

# Gaussian Process Regression for foreground removal in HI Intensity Mapping experiments

Paula S. Soares <sup>1</sup>★, Catherine A. Watkinson,<sup>1</sup> Steven Cunnington <sup>1</sup> and Alkistis Pourtsidou <sup>1,2</sup>

<sup>1</sup>*School of Physics and Astronomy, Queen Mary University of London, Mile End Road, London E1 4NS, UK*

<sup>2</sup>*Department of Physics and Astronomy, University of the Western Cape, Cape Town 7535, South Africa*

Accepted 2021 September 9. Received 2021 September 9; in original form 2021 June 3

## ABSTRACT

We apply for the first time Gaussian Process Regression (GPR) as a foreground removal technique in the context of single-dish, low redshift HI intensity mapping, and present an open-source PYTHON toolkit for doing so. We use MeerKAT and SKA1-MID-like simulations of 21 cm foregrounds (including polarization leakage), HI cosmological signal, and instrumental noise. We find that it is possible to use GPR as a foreground removal technique in this context, and that it is better suited in some cases to recover the HI power spectrum than principal component analysis (PCA), especially on small scales. GPR is especially good at recovering the radial power spectrum, outperforming PCA when considering the full bandwidth of our data. Both methods are worse at recovering the transverse power spectrum, since they rely on frequency-only covariance information. When halving our data along frequency, we find that GPR performs better in the low-frequency range, where foregrounds are brighter. It performs worse than PCA when frequency channels are missing, to emulate RFI flagging. We conclude that GPR is an excellent foreground removal option for the case of single-dish, low-redshift HI intensity mapping in the absence of missing frequency channels. Our PYTHON toolkit GPR4IM and the data used in this analysis are publicly available on GitHub.

**Key words:** methods: data analysis – cosmology: observations – large-scale structure of Universe – radio lines: general.

## 1 INTRODUCTION

Neutral hydrogen (HI) intensity mapping (IM) is a novel probe able to trace the large-scale structure (LSS) of the Universe using the 21cm hyperfine transition of hydrogen (Bharadwaj, Nath & Sethi 2001; Battye, Davies & Weller 2004; Chang et al. 2008). After reionization, most of the neutral hydrogen in the interstellar medium became ionized, and most of the remaining neutral hydrogen can be found inside of self-shielding galaxies. Hence, HI is a good probe for the distribution of matter in the Universe. Instead of using HI to trace individual galaxies, IM uses the combined emission of HI from numerous galaxies to probe how much structure is in a given area of the sky. As such, it can map large areas of the sky very quickly, posing an advantage over traditional spectroscopic galaxy surveys. See Kovetz et al. (2017) and Liu & Shaw (2020) for reviews on the subject.

For single-dish IM (Battye et al. 2013), several telescope dishes can be used in autocorrelation mode, probing large areas of the sky very quickly (Bull et al. 2015; Santos et al. 2017; Wang et al. 2020). While fast, single-dish IM suffers from low angular resolution, since any structure smaller than the angular resolution of the telescope will not be resolved. This has the effect of damping small-scale structure. Despite this, studies have found that it is possible to probe cosmological parameters to significant precision with future HI IM single-dish surveys (e.g. Chang et al. 2008; Peterson et al. 2009; Seo et al. 2010; Ansari et al. 2012; Battye et al. 2013; Bull et al. 2015;

Villaescusa-Navarro, Alonso & Viel 2016; Pourtsidou, Bacon & Crittenden 2017; Kennedy & Bull 2021; Soares et al. 2021).

In order to access the cosmological HI IM signal, systematic, and instrumental effects must be addressed, of which astrophysical foregrounds pose a particular problem. These foregrounds emit in the same radio frequencies at which we observe the HI signal, and are several orders of magnitude brighter. However, they vary slowly in frequency (i.e. their spectrum looks very smooth), while the HI signal is tracing the large-scale structure of the Universe and so is not smooth in frequency.

We can exploit the spectral smoothness of the foregrounds to separate them from the cosmological signal (Chang et al. 2010; Liu & Tegmark 2011; Alonso et al. 2014b; Bigot-Sazy et al. 2015; Olivari, Remazeilles & Dickinson 2015; Switzer et al. 2015; Wolz et al. 2015). Blind foreground removal methods such as principal component analysis (PCA) do not require previous knowledge of what the foregrounds look like, using the fact that they should be statistically separable from our HI signal (for a study into some of the main foreground removal methods for HI IM, see Cunnington et al. 2021). However, it is difficult to avoid also removing some of the HI signal in the process, leading to biased cosmological parameter estimates (Wolz et al. 2015; Cunnington, Camera & Pourtsidou 2020b; Soares et al. 2021).

In reality, foreground signals are not perfectly smooth in frequency, since telescope effects such as polarization leakage can introduce structure that is difficult to mitigate, posing further complications to foreground removal (Jelic et al. 2008, 2010; Moore et al. 2013; Liao et al. 2016; Carucci, Irfan & Bobin 2020). While it is possible to try to simulate polarization leakage effects, it is important to keep in mind

\* E-mail: [p.s.soares@qmul.ac.uk](mailto:p.s.soares@qmul.ac.uk)

that, in the absence of end-to-end simulations, current foreground simulations are considered overly idealized. Real data analyses of HI IM experiments conduct much more aggressive foreground cleaning than simulations require. In fact, an autocorrelation HI IM detection is yet to be observed, with current detections relying on cross-correlation with optical galaxy surveys. Since optical galaxy surveys have different systematic effects than HI IM surveys, these drop out in cross-correlation, making a detection achievable (Masui et al. 2013; Switzer et al. 2013; Wolz et al. 2016; Anderson et al. 2018; Wolz et al. 2021).

In the realm of higher redshift ( $z > 6$ ), interferometric HI experiments, the aim is to probe the Epoch of Reionization (EoR) and Cosmic Dawn, learning about the first stars and the Universe before reionization. While some instrumental and systematic effects differ between this and our low redshift, single-dish HI IM case, both are subject to astrophysical foreground contamination. Gaussian Process Regression (GPR; Rasmussen & Williams 2006) has been used as a foreground removal technique in the context of EoR (Mertens, Ghosh & Koopmans 2018; hereafter M18), and has been found to work well for both simulations and real data (Gehlot et al. 2019; Offringa, Mertens & Koopmans 2019; Ghosh et al. 2020; Hothi et al. 2020; Kern & Liu 2020; Mertens et al. 2020). GPR, like other foreground removal methods such as PCA, can also lead to HI signal loss. Recent studies have looked at how to correct for this signal loss within GPR (Kern & Liu 2020; Mertens et al. 2020), and in this paper we will consider the method proposed in Mertens et al. (2020).

In this paper, we aim to apply GPR for the first time as a foreground removal technique in the context of single-dish, low redshift HI IM. Our analysis follows on from Cunningham et al. (2021) (hereafter C21), which used MeerKAT and SKA1-MID-like HI IM simulations, including HI cosmological signal, smooth foregrounds, polarized foregrounds, and instrumental noise, to thoroughly investigate and compare different foreground removal techniques. We use these simulations to determine how GPR would perform in our case, and compare it to the widely used foreground removal technique PCA.

We aim to investigate first if it is possible to perform GPR as a foreground removal technique in our case, and if yes, we aim to explore the following questions: How well does it perform compared to PCA? What are its advantages? What are its limitations? How does it perform in the context of different instrumental and systematic effects? What should we keep in mind, and be careful about, if we want to use it in the context of real data analyses?

We base our work on the publicly available code `PS_EOR`<sup>1</sup> (M18), which outlines how to perform GPR in the context of EoR observations. Using it as a great starting point, we created the publicly available `PYTHON` package `GPR4IM`<sup>2</sup>, a user-friendly code that can be used to perform GPR as a foreground removal technique for any real space HI intensity map, either simulated or real. The URL in the caption of each figure shows how the figure was produced.

The paper is organized as follows. We introduce our simulations in Section 2. We explain how GPR and PCA work to remove foregrounds in Section 3. In Section 4, we show power spectrum results of the recovered HI power spectrum using GPR and PCA for the different cases considered. We conclude in Section 5.

## 2 SIMULATIONS

We make use of the HI IM simulations presented in C21, choosing the ones centred on the Stripe82 field, a popular target area for galaxy surveys. These simulations include the smooth foreground signal, the polarized foreground signal, the HI cosmological signal and instrumental noise. The polarization leakage is simulated using the software `CRIME`<sup>3</sup>, and the smooth foregrounds are extrapolated from observations, which contain sky structure. For more details on the simulations, see the Appendix and C21. They are an idealized representation of what a future HI IM survey will look like, such as the proposed MeerKLASS survey (Santos et al. 2017). We assume throughout that our sky area is small enough that we can use a Cartesian grid projection without much distortion. This flat-sky approximation is widely used in large-scale structure surveys, where curved sky effects are not a major limitation (Blake, Carter & Koda 2018; Castorina & White 2018).

Each frequency channel ( $\nu$ ) of our simulation is binned into pixels ( $\theta$ ), therefore the total observed temperature fluctuations in our data can be described as

$$\delta T_{\text{obs}}(\nu, \theta) = \delta T_{\text{FG}}(\nu, \theta) + \delta T_{\text{HI}}(\nu, \theta) + \delta T_{\text{noise}}(\nu, \theta). \quad (1)$$

The foreground signal is itself composed of the Galactic synchrotron emission, free-free emission, extragalactic point sources, and polarization leakage:

$$\delta T_{\text{FG}} = \delta T_{\text{sync}} + \delta T_{\text{free}} + \delta T_{\text{point}} + \delta T_{\text{pol}}. \quad (2)$$

We centre our simulations at an effective redshift of  $z_{\text{eff}} = 0.39$ , and assume a redshift range of  $0.2 < z < 0.58$ , though our cosmological signal does not evolve with redshift (see the Appendix). The frequency at which we observe the HI signal changes with redshift as  $\nu = 1420\text{MHz}/(1+z)$ , so our redshift range corresponds to a frequency range of  $899 < \nu < 1184\text{MHz}$ , which is representative of the MeerKAT telescope in the  $L$  band (Santos et al. 2017). The dimensions of our simulation box are, in comoving distance units,  $L_x = 1000$ ,  $L_y = 1000$ ,  $L_z = 924.78\text{Mpc } h^{-1}$ , and we grid it into volume-pixels (voxels) of dimensions  $N_x = 256$ ,  $N_y = 256$ ,  $N_z = 285$  [ $z$  denotes the parallel to the line-of-sight (LoS) direction, while  $x$  and  $y$  are perpendicular to the LoS]. This gridding corresponds to a frequency resolution of  $\delta\nu = 1\text{MHz}$ . The sky area covered by our simulation is approximately  $3000\text{deg}^2$ , also representative of a MeerKLASS-like survey (Santos et al. 2017).

