Inpainting RFI in Visibility Waterfall Plots with a U-Net

Jing Liu and Adrian Liu
McGill University Department of Physics & McGill Space Institute

Siamak Ravanbakhsh
McGill University School of Computer Science & Mila

PHYS 489 Final Report
Lilian Childress
(Dated: April 21, 2021)

The Hydrogen Epoch of Reionization Array (HERA) measures 21-cm emissions from neutral hydrogen during Cosmic Dawn. HERA data can be contaminated with radio frequency interference (RFI), which must be filled in before the data can be further used. Inspired by machine learning algorithms that inpaint missing data in other cosmological applications [1, 2], we propose convolutional neural networks (CNNs), including a U-Net, to inpaint HERA visibility waterfall plots. We found good results on simulated data with simulated RFI masks, but require further work to achieve comparable results on real data with simulated RFI masks.

I. INTRODUCTION

The Hydrogen Epoch of Reionization Array (HERA; `http://reionization.org`) is a radio interferometer located in South Africa. HERA is part of an international effort to map the universe during the Epoch of Reionization (EOR) using 21-cm cosmology techniques [3, 4]. During this period and the Dark Ages that preceded it, generally known as Cosmic Dawn, neutral hydrogen emitted photons as its spin states underwent changes—known as 21-cm emission [5]. This radiation can map out far more of the observable universe and allow us to better understand aspects of the early universe such as the properties of the first galaxies and the evolution of large-scale structure [3].

Data from HERA is returned as visibility waterfall plots of local sidereal time (LST) vs. frequency channel. Visibility is a complex quantity that is essentially a Fourier transform of the brightness of the sky [6]. (A more in-depth treatment of HERA and radio astronomy can be found in Section II B.) The end goal is to recover power spectra—distributions of power in discrete Fourier frequency components—of spatial variations in the 21cm signal. In practice, various factors may contaminate the raw data; these must be accounted for before the data can be sent further down the pipeline.

One pervasive contaminant in HERA waterfall plots is radio frequency interference (RFI)—transmissions within the observed frequency band from non-celestial sources such as FM radio channels or car engine ignitions [7]. Currently, machine learning models such as [8] are able to delineate RFI areas directly in waterfall plots, allowing us to mask over them. Afterwards, the missing data must be filled in; otherwise, sharp boundaries in the waterfall plots will ruin Fourier transforms by creating additional artificial high-k power in the final power spectra.

There are two algorithms that can be used to inpaint, or fill in the missing data, detailed in Section II B. The goal of this project is to provide an alternative inpainting algorithm that leverages the capabilities and theoretical advantages of machine learning (introduced in Section II C), with the hope that it will surpass the two current algorithms. Machine learning inpainting algorithms are already deployed in other areas of cosmology, such as cosmic microwave background analysis [1, 2], but to our knowledge, this is the first application to RFI. Our main contribution is a machine learning algorithm that inpaints well on simulated visibility waterfall plots with simulated masks, though a thorough comparison to existing inpainting algorithms is left to future work. However, once good performance is achieved on real data with simulated masks, the model can be applied to inpaint real masks in real data—the end goal of this project.

II. BACKGROUND

A. Cosmic Dawn

About 370,000 years after the Big Bang, as the universe expanded and cooled, electrons and protons that had been part of an ionized fluid came together to form the first neutral atoms in a process called recombination [9]. During recombination, photons were emitted; these decoupled photons, redshifted into radio waves over billions of years, form the cosmic microwave background (CMB) that permeates all space today [10].

After recombination came the period known as the Dark Ages, which lasted until 1 billion years after the Big Bang. The universe had cooled sufficiently for light to travel long distances, but there were only two sources of photons: the aforementioned CMB, and 21-cm line emission from neutral hydrogen due to changes in its spin states [5, 11]. A few hundred million years later, the first stars, galaxies, and quasars formed, marking the end of the Dark Ages [11]. They emitted radiation and

ionized the hydrogen [11]. This process of reionization completed with the emergence of large galaxies, and the universe became the way we see it today [11].

While the period between the CMB's release and the completion of reionization, generally referred to as Cosmic Dawn, is currently lacking observations, astronomers are probing 21-cm emission using tools like HERA in an attempt to create a map of the sky during this era [11].

