Draft version September 13, 2024
Typeset using LaTeX **twocolumn** style in AASTeX631

Using Neural Networks to Automate the Identification of Brightest Cluster Galaxies in Large Surveys

Patrick Janulewicz,[1,2,3,4] Tracy Webb,[1,2] and Laurence Perreault-Levasseur[2,3,4,5,6,7]

[1] *Department of Physics, McGill University, Montréal, Canada*
[2] *Trottier Space Institute, McGill University, Montréal, Canada*
[3] *Ciela Institute, Montréal, Canada*
[4] *Mila - Quebec Artificial Intelligence Institute, Montréal, Canada*
[5] *Department of Physics, Université de Montréal, Montréal, Canada*
[6] *Center for Computational Astrophysics, Flatiron Institute, New York, USA*
[7] *Perimeter Institute for Theoretical Physics, Waterloo, Canada*

ABSTRACT

Brightest cluster galaxies (BCGs) lie deep within the largest gravitationally bound structures in existence. Though some cluster finding techniques identify the position of the BCG and use it as the cluster center, other techniques may not automatically include these coordinates. This can make identifying and studying BCGs in such surveys difficult, forcing researchers to either adopt oversimplified algorithms or to perform cumbersome visual identification. For large surveys, there is a need for a fast and reliable way of obtaining BCG coordinates. We propose machine learning to accomplish this task and train a neural network to identify positions of candidate BCGs given no more information than multiband photometric images. We train on both mock observations from The Three Hundred project and real ones from the Sloan Digital Sky Survey (SDSS), and we quantify the performance in each case. While the network trained on real data selects the true BCG with high accuracy, the network trained on simulations struggles to identify some subsets of real galaxy clusters. Nonetheless, we find this technique to be a promising method to automate and accelerate the identification of BCGs in large datasets.

1. INTRODUCTION

Galaxy clusters are the largest gravitationally bound objects in the universe. At the heart of these clusters lies the brightest cluster galaxy (BCG). These BCGs are among the most massive and luminous galaxies in the universe today. They also play a pivotal role in our understanding of galaxy evolution. The BCG is often considered to be the optical center of the galaxy cluster, as it lies at the base of the cluster's gravitational potential well (van den Bosch et al. 2004). Some optical cluster finding techniques naturally identify the most likely BCG candidate and set the center of the cluster to be at that point. A well-known example of this is redMaPPer (Rykoff et al. 2014), which assigns up to 5 candidates ranked by probability. Another example is the method seen in Wen et al. (2012) (henceforth WHL12), which selects the brightest galaxy within a certain distance from the cluster center.

However, not all cluster finding techniques use the BCG as a center. X-ray surveys (Ebeling et al. 1997; Vikhlinin et al. 1998), for example, select the X-ray emission peak instead. Another popular method is to use fluctuations in the cosmic microwave background caused by the Sunyaev-Zeldovich (SZ) effect as tracers of galaxy clusters (Bleem et al. 2015). In this case, the center is selected to be the peak of the hot gas distribution. The BCG, however, has been shown to often have an offset with the center found in X-ray or SZ surveys, particularly as redshift increases (Gozaliasl et al. 2019). Similar results are also suggested by cosmological simulations (Cui et al. 2016). Even in optical surveys, it is not guaranteed that the BCG is automatically identified. For instance, some algorithms select clusters by searching for overdensities of galaxies that form a red sequence (Gladders & Yee 2000). These include the Spitzer Adaptation of the Red-sequence Cluster Survey (Muzzin et al. 2009; Wilson et al. 2009) and the Massive and Distant Clusters of WISE Survey (Gonzalez et al. 2019). Because these techniques focus on overdensities rather than individual galaxies, the coordinates of the BCG centroid are not necessarily provided. In addition to this, the BCG has been predicted to lose its unique identity around $z \sim 0.7$ (De Lucia & Blaizot 2007). Combining all these factors, we find that correctly identifying the BCG is not a trivial task even at low redshift, and that it becomes increasingly difficult at higher redshift.

With recent and upcoming projects such as Euclid (Laureijs et al. 2011) and the Vera C. Rubin Observatory (Ivezić et al. 2019) expected to greatly increase the quantity of available wide-field data, having a quick and accurate method of identifying the BCG from images will become crucial. Such large amounts of data will make manual identification infeasible, while overly simplistic algorithms may fail given the complexity of the task. As a result, there is a need to better automate this process. We investigate a model capable of appropriately weighing features including optical color, morphology, and size, while also being adaptable to different redshifts and surveys. Such requests naturally lend themselves to the use of machine learning.

In this work, we test the abilities of neural networks to identify the BCG centroid from galaxy cluster images. We train a neural network on simulated cluster images from The Three Hundred project (Cui et al. 2018) and test its performance on real cluster images from the Sloan Digital Sky Survey (SDSS) (York et al. 2000; Aihara et al. 2011). We train another neural network on real galaxy cluster images for comparison. Though we explore higher redshifts later in the paper, we introduce this work as a proof of concept for a relatively narrow range of low-redshift observations. By doing so, we hope to minimize evolutionary effects and limitations caused by resolution or depth, thus reducing systematic uncertainties.

We argue that training on simulations has many advantages over training on real data. For one, simulations allow us to be certain about our choice of BCG. In observations, there is always a chance that the galaxy selected as the BCG is incorrectly identified. This is especially useful for generalizing the task to higher redshift, where the true BCG may be difficult to discern using real observations. Secondly, using simulated data gives access to a variety of characteristics not necessarily known to the observer, allowing for the possibility of further inference. While we recognize that training on mock observations may introduce biases when the simulations stray from reality, we believe that their various benefits make them particularly attractive.

The paper is laid out as follows. In Section 2, we describe the dataset, which contains both simulated observations and real ones. In Section 3, we describe the methods used to post-process the data, train the machine learning model, and assess its accuracy. In Section 4, we present the results, which include training and testing the model, studying the differences between simulated and real observations, and performing relevant tests for those wishing to apply these methods to other datasets. In Section 5, we summarize the contents of the paper and discuss the application of this technique to large surveys.

