Draft version April 26, 2021
Typeset using LaTeX default style in AASTeX63

Pure B Filtering with a Convolutional Neural Network

Charles Skerbec, Advisor: Dr. Colin Bischoff[1]

[1] *Department of Physics, University of Cincinnati, Cincinnati OH 45221, USA*

ABSTRACT

By masking cosmic microwave background (CMB) polarization maps, EB leakage occurs, namely at the edges at the mask. To detect B-mode polarization, filtering out E-mode polarization is a necessary step as well as filtering out EB leakage. Utilizing a convolutional neural network (CNN) with an architecture similar to UNet, the network is trained to filter CMB polarization maps. This method could prove to be a computationally efficient method in searching for B-modes. B-mode polarization at low ℓ values is a prediction of inflation theory. The search for B-modes in the CMB is to provide strong experimental evidence for primordial gravitational waves. Correlation of power spectra and bandpower window function were used to test the effectiveness of the CNN filter.

1. INTRODUCTION

The Big Bang was first proposed in 1927 by Georges Lemaître. In 1929, Edwin Hubble made an observation that farther galaxies were moving faster away than closer ones, and formulated the famous result, Hubble's Law. This showed that the universe was still expanding. This implied that because celestial bodies were still moving apart, then they were together at some point. At the beginning of the universe, the density and temperature were extremely high and an opaque fog of hot ionized gas existed. Photons could not freely travel without scattering. As the universe expanded and cooled, neutral atoms formed, and the photons could travel freely through the universe. These photons scattered for the last time as the ionized plasma transitioned to neutral gas. The light traveling from this moment is seen today as the cosmic microwave background (CMB), a prediction of the Big Bang. This period of time where neutral atoms formed was about 380,000 years after the Big Bang. The CMB was discovered by Penzias and Wilson in 1964, giving strong evidence for the Big Bang. The CMB has a thermal black body spectrum at a temperature of approximately 2.73 K.

A fraction of CMB photons were linearly polarized due to Thomson scattering off free electrons from the moment of last scattering. Polarization describes the orientation of light perpendicular to the direction of propagation. The theory of inflation explains the universe expanded extremely rapidly in a fraction of a second after the big bang. A consequence of the theory leads to the prediction of a background of gravitational waves in the early universe. These faint gravitational waves can be observed by studying the mark they left on the CMB in the form of polarization patterns [1].

1.1. *Polarization*

CMB polarization was first discovered in 2002 by the DASI telescope. Polarization of CMB photons can be caused by density perturbations or tensor perturbations. There are two types of polarization of concern: E mode and B mode polarization. Density or scalar perturbations only generate a specific polarization pattern, E modes. The CMB is dominated by mostly E modes. Gravitational waves create both E and B modes, so detecting B modes would give observational evidence that in the early universe, a background of gravitational waves was present. Models of inflation predict these B modes will be present at angular scales of about a degree or higher (at low spherical multipole moment, ℓ). To detect B-modes, this requires decomposition of E and B-mode polarization. E and B modes can be geometrically seen in Fig. 1 below. E-modes run parallel or perpendicular to the wave vector, $\vec{k}$, while B-modes run at 45° to $\vec{k}$.

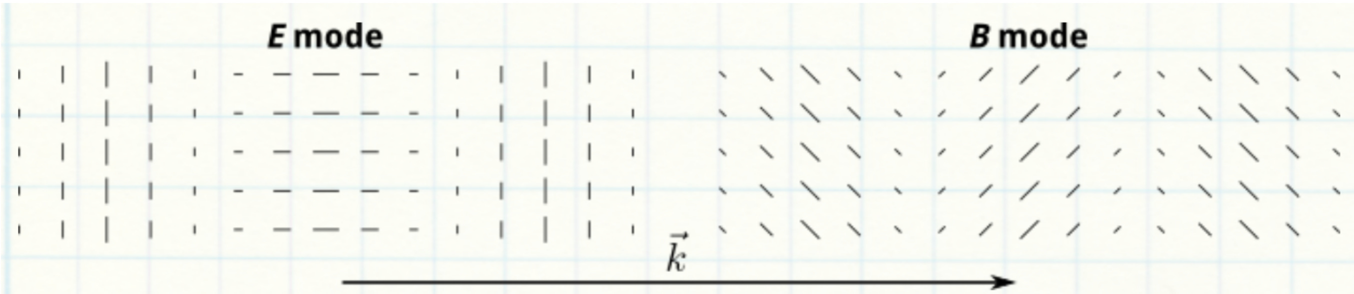

Figure 1. Polarization shown by headless vector lines across a horizontal wave vector, $\vec{k}$. The polarization of an E mode is parallel or perpendicular to the wave vector, and the polarization of a B mode is rotated 45° with respect to $\vec{k}$.

To detect B-mode polarization, E-modes must be separated from the B-modes in polarization maps. E and B-modes can be transformed into Stokes parameters Q and U, where circular polarization is not excepted in the CMB. To translate from E and B to Q and U, the modes can undergo the rotation matrix, [6],

$$\begin{pmatrix} Q \\ U \end{pmatrix} = \begin{pmatrix} cos(2\phi) - sin(2\phi) \\ sin(2\phi) + cos(2\phi) \end{pmatrix} \cdot \begin{pmatrix} E \\ B \end{pmatrix}. \tag{1}$$

Polarization maps were expressed as Q/U maps and can be transformed back to E and B modes with the inverse rotation matrix from Eq. 1. Since E-modes dominate the CMB, they leave a distinct pattern in Q and U maps that can be seen in the left image of Fig. 2. B-modes leave a diagonal pattern in Q maps (horizontal and vertical in U maps) as seen in the right image of Fig. 2. The intensity is much lower in the Q map when dominated by B-modes compared to E-modes. Here a small portion of the sky, 20 by 20 degrees were taken in these simulated maps.