B. HERA and Radio Interferometry

HERA is a type of radio telescope known as an interferometer [4]. Because radio waves are longer than visible light waves, for a given diameter, a radio telescope will have a lower resolution than an optical telescope [6]. Interferometry provides a solution: two antennas a certain distance (called a baseline) apart are pointed at the same source. The antennas measure visibility, a complex quantity that quantifies the strength of a Fourier mode of the sky [7]. These signals are combined in a way that mimics the operation of a telescope dish with a diameter equivalent to the baseline, creating a higher resolution signal [12].

HERA has 350 14-m parabolic dishes measuring data from $50-250$ MHz, corresponding to a redshift of $z\sim 5-25$, where the higher the redshift, the farther back in time the signal was emitted. Different combinations of these dishes create different baselines: each baseline measures a certain Fourier mode, and by combining different baselines (as well as variations in individual baselines, as the Earth rotates) we can map out the sky at all frequencies [3, 6, 12].

More precisely, visibility is defined as a function of baseline and frequency:

$$V(\boldsymbol{b},\nu)\propto\int d^2\theta T(\boldsymbol{\theta},\nu)e^{-i2\pi\boldsymbol{b}\cdot\boldsymbol{\theta}/\lambda}, \tag{1}$$

where $T(\boldsymbol{\theta},\nu)$ is the brightness of the sky as a function of flat-sky coordinates $\boldsymbol{\theta}\equiv(\theta_x,\theta_y)$ and frequency ν. The baseline $\boldsymbol{b}$ probes the Fourier mode given by $\boldsymbol{u}=\boldsymbol{b}/\lambda$, where $\boldsymbol{u}$ are spatial frequencies dual to angular position $\boldsymbol{\theta}$ [13].

HERA returns visibilities as waterfall visibility plots, an example of which is shown in Figure 1. These show the evolution of the amplitude and phase of the visibility function of a single baseline as a function of local sidereal time (LST) and frequency. As the Earth rotates, a single baseline traces out an arc in the complex uv-plane, sampling different uv-modes. By taking combinations of baselines and frequencies, we are able to map out a fuller picture of the visibility function.

Raw voltages from the antennas, combined in a digital system called a correlator, are outputted as visibilities and subsequently fed through a data analysis pipeline to eventually return power spectra [14]. First, quality checking is performed to find antenna defects and calibrate data from the same baselines against one another. Next, algorithms such as [8] flag RFI areas, which are then masked. The following step, the topic of this project, is to inpaint: to fill in missing data values masked by RFI. Two algorithms currently serve this purpose. First, a variant of the CLEAN algorithm [15], currently deployed in the HERA pipeline, deconvolves data with the assumption that the original image consists of point sources in Fourier space (i.e. sinusoids in the original frequency space) [16]. Another proposed algorithm is a technique that draws Gaussian-distributed samples in a way that respects priors on the data's covariance. For both of these algorithms, error quantification—which has not been done in prior work—is difficult, requiring repeated simulations.

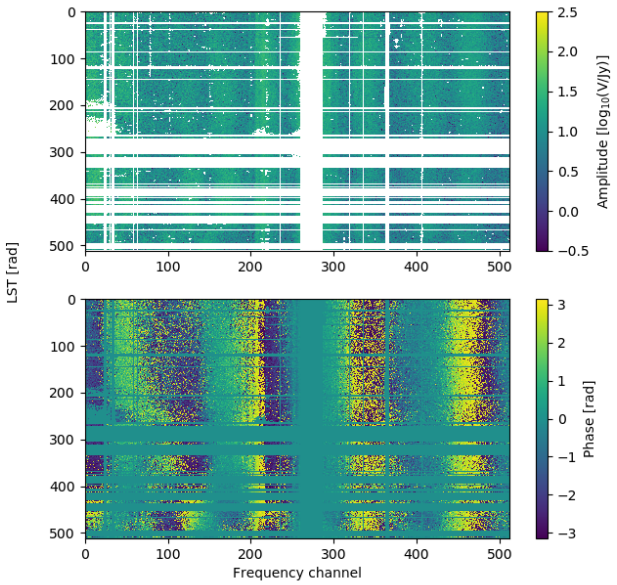

FIG. 1. An example waterfall plot from HERA with RFI masked. The top plot shows the amplitude of the visibility while the bottom plot shows the phase.

After inpainting and correcting some systematic effects, the data are averaged by time and baseline to produce a dataset for power spectrum estimation [14].

