Inpainting the Cosmic Microwave Background with Partial Convolutional Neural Networks

Emmy Li
liemily@stanford.edu
Stanford Physics Department

(Dated: March 19, 2021)

We develop a novel application of a partial convolutional neural network (PCNN) to inpaint noisy portions of experimentally retrieved Cosmic Microwave Background (CMB) temperature and polarization maps. Due to its weak radiative signal, CMB maps can often be contaminated by extragalactic radio sources and galactic dust, leading to incomplete observations. Inpainting, or restoring missing patches of pixels in images, has seen marked progress in other fields with generative deep learning algorithms. Adapting these frameworks to cosmology, we will reconstruct sections of CMB maps cropped along the equatorial plane and overlayed with a masking mimicking thermal dust emissions. In generating predictions for this missing data, our PCNN achieves a runtime three orders of magnitude faster than Markov Chain Monte Carlo numerical simulations per retrieved patch and accurately recovers the angular power spectrum with 4% error down to angular scales of roughly 5 arc minutes. Implementing this flexible, yet accurate result drastically reduces the computational expense when compared to traditional numerical methods and shows promise with inpainting other complex, non-Gaussian data distributions beyond astrophysics.

I. INTRODUCTION

Left over from the explosive beginnings of the cosmos, the Cosmic Microwave Background (CMB) provides a critical source of information about the nature and origins of the early universe [1]. While largely uniform, minuscule fluctuations in this old, faint electromagnetic radiation captured by high-precision telescopes contribute to a robust portfolio of evidence to form the current cosmological model. Proper measurement of these asymmetries leads researchers to estimate cosmological parameters with greater accuracy, potentially answering questions about the curvature of the universe, the position of large scale structures, the ratio of baryonic to dark matter, the distribution of dark energy, and much more.

Experimentally, collecting high-fidelity data poses a unique challenge due to both the remarkable isotropy of the CMB as well as our own inauspicious vantage point from within a cloudy galaxy. Indeed, interference due to galactic dust and extragalactic radio sources muddies the observation, and removing the noisy foreground proves to be a challenging computational problem. Typically, missions record locations of contaminated portions of the sky to be removed from maps such that some estimator may reconstruct the obfuscated, masked patch. Inpainting, or restoring missing patches of pixels in images, is a burgeoning sub-topic in CMB data analysis, and in this paper, we demonstrate the success of a method of estimation using a partial convolutional neural network (PCNN) to restore the missing measurements.

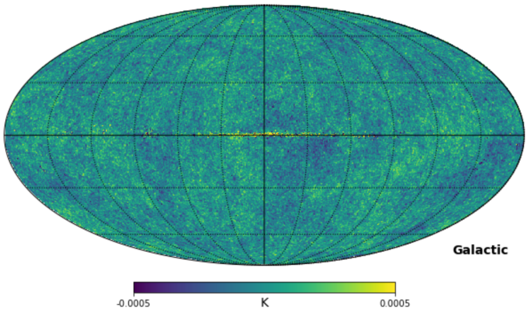

FIG. 1: Planck 2018 Commander CMB Temperature Map, shown in Galactic Coordinates on a Mollweide projection. [2]

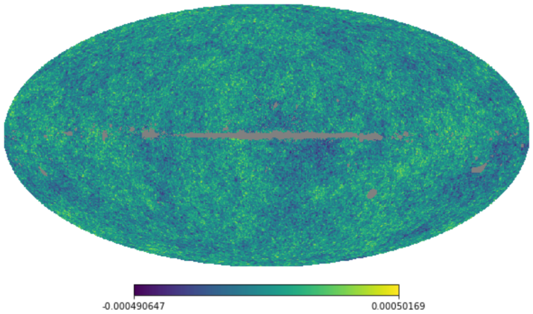

FIG. 2: Planck 2018 Commander CMB Temperature Map, with applied masking (in gray) to indicate regions of missing pixels to be restored through inpainting. Notice the most contamination exists in the equatorial region, centered on the galactic plane. [2].