After adding our different signals together, we smooth them by a constant Gaussian telescope beam, to make the angular resolution representative of future surveys. The resolution of the beam is determined by the frequency of observation ( $\nu$ ) and the size of the telescope dish ( $D_{\text{dish}}$ ):

$$\theta_{\text{FWHM}} = \frac{1.22c}{\nu D_{\text{dish}}}, \quad (3)$$

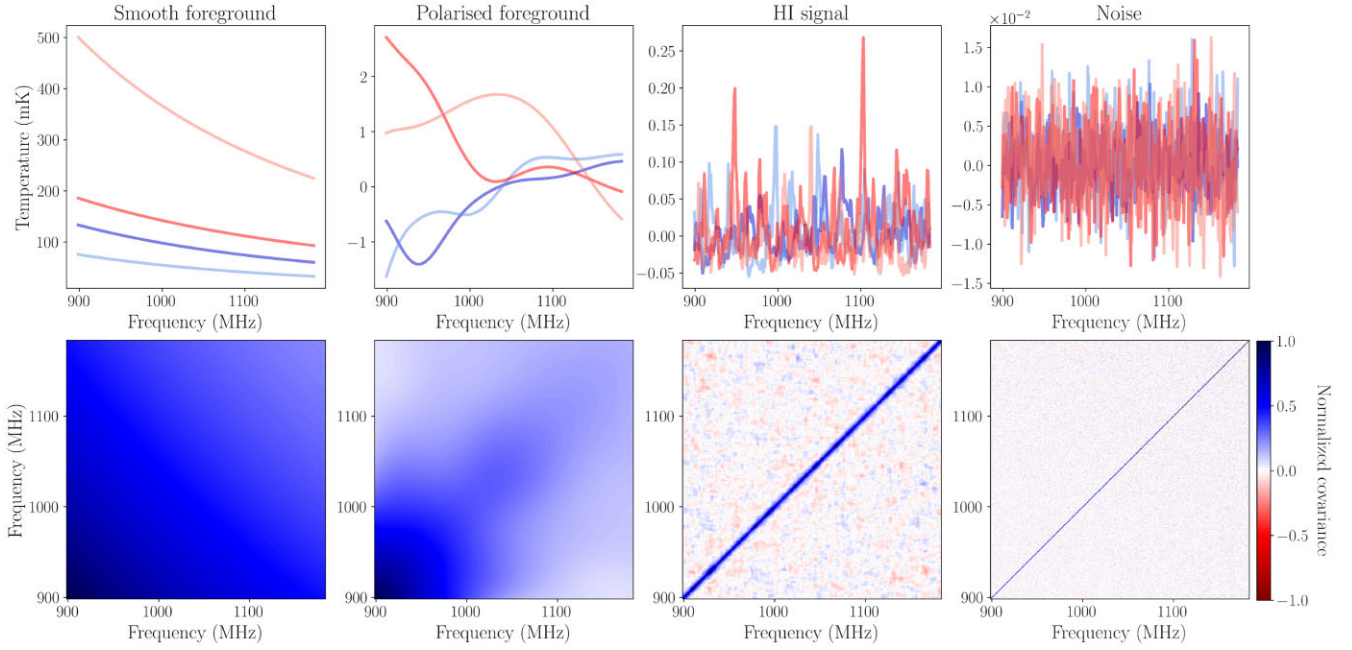
where  $c$  is the speed of light. We choose  $D_{\text{dish}} = 15\text{m}$  based on the SKA1-MID dishes, but this is also similar to the MeerKAT dish size. Although  $\theta_{\text{FWHM}}$  is frequency dependent, we smooth every frequency channel to the same resolution to aid the performance of our foreground removal algorithms (Matshawule et al. 2020). We choose the common resolution to be that at our minimum frequency,  $\nu_{\text{min}} = 899\text{MHz}$ , since this will give the largest beam size, equivalent to  $\theta_{\text{FWHM}} = 1.55\text{deg}$ .

The top row of Fig. 1 shows each component of our simulation as a function of frequency. Each line shows a different random

<sup>1</sup>[https://gitlab.com/flomertens/ps\\_eor](https://gitlab.com/flomertens/ps_eor)

<sup>2</sup><https://github.com/paulassoares/gpr4im>

<sup>3</sup><http://intensitymapping.physics.ox.ac.uk/CRIME.html>



**Figure 1.** *Top:* Random LoS samples drawn from each component of our simulation, namely from left to right: the smooth foregrounds, the polarized foregrounds, the HI (21cm) signal, and the instrumental noise. *Bottom:* Normalized frequency covariance for each component. The panels are arranged in order of decreasing signal amplitude and frequency correlation, from left to right. [URL](#)

pixel, to give a better idea of what the signal looks like throughout our simulation box. The foreground signals are much smoother in frequency than the HI signal, and this will be used later in order to perform the foreground removal. The polarized foreground is less smooth in frequency than the smooth foregrounds, which will be important for foreground removal too. On the bottom row, we show the normalized frequency covariance of each component of our signal. It will be relevant later that the smooth foreground signal is brighter and more correlated at lower frequencies.

### 3 FOREGROUND REMOVAL METHODS

Here, we describe the main foreground removal technique studied in this paper, GPR. We also present the PCA technique, which is a widely used foreground removal technique that we will compare the performance of GPR to.

#### 3.1 Gaussian process regression

In this section, we will introduce the framework of GPR and discuss how it applies to our foreground removal problem. For a comprehensive and in-depth discussion of Gaussian processes, please see Rasmussen & Williams (2006), and for an intuitive description, see Luger, Foreman-Mackey & Hedges (2021). We used these, as well as M18, Mertens et al. (2020) and Kern & Liu (2020), as excellent references for writing this section.

We can model our simulated data as a vector  $\mathbf{d}$ , where each element is one ‘observation’ (while our data is simulated, we will use the term ‘observation’ in this paper for simplicity). The length of this data vector is determined by the number of observations made (in our case,  $N_v = N_z = 285$  observations made at  $N_v$  frequency channels). Each element (or observation) is an image *slice* with total number of pixels given by  $N_\theta = N_x \times N_y$ , and containing all the spatial information observed at that frequency. Therefore, our

data vector is actually a data *matrix* with dimensions  $N_\theta \times N_v$ . The frequency range of our data, a vector with length  $N_v$ , is denoted as  $\mathbf{v}$ .

Our data is composed of four signal components: the smooth foregrounds ( $\mathbf{f}_{\text{smooth}}$ ), the polarized foregrounds ( $\mathbf{f}_{\text{pol}}$ ), the HI cosmological signal ( $\mathbf{f}_{21}$ ) and the instrumental (Gaussian) noise ( $\mathbf{n}$ ). We write our data matrix as the sum of these components:

$$\mathbf{d} = \mathbf{f}_{\text{fg}} + \mathbf{f}_{21} + \mathbf{n}, \quad (4)$$

where our foregrounds  $\mathbf{f}_{\text{fg}}$  can be made up of either only a smooth component ( $\mathbf{f}_{\text{fg}} = \mathbf{f}_{\text{smooth}}$ ), or a smooth component and a polarized component ( $\mathbf{f}_{\text{fg}} = \mathbf{f}_{\text{smooth}} + \mathbf{f}_{\text{pol}}$ ). The smooth component is smooth in frequency, while the polarized one is less so, as seen in Fig. 1.

Because we are assuming that our signal components are separate from each other, the (frequency) covariance ( $\mathbf{C}$ ) of our data ( $\mathbf{d}$ ) can be written as

$$\mathbf{C} = \langle \mathbf{d}\mathbf{d}^T \rangle = \mathbf{d}\mathbf{d}^T / (N_v - 1) = \mathbf{C}_{\text{fg}} + \mathbf{C}_{21} + \mathbf{C}_{\text{n}}, \quad (5)$$

where the foreground covariance is written as  $\mathbf{C}_{\text{fg}} = \mathbf{C}_{\text{smooth}}$  in the absence of polarization, and  $\mathbf{C}_{\text{fg}} = \mathbf{C}_{\text{smooth}} + \mathbf{C}_{\text{pol}}$  in the presence of polarization. Here, we are assuming that the polarized foregrounds have an additive effect, which is a suitable approximation to first order. Formally, due to mode-mixing, polarization leakage would also have a multiplicative effect (Mertens et al. 2020).

##### 3.1.1 What is a Gaussian process?

The main idea behind GPR is that we can describe each component of our data  $\mathbf{d}$  as a *Gaussian process*. A Gaussian process is a Gaussian distribution over infinite dimensions. Think of a multivariate Gaussian distribution over two variables, but extend the number of variables to infinity. And instead of variables, it is functions of frequency  $\mathbf{v}$ . Each line in the top panel of Fig. 1 is a function,

representing signal in our data ( $\mathbf{d}$ ) which is a function of frequency ( $\mathbf{v}$ ).

To understand this better, think of how a multivariate Gaussian distribution over variables is defined by a mean  $\boldsymbol{\mu}$  (of size  $N \times 1$ ) and covariance  $\boldsymbol{\Sigma}$  (of size  $N \times N$ ). A random variable  $f$  ( $N \times 1$ ) that has a multivariate Gaussian distribution defined by  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  is written as

$$f \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (6)$$

Now, assume that the mean is actually a *mean function*, in our case a function of frequency:  $\boldsymbol{\mu} = m(\mathbf{v}) \equiv m$ . The only difference is that now each element of the mean depends on frequency:  $\mu_i = m(v_i)$ . Assume also that the covariance is defined by a *kernel function*:  $\mathbf{K} \equiv k(\mathbf{v}, \mathbf{v})$ , so each element of the covariance is given by  $\Sigma_{i,j} = k(v_i, v_j)$ . If we assume a random variable  $f$  is a Gaussian process, we write

$$\begin{aligned} f &\sim \mathcal{GP}(m, \mathbf{K}), \\ f(\mathbf{v}) &\sim \mathcal{N}(m(\mathbf{v}), k(\mathbf{v}, \mathbf{v})). \end{aligned} \quad (7)$$

Multivariate Gaussian distributions have several useful properties, many of which are shared by Gaussian processes. We outline a few of these below:

- (i) Multivariate Gaussian distribution (and Gaussian processes) are easy to sample from. All one needs is a mean (or mean function) and a covariance (or kernel function).
- (ii) Multivariate Gaussian distributions (and Gaussian processes) have a well-defined marginal likelihood function, assuming only Gaussian noise is present. It can be calculated analytically given a mean (or mean function) and a covariance (or kernel function), and can be maximized for inference problems.
- (iii) A Gaussian process is a *stochastic* process. Since we drawing from an infinite Gaussian distribution, each draw is random and can take any functional form consistent with the mean and kernel function.