C. Machine Learning

Machine learning refers to a class of algorithms that automatically discern patterns in data, as opposed to being provided sequential instructions [17]. They have applications in a vast number of areas including image classification, natural language processing, and product recommendations.

1. Neural Networks

Neural networks are a powerful type of machine learning algorithm. Neurons, the fundamental computing units, receive inputs x_i weighted by weights w_i, add a bias b, then send the result through a nonlinear activation function ϕ, which determines if the output y is kept or discarded. This process can be summarized as $y = \phi(\sum_i x_i w_i + b)$. When layers of neurons are interconnected in a stack, the result is a neural network. Due to their layered architecture (see Figure 2), neural networks can hierarchically extract features in inputs, leading to excellent performance on problems such as regression or classification [17].

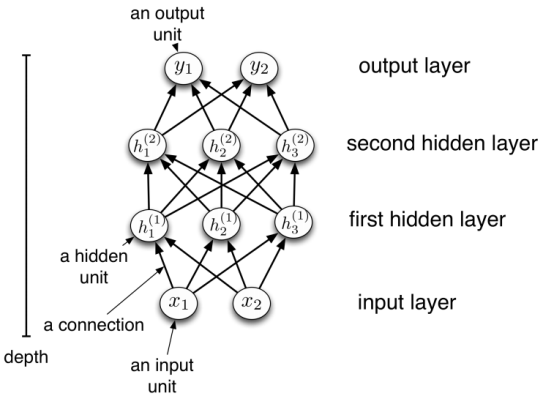

FIG. 2. An example neural network [17].

In this project, we use a neural network to learn the features of visibility waterfall plots in order to inpaint in the masked areas. Given two versions of the waterfall plots—masked and unmasked—as inputs, the network treats the unmasked versions as ground truths and adjusts its weights in order to produce image outputs that best match these ground truths. This setup, where the network is provided with ground truths and attempts to fit data to them, is called supervised learning, and this process of learning through repeated iterations through a dataset is called training. We can define what the network considers a best fit by defining a loss function, which the network will seek to minimize through training. In a forward pass, an input propagates through the network and produces an output. This output is compared to the ground truth using the loss function and an error is calculated. The error is propagated back through the network and weights are adjusted through gradient descent, an iterative algorithm that moves the weights in the direction of greatest decrease for the loss function. (Mathematically, each step can be written as $\boldsymbol{w_{i+1}} = \boldsymbol{w_i} - \alpha \frac{\partial L}{\partial \boldsymbol{w_i}}$, where $\boldsymbol{w_i}$ is the weight vector at step i and α is a predefined learning rate.) The goal is to minimize the loss function using its gradient with respect to the weights and biases, and stepping in the direction that decreases the loss [17].

The mean squared error (MSE) between the predicted values and the ground truth values is a common loss function, but in our problem we restrict the MSE to be taken over masked areas since we are only interested in inpainting in those areas:

$$L(\hat{Y}, Y) = \frac{1}{N} \sum_{\substack{\text{masked} \\ \text{areas}}} |Y - \hat{Y}|^2, \tag{2}$$

where N is the number of data points, $\hat{Y}$ are the predictions, and Y are the ground truths.

In supervised learning, the goal is to eventually generalize the network's training to unseen data. In our case, the goal is to train the network to inpaint simulated RFI masks, where the data underneath is provided as ground truths, such that it might be successfully applied to real RFI masks for which there is no correct data. As such, we want a network that generalizes well to the overall problem and without overfitting to its given training examples. Dropout, explained in Section III, is one method of countering overfitting.

2. Convolutional Neural Networks

The layers discussed so far are called fully-connected, because every neuron in an intermediate layer l_j is connected to every neuron in layers l_{j-1} and l_{j+1}. With image inputs, where each pixel is served by a neuron, the computational cost quickly adds up. Convolutional layers are an alternative that require less computational complexity. Used in conjunction with pooling layers (defined below), they form a convolutional neural network (CNN), which have two benefits: they are able to consider shared structures across the network, in that knowledge about a certain feature in one part of the input can be reused in another part. This property, combined with invariance, the ability to recognize a transformed (translated, rotated, resized, etc.) input as a version of the original, makes CNNs the current gold standard for applications involving image data [17].

