

DATA ANALYSIS METHODS FOR THE DETECTION OF THE EPOCH OF REIONIZATION

CARINA CHENG¹, ET AL.

¹Astronomy Dept., U. California, Berkeley, CA

ABSTRACT

The Epoch of Reionization (EoR) is an uncharted era in our Universe’s history during which the birth of the first stars and galaxies led to the ionization of neutral hydrogen. This important epoch of our cosmic dawn harbors a wealth of information regarding the environment during this transformative time, including insight into the nature of the first luminous sources and implications about cosmological parameters. There are many experiments investigating the EoR by tracing the 21 cm line of neutral hydrogen, a signal which is very faint and difficult to isolate. With a new generation of instruments and a statistical power spectrum detection in our foreseeable future, it has become increasingly important to develop techniques that help maximize sensitivity and validate results. Additionally, it is imperative to understand the trade-offs between different methods and their effects on common power spectrum themes. In this paper, we focus on three major themes - signal loss, power spectrum error bar estimation, and bias in measurements. We describe techniques that affect these themes using both a toy model and data taken by the 64-element configuration of the Donald C. Backer Precision Array for Probing the Epoch of Reionization (PAPER).

1. INTRODUCTION

By about one billion years after the Big Bang, the very first stars and galaxies are thought to have ionized all the neutral hydrogen that dominated the baryonic matter content in the early Universe. This transition period, during which the first luminous structures formed from gravitational collapse and began to emit intense radiation that ionized the cold neutral gas into a plasma, is known as the epoch of reionization (EoR). The EoR is a relatively unexplored era in our cosmic dawn. Its history encodes important information regarding the nature of the first galaxies and the processes of structure formation. Direct measurements of the EoR would unlock powerful information about the intergalactic medium, revealing connections between the smooth matter distribution exhibited via cosmic microwave background (CMB) studies and the highly structured web of galaxies we observe today.

One promising technique to probe the EoR is to target the 21 cm wavelength emission that is emitted by neutral hydrogen via its spin-flip transition. This technique is powerful because it can be observed as a function of redshift — that is, the wavelength of the signal reaching our telescopes can be directly mapped to a distance from where the emission originated before stretching out as it traveled through expanding space. The 21 cm line therefore offers a window into the evolution of ionization, temperature, and density fluctuations on cosmic scales.

Although a detection of the EoR remains elusive, there are several radio telescope experiments that have succeeded in using the 21 cm signal from hydrogen to place constraints on the brightness of the EoR. Examples of

experiments investigating the mean brightness temperature of the EoR relative to the CMB are the Experiment to Detect the Global EoR Signature (EDGES, [Bowman & Rogers 2010](#)), the Long Wavelength Array (LWA, [Ellingson et al. 2009](#)), the Large Aperture Experiment to Detect the Dark Ages (LEDA, [Greenhill & Bernardi 2012](#)), the Dark Ages Radio Explorer (DARE, [Burns et al. 2012](#)), the Sonda Cosmológica de las Islas para la Detección de Hidrógeno NeutroSciHi (SCI-HI, [Voytek et al. 2014](#)), the Broadband Instrument for Global Hydrogen Reionisation Signal (BIGHORNS, [Sokolowski et al. 2015](#)), and the Shaped Antenna measurement of the background RAdio Spectrum (SARAS, [Patra et al. 2015](#)). Radio interferometers which seek to measure statistical power spectra include the Giant Metre-wave Radio Telescope (GMRT, [Paciga et al. 2013](#)), the LOw Frequency ARray (LOFAR, [van Haarlem et al. 2013](#)), the Murchison Widefield Array (MWA, [Tingay et al. 2013](#)), the 21 Centimeter Array (21CMA, [Peterson 2004](#), [Wu 2009](#)), and PAPER ([Parsons et al. 2010](#)). The Hydrogen Epoch of Reionization Array (HERA), which is currently being built, is a next-generation instrument that aims to combine lessons learned from previous experiments and is forecasted to be able to make a successful 23σ detection with an eventual 350 elements, using current analysis techniques ([DeBoer et al. 2017](#), [Pober et al. 2014](#)).

The major challenge that faces all 21 cm experiments is isolating a small signal that is buried underneath foregrounds and instrumental systematics that are 4-5 orders of magnitude brighter. A clean measurement therefore requires an intimate understanding of the instrument and a rigorous study of data analysis choices. With

continual progress being made in the field and HERA on the horizon, it is becoming increasingly important to understand how the methods we choose interact with each other to affect power spectrum results. In this paper, we discuss three themes essential to a 21 cm power spectrum analysis and how data analysis choices affect each. We approach these themes from a broad perspective, and then perform a detailed case study using data from the 64-element configuration of PAPER.

This paper is organized as follows. In section 2 we introduce the three themes of our focus, using a toy model to develop intuition into each one. Section 3 presents an overview of the PAPER-64 array and observations, highlighting key changes from [Ali et al. \(2015\)](#). Sections 4, 5, and 6 detail how the new PAPER-64 analysis quantifies signal loss, estimates error bars, and eliminates bias, respectively. We conclude in Section 7.

2. POWER SPECTRUM THEMES AND TECHNIQUES

There are many choices for a data analyst, such as how to combine time-ordered measurements, how to estimate its variance, how to weight data in a way that suppresses contaminated modes while not destroying an EoR signal, and how to identify the source of a detection. Many common techniques, such as averaging data, weighting, bootstrapping, and jack-knife testing, each address these issues yet harbor additional trade-offs. For example, an aggressive filtering method may succeed in eliminating interfering systematics but comes at the cost of losing some EoR signal. A chosen weighting scheme may maximize sensitivity but fail to suppress foregrounds.

Measuring the statistical 21 cm power spectrum requires robust methods for determining accurate confidence intervals and rigorous techniques to identify and suppress systematics. Additionally, we desire to maximize our sensitivity with filtering and weighting techniques. In this paper, we focus on four techniques and their effects on three overarching themes of 21 cm power spectrum estimation. We will give brief definitions now, and build intuition for each theme in the sections to follow.

Power Spectrum Themes

The accuracy and interpretation of a 21 cm power spectrum can be encompassed in the following three themes.

- **Signal Loss** (Section 2.1): As explained in the next section, certain analysis techniques can lead to the loss of the EoR signal. If not corrected for, it could lead to a false non-detection. Computing signal loss has subtle challenges but is crucial to the accuracy of any result.
- **Error Bar Estimation** (Section 2.2): Confidence intervals on the 21 cm power spectrum result determine the difference between a detection and a

null result, which have two very different implications. Errors can be estimated in a variety of ways, and we will discuss a few of them.

- **Bias** (Section 2.3): There are several possible sources of bias in a visibility measurement that can leak its way into a power spectrum, such as bias from noise and foregrounds. Proving a bias is EoR may be the most difficult challenge for 21 cm analyses, as it is crucial to be able to distinguish a detection of foreground leakage, for example, from that of EoR. In this paper we will highlight some sources of bias and discuss ways to mitigate their effects.

Power Spectrum Techniques

The following techniques each have advantages when it comes to maximizing sensitivity and understanding systematics in data. However, they also have limitations and circumstances in which there are trade-offs.

- **Fringe-rate filtering:** Fringe-rate filtering can be considered an averaging scheme for time-ordered data ([Parsons et al. 2016](#)), and we explain the trade-offs of filtering in more detail in Section 2.1. Broadly, a fringe-rate filter increases the sensitivity of a dataset and reduces the number of independent samples by an amount dependent on the width of the averaging window. However, it can also affect the presence of foregrounds and systematics.
- **Weighting:** A dataset can be weighted to emphasize certain spectral features and minimize others. One particular flavor of weighting is inverse covariance weighting, which weights a dataset by minimizing the covariance between frequency channels. This weighting has the effect of down-weighting correlated information (i.e. foregrounds) and up-weighting noise-like information (i.e. EoR). However, one challenge of weighting is in accurately determining a covariance matrix that best describes the data.
- **Bootstrapping:** Because applying theoretical errors to a 21 cm measurement are insufficient [[why?](#)], bootstrapping is a useful method for estimating errors of a dataset from itself. By randomly drawing many samples of the data, we get a sense of its inherent variance, though there are subtleties to consider such as bootstrapping sample sizes.
- **Jack-Knife testing:** A resampling technique useful for estimating bias, jack-knives can be taken along different dimensions of a dataset to cross-validate results. In particular, null tests can be used to verify whether results are free of systematics ([Keating et al. 2016](#)).

In the next three subsections, we go into each theme in depth, focusing on how power spectrum technique trade-offs affect each. We use a toy data model to develop intuition into why certain analysis choices may be appealing and discuss ways in which they are limited. We highlight problems that can arise regarding each theme and offer suggestions to mitigate the issues. Ultimately, [some kind of summarizing sentence here].

2.1. Signal Loss

Signal loss refers to the attenuation of the target cosmological signal in a dataset. It can arise in a variety of ways in the analysis pipeline, such as fitting a polynomial during spectral calibration, applying a delay-domain filter, or by weighting data by itself. Here we focus on signal loss associated with applying a weighting matrix to data. Driven by the need to mitigate foreground bias, we would like to use a weighting method that succeeds in down-weighting foregrounds. However, as we'll investigate, this carries the risk of large amounts of signal loss.

Before we demonstrate how different weighting matrices affect signal loss, we first recap how weights are related to the power spectrum estimation technique of optimal quadratic estimators (OQE, Liu et al. 2014). A summary of the method is as follows.

We begin with our data vector, \mathbf{x} , which contains our measured visibilities in mK that have the shape (*times, frequencies*). We form the un-normalized power spectrum estimate \hat{q}_α :

$$\hat{q}_\alpha = \frac{1}{2} \mathbf{x}^\dagger \mathbf{w} \mathbf{Q}_\alpha \mathbf{w} \mathbf{x} \quad (1)$$

where \mathbf{w} [change to capital letter that isn't taken yet!] is a weighting matrix (for example, inverse covariance weighting would set $\mathbf{w} = \mathbf{C}^{-1}$). The matrix \mathbf{Q} is an operator that takes our frequency-domain visibilities and Fourier-transforms them into power spectrum space. The index α denotes a waveband in k_{\parallel} , where k_{\parallel} is the Fourier-dual to frequency.

We normalize our power spectrum estimates using the matrix \mathbf{M} :

$$\hat{\mathbf{p}} = \mathbf{M} \hat{\mathbf{q}} \quad (2)$$

where $\hat{\mathbf{p}}$ is the normalized estimate of the true power spectrum. We compute \mathbf{M} using the Fisher matrix \mathbf{F} , defined as:

$$\mathbf{F}_{\alpha\beta} = \frac{1}{2} \text{tr}[\mathbf{w} \mathbf{Q}_\alpha \mathbf{w} \mathbf{Q}_\beta]. \quad (3)$$

We have a choice for \mathbf{M} , and for simplicity in this section we set $\mathbf{M} \propto \mathbf{I}$, the identity matrix. This is not the case for the analysis of PAPER-64 data, which we describe in Section [not sure which section to point to yet] as part of the discussion of bias.

In the next sections, we will experiment with different weighting matrices \mathbf{w} and their impact on the resulting power spectra estimate $\hat{\mathbf{p}}$.

2.1.1. Toy Model: Inverse Covariance Weighting

To build our intuition into how a particular choice of weighting, namely inverse covariance weighting, can lead to signal loss, we use a toy model. Suppose we have a very simple dataset that contains 2-dimensional data (representing 100 time integrations and 20 frequency channels). This model represents realistic dimensions of about an hour of PAPER data which could be used for a power spectrum analysis. We create a mock bright foreground signal, \mathbf{x}_{FG} , as a complex sinusoid that varies in time and frequency, as well as a mock EoR signal, \mathbf{x}_{EoR} , as complex, random Gaussian noise. The sum of the two forms our combined data vector, \mathbf{x} , to which we will be applying different weighting schemes. The three data vectors are shown in Figure 1. Our ultimate goal in experimenting with weighting is to suppress the foregrounds perfectly without destroying EoR.

