STAT 626 (PRACTICAL WORK) REPORT: UNDERSTANDING THE EFFECTS OF ENERGY BAND SPLITTING IN X-RAY CLUSTER MASS ESTIMATION

MAX LOVIG

Abstract. An important statistical question in astronomy involves trying to predict the mass of a Galaxy Cluster as it provides information on fundamental cosmological parameters. Previous attempts have shown that compressing X-ray observations into an image and using a Convolutional Neural Network (CNN) is a viable way to predict the mass of a cluster. In this work, we display that accurate prediction is possible and demonstrate a phenomenon that prediction dramatically increases in accuracy upon splitting the observed energy into at least two bins. We discuss the extent to which these phenomena occur and show that it is unrelated to a specific feature of our image, Average Galactic Nucleus (AGN). We present a new machine learning visualization to certify that not only the predictions with and without AGN are similar but the training dynamics when splitting energy bands is the same as well. Ultimately, we answer the following question: How does varying the bins of the X-ray energy spectrum dictate how a model can learn to predict the mass of galactic medium?

1. Introduction

Galaxy clusters are large gravitationally bound celestial bodies on the mass[1] $\gtrsim 10^{14} M_\odot$. Understanding the mass of these clusters is important as the mass is correlated with fundamental parameters that dictate how galaxies evolve in space. Making the ability to observe galactic clusters in space and predict their mass with relatively high accuracy is a vital problem in understanding the universe.

Many classical approaches to this problem have been used previously, two examples are cluster luminosity and global temperature. Cluster luminosity is a single dimensional statistics that counts the number of observations that exceed a certain threshold. A common choice may be to count the number of x-ray signals that hit a receiver which fall into a specified energy band (say[2] $[0, 3]keV$). This statistic is then fitted into a (possibly non-linear) regression model that then fits the cluster luminosity to the mass of a cluster. A further option is to restrict this counting statistics to specific areas of the cluster in order to reduce variance in estimation, say to ignore observations near the center of the galaxy due to their high levels of noise. Global temperature is another commonly used statistic as it has a direct theoretical scaling relationship with the mass of the cluster.

Unfortunately, many of these finite dimensional statistics neglect valuable information that may be helpful in predicting the mass of the cluster. There is also ambiguity in the choice of statistic and it would be preferable to have a purely data driven embedding of a galactic cluster which we can then fit a linear regression model to. Motivated by this data-driven embedding, many previous works have attempted to use a (convolutional) neural network to learn a data driven embedding for cluster mass estimation. The final layer of this network is then essentially a finite dimensional statistic that has been trained over many observations. Machine learning has had recent success in solving many astronomical problems [2, 3, 6] and in particular [4] applied a CNN on X-ray post stamps of galactic clusters in an attempt to estimate their mass. The goal of this work is to extend their findings in three ways:

(1) We will consider a more complex set of images than those considered in [4].

(2) We will consider many transformations of the classical single channel image from [4] and attempt to understand how these transformations will help increase the ability to predict the cluster's mass.

(3) We identify that splitting the energy values into distinct bands and then reporting the count shows a significant improvement in prediction. We develop new visualizations to address why this may be happening and for what types of galaxies does the model improve on.

Date: May 2024.

[1]Here $M_\odot$ is the mass of the Sun

[2]keV stands for Kilo Electron Volts

2. Problem and Data Description

Naively, it is impossible to train a model to predict the mass of a galactic cluster in a supervised way, since the true mass of the galaxy is unknown due to our lack of full understanding of the universe. In order to circumvent this problem, the astronomy community has developed high quality simulations of how galaxies evolve in space. In these simulations we can set the specific cosmological parameters for the simulation and then can generate galaxies, the masses of which can be easily recorded. Thus, we can now train our model on these simulated galaxies with known masses. In turn, the hope is that these models will generalize well to real world observations as the simulations included measurement error on the receiver, dithering, and a wider variety of clusters than was used in [4].

Stated formally, we have the following estimation problem. Given a probability distribution P on

$$\underbrace{\mathbb{R}_+}_{\text{Mass}} \times \underbrace{\mathbb{N}}_{\substack{\text{Number} \\ \text{of Observations}}} \times \underbrace{\mathbb{R}^2}_{\substack{(X,Y) \text{ location} \\ \text{on reciever}}} \times \underbrace{\mathbb{R}_+}_{\substack{\text{Enegy of} \\ \text{each X-ray} \\ \text{observation}}},$$

we generate a sequence of X-ray observations of a single galactic cluster. Our goal is too then to try to find the conditional expectation of the mass given the remaining observations. These observations were created using the SIXTE [1] to simulate the types of data received by the Chandra telescope.

Our dataset consists of 3286 sets of X-ray observations. Each set varies in size from 736 to 6558594 observations. Figure 1 is a histogram of the number of observations, binned to remove any cluster with more than 100000 observations. Previous work on this problem was done by Zehao Dou which employed a thinning technique to reduce the number of observations to make this dataset easier to deal with. Since we are processing this data into an image we can avoid this issue as, after transformation, our data is completely converted into a fixed dimension tensor.