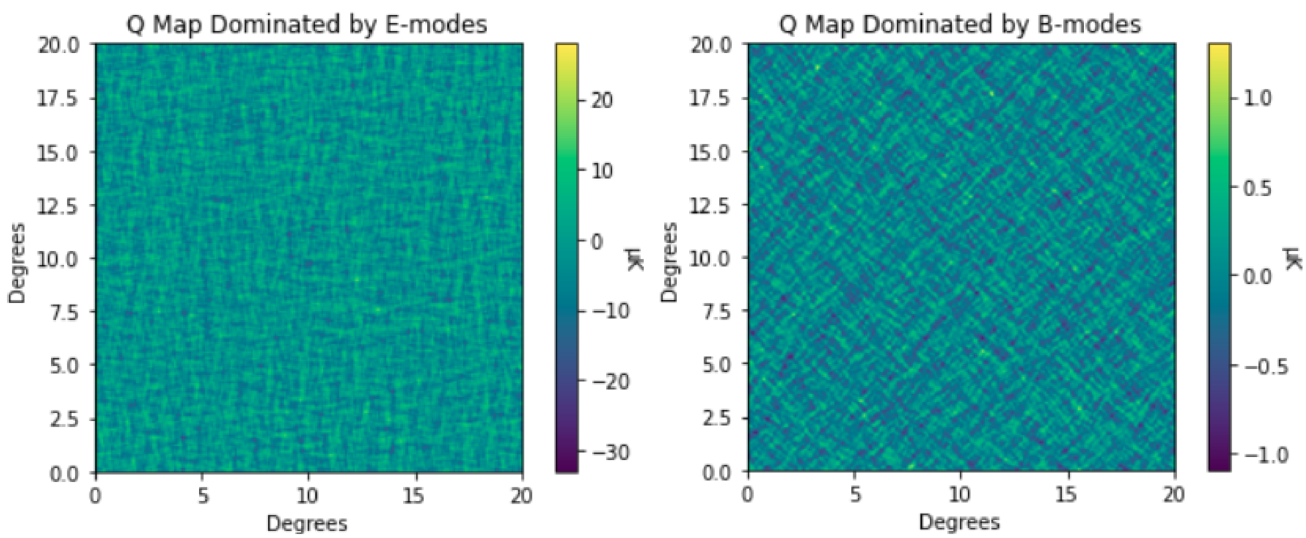

Figure 2. Polarization maps displaying Q maps dominated by E-modes (on left) and dominated by B-modes(on right). In E/B dominated U maps, U would display diagonal patterns from E-modes and horizontal/vertical patterns from B-modes.

1.2. *EB Leakage*

Limited patches of sky restrict analysis to a subset of polarization modes obtained from observations. Polarization maps must be masked to remove pixels with high noise, but this creates a problem known as EB leakage [2]. When Fourier transforming, power is inputted at each ℓ value when attempting to go across masked pixels this results in leaked E to B modes, apparent around the edges of the mask as seen in Fig. 3. To create an example of this leakage, start from zeroed out B modes and apply a mask to the polarization maps that contain only E-modes. The E modes are zeroed out after the mask is applied and the maps are transformed to Q/U maps, displaying only B-modes from EB leakage.

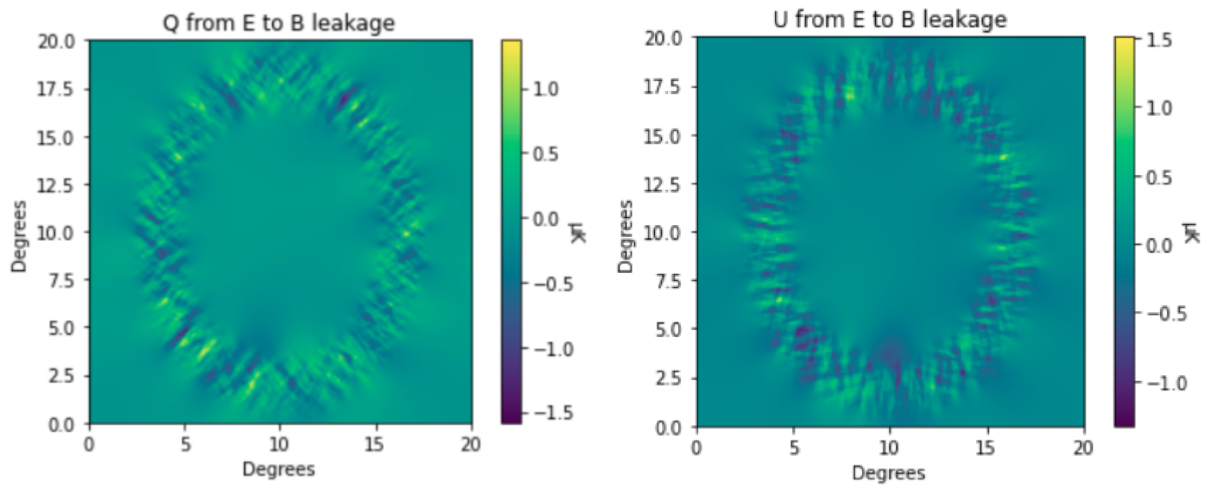

Figure 3. Polarization maps displaying EB leakage at the edges of a mask.