Conventional approaches to CMB inpainting relied on a Gaussian constrained realization of a map generated by an angular power spectrum C_ℓ. Some groups have estimated foreground contamination with a variety of highly accurate solutions to recover low-ℓ spherical harmonic coefficients using sparsity, energy, and isotropy as priors in Markov Chain Monte Carlo (MCMC) simulations [3] [4] [5] [6] [7] [8] [9] [10]. `COMMANDER`, an MCMC based mapmaker employed by the European Space Agency's *Planck* satellite, uses Gibbs sampling to yield the density function of power spectrum C_ℓ and Bayesian inference define likelihood of C_ℓ [2] [11] [12]. Later on, we use their results as an evaluation metric for the physicality of our test outputs.

However, facing pressure from the rapidly increasing amount of data, numerical solutions became less tractable for higher resolution maps. With increasingly detailed maps, the astronomical amount of data precludes one from engaging in the computationally expensive numerical estimation process without sacrificing finer granularity. In response to the influx of high-resolution data, researchers have been applying generative deep learning neural networks to restore the missing information. These machine learning techniques do not rely on an a-priori model, but rather learn statistical features organically from training. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have been independently shown to mimic the galactic foreground and other sources of contamination by leveraging the surrounding structures of missing patches to create synthetic, life-like data [13] [14] [15] [16] [17] [18]. Another promising emerging sub-section parallel to GANs and VAEs are PCNNs which have been demonstrated to have the flexibility to reconstruct the maps and power spectra with irregular masking to a high degree of accuracy [19] [20] [21].

Building on the rapidly developing and diverse experimentation with deep learning algorithms, we implement a novel algorithm that leverages the existing architecture of PCNNs and extends them to inpainting experimental data obtained from *Planck*'s `COMMANDER` 2018 Data Release [2]. While general deep learning based inpainters utilized in computer vision and satellite imaging optimize largely on visual appeal, cosmological inpainting must reflect the underlying statistical distributions which define the data. Traditional methods used both masked (empty) pixels and the surrounding pixels to inpaint the masked region, which leads to blurriness in the recovered region. By masking the convolutional layer (i.e. partial convolutions), the network only draws knowledge from non-masked pixels.

II. METHODS

A. Generating the Dataset

In our experimentation, we take advantage of the latest data release from the European Space Agency's *Planck* satellite to access the latest CMB temperature maps [2]. There exist two fundamental challenges when analyzing cosmological phenomena in the context of deep learning, namely, the original data rests on a three-dimensional sphere and we must discretize the one map into data to use in training, validation, and test sets. In order to navigate these obstacles, we first project the spherical CMB map onto a Cartesian plane using `HEALPix` (Hierarchical Equal Area Isolatitude Pixelation) which provides a framework for processing functions on a sphere, critical to translating the statistical summaries into data visualizations [22] [23]. Using the same software, we slice the map, containing over 50 million pixels into 4,042 400×400 pixel images. Since the CMB is largely isotropic, it is reasonable to assume that the distribution on these smaller patches would be random fields, and therefore we may treat them as independently and identically distributed. Due to deep learning algorithms' heightened performance with larger data sets and the relative scarcity of viable CMB maps to sample from, I apply data augmentation techniques including randomly flipping patches horizontally and vertically and rotating by integers of 90°to artificially increase the number of training examples.

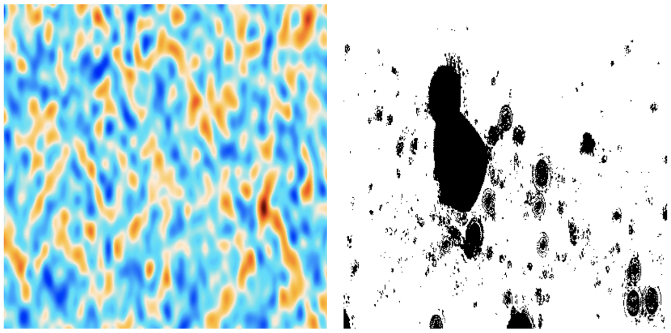

FIG. 3: An example 400×400 pixel map patch, projected from the spherical *Planck* data onto a two-dimensional cutting on the left. This was generated by the `COMMANDER` numerical estimator employed by the European Space Agency. On the right, an example 400×400 pixel mask patch corresponding to the coordinates of the left map.