One choice for a weight matrix \mathbf{w} is an inverse covariance matrix. This type of weighting is attractive for power spectrum analyses because it is an aggressive way to down-weight foregrounds. The covariance matrix \mathbf{C} , which describes covariances between frequency channels, can be estimated in a variety of ways. Should we have perfect foreground, instrumental, and EoR models, we could form \mathbf{C} in a way that accurately describes our real data. However, if our foreground model is flawed for example, our estimate of \mathbf{C} would not be successful at down-weighting the real foregrounds in the data. One attractive way to estimate \mathbf{C} then is to empirically derive it from the data vector \mathbf{x} itself:

$$\mathbf{C} \equiv \langle \mathbf{x} \mathbf{x}^\dagger \rangle, \quad (4)$$

assuming $\langle \mathbf{x} \rangle = 0$. The inverse covariance matrix is therefore $\mathbf{w} = \mathbf{C}^{-1}$.

If \mathbf{C} is computed from the data itself and its inverse is then applied to the data, it carries the risk of over-fitting information in the data and introducing a multiplicative bias to estimates of the signal. If instead we choose $\mathbf{w} = \mathbf{I}$, the identity matrix, there will be no signal loss (although it may suffer from foreground or instrumental bias). We will refer to this choice as ‘unweighted’, since using the identity does not alter (doesn't upweight or downweight) the data at all.

First, we compute the power spectrum of \mathbf{x} using the OQE formalism and a weighting matrix of \mathbf{C}^{-1} . The result is shown in green in the left plot of Figure 2. Also plotted in the figure are the unweighted power spectrum of \mathbf{x}_{FG} (blue) and \mathbf{x}_{EoR} (red).

As shown in Figure 2 (left plot), our inverse covariance weighted result successfully suppresses foregrounds. This illustrates why inverse covariance weighting is attractive for 21 cm data. It is also evident that our result fails to recover the EoR signal — it exhibits the correct shape, but the amplitude level is slightly low. This is evidence of signal loss. In order to understand how this occurs, we need to closely study our covariance matrix, \mathbf{C} .

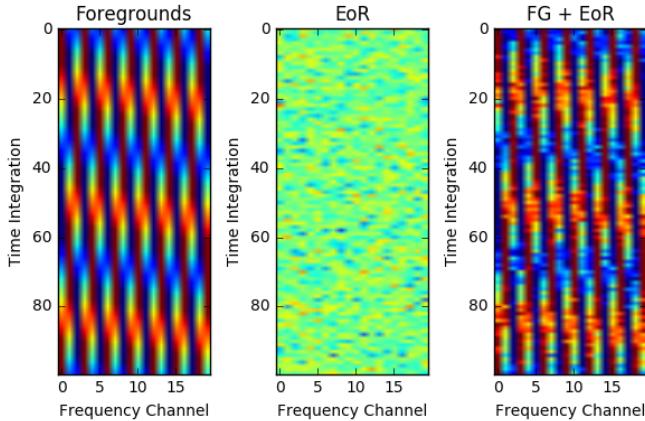


Figure 1. Our toy model dataset contains a sinusoid foreground that varies in time and frequency and random Gaussian noise as a mock EoR signal. Real parts are shown here. [colorbar?]

It turns out that because we estimated \mathbf{C} from our data, its eigenspectrum offers insight into understanding how inverse covariance weighting affects our result. An eigenspectrum ranks the eigenvalues of \mathbf{C} from highest to lowest and can be thought of as a spectrum of weights that are given to each frequency mode in the data. The eigenspectrum of the identity matrix \mathbf{I} is flat (all 1's) because it gives equal weighting to all modes.

For our toy model, the eigenspectra of \mathbf{C}_{FG} , \mathbf{C}_{EoR} , and \mathbf{C} are shown in Figure 3. The blue curve peaks only at the zeroth eigenmode, revealing that the foregrounds can be described by a single eigenmode. This is a direct consequence of how it was constructed (only one sinusoid). In fact, the blue peak in Figure 2 is also a direct result of our foreground's construction. The eigenspectrum of EoR, however, is fairly flat, signifying that its covariance matrix is similar to an identity matrix. This makes sense because the EoR signal consists of random noise, which should only have strong frequency-frequency correlations when two frequencies are the same.

The eigenmodes with the highest eigenvalues will be down-weighted the most (i.e. when the spectrum is inverted, they will be given the smallest weights). In contrast, the modes with the smallest eigenvalues will be up-weighted the most. In other words, if an eigenspectrum is not perfectly flat, all modes will be given different weights and it is possible to up-weight a few modes much more than others. The subtle slope of the eigenspectrum in this toy example gives rise to a small amount of signal loss because it is possible to overfit a few modes (the highest weighted ones) ever so slightly. We will see in the next section how this effect can be exaggerated as the ‘steepness’ of the spectrum changes.

Using what we've learned about the eigenspectrum, we can tweak it in a simple way to suppress foregrounds and yield zero signal loss. Namely, we can project out the zeroth eigenmode (i.e. zero out all eigenmodes except the first one), thereby down-weighting the foregrounds perfectly and nothing else. The result is shown

in the right plot of Figure 2. In this case, we perfectly recover EoR, demonstrating that if we can perfectly model our foregrounds, we can down-weight them without signal loss. However, this is not the case in reality since we aren't able to distinguish between foreground and EoR modes.

2.1.2. Toy Model: Fringe-Rate Filtering

We have shown how signal loss can arise due to unintentionally up-weighting eigenmodes with small eigenvalues (overfitting the noise). We will next show how this effect is exaggerated with a reduction of the total number of independent samples in a dataset.

A fringe-rate filter is an analysis technique designed to maximize sensitivity by integrating in time (Parsons et al. 2016). To mimic this filter, we average every 4 time integrations of our toy model dataset together, yielding 25 independent samples in time (Figure 4, left). We choose these numbers so that the total number of independent samples is similar to the number of frequency channels — therefore, our matrices will be full rank. The resulting eigenspectra (Figure 4, right), as compared to those in Figure 3, fall more steeply for the last few eigenmodes. This shape has large consequences on the resulting power spectra.

Applying inverse covariance weighting to this dataset results in the power spectra shown in Figure 5. There is a much larger amount of signal loss for this time-averaged dataset. To reiterate, the drop-off in the eigenspectrum shape suggests that when the spectrum is inverted, a few modes are up-weighted much more strongly than the rest. This dramatic up-weighting drowns out the EoR signal, giving rise to signal loss. Additionally, most of our power spectrum information is coming from only the last few modes and as a result, is a noisier estimate. This is evident by noticing that the green curve in Figure 5 fails to trace the shape of the unweighted EoR power spectrum.

Using our toy model, we have seen that a sensitivity driven analysis technique like fringe-rate filtering has trade-offs of signal loss and noisier estimates. Longer integrations increase sensitivity but reduce the number of independent samples, resulting in steep drop-offs in eigenspectra. For PAPER-64 data, we see ~ 3 orders of magnitude of signal loss when combining the use of an optimal fringe-rate filter and inverse covariance weighting.

2.1.3. Toy Model: Other Weighting

In a previous section we showed how altering an eigenspectrum can make the difference between zero and some signal loss, if we can distinguish between foreground eigenmodes and EoR eigenmodes. We will now use our toy model to describe several other ways to alter a covariance matrix \mathbf{C} in order to minimize signal loss.

As a first test, we know that our simulated EoR should have a covariance matrix that mimics the identity matrix, with its variance encoded along the diagonal. If we model \mathbf{C}_{EoR} as such, instead of computing it based

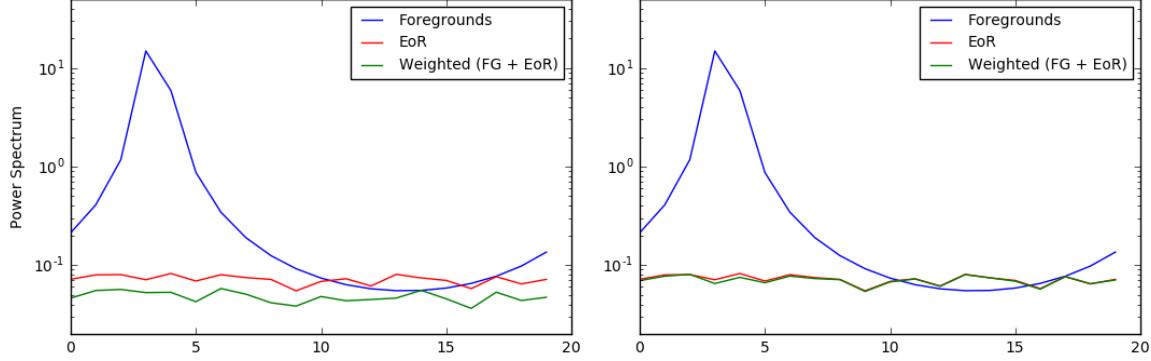


Figure 2. Resulting power spectrum estimates for our toy model simulation — foregrounds only (blue), EoR only (red), and the weighted FG + EoR dataset (green). We use inverse covariance weighting where \mathbf{C} is derived from the data (left), contrasted with projecting out the first eigenmode only (right). There is signal loss for the former but not the latter.

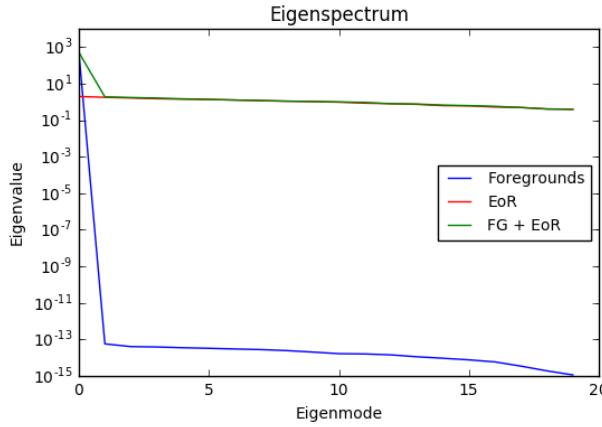


Figure 3. Eigenspectrum of foregrounds only (blue), EoR only (red), and FG + EoR (green). The eigenspectrum of foregrounds only peaks at the zeroth eigenmode, a consequence of the presence of only one sinusoid. All eigenspectra steepen slightly for the final few eigenmodes, a behavior that results in some modes being upweighted more strongly than others, thereby overfitting noise and causing signal loss in the final power spectrum estimate (Figure 2, left). If only the first eigenmode is used, all others get equal weights and hence, there is no signal loss (Figure 2, right)

on the dataset itself, and add it to $\mathbf{C}_{FG} = \mathbf{x}_{FG}\mathbf{x}_{FG}^\dagger$ to obtain a final \mathbf{C} to use in weighting, we see that there is negligible signal loss (Figure 6, upper left).

The next three panels of Figure 6 each involve methods that regularize the matrix \mathbf{C} after it is computed from the foreground + EoR dataset. By regularizing \mathbf{C} , we are flattening out its eigenspectrum. Therefore, some modes will be down-weighted, but the rest will be treated with similar weights. Hence, we won't be up-weighting the last few noisy modes.

Regularization schemes that we are highlighting here are adding an identity matrix to \mathbf{C} (Figure 6, upper right), killing all but the first three eigenmodes of \mathbf{C} (lower left), and multiplying an identity matrix to \mathbf{C} (lower right). Each avoids signal loss, but it is clear that two of the methods down-weight foregrounds much better than the third. For this toy model, our fore-

grounds are spread out in frequency and therefore have non-negligible frequency-frequency correlations. Therefore, when multiplying \mathbf{C} by an identity matrix, we are only preserving correlations between the same frequencies. Because we killed off information from all other frequency combinations, we do a poor job suppressing the foreground.

Although this method did not successfully recover EoR for this particular simulation, it is important that we show that there are many options for estimating a covariance matrix, and some may work better than others based on the dataset. One may imagine a situation where a particular systematic is contained to an isolated frequency. In such a case, preserving only the diagonal elements of \mathbf{C} would be an effective way of removing this contamination.

