
Recovering Galaxy Cluster Convergence from Lensed CMB with Generative Adversarial Networks

Liam Parker
Princeton University
Princeton, NJ
lharker@princeton.edu

Dongwon Han
University of Cambridge
Cambridge, UK
dh689@cam.ac.uk

Pablo Lemos Portela
Mila
Montreal, Quebec
plemos91@gmail.com

Shirley Ho
CCA, Flatiron Institute
New York, NY
shirleyho@flatironinstitute.org

Abstract

We present a new method which leverages conditional Generative Adversarial Networks (cGAN) to reconstruct galaxy cluster convergence, κ , from lensed CMB temperature maps. Our model is constructed to emphasize structure and high-frequency correctness relative to the Residual U-Net approach presented by Caldeira, et. al. (2019). Ultimately, we demonstrate that while both models perform similarly in the no-noise regime (as well as after random off-centering of the cluster center), cGAN outperforms ResUNet when processing CMB maps noised with $5\mu K/arcmin$ white noise or astrophysical foregrounds (tSZ and kSZ); this out-performance is especially pronounced at high ℓ , which is exactly the regime in which the ResUNet under-performs traditional methods.

1 Introduction

The universe’s oldest light, the cosmic microwave background (CMB), encodes in its temperature and polarization a wealth of information. While the study of the CMB has a long and multi-faceted history, the study of the gravitational effects imprinted on the CMB at later stages of the universe has captured the attention of modern cosmology.

Specifically, the study of gravitational lensing of the CMB by galaxy clusters has enabled the reconstruction of the gravitational potential field, ϕ , around these clusters. For one, this field contains information about the spatial distribution of these clusters, which enables insights into the parameters that govern their formation, such as dark energy and massive neutrino properties [14]. Moreover, a variety of cosmological parameters, such as Ω_m , the mass density of the universe, can be constrained by recovering the total mass of these clusters from ϕ [5]. However, with recent and upcoming CMB surveys - e.g. AdvancedACTPol [20] and Simons Observatory [2] - expected to amass lensed CMB measurements at unprecedentedly high signal-to-noise ratios, discoveries tied to the study of CMB lensing are likely to only become more significant.

Currently, the most prevalent method for reconstructing ϕ from lensed CMB is the quadratic estimator (QE), an estimator formed from quadratic combinations of data [10]. However, QE is shown to be sub-optimal for low-noise polarization data due to the lensing itself [21], as well as for low-noise temperature data due to the cosmic variance of the background CMB gradient [8] [9], and thus not suited for the higher signal-to-noise ratios promised by this novel generation of CMB surveys. As such, a variety of alternatives have been proposed, including a gradient-inversion technique [8], a maximum-likelihood-estimator [18], and a hierarchical Bayesian inference method [16].

Machine learning (ML)-backed methods also present an attractive alternative to QE. Indeed, Caldeira, et. al. [7] use a Residual U-Net (ResUNet) to recover κ maps (the dimensionless surface-mass density along the line of sight) around galaxy clusters from input lensed CMB with higher signal-to-noise ratios than QE over a broad range of angular scales in the low-noise regime. However, the quality of ResUNet’s predictions materially degrades in noisier conditions, as well as at higher angular scales, where it underperforms QE.

One disadvantage of using the ResUNet method is its reliance on a static loss function (such as L1/L2) during network optimization. While this static loss function has been shown to capture low-frequency components, it essentially formulates the image-to-image translation problem as a per-pixel regression problem, thereby ignoring dependence between pixels in the output space and often leading to a loss of sharpness and structure [13].

In this work, we aim to overcome this disadvantage by optimizing the ResUNet with a trainable loss function (discriminator) in conjunction with an L1 loss, effectively transforming the ResUNet into a modified Pix2Pix conditional generative adversarial network (cGAN) [11]. This architecture is particularly adept at extrapolating structure and high-frequency components in image-to-image translation tasks while maintaining low-frequency correctness. We train both our cGAN and a ResUNet to recover κ maps from lensed CMB temperature maps around galaxy clusters under four conditions: no noise, astrophysical (tSZ and kSZ) foreground noise, $5\mu K/arcmin$ white noise (which mimics instrumentation noise), and random off-centering (which mimics the fact that the clusters centers will not always be perfectly centered). We demonstrate that our cGAN outperforms ResUNet under all four conditions, and that this out-performance becomes especially pronounced in the noisier regimes and at high ℓ ¹.

