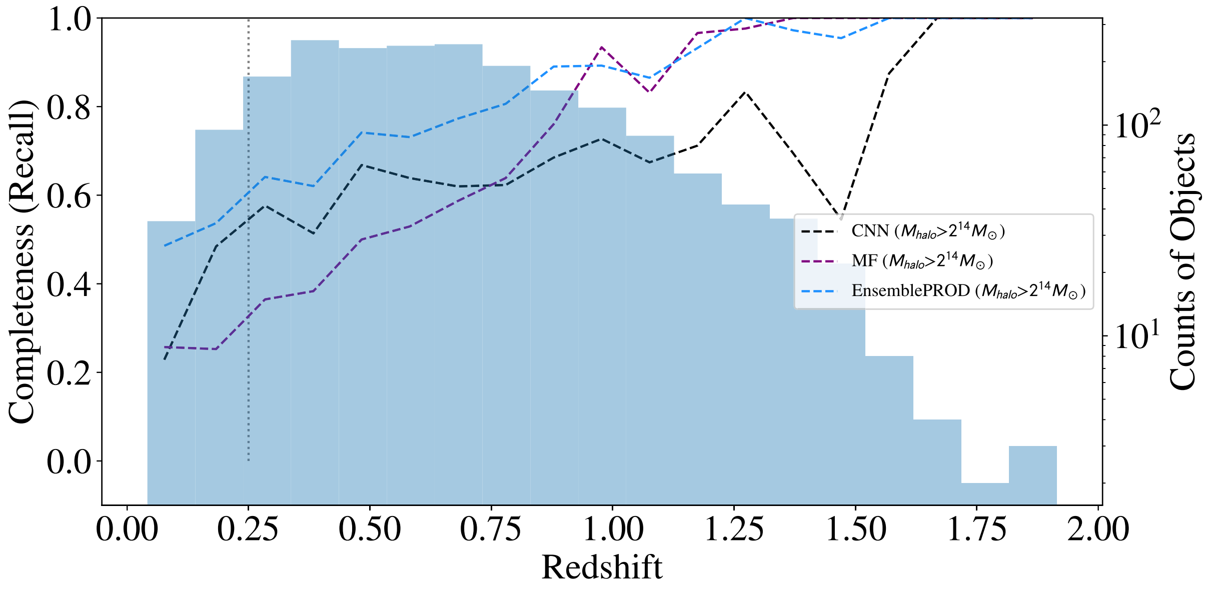


Figure 13. The blue histogram shows the underlying number of galaxy clusters in each redshift bin that satisfy our mass threshold of $M_{\mathrm{halo}} > 2 \times 10^{14} \mathrm{M}_\odot$ used for training (Counts of Objects versus Redshift). The lines show the completeness curves for each method on the subset of objects over the same mass threshold. The EnsemblePROD completeness curve sits above the other methods until $z \approx 0.9$, at which point it is comparable to MF.

To better assess the complementarity, we visualize the performance of each method in Fig. 14. Here, we show the distribution of the virial mass as a function of three cluster parameters in our sample. From top to bottom, we show the virial mass as a function of redshift, virial radius, and angular size of the cluster. From left to right, we show the per cent of true positives identified by the CNN, EnsemblePROD, and MF, where we colour code the parameter space by the true positive percentage. The darker shades of green correspond to higher true positive rates in that region of parameter space. Each pixel in the colour-coded parameter space corresponds to