After generating patches of CMB, we must slice the mask data set using Planck's 2018 Component Separation Inpainting Common mask in Intensity, downloaded from the Planck Legacy Archive [2]. This spherical mask map is matches the CMB map sliced in the previous step and covers 772 out of 4,042 patches. For the training set, we pair maps and masks randomly to augment the sampling pool, and for the validation and test sets, we divide the 772 corresponding map/mask pairs into each. A visualization of the unaltered Planck data can be viewed in FIG. 1, and the mask may be seen in FIG. 2. Additionally, a random example of a cropped patch and its corresponding mask may be viewed in FIG. 3.

B. Network Architecture

Implemented in Keras, the model follows a U-Net architecture with a partial convolutional layer and loss function [24]. The following description adhere to the description in this pivitol image inpainting paper using PCNNs [25]. For the partial convolutional layer, given filter weights $\mathbf{W}$, bias b, and a binary mask of 0s and 1s $\mathbf{M}$, the partial convolution on the current pixel values $\mathbf{X}$ can be defined as:

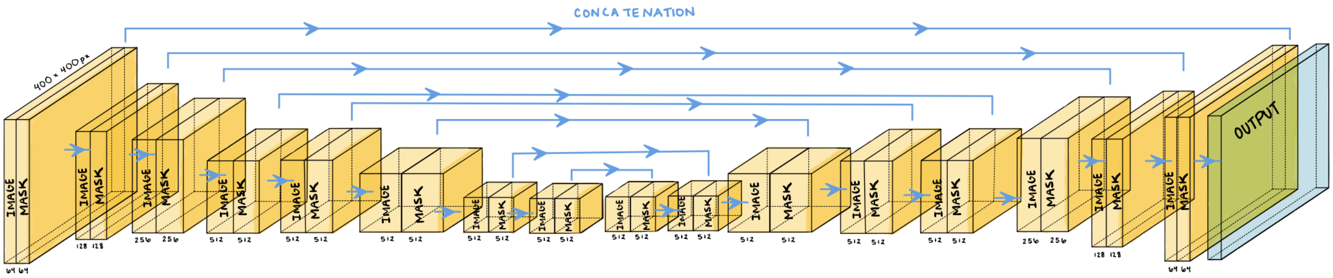

FIG. 4: Illustration of the partial convolutional neural network. The encoder and decoder, which are connected by channel-wise latent variables, have U-Net architecture, each with eight blocks. A more detailed description of each layer can be found in TABLE I in Appendix A.

$$x' = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M})\frac{sum(\mathbf{1})}{sum(\mathbf{M})} + b & \text{if sum}(\mathbf{M}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where x' denotes the updated pixel values and $\odot$ denotes element-wise multiplication. $\frac{sum(\mathbf{1})}{sum(\mathbf{M})}$ adjusts the output to accommodate the varying number of masked inputs. Then, portions of the mask are removed in areas in which the algorithm can successfully condition the output on the valid inpust such that the updated mask pixel values *m'* can be described by:

$$m' = \begin{cases} 1 & \text{if sum}(\mathbf{M}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

On a macro-level, a U-Net architecture defines the model where all convolutional layers are replaced with partial convolutional layers split between encoding and decoding methods. During encoding, the input passes through seven layers with the number of filters increasing incrementally from 32 to 128 while the filter size decreases from 7 to 3. During decoding, the data passes again through seven layers as the number of filters decreases until it meets the final concatenation with the input. The final combining step allows the model to fill in masked pixels with generated values. For more details on the network architecture, refer to TABLE I in Appendix A.