2 Data

We employ the Websky Extra-galactic CMB Simulations [19] to model the lensed CMB temperature anisotropies, tSZ and kSZ effects, and κ maps around corresponding galaxy clusters. These simulations are tailored to the upcoming ground-based CMB surveys such as the Simons Observatory and Advanced ACTPol [19], rendering them ideal for our purposes². We select 50,000 clusters matching the following criterion: $M_{200m} \in [10^{13}, 5 * 10^{14}]M_{\odot}$ and $z \in [0.47, 0.6]$, and cut out the lensed CMB temperature, κ , tSZ, and kSZ maps in 128×128 arcmin squares around the cluster center. Additionally, we create random $5\mu K/arcmin$ 128×128 white noise maps. Notably, all maps are projected into 2D euclidean space, as our models are only capable of handling such formats, using Orphics [15]. From these maps, we compile three feature datasets, X_{CMB} , $X_{CMB+tSZ+kSZ}$, and $X_{CMB+5\mu K}$, and use the κ maps as our target dataset. Additionally, we repeat this process for the pure CMB and κ maps with random off-centering in both RA and Dec directions by a Gaussian with mean 0 and 1 arcmin variance, resulting in one additional feature/target pair, $\{X_{oc}, \kappa_{oc}\}$. All datasets are thus size (50000, 128, 128, 1), and we split each using a 80:10:10 split.

3 Method

Our cGAN is made up of two main components: G , the generator, and D , the discriminator. Both G and D are convolutional neural networks (CNNs), which have demonstrated unparalleled power in dealing with image-data [3]; moreover, G is a Residual U-Net, a type of CNN which uses an archetypal encoder-decoder scheme that has proven adept at a wide array of image-to-image translation tasks [22]. G learns to map the observed CMB temperature map, X , to a predicted κ_{pred} map, $G : X \rightarrow \kappa_{pred}$, while D takes as input the concatenated generator-predicted κ_{pred} and ground-truth κ maps, and predicts a grid of the likelihood $[0, 1]$ that each 70×70 patch in κ_{pred} is real, $D : \{\kappa_{pred}, \kappa\} \rightarrow D_{output}$. We model G after the ResUNet proposed by Caldeira, et. al. [7], albeit with some modifications, and D after the convolutional PatchGAN architecture proposed by [11]. Ultimately, G is trained to produce κ_{pred} maps as similar as possible to the ground truth κ map, while D is trained to discriminate between the "fake" κ_{pred} and the ground-truth κ .

¹As the present paper focuses on reconstruction in the high- ℓ regime, and because of the fact that QE is computationally prohibitively expensive, we focus on comparing our cGAN architecture exclusively to ResUNet.

²The following cosmological parameters are used in the Websky simulations: $\Omega_m = 0.31$, $\Omega_b = 0.049$, $\Omega_c = 0.261$, $H_0 = 100 * 0.68$, $n_s = 0.965$, $\tau = 0.0943$ [19].