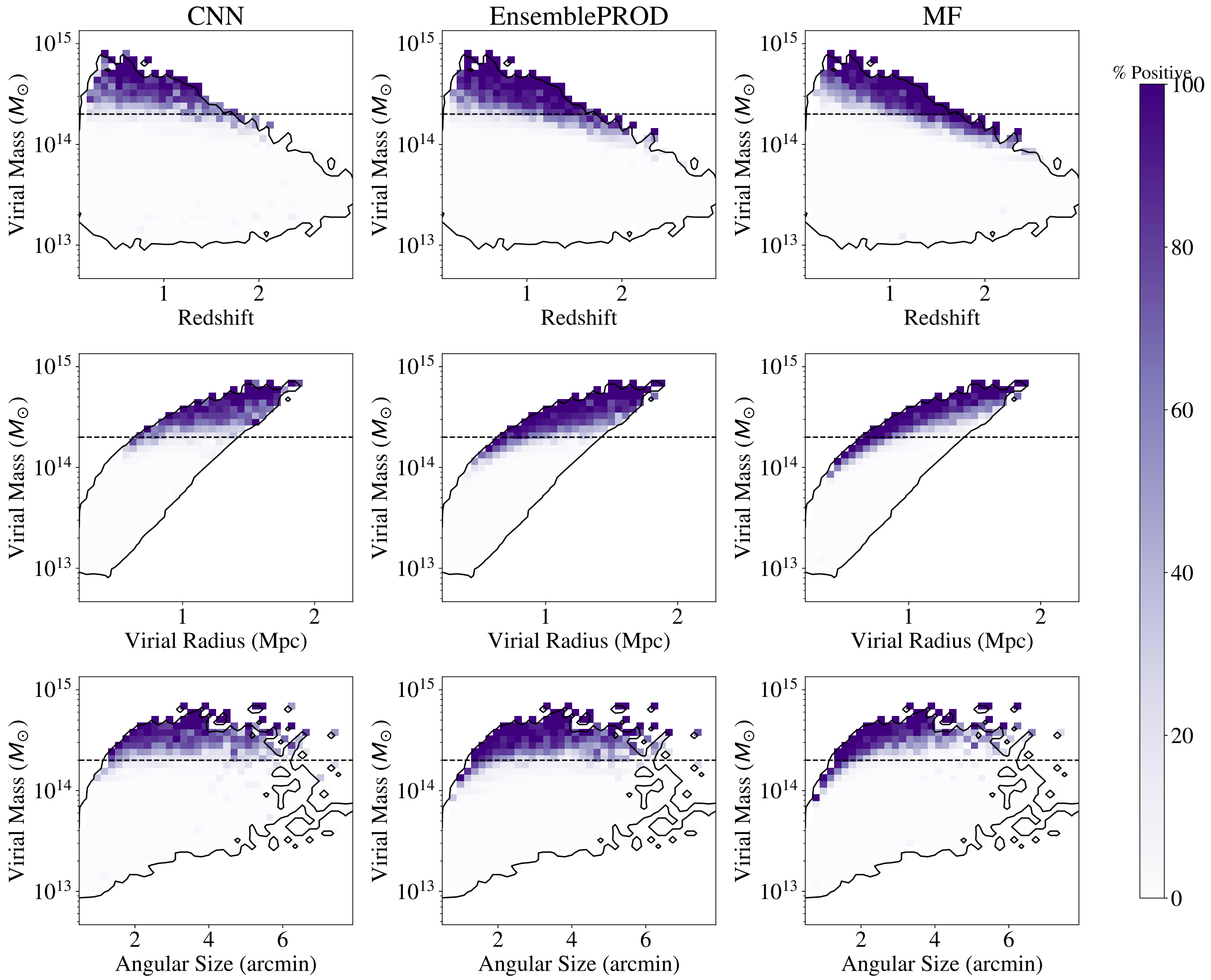


Figure 14. We plot the distribution of cluster properties and colour code by true positive identification of each method. The left-hand column corresponds to true positive identification by our CNN, centre by the rank product, and right by MF. The black-dashed line illustrates our chosen mass cut for what we labelled as true positives for the training set in our CNN. The colour coding of per cent Positive colours the per cent of clusters (cut-out containing clusters) in that cluster property bin that was positively identified by each method. The property bins have at least three clusters from which we calculate a percentage. One key feature to note is that the CNN method has a relatively stronger mass-limited selection function that is driven by our mass cut in labelling true positives. This selection does not strongly depend on redshift, virial radius, or angular size. On the other hand, the matched filter preferentially picks out lower mass clusters that are more compact, which are those at higher redshifts. The combined method, using the rank product, also has a selection function more aligned with our chosen mass cut.

at least three clusters from our sample. For reference, the horizontal dashed line indicates the mass threshold for clusters in cut-out that we labelled as 'cluster-containing'.

