Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **488,** 2605–2615 (2019)
Advance Access publication 2019 July 8

doi:10.1093/mnras/stz1865

Optimizing sparse RFI prediction using deep learning

Joshua Kerrigan,[1]★ Paul La Plante,[2] Saul Kohn,[2] Jonathan C. Pober,[1] James Aguirre,[2] Zara Abdurashidova,[3] Paul Alexander,[4] Zaki S. Ali,[3] Yanga Balfour,[5] Adam P. Beardsley,[6] Gianni Bernardi,[5,7,8] Judd D. Bowman,[6] Richard F. Bradley,[9] Jacob Burba,[1] Chris L. Carilli,[4,10] Carina Cheng,[3] David R. DeBoer,[3] Matt Dexter,[3] Eloy de Lera Acedo,[4] Joshua S. Dillon,[3] Julia Estrada,[11] Aaron Ewall-Wice,[12] Nicolas Fagnoni,[4] Randall Fritz,[5] Steve R. Furlanetto,[13] Brian Glendenning,[10] Bradley Greig,[14,15] Jasper Grobbelaar,[5] Deepthi Gorthi,[3] Ziyaad Halday,[5] Bryna J. Hazelton,[16,17] Jack Hickish,[3] Daniel C. Jacobs,[6] Austin Julius,[5] Nicholas S. Kern,[3] Piyanat Kittiwisit,[6] Matthew Kolopanis,[6] Adam Lanman,[1] Telalo Lekalake,[5] Adrian Liu,[18] David MacMahon,[3] Lourence Malan,[5] Cresshim Malgas,[5] Matthys Maree,[5] Zachary E. Martinot,[2] Eunice Matsetela,[5] Andrei Mesinger,[19] Mathakane Molewa,[5] Miguel F. Morales,[16] Tshegofalang Mosiane,[5] Abraham R. Neben,[12] Aaron R. Parsons,[3] Nipanjana Patra,[3] Samantha Pieterse,[5] Nima Razavi-Ghods,[4] Jon Ringuette,[16] James Robnett,[10] Kathryn Rosie,[5] Peter Sims,[1] Craig Smith,[5] Angelo Syce,[5] Nithyanandan Thyagarajan,[6,10]† Peter K. G. Williams[20] and Haoxuan Zheng[11]

Affiliations are listed at the end of the paper

Accepted 2019 June 24. Received 2019 May 29; in original form 2019 February 21

ABSTRACT

Radio frequency interference (RFI) is an ever-present limiting factor among radio telescopes even in the most remote observing locations. When looking to retain the maximum amount of sensitivity and reduce contamination for Epoch of Reionization studies, the identification and removal of RFI is especially important. In addition to improved RFI identification, we must also take into account computational efficiency of the RFI-Identification algorithm as radio interferometer arrays such as the Hydrogen Epoch of Reionization Array (HERA) grow larger in number of receivers. To address this, we present a deep fully convolutional neural network (DFCN) that is comprehensive in its use of interferometric data, where both amplitude and phase information are used jointly for identifying RFI. We train the network using simulated HERA visibilities containing mock RFI, yielding a known 'ground truth' data set for evaluating the accuracy of various RFI algorithms. Evaluation of the DFCN model is performed on observations from the 67 dish build-out, HERA-67, and achieves a data throughput of 1.6×10^5 HERA time-ordered 1024 channelled visibilities per hour per GPU. We determine that relative to an amplitude only network including visibility phase adds important adjacent time–frequency context which increases discrimination between RFI and non-RFI. The inclusion of phase when predicting achieves a recall of 0.81, precision of 0.58, and F_2 score of 0.75 as applied to our HERA-67 observations.

Key words: methods: data analysis – techniques: interferometric.

★ E-mail: joshua_kerrigan@brown.edu
† Jansky Postdoctoral Fellow.

1 INTRODUCTION

Next generation radio interferometers are now beginning to become operational. These arrays are looking to detect and measure some of the weakest signals the Universe has to offer, such as the brightness-temperature contrast of the 21cm signal during the Epoch of Reionization (EoR). By measuring this highly redshifted signal we can characterize the progression of the EoR. The understanding gained from this characterization has the potential to help us unravel how the first stars and galaxies formed and reionized their surrounding neutral hydrogen. While instruments like the Hydrogen Epoch of Reionization Array (HERA, DeBoer et al. 2017) have the intrinsic sensitivity required to detect the EoR signal through a power spectrum, they are afflicted with anthropogenic noise which we refer to as radio frequency interference (RFI). Interference from RFI in 21cm EoR observations is an especially significant obstacle because it can have a brightness anywhere from on the order of the EoR signal to orders of magnitude beyond even Galactic and extragalactic foregrounds. RFI unfortunately introduces a reduction in sensitivity in two separate but distinct ways, one being the direct contamination by having similar spectral characteristics and overpowering of the 21cm signal, and the other being the introduction of a complex sampling function due to missing data. This produces correlations between modes when computing the Fourier transform along the frequency axis (Offringa, Mertens & Koopmans 2019). It is therefore important to strike a balance between identifying RFI while not falsely identifying non-RFI as RFI, which leads to further complicating our sampling function over frequency. Many approaches have recently been developed to identify and extract RFI from radio telescope data. RFI algorithms of particular interest include AOflagger (Offringa, van de Gronde & Roerdink 2012), which uses a Scale-invariant Rank operator to identify morphologies that are scale-invariant in time or frequency which is a characteristic of many RFI signals. This RFI detection strategy has been used successfully on instruments such as the Murchison Widefield Array (MWA) (Offringa et al. 2015) and the Low-Frequency Array (Offringa et al. 2013). Alternative approaches to RFI identification include the application of neural networks. More specifically, a deep fully convolutional neural network (DFCN) based on the U-Net architecture (Ronneberger, Fischer & Brox 2015) has been used on single-dish radio telescope data (Akeret et al. 2017), and a recurrent neural network (RNN) has been applied to signal amplitudes from radio interferometer data (Burd et al. 2018).

In this paper, we expand upon the RFI identification approach using a DFCN developed in TENSORFLOW (Abadi et al. 2016). DFCNs have seen a growing trend of use (Long, Shelhamer & Darrell 2014; Ronneberger, Fischer & Brox 2015; Akeret et al. 2017; Fu et al. 2017; Chen et al. 2018; Guha Roy et al. 2018) for semantic segmentation in images and provide a good starting point for our use in identifying RFI. For our RFI DFCN application, we use both the amplitude and phase information from an interferometric visibility. This technique is prompted by examples such as what is shown in Fig. 1, which demonstrates how the phase of time-ordered visibilities (waterfall visibilities) can provide supplemental information in identifying RFI beyond that of an amplitude-only approach. Note that in this paper, all time-ordered visibility plots of real data are in the yellow–purple palette (e.g. Fig. 1) whereas all simulated data are in the blue–white palette (e.g. Fig. 8). To understand the improvements afforded by our joint amplitude–phase network, we compare it to both an amplitude only network and the Watershed RFI algorithm (See Appendix A) which is the current RFI-flagging algorithm of choice for the HERA data processing pipeline.

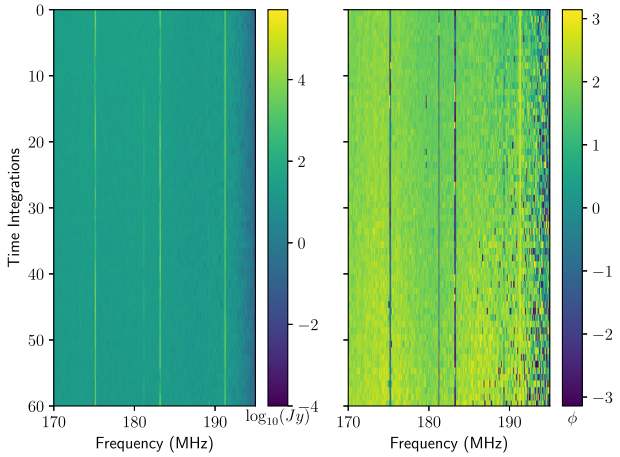

Figure 1. An example of a HERA 14 m baseline waterfall visibility between 170 and 195 MHz in amplitude (left) and phase (right). The phase waterfall visibility demonstrates how it can provide complementary information about the presence of RFI such as in the 181.3 MHz channel which has constant narrow-band RFI and the more spontaneous 'blips' in the 179.5 MHz channel at time integrations of 13, 22, and 23. The significant contrast between the phase of the sky fringe, and how it is restricted to a narrow band is an obvious indication of being RFI.