In summary, we have a choice of how to weight 21 cm data. Ideally, we want to down-weight bright foregrounds without removing the underlying cosmological signal. Inverse covariance weighting is an optimal weighting scheme for dealing with foregrounds, but its aggressiveness results in sloped eigenspectra shapes, the overfitting of noise, and thus signal loss. The application of a fringe-rate filter heightens this effect by reducing the number of independent samples and steepening eigenspectra further. Rather than use inverse covariance weighting, there are several alternate ways to regularize covariance matrices and flatten out eigenspectra to minimize signal loss. In Section 4, we apply the lessons we've learned about signal loss and its interaction with fringe-rate filtering and inverse covariance weighting to PAPER-64 data.

2.2. Error Estimation

Two ways of estimating errors on a power spectrum measurement are exploring the variance of a dataset, and computing a theoretical error estimate based on an instrument's system temperature and observational parameters. In a perfect world, both methods would match up. However, in practice the two don't always agree due to contaminants in the data that prevent it from being

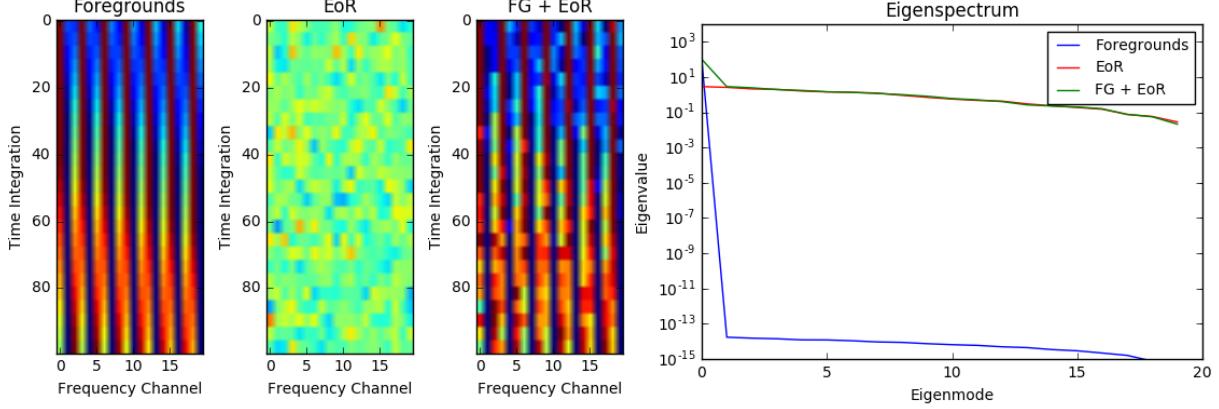


Figure 4. Left: Our ‘fringe-rate filtered’ (time-averaged) toy model dataset. It contains 25 independent samples in time. Real parts are shown here. Right: Eigenspectrum of the covariance matrices of foregrounds only (blue), EoR only (red), and FG + EoR (green). The eigenspectrum of foregrounds only peaks at the zeroth eigenmode, a consequence of the presence of only one sinusoid. All eigenspectra steepen for the final few eigenmodes, a behavior that results in a few noisy modes being upweighted more strongly than others, thereby overfitting noise and causing signal loss in the final power spectrum estimate.

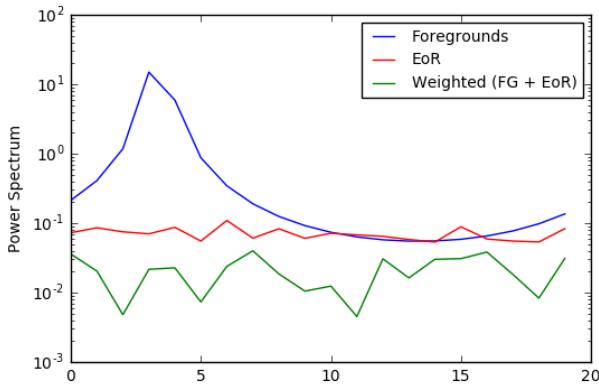


Figure 5. Resulting power spectrum estimate for our fringe-rate filtered toy model simulation — foregrounds only (blue), EoR only (red), and the weighted FG + EoR dataset (green). We use inverse covariance weighting where \mathbf{C} is derived from the data. There is a larger amount of signal loss than unfringe-rate filtered data. The power spectrum estimate is also noisier, since most of the information used in constructing the estimate came from only a few eigenmodes.

perfectly noise-limited. Therefore, it is important to place error bars on our measurements that have been derived from its inherent variance.

A common technique used to estimate the variance in a dataset is bootstrapping. Bootstrapping is a metric that relies on sampling with replacement and allows the estimation of a sampling distribution. For example, power spectrum measurements of 21 cm data can be made along many axes, including time and baselines. Through the process of bootstrapping along these axes, we can obtain error bars for our results which represent the underlying distribution of power spectra values that are allowed by our measurements.

Again, we focus on simple toy models to highlight two traps that one can fall into when bootstrapping power spectra. Suppose we have a Gaussian random noise dataset of length 1000 and variance of 1. We are interested in the average of the dataset, and therefore predict

that the error on the mean should obey $1/\sqrt{N}$, where N is the number of samples (in this case, 1000).

We form 100 bootstraps, each comprised of an array of length 1000 that is created by random sampling the original data with replacement. The standard deviation over the 100 bootstraps gives an error estimation for our dataset. As shown in Figure 7 (grey star), the error computed from bootstrapping matches our theoretical prediction.

One major caveat of bootstrapping is that the method is no longer valid when working with correlated data. In other words, bootstrapping assumes completely independent samples. Therefore, the use of a fringe-rate filter, which averages data in time to increase sensitivity, creates a situation in which bootstrapping can under-estimate errors. We will now show this effect using our toy model.

Going back to our toy model, we apply a sliding boxcar average to 10 samples at a time, thus reducing the number of independent samples to $1000/10 = 100$. Bootstrapping this time-averaged noise, using the same exact method as described earlier (drawing the same number of samples as the length of the dataset), under-estimates the error by a factor of ~ 3 . This occurs because we are drawing more samples than independent ones available. In fact, the error derived from bootstrapping is a strong function of the number of samples that are drawn (Figure 7, black points), and we can both under-estimate the error by drawing too many or over-estimate it by drawing too few. However, in this case we know that we have 100 independent samples, and the error associated with drawing 100 samples with replacement does match the theoretical prediction as expected.

This examples highlights the importance of understanding how analysis techniques (like a fringe-rate filter) can affect a common statistical procedure (like bootstrapping). Bootstrapping as a means of estimating power spectrum errors from real fringe-rate filtered data requires precise knowledge of the number of independent

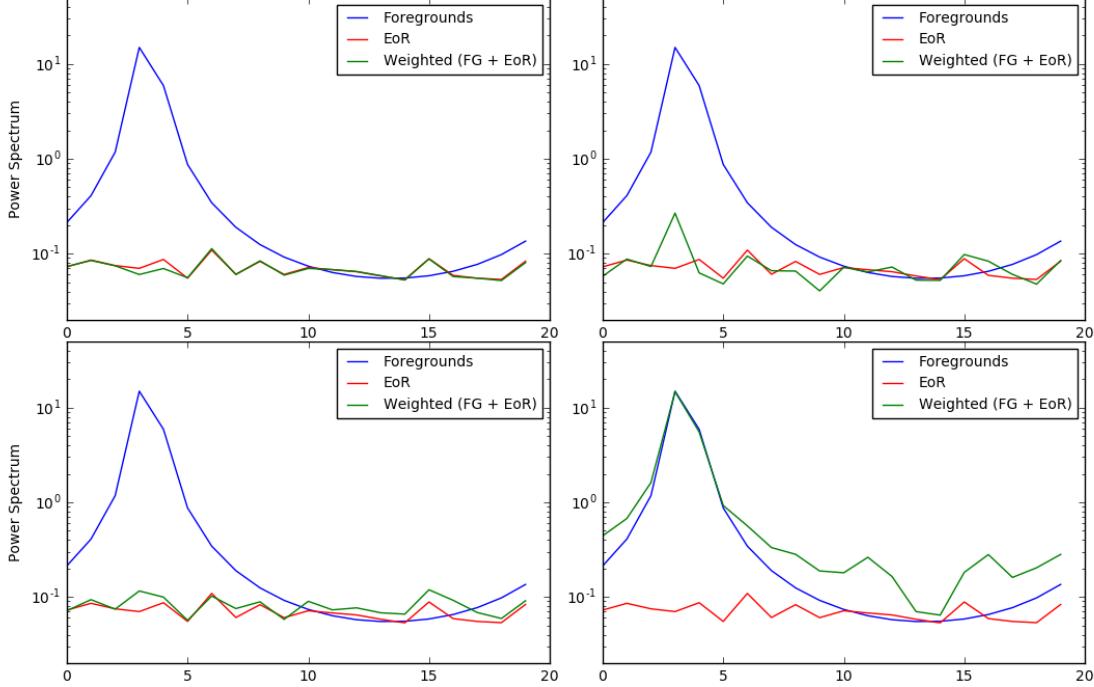


Figure 6. Resulting power spectra estimates for our fringe-rate filtered toy model simulation — foregrounds only (blue), EoR only (red), and the weighted FG + EoR dataset (green). We show four alternate weighting options that each avoid signal loss, including modeling the covariance matrix of EoR (upper left), regularizing \mathbf{C} by adding an identity matrix to it (upper right), using only the first three eigenmodes of \mathbf{C} (lower left), and multiplying an identity matrix to \mathbf{C} (lower right). There are many regularization options available to minimize signal loss, and some work better than others depending on the dataset.

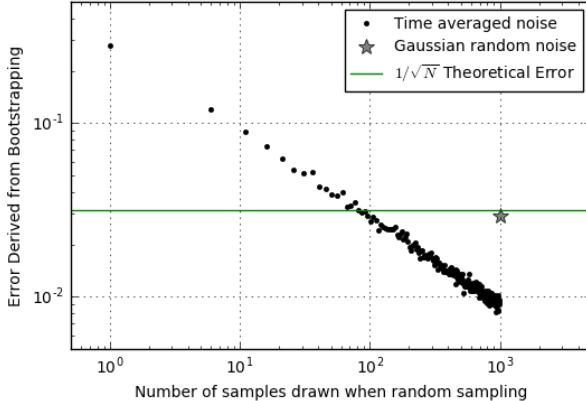


Figure 7. Power spectrum error estimation produced by bootstrapping as a function of the number of samples drawn when sampling with replacement. The grey star represents the error associated with drawing 1000 samples from a length 1000 array of random noise. The black points correspond to time-averaged data and illustrate how errors can be underestimated if drawing more samples than independent ones in the data. There are 100 independent samples in the time-averaged data — hence, using this number does indeed match the theoretical error prediction.

samples, which is not a trivial task.

We will now discuss a second subtle feature of bootstrapping that can lead to an over-estimation of errors. Suppose we have 1000 baselines (another appropriate axis for bootstrapping power spectra), each an independent measurement of the sky. Bootstrap theory requires a bootstrap to comprise of some combination of the 1000

baselines. If all 1000 spaces are filled randomly, there is in fact a high probability that baselines will be repeated in a sample (sometimes many times). Thus, bootstrapping in this way negatively affects power spectrum sensitivity because the number of independent baselines in a sample can be significantly lower than the total number of baselines.

In order to maximize sensitivity, we use a slightly modified bootstrapping method. We first shuffle all the baselines for a bootstrap, take the first 999, and then fill the last slot randomly with replacement. There's a small chance it will yield 1000 independent samples, but even with 999 our sensitivity is nearly maximized. One may wonder whether this change is still a legitimate way of error estimating since the number of possible random samples for a single bootstrap has dramatically decreased. However, as long as this number is still much greater than the number of bootstraps we perform, it is a valid way to uncover the inherent variability in a dataset.