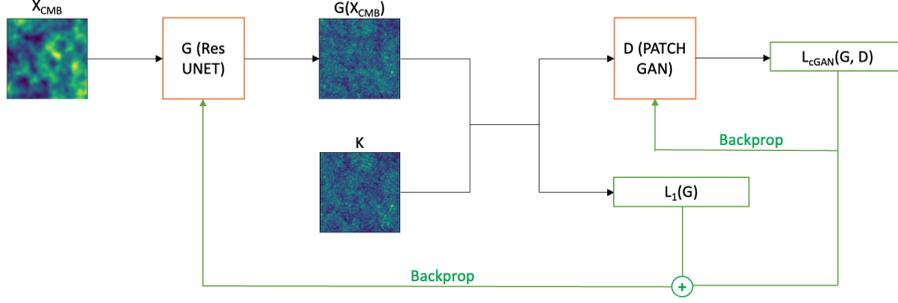


Figure 1: Structure of the cGAN. As illustrated, the generator, G , based on the ResUNet architecture takes as input X_{CMB} and produces a predicted κ map. These are fed into the discriminator, D , to produce the first loss function, $L_{cGAN}(G, D)$ used to update the parameters of both G and D . An L_1 loss is also included, as it has been shown to emphasize low-frequency correctness.

G is composed of an encoder, a bottleneck, and decoder. The encoder is made of six encoding blocks, each of which consists of a down-sampling convolutional layer followed by a simple convolutional layer. The number of filters for each block increases as $32 * 2^n$, where n is the index of the encoder block from $n \in \{0, 5\}$. The decoder follows the same structure, however the first convolutional layer of each decoder block is a 2D transpose convolutional layer, and the number of filters goes as $32 * 2^{5-n}$. Residual connections are established between encoder and decoder blocks with same n to allow low-level feature information to flow more easily through the network. The output of the decoder block is fed into a final 2D transpose convolutional layer after which \tanh activation is applied³. D is composed of four encoding convolutional layers with filter size $64 * 2^n$, followed by a simple convolutional layer with filter size 1, and a sigmoid activation function. Batch normalization is applied to all convolutional layers in G and D except the first, and LeakyReLU activation functions follow each convolutional layer except for the last. Additionally, the first two decoding blocks in G have $\text{dropout} = 0.5$ applied. The models are constructed using the Keras interface of the Tensorflow API [1], and we use the general GAN update structure laid out by Brownlee [6].

We train both our cGAN and the ResUNet three separate times on the 40,000 cluster $\{X_{CMB}, \kappa\}$, $\{X_{CMB+tSZ+kSZ}, \kappa\}$, $\{X_{CMB+5\mu K}, \kappa\}$, and $\{X_{oc}, \kappa_{oc}\}$ training datasets. Notably, cGAN is trained to minimize both the discriminator loss and an L1 loss; more specifications on optimization are provided in the Appendix. We train both models for 100 epochs using mini-batches of 32 samples on a single NVIDIA Tesla P100 GPU with 16GB of Graphic RAM, which takes 3.8 hours for the ResUNet and 8.1 hours for the cGAN.

4 Results

All results are generated over the held-out 5000-cluster test dataset. Figure 2 provides a sample visualization of ResUNet and cGAN predictions under various noise conditions for a random test cluster. From visual inspection, it appears as if cGAN captures significantly more information than ResUNet under all noise conditions; this is especially apparent in the noised (astrophysical foreground and $5\mu K/arcmin$) regimes, where cGAN continues to recover the majority of structural information, whereas significant blurring and/or loss of structure has occurred in the ResUNet predictions. In order to quantitatively test the performance of cGAN, we compare the power spectrum (generated using the Orphics package [15]) and one-point PDF of the predictions of cGAN and ResUNET to the ground-truth κ maps in Figure 3.

In observing the mean power spectra, it is clear that the ResUNet power spectrum is able to mimic that of the ground-truth relatively well in the noiseless regime, as well as after random off-centering. However, it materially diverges from the ground-truth power spectrum under both astrophysical foreground and $5\mu K/arcmin$ noise regimes; this divergence is especially pronounced at high ℓ .

³For consistency, we use this same network architecture and parameters for the ResUNet to which we compare our cGAN.