The paper is outlined as follows. Section 2 introduces the architecture of our network, discusses how it compares to previous work, and describes the training data set. We then demonstrate the effectiveness by evaluating both DFCNs on simulated and real HERA observations in Section 3. Finally, in Section 4, we conclude with discussion of further applications and future work.

2 METHOD

2.1 DFCN architecture

The standard 2D convolutional neural network (CNN, LeCun & Bengio 1998) is structurally similar to that of a typical artificial neural network (ANN, Lecun et al. 1998), but it differs to an ANN's dense layers of 'neurons' by its successive convolutions of an input image, which preserves spatial dependence. Each convolutional layer contains a set of learn-able filters which represent a response for particular shapes at different scales (e.g. the edges of an object in an image). The convolved output for every layer is then typically downsampled using a process known as max pooling that strides a window across the image keeping the highest pixel value within the window. Max pooling provides both a computational improvement due to a decreased image size, and an added level of abstraction relative to the initial image. After the convolution and max pooling layers, the image typically is then passed through a non-linear activation function (e.g. sigmoid function) which produces a spatial activation map describing the convolutional layer's response to every pixel contained within the image. The eventual output of these successive convolution, max pooling, and activation layers is then used to predict (or regress) based on the classification of the image. The error between the predicted class and the true class is then computed through a loss function such as the cross-entropy loss (or mean-squared error for regression) and the error is back-propagated through the network updating the learn-able parameters.

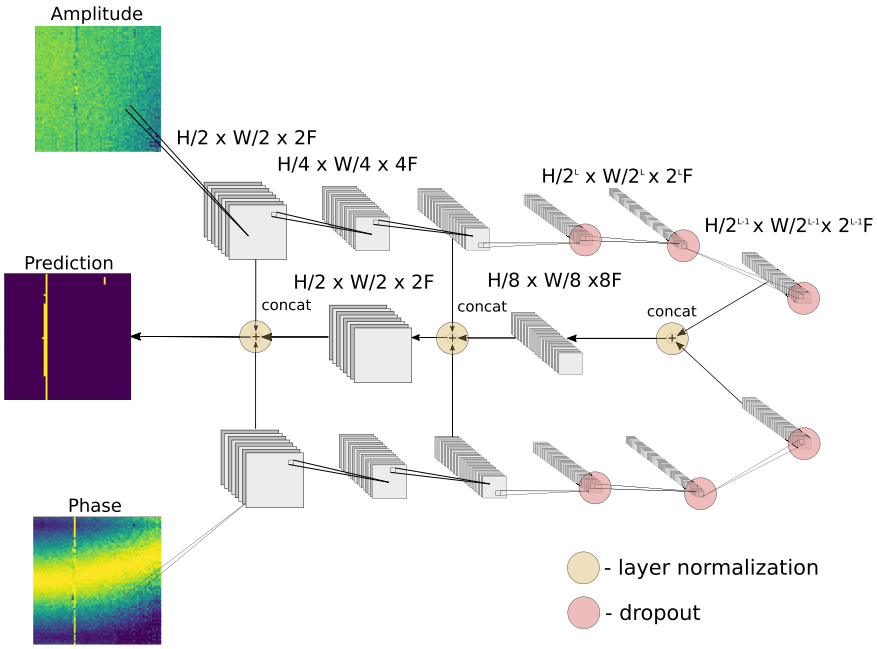

Figure 2. The general architecture for the amplitude–phase DFCN demonstrating the sliced in frequency, padded in both time and frequency, and finally normalized amplitude and phase input layers. H and W correspond to the input visibility dimensions in time and frequency, while F is the number of feature (filter) layers with L corresponding to the total number of layers between input and the fully convolutional layer. For reasons explained in Section 2.2, we use layer normalization at each skip connection and concatenation due to the difference in distributions of the amplitude and phase downsampling pathways. Every convolutional layer in the downsampling pathway is a three times stacked set of convolutional layers with 3×3 kernels leading into an output convolutional layer with a 1×1 kernel, similar to the 'Network in Network' architecture of Lin et al. (2013). All convolutional layers including the final output layer use the Leaky ReLU as their activation function.

The style of network we describe in this paper deviates from a traditional CNN by being developed as a multi-input/output fully CNN, where the number of inputs is equal to outputs. For a deeper understanding of this kind of network architecture, see Krizhevsky, Sutskever & Hinton (2017). We begin with a deep fully convolutional network architecture similar to the U-Net RFI (Ronneberger et al. 2015; Akeret et al. 2017) implementation. However,instead of using a uniform number of feature layers per convolutional layer we increase the feature layers as you get deeper into the network based on the minimum number of feature layers needed for the initial convolutional layer. This approach offers a decrease in computational time as the number of convolutions is reduced for the largest image inputs. Some caution should be taken here as a reduced number of feature layers for the input convolutional layers could cause prediction performance to suffer as the neural network may be incapable of capturing important structure in the input images due to underfitting. This initial layer feature size is another hyperparameter that needs to be tuned where the fewest number of layers required to model our data is optimal. All convolutional layers in our neural network consist of several stacked layers which due to their respective max pooling stage gives the output dimensions of $(\frac{H}{2^L} \times \frac{W}{2^L} \times 2^L F)$ where F is the number of feature layers, H and W are the layer height and width in pixels, and L is the layer of interest. Every output for each stacked layer contains a 1x1 convolutional layer with a Leaky ReLU activation function which is analogous to the 'Network in Network' introduced in Lin, Chen & Yan (2013). By introducing 1x1 convolutions with non-linearities to each output we yield a per-pixel discrimination independent of neighbouring pixels which increases the network's ability to model higher level abstractions. This is important because abstractions of our data are typically invariant to initial changes (Bengio, Courville & Vincent 2012) thus having the potential for more robust predictive power.

To adapt the network to use the visibility phase component, we mirror the amplitude only network as shown in Fig. 2. We then combine successive amplitude and phase convolution layers at each transpose convolution layer with the technique known as 'skip connections' introduced in Long et al. (2014) and He et al. (2016). This is implemented by taking the output of a downsampled convolutional layer and concatenating it with an upsampled transpose convolutional layer of equal time, frequency, and feature dimensions. By using these skips in the convolutional pathway, the network is provided with a 'template' from which to make small deviations. More simply speaking, the skip connections re-introduce higher resolution information that may have become lost due to the successive convolution and pooling operations that further remove the network's ability to relate higher order abstraction to our raw input images. This fixes an issue within deep networks where fits to non-linearities become dominant in a layer, leading to training and overfitting issues. Empirically, we find that using skip connections in conjunction with phase information allows for training a deeper network that converges in fewer iterations than the simple amplitude-only network. This effect is most likely due to the fact that while our amplitude and phase visibilities are broadly dissimilar the one aspect where they can agree is where RFI exists. Layers deeper into our network rely on more abstract features and by stacking these skip connections from amplitude and phase pathways, activations due to RFI present in pixels strengthens inferences at later layers.

For each of the skip layer concatenations between the amplitude and phase pathways, we subtract the mean and normalize over both time and frequency, which assists in standardization as amplitude

Table 1. Architecture overview of the DFCNs demonstrated in this analysis. The coloured rows correspond to the concatenations on the outputs between those respective layers, where prior to the concatenation each layer undergoes a batch normalization. The depth of a layer here means that there are multiples of the layer stacked all having the same properties. The amplitude–phase DFCN has two input pathways mirrored up until the first transpose convolution layer.

layer type	kernel size	stride	filters	depth
convolution	3x3	1	16	2
convolution	1x1	1	16	1
maxpool	2x2	2		1
batch norm.				
convolution	3x3	1	32	2
convolution	1x1	1	32	1
maxpool	2x2	2		1
batch norm.				
convolution	3x3	1	64	2
convolution	1x1	1	64	1
maxpool	2x2	2		1
batch norm.				
convolution	3x3	1	128	2
convolution	1x1	1	128	1
maxpool	2x2	2		1
batch norm.				
convolution	3x3	2	256	2
convolution	1x1	1	256	1
maxpool	2x2	2		1
batch norm.				
transpose conv.	3x3	2	128	1
transpose conv.	3x3	2	64	1
transpose conv.	3x3	2	32	1
transpose conv.	5x5	4	2	1

and phase features can be quite disparate. The amplitude only DFCN we use has $\sim 6 \times 10^5$ trainable parameters, while the addition of the phase downsampling layers for the amplitude and phase DFCN pushes the number of trainable parameters to $\sim 9 \times 10^5$. The specific per layer attributes employed in our networks can be seen in Table 1, where it should be noted that per layer dimension sizes are not specified because this style of network is agnostic to the input height and width. Additionally, all convolutional layers are zero padded so that input height and width dimensions are equal to the output which helps with reducing the minimum allowable dynamic input image size.