A method to solve both the problems of decomposing polarization maps into E and B modes separately and dealing with the issue of EB leakage, is using a machine learning model.

2. MODEL DISCUSSION

Current image processing methods to fix the problem of EB leakage are computationally expensive. The goal is to fix this problem of EB leakage through a machine learning model trained to filter out E mode polarization and EB leakage. The model is regression based and requires at minimum two inputs to begin training. One is input data and one is "target" data or what the model tries to learn what that input should transform into. It attempts to change input maps that contain both E and B modes (expressed as Q/U polarization maps) and filter out leakage and E-modes. The target data maps are also Q/U maps, but they contain only B-modes as discussed further in CMB Maps section. These maps can be seen at the top of Fig. 5. Convolutional neural networks (CNNs) are able to reduce images into a form that is easier for the model to process without losing important features, namely polarization patterns at different resolutions.

Neural networks have a collection of nodes interconnected called neurons. Each neuron can transmit information to other neurons. Connections of neurons are modeled with arrays of weights. The weights are multiplied to the inputs and summed. Then an activation function is applied to modify the value of the output. CNNs are also very capable of processing large datasets. CNNs pass information through layers where each layer performs an operation on the data.

2.1. *Convolutional Layers*

Convolutional neural networks are comprised of layers of convolutional operations. A convolution is a linear operation that acts on a given data set with a multiplication of weights. The array of weights are called a filter or kernel. The kernel is smaller than the input data and travels across the data during a convolution layer. The kernel performs element-wise multiplication on the data. Kernels are learned during training for optimizing the weights during training. A kernel can be given a stride length or the amount of pixels the array shifts over per element-wise multiplication. This convolution operation can be seen in detail in Fig. 4. An example of a 3x3 kernel is given and multiplied (element-wise) across the input data, then summed together.

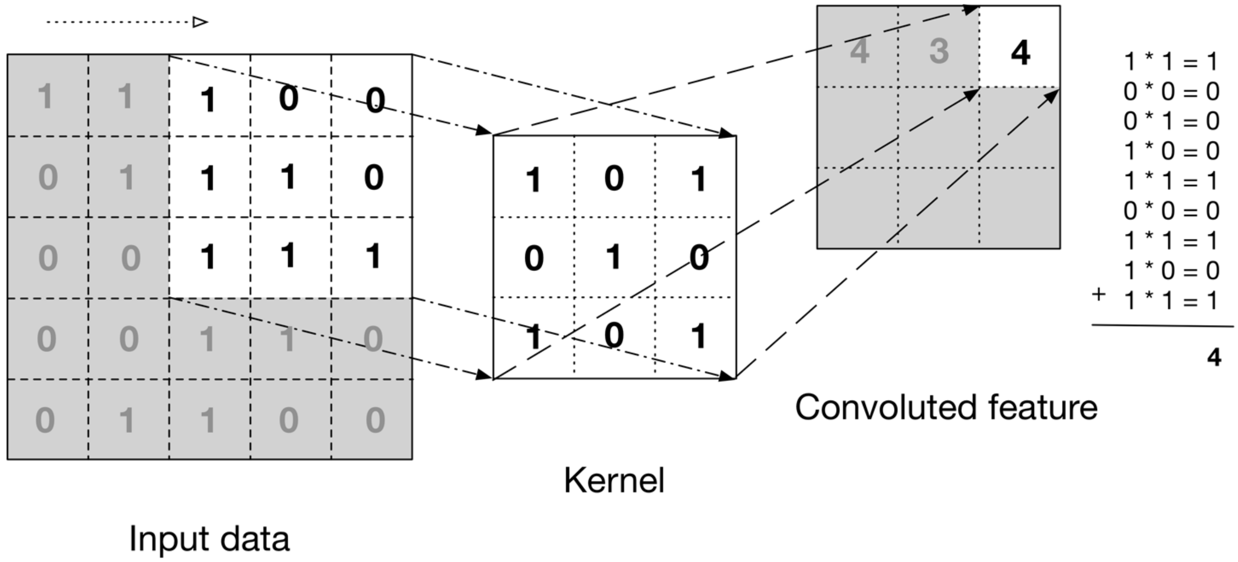

Figure 4. Example of convolutional layer with a kernel size of 3x3 across a 2 dimensional input data array. The layer applies a dot product operation of the kernel and input data taking strides across the input data from left to right and down from the top of the input array [5].

There are two options for convolutions in terms of padding. Same padding retains the dimension of the original image while valid padding prevents the kernel operation around the edges. Valid padding causes the borders to be padded, preventing convolution operations. Valid padding was used for all convolutions. This however, caused trimming of the dimensions every convolutional layer. This trimming can be seen in Fig. 5 labeled by the green arrows.