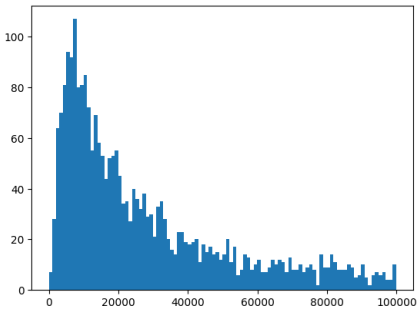

Figure 1. A histogram representing the number of observations per each galaxy. Note that this is only for sets which have less than 100000 observations.

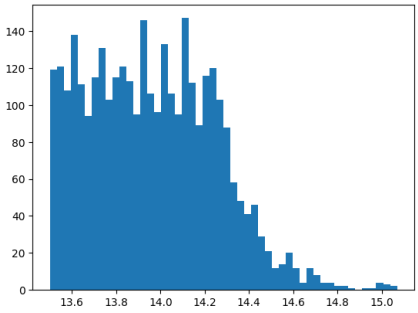

Figure 2. A histogram of the masses of galactic clusters in the dataset, notice that a majority of clusters fall within 13.6-14.3 $\log_{10} M_\odot$ and then we have a tail off with larger clusters.

It is also important to understand the range of cluster masses we will attempt to fit a model to, Figure 2 is a histogram corresponding to the galactic mass (in $\log_{10} M_\odot$ units):

3. Methodology and Results

3.1. **Data Processing.** Once we have our observational data, we will convert each observational set to a $(128 \times 128 \times k)$ image, with $k \in \{1, 2, 3, 4\}$. We have four special process which we hope to test against each other to see which is an optimal transformation for predicting the mass of the cluster. Before we go into the specific methods, we detail how we in general convert our observational data into generic k channel images.

Consider a single observational vector with n entries in $\left(\mathbb{R}_+ \times \mathbb{R}^2\right)^n$. Where the 3 indices are (energy of an X-ray, its x-position, its y-position).

(1) First, we find the maximal and minimal x and y values. This is used to find a box which we will then divide into pixels.

(2) Given some resolution, r, we then divide this box into r^2 pixels. We then create a $r \times r$ array A to fill with our observations.

(3) For each pixel we then search through the observation vector and collect those observations with (x,y) coordinates which belong in

the bounds for said pixel. For each such observation, we append its energy value to the corresponding location in the array A.

(4) Once this is done for all pixels, we now create a single channel as follows:

Given a function, $f : \mathbb{R}^s \to \mathbb{R}$ for any $s \in \mathbb{N}$, we apply this function of each list in the array A.

(5) We then replicate this operation, with a different function, for our number of desired channels k.

Using this general formula, we need just specify r and our collection of functions $\{f_i\}_{i\in[k]}$ for each of our k channels. Below we detail each of these choices for our given models and we give corresponding images for some clusters in our dataset, note that for all of our analysis we have chosen $r = 128$.

3.1.1. *Single Full Band.* This model in the most comparable to [4], where their images were composed of the number of observations in a given pixel within the $[0, 7.5]keV$ energy band, in base 10 logarithmic units. In our notation, [4] considered $k = 1$ and

$$f_1(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i\in[0,7.5]}\right)$$

Where $\mathbb{1}_E$ is the indicator for event E occurring. In order to create a better bench-mark with our multi-channels models, we will use a similar function without the thresholding[3]. So for this model we will still have $k = 1$ but instead will use the function

$$f_1(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}(1 + s)$$

We call this model the Single Band-Model with the 0+ channel. In Figure 3 we have shown 6 examples of these single channels images.

3.1.2. *Multi-Band Energy Bins.* Based on the success of [4] we will consider a similar construction but will now consider splitting the energy bands into four bins and then see if this differentiation of X-ray energy levels aids in the ability to predict the cluster mass more accurately.

[3]We had found that there is little difference in performance in the thresholded versus non-thresholded images

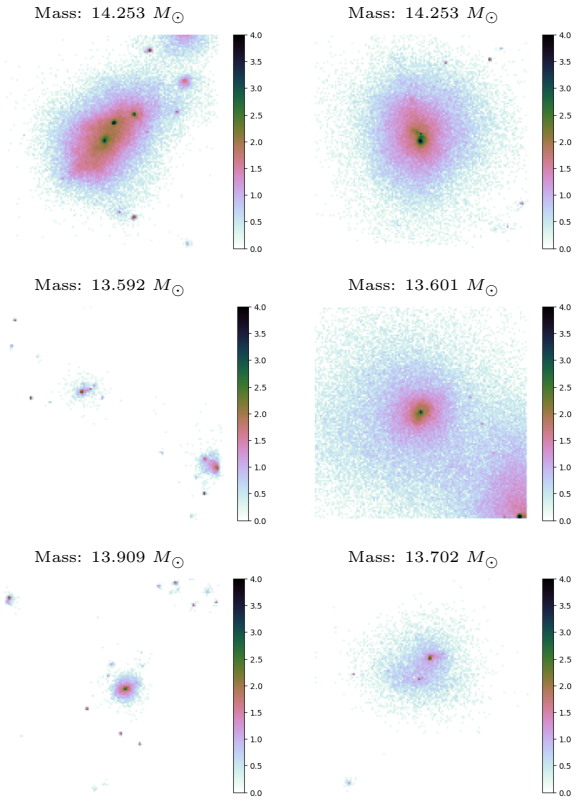

FIGURE 3. An assortment of images under the single band transformation with no thresholding.