To optimize the network hyperparameters, a coarse grid search was performed over activation function, dropout rate, learning rate, and batch size; the optimal results from this search are found in Table 2. The depths of our convolutional layers are chosen to maximize learning and minimize prediction times, while trying to retain abstractions of the input visibilities that can properly describe our RFI. These dimensions are thus determined by initially training at an arbitrarily high number of feature layers and scaling back to the minimum number of layers we need to retain for convergence of the training loss.

2.2 Data preparation

The analysis in this paper is performed entirely on HERA data (both simulated and real) with PYUVDATA (Hazelton et al. 2017) used for data handling and therefore should be noted that any data preparation techniques outlined here may be unique to HERA. This does not imply that they are unsuitable for other radio interferometers but additional precautions may need to be taken into consideration. To prepare the amplitude–phase input space to be as robust to as many visibility scenarios as possible, we must adopt several standardizations. The amplitude of the visibility can vary drastically by local sidereal time (LST), day, and baseline type while having significant differences in dynamic range. In contrast, the phase of a visibility is intrinsically more standardized: it is constrained between $-\pi \leq \phi \leq \pi$ and should have a mean that is approximately $\mu_\phi = 0$, so we should only expect substantial deviations across baseline type, which are due to changing fringe rates. Therefore to lessen the dynamic range issues in amplitude, we standardize our waterfall visibilities $V(t, \nu)$, according to $\hat{V}(t, \nu) = (\ln|V| - \mu_{\ln|V|})/\sigma_{\ln|V|}$, by subtracting the mean, $\mu_{\ln|V|}$, and dividing by the standard deviation, $\sigma_{\ln|V|}$, across time and frequency of the logarithmic visibilities.

Table 2. Parameters and network architecture features that were determined by grid-search cross-validation which included the activation function but not the loss function. The dropout rate is uniform across all nodes as highlighted in Fig. 2.

Parameter	Values
Batch size	256
Optimizer	ADAM[a]
Learning rate	0.003
Activation function	Leaky Rectified Linear Unit[b]
Dropout rate	0.7
Loss function	Cross entropy

Notes: [a]Kingma & Ba (2014).
[b]Maas, Hannun & Ng (2013).

To further increase the robustness and generalizability of our network for different time and frequency sub-bands, we slice the HERA visibilities into 16 spectral windows of dimensions 64 frequency channels by 60 time integrations (6.3 MHz × 600 s). We then pad both time and frequency dimensions by reflecting about the boundaries, extending the data set in both directions. This allows for making predictions for the edge pixels, which otherwise would be ignored due to the size of our convolution layer kernel size of 3 × 3 (98.44 kHz × 30 s). Furthermore, we want to use square input channels to maintain a 1:1 aspect of time to frequency pixels. These considerations inform the decision to use a size of 68 × 68 for our input waterfall visibility.

Combined with our training batch size, N, of 256, for our amplitude–phase DFCN, this gives us an input training space of size $N \times H \times W \times C = (256 \times 68 \times 68 \times 2)$, where C is the number of input channels (e.g. C_0, $C_1 = \hat{V}(t, \nu)$, $\phi(t, \nu)$).

2.3 Training data set

The training data set was composed of simulated HERA visibilities using the simulator, hera_sim. .[1]

This simulator creates visibilities according to a 'pseudo-sky', which means that modelled point sources have no relationship, in either time or frequency, to any real extragalactic source on the sky (e.g. Fornax A). Extragalactic point sources are modelled using the discrete form of the visibility equation

$$\tilde{V}(t, \nu) = \sum_n \left[\tilde{A}(\tau, \hat{s}) * \tilde{S}_n(\tau) * \delta(\tau_n - \tau) \right] \quad (1)$$

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

(Parsons et al. 2012) which depends upon the source delay position on the sky, τ, the source spectrum, $S_{\rm n}(\nu)$, and the delay-dependent interferometer gains, $\tilde{A}(\tau, \hat{s})$, where a tilde represents the Fourier transform converting between frequency ν and delay τ, and $*$ represents a convolution. Point source flux densities are drawn from a power-law distribution of the form $\Pr(S > S_{0.3\ \rm Jy}) = (\frac{S}{S_{0.3\ \rm Jy}})^{-1.5}$ with a lower bound of 0.3 Jy. The spectral indices for these sources are then assigned uniformly at random between $-1 \leq \alpha_{\rm r} \leq -\frac{1}{2}$ as per Hurley-Walker et al. (2017), where $S_\nu \propto (\frac{\nu}{\nu_{\rm centre}})^{\alpha_{\rm r}}$. The source delays (sky positions) are also chosen according to a uniform random distribution. Each simulated waterfall visibility contains between $10^3 \leq N_{\rm srcs} \leq 10^4$ sources with the aforementioned characteristics. We simulate diffuse galactic emissions with the de Oliveira-Costa et al. (2008) Global Sky Model (GSM) and an analytic form of the HERA primary beam (Parsons 2015). GSM diffuse emissions are not precisely modelled but created to give a sky-like realization by sampling across LST and frequency, and applying a filter in time that has a fringe-rate corresponding to the baseline type being simulated. The visibility baseline types are uniformly sampled across LST, where baseline length, $|\boldsymbol{b}|$, is chosen according to a half-normal distribution with $\mu_{|\boldsymbol{b}|} = 7.5\ \lambda$ and $\sigma_{|\boldsymbol{b}|} = 150\ \lambda$. This is done to closely resemble the distribution of baseline lengths seen in HERA which is weighted towards short baselines. The learned model can be further tuned as longer baseline types are introduced.

We model RFI with four distinct classes: narrow-band persistent (e.g. ORBCOMM), narrow-band burst (e.g. ground/air communications), broad-band burst (e.g. lightning), and random single time–frequency 'blips'. Narrow-band persistent constitutes the majority of RFI and are most often the brightest sources in HERA observations; these are empirically modelled. Narrow-band bursts have no preference in duration or frequency but typically persist $>$ 30 s and are simulated with a Gaussian profile in time to mimic the roll on/off seen in HERA observations. Broad-band bursts are rare events that exist across the entire HERA band at specific time integrations. These events are introduced in only 3 per cent of the training data. We randomly inject 'blips' that are RFI with a duration of $\Delta t \leq 10$ s and frequency width, $\Delta\nu \leq 100$ kHz, which when taking into account HERA's time and frequency resolution places this class of RFI into single visibility pixels.

To create the most comprehensive HERA visibility simulations to mimic real observations we include simplistic models of several important effects seen in the HERA signal chain. These effects include:

Cross-talk – an effect due to over-the-air coupling between nearby HERA receivers and dipole-arm coupling. This spurious correlation is mocked by convolving the simulated visibility with white noise.

HERA bandpass – empirically derived from HERA bandpass measurements and fit to a seventh-order polynomial (Parsons & Beardsley 2017).

Gain fluctuations – fluctuations are applied to the analytic HERA bandpass by introducing individual phase delays with a uniform spread between $-20 \leq \delta\tau \leq 20$ ns.

We simulate a training data set of 1000 HERA observations of 10 min (60 time integrations) over the frequency range of 100–200 MHz (1024 frequency channels). The mean RFI occupancy rate for these simulated observations was $\sim$ 10 per cent. This value differs from the $\sim$3 per cent which is the typically observed RFI environment in the Karoo Desert, South Africa seen in past HERA observations (Kohn 2016) which used a simple statistical thresholding RFI algorithm. The comparison between our simulated and more recent real HERA data set RFI occupancy rates across the band can be seen in Fig. 3. We further expand this training data set by performing data augmentation techniques on the reduced spectral windows. These techniques include mirroring over time and frequency, Gaussian random noise injection (correlated between amplitude and phase) with an amplitude that is at most 10 per cent of $|V|$ the visibility amplitude and by translating a spectral window across the band creating unique window samples at varying frequency intervals. Using a translation in frequency has the intent of reducing overfitting to steady-state narrow-band RFI (e.g. ORBCOMM) because of repetitive sub-band samples entering the training data set.