If we look at the performance of the methods in the Virial Mass versus Redshift space, we can see that the CNN positively identifies more of the clusters at low redshifts whose masses lie just above the mass cut than the MF identifies (i.e. $2 \times 10^{14}\,\mathrm{M}_\odot \lesssim M_{\mathrm{vir}} \lesssim 3 \times 10^{14}\,\mathrm{M}_\odot$ and $z \lesssim 0.7$). On the other hand, the MF positively identifies more of the clusters at higher redshifts whose masses lie just above the mass cut (i.e. $2 \times 10^{14}\,\mathrm{M}_\odot \lesssim M_{\mathrm{vir}} \lesssim 3 \times 10^{14}\,\mathrm{M}_\odot$ and $z \gtrsim 1$). As a result, the EnsemblePROD has more positive detections of clusters above the mass cut across the redshift range.

We can do a similar comparison in the virial mass versus virial radius space. Here, we see that the MF preferentially picks out the highest mass haloes at fixed virial radius, missing some of the clusters near the mass threshold that are more spatially extended. The CNN manages to identify more of these 'missed' clusters, with identified clusters that appear to be more complete with a limiting mass. Again, we see how the complementarity manifests itself in the EnsemblePROD, which identifies clusters missed by either method on its own.

The last row of this figure, showing the per cent of positively identified clusters in the space of virial mass versus angular size, summarizes the effects from the first two rows. Clusters with large angular size are lower redshift clusters with larger virial radii. These are 'missed' clusters that the CNN identifies more than the MF, leading to a better performance for the EnsemblePROD.

5 SUMMARY AND DISCUSSIONS

In this paper, we compare the performance of a MF method and a CNN in the task of identifying galaxy clusters in mock millimeter maps of the CMB. For the neural network, we use a modified version of ResNet, a architecture used in popular image classification. We use simulated microwave maps at 90, 148, and 219 GHz channels with added observational components to train and test our CNN. We also use the F1 score (see Section 4.2), a quantity that accounts for both precision (purity) and recall (completeness), to compare method performance and to define an identification procedure that combines both the MF and CNN. We find the following:

(i) At the selected redshift and mass thresholds, the CNN does comparably to the MF (see Table 1). The precision (purity) of the CNN is slightly lower, but the the recall (completeness) slightly higher.

(ii) The CNN achieved comparable performance in the absence of standard image pre-processing, e.g. normalization, point source subtraction, etc.

(iii) A cluster identification procedure that combines both the MF and CNN scores significantly improves performance (e.g. see Figs 12–14), indicating complementarity between the methods.

(iv) We note that the cut-out nature of the train/test/validation data set for the CNN impacts the model performance; clusters further from the cut-out centre are more difficult for the CNN to identify (see Section 4.4.2). An algorithm that minimizes the distance between the cluster centre and the cut-out centre would further improve the CNN performance.

We note our experiment only includes the approximated effect of the instrument beam. The effect of the beam is to convolve the input with a Gaussian of full width at half-maximum of ∼1 arcmin for SPT-3G-like map. An SPT-3G-like beam would therefore blur features smaller than the 1 arcmin scale. For the MF method, we expect the impact of a more accurate beam approximation on the model performance to be fairly minor. Only the clusters with the smallest angular sizes would be blurred. On these spatial scales, low-flux point sources tend to dominate the map, leading to a downweighting for these smaller clusters. Therefore, we expect a minimal difference in relevant information with our approximation of the beam. For CNN, a more accurate beam approximation is likely to negatively affect the accuracy, but will require additional experiments to verify. However, we emphasize that as long as the method for simulating the beam consistent between the training and testing data, CNN models should be relatively resilient to it (Dodge & Karam 2016; Vasiljevic, Chakrabarti & Shakhnarovich 2016).

We note that some cut-out contributing to false positive identification are cut-out that contain clusters just below the mass threshold of $M_{\rm vir} = 2 \times 10^{14}$ that we label as ‘cluster-containing’. The cluster-finding task differs from most other standard classification applications in that galaxy clusters may be defined on a continuum; the separation between galaxy clusters and galaxy groups is largely definition-based. Admittedly, galaxy clusters would be ideal objects to apply regression methods that predict continuous values of cluster parameters. We leave this to a follow-up paper.