We assume that our data  $\mathbf{d}$  (and each of its components) is one such variable  $f$ , which is drawn from a Gaussian process with a particular mean and kernel function as in equation (7). As is commonly done with GPR, we assume a zero mean function, which is true for our data since we mean-centre each frequency slice.

### 3.1.2 Foreground removal with GPR

If we consider our data  $\mathbf{d}$  to be a Gaussian process defined over the frequency range  $\mathbf{v}$ , we can obtain its joint probability with a Gaussian process defined at another set of frequencies  $\mathbf{v}'$  as

$$\begin{bmatrix} \mathbf{d} \\ \mathbf{d}' \end{bmatrix} = \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{v}, \mathbf{v}) & k(\mathbf{v}, \mathbf{v}') \\ k(\mathbf{v}', \mathbf{v}) & k(\mathbf{v}', \mathbf{v}') \end{bmatrix} \right). \quad (8)$$

Since our data is made up of separate components (each of which is a Gaussian process), we split its kernel function ( $\mathbf{K} \equiv k(\mathbf{v}, \mathbf{v})$ ) into separate components:

$$\mathbf{K} = \mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n, \quad (9)$$

where if we have polarized foregrounds presents, we write  $\mathbf{K}_{\text{fg}} = \mathbf{K}_{\text{smooth}} + \mathbf{K}_{\text{pol}}$ . Otherwise,  $\mathbf{K}_{\text{fg}} = \mathbf{K}_{\text{smooth}}$ .

We are interested in separating our foreground signal from the rest of our data. If we know our data's kernel function  $\mathbf{K}$ , we can write the joint probability distribution of our data  $\mathbf{d}$  and the foreground model

$f_{\text{fg}}$  as (M18):

$$\begin{bmatrix} \mathbf{d} \\ \mathbf{f}_{\text{fg}} \end{bmatrix} = \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n & \mathbf{K}_{\text{fg}} \\ \mathbf{K}_{\text{fg}} & \mathbf{K}_{\text{fg}} \end{bmatrix} \right) \quad (10)$$

Our foreground model  $f_{\text{fg}}$  is also a Gaussian process, with expectation value ( $\mathbb{E}[\mathbf{f}_{\text{fg}}]$ ) and covariance ( $\text{cov}[\mathbf{f}_{\text{fg}}]$ ) defined by

$$\mathbf{f}_{\text{fg}} \sim \mathcal{N}(\mathbb{E}[\mathbf{f}_{\text{fg}}], \text{cov}[\mathbf{f}_{\text{fg}}]). \quad (11)$$

The expectation value is a prediction of the foreground signal, and the covariance is the uncertainty in this prediction. These are given by (M18):

$$\begin{aligned} \mathbb{E}[\mathbf{f}_{\text{fg}}] &= \mathbf{K}_{\text{fg}}[\mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n]^{-1} \mathbf{d}, \\ \text{cov}[\mathbf{f}_{\text{fg}}] &= \mathbf{K}_{\text{fg}} - \mathbf{K}_{\text{fg}}[\mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n]^{-1} \mathbf{K}_{\text{fg}} \end{aligned} \quad (12)$$

The expectation value  $\mathbb{E}[\mathbf{f}_{\text{fg}}]$  is dependent not only on the foreground kernel  $\mathbf{K}_{\text{fg}}$  but also on the full kernel of our data  $\mathbf{K} = \mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n$ . It is thus important to ensure we have an appropriate kernel for all the elements in our data.  $\mathbb{E}[\mathbf{f}_{\text{fg}}]$  is also dependent on our data  $\mathbf{d}$ , so it can be thought of as: given the known data  $\mathbf{d}$ , the foreground kernel  $\mathbf{K}_{\text{fg}}$  and the full data kernel  $\mathbf{K}$ , what is the expected (mean) value of our foregrounds in our data's frequency range?

Thus, by estimating  $f_{\text{fg}}$ , we are making an informed prediction for what our foregrounds look like. We can then subtract this prediction from the data, and hopefully be left with only our cosmological signal and instrumental noise. This is how one performs foreground removal with GPR. The residual vector  $\mathbf{r}$  (what is leftover after GPR foreground removal) is defined as (M18):

$$\mathbf{r} = \mathbf{d} - \mathbb{E}[\mathbf{f}_{\text{fg}}]. \quad (13)$$

We are also interested in  $\text{cov}[\mathbf{f}_{\text{fg}}]$ , the 'error margin' of our foreground prediction  $\mathbb{E}[\mathbf{f}_{\text{fg}}]$ . This is important for bias correction techniques (see Section 3.1.6).

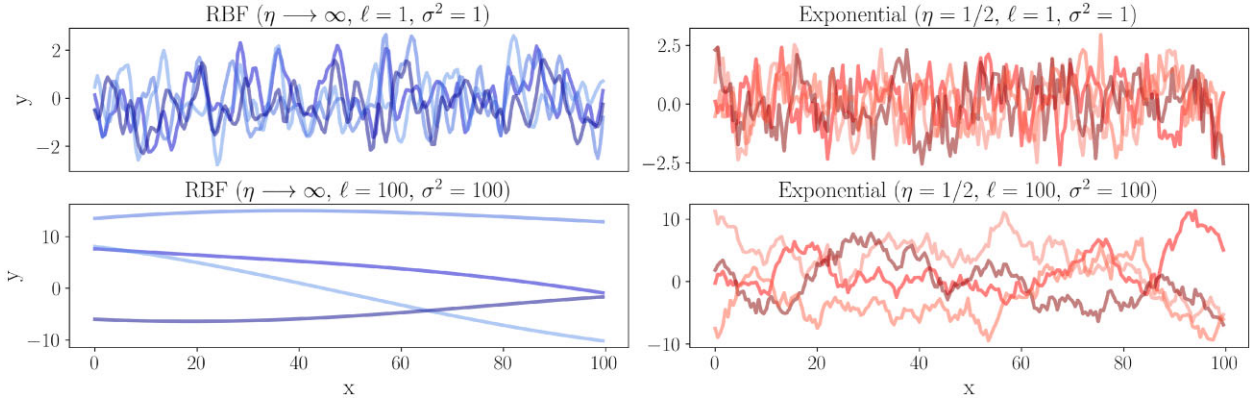
So far we have described the process of foreground removal with GPR assuming we know the kernel  $\mathbf{K} = \mathbf{K}_{\text{fg}} + \mathbf{K}_{21} + \mathbf{K}_n$ . In reality, we need to use the data to try and find the best-fitting kernel for each signal component.

### 3.1.3 Kernel functions

To understand how we find the optimized kernel functions for our data, let's first think of a simple linear regression problem. Consider a vector  $\mathbf{y}$  of observed values (dependent variable), and a vector  $\mathbf{x}$  of input values (independent variable), both of length  $N$ . If the variables follow a linear trend, it can be fit with a linear model:  $y_i = mx_i + b$ ,  $i = 1, \dots, N$ . This might be a good model for the data, but the model parameters ( $m$  and  $b$ ) still need to be optimized, which can be done using a maximum likelihood estimation, for example. Thus, this is a two-step process: first, a model is chosen to best describe the data, and then the parameters of the model are optimized.

Our kernel optimization problem is similar, but not as simple as this two-step process. We wish to find the kernel functions  $\mathbf{K}$  that best fit the covariance of our data  $\mathbf{C}$  (equation 5). For example, for the foreground covariance  $\mathbf{C}_{\text{fg}}$ , we want to find the kernel function  $\mathbf{K}_{\text{fg}}$  that best describes that covariance, and its best-fitting hyperparameters. The parameters of the kernel functions are called hyperparameters to emphasize that a Gaussian process is a non-parametric model.

We henceforth use the terms 'kernel', 'kernel function' and 'covariance function' interchangeably, but the *data* covariance is not equivalent to the covariance *function* – the covariance function



**Figure 2.** *Left:* RBF covariance samples. *Right:* Exponential covariance samples. We can see that overall, the RBF samples look smoother than the exponential samples. The top row shows the case of small variance and small length-scale ( $\ell = 1$ ,  $\sigma^2 = 1$ ), meaning the signal amplitude is small and the data is not very correlated in frequency. The bottom row is showing the opposite case of a higher variance and length-scale ( $\ell = 100$ ,  $\sigma^2 = 100$ ), with higher signal amplitude and more frequency correlation. [URL](#)

is the model we are optimizing in order to best describe the data covariance.

The covariance function must be a function of frequency ( $K = k(\nu, \nu')$ ), since our data is a function of frequency. There exist several widely used kernels in GPR. One of them is the Matérn kernel, which has the form (Rasmussen & Williams 2006):

$$K_{\text{Matern}}(\nu, \nu') = \sigma^2 \frac{2^{1-\eta}}{\Gamma(\eta)} \left( \sqrt{2\eta} \frac{|\nu - \nu'|}{\ell} \right)^\eta K_\eta \left( \sqrt{2\eta} \frac{|\nu - \nu'|}{\ell} \right), \quad (14)$$

where  $\Gamma$  is the gamma function and  $K_\eta$  is the modified Bessel function of the second kind. The three hyperparameters of our Matérn kernel model that we want to optimize are:

(i)  $\sigma^2$  is the *variance*, which describes the overall amplitude of the signal. We expect  $\sigma^2$  to be larger for our foregrounds than for our HI signal since the foregrounds are brighter (see Fig. 1);

(ii)  $\ell$  is the *length-scale*, which describes the typical scale of correlations in our data, across frequency. A larger value of  $\ell$  means the data is more correlated in frequency, so we expect  $\ell$  to be larger for our foregrounds than for our HI signal, since the foregrounds are more correlated in frequency (see Fig. 1);

(iii)  $\eta$  is the *spectral parameter*, which determines the overall ‘smoothness’ of our data. We expect this to be larger for our foregrounds than for our HI signal, since our foregrounds look smooth in frequency while the HI signal is more Gaussian (see Fig. 1).