To define these energy bin channels, we have the following four functions:

(1)
$$f_1(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i\in[0,2.5]}\right)$$

(2)
$$f_2(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i\in(2.5,5]}\right)$$

(3)
$$f_3(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i\in(5,7.5]}\right)$$

(4)
$$f_4(e_1, \cdots, e_s) = \mathbb{1}_{s>0} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i\in(7.5,\infty)}\right)$$

We then define the following 4 Multi-channel transformations as follows

(1) Multi-band (0-2.5): $k = 1$, $\{f_1\}$

(2) Multi-band (0-2.5, 2.5-5): $k = 2$, $\{f_1, f_2\}$

(3) Multi-band (0-2.5, 2.5-5, 5-7.5): $k = 3$, $\{f_1, f_2, f_3\}$

(4) Multi-band (0-2.5, 2.5-5, 5-7.5, 7.5+): $k = 4$, $\{f_1, f_2, f_3, f_4\}$

We can also see how the differing energy bands will look in our examples in Figure 4.

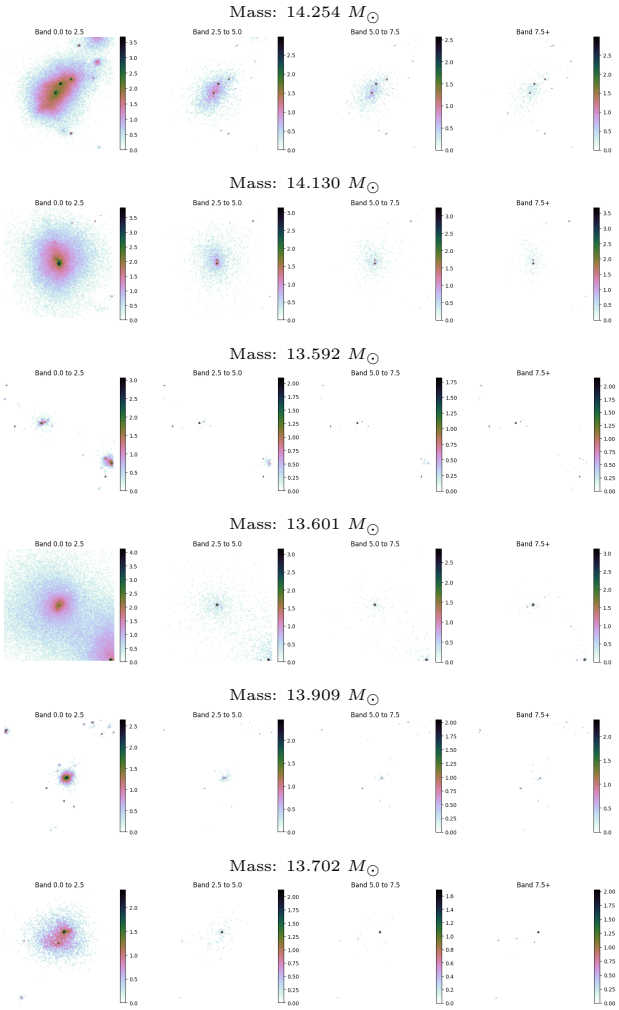

FIGURE 4. Higher order energy band images of the galaxies seen in Figure 3, from left to right we have that bins (0-2.5, 2.5-5, 5-7.5, 7.5+)

3.1.3. *Large Observation Excision.* A common belief on the use of CNNs to predict the mass of galaxies is that the model will predominately use information from the edges of the galaxy while ignoring the content contained in the core of the image. For example a model should, in theory, ignore the AGN (Active galactic nucleus) spikes in the center of the cluster. We can recognize the AGN as the bright dots in Figure 3 & 4. The reason is that these spikes represent a different source of X-ray observations that are unrelated to the overall mass of the cluster. To test this theory, we will remove the inner core of the image by only retaining the bottom $p\%$ of observations, this naturally will only keep the edge of the galactic cluster. Thus, we define $p\%$ retained model with $k = 1$ and

$$f_1(e_1, \cdots, e_s) = \mathbb{1}_{\substack{s>0,\\ s \text{ in the below}\\ \text{the } p\%\text{percentile}}} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i \in [0,2.5]}\right)$$

$$f_2(e_1, \cdots, e_s) = \mathbb{1}_{\substack{s>0,\\ s \text{ in the below}\\ \text{the } p\%\text{percentile}}} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i \in [2.5,5)}\right)$$

$$f_3(e_1, \cdots, e_s) = \mathbb{1}_{\substack{s>0,\\ s \text{ in the below}\\ \text{the } p\%\text{percentile}}} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i \in [5,7.5)}\right)$$

$$f_4(e_1, \cdots, e_s) = \mathbb{1}_{\substack{s>0,\\ s \text{ in the below}\\ \text{the } p\%\text{percentile}}} \log_{10}\left(1 + \sum_{i=1}^{s} \mathbb{1}_{e_i > 7.5}\right)$$

We can see in Figure 5 for this transformation applied to a cluster with $p \in \{99, 95, 90, 75, 50, 25\}$.