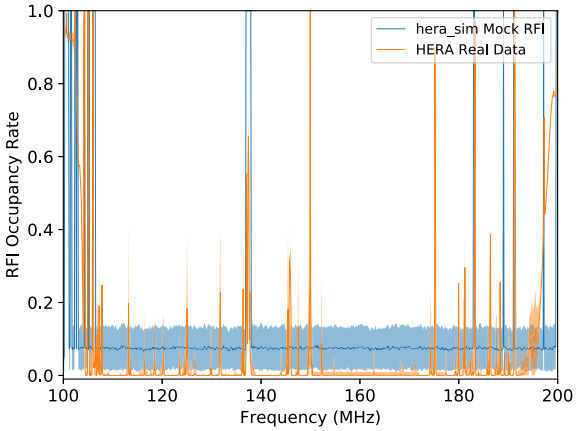

Figure 3. The hera_sim mock RFI (blue) occupancy rates across the band with its variance (blue region), as compared to RFI flagged in our real HERA data evaluation data set (orange) with its own variance (orange region). The simulated RFI is overemphasized ($>$10 per cent) in the training data set. This is done in an attempt to balance the training due to RFI being a significantly sparse class which without would lead to more significance placed on non-RFI when computing the loss.

After increasing our simulated data set volume through augmentation it is sliced into 16 spectral windows and padded according to Section 2.2 which results in 44 800 unique spectral window visibilities each of size 68 time samples $\times$ 68 frequency channels. We separate this simulated data set by an 80–20 split, where 80 per cent of the simulated data set is used for training and 20 per cent is used for validation.

3 EVALUATION

For the evaluation of our networks, we used several data sets unseen in training. The real observed data set used for evaluation consisted of HERA observing data from the 2017–2018 season, more specifically between the Julian Dates of 2458098–2458116, which we will just refer to as our real HERA data set . The real HERA data were composed of raw uncalibrated visibilities that have been visually inspected and manually flagged by hand with high- and low-frequency band edges removed. Hand flagging was accomplished by looking in both amplitude and phase for sharp discontinuities and structure that exhibited an increase in power when compared to a fringing sky signal. The band edge removal is a precaution due to the large dynamic range roll-off, which makes discriminating between RFI and sky observations nearly impossible for humans and algorithms alike. This reduces our actual data

evaluation passband to 896 frequency channels which covers 106 $\leq \nu \leq$ 194 MHz.

A simple approach to evaluation would rely on using the accuracy of predictions meaning we only look at the number of correctly predicted RFI pixels relative to all RFI, although this metric hides important details to the performance of our networks. This is an important consideration in this instance because our HERA data observations contain on average 3 per cent of data corrupted by RFI, which means a blanket classification of 'No-RFI' would yield an accuracy rate of 97 per cent. To account for this class imbalance we evaluate the effectiveness of our networks by using several metrics commonly employed for classification. We use the standard metrics of Recall and Precision, which are defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{RFI}_{\text{Correct}}}{\text{RFI}_{\text{Correct}} + \text{RFI}_{\text{Incorrect}}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{RFI}_{\text{Correct}}}{\text{RFI}_{\text{Correct}} + \text{NoRFI}_{\text{Incorrect}}} \tag{3}$$

where we consider true positives (TP) to be correctly predicted RFI pixels, false negatives (FN) are RFI pixels identified as No-RFI, and false positives (FP) are no-RFI pixels incorrectly identified as RFI. We can therefore understand Recall as the fraction of all RFI events identified by the flagging algorithm, and Precision the fraction of identified RFI that is actually RFI.

We also need a metric that is sensitive to a data set with a sparse class, as in our case where RFI represents < 3 per cent of our observations, and one that can condense our overall performance into a single metric. We therefore can use the binary classifier performance metric, the F-score which has the general form of

$$F_\beta = (1 + \beta^2)\,\frac{\text{Recall} \cdot \text{Precision}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{4}$$

where we set $\beta = 2$ to preferentially weight Recall.[2] The F_2 score therefore provides us with a metric that has an aggressive stance towards RFI while still being somewhat sensitive to false positive flagging. Due to the nature of measuring the 21 cm EoR signal with HERA where we have collected sufficient data to not be noise limited, we can sacrifice good quality observations for the sake of reducing as much RFI contamination as possible; thus allowing us a higher rate of FPs.

We compare three distinct algorithms: the amplitude-only DFCN, amplitude–phase DFCN, and the Watershed RFI algorithm. For a fair comparison, we evaluate the DFCNs after both have converged independently of the number of training epochs, ensuring that they have learned the training data set to their maximum capability. The networks are then checked for overfitting by comparing that the evaluation loss is similar to that of each networks training loss when applied to the unseen 20 per cent of simulated visibilities set aside for evaluation.

Before we approach the evaluation on our HERA data, we can further optimize how our networks handle the shift in domain from simulation to observed. This optimization can be seen by looking at the receiver operating characteristics (ROC) curves in Fig. 4 of both the networks and the Watershed algorithm. The ROC curve gives us a general idea of the performance between our networks by looking at how the TP and FP rates respond to different thresholding values of each networks' output layer. An optimal threshold is found

[2]An F_β score with $\beta < 1$ describes a preference for weighting Precision over Recall.

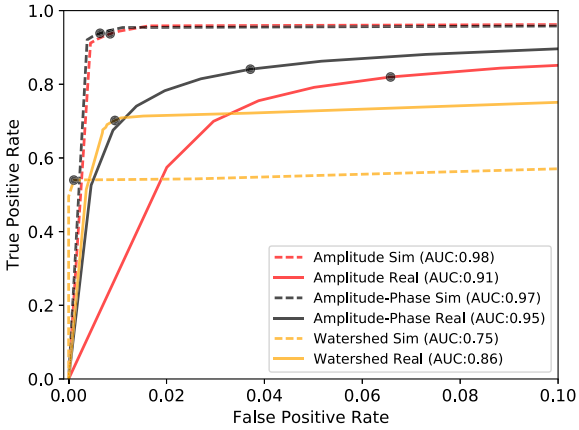

Figure 4. ROC curve showing a broad model comparison between all three RFI flagging algorithms, amplitude DFCN (red), amplitude–phase DFCN (black), and Watershed (orange). The ROC curves were derived from each algorithm predicting on real HERA data visibilities (solid) and simulated HERA visibilities (broken). Black circles represent the optimal F_2 score. The area under the curve (AUC) metric condenses the overall performance of our algorithms and tells us that the amplitude–phase network exhibits the best response on our real data with an AUC of 0.95. The TPR and FPRs for the real data (solid) are based on manually flagging RFI to the best of our ability to discern RFI from signals on the sky and therefore should not be taken as a ground truth.

per network/algorithm and per data set allowing us the freedom to tune our predictions for a particular instance. We determine this optimal thresholding value by choosing the maximum F_2 score across all thresholds which is shown to find a reasonable balance in TPR and FPR by locating the 'knee' of the ROC curve. By using an ROC curve in conjunction with the F_2 score which is reliant on our combined Recall and Precision we are able to compare our predictive power per data set while still being sensitive to our RFI/Non-RFI class imbalance.

For a more comprehensive understanding of each algorithm's behaviour, we present the performance metric results in Table 3 as applied to simulated data sets. These metrics are performed on data that are unique from the previous training/evaluation data sets and include a control data set which has no RFI present and four others that contain a single distinct class of RFI. An example of what each RFI class is modelled after is shown in Fig. 5. In doing this, we can gauge how sensitive each algorithm is to a certain class of RFI. The DFCN networks both perform well on the narrow-band time persistent and burst RFI which is unsurprising as these simulations closely resemble the evaluation data set and only differ in occurrence of events. However, both networks are inadequate for identifying broad-band bursts and 'blips'. This is understandable from a training perspective as both of these classes of RFI are going to be the last to be modelled in our networks as they account for only a minor fraction of all simulated RFI and lead to little overall optimization of the loss. This could potentially be remedied by placing more emphasis on these two classes of RFI in the training data set or a much more in depth hyperparameter optimization of per-layer kernel sizes.