The spectral parameter  $\eta$  is particularly important because it can simplify the Matérn kernel. Some common choices are as follows:

(i)  $\eta \rightarrow \infty$ : in this limit, the Matérn kernel simplifies to a radial basis function (RBF), also called a squared exponential function or Gaussian function. This limit describes a very smooth signal, which is ideal for our foreground signal. Both Kern & Liu (2020) and Mertens et al. (2020) use this kernel to describe their smooth foreground component. It has the form:

$$K_{\text{RBF}}(\nu, \nu') = \sigma^2 \exp \left( -\frac{(\nu - \nu')^2}{2\ell^2} \right); \quad (15)$$

(ii)  $\eta = 5/2$ : In this limit, the Matérn kernel is relatively smooth but not as smooth as the RBF kernel. M18 uses this to describe their smooth foreground component;

(iii)  $\eta = 3/2$ : In this limit, the Matérn kernel is even less smooth in frequency, and it can be used to describe foreground components

that have medium frequency smoothness, as is done in e.g. M18 and Mertens et al. (2020);

(iv)  $\eta = 1/2$ : In this limit, the Matérn kernel is the least smooth in frequency, and is known as the exponential function. It agrees well with a spectrally varying signal such as the HI signal. The form of this kernel, which is used in e.g. M18, Mertens et al. (2020) and Kern & Liu (2020) to describe the HI signal, is

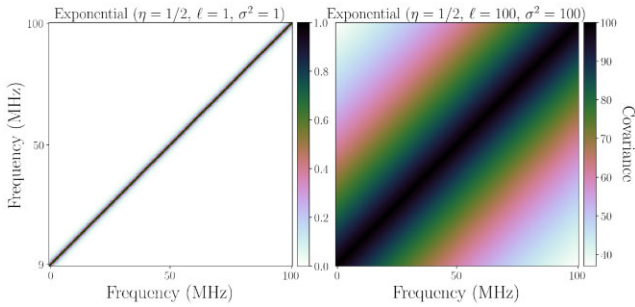
$$K_{\text{exp}}(\nu, \nu') = \sigma^2 \exp \left( -\frac{|\nu - \nu'|}{\ell} \right). \quad (16)$$

The Matérn kernel is very useful and widely applicable due to its flexibility. With different hyperparameters, it can describe a variety of different signals. We find that the Matérn kernel is sufficient to describe all components in our data, with differing hyperparameters. Other kernel functions, such as the rational quadratic function, are also useful. Gehlot et al. (2019) for example use it to describe frequency varying foregrounds, which are qualitatively similar to our polarized foregrounds, however we did not find it to perform as well as the Matérn kernel in our data.

There also exist kernel functions with periodic behaviour, for more oscillatory signals. Our simulations do not include this type of behaviour hence we do not consider these kernels. However, with real data, there might exist systematic effects with this type of behaviour.

In Fig. 2, we exemplify what samples drawn from an exponential and RBF covariance look like. We consider the cases of small variance and length-scale ( $\ell = 1$ ,  $\sigma^2 = 1$ ), as well as a larger variance and length-scale ( $\ell = 100$ ,  $\sigma^2 = 100$ ) to illustrate the difference. In both cases, the RBF samples are much smoother than the exponential samples, due to it having a higher spectral parameter ( $\eta \rightarrow \infty$ ). For both the RBF and exponential kernels, increasing the variance  $\sigma^2$  leads to an increase in the amplitude of the signal, and increasing the length-scale  $\ell$  makes the samples more correlated in frequency.

We also demonstrate what the exponential kernel function looks like. In Fig. 3, we show the exponential kernel in two different cases: the low variance, low length-scale case ( $\ell = 1$ ,  $\sigma^2 = 1$ ), and the high variance, high length-scale case ( $\ell = 100$ ,  $\sigma^2 = 100$ ). Increasing the variance changes the amplitude of the covariance, as seen in the colour bar, while increasing the length-scale changes the correlation of the signal, as seen by how the diagonal correlation is much wider in the second plot. A wider diagonal means that the data are more correlated in frequency, while a smaller diagonal means that the data



**Figure 3.** Exponential kernel for two different cases. *Left:* Low variance, low length-scale case ( $\ell = 1, \sigma^2 = 1$ ). *Right:* High variance, high length-scale case ( $\ell = 100, \sigma^2 = 100$ ). [URL](#)

points are mostly correlated with themselves and those immediately around them.

### 3.1.4 Hyperparameter optimization and model selection

We have established that the kernel hyperparameters affect the behaviour of the samples being drawn from it. How do we simultaneously pick the best kernel model (e.g. choice of  $\eta$  for a Matérn kernel), and also the best hyperparameters (e.g. length-scale and variance)? These steps are interdependent. When comparing different models, we want to compare the best overall performance of each model. Therefore, we first optimize the hyperparameters of our model by maximizing the marginal likelihood over the function parameters (which are distinct from the model hyperparameters defined above); we then compare its evidence (i.e. the marginal likelihood over the function parameters) to other optimized models. When we talk about ‘model’ in this section, we are talking about finding the kernel function choices that best model our data (what we have referred to as  $\mathbf{K}$  in previous sections).

Given a model (kernel function) with hyperparameters  $\theta$ , the marginal likelihood of the data (also known as the Bayesian evidence) can be calculated as the integral of the data likelihood  $\mathcal{L}$  times the prior:

$$p(\mathbf{d}|\mathbf{v}, \theta) = \int \mathcal{L}(\mathbf{d}|\mathbf{x}, \mathbf{v}, \theta) p(\mathbf{x}|\mathbf{v}, \theta) d\mathbf{x}, \quad (17)$$

where  $\mathbf{x}$  is another vector occupying the same space as our data, which describe the functions of the Gaussian process. The result of this integration is a constant, called the log-marginal likelihood (LML):

$$\log p(\mathbf{d}|\mathbf{v}, \theta) = -\frac{1}{2} \mathbf{d}^T \mathbf{K}^{-1} \mathbf{d} - \frac{1}{2} \log |\mathbf{K}| - \frac{n}{2} \log 2\pi, \quad (18)$$

where  $n$  is the number of data points sampled, and  $\mathbf{K}$  is again the full kernel function for the data (the sum of the kernels for each component of the data, including noise). We assume that each component of our data is a Gaussian process, and that the noise is Gaussian distributed, thus the LML can be calculated analytically. The LML not only favours models that fit our data well but it also disfavors overly complex models (Rasmussen & Williams 2006).

The LML tells us the ‘evidence’ of the model, i.e. how well our covariance function fits our data. We can take the ratio of two evidences to compare how well different models fit our data. This is known as the Bayes factor  $\mathcal{Z}$ . When working with the log evidence, the log of the Bayes factor is achieved by taking the difference between the LML of two different models (called  $m_1$  and  $m_2$ ):

$$\log \mathcal{Z} = \text{LML}_{m_1} - \text{LML}_{m_2}, \quad (19)$$

where larger values of  $\mathcal{Z}$  or  $\log \mathcal{Z}$  indicate that  $m_1$  fits the data better than  $m_2$ .

We are also interested in obtaining the posterior distribution of our hyperparameters. According to Bayes’ theorem, we can obtain the posterior probability density by multiplying the likelihood with the prior. In log space, the log posterior probability density can be written as

$$\log p(\theta|\mathbf{d}, \mathbf{v}) \propto \log p(\mathbf{d}|\mathbf{v}, \theta) + \log p(\theta). \quad (20)$$

Because the LML is easy to calculate with equation (18), we can calculate it for several choices of hyperparameters  $\theta$  and find the combination of  $\theta$  that gives us the *maximum likelihood* – the largest value for LML. This method for maximizing the LML is known as Type-II maximum likelihood (ML-II, e.g. gradient descent), and yields point estimates for the hyperparameters. While fast, ML-II has several caveats, as it is prone to: overfitting when there are many hyperparameters, getting stuck in local minima, and underestimating predictive uncertainty by yielding only point estimates for the best-fit values of the hyperparameters (without any uncertainty) (Simpson, Lalchand & Rasmussen 2020). Some ML-II methods, such as simulated annealing or ensemble sampling, could lead to better global maxima estimates, but are still unable to provide uncertainties on the hyperparameters. We use the PYTHON package GPY<sup>4</sup> (GPy 2012) to run gradient descent in the context of GPR.

Nested sampling (NS) is a technique able to sample complex distributions, and it focuses on estimating the model evidence (equation 17) and well as its uncertainty. It also yields the posterior distribution of the hyperparameters as a by-product, which is useful. We used the PYTHON package PYMULTINEST<sup>5</sup> (Buchner et al. 2014; Feroz, Hobson & Bridges 2009) to run NS.

A downside to this method is that it is computationally expensive. We compared our model evidence estimate from NS to that of gradient descent, and found them to be consistent. We therefore use GPY for quickly comparing different models, and NS for obtaining the final best-fitting hyperparameter results of our chosen model. Our model was relatively simple, with few hyperparameters, so both gradient descent and NS were able to sample the hyperparameter space easily. For more complex models or real data, the hyperparameter space will likely be more complex, leading to differences between gradient descent and NS, so it would be better to use NS in practice. Our simulations are idealized and do not suffer from systematic effects other than polarization leakage, thus it was straightforward to find the best-fitting models since the differences between the evidences of different kernel choices were very large (much larger than the offset between the NS and gradient descent evidences).

We summarize our model selection and hyperparameter optimization steps below:

- (i) Choose a kernel model  $\mathbf{K}$ , and optimize it using gradient descent, obtaining an estimate for the evidence.
- (ii) Compare the evidence of the model with that of other models (e.g. try a different kernel for the smooth foregrounds  $\mathbf{K}_{fg}$ ).
- (iii) Conclude the model selection by using the Bayes factor to find the best model.
- (iv) Once a model has been selected, run NS to obtain a more robust estimate of the evidence and hyperparameters.
- (v) If the posterior distributions from NS look sensible, take the peaks of the distributions as the final optimized hyperparameter

<sup>4</sup><https://github.com/SheffieldML/GPy>

<sup>5</sup><https://johannesbuchner.github.io/PyMultiNest>

values. Though not done in this analysis, it is also possible to take into account the error on this prediction by looking at the standard deviation of the posterior distributions.

During model selection, one should carefully monitor the best-fit hyperparameter estimates to check if the result is hitting against a prior or seems unphysical given what is known about the data. For example, we would not expect the variance of the HI kernel to be orders of magnitude larger than the variance of the foreground kernel – we would expect the opposite. If this is happening, the model is not a good fit for the data, regardless of the evidence.