3.2. **Training and Testing Protocols.** We consider a CNN with a generic input of a 128×128 images with k channels using TensorFlow. According to the *model.summary()* output, the structure of our CNN is given in Figure 6

The choice of this format is to attempt to mirror the success of [4]. Our model is nearly identical to theirs with the exclusion of their dropout layers. In practice, we found that these layers are unnecessary for this task. A key feature of this model is the global average pooling, which hopefully gives some type of invariance to small rotations and translations in the image. For a four channels image ($k = 4$), Figure 7 gives a visualization of our network architecture. For each of our convolutional layers we considered a 3×3 kernel, for each max pooling layer we considered a 2×2 window with a 2×2 stride.

For both our training and testing data set, we consider all 90° rotations and axial flips. The reason for training is to hopefully make our model robust to natural symmetries that would be found in clusters in real life. Since the rotation and axial flip we observe for our data is essentially random, it is important to test the model's performance on all such transformations, motivating us to apply this change to the test set as well. As an example, here is a sample image and two of its corresponding rotations/flips can be seen in Figure 8. Importantly, we do this transformation only after we conduct a train/test split of our data. This is to ensure that the model does not train

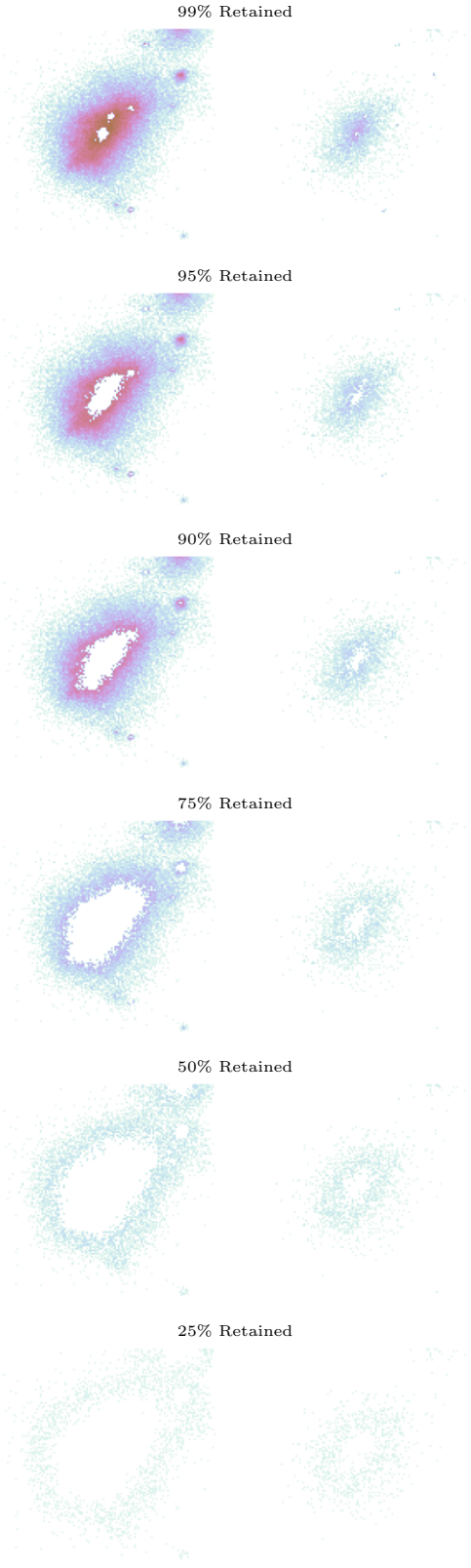

FIGURE 5. The (0,2.5) and (2.5,5) energy bands from the top galaxy in 4 percent retained transformation, for reference we have keep the color bar the same as the corresponding image in Figure 4

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 16)	592
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 64)	0
dense (Dense)	(None, 200)	13000
dense_1 (Dense)	(None, 100)	20100
dense_2 (Dense)	(None, 20)	2020
dense_3 (Dense)	(None, 1)	21

Total params: 58869 (229.96 KB)
Trainable params: 58869 (229.96 KB)
Non-trainable params: 0 (0.00 Byte)

FIGURE 6. The structure of our CNN according to Tensorflow's *model.summary()* output.

and test on the same galaxy, just with different rotations.

For each of our data transformation (Single Band, Multi-Band, Core-Removed and AGN-less) we implemented the same training protocol. Training occurred for 100 epochs at a batch size of 16. We used ADAM optimization at a learning rate of .0005 (half of the default). Throughout training, we also shuffled the order of our dataset at each epoch.

3.3. **Training Results.** For each transformation, we consider 10 cross-validation folds. For each fold we had a 90/10 training/testing split that was randomly generated (with a fixed seed for each transformation) we then collected the residuals for each data point when it occurred in the testing set and got the empirical bias and standard deviation for each given transformation. These results are collected in Table 1.