2.2. *Loss Function*

The model utilizes a loss function to predict numerical quantities. The mean square error (MSE) calculates the average squared difference between the input data and target data where a value closer to zero is desired. The MSE is given as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - T_i)^2, \tag{2}$$

where n is the number of data points, Y_i is the input data (after passing through the model), and T_i is the target data. The model minimizes the value of the loss function with more training. The model was trained over 100 epochs (amount of passes through data) with epochs over 150 resulting in overtraining and poor results. More epochs may return better results, but methods to reduce overtraining would need to be considered such as adding dropout layers within the model.

2.3. *Network Pathways and Skip Connections*

The model consists of encoder and decoder paths that downscale and upscale the images. The model follows a similar model as UNet [4] a model made by Münchmeyer and Smith [3]. The encoder and decoder paths are particularly useful as the model learns about the polarization patterns at different resolutions. The model retains the information it learns from the encoder path at low resolutions over to the decoder path through skip connections. A separate path for the mask is added to add emphasis on certain pixels. The details of the model can be seen in Fig. 5 with each (red or blue) arrow representing a new layer in the model. Red arrows represent two-dimensional convolutional layers where the image's resolution is reduced by a factor of 2. Blue arrows represent two-dimensional upsampling or increasing the image's resolution by a factor of two. The model followed closely to the architecture as UNet. Skip connections were applied along the low resolutions. Skip connections used were concatenations from the encoder path to the decoder path. Adding skip connections improved the correlation calculation explained further in the Results section.

Two pathways were created in the model. One performing linear operations with no activation function and the other pathway performing non-linear activation functions. The linear pathway was created to keep operations on the polarization maps linear, a common practice in CMB filtering. The non-linear pathway consisted of two dimensions of smooth masks (following the same two dimensions of polarization types, Q/U). Rectified Linear Unit (ReLU) was the activation function used in the non-linear pathway, a commonly used activation function. ReLU is given by $y = x^+$ where negative-numbers all return zero. The non-linear and linear pathways were pixel-wise multiplied together at each layer.

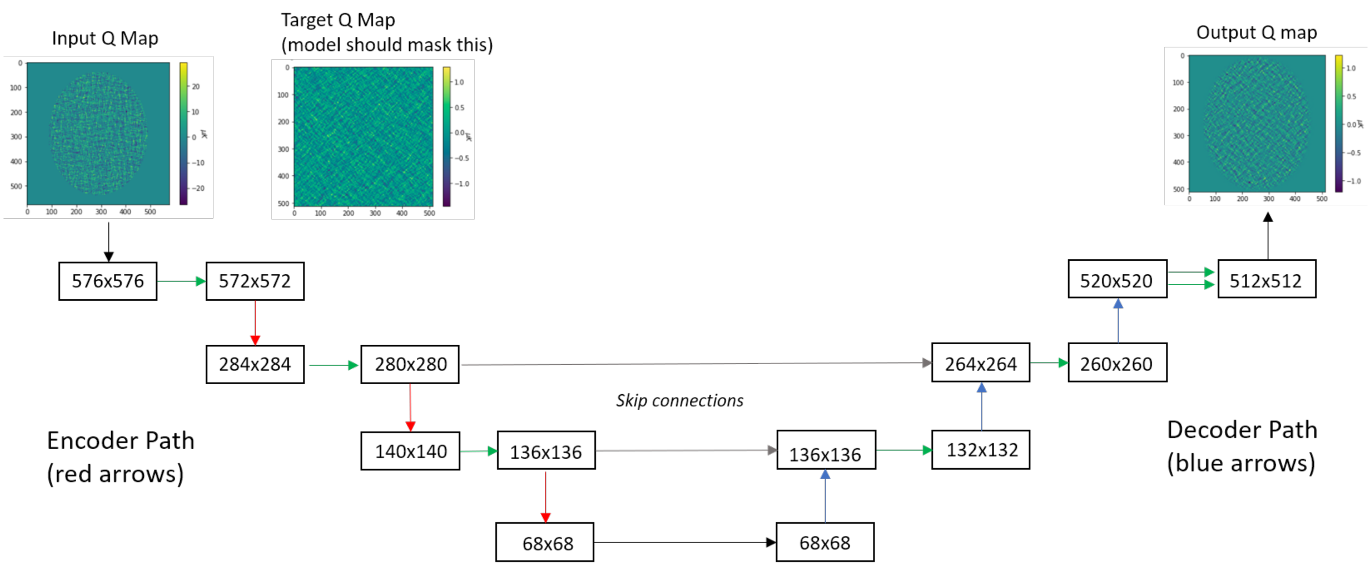

Figure 5. Model architecture showing input map size of 576x576 pixels scaling down through encoder path and upsampling through the decoder path to 512x512 pixels. This model was is an example of a four layer model. An additional layer was added into the encoder and decoder path for the final model (this changes input resolution to 640x640 pixels and retains output resolution of 512x512). Separate mask pathway not shown but is identical to path of polarization maps. The mask's encoder/decoder path is applied to the polarization paths through multiplication layers. Skip connections are applied for the model to retain information from low resolutions.

3. MODEL IMPLEMENTATION

The model was written using Python and a combination of Keras and Tensorflow libraries. The model's inputs followed examples done with CNNs on RGB images with multidimensional arrays where instead polarization types (Q/U) were in place of the RGB dimensions. The model used was 5 layers deep. Training took between one to two minutes per epoch (epochs will be discussed further in the Model Parameters section) for a total of somewhere around 3 hours to train the model through 100 epochs. The model was trained on a host computer, NERSC Cori.