Mathematically, convolution can be thought of as a sum of multiplication of parts of a kernel w over parts of an input array x. The 2D convolution function is given by

$$Y_{a,b} = \sum_{i=0}^{K} \sum_{j=0}^{K} (X_{a+i,b+j})(W_{i,j}),\ 0 < a, b < (D-K), \tag{3}$$

where X is $K \times K$, W is $D \times D$, and the output Y is $(D-K) \times (D-K)$ [18]. An example is described in Figure 3. Similarly to fully-connected layers, the output of a convolution—called a feature map—is usually passed through a nonlinear function in order to provide greater expressive power.

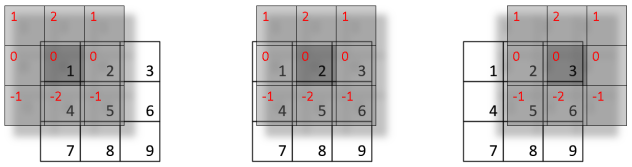

FIG. 3. First three steps of a 2D convolution example, where X is the bottom matrix [19]. The output of the entire operation is a 3×3 matrix Y where $Y_{1,1} = 0\cdot1+0\cdot2+0\cdot1+0\cdot0+1\cdot0+2\cdot0+0\cdot(-1)+4\cdot(-2)+5\cdot(-1) = -13, Y_{2,1} = 0\cdot1+0\cdot2+0\cdot1+1\cdot0+2\cdot0+3\cdot0+4\cdot(-1)+5\cdot(-2)+6\cdot(-1) = -20, Y_{3,1} = -17$, etc.

Often used in conjunction with convolutional layers are pooling layers, which downsample, or reduce the size of an input by separating it into sections and aggregating the values in each section. Pooling contributes to invariance by creating a summarized, lower-resolution representation of the important features in the input. All in all, convolutional layers detect spatial features and pooling layers contribute local invariance—providing a powerful tool for learning features in image data [20].

3. U-Nets

U-Nets are a type of neural network architecture consisting of a symmetrical pipeline in which images are first repeatedly downsampled through convolutional and pooling layers before being repeatedly upsampled. The downsampling phase learns features and their locations, which can be thought of as an abstract representation of an image. Then, upsampling reconstructs the original image using that representation.

Originally introduced for biomedical image segmentation in [21], U-Nets are also effective for other types of image problems—such as inpainting. Their name comes from the shape of their visualization, as seen in Figure 4.

III. PROCEDURE

We began by generating simulated visibility waterfall plots using the `hera_sim` software package [22]. We also created a framework for generating simulated, randomized RFI masks by extracting the widths and heights of RFI bars in a real mask and shuffling them. Simulated masks serve as an easier prediction problem, as we only generate rectangular RFI bars in contrast to the irregularities found in real data. We then created training datasets by combining simulated waterfall plots with simulated masks. We ran different neural network architectures on these datasets and compared performance, using the same loss function (Equation 2) in all cases. Finally, we also applied the best-performing networks to real HERA data with simulated masks.

Below are details about the layers in our final architecture, a U-Net. The exact architecture is detailed in Figure 4, with the implementation in TensorFlow Keras based off of [23].

Conv2D: Standard convolutional layers, as described in Section II C 2. Some of the convolutional layers are strided, which means certain pixels are sampled while others are skipped. Strided convolutions act as pooling layers [18].

Dropout: Layers where some neuron connections are temporarily discarded at random. Dropout counters overfitting by preventing the network from over-relying on a specific set of neurons [24].

Conv2DTranspose: Transposed convolution is an upsampling operation that is the reverse of convolution, used in the reconstruction phase of the U-Net [18].

Concatenate: Concatenation of two inputs to produce a larger output. In the U-Net, concatenation serves as a form of long skipped connections, whereby output from one layer can bypass multiple layers. Practically, outputs from feature maps in the downsampling phase are concatenated to corresponding feature maps in the upsampling phase, which helps the network recover spatial information lost during downsampling [25].

Dense: Equivalently called fully-connected layers, described in Section II C 1. Several of these are usually found at the end of CNN architectures and they are also used as output layers.