For our toy model, the fraction of independent baselines using this method (999/1000) is over 1.5 times greater than the aforementioned approach. These two methods converge for small numbers of samples, when filling a few spots randomly is nearly the same as filling only the last spot randomly. The evolution of both of these methods as a function of number of baselines is shown in Figure 8. In order to reduce the number of baselines, we divide our 1000 baselines into N groups, where the number of baselines in a group is $1000/N$. No

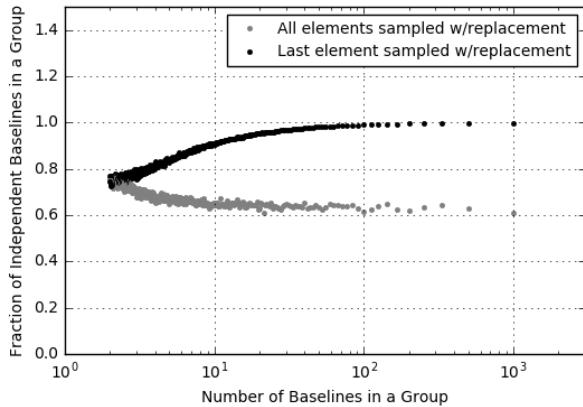


Figure 8. Fraction of independent baselines in a baseline group as a function of the number of baselines in a group. A fraction of 1.0 means that all baselines in a group are independent, and therefore sensitivity is maximized (alternatively, this axis can be treated as a power spectrum sensitivity in temperature-squared space). Two bootstrapping methods are shown here — sampling all elements with replacement (grey) and sampling only the last element with replacement (black). The former over-estimates error bars since some baselines end up being repeated due to random sampling. Therefore, baseline sensitivity is not maximized to its full capacity.

baselines are repeated within or between groups, until a bootstrapping method is applied to each (either sampling all elements with replacement, or only the last one).

It is clear that the fraction of independent baselines in a group is always greater using our revised sampling approach, with the greatest discrepancy for small number of groups (large number of baselines per group). The more baselines per group, the more sensitivity we can gain by ensuring that most of the baselines are independent ones. An analog for the y-axis in Figure 8 is therefore the final power spectrum sensitivity in temperature-squared space, which decreases as the square root of the number of baselines that are averaged together for a measurement.

In summary, bootstrapping is still an effective and straightforward way to estimate errors of a dataset. However, one must be careful about violating independence assumptions, as it is possible to under-estimate errors by over-sampling data. Additionally, small changes in the way a dataset is randomly sampled can ensure that its overall sensitivity is maximized.

We have discussed the interplay between two of our power spectrum techniques — fringe-rate filtering and bootstrapping — and how they affect the theme of error bar estimation. In addition to these, inverse covariance weighting can also affect error bars. Specifically, a power spectrum measurement and its errors are susceptible to blowing up when the number of non-zero eigenmodes are less than the number of independent samples in a dataset (i.e. the covariance matrix is not full rank). **[Not really sure how to explain this or how much detail to get into here...]**

Our fourth power spectrum technique, jack-knife test-

ing, is also related to error estimation. Splitting data along various axes and performing the same power spectrum analysis on each provides a test of robustness. If the same errors are obtained for each data subset, we can have confidence that they are unbiased.

2.3. Bias

There are several sources of bias that could leak into a power spectrum measurement, including noise and foreground biases. In this section we will describe analysis techniques that help mitigate these effects and how jackknife tests can tease out biases that are present in data. Ultimately, while different weighting and data combination techniques can help identify and remove biases, it is important to remember that a true detection of EoR will be robust to these tests. In other words, detections that aren't stable across multiple jack-knives are likely false ones and attributable to systematic errors.

Noise bias can arise in multiple ways. Every 21 cm visibility measurement contains thermal noise that is comprised of receiver and sky noise. We expect this noise to be independent between antennas and thus we can beat it down (increase sensitivity) by integrating longer, using more baselines, etc. However, a noise bias can result if the same thermal noise is multiplied by itself. This squaring of noise occurs when cross-multiplying visibilities to enter temperature squared space and results in power spectrum measurements that are higher than those predicted by the thermal noise of the instrument. One way to avoid this type of noise bias is to avoid cross-multiplying data from the same baselines or days. This ensures that the two quantities that go into a measurement have separate noises that don't correlate with each other.

Another way of introducing a noise bias into data is via cross-talk subtraction. Cross-talk is caused by the cross-coupling of signals between antennas and shows up as a constant bias in time in visibilities. For PAPER-64, cross-talk is removed during redundant calibration, when the deviation from an average visibility (across similar baseline types) is subtracted off of each visibility. This type of removal preserves signals common to all baselines, such as the EoR signal, but also has the consequence of introducing a low level bias. This bias term comes from the averaged cross-talk across a redundant baseline group and hence every baseline has a piece of all the other baseline's (of the same type) noise. One way to remove this noise bias is with the use of a fringe-rate filter, which can be chosen to filter out static signals (i.e. filter out zero 'fringe-rates').

Foreground bias is perhaps one of the main limiting factors pushing up against 21 cm results. Foreground signals lie $\sim 4 - 5$ orders of magnitude above the cosmological signal, and there are many existing techniques to avoid or remove these strong signals. Despite current best efforts to do so, however, there remains some foreground leakage that can show up as detections in a power spectrum. This leakage is strongest at low k

values, corresponding to large spatial scales on the sky.

As with noise bias, there are analysis techniques to mitigate the effects of foreground leakage and prevent information from low k 's from spreading to high k values. Narrow window functions can be used to minimize the influence of a particular k value on other ones ([[citation?](#)]). This in turn compromises power spectrum sensitivity (narrow window functions increases errors for each k -mode) as a trade-off for the minimization of foreground leakage. Another option is to construct an estimator using OQE that forces a window function to have a minimum response to low k values. The PAPER-64 window function is constructed in such a way, specifically to prevent foregrounds that live at low k 's from contaminating higher k -modes (Section 3.2).

One last source of foreground bias that we will highlight here is polarization leakage. When forming Stokes I from XX and YY polarization visibility data, we ignore the beam asymmetry between the two. This mismatch can cause leakage from Stokes Q ($V_Q = \frac{1}{2}(V_{XX} - V_{YY})$) into Stokes I ($V_I = \frac{1}{2}(V_{XX} + V_{YY})$). Another way of describing this leakage is that Faraday Rotation, which causes a rotation of the polarization angle of a signal, can induce spectral structure at certain delays (Moore et al. 2013). This type of sky bias is dependent on rotation measures, which characterize the amount of polarization rotation. If a particular rotation measure is such that a Stokes Q signal ends up falling at a power spectrum delay of interest, the Stokes I result for that k would be biased.

One preventative measure for polarization leakage could be tailoring a fringe-rate filter in a way that improves the match between XX and YY polarization beams (Parsons et al. 2016). There are also multiple ways of tracing a power spectrum detection to a polarization bias. One can look at Stokes Q power spectra to estimate the amount of expected leakage (Moore et al. 2013). In addition, jack-knife tests can be used to investigate whether polarization leakage affects a result. For example, rotation measures are highest when observing into the Galactic plane due to increased Faraday rotation ([[citation?](#)]), so one could imagine looking at different rotation measure bins along the Galactic latitude axis in order to identify whether a strong rotation measure coincides with a particular power spectrum bias.

In general, jack-knife tests are helpful in revealing a range of biases, including polarization and foreground biases. Slicing data along LST, for example, can determine whether excess detections are traceable to certain foreground sources. Slicing along baselines can uncover poorly operating baselines, or baselines whose weighting matrices aren't properly down-weighting foregrounds. Jack-knives can be taken along polarization axes, the frequency axis, daily axes, etc., and each can help trace a bias to a physical reason. A real 21 cm signal, however, should exist regardless of these tests, making jack-knives crucial for confidence in a successful EoR detection.

3. CASE STUDY: PAPER-64

Now that we have developed our intuition regarding signal loss, error estimation, and bias, we will look at each in more detail as applied to data taken by PAPER. We begin with an overview of the telescope array, its observations, and analysis pipeline.

3.1. Observations

The Donald C. Backer Precision Array for Probing the Epoch of Reionization (PAPER) is a dedicated 21 cm experiment located in the Karoo Desert in South Africa. The PAPER-64 configuration consists of 64 dual-polarization drift-scan elements, each 2 m on a side. The antenna layout is formatted in a grid layout (Figure 9), with 8 antennas on a side, 30 m spacing between antennas along the East/West direction, and 4 m spacing between antennas along the North/South direction. For the rest of this paper, we focus on data from only the 30 m pure East/West baselines.

PAPER-64 observed for a total of 135 nights between 2012-2013. The correlator processes a bandwidth of 100-200 MHz, corresponding to a redshift range of 6-12. For more information about the backend system of PAPER-64 and its observations, we refer the reader to [??] and Ali et al. (2015).

Because there is a detailed discussion of the PAPER-64 data reduction pipeline in Ali et al. (2015), here we will only briefly summarize the data processing steps prior to the power spectrum analysis.

Beginning with compressed data products which have been cleaned of radio frequency interference (RFI) at the 6σ level and then down-sampled (to 42.9 s integrations, 203 frequency channels), we then employ the technique of redundant calibration as a means of calibrating for individual antenna gains without needing to use information about the sky. Because PAPER is laid out on a grid, antenna calibration parameters can be found through the use redundancy. Namely, baselines of the same length and orientation should measure the same sky signal, but in practice there are differences due to instrumental effects caused by antennas, cables, and receivers. Using the basis of redundancy, however, allows us to set up a system of equations to solve for a complex gain per antenna that brings all baseline of a particular type into alignment. We perform these computations using the package OMNICAL (Zheng et al. 2014).

We next solve for the array's overall phase and gain calibration parameters using a standard self calibration method. We used radio sources Pictor A, Fornax A, and the Crab Nebula to fit for an overall phase solution, and set our overall flux scale using Pictor A as a calibrator source. Although PAPER-64 data exists for all 4 polarizations, we only use the xx and yy polarization data to form Stokes I visibilities as $V_I = \frac{1}{2}(V_{XX} + V_{YY})$.

Eliminating bright foregrounds remains a challenging yet crucial component of any 21 cm data analysis. There are many techniques to go about foreground removal, including spectral polynomial fitting, principle compo-

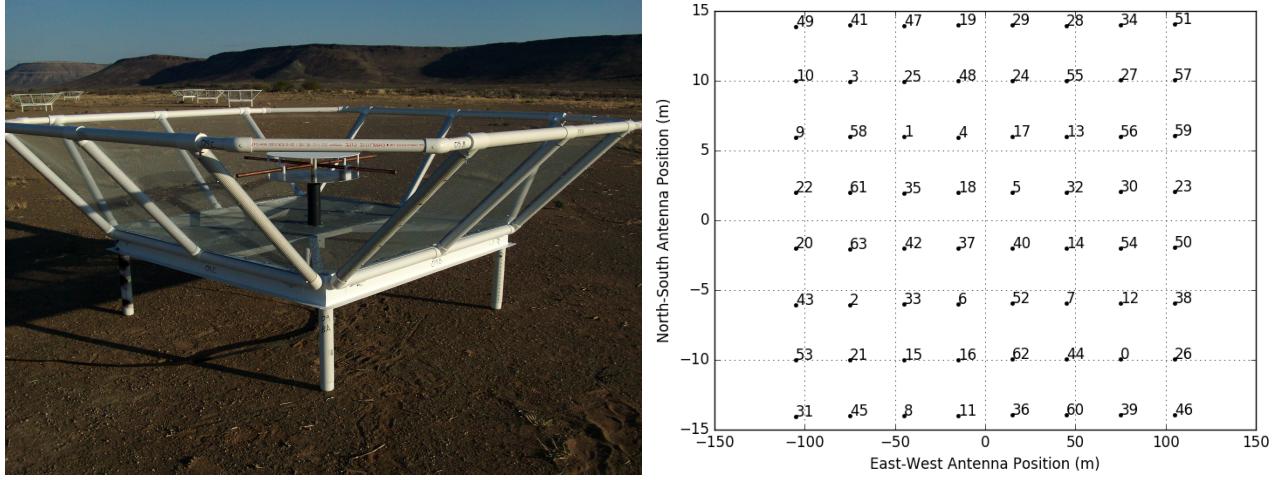


Figure 9. PAPER dipole in South Africa (left) and PAPER-64 antenna layout (right).

ment analysis, Fourier-mode filtering, non-parametric subtractions, and inverse covariance weighting. PAPER uses a delay-spectrum filtering method which was first explained and applied in Parsons et al. (2014). In short, the delay-filtering technique employs the spectrally smooth nature of foregrounds, which are consequently localized in delay-space, the Fourier dual to frequency. We subtract all components that fall within the horizon limit for a specific baseline type in delay-space, plus a 15 ns buffer, thus gaining about ~ 4 orders of magnitude in sensitivity.