We compare all three algorithms as applied to a simulated hera_sim and real HERA data set in Table 4. Looking at the prediction rates, both DFCN networks display an immense improvement over the Watershed RFI algorithm, boasting rates of 32 and 22 times better than the Watershed, for the amplitude and amplitude–phase networks respectively. The faster amplitude only prediction rate

Table 3. RFI recovery metrics based on individual type of simulated RFI. We look at the Recall, Precision, and F_2 score for each of the three algorithms as simulated with hera_sim. All metrics here are determined under the condition that an optimal threshold value was found by maximizing the F_2 score. The Recall and Precision rates are the average over 1000 simulated waterfall visibilities with the same simulation parameters for foregrounds and signal chain outlined in Section 2.3. Values in bold indicate the best achieved rate within each RFI type across algorithms.

		RFI classes			
Algorithm	No RFI Accuracy (per cent)	Narrow band Recall – Precision – F_2	Narrow-band burst ‘ ’	Broad-band burst ‘ ’	‘Blips’ ‘ ’
Amp DFCN	94	0.98 – 0.82 – 0.94	**0.77** – 0.65 – **0.74**	0.16 – 0.67 - 0.19	0.35 – 0.01 – 0.07
Amp–Phs DFCN	**98**	**0.99** – 0.83 – **0.95**	**0.77** – 0.67 – **0.74**	0.18 – 0.68 – 0.21	0.35 – 0.02 – 0.08
Watershed RFI	**98**	0.49 – **0.95** – 0.54	0.32 – **0.97** – 0.37	**0.99** – **0.74** – **0.98**	**0.71** – **0.73** – **0.71**

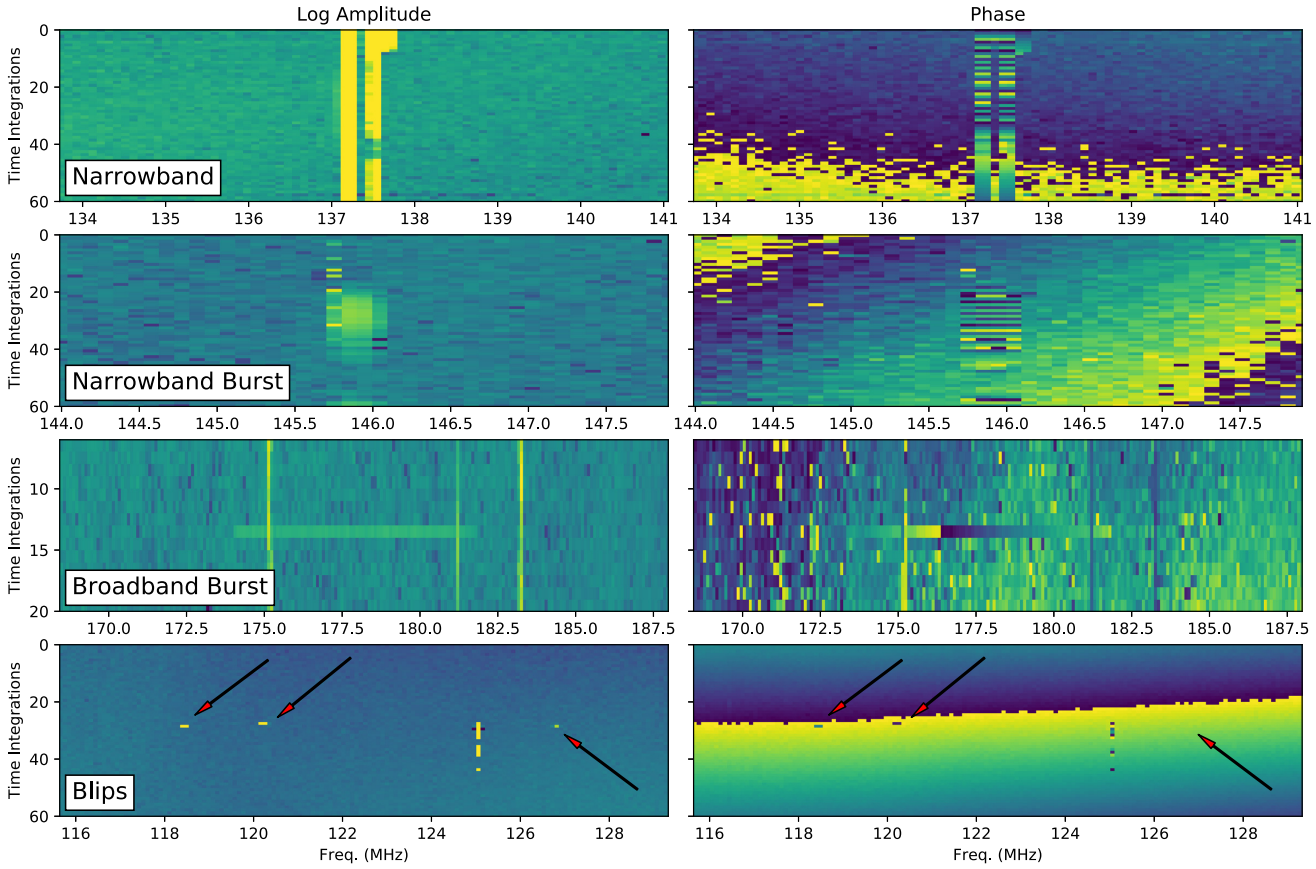

Figure 5. Examples of the four RFI classes from HERA data as they appear in amplitude and phase that we model in our simulations. Note the different time and frequency scales on each plot. The narrow-band example (row 1) centred at a frequency of ∼137.2 MHz is the ORBCOMM satellite system which is occasionally intermittent. Narrow-band burst (row 2) is typically limited across only a few frequency channels (≤500 kHz) and has no consistent operating pattern over time. Broad-band burst events (row 3) are short time duration (≤40 s) and can exist across the entire band (e.g. lightning) or in a sub-band as seen here flanked by the South African Broadcasting Corporation’s channel 4 video (175.15 MHz) and audio (181.15 MHz) broadcasts (Kohn 2016). The ‘blips’ (row 4) demonstrate the one off nature of this sparse class as compared to the intermittent transmitter at frequency 125 MHz.

Table 4. RFI recovery metrics for hera_sim simulated data containing signal chain effects with all classes of RFI and raw (uncalibrated) HERA observations from the 2017–2018 observing season. All results in the real HERA data column are based off of manually identified RFI and therefore the ground truth is uncertain especially in the low SNR limit for RFI. Our real HERA data included observations from LSTs of $0 \leq t \leq 5$ h and across baseline lengths of $7 \leq |\boldsymbol{b}| \leq 100$ λ. Values in bold correspond to the best achieved result for that metric. All three algorithms are run on a single NVIDIA GeForce GTX TITAN GPU.

Algorithm	hera_sim Recall – Precision – F_2	HERA real ‘ ’	Prediction rate waterfall h^{-1} GPU^{-1}
Amp DFCN	**0.90** – 0.61 – 0.82	0.76 – 0.42 – 0.65	2.4×10^5
Amp–phs DFCN	**0.90** – 0.82 – **0.88**	**0.81** – 0.58 – **0.75**	1.6×10^5
Watershed RFI	0.53 – **0.95** – 0.58	0.64 – **0.88** – 0.68	7.4×10^3