IV. RESULTS AND DISCUSSION

Using simulated waterfall plots, the first model we tried was a 13-layer CNN, which was unable to fully remove RFI masks. Trained on 500 1500×818 examples, its ending MSE was ~ 500. Our second architecture was a deeper (i.e. having more layers) CNN based on AlexNet, a famous architecture tuned for image classification [27]. It returned a slightly lower MSE of ~ 123. After switching architectures and cropping the data to be 512×512, we found that a U-Net performed far better, reaching an ending MSE of ~ 70 (results shown in Figure 5). The results look good visually; the network is able to make roughly correct predictions.

We subsequently applied the U-Net to 544 512×512 samples of real HERA data, with results shown in Figure 6. Visually, the results are much worse than for the simulated data. The MSE is not reported because there are evidently significant differences between the statistical distributions underlying the simulated and real data; as a result, the MSE for the real data cannot be directly compared to the MSE for the simulated data.

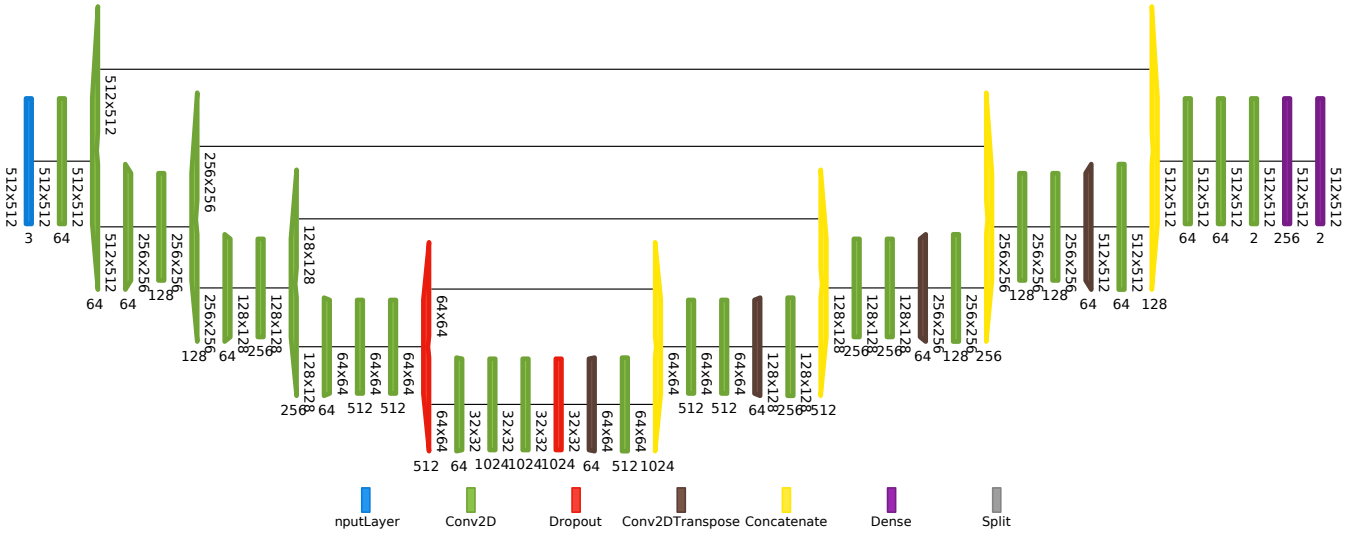

FIG. 4. Our best U-Net architecture. The vertical numbers are the dimensions of the input in each layer, while the horizontal numbers represent the number of kernels used in each layer (and thus the dimension of the layer's output). The blue rectangle on the far left represents the input image, which is $512 \times 512 \times 3$ (3 channels, consisting of real values, imaginary values, and a binary RFI mask), while the purple rectangle on the far right is the output image, which is $512 \times 512 \times 2$ (2 channels, one each for real and imaginary values). Image created with [26].

Qualitatively, what's missing between the predicted images and the ground truths are bits of noise that were masked in the original (e.g. around frequency channel 100 in Figure 5, the predicted output is much smoother than the true output). This observation makes sense as we're taking the MSE, which doesn't give us local information—it only seeks to minimize the total loss value across the entire masked area, which may not result in optimal accuracy in all parts of the domain. In other words, the information necessary to correctly predict the noise is simply unavailable, in which case the predicted values in the noisy region will simply be what the model thinks is the mean value of the pixels in that region. More complex architectures or loss functions with spatial consideration would probably be required to improve prediction performance.