2. DATA

2.1. *The Three Hundred data*

Simulated clusters are obtained from The Three Hundred project cosmological simulation, which consists of 324 galaxy clusters modeled with different full-physics hydrodynamical re-simulations and semi-analytical models. We use clusters obtained from the gizmo-simba run, as this has been shown to produce more realistic BCGs compared to other codes from the same simulation (Cui et al. 2022). We select a redshift range between $z = 0.15$ and $z = 0.25$. The full dataset of simulated clusters contains four snapshots within this range, all of which are included in our dataset.

Simulated cluster images are then generated using the Python package PyMGal[1], which calculates magnitudes using techniques from EzGal (Mancone & Gonzalez 2012) and then projects them to create mock observations. When using the software, we assume a BC03 simple stellar population model (Bruzual & Charlot 2003) with a Chabrier initial mass function (Chabrier 2003), and we select the Gaussian smoothing length of a given particle to be the distance to its 30th nearest neighbor.

For each snapshot, 10 random projection angles are chosen, giving 40 different projections for each of the 324 clusters. This makes for a total of nearly 13000 projections. For each one, we generate images in all five of the SDSS bands. These bands are ultraviolet (u), green, (g), red (r), near-infrared (i), and infrared (z). Each projection is thus made up of a 5-channel 256x256 image. We select the side lengths to represent a physical distance of roughly $1\frac{1}{3}$ Mpc. A sample can be found in the first row in Figure 1.

2.2. *SDSS data*

The data obtained from SDSS can be split into two separate parts: noise images and cluster images.

We begin with the noise. To make the simulated clusters similar in appearance to observed ones, we must add realism to them. To do this, we superimpose randomly selected images of the sky obtained from SDSS. Coordinates are chosen randomly from the area of the BOSS survey (Alam et al. 2017). We then cut out an image with side length equal to approximately $1\frac{1}{3}$ Mpc to match the scale of simulated images. This side length

[1] https://pymgal.readthedocs.io

is calculated by producing a random redshift between $z = 0.15$ and $z = 0.25$. For a given redshift, the length is converted to a physical distance by assuming a cosmology of $H_0 = 67.8$ and $\Omega_{\mathrm{m},0} = 0.307$. The number of unique noise samples collected is equal to the number of simulated observations, meaning that no noise sample is repeated.

Though the mock observations each have a side length of 256 pixels, this is not necessarily true for SDSS data. We must therefore resize SDSS images to match the dimensions of the simulations. We use the CCD rebinning algorithm from the RealSim package (Bottrell et al. 2019). This ensures that the total flux is conserved and maximizes the fidelity of the mock observations. Clean simulated images and SDSS noise are then added together in FITS format as shown in Figure 1. They are then converted to PNG format using logarithmic scaling.

Images of real galaxy clusters are also obtained from SDSS using the WHL12 catalog. The catalog contains the name, celestial coordinates, and photometric redshift of each galaxy cluster. We use this photometric redshift to select all clusters between $z = 0.15$ and $z = 0.25$. We then cut out regions with a side length of $1\frac{1}{3}$ Mpc and select all images that fit within the boundaries of their SDSS field. This leaves us with approximately 7000 real cluster images properly centered on the BCG identified by the catalog. Cluster images are then processed using the same techniques used for noise images. An example of a real galaxy cluster image is shown at the bottom of Figure 1.

The WHL12 catalog also contains data describing the BCG brightness and the richness of the cluster. The brightness is defined as the Petrosian magnitude of the BCG in the r-band, which we will refer to as r_{BCG}. The number of galaxies within R_{200} of the cluster, which we denote N_{200}, is included as a measure of richness. We will refer to these properties in Section 4 and when isolating subsets of BCGs that may be easier or more challenging to identify.

3. METHODS

3.1. *Machine learning details*

Because both the real and simulated clusters are centered on their BCG, we must modify them to ensure that the model is capable of generalizing to instances where the BCG may be offset, as this will be the case for many surveys. However, we want to instill in the model a tendency to prefer central galaxies, as the BCG in real cluster images should still be near the cluster center defined by the given technique. To do this, we offset the images using a Gaussian perturbation from the center. We randomly draw x and y values from a normal distribution and offset the images by the result. The distribution has a mean of 0 kpc and a standard deviation of 150 kpc. This roughly covers the distribution of offsets found in other works such as the offset between the BCG and the X-ray peak (Seppi et al. 2023). To ensure that the smaller image does not exceed the boundaries of the larger one, we truncate the Gaussian at the maximum possible offset. We then cut out a region of 1 Mpc from the larger $1\frac{1}{3}$ Mpc image, meaning that the resulting images have pixel dimensions of 192x192.

The images are then divided into training, validation, and test datasets. For real observations, we use 80% of images for training, 10% for validation, and 10% for testing. For the simulations, we split by cluster number to ensure that no simulated galaxy cluster appears in two datasets. These numbers are drawn at random and assigned to a dataset. In other words, we select 80% of the 324 clusters for training, 10% for validation, and 10% for testing.

We then pass the off-centered 5-channel training images to the neural network. We select a ResNet18 architecture (He et al. 2015) and pass the x and y pixel values of the BCG centroid as learned parameters. Details regarding the model architecture and hyperparameters can be found in Table 1.

Parameter	Value
Main model	
Architecture	ResNet18
Learning rate	0.001
Batch size	16
Optimizer	Adam
Loss function	MSE
Activation function	ReLU
Transfer learning	
Learning rate	0.0001

Table 1. Details and hyperparameters of the model used to determine the BCG position.