We also emphasize the use of the F1 score as a mechanism for apples-to-apples cluster-finding method comparisons and method combinations. The F1 score plays a distinct role of enabling a performance comparison between the two methods considered. The purity and completeness of a method depends on a threshold for positive identification. Shifts in that threshold for a given method will change the quoted purity and completeness. We therefore choose a threshold for each method that maximizes the F1 score in that method. In other words, our cluster-finding comparison compares the best version of the CNN and MF to one another, using the F1 score as a metric for ‘best’. We additionally use the F1 score as a metric that allows us to combine the two cluster-finding methods to further improve performance. We present a use case of the F1 score as a comparison metric and combination mechanism that can be used as a template for other cluster-finding comparisons and combinations. We acknowledge, however, the additional complications that would arise in characterizing the selection function for a combined method. The most straightforward method would be a Monte Carlo simulation. While this is computationally expensive, the characterization would only need to be done once after the model has been trained.

Finally, we comment on the complementarity of MF and CNN. MF inherently relies on an understanding of the expected signal and noise in the filter definition. The CNN relies on an understanding of the data when generating the training data set. Furthermore, the CCN does *not* rely on any assumptions in the network architecture nor did it require that the data undergo pre-processing steps such as point source removal, input of SZ frequency spectrum or assumed cluster profile (typically a beta model for MF). Another distinguishing aspect between the two is that the MF has an explicit fully analytic formulation as a discriminative model, while the CNN has a quasi-analytic (semiparametric) formulation via the simulation data used to train the model.

A common criticism of machine learning methods, particularly neural networks, is in the lack of interpretability. While the MF method has physically motivated, or a priori physically denoted, features the method is designed to detect, the CNN picks up on a non-linear combination of features that do not necessarily have obvious physical motivation. The non-linearity is likely what enabled the CNN to have the complementary model performance seen in Fig. 14. Here, we see that the CNN has more positive detections in the space of low mass/low redshift (or low mass/larger angular scale). Finally, we note that other architectures using neural networks have demonstrated advantages in other tasks that can be applied to SZ clusters. For example, there has been recent advances in the context of galaxy–galaxy blending Arcelin et al. (2020). This approach could be useful in the context of deblending low mass galaxy clusters.

Last, we note that the MF method had taken human time to calibrate. While use of the CNN did not require significant calibration, the human time went into developing the CNN architecture and training the model.

ACKNOWLEDGEMENTS

Author Contributions

(i) **Lin**: led the design and experiments of the standalone NN and ensemble methods, prepared the cut-out from raw data after appropriate checks, performed the evaluation and diagnostic analysis for MF, NN, and ensemble models, contributed to manuscript writing, and kept the work going through challenging phases.

(ii) **Huang**: optimized and ran the MF, producing a catalogue for comparison, and contributed to manuscript writing.

(iii) **Avestruz**: contributed to the initial project conception, project management, manuscript writing, and student mentorship, including guidance on analysis direction.

(iv) **Wu**: generated simulations for testing and validation; did initial classification; contributed to manuscript writing.

(v) **Trivedi**: prototyped initial architecture and experiments, provided guidance and student mentorship on subsequent deep learning components, including sharing expertise on ML methodology.

(vi) **Caldeira**: contributed to comparisons between results of the two methods and manuscript writing.

(vii) **Nord**: developed the initial project conception, helped assemble the research team, contributed to technical development, project management, manuscript writing, and student mentorship, including guidance on analysis direction.

We would like to thank Arya Farahi for discussions that helped improve this manuscript. We also thank Phil Mansfield for help on manuscript presentation at the end of the project. CA would like to thank support from the LSA Collegiate Fellows program and the Leinweber Center for Theoretical Physics at the University of Michigan. This project was supported in part by NSF-AAG awards AST-200994 and AST-2009121. During part of the project, ST was supported by by the National Science Foundation under Grant No. DMS-1439786 while he was in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the non-linear algebra program.

We acknowledge the contributions of members of the Deep Skies collaboration, https://deepskieslab.com, as a community of multidomain experts and collaborators who have facilitated an environment of open discussion, idea-generation, and collaboration. This community was important for the development of this project. We would also like to extend our thanks to the SPT collaboration for providing the MF implementation.

Work supported by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1752814.

DATA AVAILABILITY

As noted in Section 2.1, the simulation data is from Sehgal et al. (2010), and can be downloaded from https://lambda.gsfc.nasa.gov/toolbox/tb_sim_ov.cfm. The detailed data generation method has been described in Section 2.