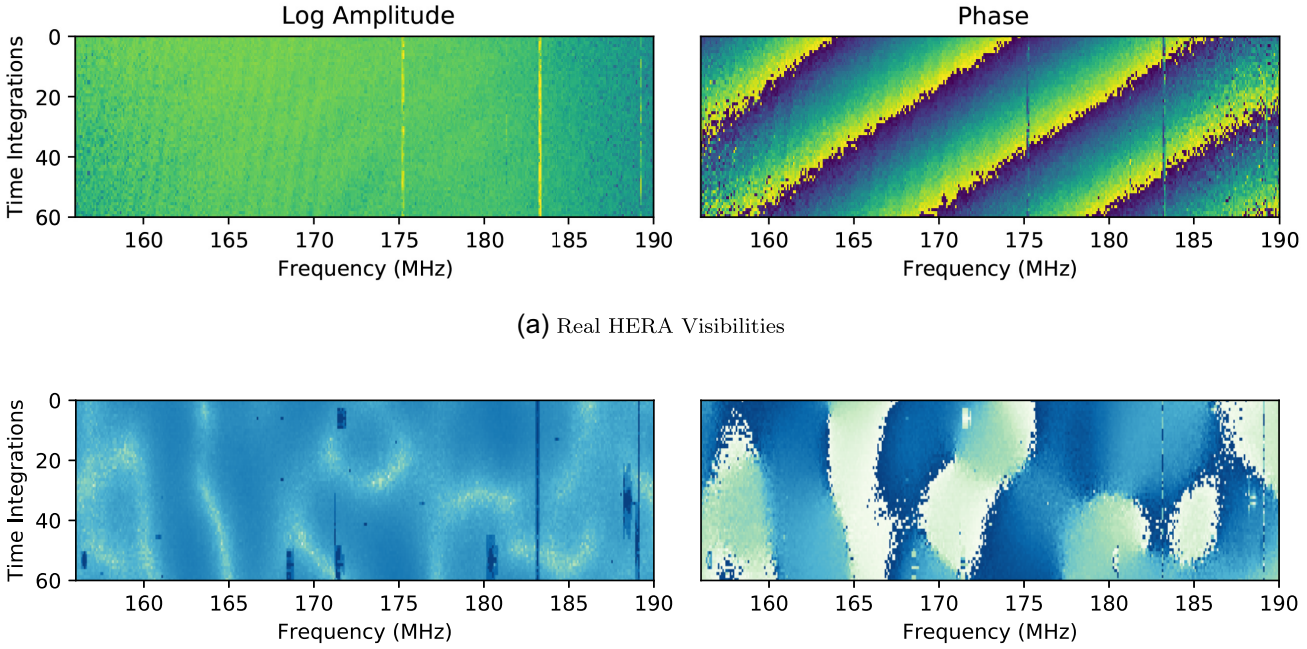

Figure 6. (a) Example of the real HERA waterfall visibilities used in Fig. 7 prior to any RFI flagging in amplitude and phase. (b) Simulated HERA waterfall visibilities used in Fig. 8 without RFI flagging.

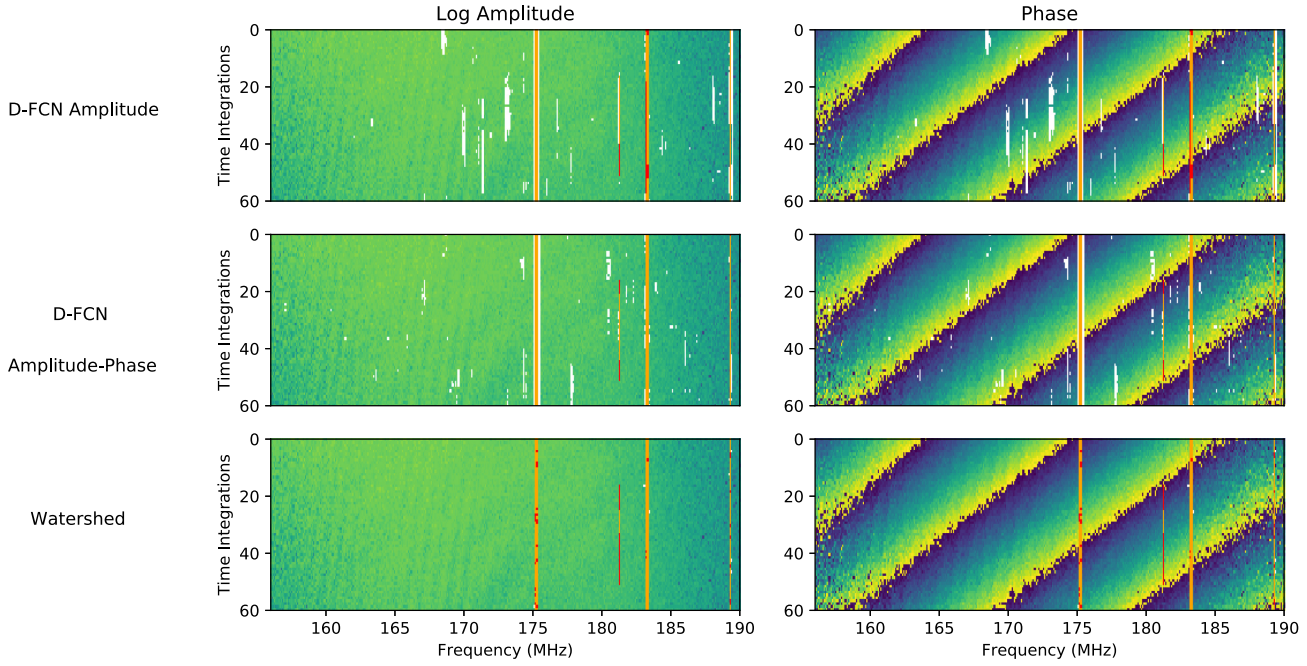

Figure 7. A comparison between the three flagging algorithms described in this paper as applied to a sub-band (157−193 MHz) from the real HERA data set, which has been flagged manually and is used as the approximate ground truth. Orange indicates TP, white is FP, and red represents FN. In this example, the amplitude–phase fed DFCN ultimately has the best TP outcome but, as seen in Table 4, both the DFCN algorithms take a more aggressive stance towards RFI resulting in higher rates of FP when compared to the Watershed algorithm.

compared to the amplitude–phase is unsurprising, as the number of parameters involved in an amplitude–phase prediction is roughly 1.5 times more and scales approximately proportional with the prediction rate.

We demonstrate each of these algorithms on a sample of real HERA data and a simulation from hera_sim, as shown in Fig. 6. An example of these results, which serves to give an appropriate idea of the average performance as applied to a real HERA waterfall

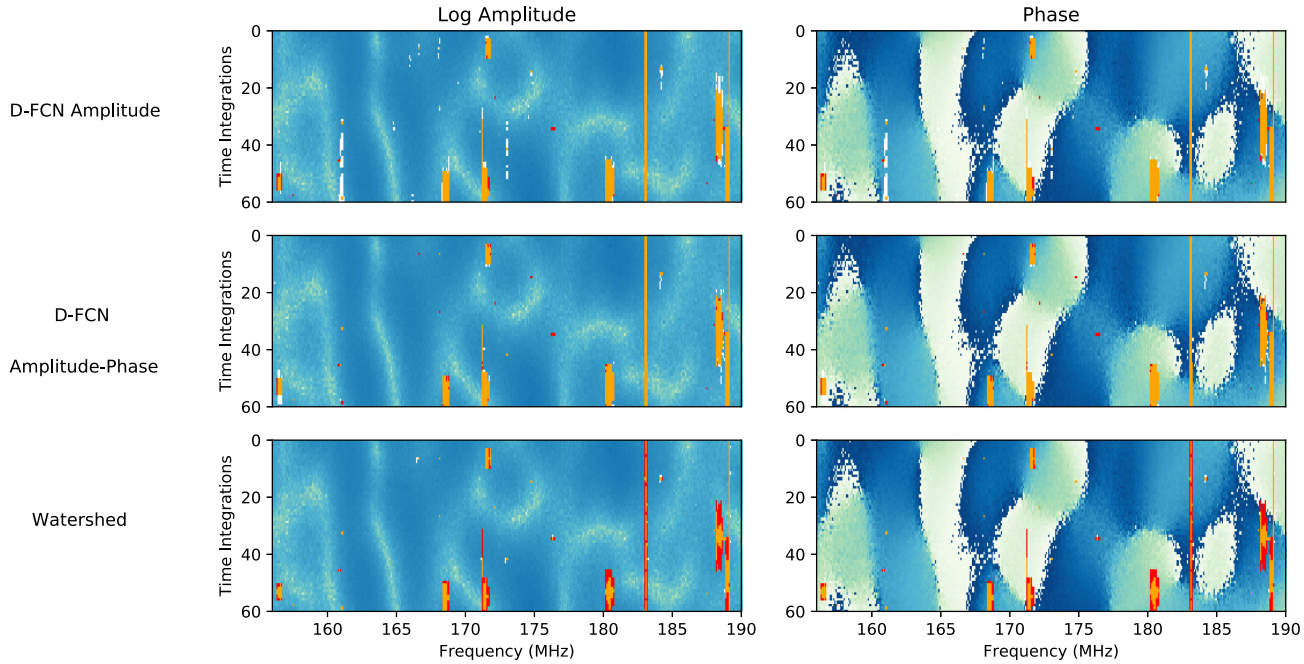

Figure 8. A similar comparison as in Fig. 7 demonstrating how each RFI flagging algorithm performs on simulated HERA data from hera_sim. Orange indicates TP, white is FP, and red is FN. The simulated waterfall visibility is of a 25λ baseline dominated by strong diffuse emissions from the de Oliveira-Costa et al. (2008) GSM. The Watershed algorithm's inability to discriminate between RFI and sky, as indicated by its higher FN rate, in this instance hints that there is required fine-tuning of its kernel size and initial threshold hyperparameters, due to the spectral structure in our simulations.

visibility is shown in Fig. 7; how each compares on simulated data can be seen in Fig. 8. Both DFCN networks have a tendency to overpredict RFI where there may not be any, however in the case of the narrow-band RFI seen in frequency channels 175 and 189 MHz, it may not be unreasonable to be more aggressive as leakage into adjacent channels can occur. This can be difficult to quantify of course as the ground truth of our real HERA data is unknown and RFI leakage can be easily masked by the sky.