3.1. *CMB Maps*

The simulated polarization maps were created from angular power spectra data created using CAMB. The Q and U maps had a pixel size of 512x512 and an angular size of 20 degrees by 20 degrees, a small region of the sky to approximate the maps as flat. Using this approximation, two-dimensional Fourier transforms could be used instead of computationally heavy spherical harmonic transforms. B-mode polarization was created with gravitational lensing. From the angular power spectra data, E and B maps were produced by linearly interpolating the square root of the power spectra data over Fourier space. Maps were randomly generated and transformed to Fourier space and temperature and E-modes had partial correlation. These randomized maps were multiplied to the E and B maps. E and B modes were transformed to Q and U by a spin-2 rotation matrix, seen in Eq. 1 and converted to pixel space.

Q and U maps were then masked with a binary mask sharing the same pixel size that masked the edges of the maps. Masking is necessary in real CMB polarization maps to remove pixels containing a high amount of noise. These maps were padded along the edges to obtain a 640x640 pixel size for the model. These inputs can be seen in Fig. 7. These polarization maps contain both E and B-mode polarization.

Layer	Pixel size	# filters	Kernel size	Stride
Conv2D	640x640	16	(5,5)	(1,1)
Conv2D	636x636	16	(5,5)	(2,2)
Conv2D	316x316	32	(5,5)	(2,2)
Conv2D	156x156	32	(5,5)	(2,2)
Conv2D	76x76	32	(5,5)	(2,2)
Conv2D + Upsampling2D	36x36	32	(5,5)	(1,1)
Conv2D + Upsampling2D	68x68	32	(5,5)	(1,1)
Conv2D + Upsampling2D	132x132	32	(5,5)	(1,1)
Conv2D + Upsampling2D	260x260	16	(5,5)	(1,1)
Conv2D + Upsampling2D	516x516	16	(5,5)	(1,1)
Output	512x512x2			

Figure 6. Model outline displaying encoder/decoder path of linear pathway polarization maps undergo. Kernel size was set at 5x5 for all layers including non-linear path. Non-linear path followed same parameters as linear path with one less layer in the decoder path. The non-linear path gets to an output of 516x516 pixels.

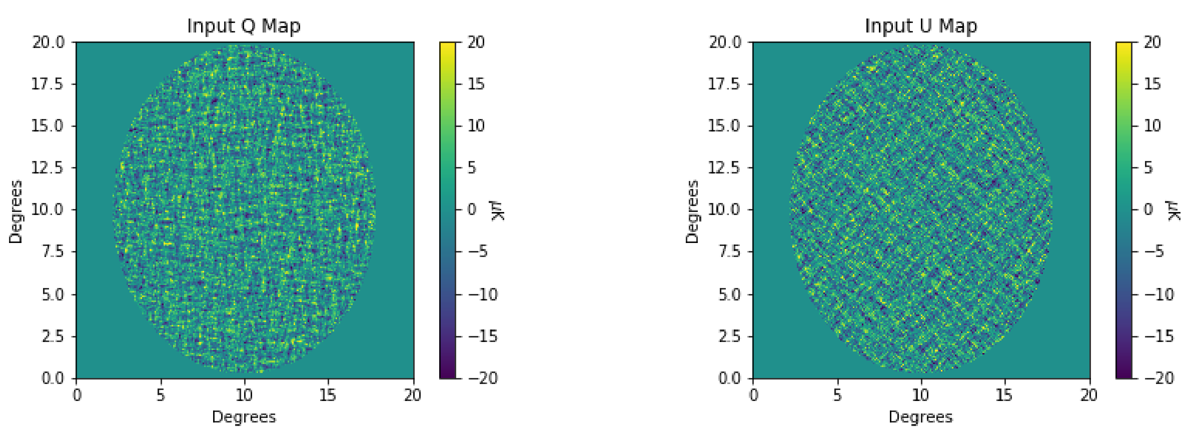

Figure 7. 100 input maps of each polarization type was created to train the model. These are similar to maps shown in Fig. 2, but have a binary mask around the edges.

The masked maps Q and U maps containing both E and B-mode polarization were the first two dimensions of input maps to the CNN model. The last two dimensions of the input maps to the CNN model were 640x640 pixel sized smooth masks where the center had a value of 1 and progressively move to values of 0 around the edge of the maps. Target Q and U maps were created in a similar way to the input Q/U maps, but E modes were taken out. The best results from the model were with target maps left unmasked. These target maps can be seen in Fig. 8. These are B-mode dominated maps with a lower signal compared to Fig. 7 which contained both polarization types.