The loss function compares the inpainting accuracy at the pixel level, summarized as:

$$\begin{aligned} \mathcal{L}_{total} =& \mathcal{L}_{valid} + 6\mathcal{L}_{hole} + 0.05\mathcal{L}_{perceptual} \\ &+ 120\mathcal{L}_{style} + 0.1\mathcal{L}_{tv} \end{aligned} \tag{1}$$

where $\mathcal{L}_{valid}$ and $\mathcal{L}_{hole}$ represent the loss on the output for the non-masked and masked pixels, $\mathcal{L}_{perceptual}$ and$\mathcal{L}_{style}$ are derived from the ImageNet VGG 16 pretrained weights, and $\mathcal{L}_{tv}$ corresponds to the smoothing penalty on pixels of the masked region. For a more detailed breakdown of the loss function terms, refer to [25].

C. Training

For the training procedure, we initialize weights using the pre-trained VGG 16 weights from ImageNet (an experiment trained on millions of images and widely accepted as a good initialization for computer vision applications) [26]. After this, we use the Adam optimizer twice: once, in the first pass with batch normalization in all layers and a learning rate of $\alpha = 0.0001$, and another in the second pass with batch normalization only in the decoder with a learning rate of $\alpha = 0.00005$ [27]. This two-stage process is performed in inpainting application to avoid a mean and variance "drift" arising from the hole pixels interfering with the calculation and to achieve a faster convergence. Due to memory constraints, we used a batch size of 4 and trained 30 epochs. Of all hyperparameters tuned, the learning rate proved to be most consequential, contributing to time to convergence as well as significant bias when $\alpha > 0.001$ during either encoding or decoding.

This experiment was conducted on Google Colab using an Nvidia Tesla K80 GPU.

III. RESULTS

Shown in FIG. 5, we visually compare the ground truth CMB temperature map constructed using *Planck*'s 2018 `COMMANDER` MCMC inpainter on the left with our PCNN retrieved patches on the right. In the center, we have the original masked images which both techniques seek to reconstruct.

Quantitatively, a key way to evaluate this PCNN approach's success is to compare the theoretical angular power spectra deerived from the highly accurate but computationally expensive numerical estimators with spectra extracted from our PCNN inpainted map. As shown in FIG. 6, the angular power spectrum may be computed via HEALPix on a macro scale from restitched together complete patches [22] [23]. By calculating the power spectra and pixel distribution residuals, we aim to verify that generative algorithms like PCNN are not only