### 3.1.5 Noise treatment

There are a few different ways to treat Gaussian noise in GPR. We note some below:

(i) Constant noise: Assume the data has some random Gaussian noise that is constant in frequency, with variance given by  $\sigma_n^2$ . GPR will then assume that the observed target value  $y$  is given at a particular frequency by

$$y = f(\mathbf{v}) + \epsilon, \quad (21)$$

where  $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ , and  $f$  is defined in equation (7). We can think of this as adding a new kernel  $K_n$  to our model  $K$ , given by  $K_n(\nu, \nu') = \sigma_n^2 \delta_{\nu, \nu'}$  where  $\delta_{\nu, \nu'}$  is a Kronecker delta which is one where  $\nu = \nu'$  and zero elsewhere. The kernel can also be written as  $K_n = \sigma_n^2 I$  where  $I$  is the identity matrix. If one has a reasonable estimate for the noise in the data,  $\sigma_n^2$  can be fixed. Otherwise, it is a free hyperparameter that can be optimized.

(ii) Heteroscedastic noise: This is similar to the constant noise case, with one difference: we assume that the Gaussian noise variance is *not* constant in frequency. We would then have that for a given frequency  $\nu_i$ :

$$y_i = f(\nu_i) + \epsilon_i \quad (22)$$

where  $\epsilon_i \sim \mathcal{N}(0, \sigma_{n,i}^2)$ ,  $\sigma_{n,i}^2$  being the frequency dependent noise variance. One would then write the noise kernel as  $K_n(\nu, \nu') = \sigma_n^2(\nu, \nu')$ , where  $\sigma_n^2(\nu_i, \nu'_i) = \sigma_{n,i}^2$  for  $\nu_i = \nu'_i$  and zero otherwise. It is also possible to either fix this noise variance for each frequency, or let them be free hyperparameters. As discussed in Section A4, in our case the noise variance is changing with frequency.

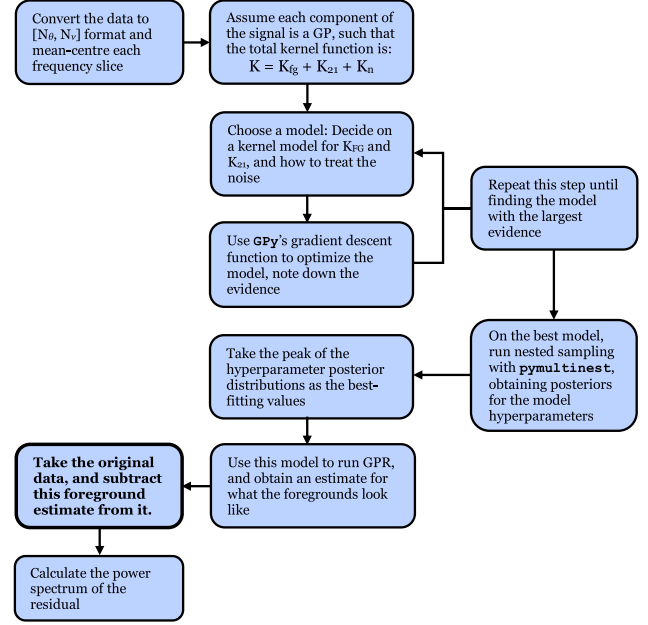
(iii) Indistinguishable noise: Assume that there is indistinguishable noise present in the data, such that  $y = f(\mathbf{v})$ . This might mean that there is either no noise present in the data, or that there is noise present but it is indistinguishable from other signals, and so the covariance function of other signals is also describing the noise.

(iv) Another noise kernel: Choose a kernel to represent the noise signal, such as an exponential kernel.

For our data, we tried each of these methods and used the Bayes factor to decide on the best approach. We find that for our analysis, case (iii) is the best, possibly due to how small the noise is making it indistinguishable from the faint HI cosmological signal.

In real data, more complex instrumental noise is likely to be present. For example, time correlated  $1/f$  noise is a problem in HI intensity mapping experiments, and would require more careful treatment, such as its own kernel (see e.g. Li et al. 2020).

We summarise our pipeline for removing foregrounds from our data using GPR and obtaining the residual power spectrum in Fig. 4.



**Figure 4.** A flowchart outlining our foreground removal pipeline, starting with our data in  $[N_x, N_y, N_z]$  format, and ending up with the power spectrum estimate of our foreground cleaned residual. In bold is the step where foreground removal is performed.

### 3.1.6 Bias correction

Ideally, we want our estimate of the foreground removed residual  $\mathbf{r}$  to be unbiased. Unfortunately, for current foreground removal methods this is not the case. Foreground removal methods especially struggle to recover the true HI power spectrum on large scales, where our desired HI signal looks similarly correlated in frequency to the foregrounds and is erroneously removed (see e.g. C21).

For GPR, this is also a problem. Previous works have tried to account for over or undercleaning with GPR by using *bias correction* techniques. Mertens et al. (2020) presented an additive bias correction to the residual covariance and power spectrum, which arises analytically assuming the data covariance is perfectly known. Kern & Liu (2020) showed that this correction fails to recover the true EoR 21cm power spectrum when the EoR signal covariance is misestimated. While the EoR signal is still undetected and not fully understood, the HI signal from IM follows the well-known large-scale structure of the Universe. Therefore, we are more likely to estimate the HI signal covariance of our data accurately, making this bias correction a better approximation in our case (especially when considering idealized simulations).

Kern & Liu (2020) point out that such an additive bias correction by definition cannot account for *under*-cleaning, only *over*-cleaning of the signal. They derive a multiplicative bias correction as part of their optimal quadratic estimator for the power spectrum, and show that it is robust to imperfect estimations of the EoR signal covariance. Their correction by construction cannot under-predict the 21cm signal. However, it requires perfectly knowing the foreground covariance.

Since our simulations are idealized and we are confident about our data covariance estimations, we show the effect of the Mertens et al. (2020) bias correction to our results, and leave comparison between the Mertens et al. (2020) and Kern & Liu (2020) corrections for future work. We only show this correction to demonstrate its effect

in our case, and do not consider it when comparing our GPR results with PCA. The bias correction is analogous to the transfer function, commonly applied to PCA cleaned data, in that it is a data-driven approach for estimating signal loss in a foreground clean (Switzer et al. 2015). Since we do not apply a transfer function correction to PCA, we do not consider the bias correction to GPR when comparing the two. We want to establish their differences before any signal loss correction is applied. Ideally we want the need for correction to be as minimal as possible, hence it is important to compare methods before corrections.

The Mertens et al. (2020) bias correction is well described in their Section 3.3.2, and we summarise it here. For the covariance of the residual,  $\langle \mathbf{r}\mathbf{r}^T \rangle$ , the bias correction is applied by adding the foreground model covariance ( $\text{cov}[\mathbf{f}_{\text{fg}}]$ ) to the result:

$$\langle \mathbf{r}\mathbf{r}^T \rangle_{\text{unbiased}} = \langle \mathbf{r}\mathbf{r}^T \rangle + \text{cov}[\mathbf{f}_{\text{fg}}]. \quad (23)$$

For the radial power spectrum, we calculate the bias correction as outlined below.

- (i) Draw a number of random realizations from a multivariate Gaussian distribution with mean zero and covariance given by  $\text{cov}[\mathbf{f}_{\text{fg}}]$ . The realizations are drawn in the same dimensions as our data ( $N_\theta \times N_\nu$ );
- (ii) Measure the radial power spectrum for each of these realizations;
- (iii) Take the average of these power spectra;
- (iv) Add the averaged power spectrum to the residual power spectrum obtained with GPR.

The procedure for calculating the bias correction for the spherically averaged power spectrum is similar. The only difference is that in step (ii), we bin the measured radial power spectrum into the power spectrum  $k$ -bins. This ensures we are ignoring the transverse  $k_\perp$  modes in our spherically-averaged power spectrum. This is essential since these will only contain random Gaussian noise due to how the simulations are generated in step (i). The only physically relevant information in our simulations is in the radial direction since  $\text{cov}[\mathbf{f}_{\text{fg}}]$  only contains radial information. No bias correction is applied to the transverse power spectrum because  $\text{cov}[\mathbf{f}_{\text{fg}}]$  does not give us any information about what uncertainties might be present in the transverse direction.

### 3.2 Principal Component Analysis

PCA is a blind component separation method able to remove astrophysical foregrounds from Hi IM data. It does not require prior knowledge of what the foregrounds look like in order to work. Methods similar to PCA have been used in the analysis of real Hi IM data since it does not make many assumptions about the data. This is useful since we do not yet understand the instrumental systematic effects of Hi IM experiments well enough (Masui et al. 2013; Wolz et al. 2016; Anderson et al. 2018). We choose this popular foreground removal method to compare the performance of GPR to.

PCA tries to estimate the foreground signal by transforming the data to a dimensional basis which maximises variance. In this new context, we expect the first few basis vectors - which contain highly correlated signal with the largest amplitude - to represent the foregrounds. These basis vectors are called the *principal components*, and we designate the number of principal components containing the foreground information as  $N_{\text{FG}}$ .

Recall that we can describe our data as a matrix  $\mathbf{d}$  with dimensions  $N_\theta \times N_\nu$ . We assume that our data can be represented as a linear

system

$$\mathbf{d} = \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}, \quad (24)$$

where  $\mathbf{A}$  is the mixing matrix, containing the amplitude of the  $N_{\text{FG}}$  separable components ( $\mathbf{s}$ ), and  $\boldsymbol{\epsilon}$  is the residual signal, which includes the noise and cosmological signal. The residual is what we want to obtain, and can be obtained simply by rearranging the above equation

$$\boldsymbol{\epsilon} = \mathbf{d} - \mathbf{A}\mathbf{s}. \quad (25)$$

This foreground removed residual  $\boldsymbol{\epsilon}$  is the PCA equivalent to the residual  $\mathbf{r}$  we obtain with GPR, and we will compare which of these two are closer to the true residual. For more detail of how exactly PCA finds  $\mathbf{A}$  and  $\mathbf{s}$  and performs foreground removal, please see C21.

## 4 RESULTS

Here we present our main results, which compare the foreground removal performance of GPR to that of PCA. We do this first in the idealized case of no polarization leakage, then add in the polarization leakage to see how it affects our results. We also investigate how dependent our results are on bandwidth and redshift, and whether a foreground removal transfer function is possible with GPR.