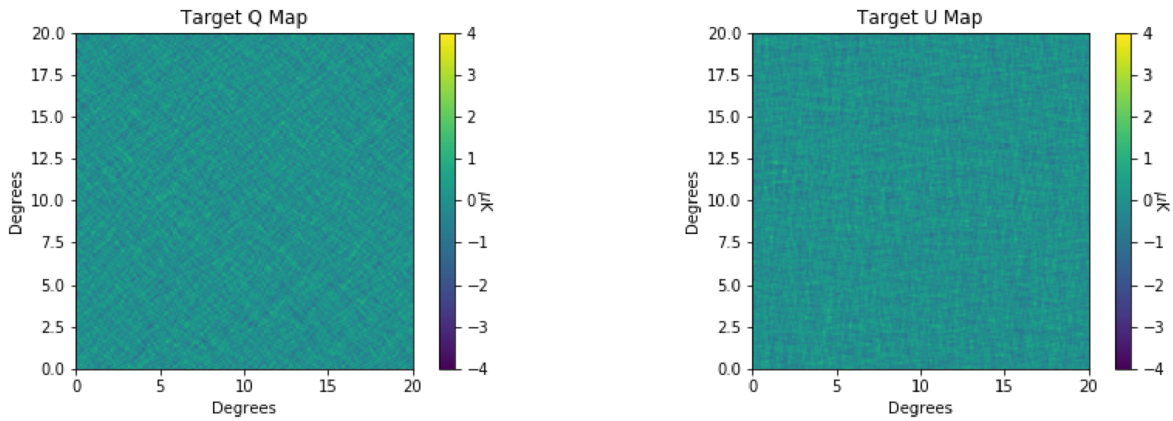

Figure 8. Target maps in the CNN model, containing only B-mode polarization.

New Q/U maps were created similarly to the maps in Fig. 7 as inputs to the trained model. Once the model was trained, the time to filter maps only took a few minutes. Visually, the model was able to filter out E-mode polarization as the output maps in Fig. 9 display B-mode polarization patterns in Q/U maps. The signal also matched around the signal intensity of B-modes. The model appears to continue the B-mode pattern outside the confines of the mask.

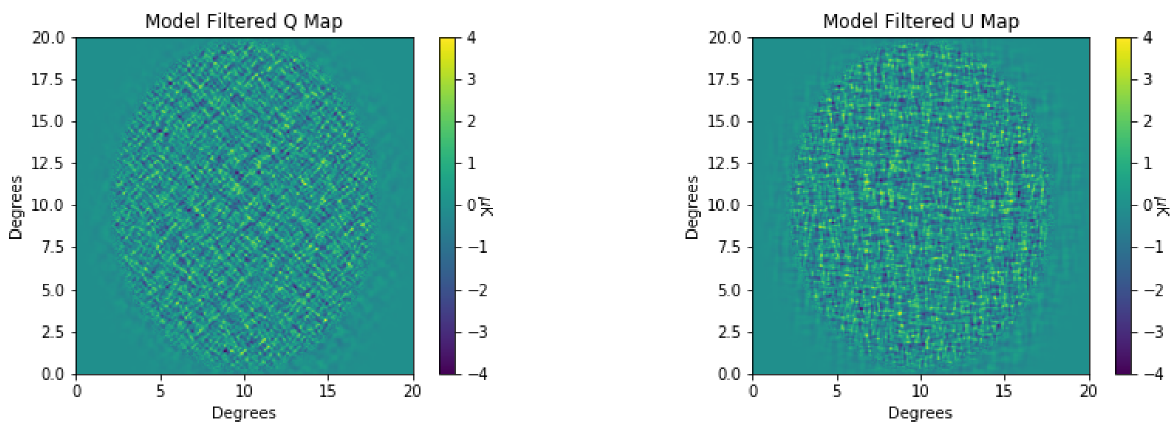

Figure 9. Example of output maps of CNN filter.

3.2. *Angular Power Spectrum*

Angular power spectra is a method to measure the power of a signal over different angular scales (in spherical harmonic coordinates). Q and U maps are converted to Fourier space using a two-dimensional Fourier transform and rotated using the inverse of the rotation matrix in Eq. 1. The power is calculated by the absolute square of the desired spectrum. The original data was measured in D_ℓ related to C_ℓ by $D_\ell = \frac{\ell(\ell+1)}{2\pi} C_\ell$. Angular power spectra is measured in bins of ℓ where the absolute square is averaged over these bins.

3.3. *Model Parameters*

The model uses a gradient descent based algorithm called Adam. Setting the learning rate to 10^{-4} led to the best results. The Q/U polarization maps were standardized. The best results for masking the polarization maps were from using a binary mask on the polarization maps while leaving the target maps unmasked. 100 input maps and target maps were created for the data set. A set of 20 validation maps of similar input and target maps were also created to test the model in overtraining as it trained. The non-linear path or the mask path in the model followed the same number of input/target maps.

Epochs are the amount of times the data entire data set is passed through the model was set at 100. More epochs may result in marginally better filtering results, but is hindered by overtraining. A possible solution may be to add dropout layers within the model to increase the number of epochs possible. The model did not display any significant evidence for overtraining at 100 epochs. Two skip connections were placed in each pathway. There was an upper limit on the number of skip connections possible given the different number of resolution layers. The model performed slightly worse with less skip connections, using the analysis described in the cross spectra section.

The best performing model had 5 resolution layers where in the encoder path the resolution was brought down to 68x68 pixels and the decoder path upsampled the maps to a pixel size of 512x512 as seen in Fig. 5. The input maps were arrays with the following dimensions: (# maps, pixels (x), pixels (y), polarization/mask) where polarization/mask contained binary masked Q/U maps as the first two dimensions and a smooth mask in the last two dimensions. The array ordering follows TensorFlow's requirements.

4. RESULTS

Analyzing the outputs of the model was done in terms of the power spectra or cross spectra of the Q/U predictions made by the model.