4 CONCLUSIONS

Machine learning applications in the fields of astronomy and cosmology are rapidly developing, and in many cases are beginning to outmaneuver the classical algorithms by way of increased speed and more accurate predictions. In this paper, we described an RFI identification approach to using a DFCN, which combined the amplitude and phase of an interferometer's measured visibility to predict which time–frequency pixels contained RFI. We compared this result to the Watershed RFI algorithm in the HERA data processing pipeline, and demonstrated that the DFCN approach was vastly more time efficient in its prediction with comparable to improved RFI identification rates. We also show that by including the phase component of the visibility we can mitigate the effects of domain shift between an entirely simulated HERA visibility training data set and the observed validation data set. This means that by improving our simulated model for HERA visibilities, coupled with an amplitude–phase DFCN we should be able to achieve an extremely effective first-round RFI flagger that reduces a common pipeline bottleneck. For reference, in the current HERA data processing pipeline, the Watershed RFI algorithm consumes 60 per cent of the total computational time, where the other 40 per cent includes redundant and absolute calibration. We do however recognize that the DFCN approach can have issues with identifying RFI bursts that occupy single time–frequency samples, what we called 'blips', and broad-band bursts. This is most likely due to an imbalanced representation in our training data set, and the loss optimization not being rewarded enough to drive the DFCNs to learn a subclass that appears at a rate of < 0.1 per cent. This can be potentially overcome by fine-tuning the model by using transfer learning (Yosinski et al. 2014), and would involve a training data set which consists almost entirely of these two subclasses, where the trained DFCN model shown here would serve as the starting point.

In near future build-outs of HERA, there will need to be an extreme importance placed on reducing bottlenecks in the HERA data processing pipeline. The current Watershed RFI flagging algorithm does not scale particularly well, which puts this class of fully CNN as an ideal alternative. The eventual number of HERA dishes will total 350, which for a single 10 min observation gives us 61 075 unique waterfall visibilities. In the amplitude–phase DFCN design outlined in this paper the RFI flagging throughput is 1.6×10^5 waterfall h^{-1} GPU^{-1}[3] which compares to the Watershed RFI flagger at 7.4×10^3 on the same resources.

Future work related to the amplitude–phase DFCN could include a modification to a similarly styled comprehensive data quality classifier which should in-turn lead to improved results for sky based (Barry et al. 2016) and redundant calibration (Zheng et al. 2014), both of which requires exceptionally conditioned data. A strict binary classifier could be achieved by developing a training data set that does not use a mock sky, but an accurately modelled sky with a proper HERA beam model. Of course it would also be possible and might be better suited by developing an observation derived training data set in this instance, as failure modes are generally easier to identify in visibilities as opposed to contamination by RFI.

[3]Performed on a single NVIDIA GeForce GTX TITAN.

It should also be possible to extend this work to arrays with better temporal resolution such as the MWA (Tingay et al. 2015) in the search for transients like fast radio bursts (Zhang et al. 2018). The additional phase information could potentially reduce the low-end limit of fluence for identification due to a more significant contrast between RFI and sky fringes.

The github repository for the RFI DFCN described in this paper can be found at https://github.com/UPennEoR/ml_rfi.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant nos 1636646 and 1836019 and institutional support from the HERA collaboration partners. This research is funded in part by the Gordon and Betty Moore Foundation. HERA is hosted by the South African Radio Astronomy Observatory, which is a facility of the National Research Foundation, an agency of the Department of Science and Technology. This work was supported by the Extreme Science and Engineering Discovery Environment, which is supported by National Science Foundation grant number ACI-1548562 (Towns et al. 2014). Specifically, it made use of the Bridges system, which is supported by National Science Foundation award number ACI-1445606, at the Pittsburgh Supercomputing Center (Nystrom et al. 2015). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research. SAK is supported by a University of Pennsylvania SAS Dissertation Completion Fellowship. Parts of this research were supported by the Australian Research Council Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D), through project number CE170100013. GB acknowledges support from the Royal Society and the Newton Fund under grant NA150184. This work is based on research supported in part by the National Research Foundation of South Africa (grant no. 103424). GB acknowledges funding from the INAF PRIN-SKA 2017 project 1.05.01.88.04 (FORECaST). We acknowledge the support from the Ministero degli Affari Esteri della Cooperazione Internazionale – Direzione Generale per la Promozione del Sistema Paese Progetto di Grande Rilevanza ZA18GR02. This work is based on research supported by the National Research Foundation of South Africa (grant number 113121).

REFERENCES

Abadi M. et al., 2016, Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI'16. USENIX Association, Berkeley, CA, USA, p. 265
Akeret J., Chang C., Lucchi A., Refregier A., 2017, Astron. Comput., 18, 35
Barry N., Hazelton B., Sullivan I., Morales M. F., Pober J. C., 2016, MNRAS, 461, 3135
Bengio Y., Courville A., Vincent P., 2013, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, 1798
Burd P. R., Mannheim K., März T., Ringholz J., Kappes A., Kadler M., 2018, Astron. Nachr., 339, 358
Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H., 2018, preprint (arXiv:1802.02611)
de Oliveira-Costa A., Tegmark M., Gaensler B. M., Jonas J., Landecker T. L., Reich P., 2008, MNRAS, 388, 247
DeBoer D. R. et al., 2017, PASP, 129, 045001
Fu J., Liu J., Wang Y., Lu H., 2017, preprint (arXiv:1708.04943)
Guha Roy A., Conjeti S., Navab N., Wachinger C., 2018, preprint (arXiv:1801.04161)
Hazelton B., Jacobs D., Pober J., Beardsley A., 2017, J. Open Source Softw., 2, 10
He K., Zhang X., Ren S., Sun J., 2016, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Deep Residual Learning for Image Recognition. p. 770
Hurley-Walker N. et al., 2017, MNRAS, 464, 1146
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Kohn S. A., 2016, RFI in HERA19, Memo Series 19, HERA Collaboration. Available at: http://reionization.org/science/memos/
Krizhevsky A., Sutskever I., Hinton G. E., 2017, Commun. ACM, 60, 84
LeCun Y., Bengio Y., 1998, Chapt. Convolutional Networks for Images, Speech, and Time Series. MIT Press, Cambridge, MA, USA, p. 255
Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, Proc. IEEE, 86, 2278
Lin M., Chen Q., Yan S., 2013, preprint (arXiv:1312.4400)
Long J., Shelhamer E., Darrell T., 2014, preprint (arXiv:1411.4038)
Maas A. L., Hannun A. Y., Ng A. Y., 2013, ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Rectifier nonlinearities improve neural network acoustic models
Nystrom N. A., Levine M. J., Roskies R. Z., Scott J. R., 2015, Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure (XSEDE'15). ACM, New York, NY, p. 30
Offringa A. R. et al., 2013, A&A, 549, A11
Offringa A. R. et al., 2015, PASA, 32, e008
Offringa A. R., van de Gronde J. J., Roerdink J. B. T. M., 2012, A&A, 539, A95
Offringa A. R., Mertens F., Koopmans L. V. E., 2019, MNRAS, 484, 2866
Parsons A., 2015, Power Spectrum Normalizations for HERA, Memo Series 27, HERA Collaboration, Available at: http://reionization.org/science/memos/
Parsons A., Beardsley A., 2017, Modeling Sky Temperature, Memo Series 34, HERA Collaboration, Available at: http://reionization.org/science/memos/
Parsons A. R., Pober J. C., Aguirre J. E., Carilli C. L., Jacobs D. C., Moore D. F., 2012, ApJ, 756, 165
Roerdink J. B., Meijster A., 2000, Fundam. Inf., 41, 187
Ronneberger O., Fischer P., Brox T., 2015, preprint (arXiv:1505.04597)
Tingay S. J. et al., 2015, AJ, 150, 199
Towns J. et al., 2014, Comput. Sci. Eng., 16, 62
Yosinski J., Clune J., Bengio Y., Lipson H., 2014, Proc. 27th International Conference on Neural Information Processing Systems (NIPS'14) – Vol. 2. MIT Press, Cambridge, MA, USA, p. 3320
Zhang Y. G., Gajjar V., Foster G., Siemion A., Cordes J., Law C., Wang Y., 2018, ApJ, 866, 149
Zheng H. et al., 2014, MNRAS, 445, 1084