When comparing GPR and PCA, we always show two choices of  $N_{\text{FG}}$  for PCA: one that leads to an under-cleaning of the foregrounds, and one that leads to an over-cleaning. This will help us more robustly compare the GPR and PCA performances. We also show the residuals between what GPR or PCA predict versus the true underlying signal power spectrum. We conclude that a technique is working better if it yields residuals that are close to zero, i.e. a prediction that is close to the truth. We point out when methods under- or overpredict the true power spectrum, but make no conclusion about which of these predictions is best. Overpredicting the power spectrum means there are residual foregrounds left in the data, which can bias results and boost errors. Underpredicting is caused by a loss of the desired underlying signal, which also biases results. However, signal loss can be in principle be corrected with transfer functions (Switzer et al. 2015), but since we do not fully explore this, leaving it to future work, we make no conclusions about which circumstance is more desirable between under- and overprediction of the power spectrum.

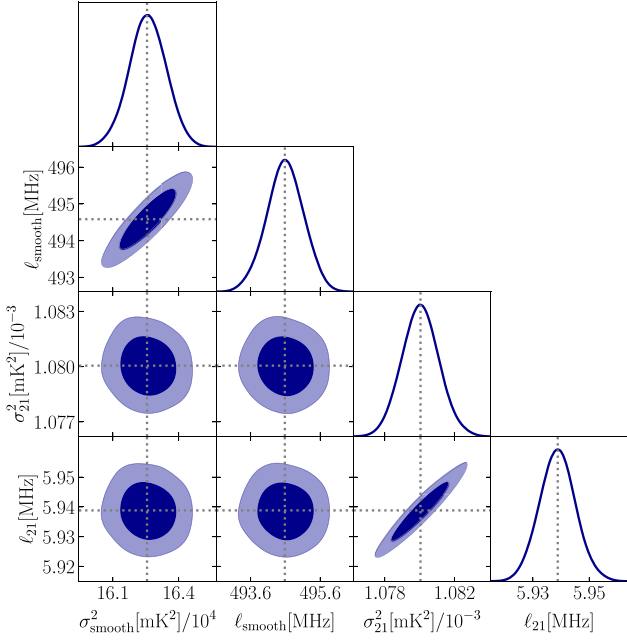
Throughout, we also show the power spectrum results including the bias correction discussed in Section 3.1.6. This is only to show the effect of this bias correction, and we do not consider it when comparing the performance to PCA which would also require some treatment for signal loss, such as a transfer function, to ensure a fair comparison. We use the terms ‘spherically averaged power spectrum’ and ‘power spectrum’ interchangeably, but specify when we are talking about transverse or radial power spectra.

We define the largest accessible scale for the spherically averaged power spectrum to be  $k_{\text{min}} = 2\pi/V^{1/3}$ , where  $V$  is the total comoving volume of our data. For the radial power spectrum, this becomes  $k_{\text{min}, \parallel} = 2\pi/L_z$ , and for the transverse power spectrum  $k_{\text{min}, \perp} = 2\pi/\sqrt{(L_x^2 + L_y^2)}$ . Throughout, we assume a bin width of twice the smallest accessible wavenumber.

### 4.1 No polarization

In the no polarization case, we want to find what kernel best describes the smooth foregrounds and the Hi signal, and choose a way to treat the noise, as discussed in Section 3.1.5. Using our model selection pipeline (which includes a Bayes factor analysis, see Section 3.1.4), we find that the exponential kernel (equation 16) best describes the Hi





**Figure 5.** Posterior distribution of our model hyperparameters, obtained using nested sampling in the no polarization case. Dotted grey lines show the best-fitting hyperparameter results obtained with gradient descent. [URL](#)

signal, and the RBF kernel (equation 15) best describes the smooth foregrounds.

We tested the noise scenarios outlined in Section 3.1.5 and find that the *indistinguishable noise* case performs best. In our data, the noise is so small that the exponential covariance for the HI signal is probably working well to describe both the HI signal and the noise. For real data, especially if the noise is larger or more correlated in frequency, this might not be the case.

Our kernel model for this no polarization case can be summarized as: Exponential (HI signal kernel, also describing indistinguishable noise) + RBF (smooth foreground kernel) + indistinguishable noise assumed throughout. Our hyperparameters for this model are:  $\sigma_{\text{smooth}}^2$  is the RBF kernel variance (describing the amplitude of our smooth foregrounds),  $\ell_{\text{smooth}}$  is the RBF kernel length-scale (describing the frequency correlation of our smooth foregrounds),  $\sigma_{21}^2$  is the exponential kernel variance (describing the amplitude of our HI signal and indistinguishable noise), and  $\ell_{21}$  is the exponential kernel length-scale (describing the frequency correlation of our HI signal and indistinguishable noise).

We run nested sampling for this model using `PYMULTINEST`, obtaining a better estimate for our evidence as well as our hyperparameter posteriors. We find that our evidence obtained with gradient descent (using `GPY`) and with nested sampling are consistent, as are the best estimates for the hyperparameters (see Fig. 5). This level of consistency is sufficient for our simulations, where the signal is idealized and the kernel model is simple. For real data the kernel model and hyperparameter posterior distributions will be more complex, so nested sampling should be used instead of gradient descent.

We plot the posterior distribution of our hyperparameters (obtained from nested sampling) in Fig. 5, and the median and  $1\sigma$  of the distributions can be found in Table 1. We also quote the uniform priors imposed in each hyperparameter in Table 1. We find that priors are required (especially on length-scales) to successfully separate the HI and foreground signal, otherwise the HI kernel will try to fit to the

foregrounds. We tested different prior ranges and found that making it narrower did not improve results, however this might not be the case for real data.

From Fig. 5 we see that the hyperparameters of each kernel (e.g.  $\sigma_{21}^2$  and  $\ell_{21}$ ) are correlated with each other but not with those of other kernels, as expected since the signals are different. The dotted lines show the best-fitting hyperparameter estimates from `GPY`'s gradient descent optimization, and they agree well with the peaks of the posterior distributions obtained with NS.

We use our best-fitting, optimized kernel model to perform foreground removal with GPR and compare this with PCA. We show two cases for PCA: one with  $N_{\text{FG}} = 3$ , and the other with  $N_{\text{FG}} = 2$ . We plot in Fig. 6 the power spectra of these residuals as well as the true residual (HI and noise) power spectrum. We choose to show the spherically averaged, radial, and transverse power spectra to compare how the foreground removal works in different directions. On the bottom panel we show the percentage difference between the foreground removed power spectra and the true residual power spectra, which we want to be as close to zero as possible.

For the spherically averaged power spectrum, the residuals in Fig. 6 show that GPR performs well on small scales, and increasingly worse on large scales. It yields residuals within 10 per cent of the truth down to a  $k$ -bin of  $k = 0.07 h \text{ Mpc}^{-1}$ , which we have marked with a vertical pink line. This is larger than the radial and transverse limits ( $k_{\parallel} = 0.01 h \text{ Mpc}^{-1}$ ,  $k_{\perp} = 0.02 h \text{ Mpc}^{-1}$ ) combined, meaning that the individual radial and transverse power spectra can access larger scales than the spherically averaged power spectrum. PCA ( $N_{\text{FG}} = 3$ ) also diverges at this  $k$ -bin, but PCA ( $N_{\text{FG}} = 2$ ) recovers larger scales. On small scales, GPR performs better than PCA ( $N_{\text{FG}} = 3$ ). While PCA ( $N_{\text{FG}} = 2$ ) can access larger scales, it overestimates the power spectrum, which can boost errors.

For the radial power spectrum in Fig. 6, GPR performs remarkably well, able to recover the largest scales down to a  $k_{\parallel}$ -bin of  $k_{\parallel} = 0.01 h \text{ Mpc}^{-1}$  within 10 per cent residual. Both PCA cases fail beyond this  $k_{\parallel}$ -bin. GPR also recovers small scales better than both PCA cases, making it a superior method for the radial power spectrum recovery.

The thin green line in Fig. 6 shows the bias corrected GPR recovery. Including it leads to an overestimation of the spherically averaged and radial power spectrum at large scales, and no obvious difference on small scales. The transverse power spectrum does not have a bias correction, as discussed in Section 3.1.6.

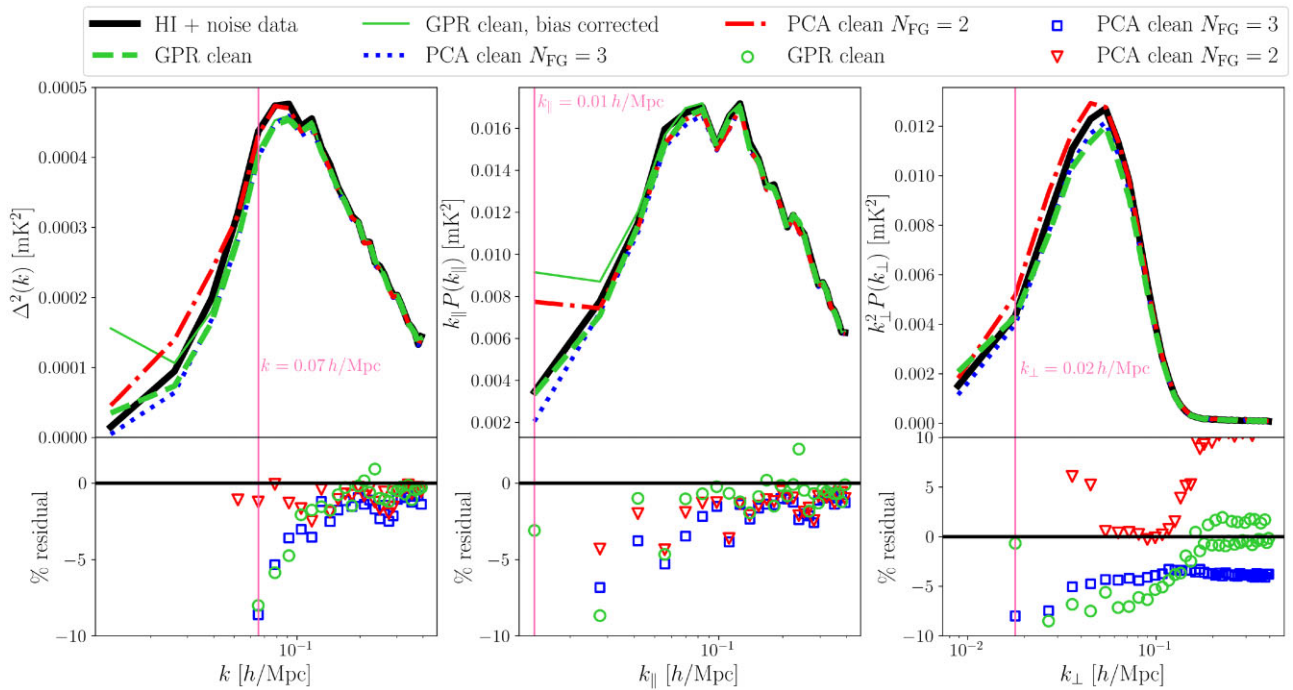
In the transverse power-spectrum case in Fig. 6, all methods perform worse than in the radial and spherically averaged power-spectrum cases. While GPR can still recover the truth within 10 per cent residual down to a  $k_{\perp}$ -bin of  $k_{\perp} = 0.02 h \text{ Mpc}^{-1}$ , the residuals tend to lie further away from zero than in the other power spectra cases. GPR again recovers small scales very well, better than PCA. However, since the signal is almost zero here due to the telescope beam, this is not very useful. We attribute the oscillatory nature of the true transverse power spectrum on small scales to the fact that the power here is near zero, so residual differences are floating point precision. Overall, the recovery of the transverse power spectrum is worse than of the radial power spectrum. This makes sense, since GPR mainly takes into account frequency (radial) information.