4.1. *Correlation*

The cross spectra was calculated finding the absolute square of BB, C_ℓ^{BB}. This was calculated for the target Q/U maps and the predicted output of Q/U by the CNN model. Then the cross spectra was calculated, or $B^*_{target}B_{output}$ between B-modes for target Q/U maps and the predicted B-mode output by the CNN model. The correlation is given by

$$CC = \frac{C_\ell^{B_T^* B_O}}{(C_\ell^{B_T^* B_T})^{1/2}(C_\ell^{B_O^* B_O})^{1/2}}, \tag{3}$$

where CC is for correlation coefficient. This was the main test to determine if the model was performing at low ℓ values.

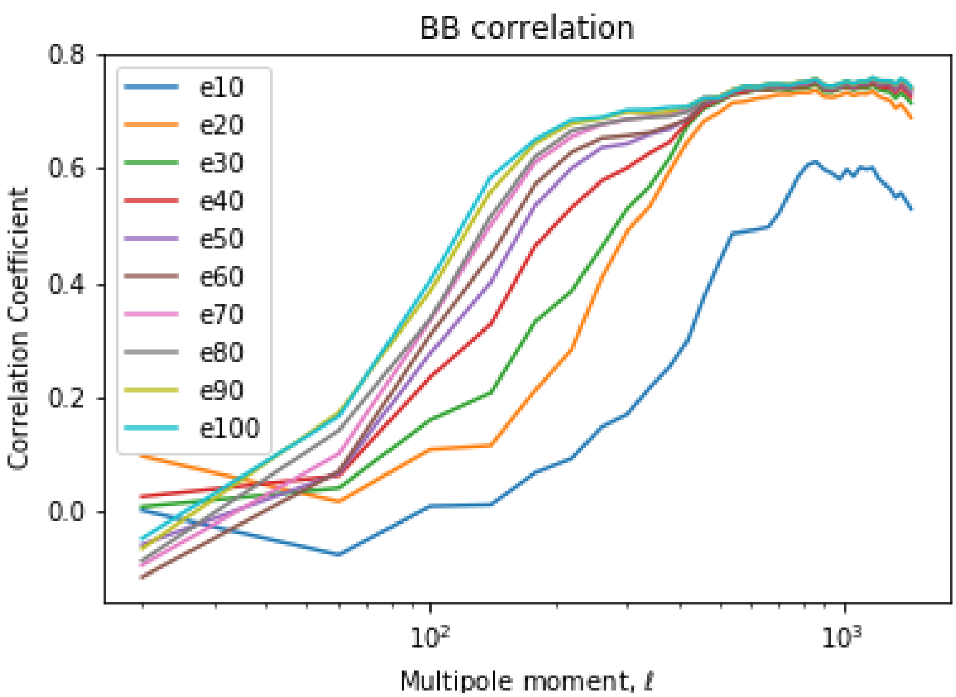

Figure 10. Correlation graph calculated with angular power spectra and cross spectra. Horizontal axis displays the multipole moment, ℓ from spherical harmonics and the vertical axis is calculated using Eq. 3. The horizontal axis was plotted with a log scale. The correlation was plotted over different number of epochs ranging from epochs 10 - 100 displayed with the legend on the left.

The graph shows a correlation climbs to about 0.70 at an ℓ value of about 150. An increase in epoch number shows an increase of correlation value with 10 epochs clearly giving the worst results. The five layer model was tested with more epochs, around 200 and this resulted in overtraining after about 140 epochs. This caused the correlation coefficients to dip rapidly. No significant increase in correlation occurred with more epochs after training the model with 100 epochs.

4.2. *Bandpower window function*

The bandpower window function (bpwf) was another method to test the effectiveness of the filter created by the model. The bpwf measures the signal of a small range of ℓ values over all frequencies. The power spectra was calculated with the same method described in the Angular Power Spectra section, but with only certain ℓ values receiving power. A low power was given at specified ℓ's while all other ℓ values were given zero power. The ℓ value range was given a maximum of 800, where *ell* values were given power in steps of 10. In the ideal case where the polarization maps are unmasked and is signal dominated, there will be peaks along the specified values of ℓ while the signal does not leak into other frequencies as seen in Fig. 11.

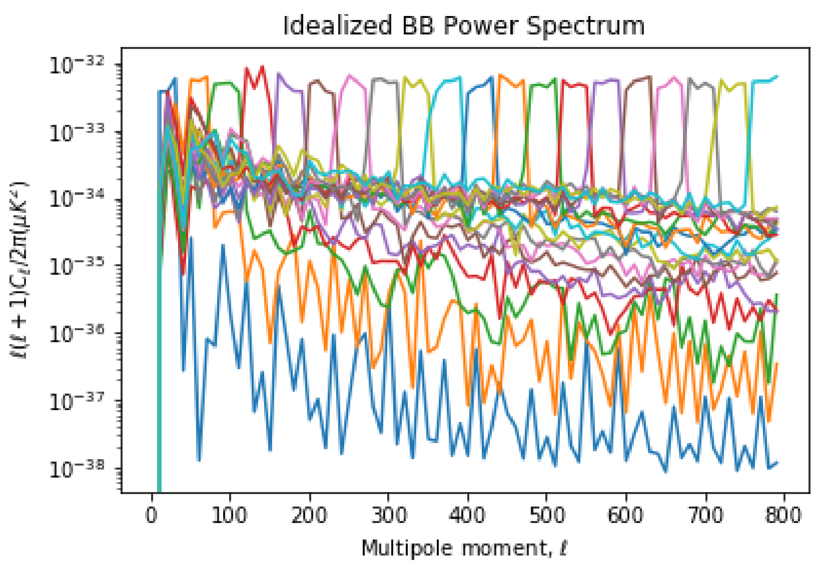

Figure 11. Example of ideal bpwf for BB angular power spectra where maps are unmasked and unfiltered. Shown to display how power leaks marginally into other other angular scales. Vertical axis displays power in terms of D_ℓ.