We also use real cluster images to train different networks with the same architecture. We compare one model trained only on simulations, one trained only on real cluster images, and one trained on both via transfer learning. For the transfer learning model, we partially train on simulated images and use the result to initialize the network's weights. We then decrease the learning rate by one order of magnitude and fine-tune on real images. This allows the model to gain information on both

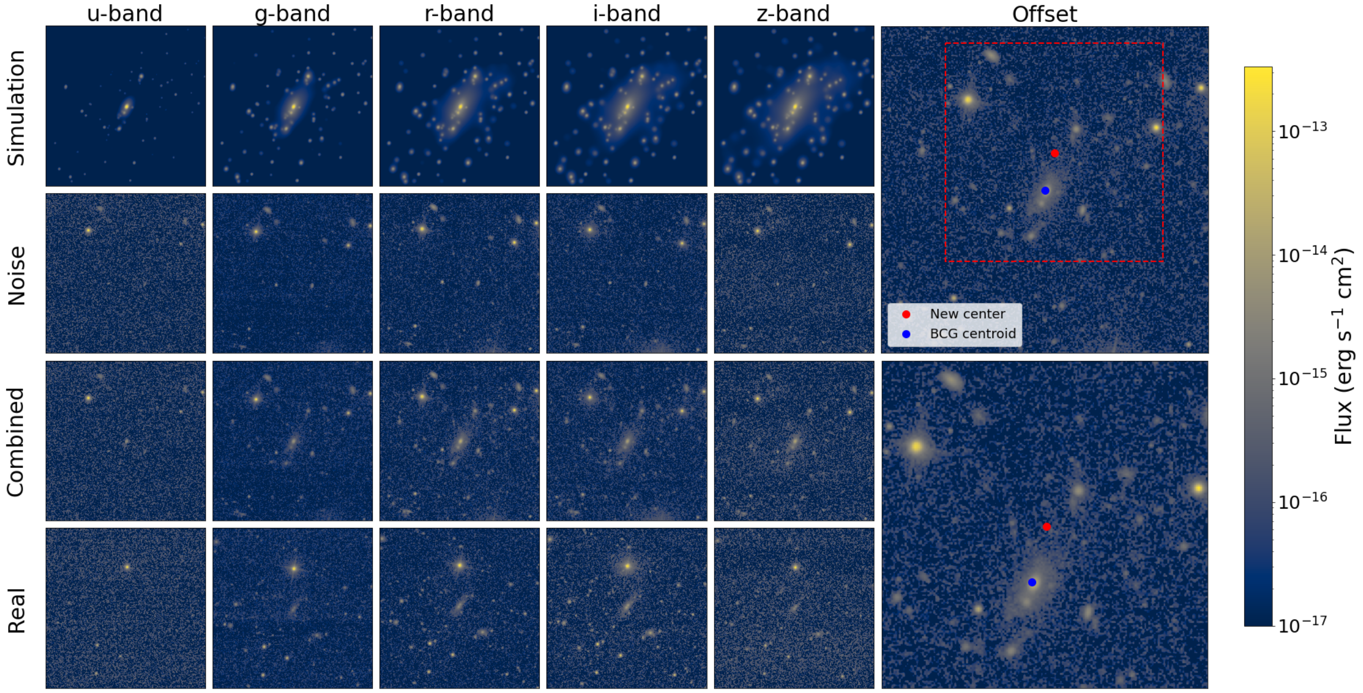


Figure 1. The five columns on the left represent photometric bands from SDSS. Simulated galaxy clusters are shown in the first row along with random SDSS noise in the second row. The combination of the two is shown in the third row, and a real WHL12 cluster is shown in the final row for comparison. In the rightmost column, we give an example of a cluster being off-centered by illustrating the process on the simulated r-band.

simulated and real observations. More details on these tests can be found in Section 4.

3.2. *Assessing accuracy*

We now define the statistics used to quantify the model's accuracy. The first statistic is the coefficient of determination, denoted R^2. To assess the quality of the predictions, we can plot the values predicted by the model as a function of the ground truth. A strong correlation indicates that predictions rarely deviate from the truth, meaning that a perfectly accurate model would produce $\mathrm{R}^2 = 1$. Another way of quantifying success is by measuring the proportion of predictions that are correct within some reasonable error. In this work, we consider a prediction to be correct if it lies within some distance of the true position and incorrect otherwise. Such a threshold should be large enough to encompass the entire BCG, but small enough to avoid false positives. Unless otherwise specified, we will use a threshold of 25 kpc, as this is a conservative estimate for the region of the cluster dominated by the BCG (Brough et al. 2024).

We define one coefficient of determination per predicted parameter, which we write R_x^2 and R_y^2. We also define the proportion of predictions that are correct within the distance threshold and refer to it as the accuracy $\mathrm{A_T}$. A mathematical description of these statistics is shown in the three equations that follow, where we denote predicted values with a hat operator, averages with an overline, and true values without an operator symbol.

$$\mathrm{R}_x^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{1}$$

$$\mathrm{R}_y^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

$$\mathrm{A_T} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\leq\,\mathrm{T}}(d_i) \tag{3}$$

Note that d_i is equal to the Euclidean distance $\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}$ between the true and predicted positions. We use $\mathbf{1}_{\leq\,\mathrm{T}}(d_i)$ to denote the indicator function, which is equal to one if the error is less than the threshold and zero otherwise.

We will refer to these statistics in Section 4 when assessing the model's performance.

4. RESULTS

4.1. *Training the model*

We study the model's behavior during training via a learning curve. The learning curve tracks the loss as a function of training epochs. In machine learning, the loss refers to the penalty for incorrect predictions and can be used to indicate the model's accuracy. Training epochs refer to the number of complete passes through the dataset, during which the network's weights are adjusted using gradient descent. The learning curve shows how quickly and how well the model learns to make correct predictions.

Examining this curve can provide important insights about the model's training process. For instance, learning curves help identify when the model begins to overfit the training set, resulting in difficulty generalizing to unseen data. Preventing overfitting is particularly important for this task, as the simulations contain a limited number of unique training samples.

The learning curve shown in Figure 2 tracks the loss in three different datasets. The training and validation set of simulated galaxy cluster images make up two of these datasets. The third is the validation set of real galaxy clusters. We can therefore look at both the real and simulated validation sets to identify potential overfitting.