After removing an additional layer of RFI (values 3σ above the median), we average together all our data into two datasets: even julian dates and odd julian dates. A total of 124 nights of data comprises this average.

Finally, our last analysis step before power spectrum estimation is the use of a fringe-rate filter. As shown in Parsons et al. (2016), a fringe-rate filter allows us to maximize our sensitivity even further by averaging visibilities in time. We apply this filter in the fringe-rate domain, the Fourier-dual to time, and it effectively amounts to increasing our data's integration time. Another way to describe the filter is that it weights different parts of the sky (which rotate at different "fringe-rates") by the antenna beam. This allows us to up-weight parts of the sky high in the primary beam, and down-weight those that are less sensitive.

New to our updated PAPER-64 analysis is the use of an optimal fringe-rate filter, which maximizes our sensitivity. In Ali et al. (2015), the filter that was applied was degraded by widening it in fringe-rate space. This was chosen in order to decrease the resulting integration time, therefore increasing the number of independent modes and reducing signal loss associated with applying inverse covariance weighting on data with too few modes.

With the development of a robust method for assessing signal loss (see Section 4), we feel comfortable using a narrow filter (Figure 10), which yields an effective integration time of 3857 s (see Section 5.1 for calcula-

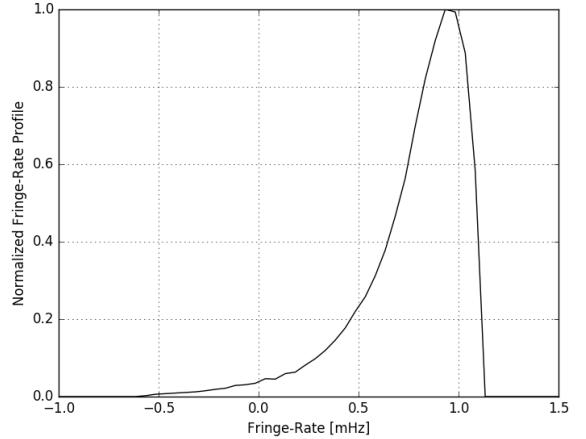


Figure 10. The optimal fringe-rate filter used in the analysis, normalized to 1.

tion) and 8 total independent modes for our 8.5 hours of LST. The filter is computed for a fiducial 30 m baseline at 150 MHz, the center frequency in our band. It is then implemented on a per baseline basis by convolving the time-domain visibilities with the Fourier transform of the fringe-rate filter, resulting in averaged visibilities yielding ~ 2 orders of magnitude in sensitivity.

3.2. Power Spectrum Estimation

To form power spectrum quantities using our two fringe-rate filtered, LST-binned datasets, we use OQE methods as summarized in Section 2.1. Here we will describe its application to PAPER-64 data.

We have a 2-dimensional data vector \mathbf{x} for each baseline and data group ('even' or 'odd'). As part of our bootstrapping routine (and to speed things up), we divide up our total number of baselines into 5 random, equal-sized groups (the way we bootstrap baselines into groups is explained in Section 5.2). Taking \mathbf{x} to be the averaged data from all baselines within a group, we now have 10 possible data vectors (for 5 groups and 2

datasets).

Notice that there are two quantities of \mathbf{wx} in the expression for $\hat{\mathbf{q}}$ in Equation 1, representing two copies of weighted data. We perform all possible cross-multiplications of this quantity, except between the same datasets ('even' with 'even', for example) and same groups ('baseline group 1' with 'baseline group 1', for example). We avoid these computations to prevent the introduction of noise biases from shared auto power.

As mentioned previously, we have a choice for our normalization matrix \mathbf{M} . For our analysis, we choose to compute it by taking the Cholesky decomposition of \mathbf{F} . Namely, we decompose the Fisher matrix such that $\mathbf{F} = \mathbf{LL}^\dagger$, where \mathbf{L} is a lower triangular matrix. Next, we construct $\mathbf{M} = \mathbf{DL}^{-1}$, where \mathbf{D} is a diagonal matrix. Hence, our window function $\mathbf{W} = \mathbf{MF}$ becomes $\mathbf{W} = \mathbf{DL}^\dagger$ and is an upper triangular matrix. This window function was constructed in a way to prevent the leakage of foreground power from low k to high k modes. Specifically, we order the elements in \mathbf{F} in such a way so that power can leak from high k modes to low k modes, but not vice versa. Since most foreground power shows up at low k 's, this method ensures a window function that retains clean, noise-dominated measurements while minimizing foreground bias.

In practice, our final power spectrum result is a 2-dimensional quantity (a function of both time and k). We compute 20 bootstraps of these quantities, where each bootstrap creates baseline groups randomly. We form both the quantity $\mathbf{P}(\mathbf{k})$ and the folded version $\Delta^2(\mathbf{k})$, defined as:

$$\Delta^2(\mathbf{k}) = \frac{k^3}{2\pi^2} \hat{\mathbf{p}}(\mathbf{k}) \quad (5)$$

4. APPLICATION: SIGNAL LOSS

As detailed in Section 2.1, signal loss can arise from the interplay between fringe-rate filtering and inverse covariance weighting. In this section, we explain a methodology that simulates the injection and recovery of a cosmological signal in order to quantify the amount of signal loss. Before doing so however, we first explain how noise and EoR are created in our power spectrum pipeline, both of which are used to verify signal loss.

4.1. Noise and EoR

One critical new component of our power spectrum pipeline is that we have many different power spectrum channels that simultaneously get processed at the same time and are especially useful for computing signal loss and sensitivity estimations. Two important channels, in addition to our PAPER-64 data, are a noise dataset and an EoR dataset (in addition to these, we also look at different combinations of all three). We will briefly describe each.

We create random noise with a variance determined by the sensitivity of our instrument in order to best represent the thermal noise level of our data. We calculate

our system temperature for our frequencies of interest as:

$$T_{sys} = 180 \left(\frac{\nu}{0.18} \right)^{-2.55} + T_{rcvr} \quad (6)$$

where ν are frequencies in GHz. We use a receiver temperature of 200 K, yielding $T_{sys} = 487$ at 150 MHz. This is lower than in [Ali et al. \(2015\)](#) because [why?]. We convert this temperature into a variance statistic using:

$$T_{rms} = \frac{T_{sys}}{\sqrt{\Delta\nu \Delta t N_{days} N_{pol}}} \quad (7)$$

where $\Delta\nu$ is channel spacing, Δt is integration time, N_{days} is the number of data samples for a particular time and frequency that went into our LST binned set, and N_{pol} is the number of polarizations (2 for Stokes I). Our simulated noise has the same shape as our data, and we fringe-rate filter it in the same way to best mimic the real noise floor of our data.

A second additional important channel in our pipeline is a simulated EoR signal. We again simulate random noise (with a default scale of 1) with the same shape as our data, and we fringe-rate filter this signal twice. The first filter transforms the unattached noise into a signal that's attached to the sky (i.e. what our instrument observes). The second filter represents the fringe-rate filtering step in our data analysis pipeline. This fake EoR signal is injected on top of data at various amplitude levels in order to measure its loss.

4.2. Signal Loss Methodology

The loss of a cosmological signal when analyzing PAPER-64 data is a real and significant issue. To reiterate, when applying inverse covariance weighting, \mathbf{C}^{-1} is empirically estimated from the data itself, which has the consequence of over-fitting the noise in the data, producing power spectra values well below the thermal noise limit that is predicted based on observation parameters. This is especially prevalent when weighting fringe-rate-filtered data, which has so few independent time modes to begin with, leading to a noisier dataset. Being able to accurately quantify this loss is crucial in interpreting and providing credibility to any power spectrum limits.

New to the PAPER-64 analysis is a robust method to estimate signal loss. This method is now a standard analysis step for all PAPER analyses and one that will be used for HERA moving forward.

As discussed previously, our power spectrum pipeline runs on a standardized set of channels (pure data, pure noise, pure EoR, and combinations of the three). As part of our signal loss routine, we also compute power spectra with various levels of the created EoR signal, dialing its amplitude from well-below the data level, to well-above. Suppose that \mathbf{e} is the injected EoR (at some amplitude level), and \mathbf{x} is our data vector. We define \mathbf{r}

to be:

$$\mathbf{r} = \mathbf{x} + \mathbf{e} \quad (8)$$

Using OQE formalism, we are interested in the following two quantities: P_{in} and P_{out} . The input power spectrum, P_{in} represents the unweighted power spectrum of only \mathbf{e} , our simulated EoR signal. The output power spectrum, P_{out} , is the weighted power spectrum of \mathbf{e} that would result from our pipeline if the signal was mixed with our data. Comparing the two quantities yields insight into how much of \mathbf{e} is lost due to our choice of weighting. Ignoring normalization, factors:

$$P_{in} \propto \mathbf{e}^\dagger \mathbf{I} \mathbf{Q} \mathbf{I} \mathbf{e} \quad (9)$$

$$\begin{aligned} P_{out} &\equiv \mathbf{P}_e = \mathbf{P}_r - \mathbf{P}_x \\ &\propto \mathbf{r}^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r \mathbf{r} - \mathbf{x}^\dagger \mathbf{w}_x \mathbf{Q} \mathbf{w}_x \mathbf{x} \end{aligned} \quad (10)$$

It is noted that the output power spectrum is comprised of two terms: the power spectrum associated with \mathbf{r} , and that of data \mathbf{x} alone.

One may wonder why P_{out} cannot be computed simply as the weighted power spectrum of \mathbf{e} alone, namely $P_{out} \propto \mathbf{e}^\dagger \mathbf{w}_e \mathbf{Q} \mathbf{w}_e \mathbf{e}$. Expanding Equation 10:

$$\begin{aligned} P_{out} &\propto (\mathbf{x} + \mathbf{e})^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r (\mathbf{x} + \mathbf{e}) - \mathbf{x}^\dagger \mathbf{w}_x \mathbf{Q} \mathbf{w}_x \mathbf{x} \\ &\propto \mathbf{x}^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r \mathbf{x} + \mathbf{e}^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r \mathbf{e} + \mathbf{x}^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r \mathbf{e} \\ &\quad + \mathbf{e}^\dagger \mathbf{w}_r \mathbf{Q} \mathbf{w}_r \mathbf{x} - \mathbf{x}^\dagger \mathbf{w}_x \mathbf{Q} \mathbf{w}_x \mathbf{x} \end{aligned}$$

And taking the case of very large \mathbf{e} , so that $\mathbf{w}_r \sim \mathbf{w}_e$ and any terms involving only \mathbf{x} are small:

$$\begin{aligned} P_{out, e \gg x} &\propto \mathbf{e}^\dagger \mathbf{w}_e \mathbf{Q} \mathbf{w}_e \mathbf{e} + \mathbf{x}^\dagger \mathbf{w}_e \mathbf{Q} \mathbf{w}_e \mathbf{e} \\ &\quad + \mathbf{e}^\dagger \mathbf{w}_e \mathbf{Q} \mathbf{w}_e \mathbf{x} \end{aligned} \quad (11)$$

We see that our naive expression for P_{out} is the first term, but there also two additional terms. An initial assumption would be that the cross-terms that involve both \mathbf{e} and \mathbf{x} should be zero, since the two quantities are un-correlated. However, [need explanation about power in cross-terms here]. Therefore, in our investigation of signal loss, we use the full quantity for P_{out} as in Equation 10.