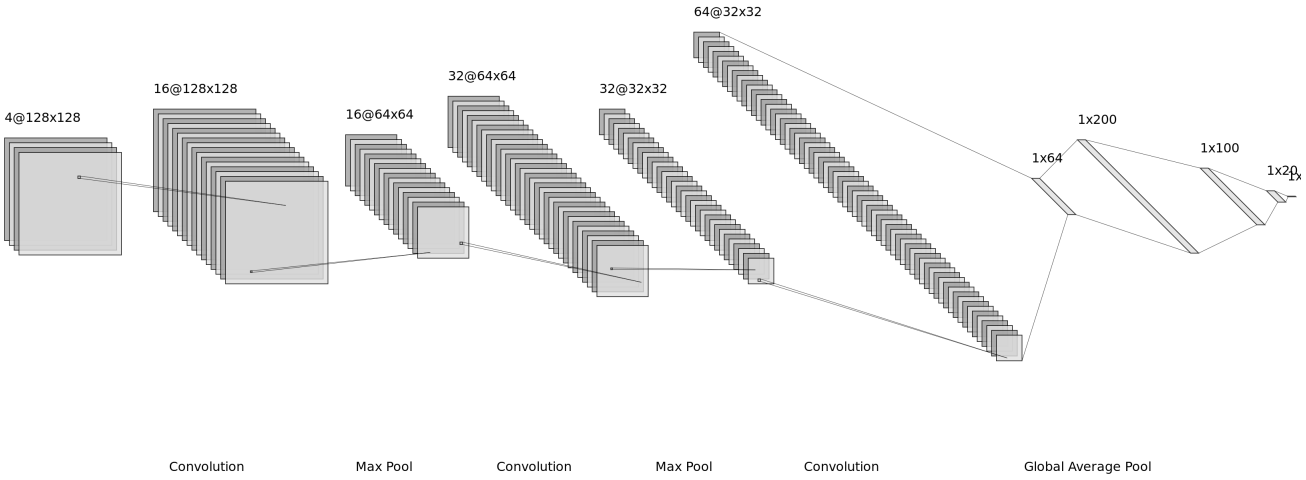

FIGURE 7. A graphical visualization of our model architecture when the input is a four channel image.

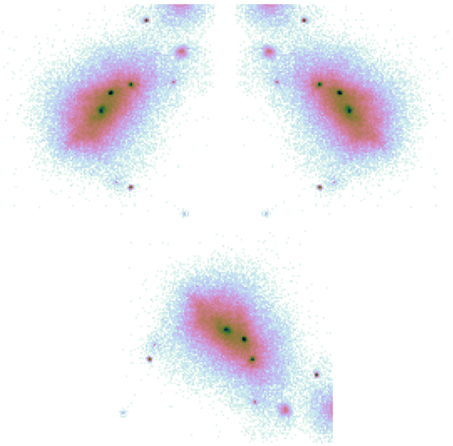

FIGURE 8. An example of rotation and axial flips that we have applied to both our training and testing sets, note that this transformation was applied after splitting so the model will not train on one rotation and test on another rotation of the same galaxy.

Model Type	Channels	Bias	Std Dev
Single Band	0+	.00086	.11253
Multi-Band	0-2.5	-.00173	.11744
Multi-Band	0-2.5,2.5-5	.00317	.06345
Multi-Band	0-2.5, $\cdots$, 5-7.5	-.00059	.05903
Multi-Band	0-2.5, $\cdots$, 7.5+	.00333	**.05818**
99% Retained	0-2.5, $\cdots$, 7.5+	**.00013**	.05903
95% Retained	0-2.5, $\cdots$, 7.5+	.00075	.06131
90% Retained	0-2.5, $\cdots$, 7.5+	-.00161	.06076
75% Retained	0-2.5, $\cdots$, 7.5+	.00365	.06253
50% Retained	0-2.5, $\cdots$, 7.5+	.00190	.06363
25% Retained	0-2.5, $\cdots$, 7.5+	-.00438	.06481

TABLE 1. The empirical bias and standard deviation of each of our proposed models, bold is the best overall.

We can clearly see that the major difference in predictive accuracy between models is when we decide to split the energy bands into more than one group, nearly halving the loss of our model. This is peculiar considering that our core-removed model does not show a similar behavior. This means that there is some fundamentally important information in our image that is revealed after splitting energy bins that go beyond ignoring the AGN, which may be unrelated to the overall mass of the cluster. This is further justified by our results with the AGN-less model, as they achieve close to the same level of prediction as the multi-band images yet lack the AGN spike features in the image. This suggests that a different phenomenon is responsible for the improvement when adding higher energy bands. Perhaps more interesting is that there is immediately a rate of diminishing returns on adding more energy bands, as we can see by the minor improvements when adding in the higher energy band channels.

It is also worth noting how our results compare to [4]. In [4] they had around 7000 observations, in contrast to our 3300, and their clusters were much more well conditioned compared to our more realistic observations. This is a possible explanation for why they are able to get a standard deviation of $\sigma = .051$ compared to our best case of $\sigma = .06196$. It is believed that with more data and perhaps some model architecture improvements that we can approach this level of accuracy.

In addition to our evaluation metrics, we can also plot the histogram of residuals and a scatter plot of true

responses versus predictions for each data transformation, these can be seen in Figure 9.

4. Why Does Energy Binning Work?

Based on our results in the previous section, it seems like splitting energy bands induces an increase in the predictive capabilities of our CNN. The goal of this section is to try to understand why the higher energy bands aid in the prediction of the cluster masses through an evaluation of their training dynamics.