V. CONCLUSION

In conclusion, the U-Net architecture worked well on simulated visibility waterfall plots with simulated RFI masks and surpassed AlexNet and simpler CNN's. However, its performance on real HERA data with simulated masks is not yet comparable; this is the next step to be completed before inpainting real data with real masks.

Further work could incorporate a "noise generator" that would account for the missing noise in the output images. Though the resultant noise can be fine-tuned with an appropriate loss function, there is no guarantee it will be able to produce better images since the underlying noise distribution is presumably random. Additionally, there are currently measurable differences in inpainting performance between applications to thick and thin RFI bars; it would be interesting to quantify whether the U-Net exhibits the same tendencies. Alternative loss functions better suited to feature localization could also be explored.

Finally, to provide a complete picture of the performance of our U-Net algorithm, we require a quantitative comparison to CLEAN and the Gaussian method. Since power spectra are HERA's end products, this comparison could involve running repeated simulations of all models and comparing errors in the generated power spectra.

VI. ACKNOWLEDGEMENTS

My heartfelt thanks to Adrian Liu and Siamak Ravanbakhsh for their supervision, guidance, and life advice. Also, I found conversations with Samuel Gagnon-Hartman, Saurabh Singh, Robert Pascua, Sabrina Berger, and Lilian Childress invaluable.

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

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

[3] D. R. DeBoer, A. R. Parsons, J. E. Aguirre, P. Alexander, Z. S. Ali, A. P. Beardsley, G. Bernardi, J. D. Bowman, R. F. Bradley, C. L. Carilli, *et al.*, Hydrogen epoch of reionization array (HERA), Publications of the Astronomical Society of the Pacific **129**, 045001 (2017).

[4] HERA Hydrogen Epoch of Reionisation Array radio telescope.

[5] HI line.

[6] Information@eso.org, Interferometry.

[7] CASA Radio Analysis Workshop – Caltech, 19-20 January 2012.

[8] J. Kerrigan, P. L. Plante, S. Kohn, J. C. Pober, J. Aguirre, Z. Abdurashidova, P. Alexander, Z. S. Ali, Y. Balfour, A. P. Beardsley, *et al.*, Optimizing sparse RFI prediction using deep learning, Monthly Notices of the Royal Astronomical Society **488**, 2605 (2019).

[9] M. Haynes, Recombination era.

[10] WMAP Big Bang CMB test.

[11] A. Loeb, The dark ages of the universe, Scientific American **295**, 46 (2006).

[12] MIT Haystack UREI - Undergraduate Research Educational Initiative.

[13] A. Liu and J. R. Shaw, Data analysis for precision 21 cm cosmology, Publications of the Astronomical Society of the Pacific **132**, 062001 (2020).

[14] J. E. Aguirre, Validation of the hera phase i epoch of reionization 21 cm power spectrum software pipeline (2021).

[15] J. Högbom, Aperture synthesis with a non-regular distribution of interferometer baselines, Astronomy and Astrophysics Supplement Series **15**, 417 (1974).

[16] A. R. Parsons and D. C. Backer, Calibration of low-frequency, wide-field radio interferometers using delay/delay-rate filtering, The Astronomical Journal **138**, 219–226 (2009).

[17] R. Grosse, CSC 321 intro to neural networks.

[18] S. Ravanbakhsh, Applied machine learning, winter 2020 (comp551-002).

[19] S. H. Ahn, Example of 2d convolution.

[20] J. Brownlee, A gentle introduction to pooling layers for convolutional neural networks (2019).

[21] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation (2015), arXiv:1505.04597 [cs.CV].

[22] `https://github.com/HERA-Team/hera_sim`.

[23] Zhixuhao, zhixuhao/unet.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research **15**, 1929 (2014).

[25] N. Adaloglou, Intuitive explanation of skip connections in deep learning (2020).

[26] A. Bauerle, C. Van Onzenoodt, and T. Ropinski, Net2vis - a visual grammar for automatically generating publication-ready cnn architecture visualizations, IEEE Transactions on Visualization and Computer Graphics , 1–1 (2021).

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, Imagenet classification with deep convolutional neural networks, Neural Information Processing Systems **25** (2012).