For the unweighted case ($\mathbf{w} \equiv \mathbf{I}$), we expect P_{out} and P_{in} to be equal, and hence the ratio of P_{in}/P_{out} to be 1. For the weighted case, this is not true due to signal loss. In order to quantify the loss, we look at the ratio of P_{in} to P_{out} as the amplitude level of the injected signal \mathbf{e} is increased. In the next section, we highlight two methods that yield similar results for the determination of signal loss using P_{in} and P_{out} .

4.3. Signal Loss in Practice

We now shift our attention towards computing signal loss for fringe-rate filtered PAPER-64 dataset, using inverse covariance weighting ($\mathbf{w} = \mathbf{C}^{-1}$). Our expressions for P_{in} and P_{out} become:

$$P_{in} \propto \mathbf{e}^\dagger \mathbf{I} \mathbf{Q} \mathbf{I} \mathbf{e} \quad (12)$$

$$P_{out} \propto \mathbf{r}^\dagger \mathbf{C}_r^{-1} \mathbf{Q} \mathbf{C}_r^{-1} \mathbf{r} - \mathbf{x}^\dagger \mathbf{C}_x^{-1} \mathbf{Q} \mathbf{C}_x^{-1} \mathbf{x} \quad (13)$$

Using the input and output power spectra for range of EoR amplitudes, we use two methods to determine signal loss associated with the over-fitting of noise during inverse covariance weighting.

[Determine whether we want to show pure noise plots, or just data, or both!!!]

The first method is very straightforward. Post-bootstrapping, our final power spectra are 1-dimensional and only a function of k . For each k , we simply look at P_{out} as a function of P_{in} , as shown in Figure 11. The shape of this function can be explained as follows. At small injection levels (small \mathbf{e}), P_{out} and P_{in} are equal, and there is no signal loss. As the amplitude of EoR increases, we then move into a regime where the final output power spectrum is lower than the unweighted input one. This is dangerous, because without correcting for this effect one might be led to underestimate the EoR signal. The peculiar tail at very low injection levels (where $P_{out} > P_{in}$) is an unphysical feature, but rather illustrates that there is some non-negligible cross-term power between \mathbf{n} and \mathbf{e} .

For this method, we interpolate the signal loss factor (per k), computed as P_{in}/P_{out} , at a P_{out} value equal to the 2σ power spectrum upper limit of noise alone. In other words, we look at $P_{noise} \propto \mathbf{n}^\dagger \mathbf{C}_n^{-1} \mathbf{Q} \mathbf{C}_n^{-1} \mathbf{n}$, compute its 2σ upper limit (mean over bootstraps + $2\times$ standard deviation over bootstraps), and interpolate the value of P_{in}/P_{out} at this value. We therefore end up with one signal loss correction factor per k .

Figure 12 shows the power spectrum of our noise simulation, using full inverse covariance weighting, both before and after signal loss correction. Prior to signal loss correction, it is obvious that the power spectrum is unfeasible because it is well below the theoretical noise level prediction. Post-correction, the power spectrum values blow up much higher than both the theory and unweighted power spectrum. This is an effect caused by the steep nature of the eigenspectrum of \mathbf{C} , and is explained more in Section 4.4.

[Need a plot that shows our signal loss factors are CORRECT. How to do that??]

Our second method for estimating signal loss is similar to the first, but more comprehensive in a statistical sense. Instead of looking at input and output power spectra after bootstrapping, we now look at their values for every bootstrap in order to get a sense of their distributions. Figure 13 plots P_{in} vs. P_{out} for 20 bootstraps, and as expected, the function now has a spread in the width-direction in comparison to what was plotted in Figure 11, but otherwise shows a familiar trend. Similarly, our weighted noise power spectra also has a defined spread due to bootstrapping.

Using these two distributions (P_{in}/P_{out} and P_{noise}), we can create bins along the P_{noise} axis to yield his-

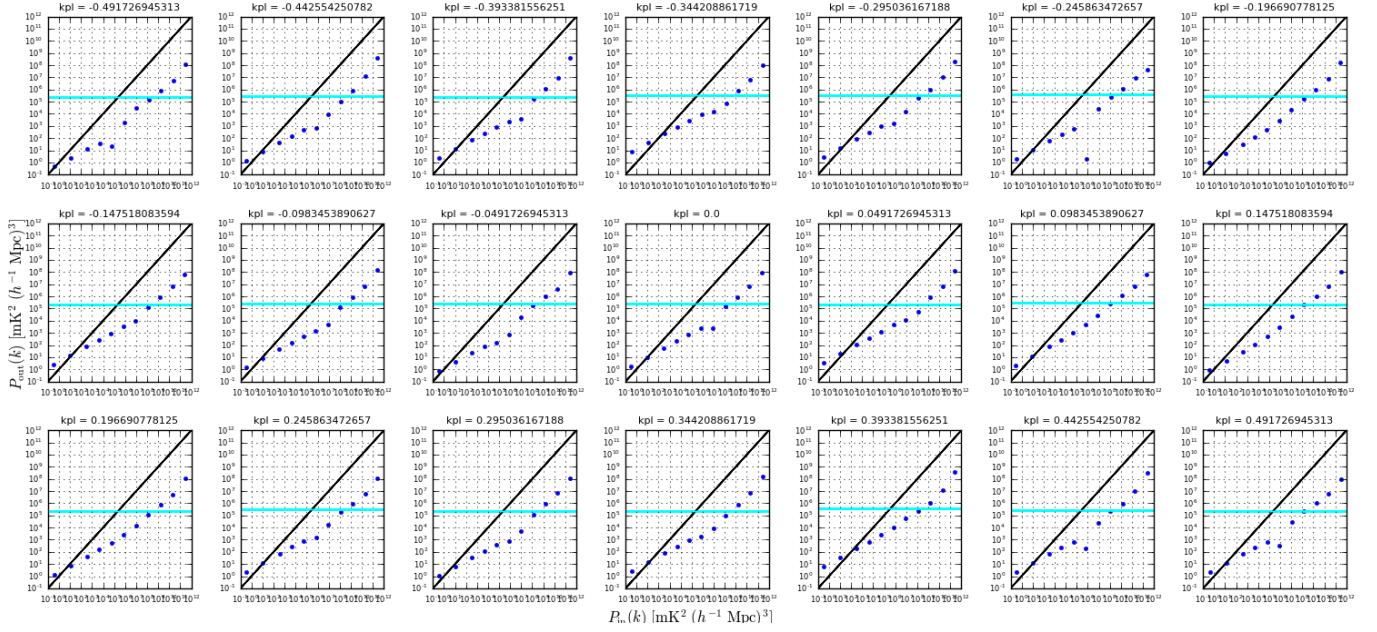


Figure 11. P_{in} vs. P_{out} (blue points) for 15 injection levels and 21 k 's. The solid cyan line is the 2σ upper limit for the weighted power spectrum of noise alone, and it is at this level where the signal loss factor P_{in}/P_{out} is computed by interpolation. [Maybe re-do this plot with closer-together points]

tograms of signal loss factors for each bin. We similarly sort the values of P_{noise} into the same bins, and multiply the probability of P_{noise} per bin (the number of values falling into that bin, divided by the total) with the signal loss factors in that bin, essentially computing a weighted average across all bins to obtain a final signal loss factor per k . As shown in Figure 14, the results are very similar to the previous method. For future power spectrum results, we choose to use the second method because it computes final signal loss values using our full distributions of measurements.

One thing to note is that for both methods, we have been careful to validate that the computations yield no signal loss (signal loss factors of 1) for the unweighted power spectrum case, as is expected. This is important in confirming that signal loss is a direct result of the choice for \mathbf{C} .

4.4. Data Weighting

With our signal loss formalism established, we now have the capability of experimenting with different weighting options for \mathbf{w} . Our goal here is to choose a weighting method that successfully down-weights foregrounds and systematics in our data without generating large amounts of signal loss. We have found that the balance between the two is a delicate one and requires a careful understanding and altering of covariance matrices.

We now turn our attention to power spectra using the 30 m East/West baselines of PAPER-64. Our dataset spans 8.5 hours of LST (.1-8.6 hrs), includes a total of 51 baselines, and is fringe-rate filtered using an optimal fringe-rate filter. We have two datasets (even days and

odd days), and only cross-multiply data from different days and different baselines. We are interested in 21 frequency channels (channels 95-115), which yields a power spectrum for a redshift of $z = 8.4$.

Using full inverse covariance weighting, our results are not too dissimilar to that of pure noise. Signal loss factors (Figure 15) are of similar order of magnitude, and our power spectrum blows up past the unweighted version after signal loss correction (Figure 16).

Looking into this behavior in more detail, we investigate the shape of the eigenspectrum of \mathbf{C} for a typical baseline used in the analysis. Figure 17 shows this spectrum for baseline (1,4). Most obviously, the spectrum is steep, spanning 4 orders of magnitude. Not as obvious is the effect of this shape on our results. When the matrix \mathbf{C} is inverted to form \mathbf{C}^{-1} , the effect of the steepness of the eigenspectrum is to up-weight very few modes of the sky while the rest are drastically down-weighted. More specifically, our fringe-rate filtered data contains a finite, small number of independent modes, thereby resulting in a covariance matrix that can be described by just a few modes. Beyond the first few modes, the eigenvalues of each additional mode falls off dramatically. When inverting, we end up not only down-weighting those initial modes but severely up-weighting a few insignificant ones. Because of our weighting choice, signal loss blows up as it thinks we only have a couple modes in our data.

Clearly the full inverse covariance treatment of our data is suboptimal to even the unweighted case, but we would like to find a weighting method that does successfully down-weight contaminants in our data and make some improvement over the unweighted power spectrum. There are many choices for determining the covariance

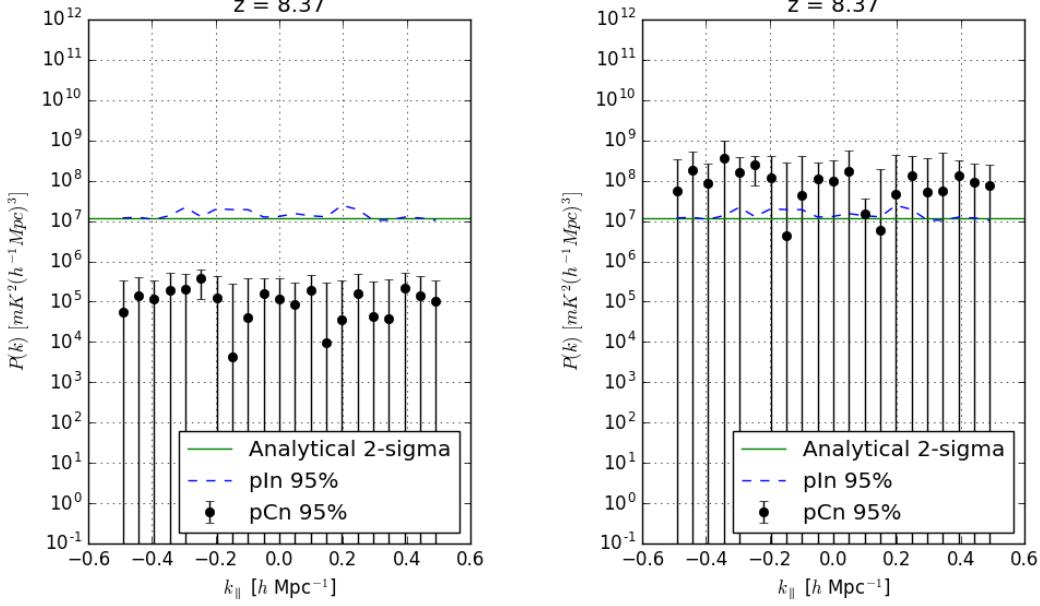


Figure 12. Full inverse covariance weighted power spectrum of pure noise (black points, with 2σ error bars) before signal loss correction (left) and after (right). The dashed blue line is the unweighted power spectrum (2σ upper limit). The solid green line is the theoretical noise level prediction based on observational parameters. [Color negative points grey]