4.1. **Channel Dynamics Tracking.** Naively comparing the learned features of our single channels models to our multi-channels models is apriori difficult, since there is not a guaranteed relationship between the two models training dynamics. After initialization, both the single and multi-band models may learn fundamentally different features that make it difficult to understand what is the added benefit to having split energy bands and on what types of galaxies is this extra information relevant. In order to bypass this issue, we used a special training protocol. Over time, this protocol feeds the model subsequent higher valued energy bands in an attempt to track how the model's features change before and after this new information is given. To be more specific, our training protocol follows the following four phases:

Phase 1: We first zero out every channel except the one with the lowest energy band, we train the model for 50 epochs[4]. We then add a small amount of Gaussian noise to the weights of the model to remove the model from a local minimum. This step is vital as if two little noise is added then overfitting will occur as the model was already near interpolation, too much noise, and one would essentially restart the dynamics at initialization. For future work, the noise level will need to be changed on a problem by problem basis. The model is then further trained for one epoch.

Phase 2-4: We now add in the next highest binning channel by removing its zeros. We train again for 50 epochs, apply a permutation for the weights, and then train for an additional epoch.

Our goal is to understand what new information and features has been provided by the higher energy bands. To accomplish this, we provide a high level visualization for what galactic clusters we are able to predict more accurately with the added energy bands. This visualization[5] can be seen in Figure 10. On the X-axis we have the number of epochs of overall training (notice that each phase has 50 epochs) with while lines separating each phase. For each column, we have plotted the prediction by the model on all unobserved test observations. We have ordered the test observations in ascending order based on their response (as seen in the final column of the image). This helps us to understand what ranges of clusters the model is able to sort better before and after adding additional information.

Three observations from Figure 10:

(1) Going from a single channel of data to many channels helps the colors better align with the last column, thus giving us higher quality predictions. There is no noticeable improvement for subsequent channels with respect to the 2nd channel's addition.

(2) The single channels model seems to be able to get the general ordering of the test points, particularly at the extremes of the mass range. The second channel's addition is able to sort these images at a finer resolution, especially those in the medium range of mass. Thus, we would expect that clusters in this section of the range have nuanced features that splitting energy bands are required to understand.

(3) We are also able to identify training examples which received bad predictions with a single band but then become much improved after adding subsequent channels of information (for example, in the 50-150 range). This technique is also able to identify testing points which our model cannot figure out even with the full four channels of information (in line 249), this is helpful to know what may be fundamentally hard observations to understand; further analysis can be done on these observations.

4.2. **Dynamical Comparison of Multi-Band versus AGN-Less Models.** In Table 3.3 we saw that there seems to be very little difference to the predictive capabilities of our models when feed the original X-ray image versus an AGN-less one. Moreover, the energy splitting phenomena between the two data transformations is quite similar as well. Using our dynamics visualization, we can attempt to track the differences in the dynamics for these two models. In

[4]This value was tested empirically to show good results

[5]As an aside, of all the work that I did for the Interpretable ML Astronomy group, this was the one thing they liked the most. Apparently this visualization is the first of its kind and I think I might try to build a library of these things since they give a nice picture of how the model's dynamics evolve with new information.