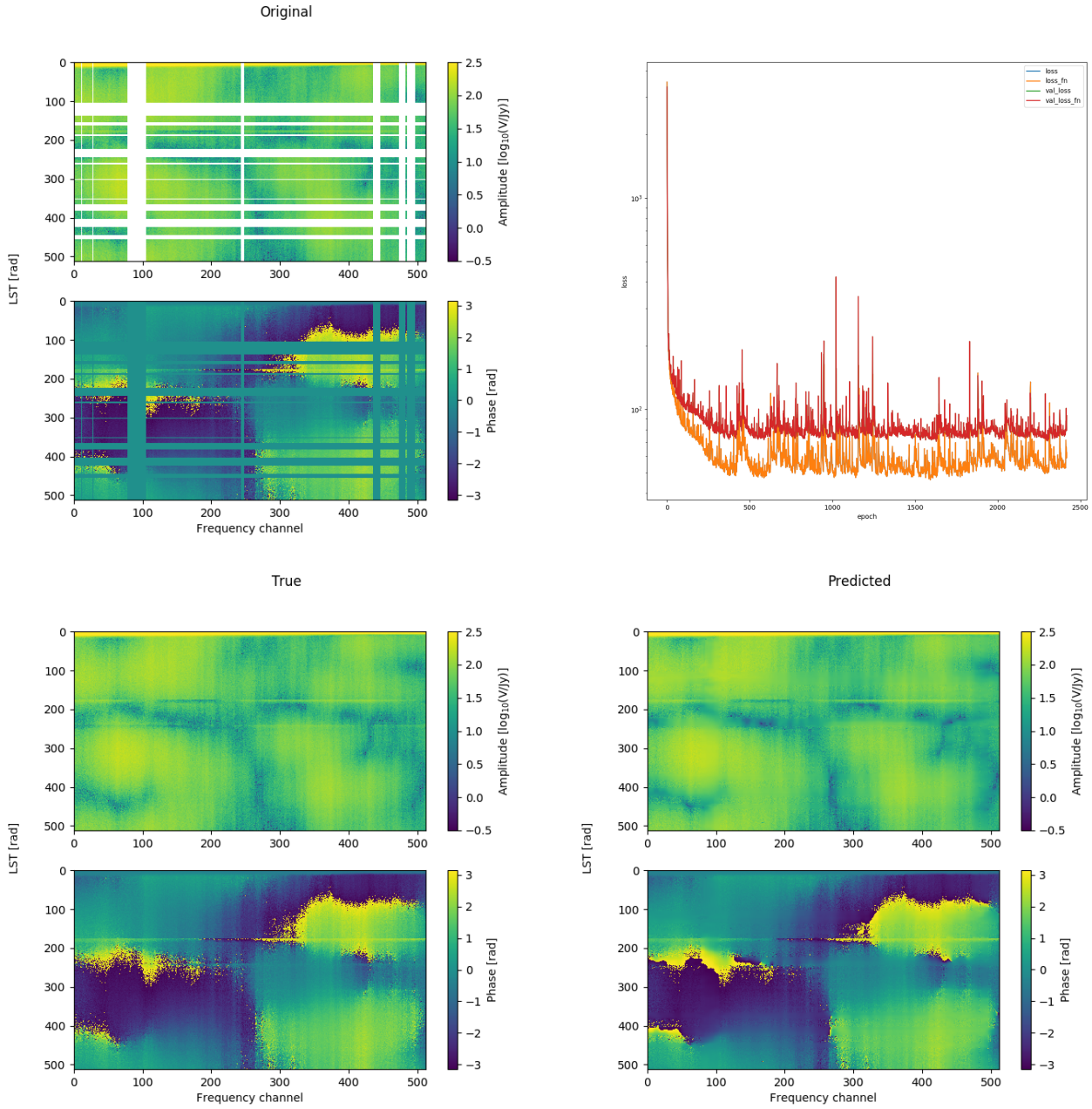

FIG. 5. The best U-Net results on simulated data so far, resulting in a MSE of around 70. The U-Net was trained on 500 different 512×512 waterfall plots. In this case, "Original" refers to one particular visibility plot with a simulated mask, passed into the network as a training example."True" refers to the same simulated visibility plot but without any mask, i.e. the ground truth. "Predicted" is the network's predicted output for this particular input image. The plot in the top right shows the training curves—the loss over time. This is just a sanity check to ensure the network is trained properly, as the loss should decrease over time.