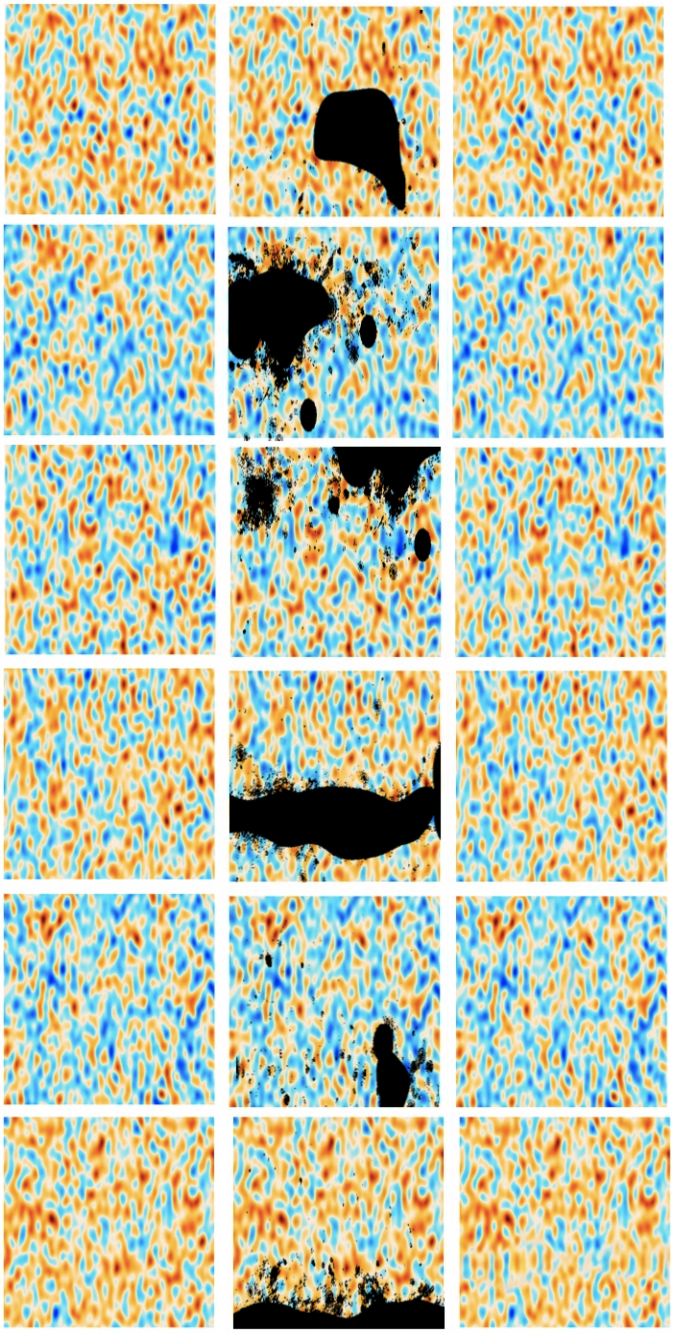

FIG. 5: Ground truth, target map patch made using *Planck*'s 2018 `COMMANDER` numerical estimator (left), the same patch overlayed with its corresponding black masking (center) and our PCNN inpainted image (right).

"hallucinating" unrealistic data, but mimicking realistic images.

The spherical harmonics, a mathematical basis orthonormal on the sphere is used to whose sum describe the variations exhibited in CMB maps. We plot the absolute value of CMB fluctuations against these spherical

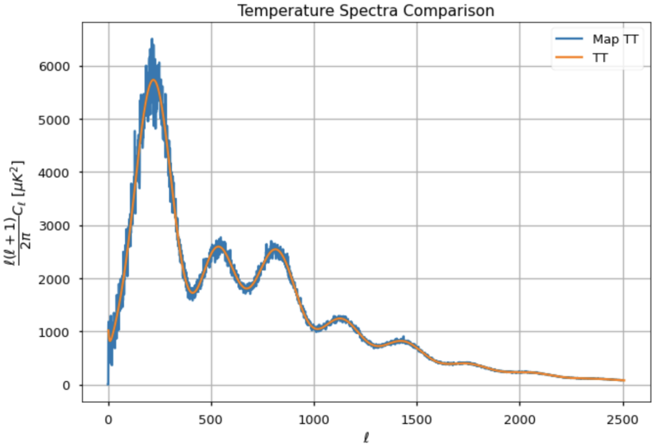

FIG. 6: Comparison of theoretical CMB temperature map angular power spectrum (orange) against PCNN extracted spectrum (blue). Theory comes from the highly accurate, but computationally expensive MCMC numerical estimators.

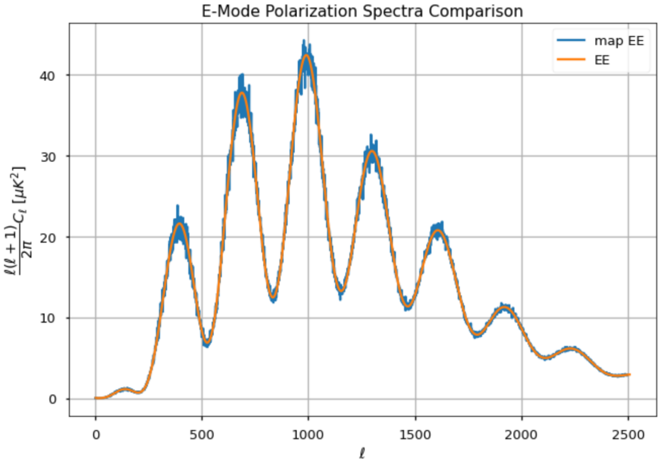

FIG. 7: Theoretical E-Mode polarization angular power spectrum (orange) against PCNN derived spectrum (blue).