## 4.2 With polarization

When including polarized foregrounds, we still find that the exponential kernel is the best fit for the HI signal, and the RBF kernel

**Table 1.** Median and  $1\sigma$  values of our model hyperparameter posterior distributions for different cases, obtained using nested sampling.

Hyperparameter	$\sigma_{\text{smooth}}^2 [\text{mK}^2]/10^4$	Median and $1\sigma$ error estimates for our hyperparameters				$\sigma_{21}^2 [\text{mK}^2]/10^{-3}$	$\ell_{21} [\text{MHz}]$
		$\ell_{\text{smooth}} [\text{MHz}]$	$\sigma_{\text{pol}}^2 [\text{mK}^2]$	$\ell_{\text{pol}} [\text{MHz}]$			
No polarization	$16.26 \pm 0.09$	$494.60 \pm 0.54$	N/A	N/A	$1.08 \pm 0.001$	$5.94 \pm 0.006$	
With polarization	$6.72 \pm 0.04$	$475.30 \pm 0.67$	$0.50 \pm 0.004$	$58.42 \pm 0.07$	$1.21 \pm 0.002$	$6.73 \pm 0.01$	
With polarization (low frequency)	$5.75 \pm 0.03$	$489.62 \pm 1.03$	$5.37 \pm 0.08$	$74.52 \pm 0.17$	$1.43 \pm 0.004$	$8.02 \pm 0.02$	
With polarization (high frequency)	$2.71 \pm 0.02$	$588.61 \pm 1.17$	$4.06 \pm 0.09$	$106.98 \pm 0.39$	$1.13 \pm 0.002$	$6.22 \pm 0.01$	
With polarization (RFI case 1)	$6.64 \pm 0.04$	$476.39 \pm 0.67$	$0.42 \pm 0.004$	$55.66 \pm 0.11$	$1.33 \pm 0.003$	$7.46 \pm 0.02$	
With polarization (RFI case 2)	$6.76 \pm 0.04$	$482.21 \pm 0.71$	$0.93 \pm 0.01$	$79.70 \pm 0.14$	$1.38 \pm 0.003$	$7.64 \pm 0.02$	
With polarization (RFI case 3)	$6.47 \pm 0.03$	$473.49 \pm 0.67$	$0.66 \pm 0.006$	$62.33 \pm 0.09$	$1.37 \pm 0.003$	$7.59 \pm 0.02$	
With polarization (+ lognormal)	$6.58 \pm 0.03$	$475.11 \pm 0.64$	$0.44 \pm 0.004$	$57.86 \pm 0.07$	$2.38 \pm 0.003$	$5.28 \pm 0.006$	
Prior	$\mathcal{U}(1000, 100000000)$	$\mathcal{U}(200, 10000)$	$\mathcal{U}(0.0001, 10)$	$\mathcal{U}(15, 500)$	$\mathcal{U}(0.000001, 0.5)$	$\mathcal{U}(0.01, 15)$	



**Figure 6.** Power spectra results for the no polarization case. *Top:* True HI + noise signal (black solid line), PCA foreground cleaned residuals (blue dotted line for  $N_{\text{FG}} = 3$  and red dash-dotted line for  $N_{\text{FG}} = 2$ ), and GPR foreground cleaned residuals without bias correction (green dashed line) and with (thin green solid line). *Bottom:* Percentage residual difference between the foreground removed residual power spectra and the true HI power spectra. *Left:* Spherically averaged power spectrum. *Centre:* Radial power spectrum. *Right:* Transverse power spectrum. [URL](#)

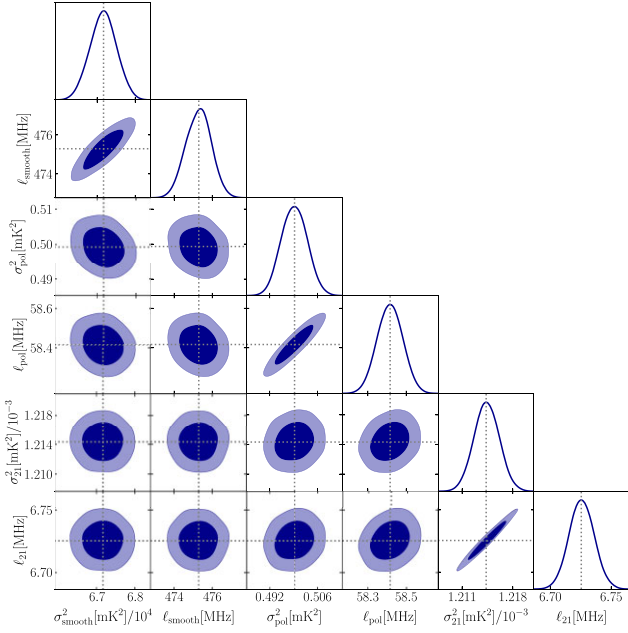
is the best for the smooth foregrounds. For the polarized signal, we find that the RBF kernel is also the best fit, though with smaller variance and length-scale than in the smooth foreground case. We again find that the indistinguishable noise treatment gives the best results.

Our kernel model in the presence of polarization is Exponential (HI signal kernel, also describing indistinguishable noise) + RBF (smooth foreground kernel) + RBF (polarized foreground kernel) + indistinguishable noise assumed throughout. We run nested sampling for this model, and plot the hyperparameter posterior distributions in Fig. 7. As in the no polarization case, the hyperparameter of each kernel is correlated with itself. The exponential kernel hyperparameters are the most correlated, and the RBF kernel describing the smooth foregrounds has the least correlated hyperparameters. However, the polarized foreground hyperparameters also seem to be negatively correlated with the smooth foreground hyperparameters, and positively correlated with the HI signal + noise parameters. This indicates that the polarized foreground signal might not be entirely described by the polarized foreground kernel, and some polarized signal is being described by the smooth foreground and

HI + noise kernels. The best-fitting hyperparameter estimates from GPY’s gradient descent again agree well with the posterior distribution obtained with NS.

We record the median and  $1\sigma$  error estimates of our hyperparameters in Table 1. The variance parameter for the smooth foregrounds ( $\sigma_{\text{smooth}}^2$ ) is a factor of  $\sim 2$  smaller than in the no polarization case, possibly due to difficulty in separating the smooth and polarized foregrounds. The estimated evidence (LML) from nested sampling, describing how well our kernel model describes our data, is smaller when including the polarized foregrounds. For the no polarization case, the evidence is 1 per cent larger than when including polarization leakage.

We plot the best-fitting kernel function for each component in Fig. 8. The HI exponential kernel is much less correlated in frequency than either foreground kernel, and the smooth foreground RBF kernel is the most correlated in frequency. From the amplitude of the colour bar, the HI exponential kernel has the smallest variance, while the smooth foreground RBF kernel has the largest variance (and consequently signal amplitude), as expected.



**Figure 7.** Posterior distribution of our model hyperparameters, obtained using nested sampling in the with polarization case. Dotted grey lines show the best-fitting hyperparameter results obtained with gradient descent. [URL](#).

We perform foreground removal with this best-fitting GPR model, and plot our residual power spectra in Fig. 9. We also show the true residual power spectrum, as well as the residual obtained with PCA (now looking at the cases with  $N_{\text{FG}} = 7$  and  $N_{\text{FG}} = 6$ , since the polarization leakage foregrounds require a larger  $N_{\text{FG}}$  to be properly removed). The bottom panel shows the percentage residual difference between the foreground cleaned and true residuals.

For the spherically averaged power spectrum, Fig. 9 shows that GPR performs better than both PCA cases on small scales, since the per cent residuals are closer to zero and it does not overestimate the power spectrum. The residuals get gradually worse for larger scales, until they diverge beyond 10 per cent below a  $k$ -bin of  $k = 0.09 h \text{ Mpc}^{-1}$ . This is a larger limit than in the no polarization case, indicating that the polarized foregrounds make the cleaning more difficult. From the residuals, PCA ( $N_{\text{FG}} = 7$ ) is able to access larger scales than GPR. PCA ( $N_{\text{FG}} = 6$ ) overestimates the residual power spectrum on both small and large scales.

For the radial power spectrum, GPR is once again good at recovering the truth, with residuals below 10 per cent down to  $k_{\parallel} = 0.03 h \text{ Mpc}^{-1}$  (a larger cut-off than in the no polarization case). The PCA residuals diverge after this point, meaning GPR can access larger scales. For the PCA  $N_{\text{FG}} = 6$  case, influence from residual foregrounds is strong and they are particularly dominant in the second smallest  $k_{\parallel}$  mode. The transverse power spectrum results are dramatically worse than in the no polarization case. Neither PCA or GPR recover the transverse power spectrum of the true signal well. This is probably due to how the foreground removal works in both cases: it mainly takes into account frequency (radial) information, so it is worse at recovering the true residual transverse power spectrum.

Once again, we see in Fig. 9 that including the bias correction leads to an overestimation of the spherically averaged and radial power spectra on large scales, and no clear difference on small scales.