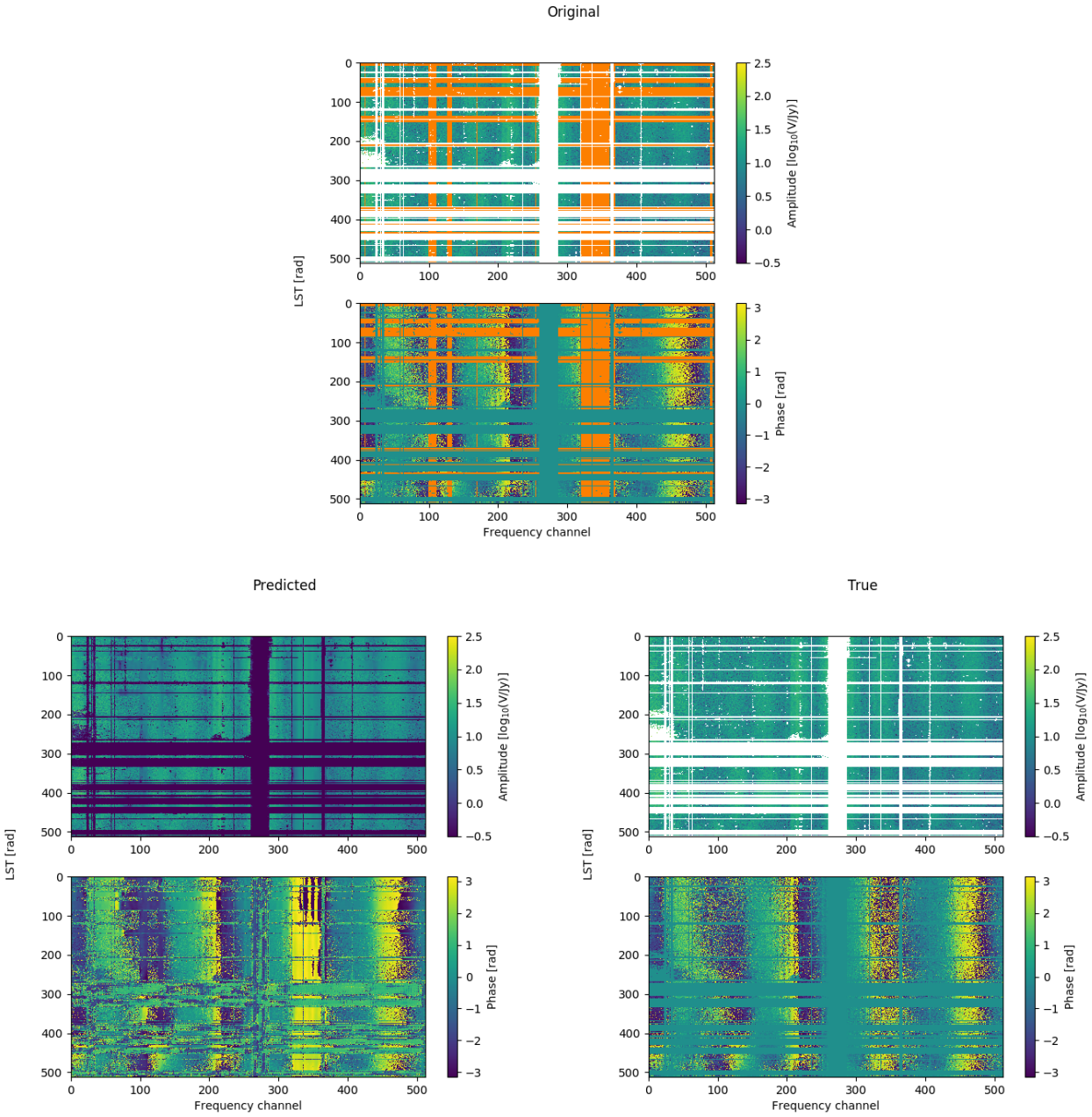

FIG. 6. The best U-Net results on real data so far. In the uppermost plot, orange represents a simulated RFI mask applied over areas with existing data, while white is the data's real mask. Using 544 512×512 waterfall plots, cut from a series of real images spanning one night of observation with 42 baselines, 4 polarizations, and 4354 time samples, the U-Net was trained to inpaint the orange areas. This figure is the clearest visualization of the problem as a whole: to summarize, the intention is to achieve good performance inpainting the orange simulated masks, for which we have ground truths, and to apply the trained network to the white real masks, for which we do not.