REFERENCES

Abazajian K. N. et al., 2016, preprint (arXiv:1610.02743)
Allen S. W., Evrard A. E., Mantz A. B., 2011, ARA&A, 49, 409
Arcelin B., Doux C., Aubourg E., Roucelle C., 2020, MNRAS, 500, 531
Benson B. A. et al., 2014, in Holland W. S., Zmuidzinas J., eds, Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VII. SPIE, Bellingham, p. 91531P
Bleem L. E. et al., 2015, ApJS, 216, 27
Bocquet S. et al., 2019, ApJ, 878, 55
Bonjean V., 2020, A&A, 634, A81
Bulbul E. et al., 2019, ApJ, 871, 50
Caldeira J., Wu W. L. K., Nord B., Avestruz C., Trivedi S., Story K. T., 2019, Astron. Comput., 28, 100307
Carlstrom J. E., Holder G. P., Reese E. D., 2002, ARA&A, 40, 643
Cavaliere A., Fusco-Femiano R., 1976, A&A, 49, 137
Costanzi M. et al., 2019, MNRAS, 488, 4779
Dodge S., Karam L., 2016, in 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX). IEEE, Lisbon, Portugal, p. 1
Green S. B., Ntampaka M., Nagai D., Lovisari L., Dolag K., Eckert D., ZuHone J. A., 2019, ApJ, 884, 33
Gupta N., Reichardt C. L., 2020, preprint (arXiv:2005.13985)
He K., Zhang X., Ren S., Sun J., 2016, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016. IEEE, Las Vegas, NV, p. 770
Hilton M. et al., 2018, ApJS, 235, 20
Hilton M. et al., 2021, ApJS, 253, 3
Ho M., Rau M. M., Ntampaka M., Farahi A., Trac H., Póczos B., 2019, ApJ, 887, 25
Hortúa H. J., Volpi R., Marinelli D., Malagò L., 2020, Phys. Rev. D, 102, 103509
Huang N. et al., 2020, AJ, 159, 110 (H19)
Hurier G., Aghanim N., Douspis M., 2017, preprint (arXiv:1702.00075)
Keisler R. et al., 2011, ApJ, 743, 28
Komatsu E. et al., 2011, ApJS, 192, 18
Krachmalnicoff N., Tomasi M., 2019, A&A, 628, A129
LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, in Proc. IEEE, 86, 2278
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
McDonald M. et al., 2017, ApJ, 843, 28
Mantz A., Allen S. W., Rapetti D., Ebeling H., 2010, MNRAS, 406, 1759
Mantz A. B. et al., 2015, MNRAS, 446, 2205
Mehrtens N. et al., 2012, MNRAS, 423, 1024
Melin J.-B., Bartlett J. G., Delabrouille J., 2006, A&A, 459, 341
Melin J. B. et al., 2012, A&A, 548, A51
Ntampaka M. et al., 2019a, BAAS, 51, 14
Ntampaka M. et al., 2019b, ApJ, 876, 82
Planck Collaboration XXIV, 2016a, A&A, 594, A24
Planck Collaboration XXVII, 2016b, A&A, 594, A27
Platt J. C., 1999, in Smola A. J., Bartlett P. L., Scholkopf B., Schuurmans D., eds, Advances in Large Margin Classifiers. MIT Press, Cambridge, p. 61
Rozo E. et al., 2010, ApJ, 708, 645
Sehgal N., Bode P., Das S., Hernandez-Monteagudo C., Huffenberger K., Lin Y.-T., Ostriker J. P., Trac H., 2010, ApJ, 709, 920
Sehgal N. et al., 2011, ApJ, 732, 44
Shirokoff E. et al., 2011, ApJ, 736, 61
Strazzullo V. et al., 2019, A&A, 622, A117
To C. et al., 2021, Phys. Rev. Lett., 126, 141301
Vanderlinde K. et al., 2010, ApJ, 722, 1180 (V10)
Vasiljevic I., Chakrabarti A., Shakhnarovich G., 2016, preprint (arXiv:1611.05760)
Vikhlinin A. et al., 2009, ApJ, 692, 1060
Zohren H., Schrabback T., van der Burg R. F. J., Arnaud M., Melin J.-B., van den Busch J. L., Hoekstra H., Klein M., 2019, MNRAS, 488, 2523

This paper has been typeset from a TEX/LATEX file prepared by the author.