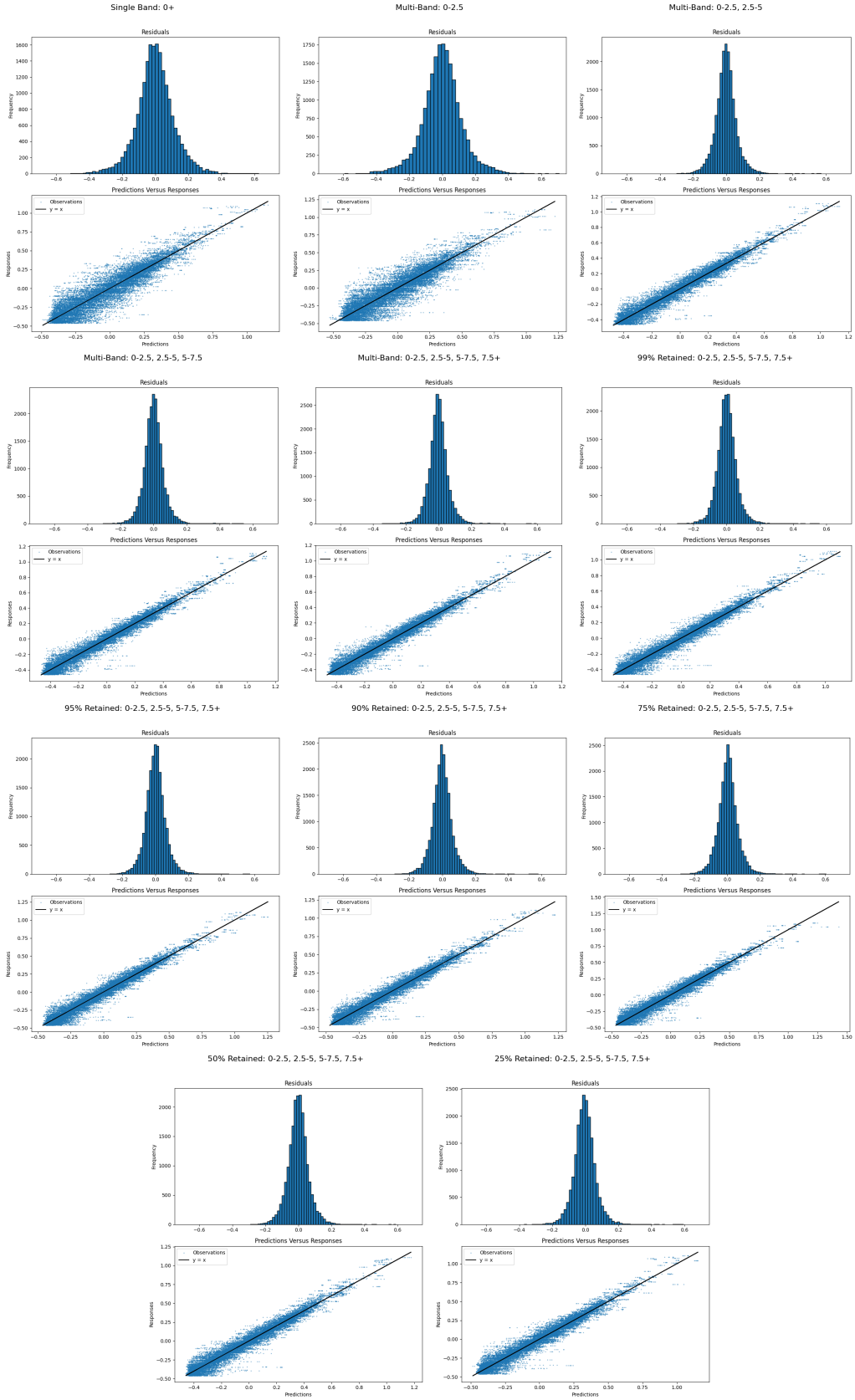

FIGURE 9. Here we have both histograms of our test prediction residuals over each of our cross-validated folds and a scatter plot of true responses versus predictions with a reference line for perfect accuracy. These transformations are, going left to right row-by-row: Single Band, Multi-Band (with subsequent added channels), Core-Removed, AGN-less (with subsequent added channels).

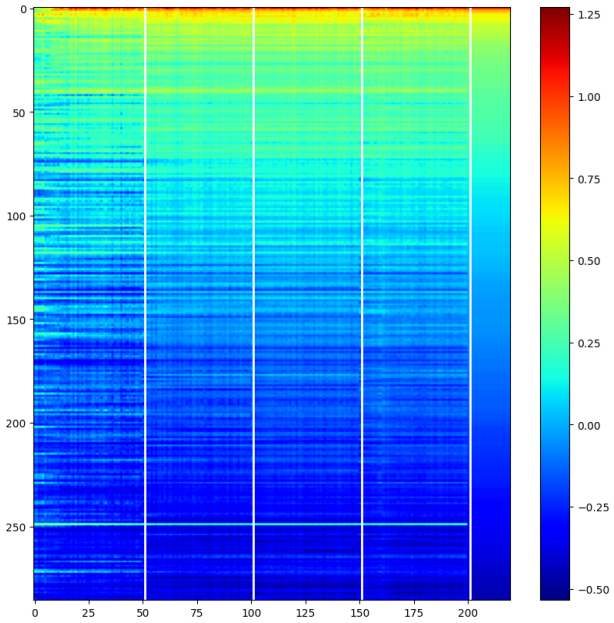

FIGURE 10. Our novel dynamical visualization: The X-axis is the overall number of epochs trained, we have white lines separating each phase as described in Section 4. On the Y-axis are our test observations ordered in ascending true response (as seen in the last column) centered by the mean. Looking left to right, we can see how a test point observations changes through training. Looking up to down, we can see at a given epoch how well our model is able to sort our data points in testing.

Figure 11 we plot the two dynamics side by side to see how they differ.

Using Figure 11 we can compare the training dynamics to make even stronger science claims about the predicting abilities about the split-energy band models. At first glance, the two dynamics visualizations appear nearly identical. They both show moderate disorder in the medium range of mass, which gets sorted out with additional energy channels. Perhaps the AGN-less model's ordering looks slightly more rough, but the large scale predictions are the same. After adding additional band there is some variation between the two models in the 50-100 range but nothing that is significantly different.

4.3. **Main Science Claim.** Due to the strong similarity between the two dynamics and the similarity in both models predictions, we can confidently state that the improved accuracy from splitting the energy bands in independent of the AGN spikes in the image. Furthermore, it seems that the model is able to provide reasonable predictions with and without AGN on a similar set of images, giving credence to the concept that these models "ignore" the core of the image. This means that any future predictor for galaxy mass can safely ignore the AGN spikes in the X-ray images.

5. Conclusion and Future Directions

This report provides two contributions to the Interpretable ML Astronomy Literature

(1) We demonstrated the split energy band phenomena, where splitting the logarithmic counts of observations by energy band provides an unexpected boost in predictive accuracy. We also showed that the phenomena mostly occurs with one additional channel, adding further channels has diminishing returns.