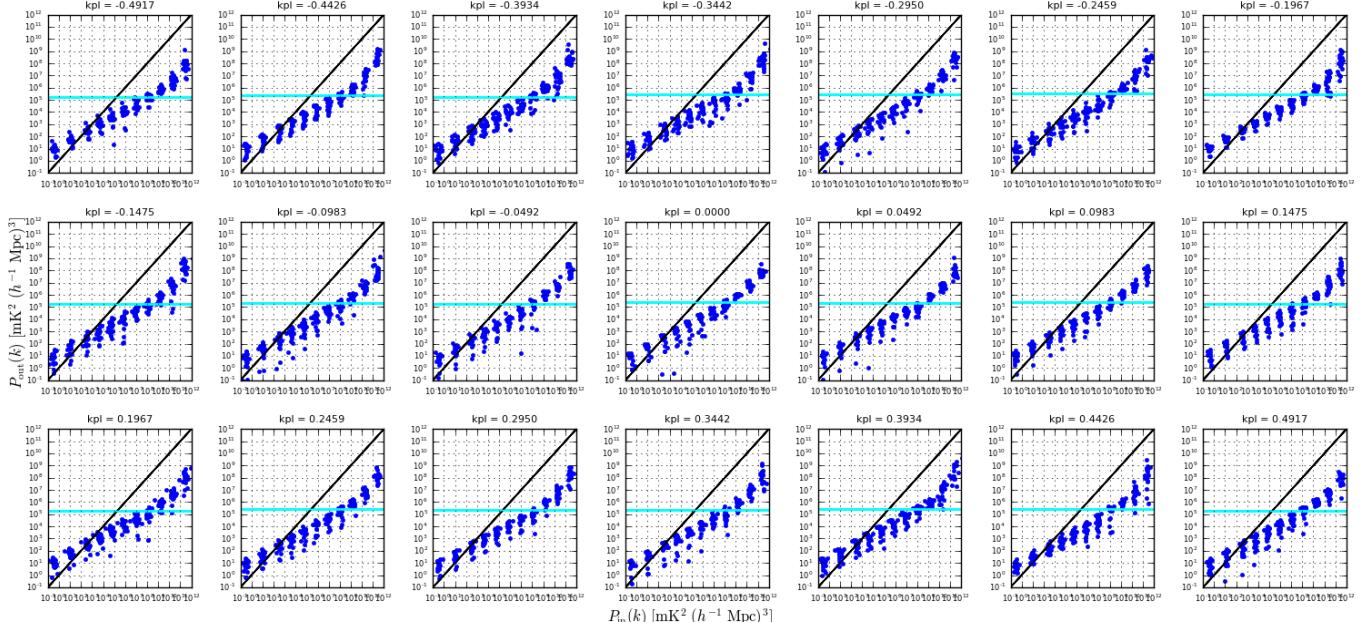


Figure 13. P_{in} vs. P_{out} (blue) for 15 injection levels, 20 bootstraps, and 21 k 's. [Plot a semi-transparent cyan range of pCn values instead of just the max]

matrix \mathbf{C} , but here we will illustrate [?] promising ones as applied to PAPER-64.

[TO DO: decide on/explain/show different weightings]

5. APPLICATION: ERROR ESTIMATION

5.1. Thermal Noise

One way to estimate errors on a 21 cm power spectrum estimate is to compute theoretical errors based on the instrumental temperature and observational parameters. A theoretical noise limit provides a useful cross-check for bootstrapped errors and is a helpful predictor of the maximum sensitivity that can be achieved by a particular observation and analysis. Here we walk through a detailed computation of a theoretical noise estimation as

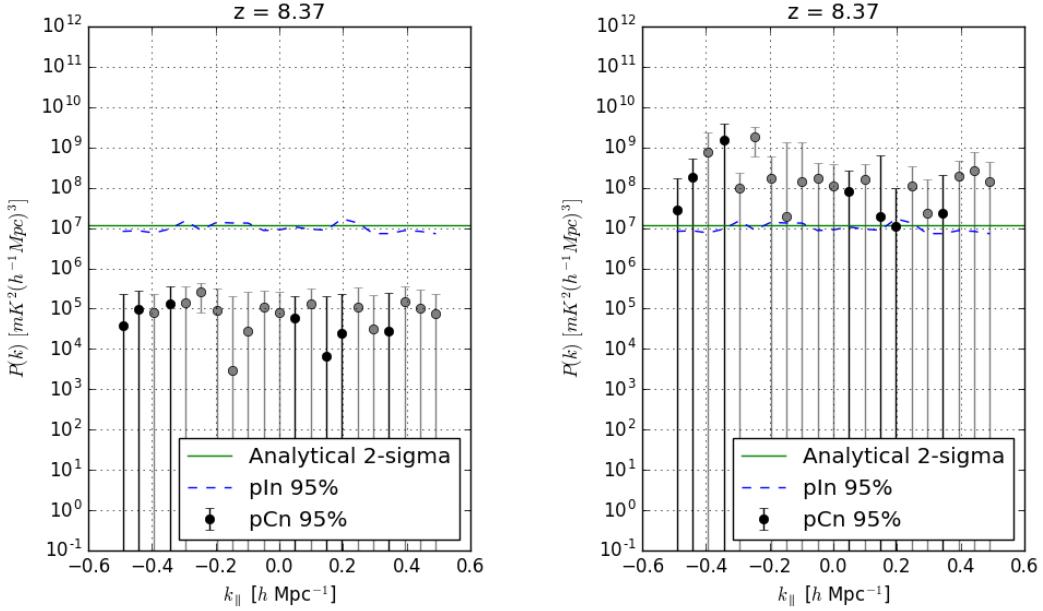


Figure 14. Full inverse covariance weighted power spectrum of pure noise (black and grey points, with 2σ error bars) before signal loss correction (left) and after (right). Black points correspond to positive values, while grey points correspond to originally negative values that have been made positive for plotting. The dashed blue line is the unweighted power spectrum (2σ upper limit). The solid green line is the theoretical noise level prediction based on observational parameters.

applied to PAPER-64 observations and highlight major changes from [Ali et al. \(2015\)](#). We first give a general formula, and then go into detail about each factor.

The sensitivity prediction for a power spectral analysis of interferometric 21 cm data, in temperature-units, is:

$$p(k) = \frac{X^2 Y \Omega_{eff} T_{sys}^2}{\sqrt{2N_{lst}s N_{seps}} t_{int} N_{days} N_{bls} N_{pol}} \quad (14)$$

- X^2Y : Conversion factors from observing coordinates (angles on the sky) to cosmological coordinates (co-moving distances). For $z = 8.4$, $X^2Y = 5 \times 10^{11} h^{-3} Mpc^3 str^{-1} GHz^{-1}$.
- Ω_{eff} : The primary beam angular size. The effective beam area changes with the application of a fringe-rate filter, since parts of the beam are up-weighted and down-weighted. Using numbers from Table 1 in [Parsons et al. \(2016\)](#), $\Omega_{eff} = 0.74^2 / 0.24$ for an optimal fringe-rate filter.
- T_{sys} : The system temperature is set by Equation 6, for a receiver temperature of 200 K and center frequency of 0.15 GHz. **[does this differ from prior analysis?]**
- $\sqrt{2}$: This factor in the denominator of the sensitivity equation comes from only using the real part of power spectrum estimates when plotting and quoting upper limits (imaginary component is discarded).

- $N_{lst}s$: The number of LST hours that go into a power spectrum estimation. The sensitivity scales as the square root because we integrate incoherently over time. For PAPER-64, $N_{lst}s = 8$ hours.

- N_{seps} : The number of baseline separation types averaged incoherently in a final power spectrum estimate. For the analysis in this paper, we only use one type of baseline, hence $N_{seps} = 1$.

- t_{int} : The integration time of the data. It is crucial to adapt this number if filtering is applied along the time axis (i.e. a fringe-rate filter). We compute the effective integration time of our fringe-rate filtered data by scaling our original integration time using the following:

$$t_{frf} = t_{int} \frac{\int 1 df}{\int w^2(f) df}, \quad (15)$$

where $t_{int} = 43$ seconds, t_{frf} is our fringe-rate filtered integration time, w is our fringe-rate profile, and the integral is taken over all fringe-rates. In other words, we calculate our effective integration time by computing how our beam is altered in fringe-rate space. For PAPER-64, this number is $t_{int} = 3857$ s.

- N_{days} : The total number of days of data analyzed. In [Ali et al. \(2015\)](#), this number was set to 135. However, because we divide our data in half (to form ‘even’ and ‘odd’ datasets), this number should be reduced by a factor of 2. Additionally, because our *LST* coverage is not 100% complete

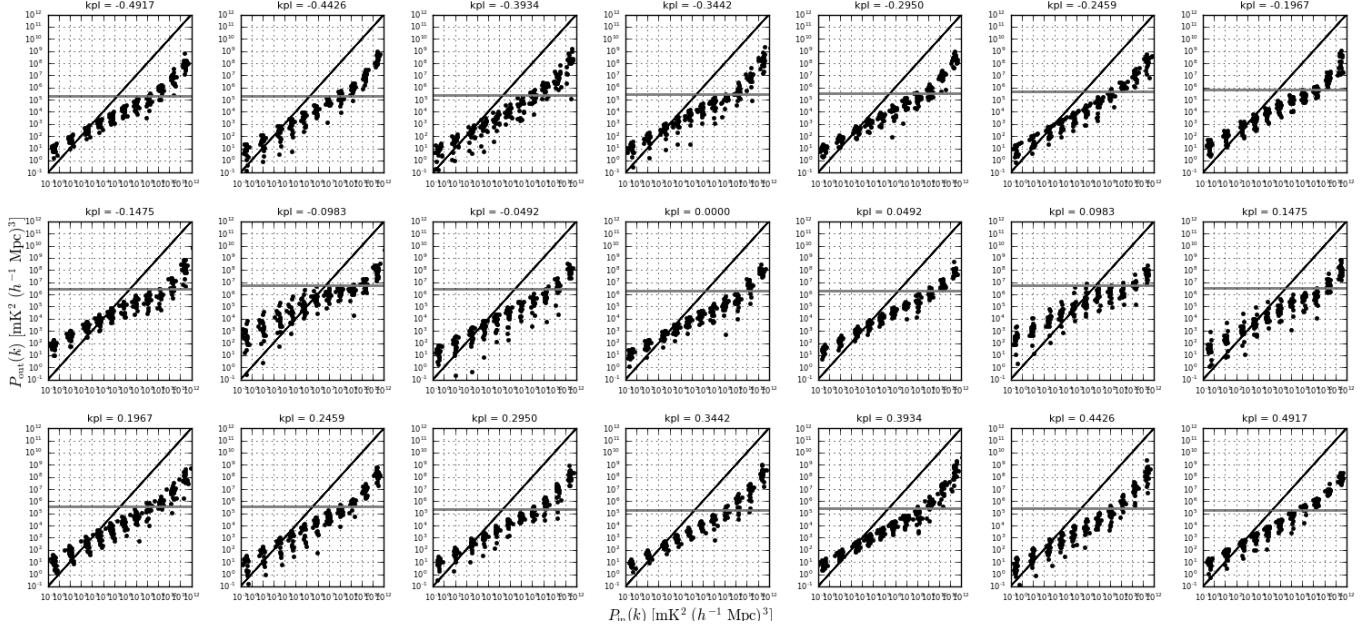


Figure 15. P_{in} vs. P_{out} (black) for 15 injection levels, 20 bootstraps, and 21 k 's. [Plot a semi-transparent grey range of pCv values instead of just the max]