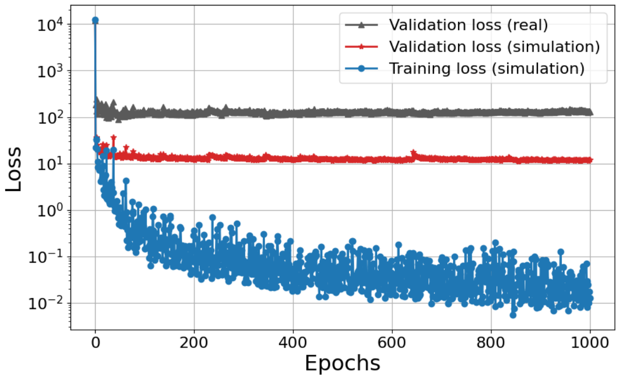


Figure 2. Learning curve when training on simulated galaxy clusters. We compare the training and validation loss for simulated clusters. We also add the validation loss for real clusters. Both validation curves plateau after relatively few epochs, with the validation loss for real cluster images remaining well above that of simulated images.

We draw two notable conclusions from this test. The first conclusion is that performance on both validation sets appears to plateau after relatively few epochs. We find that training beyond this point does not significantly reduce the validation loss. To avoid overfitting, we will use a model trained for 100 epochs for testing unless otherwise specified. The second conclusion concerns the relationship between the curves for the two validation sets. Despite both curves plateauing, the loss for the real clusters stays consistently above the loss for the simulated ones. It appears that additional training is insufficient to properly reduce the loss on real images. This suggests that there may be a discrepancy between the two datasets that cannot be corrected by training the model for longer.

4.2. *Quantifying differences between datasets*

The learning curve suggests that there are key differences between real and simulated images. It is important to understand these differences before proceeding, as any biases between the two may affect the model's performance. We consider two tests to summarize these differences. To perform these tests, we randomly select 5000 images from each dataset. This ensures a large sample size that is equivalent across the two groups.

The first test is to study the one-dimensional probability density function (1D PDF) of the flux values. This PDF can be obtained by averaging the flux over multiple images and binning the pixels to a histogram. Because this is repeated over a large number of experiments, the result should demonstrate the typical brightness distribution of a galaxy cluster image. We obtain a PDF in each band for both simulated and real images, and we show the result in the first row of Figure 3.

The second test is to compute the power spectrum of the images. The power spectrum indicates the typical variability between two pixel values at a given length scale. The power, which we denote $P(k)$, quantifies the amount of variability between samples for a given wavenumber k. This wavenumber is a spatial frequency representing how often a wave pattern repeats per unit distance. In this case, the largest wavenumber represents the smallest distance, which corresponds to one pixel or approximately 5 kpc. The smallest wavenumber represents the largest distance, which corresponds to half the image dimension.

The flux PDFs demonstrate a tendency for the simulated cluster images to be slightly brighter than the real ones. This is repeated across all SDSS bands. This may indicate that the population of simulated clusters is more luminous than the population of real ones.

The power spectra show that the simulated cluster images show slightly higher variability overall. This is particularly true for smaller wavenumbers, meaning that large distances vary more in simulated images than in real ones. This may also suggest that the simulations show brighter centers, as the larger distances compare BCGs with the outskirts of the galaxy cluster. We will take these differences into consideration in Section 4.3 by isolating subsets of clusters with more prominent BCGs.

4.3. *Testing the model*

To assess the quality of the predictions, we study how error is distributed among the model's predictions. We compute the Euclidean distance from the true BCG centroid to the value predicted by the model. We obtain

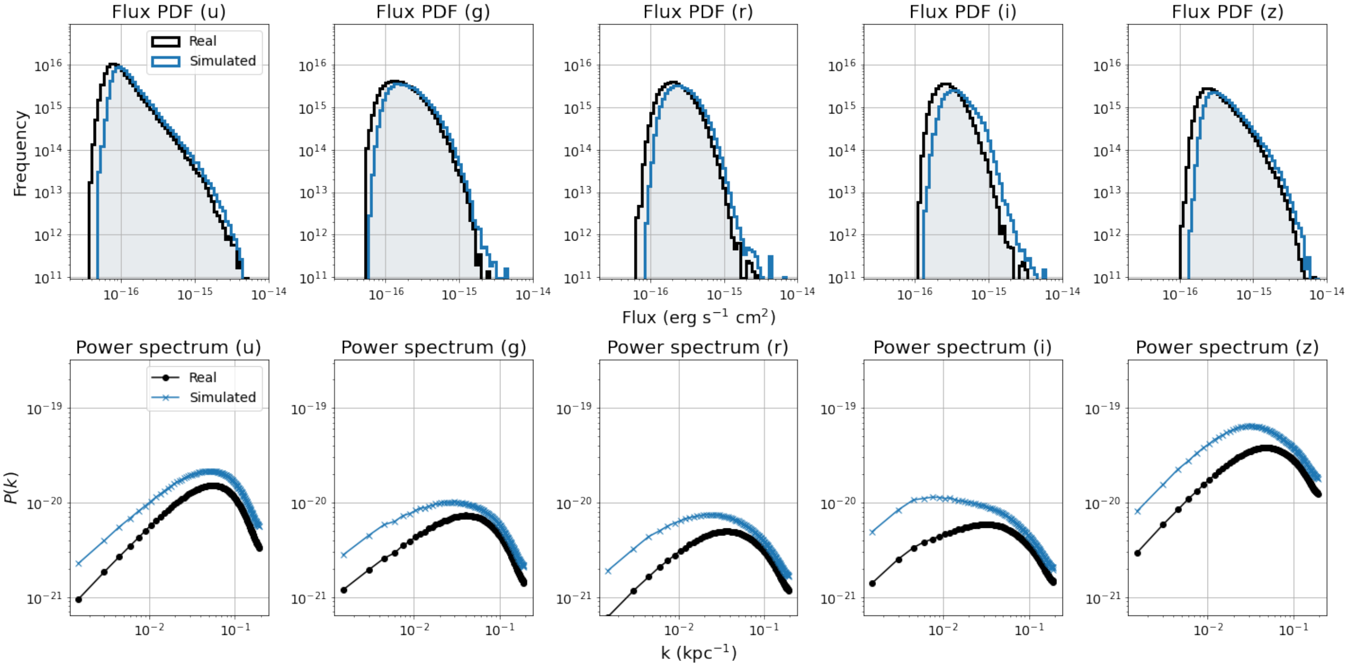


Figure 3. The first row contains the 1D PDF of flux values averaged over 5000 images for each dataset. The second row shows the power spectrum averaged over the same images for each dataset. Columns indicate SDSS bands. While the two datasets appear to be quite similar, the simulated images show higher flux and power.

the predicted positions in pixel space and convert them to physical positions. We can then examine the way this error is spread using a cumulative distribution function (CDF). This function plots the fraction of predictions with an error less than or equal to a given value. A CDF that increases steeply near the origin indicates that most predictions are fairly accurate. On the other hand, a function that increases slowly shows that more predictions are inaccurate. The error CDF can be found at the top of Figure 4.