(2) We showed that the hypothesis that energy band splitting helps the model ignore AGN better is most likely untrue. This was seen through similar predictive performances of a AGN-less image in Table 3.3 and by seeing the similar training dynamics using our dynamical visualization in Figure 11.

This leaves the next logical question: What feature is gained by our neural network when we add additional energy band channels. By the use of our dynamical visualization, we conjecture that the structure of observations for clusters in the medium range of mass is more complex and separating the energy bands provides a way for the model to better understand its mass. Future work could consist of applying saliency methods such as integrated gradients or LIME [5] to get a better understanding of what parts of the image are important to the network before and after splitting the energy bands. Another possibility would be to try to better cluster our images by their penultimate layer embedding to get a better sense of which galaxies have improved accuracy after subsequent energy bands have been added. Both of these are ongoing lines of research.

It is also of independent interest to apply the dynamical visualization in Figure 10 & 11 to other problem and start to build a new collection of machine learning interpretability tools that focus more on how a model develops features over the course of training than what are the final features are at prediction. This may aid in separating the complex features

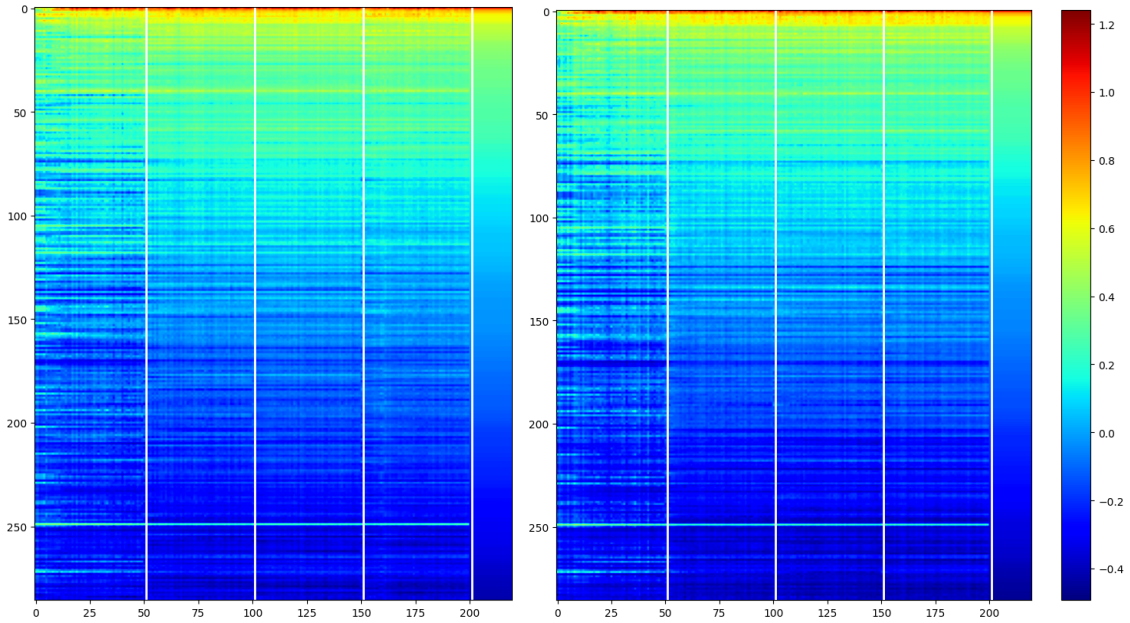

FIGURE 11. (Left) The dynamics of the original Multi-band model as seen in Figure 10. (Right) The dynamics of the AGN-less model.

that a neural network learns into small incremental changes during training.

ACKNOWLEDGEMENTS

I would like to thank Zehao Dou, Naomi Gluck, Matt Ho and Daisuke Nagai for committing their time to regular meetings to help introduce me to the interplay of astronomy and machine learning. Their comments always helped me better focus my ideas and gave me a more scientific prospective on machine learning. I hope to continue working with them and the interpretable ML group in the future.

REFERENCES

[1] Thomas Dauser, Sebastian Falkner, Maximilian Lorenz, Christian Kirsch, Philippe Peille, Edoardo Cucchetti, Christian Schmid, Thorsten Brand, Mirjam Oertel, Randall Smith, and Jörn Wilms. Sixte: a generic x-ray instrument simulation toolkit. *Astronomy amp; Astrophysics*, 630:A66, September 2019.

[2] N. Gupta and C. L. Reichardt. Mass estimation of galaxy clusters with deep learning ii. cosmic microwave background cluster lensing. *The Astrophysical Journal*, 923(1):96, dec 2021.

[3] M. Ho. *Deep Learning for Dynamical Mass Estimation of Galaxy Clusters*. PhD thesis, Carnegie Mellon University, 2022.

[4] M. Ntampaka, J. ZuHone, D. Eisenstein, D. Nagai, A. Vikhlinin, L. Hernquist, F. Marinacci, D. Nelson, R. Pakmor, A. Pillepich, P. Torrey, and M. Vogelsberger. A deep learning approach to galaxy cluster x-ray masses. *The Astrophysical Journal*, 876(1):82, May 2019.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[6] Z Yan, A J Mead, L Van Waerbeke, G Hinshaw, and I G McCarthy. Galaxy cluster mass estimation with deep learning and hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 499(3):3445–3458, October 2020.