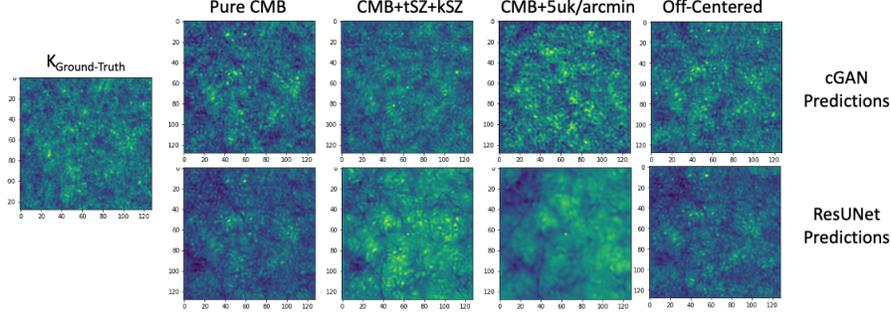


Figure 2: Visualization of sample predicted κ_{pred} maps from the cGAN and ResUNet for a random cluster in the test dataset. The predictions are made for CMB temperature maps with either no noise, astrophysical foreground noise (tSZ+kSZ), $5\mu K/arcmin$ white noise, or random off-centering according to a Gaussian with $1arcmin$ variance.

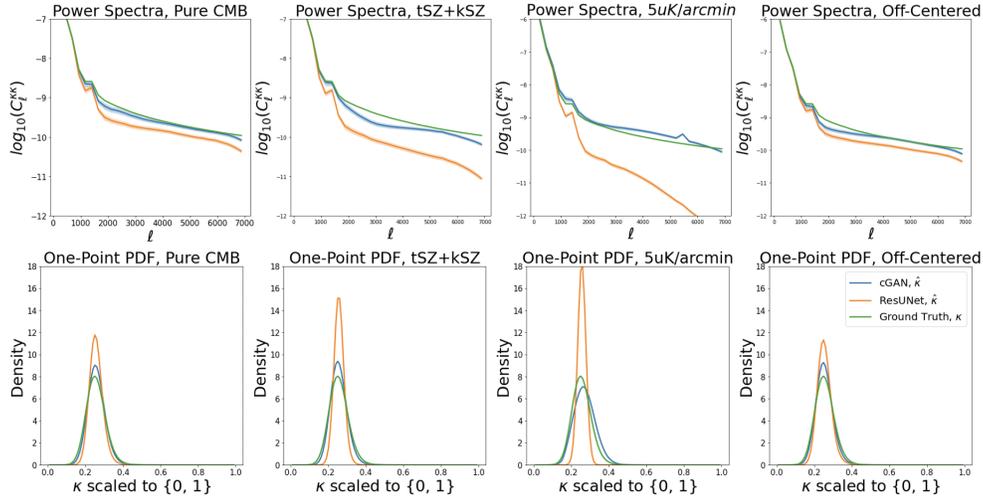


Figure 3: The mean over the 5000-cluster test dataset of the one-point PDF/power spectra of the cGAN and ResUNet predicted κ_{pred} maps under various noise conditions. For the power spectra, we also calculate the standard deviation in predicted power spectra over the test dataset using the bootstrap method, and include these standard deviations in as the shaded region in the graphs (the average standard deviation per cluster is on the order of magnitude of 10% for all predictions).

Conversely, the cGAN power spectrum is able to stay faithful to the ground-truth under all four regimes, and does not materially degrade even at high ℓ : no noticeable degradation is visible until around $\ell \approx 6500$. The relative strength of the cGAN is further emphasized in the average one-point PDFs, in which ResUNet’s one-point PDF materially diverges from the ground-truth κ map under the noised regimes while cGAN’s does not. Ultimately, cGAN’s successes at high ℓ , highlight the relative ability of the discriminator architecture at emphasizing small-scale structure in the predicted κ maps.

5 Conclusion

In the present paper, we demonstrate that the inclusion of a discriminator in the optimization of a Residual U-Net can materially improve its performance in recovering galaxy cluster convergence from lensed CMB temperature maps. Specifically, across both visualizations of predicted κ maps, as well as the power spectra and one-point PDFs of these κ maps, we show that the discriminator-enhanced network (cgAN) noticeably outperforms ResUNet under a variety of noise conditions. Moreover, we demonstrate that this out-performance becomes especially pronounced in noisy regimes (such as instrumental or astrophysical foreground noise), as well as at high ℓ . This out-performance at

high ℓ is particularly encouraging, as small-scale features are challenging to recover using traditional QE methods. In future work, it will be valuable to explore 1) cGAN's performance under a wider variety of noise conditions, 2) cGAN's relative performance with additional loss functions (such as a Fourier-space loss), and 3) alternative GAN structures (such as Wasserstein GAN [4]).