The CDF shows a discrepancy between the model's predictions across the two datasets. The simulated images tend to have very low error. In fact, we see that most predictions are accurate within roughly 25 kpc. When we look at the real images, this proportion drops by nearly half. This suggests that the model is succeeding in properly identifying some BCGs, but failing in other cases.

We also consider a subset of the brightest BCGs available in the WHL12 catalog, which we henceforth define as having $r_{\mathrm{BCG}} \leq 16.5$. By isolating these easier cases, we can determine whether the model is failing across all real cluster images or simply the more challenging cases. Most predictions on this subset are far more accurate than the predictions made on the entire dataset of real images, with results comparable to predictions on simulations. We find the error on the bright subset to be reasonably small, hinting that the model is capable of bridging the gap between the two datasets.

We also test performance by plotting the values predicted by the model as a function of the ground truth. The resulting correlation strength will then serve as a measurement of success. We use R_x^2 and R_y^2 as defined in Section 3.2 to quantify this strength.

Plots of the true and predicted results can be found at the bottom of Figure 4. The model trained on simulated images performs well when tested on simulated images, though we find a far weaker correlation on real ones. While many of the predictions fall near the true values, we see far more outliers. We test on the subset of real clusters with $r_{\mathrm{BCG}} \leq 16.5$ and find the correlation strength to be comparable to tests on simulations.

In both cases, a trend can be seen when training on simulations and testing on real data. Predictions are highly accurate for bright clusters, but they become less reliable for dimmer ones. Though much of this error is due to the neural network, it is worth noting that dim BCGs should be more challenging to identify for the WHL12 algorithm as well. As a result, looking at subtler cases can make the ground truth less reliable, therefore adding an additional source of uncertainty.

We now analyze the model's ability to adapt to different configurations of training and test datasets. One configuration is to train and test on real galaxy cluster images. Another is to invert the problem by training

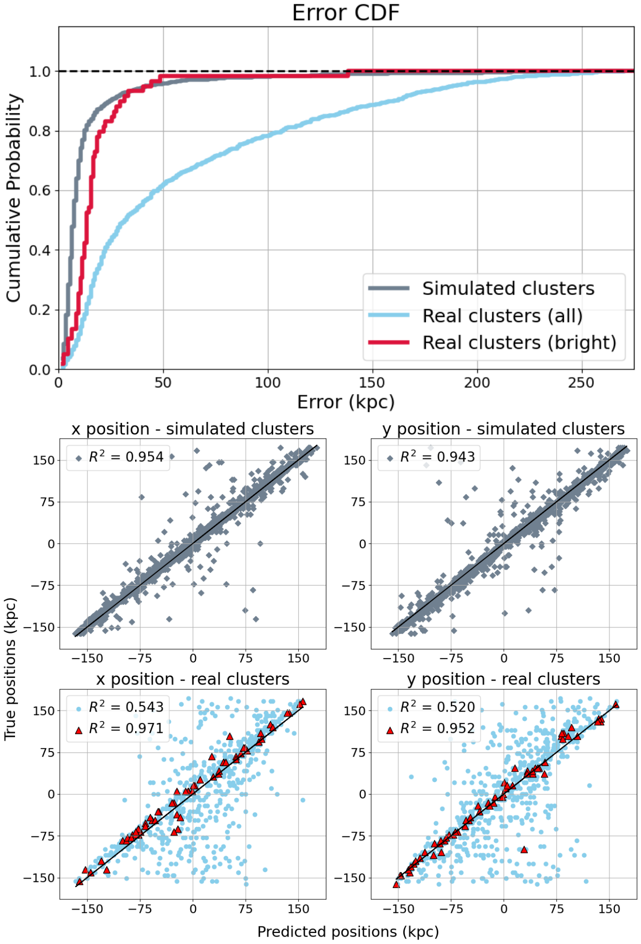


Figure 4. Performance when trained on simulated clusters. The model is tested on simulated clusters, real SDSS clusters, and a bright subset of these SDSS clusters satisfying $r_{\rm BCG} \leq 16.5$. Displayed on top is a cumulative distribution function (CDF) of the Euclidean error across the three test sets. The four figures on the bottom show the true BCG positions plotted against the model's predictions, with a black line displaying $y = x$ shown for reference. The left and right columns show the results for x and y, respectively. Performance is strong when tested on simulations and the bright subset, but it drops when tested on the full set of real SDSS clusters.

on real images and testing on simulations. If the model trained on real data performs badly when testing on simulations, and the model trained on simulations performs badly when testing on real data, then the issue is likely due to bias between the two datasets. However, if the former model performs well, and the latter model performs poorly, this may indicate that the simulated clusters form a subset of the real ones. Finally, we are also interested in whether we can improve performance using transfer learning as described in Section 3.1.

We compare three models each trained on a different dataset. We keep the architecture constant across all experiments and train each network using the same number of epochs and training images. We train one network on 5000 simulated cluster images for 100 epochs and a second network on 5000 real cluster images for 100 epochs. For the transfer learning model, we begin training with 2500 simulated images for 50 epochs and fine-tune with 2500 real images for 50 epochs. The learning rate for fine-tuning is reduced by an order of magnitude as indicated in Table 1. We show the results in Figure 5 by comparing the three statistics defined in Section 3.2.