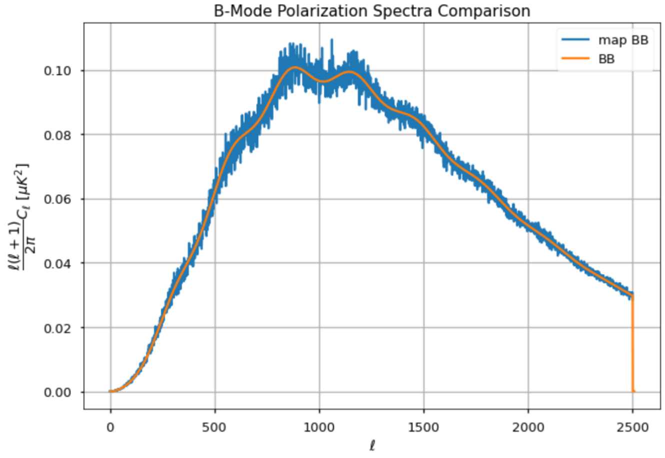

FIG. 8: Similarly, theoretical B-Mode polarization (orange) against PCNN spectrum (blue).

harmonics (the multipoles) to create an angular power spectrum useful for characterizing cosmological parameters like baryonic matter density. Each flat patch is binned into linearly spaced multipoles ($\Delta\ell = 32$ and $\ell_{max} = 2500$) with `Pixell`. In FIG. 6, a plot of comparing the theoretical and experimental temperature spectra shows that they roughly follow one another and qualitatively performs better at higher ℓ. In figures 7 and 8, we similarly see the theoretical spectra in orange and empirical in blue. While the extracted, generated spectra follows the general shape, significant noise dominates the plot, especially when comparing a far more sensitive metric like B-Mode polarization. This error could be due to an additional smoothing in the inpainted region in `COMMANDER`'s result and the inherent noise introduced when extracting a macroscopic summary statistic from restitched small patches back onto the sphere.

IV. CONCLUSION

In this paper, we presented a novel application of partial convolutional neural networks to the task of inpainting contaminated CMB maps. Using this PCNN as a baseline model, we cut the full-sky CMB maps into 4,096 400×400 pixel images consisting of mask/map pairs and apply the algorithm to restore the missing pixels. This method reconstructs the power spectrum of a CMB patch to an accuracy of $\leq 5\%$, $\leq 3\%$, and $\leq 6\%$ for temperature, E-Mode polarization, and B-Mode polarization respectively. These results show the strength of PCNNs to analyze highly non-Gaussian patterns on experimental data and displays promise for future deployment on upcoming CMB mapping missions.

In future work, hyperparameters that demand priority include the learning rate in both stages of using the Adam optimization as well as the size of each batch. Previously, special consideration was given to memory use and GPU time, and in continued experimentation, we wish to navigate this trade-off between computational expense and accuracy. Ultimately, we seek to show that deep learning algorithms provide a viable alternative to numerical estimation and separation techniques, so it is imperative to compare the runtime of one prediction generated by both means with these updates. As astronomy, astrophysics, and cosmology increasingly receive an influx of larger and larger datasets, we must place a priority on developing and applying deep learning techniques to confront this information landscape. With a continued investment in these efforts, these models may serve as a baseline for other work in computer vision including restoration, enhancement, and scaling resolution beyond cosmology.

V. ACKNOWLEDGEMENTS

I would like to thank my mentor Avoy Datta and the whole CS 230 teaching team for their valuable lessons, advice, and feedback. Guidance regarding Markov Chain Monte Carlo numerical estimations of the cosmic microwave background mentioned as an evaluation criterion is provided by Lawrence Pierson, Andrew Eberhardt, and Professor Tom Abel. Additional peer feedback of the written quality of this paper comes courtesy of Elisa Tabor, Story Frantzen, George Wang and the teaching staff of Physics 191. All contributions, unless otherwise cited in this paper, were made solely by the author: Emmy Li. Code used in this experiment can be found at https://github.com/emmyscode/deep-learning-cmb as well as a brief overview and presentation is made available at https://youtu.be/vfUTFykuzTs.

[1] A. A. Penzias and R. W. Wilson, A Measurement of Excess Antenna Temperature at 4080 Mc/s., Astrophys. J. **142**, 419 (1965).