6 Broader Impact

We employ a generative adversarial network (GAN) - a popular machine learning model - in an astrophysical context. GANs are already present in myriad social applications, and while most use cases are benign, they have been maliciously employed across multiple social media platforms, from generating fake Facebook accounts to conducting personation attacks on targeted subjects [17]. The applications of such algorithms to astrophysics however has been quite limited. Nonetheless, in applying GANs in these contexts, we can better understand where and why they fail, which can help improve our ability to spot maliciously employed GANs more broadly.

Acknowledgments and Disclosure of Funding

The contributions to the paper are as follows. **Parker**; performed all neural network computation work; innovated NN architecture; performed NN diagnostic analysis; wrote up the paper and corresponding poster; initiated problem concept. **Han**; constructed data pipeline from Websky; guided ResUNet construction; designed NN diagnostic analysis; initiated problem concept. **Portela**; consulted on paper edits; helped clarify sticky physics questions. **Ho**; designed uncertainty calculations; suggested one-point PDF analysis; guided paper through the final steps before submission.

Moreover, we are extremely thankful and indebted to Professor Suzanne Staggs for the invaluable time, advice, and effort that she has graciously provided in guiding us during the completion of this fascinating project.

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.
- [2] Peter Ade, James Aguirre, Zeeshan Ahmed, Simone Aiola, Aamir Ali, David Alonso, Marcelo A Alvarez, Kam Arnold, Peter Ashton, Jason Austermann, et al. The simons observatory: science goals and forecasts. *Journal of Cosmology and Astroparticle Physics*, 2019(02):056, 2019.
- [3] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 international conference on communication and signal processing (ICCSP)*, pages 0588–0592. IEEE, 2017.
- [4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL <https://arxiv.org/abs/1701.07875>.
- [5] S Bocquet, JP Dietrich, T Schrabback, LE Bleem, M Klein, SW Allen, DE Applegate, MLN Ashby, M Bautz, M Bayliss, et al. Cluster cosmology constraints from the 2500 deg² spt-sz survey: inclusion of weak gravitational lensing data from magellan and the hubble space telescope. *The Astrophysical Journal*, 878(1):55, 2019.
- [6] Jason Brownlee. How to develop a pix2pix gan for image-to-image translation, Jan 2021. URL <https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/>.
- [7] João Caldeira, WL Kimmy Wu, Brian Nord, Camille Avestruz, Shubhendu Trivedi, and Kyle T Story. Deepcmb: Lensing reconstruction of the cosmic microwave background with deep neural networks. *Astronomy and Computing*, 28:100307, 2019.
- [8] Boryana Hadzhiyska, Blake D Sherwin, Mathew Madhavacheril, and Simone Ferraro. Improving small-scale cmb lensing reconstruction. *Physical Review D*, 100(2):023547, 2019.
- [9] Christopher M Hirata and Uroš Seljak. Analyzing weak lensing of the cosmic microwave background using the likelihood function. *Physical Review D*, 67(4):043001, 2003.