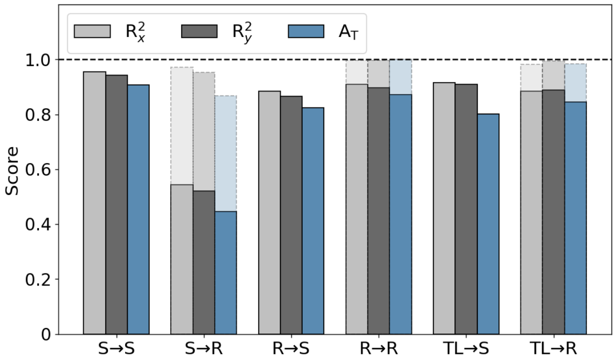
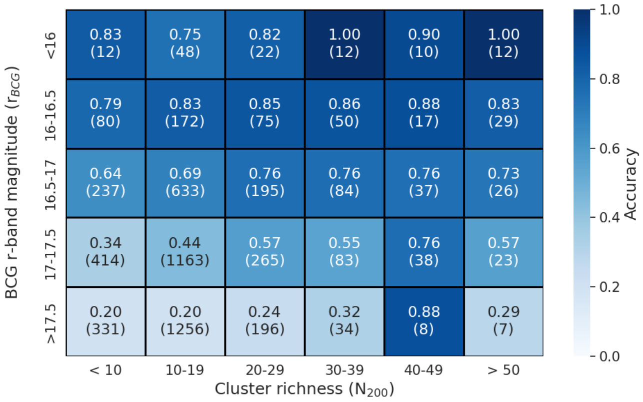


Figure 5. A comparison of R^2_x, R^2_y, and A_T for different networks tested on the real and simulated cluster datasets. The threshold distance for A_T is set to be 25 kpc. Labels to the left of the arrow represent the dataset used for training. They are abbreviated S for simulated images, R for real images, and TL for transfer learning. Labels to the right of the arrow represent the test dataset and are abbreviated similarly. Performance on the bright subset of BCGs ($r_{\rm BCG} \leq 16.5$) is indicated by shaded regions for comparison. Results are strong in all cases except for the model trained on simulations and tested on real images, although they improve drastically when tested on the bright subset.

The model trained on real clusters works well when tested on real clusters, and the model trained on simulations performs well when tested on simulations. Transfer learning shows strong performance across both datasets. When testing on a different dataset than the training set, we get mixed results. We continue to see a drop in performance when training on simulations and testing on real data. However, the model trained on real images appears to perform well when tested on simulations, hinting that the simulations form a subset of the real images. We once again isolate the subset of real observations satisfying $r_{\rm BCG} \leq 16.5$. All networks, regardless of training dataset, make high-accuracy predictions on this subset.

We can further divide the SDSS clusters into groups based on their physical properties to study where the simulation-trained model breaks down. We divide according to $r_{\rm BCG}$ and N_{200}. Brighter BCGs will tend to be more conspicuous, suggesting that they should be easier to identify. Meanwhile, richness tends to be associated with the depth of the potential well of a galaxy cluster, meaning that BCGs in rich clusters tend to be more prominent (Hansen et al. 2009). We test the fraction of predictions accurate within 25 kpc for each subset and show the results in Figure 6.



BCG r-band magnitude (r_{BCG})	< 10	10-19	20-29	30-39	40-49	> 50
<16	0.83 (12)	0.75 (48)	0.82 (22)	1.00 (12)	0.90 (10)	1.00 (12)
16-16.5	0.79 (80)	0.83 (172)	0.85 (75)	0.86 (50)	0.88 (17)	0.83 (29)
16.5-17	0.64 (237)	0.69 (633)	0.76 (195)	0.76 (84)	0.76 (37)	0.73 (26)
17-17.5	0.34 (414)	0.44 (1163)	0.57 (265)	0.55 (83)	0.76 (38)	0.57 (23)
>17.5	0.20 (331)	0.20 (1256)	0.24 (196)	0.32 (34)	0.88 (8)	0.29 (7)

Figure 6. Performance for different values of $r_{\rm BCG}$ and N_{200}. Accuracy is defined as the proportion of predictions within 25 kpc of the true position. The number of samples in each bin is indicated in parentheses. An increasing trend in accuracy can be seen with greater brightness and richness.

While predictions on bright and rich clusters are strong, accuracy drops steeply when testing on clusters with lesser $r_{\rm BCG}$ and N_{200}. This shows that the poorer performance in Figure 4 and Figure 5 is caused by a subset of subtle cases. These results raise a few interesting points. The first is that it is possible to identify subsets of the data where accuracy will be strong. When testing on these subsets, we can have confidence that the model will be reliable. When venturing beyond this subset, we must be more cautious. The second conclusion is that the simulations do not appear to encompass all possible galaxy clusters. Rather, there appears to a lack of dim BCGs in the simulation data. This is not entirely unexpected, as we only have 324 unique galaxy clusters. Nonetheless, this limits our ability to generalize well to the wide range of SDSS observations.

4.4. *Applying to upcoming surveys*

We highlight some important details that may be helpful for generalizing these tests to upcoming surveys.

The question of transfer learning is a crucial one. Ideally, we would like the model to perform well without having ever seen real data. This would make it possible to pre-train a model for upcoming surveys and apply it as soon as there is a new data release. If this is not possible, we would at least hope to be able to fine-tune the model with small amounts of data.

We test the model's accuracy with different numbers of training images. We begin with a small set of 100 randomly selected samples. We then continue to double this number until it reaches a size greater than half of the training dataset. The fine-tuned transfer learning model can then be compared with a model trained from scratch. Results are shown in Figure 7.