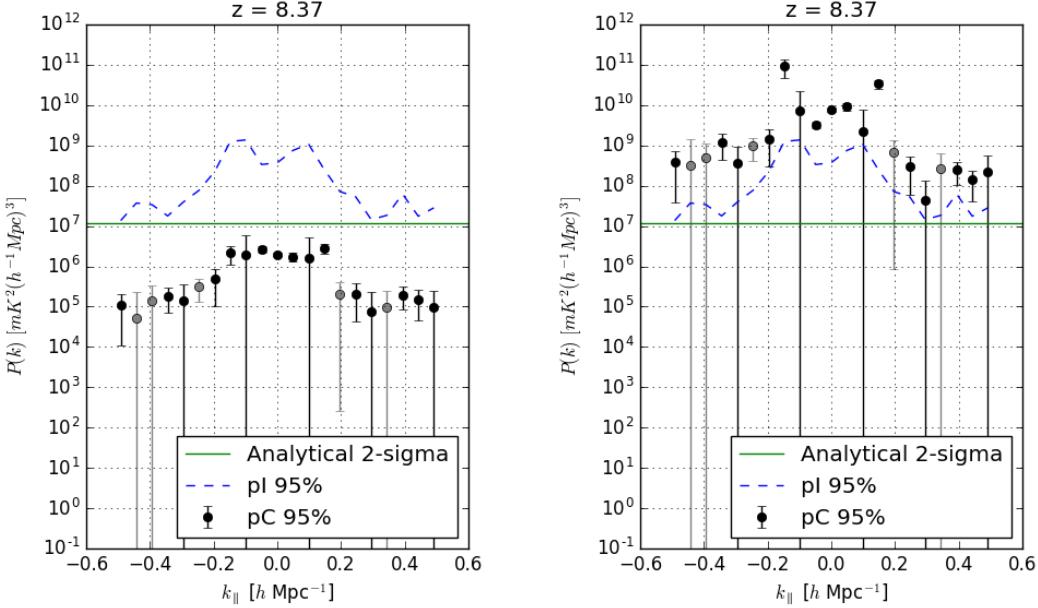


Figure 16. Full inverse covariance weighted power spectrum of PAPER-64 data (black and grey points, with 2σ error bars) before signal loss correction (left) and after (right). Black points correspond to positive values, while grey points correspond to originally negative values that have been made positive for plotting. The dashed blue line is the unweighted power spectrum (2σ upper limit). The solid green line is the theoretical noise level prediction based on observational parameters.

(it doesn't overlap for every single day), we compute a realistic value of N_{days} as the RMS of all the daily counts in one dataset ('even', for example). For PAPER-64, our revised estimate of N_{days} is ~ 34 days.

- N_{bls} : The number of baselines contributing to the sensitivity of a power spectrum estimate. In Ali

et al. (2015), this number was the total number of 30 m East/West baselines used in the analysis. However, using the total number of baselines neglects the fact that we divid baselines into 5 groups before cross-multiplying data. Our revised estimate for the parameter is $N_{bls} = \frac{N_{bls}}{N_{gps}} \sqrt{N_{gps}^2 - N_{gps}}$, where $N_{gps} = 5$. This expres-

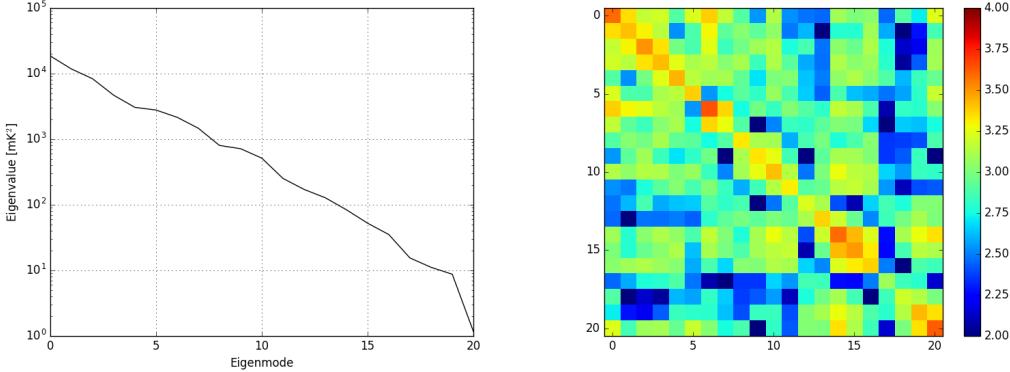


Figure 17. Eigenspectrum for \mathbf{C} for baseline (1,4) for the 21 channels of interest (left) and covariance matrix \mathbf{C} for the same baseline (right).

sion arises because N_{bls} must be scaled by the total number of cross-multiplications going into the power spectrum estimate. For 5 groups, there are 25 total cross-products, minus 5 cross-products since we avoid multiplying the same group with itself. For our PAPER-64 analysis with only one baseline separation type, this becomes $N_{bls} \sim 46$ baselines.

- N_{pol} : The number of polarizations averaged together. For the case of Stokes I, $N_{pol} = 2$.

5.2. Bootstrapping

In this section, we highlight two major changes since [Ali et al. \(2015\)](#), using the lessons we've learned about sampling with replacement and bootstrapping independent samples.

As discussed in Section 2.2, traditional bootstrapping theory is only a valid way of estimating errors if a dataset is comprised of independent samples. Recall that the PAPER-64 pipeline outputs 20 bootstraps (over baselines), each a 2-dimensional power spectrum that is a function of k and time. In [Ali et al. \(2015\)](#), a second round of bootstrapping occurs over the baseline bootstrap and time axes simultaneously. Random values are sampled with replacement along both axes, drawing as many values as there are number of bootstraps and times. Final power spectrum limits are then computed by taking the mean and standard deviation over this second bootstrap axis (which is typically large, on the order of $N_{boot} \sim 400$).

Because our dataset is fringe-rate filtered, we have substantially increased our integration time, resulting in fewer independent samples (from ~ 700 samples to 8 samples). A random draw of 700 samples from this dataset therefore has many repeated samples, and the variance between hundreds (N_{boot}) of these random samples is smaller than the true underlying variance. Another way of describing this underestimation is that there are fewer ways to randomly sample data with 8 independent modes versus 700, so bootstrapping a fringe-rate filtered dataset yields a narrower distribution of val-

ues from which we calculate its standard deviation.

To prevent error underestimation, we simply take the average along the time axis in our revised pipeline rather than bootstrapping. Hence, our final power spectrum errors are calculated from the standard deviation over the baseline bootstrapping axis only, which is still a valid axis since each baseline makes an independent measurement of the sky.

The second aspect of bootstrapping discussed in Section 2.2 is in regard to how a dataset is randomly sampled. In order to maximize sensitivity (i.e. maximize independent samples in a bootstrap), we have revised the method by which we sample baselines when bootstrapping.

First, we speed things up in our pipeline by dividing our total number of baselines into 5 groups, where there are no repeated baselines within or between groups. In [Ali et al. \(2015\)](#), each group is then sampled with replacement to create a new group of the same size, which can have repeated baselines inside it. We discover that in doing so, we are sacrificing some of our sensitivity since there are 3-4 repeated baselines per group. In order to maximize our sensitivity but still apply random sampling for use in error estimation, we instead form new groups using all independent baselines except the very last one, which we fill randomly. We use this method for every bootstrap, creating new groups and sampling the last spot randomly each time.

Power spectrum estimates for PAPER-64 fringe-rate filtered data are shown in Figure 18. The estimates use a weighting matrix of \mathbf{C}_{eff}^{-1} , where $\mathbf{C}_{eff} = \mathbf{C} * \mathbf{I}$, in order to minimize signal loss. We show upper limits (2σ error bars) for three different results. Our revised error estimation method is in black, where we only bootstrap along the baseline axis and we maximize sensitivity by random sampling only the last baseline in each group. The red curve uses a similar method except all baselines are randomly sampled, leading to a higher power spectrum estimate because some baselines are repeated per group. In blue, we bootstrap along both the baseline and time axes, drawing as many samples as there are in

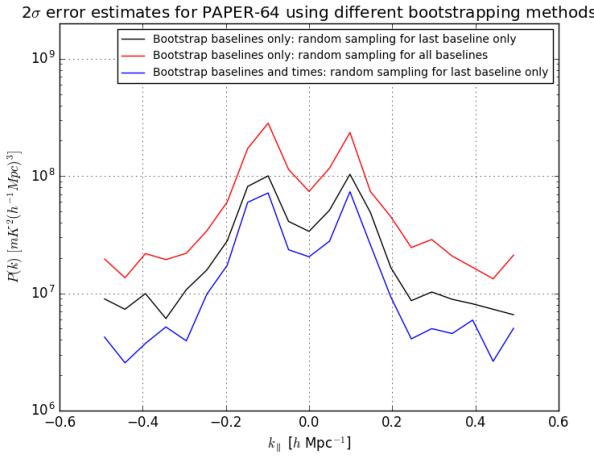


Figure 18. 2σ upper limit power spectrum estimates using PAPER-64 data and three different bootstrapping methods. The data is fringe-rate filtered and a weighting matrix of \mathbf{C}_{eff}^{-1} is used, where $\mathbf{C}_{eff} = \mathbf{C} * \mathbf{I}$, in order to minimize signal loss. Three different bootstrapping methods are shown: the black and red use methods where bootstrapping occurs on the baseline axis only (not the time axis), and they differ by how the baselines are randomly sampled. It is evident that sensitivity is maximized when baselines are sampled in a way that ensures mostly independent samples (only the last baseline spot is sampled randomly). The blue result bootstraps over both the baseline and time axes, and illustrates how errors can be under-estimated if sampling more values than independent ones (fringe-rate filtering reduces the number of independent samples).

the data. We under-estimate errors in doing so because we are drawing many more samples than independent ones since the data is fringe-rate filtered.

6. APPLICATION: BIAS

7. CONCLUSION

8. ACKNOWLEDGEMENTS

[NSF Graduate Research Fellowship Program (GRFP) Fellowship] [UC Berkeley Chancellor's Fellowship]

REFERENCES

- Ali, Z. S., et al. 2015, ApJ, 809, 61
 Bowman, J. D., & Rogers, A. E. E. 2010, Nature, 468, 796
 Burns, J. O., et al. 2012, Advances in Space Research, 49, 433
 DeBoer, D. R., et al. 2017, Publications of the Astronomical Society of the Pacific, 129, 045001
 Ellingson, S. W., Clarke, T. E., Cohen, A., Craig, J., Kassim, N. E., Pihlstrom, Y., Rickard, L. J., & Taylor, G. B. 2009, IEEE Proceedings, 97, 1421
 Greenhill, L. J., & Bernardi, G. 2012, ArXiv e-prints
 Keating, G. K., Marrone, D. P., Bower, G. C., Leitch, E., Carlstrom, J. E., & DeBoer, D. R. 2016, The Astrophysical Journal, 830, 34
 Liu, A., Parsons, A. R., & Trott, C. M. 2014, PhRvD, 90, 023019
 Moore, D. F., Aguirre, J. E., Parsons, A. R., Jacobs, D. C., & Poher, J. C. 2013, The Astrophysical Journal, 769, 154
 Paciga, G., et al. 2013, MNRAS
- Parsons, A. R., Liu, A., Ali, Z. S., & Cheng, C. 2016, ApJ, 820, 51
 Parsons, A. R., et al. 2010, AJ, 139, 1468
 —. 2014, ApJ, 788, 106
 Patra, N., Subrahmanyam, R., Sethi, S., Udaya Shankar, N., & Raghunathan, A. 2015, ApJ, 801, 138
 Peterson, U.-L. P. X.-P. W. J. 2004, ArXiv Astrophysics e-prints
 Poher, J. C., et al. 2014, ApJ, 782, 66
 Sokolowski, M., et al. 2015, PASA, 32, e004
 Tingay, S. J., et al. 2013, PASA, 30, 7
 van Haarlem, M. P., et al. 2013, A&A, 556, A2
 Voytek, T. C., Natarajan, A., Jáuregui García, J. M., Peterson, J. B., & López-Cruz, O. 2014, ApJL, 782, L9
 Wu, X. 2009, in Bulletin of the American Astronomical Society, Vol. 41, American Astronomical Society Meeting Abstracts #213, 474
 Zheng, H., et al. 2014, MNRAS, 445, 1084