- [10] Wayne Hu and Takemi Okamoto. Mass reconstruction with cosmic microwave background polarization. *The Astrophysical Journal*, 574(2):566, 2002.
- [11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016. URL <https://arxiv.org/abs/1611.07004>.
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.
- [14] Antony Lewis and Anthony Challinor. Weak gravitational lensing of the cmb. *Physics Reports*, 429(1):1–65, 2006.
- [15] Mat Madhavacheril. Orphics. <https://github.com/msyriac/orphics>, 2022.
- [16] Marius Millea and Uros Seljak. Muse: Marginal unbiased score expansion and application to cmb lensing. *arXiv preprint arXiv:2112.09354*, 2021.
- [17] Donnie O’Sullivan. Now fake facebook accounts are using fake faces. *CNN*, 2019.
- [18] Srinivasan Raghunathan, Sanjaykumar Patil, Eric J Baxter, Federico Bianchini, Lindsey E Bleem, Thomas M Crawford, Gilbert P Holder, Alessandro Manzotti, and Christian L Reichardt. Measuring galaxy cluster masses with cmb lensing using a maximum likelihood estimator: statistical and systematic error budgets for future experiments. *Journal of Cosmology and Astroparticle Physics*, 2017(08):030, 2017.
- [19] George Stein, Marcelo A Alvarez, J Richard Bond, Alexander van Engelen, and Nicholas Battaglia. The websky extragalactic cmb simulations. *Journal of Cosmology and Astroparticle Physics*, 2020(10):012, 2020.
- [20] RJ Thornton, PAR Ade, S Aiola, FE Angile, M Amiri, JA Beall, DT Becker, HM Cho, SK Choi, P Corlies, et al. The atacama cosmology telescope: the polarization-sensitive actpol instrument. *The Astrophysical Journal Supplement Series*, 227(2):21, 2016.
- [21] Jaiyul Yoo and Matias Zaldarriaga. Improved estimation of cluster mass profiles from the cosmic microwave background. *Physical Review D*, 78(8):083002, 2008.
- [22] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? **[Yes]** See Section 2.
- Did you include the license to the code and datasets? **[No]** The code and the data are proprietary.
- Did you include the license to the code and datasets? **[N/A]**

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? **[Yes]**
 - (b) Did you describe the limitations of your work? **[Yes]**
 - (c) Did you discuss any potential negative societal impacts of your work? **[Yes]**

- (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
- 2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [N/A]
 - (b) Did you include complete proofs of all theoretical results? [N/A]
- 3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code will be available upon acceptance.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] With additional training details available in the Appendix.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
- 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes] All open-source code and datasets have been cited in the body of the paper.
 - (b) Did you mention the license of the assets? [N/A]
 - (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
 - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
- 5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Network Optimization

The cGAN network is trained by optimizing the GAN loss function from Isola, et. al. [11], given by

$$L_{cGAN}(G, D) = \mathbb{E}_{\kappa}[\log D(\kappa)] + \mathbb{E}_{X_{CMB}}[\log(1 - D(G(X_{CMB})))] \quad (1)$$

Because this loss function is driven by D , and D is constructed to emphasize localized structure recovery and high-frequency correctness, we include an L_1 loss as well in order to promote low-frequency correctness in the output space,

$$L_1(G) = \mathbb{E}_{X_{CMB}, \kappa}[\|\kappa - G(X_{CMB})\|]_{L_1} \quad (2)$$

Thus, the total network loss is the weighted sum of L_{cGAN} and L_1 , where we weight L_1 by a factor of 100 relative to L_{cGAN} as suggested in Isola, et. al. [11]. This gives a composite objective function

$$G_{cGAN}^* = \underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{max}} L_{cGAN}(G, D) + 100L_1(G), \quad (3)$$

which is minimized relative to D (as D should maintain a maximum distance between real and fake in every iteration) and maximized relative to G .

During cGAN training, we alternate between one update step on D and one update step on G . Specifically, we (1) generate a batch of fake samples, $G(X_{CMB}) \rightarrow \kappa_{pred}$, corresponding to a real batch of samples κ , (2) update the discriminator using the batch of real κ , (3) update the discriminator using the batch of generated κ_{pred} , (4) update the generator using the composite loss. Importantly,

we divide the objective by 2 while optimizing D , slowing down its learning rate relative to G , as otherwise D has been found to outpace G [11]. Our training is run on the Adam optimizer [12], with a learning rate of 2×10^{-4} and $\beta_1 = 0.5, \beta_2 = 0.999$. Additionally, for the ResUNet, we use a learning rate scheduler as well, which reduces the learning rate by a half every time the training loss has not improved for five consecutive epochs.