Comparing the model to the idealized case is not practical as the maps need to be masked. Instead, the bpwf using a vanilla estimator will be used to compare to the CNN model filter. The vanilla estimator described a smooth mask on the Q/U maps and performing the bpwf calculation on the masked Q/U maps. The power ratio of the different estimators was calculated in Fig. 12. Each bpwf calculation was done twice, once for power input from E-modes and once for power input from B-modes. Giving power to a range of ℓ values with E-mode signal resulted in EB leakage and the BB spectra of the leakage was calculated. Each curve in Fig. 12 shows a power input at a given ℓ range of values. The ratio of the calculated BB spectra, resulting from E-modes and B-modes is $D_\ell^{BB,E}/D_\ell^{BB,B}$. This also shows that at an ℓ range of 100 to 200, the CNN filter is more effective than the vanilla estimator, and neither estimator is particular effective at $\ell < 100$.

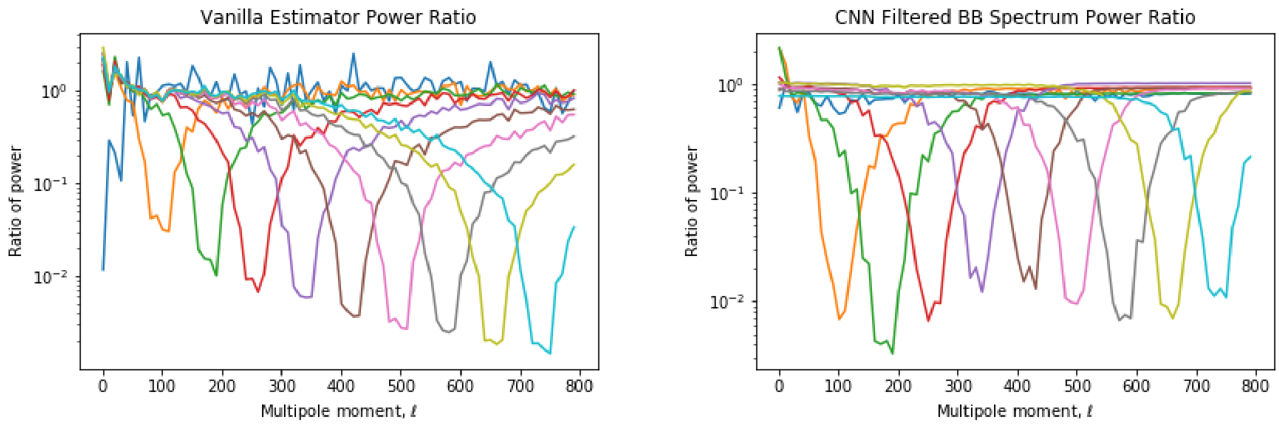

Figure 12. Ratios of BB spectra from E-mode input over B-mode input to BB spectra. Small ratios at a given ℓ shows input from B-modes results in high BB spectra and input E-modes result in low BB spectra.

5. CONCLUSIONS & FURTHER WORK

At an ℓ range of 100 to 200, the CNN estimator is more effective than the vanilla estimator, but the estimator isn't particularly effective at $\ell < 100$ as shown in Figs. 11 & 12. There is further important details that still need to be explored. A possibility would be to make maps that take up a larger portion of the sky and section them off into 20 degree by 20 degree maps. This could help in probing a larger resolution resulting in a higher signal. In the same vein, the Fourier space was defined as relatively a coarse map. Changes in the definition of Fourier space to reduce the ℓ interval in the bpwf could also be made.

Introducing different levels of noise into the polarization maps could help explore what happens at high and low noise level regimes. At low noise, the CNN filter would ideally produce better results than the vanilla estimator. This could be tested by running through the process to obtain BB spectra of the noisy maps and removing the noise bias from each spectra tested. The standard deviation of the debiased spectra could then be compared for the vanilla estimator and the CNN filter. Testing the effectiveness of the CNN filter with noise is a vital step towards making the CNN applicable to real CMB polarization maps.

REFERENCES

[1]Guzzetti, M. C., Bartolo, N., Liguori, M., & Matarrese, S. 2016, La Rivista del Nuovo Cimento, 39, 399

[2]Liu, H., Creswell, J., von Hausegger, S., & Naselsky, P. 2019, Physical Review D, 100, 023538

[3]Münchmeyer, M., & Smith, K. M. 2019, arXiv preprint arXiv:1905.05846

[4]Ronneberger, O., Fischer, P., & Brox, T. 2015, in International Conference on Medical image computing and computer-assisted intervention, Springer, 234–241

[5]Yin, L. 2019, A Summary of Neural Network Layers,

Machine Learning for Li

[6]Zaldarriaga, M. 2001, Physical Review D, 64, 103001