[2] N. Aghanim, Y. Akrami, F. Arroja, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, and et al., Planck2018 results, Astronomy Astrophysics **641**, A1 (2020).

[3] J.-L. Starck, M. J. Fadili, and A. Rassat, Low-cmb analysis and inpainting, Astronomy Astrophysics **550**, A15 (2013).

[4] M. Bucher and T. Louis, Filling in cosmic microwave background map missing data using constrained gaussian realizations, Monthly Notices of the Royal Astronomical Society **424**, 1694–1713 (2012).

[5] K. T. Inoue, P. Cabella, and E. Komatsu, Harmonic inpainting of the cosmic microwave background sky: Formulation and error estimate, Phys. Rev. D **77**, 123539 (2008).

[6] J. Kim, P. Naselsky, and N. Mandolesi, Harmonic inpainting of cosmic microwave background sky by constrained gaussian realization, The Astrophysical Journal **750**, L9 (2012).

[7] J. Bobin, F. Sureau, and J.-L. Starck, Cosmic microwave background reconstruction from wmap andplanckpr2 data, Astronomy Astrophysics **591**, A50 (2016).

[8] J. Bobin, J.-L. Starck, F. Sureau, and S. Basak, Sparse component separation for accurate cosmic microwave background estimation, Astronomy Astrophysics **550**, A73 (2013).

[9] L. Perotto, J. Bobin, S. Plaszczynski, J. L. Starck, and A. Lavabre, Reconstruction of the cmb lensing for planck (2009), arXiv:0903.1308 [astro-ph.CO].

[10] S. Plaszczynski, A. Lavabre, L. Perotto, and J.-L. Starck, A hybrid approach to cosmic microwave background lensing reconstruction from all-sky intensity maps, Astronomy Astrophysics **544**, A27 (2012).

[11] H. K. Eriksen, C. Dickinson, C. R. Lawrence, C. Baccigalupi, A. J. Banday, K. M. Gorski, F. K. Hansen, P. B. Lilje, E. Pierpaoli, M. D. Seiffert, and et al., Cosmic microwave background component separation by parame-

ter estimation, The Astrophysical Journal **641**, 665–682 (2006).

[12] H. K. Eriksen, J. B. Jewell, C. Dickinson, A. J. Banday, K. M. Górski, and C. R. Lawrence, Joint bayesian component separation and cmb power spectrum estimation, The Astrophysical Journal **676**, 10–32 (2008).

[13] K. Aylor, M. Haq, L. Knox, Y. Hezaveh, and L. Perreault-Levasseur, Cleaning our own dust: simulating and separating galactic dust foregrounds with neural networks, Monthly Notices of the Royal Astronomical Society **500**, 3889 (2020), https://academic.oup.com/mnras/article-pdf/500/3/3889/34673759/staa3344.pdf.

[14] A. V. Sadr and F. Farsian, Inpainting via generative adversarial networks for cmb data analysis (2020), arXiv:2004.04177 [astro-ph.CO].

[15] N. Krachmalnicoff and G. Puglisi, Forse: a gan based algorithm for extending cmb foreground models to subdegree angular scales (2020), arXiv:2011.02221 [astro-ph.CO].

[16] A. Mishra, P. Reddy, and R. Nigam, Cmb-gan: Fast simulations of cosmic microwave background anisotropy maps using deep learning (2019), arXiv:1908.04682 [astro-ph.CO].

[17] N. Gupta and C. L. Reichardt, Mass estimation of galaxy clusters with deep learning ii: Cmb cluster lensing (2020), arXiv:2005.13985 [astro-ph.CO].

[18] K. Yi, Y. Guo, Y. Fan, J. Hamann, and Y. G. Wang, Cosmovae: Variational autoencoder for cmb image inpainting (2020), arXiv:2001.11651 [eess.IV].

[19] G. Montefalcone, M. H. Abitbol, D. Kodwani, and R. D. P. Grumitt, Inpainting cmb maps using partial convolutional neural networks (2020), arXiv:2011.01433 [astro-ph.IM].

[20] G. Puglisi and X. Bai, Inpainting galactic foreground intensity and polarization maps using convolutional neural networks, The Astrophysical Journal **905**, 143 (2020).