APPENDIX A: WATERSHED RFI ALGORITHM

The current algorithm used in the HERA analysis pipeline is the Watershed RFI algorithm. This algorithm was developed for use on data from the Precision Array for Probing the Epoch of Reionization Array and HERA. Prior to identifying and removing suspected RFI instances, the watershed algorithm uses a median filter that is applied to the raw data. In one dimension, a median filter is defined by the radius of the kernel K, which is applied as a sliding window across the entire length of the input data vector. Specifically, given an input vector $\boldsymbol{x} = [x_0, x_1, \ldots, x_N]$, the median filtered output for a given entry $\tilde{x}_i$ can be expressed as:

$$\tilde{x}_i = \texttt{median}(x_{i-K}, x_{i-K+1}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+K}), \quad \text{(A1)}$$

where `median()` is a function which returns the median of the list of data. By construction, the list will have an odd number of elements in it, and so the median is guaranteed to be an entry in the list.

In two dimensions, the median filter is defined analogously to the ID case, except that there are two filter radii (K_t, K_ν) that define the median filter. Here, we have used the subscripts t and ν to represent the time and frequency axes found in a visibility waterfall. In general these need not be the same, but in practice as part of the HERA

Algorithm 1 Watershed XRFI Algorithm

procedure XRFI($V_{ij}(t, \nu)$)
  $\tilde{V}_{ij}(t, \nu) \leftarrow$ `medfilt2d`$(V_{ij}(t, \nu), K_t, K_\nu)$
  $\sigma_{ij} \leftarrow \left(\sum_{t,\nu} \tilde{V}^2_{i,j}(t, \nu) - \sum_{t,\nu} \tilde{V}_{i,j}(t, \nu)\right)^{1/2}$
  $z_{ij}(t, \nu) \leftarrow \left|\tilde{V}_{ij}(t, \nu)/\sigma_{ij}\right|$

  where$\sim z_{ij}(t, \nu) > 6$ ▷ set initial flags
    $f_{ij}(t, \nu) \leftarrow$ `True`
  else where
    $f_{ij}(t, \nu) \leftarrow$ `False`
  end where

  `AddedFlags` $\leftarrow$ `True`
  while `AddedFlags` **do** ▷ flood fill to neighbors
    `AddedFlags` $\leftarrow$ `False`
    for all $f_{ij}(t, \nu) \in t, \nu$ **do**
      if f_{ij} is `True` **then** ▷ grow existing flags
        for $t' \leftarrow t \pm 1$ **do** ▷ check times
          if $z_{ij}(t', \nu) > 2$ **then**
            $f_{ij}(t', \nu) \leftarrow$ `True`
            `AddedFlags` $\leftarrow$ `True`
          end if
        end for
        for $\nu' \leftarrow \nu \pm 1$ **do** ▷ check frequencies
          if $z_{ij}(t, \nu') > 2$ **then**
            $f_{ij}(t, \nu') \leftarrow$ `True`
            `AddedFlags` $\leftarrow$ `True`
          end if
        end for
      end if
    end for
  end while

  return $f_{ij}(t, \nu)$
end procedure

pipeline, both have the same value of $K_t = K_\nu = 8$. Empirically, these values seem to fall into a 'sweet spot' of parameter space, where the values were large enough that the overall algorithm catches the majority of RFI events (as verified by inspecting the visibilities by hand) while still remaining computationally tractable to run. Also, to ensure the output of the median filter has the same dimensionality as the input data, the arrays are padded with a reflection of the data that is K_t or K_ν elements long, rather than with zero values, to avoid discontinuous jumps at the boundaries.

Physically, the median filter has the property of generating a proxy for the underlying noise of the raw visibility data because of its differencing of neighbouring time–frequency pixels, and helps detrend the smooth foreground structure that is quite prominent and exhibits a strong frequency dependence. Once the 2D median filter has been computed for every point in the visibility, the output is a 'noise' visibility. The standard deviation of this 'noise' is computed, which is then used to convert the noise to modified z-scores. (That is, the value of the noise is divided by the standard deviation, to quantify how strong of an outlier a particular data point is.) An initial round of seeds is generated by identifying all of the 6σ outliers (the data points whose absolute valued z-scores is greater than six). Once the data have been pre-processed in this fashion, the Watershed algorithm is used to identify all instances of RFI.

A Watershed algorithm (or more correctly, a flood-fill algorithm, because the resulting image segments are not grouped or labelled) is then used to identify the remaining RFI instances in the waterfall.[4] Under the assumption that RFI events tend to have some coherency either in time (e.g. for narrow-band emission that is almost always on, such as ORBCOMM) or in frequency (e.g. for broad-band RFI events caused by lightning), the initial flags generated by finding 6σ outliers are extended to neighbouring pixels if the absolute value of their z-score is greater than 2. These regions are extended until no neighbouring 2σ values are encountered.

Algorithm 1 shows the pseudo-code of the XRFI flagging algorithm. The algorithm takes in a waterfall of visibility data $V_{ij}(t, \nu)$ and returns a set of flags $f_{ij}(t, \nu)$ of the same dimensionality. There are three main phases:

(i) Pre-process the visibility data.
(ii) Generate initial series of flags.
(iii) Flood-fill around initial flags to generate full set of flags.

As currently implemented, the Watershed XRFI algorithm operates on the absolute value of the visibility data, though it could be extended to operate on the real and imaginary components as well. When running the Watershed XRFI algorithm in production, the most computationally expensive part is the 2D median filter which has a time complexity of $\mathcal{O}(K_t K_\nu)$. The overall complexity is roughly $\mathcal{O}(N_t N_\nu K_t K_\nu)$ for a waterfall visibility with dimensions $N_t \times N_\nu$. Thus, speeding up the median filter operation by decreasing the kernel size or leveraging GPU computing can provide a significant speedup. Nevertheless, preliminary results have shown that this speedup is still unable to match the speed offered by neural networks.

[1]*Department of Physics, Brown University, Providence, Rhode Island, RI, USA*
[2]*Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA, USA*
[3]*Department of Astronomy, University of California, Berkeley, CA, USA*
[4]*Cavendish Astrophysics, University of Cambridge, Cambridge, UK*
[5]*SKA SA, 3rd Floor, The Park, Park Road, Pinelands 7405, South Africa*
[6]*School of Earth and Space Exploration, Arizona State University, Tempe, AZ, USA*
[7]*Department of Physics and Electronics, Rhodes University, PO Box 94, Grahamstown 6140, South Africa*
[8]*INAF – Istituto di Radioastronomia, via Gobetti 101, I-40129 Bologna, Italy*
[9]*National Radio Astronomy Observatory, Charlottesville, VA, USA*
[10]*National Radio Astronomy Observatory, Socorro, NM, USA*
[11]*Department of Engineering, University of California, Berkeley, CA, USA*
[12]*Department of Physics, Massachusetts Institute of Technology, Cambridge, MA, USA*
[13]*Department of Physics and Astronomy, University of California, Los Angeles, CA, USA*
[14]*School of Physics, University of Melbourne, Parkville, VIC 3010, Australia*
[15]*ARC Centre of Excellence for All-Sky Astrophysics in 3 Dimensions (ASTRO 3D), University of Melbourne, VIC 3010, Australia*
[16]*Department of Physics, University of Washington, Seattle, WA, USA*
[17]*eScience Institute, University of Washington, Seattle, WA, USA*
[18]*Department of Physics and McGill Space Institute, McGill University, 3600 University Street, Montreal, QC H3A 2T8, Canada*
[19]*Scuola Normale Superiore, I-56126 Pisa, PI, Italy*
[20]*Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA*

[4]Please see Roerdink & Meijster (2000) for a more in-depth understanding of the Watershed algorithm

This paper has been typeset from a TEX/LATEX file prepared by the author.