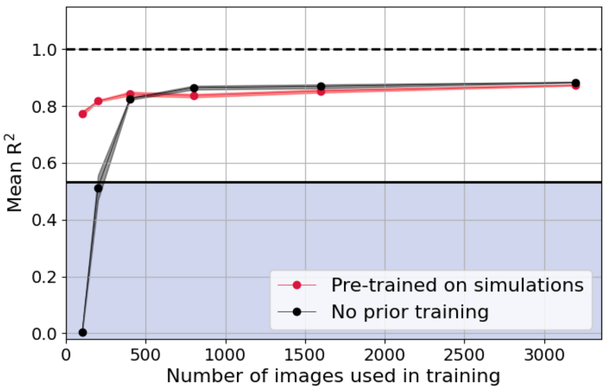


Figure 7. The average of R_x^2 and R_y^2 as a function of training set size, with the thickness of the line being equal to the spread between the two. We compare a model trained from scratch on real images to a model pre-trained on simulated images and then fine-tuned on real ones. The solid black line indicates the average R^2 value for the pre-trained model before transfer learning is applied, with worse values falling in the shaded region below it. Though both models perform well with only a few hundred images, training from scratch requires a larger set of training samples.

This fine-tuned model achieves a strong performance with very few training samples, suggesting that biases in the simulations can be corrected with small amounts of data. In the event that the simulations for large surveys are biased compared to real observations, researchers may be able to correct for this with limited early release data. It is also noteworthy that the two models begin to have similar performance after only a few hundred training images. While simulations have better ground truth and more available information, real data may potentially be less biased. We find that either method can be viable once a modest amount of data is obtained.

We also test the model's ability to adapt to higher redshift and identify the point at which it breaks down. To ensure that we have a representative sample, we collect 5000 test images up to the maximum available redshift of $z = 0.8$. We process them with the same methods used on the other datasets. We pass them to two different models. To avoid bias from the simulations, we only use models trained on SDSS observations. The first model is the network trained on 5000 real galaxy clusters between $z = 0.15$ and $z = 0.25$ described in Figure 5. The second one is trained on the same number of cluster images for the same number of epochs, but covering the entire redshift range. Results can be seen in Figure 8.

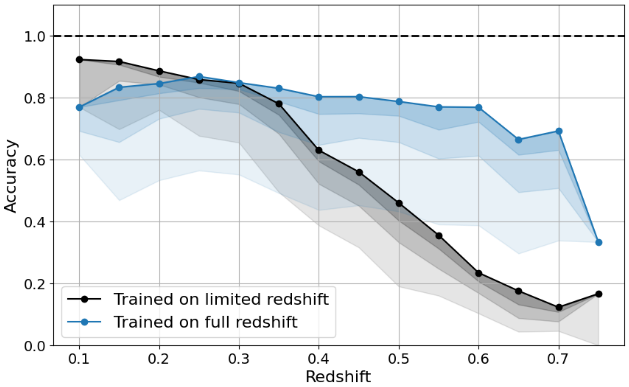


Figure 8. The proportion of predictions within a given distance to the ground truth at various redshifts. The model trained on a range of $0.15 \leq z \leq 0.25$ is compared to a model trained over the entire redshift range of the catalog. Both are tested on real SDSS clusters. Filled lines represent errors below 25 kpc. Shaded regions below the filled lines indicate a tighter error threshold. Ordered from darkest to lightest, the breaks between each shaded region indicate a threshold of 20 kpc, 15 kpc, and 10 kpc. Both models tend to perform well within the training range, though the task becomes increasingly difficult at higher redshift.

The neural network trained on the limited redshift range shows a gradual decrease as we move toward higher redshift. Results begin to deteriorate immediately as we push beyond the training values, and they decline steeply around $z = 0.35$. Even within the training range, there appears to be slightly better performance for closer objects. By the tail end of the redshifts available in the catalog, accuracy has decreased to nearly zero.

We compare this to the network trained on the entire range. This model shows much better adaptability, as performance does not sharply decrease until roughly $z = 0.6$. It does not perform as well on low-redshift clusters as the first, suggesting that there is some benefit to a more specialized model. This also shows that the task can be performed at somewhat higher redshift without significant decreases in accuracy.

Neither model is capable of accurately predicting the BCG position at the tail end of the range. While it is true that these BCGs may be more subtle than those nearby, this is not necessarily the reason for the drop in accuracy. Limitations in depth and resolution from SDSS likely play a role as well. To properly study this phenomenon, observations from more modern telescopes may be required.

5. DISCUSSION

Neural networks show promising results when tasked with identifying brightest cluster galaxies in multiband photometric images. Models trained on simulated images can identify the simulated BCGs with near-perfect accuracy. Perhaps more impressive is that training and testing on real galaxy cluster images shows similar success. While the simulations are limited to 324 unique galaxy clusters and their BCGs, each telescope observation obtained is truly unique. We therefore expect a greater challenge, as there is greater variability in the clusters and the galaxies that form them. Real data also introduces the possibility of false detections or other uncertainties, which could further hurt performance. Despite these challenges, we obtain accurate predictions even when training on relatively few samples.

The biggest challenge arises when attempting to bridge the gap between datasets. After analyzing the images in each photometric band, we find slight systematic differences between the real and simulated observations. This makes it difficult for the model to generalize between the two datasets even when precautions are taken to avoid overfitting. Nonetheless, we find some degree of success when training and testing on different datasets. The model trained on real observations produces strong results when tested on simulations. The converse, however, is not true, suggesting that the simulations cover only a subset of reality.

We are able to isolate a subset of more obvious cases where the network trained on simulations shows extremely strong results. A simple way of doing this is by taking only clusters with particularly bright BCGs. We also investigate the impact of cluster richness on accuracy. These tests show that it is possible for a neural network trained on simulations to identify BCGs in real observations, suggesting that this task can also be done in anticipation of upcoming surveys. Though preparing for these surveys by training on simulations is not trivial, it can be done with great success given the correct dataset. Those wishing to use such an algorithm should take these findings into consideration and thoroughly test the model's adaptability to unseen data. One possible strategy to address this challenge is to assign a confidence level or an uncertainty to a given prediction. Software such as Python's simulation-based inference (SBI) package (Tejero-Cantero et al. 2020) exists for these purposes. In surveys where the ground truth is not necessarily known, estimating uncertainties may help isolate simple cases from more challenging ones.

Even in the event of slight differences between datasets, transfer learning can be employed to fine-tune the network and drastically improve performance. Biases can easily be corrected with relatively few samples.