[21] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications, Astronomy and Computing **27**, 130–146 (2019).

[22] A. Zonca, L. Singer, D. Lenz, M. Reinecke, C. Rosset, E. Hivon, and K. Gorski, healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in python, Journal of Open Source Software **4**, 1298 (2019).

[23] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere, Astrophys. J. **622**, 759 (2005), arXiv:astro-ph/0409513.

[24] F. Chollet, keras, `https://github.com/fchollet/keras` (2015).

[25] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, Image inpainting for irregular holes using partial convolutions (2018), arXiv:1804.07723 [cs.CV].

[26] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition (2015), arXiv:1409.1556 [cs.CV].

[27] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), arXiv:1412.6980 [cs.LG].

[28] A. Mishra, P. Reddy, and R. Nigam, Baryon density extraction and isotropy analysis of cosmic microwave background using deep learning, Machine Learning: Science and Technology **1**, 045012 (2020).

[29] B. Thorne, L. Knox, and K. Prabhu, A generative model of galactic dust emission using variational inference (2021), arXiv:2101.11181 [astro-ph.CO].

[30] A. Lewis and A. Challinor, CAMB: Code for Anisotropies in the Microwave Background (2011), ascl:1102.026.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[32] W. Hu and S. Dodelson, Cosmic microwave background anisotropies, Annual Review of Astronomy and Astrophysics **40**, 171–216 (2002).

Appendix A: Network Architecture in Detail

For our network architecture drawn pictorially in FIG. 4, we give details in TABLE I about the sixteen partial convolutional layers (PConv) with corresponding filter size, number of filters (or channels), and stride. The "Batch Norm" column refers to whether that layer is followed by a Batch Normalization layer, and "Nonlinearity" specifies which activation is used following Batch Norm. Layers PConv1 - PConv8 constitute the encoder stage, and layers PConv9 - PConv16 make up the decoder stage. Concatenation steps link the previous nearest neighbor upsampled results with the corresponding PConv layer from encoding.

TABLE I: Partial Convolutional Neural Network Architecture

Layer Package Name	Filter Size	Number of Filters	Stride/Up Factor	Batch Norm	Nonlinearity
PConv1	7×7	64	2	-	ReLU
PConv2	5×5	128	2	Y	ReLU
PConv3	5×5	256	2	Y	ReLU
PConv4	3×3	512	2	Y	ReLU
PConv5	3×3	512	2	Y	ReLU
PConv6	3×3	512	2	Y	ReLU
PConv7	3×3	512	2	Y	ReLU
PConv8	3×3	512	2	Y	ReLU
NearestUpSample1		512	2	-	-
Concat1(w/ PConv7)		$512 + 512$		-	-
PConv9	3×3	512	1	Y	LeakyReLU(0.2)
NearestUpSample2		512	2	-	-
Concat2(w/ PConv6)		$512 + 512$		-	-
PConv10	3×3	512	1	Y	LeakyReLU(0.2)
NearestUpSample3		512	2	-	-
Concat3(w/ PConv5)		$512 + 512$		-	-
PConv11	3×3	512	1	Y	LeakyReLU(0.2)
NearestUpSample4		512	2	-	-
Concat4(w/ PConv4)		$512 + 512$		-	-
PConv12	3×3	512	1	Y	LeakyReLU(0.2)
NearestUpSample5		512	2	-	-
Concat5(w/ PConv3)		$512 + 256$		-	-
PConv13	3×3	256	1	Y	LeakyReLU(0.2)
NearestUpSample6		256	2	-	-
Concat6(w/ PConv2)		$256 + 128$		-	-
PConv14	3×3	128	1	Y	LeakyReLU(0.2)
NearestUpSample7		128	2	-	-
Concat7(w/ PConv1)		$128 + 64$		-	-
PConv15	3×3	64	1	Y	LeakyReLU(0.2)
NearestUpSample8		64	2	-	-
Concat8(w/ PConv6)		$64 + 3$		-	-
PConv16	3×3	3	1	Y	-