We also see a level of adaptability to higher redshift. While results are best when testing on the same redshift range used in training, the boundaries can be pushed to some degree without major drops in accuracy. Training over a wider range is also a viable option, but there is a limit at which performance drops steeply. Given the depth and resolution of SDSS, it is unclear whether this limit is the result of technological limitations or the evolution of the galaxy clusters. To settle this, a similar experiment would need to be performed on a deeper, higher-resolution survey.

We reiterate that this method shows strong results, and that neural networks are highly effective for identifying brightest cluster galaxies in large surveys. The primary challenge facing this method is adapting between simulated and real observations. We encourage individuals performing this task to exercise caution when training on simulations, as the quality and diversity of these simulations have a significant impact on the model's adaptability to real observations. While bridging the gap is indeed possible, it is challenging due to the limited amount of unique data from cosmological simulations. Those applying this task to upcoming surveys should consider these facts before moving forward.

ACKNOWLEDGMENTS

We acknowledge support from the Centre de recherche en astrophysique du Québec, un regroupement stratégique du FRQNT. TW and PJ acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2020-04606]. LPL acknowledges support from the Canada Research Chair Program. We also acknowledge access to the theoretically modeled galaxy cluster data via The Three Hundred collaboration[2]. The simulations used in this paper have been performed in the MareNostrum Supercomputer at the Barcelona Supercomputing Center, thanks to CPU time granted by the Red Española de Supercomputación. As part of The Three Hundred project, this work has received financial support from the European Union's Horizon 2020 Research and Innovation program under the Marie Skłodowska-Curie grant agreement number 734374, the LACEGAL project.

REFERENCES

Aihara, H., Allende Prieto, C., An, D., et al. 2011, ApJS, 193, 29, doi: 10.1088/0067-0049/193/2/29

Alam, S., Ata, M., Bailey, S., et al. 2017, MNRAS, 470, 2617, doi: 10.1093/mnras/stx721

Bleem, L. E., Stalder, B., de Haan, T., et al. 2015, The Astrophysical Journal Supplement Series, 216, 27, doi: 10.1088/0067-0049/216/2/27

Bottrell, C., Hani, M. H., Teimoorinia, H., et al. 2019, MNRAS, 490, 5390, doi: 10.1093/mnras/stz2934

Brough, S., Ahad, S. L., Bahé, Y. M., et al. 2024, MNRAS, 528, 771, doi: 10.1093/mnras/stad3810

Bruzual, G., & Charlot, S. 2003, MNRAS, 344, 1000, doi: 10.1046/j.1365-8711.2003.06897.x

Chabrier, G. 2003, PASP, 115, 763, doi: 10.1086/376392

Cui, W., Power, C., Biffi, V., et al. 2016, Monthly Notices of the Royal Astronomical Society, 456, 2566, doi: 10.1093/mnras/stv2839

Cui, W., Knebe, A., Yepes, G., et al. 2018, MNRAS, 480, 2898, doi: 10.1093/mnras/sty2111

Cui, W., Dave, R., Knebe, A., et al. 2022, MNRAS, 514, 977, doi: 10.1093/mnras/stac1402

De Lucia, G., & Blaizot, J. 2007, Monthly Notices of the Royal Astronomical Society, 375, 2–14, doi: 10.1111/j.1365-2966.2006.11287.x

Ebeling, H., Edge, A. C., Fabian, A. C., et al. 1997, The Astrophysical Journal, 479, L101, doi: 10.1086/310589

Gladders, M. D., & Yee, H. K. C. 2000, AJ, 120, 2148, doi: 10.1086/301557

Gonzalez, A. H., Gettings, D. P., Brodwin, M., et al. 2019, The Astrophysical Journal Supplement Series, 240, 33, doi: 10.3847/1538-4365/aafad2

Gozaliasl, G., Finoguenov, A., Tanaka, M., et al. 2019, MNRAS, 483, 3545, doi: 10.1093/mnras/sty3203

Hansen, S. M., Sheldon, E. S., Wechsler, R. H., & Koester, B. P. 2009, The Astrophysical Journal, 699, 1333–1353, doi: 10.1088/0004-637x/699/2/1333

He, K., Zhang, X., Ren, S., & Sun, J. 2015, Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385

Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111, doi: 10.3847/1538-4357/ab042c

[2] https://www.the300-project.org

Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv e-prints, arXiv:1110.3193, doi: 10.48550/arXiv.1110.3193

Mancone, C. L., & Gonzalez, A. H. 2012, Publications of the Astronomical Society of the Pacific, 124, 606–615, doi: 10.1086/666502

Muzzin, A., Wilson, G., Yee, H. K. C., et al. 2009, The Astrophysical Journal, 698, 1934, doi: 10.1088/0004-637X/698/2/1934

Rykoff, E. S., Rozo, E., Busha, M. T., et al. 2014, The Astrophysical Journal, 785, 104, doi: 10.1088/0004-637x/785/2/104

Seppi, R., Comparat, J., Nandra, K., et al. 2023, Astronomy & Astrophysics, 671, A57, doi: 10.1051/0004-6361/202245138

Tejero-Cantero, A., Boelts, J., Deistler, M., et al. 2020, Journal of Open Source Software, 5, 2505, doi: 10.21105/joss.02505

van den Bosch, F. C., Norberg, P., Mo, H. J., & Yang, X. 2004, Monthly Notices of the Royal Astronomical Society, 352, 1302–1314, doi: 10.1111/j.1365-2966.2004.08021.x

Vikhlinin, A., McNamara, B. R., Forman, W., et al. 1998, ApJ, 502, 558, doi: 10.1086/305951

Wen, Z. L., Han, J. L., & Liu, F. S. 2012, The Astrophysical Journal Supplement Series, 199, 34, doi: 10.1088/0067-0049/199/2/34

Wilson, G., Muzzin, A., Yee, H. K. C., et al. 2009, ApJ, 698, 1943, doi: 10.1088/0004-637X/698/2/1943

York, D. G., Adelman, J., Anderson, John E., J., et al. 2000, AJ, 120, 1579, doi: 10.1086/301513