To better understand how well our best-fitting kernel functions are describing the signals in our data, we compare the true frequency covariance of our signals to what GPR predicts. In Fig. 10,

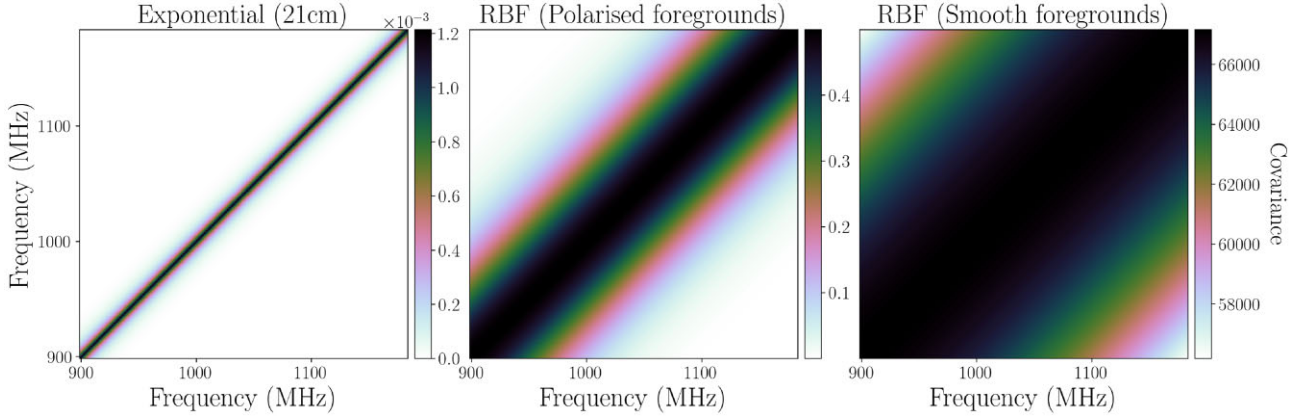
we look at two different cases of what GPR recovers, described below.

First, we take the RBF kernel for the smooth foregrounds and the RBF kernel for the polarized foregrounds, and use our full data to predict what the foreground signal looks like, i.e. we are estimating  $E[f_{\text{fg}}]$ . This is what we have already done previously to predict the foreground signal and remove it from our data, but we are focusing now on what this prediction looks like. We do this also for the smooth foreground RBF kernel, to predict what the smooth foregrounds look like (i.e.  $E[f_{\text{smooth}}]$ ). We do the same for the polarized foregrounds RBF kernel, obtaining an estimate for what our polarized foregrounds look (i.e.  $E[f_{\text{pol}}]$ ). Finally, we also do this for the HI signal and indistinguishable noise kernel. These are denoted as a ‘full data fit’, since we are using the kernels and the full data to predict the signals. This ‘full data fit’ is what we have already done to remove foregrounds using GPR, but here we are focusing on what the signal predictions look like, and how it compares to the true signal. We are also considering a new case of only fitting the smooth foreground kernel, and also only fitting the polarized foreground kernel, to try to predict these signals in isolation.

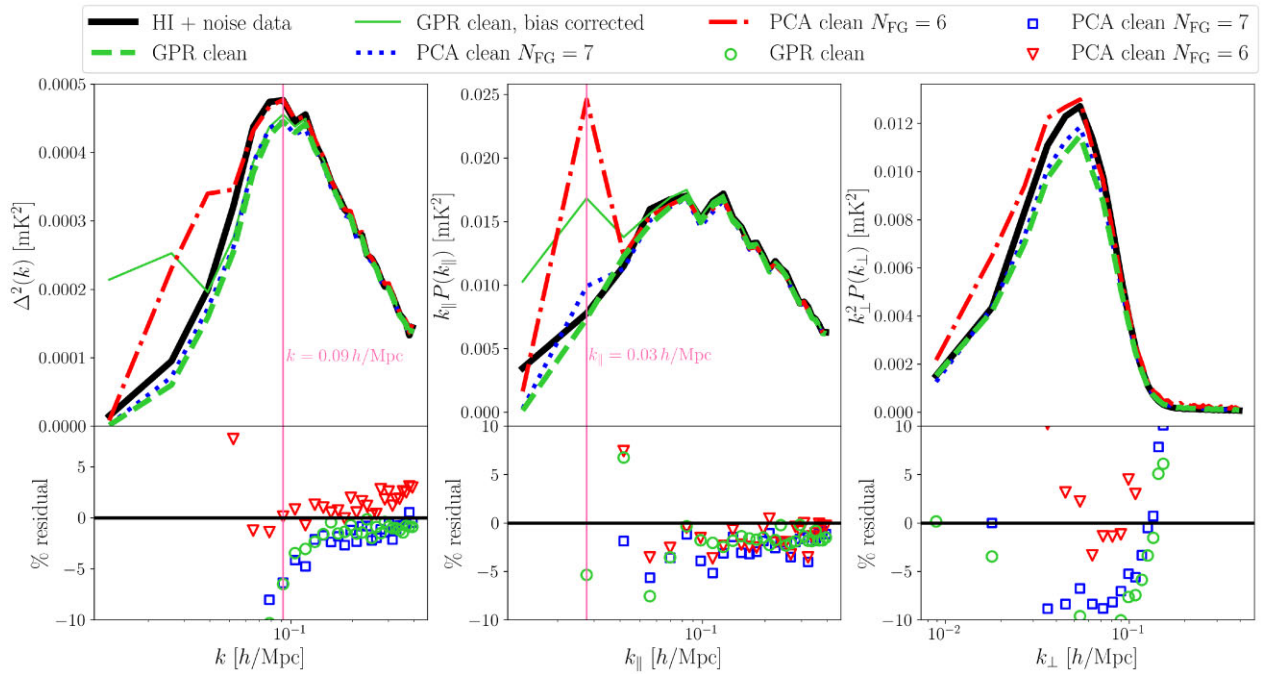
We also consider an ‘individual fit’, where instead of fitting our kernels to our full data, we fit them to the appropriate data *only*. For the RBF + RBF full foreground kernel, we fit this to the smooth + polarized foreground data. For the RBF smooth foreground kernel, we fit it only to the smooth foreground data, and similarly for the polarized foreground case. For the Exponential kernel, we fit it only to the HI and noise data. This ‘individual fit’ is a new analysis, which we are only doing for understanding GPR better, and it does not play a part in our foreground removal pipeline. The only difference from the ‘full data fit’ is that instead of calculating the signal estimates using the full data, we only use the relevant signal data (i.e. we are predicting the foregrounds given the foreground kernels and foreground data, without the need to separate it from the HI signal or noise).

We expect that the ‘individual fit’ will give predictions that are closer to the truth than the ‘full data fit’, since in this case we are not trying to separate our signals, but are instead fitting to the isolated signals. This is indeed what we find in Fig. 10, where we plot the frequency covariance of our predictions as well as the real covariance. The left-hand panel (‘individual fit’) looks similar to the right-hand panel (truth), while the middle panel (‘full data fit’) shows some disagreement. For the all foregrounds and smooth foregrounds case, the difference between these cases is not too stark. However, for the polarized foregrounds and HI + noise signal it is. We can see that the ‘full data fit’ for the polarized foregrounds is not great, since it does not match the truth so well. The fact that the ‘full data fit’ is bad, but the ‘individual fit’ is good means that the issue is not with our best-fitting foreground kernels. The issue arises when we try to pick out the polarized signal in the presence of other signals – this makes it easier for the signals to be confused with each other, and leads to more uncertainty. Our foreground kernels are good fits for the signals individually, but it is difficult to isolate the polarized foregrounds in the presence of other signals.

The difficulty in recovering the HI + noise signal in the ‘full data fit’ case shows that foreground removal in the presence of polarized foregrounds is more difficult. In the ‘individual fit’ case, the HI + noise covariance recovery matches the truth well, but when considering all signals together in the ‘full data fit’, this is no longer true. This suggests that the HI + noise signal is being confused with other signals, much like the polarized foregrounds. This confusion and inability to separate signals makes the final foreground prediction more uncertain, and ultimately hinders the foreground removal process.



**Figure 8.** Frequency covariance generated using our best-fitting kernel functions, for each signal component. These are generated for the frequency range of our data, using our best estimate for the kernel hyperparameters from nested sampling. [URL](#)



**Figure 9.** Power spectra results for the case including polarization. *Top:* True HI + noise signal (black solid line), PCA foreground cleaned residuals (blue dotted line for  $N_{\text{FG}} = 7$  and red dash-dotted line for  $N_{\text{FG}} = 6$ ), and GPR foreground cleaned residuals without bias correction (green dashed line) and with (thin green solid line). *Bottom:* Percentage residual difference between the foreground removed residual power spectra and the true HI power spectra. *Left:* Spherically averaged power spectrum. *Centre:* Radial power spectrum. *Right:* Transverse power spectrum. [URL](#)

### 4.3 Bandwidth and redshift dependence

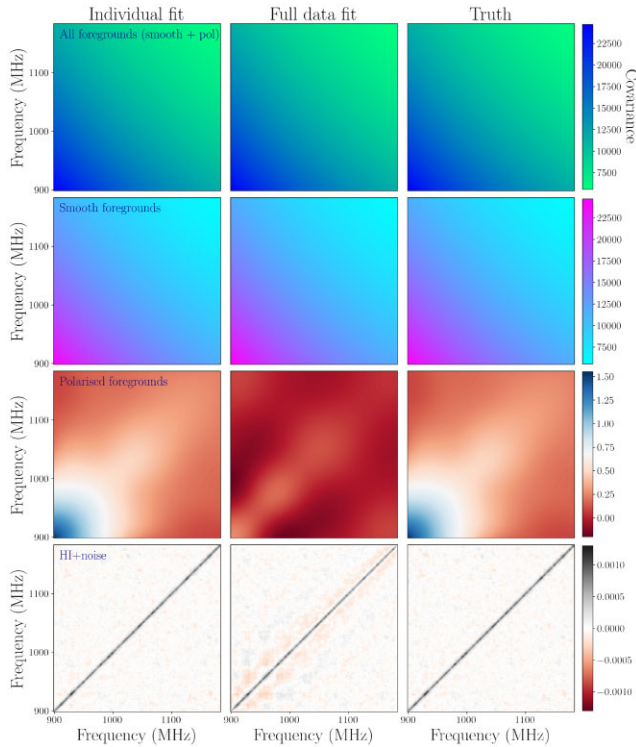
We wanted to test how our results would differ if we cut our data’s bandwidth in half, i.e. only considered half of our frequency range. Since GPR uses frequency information, it might be more difficult for it to fit the data well if there is less frequency data to learn from. Alternatively, as suggested by Hothi et al. (2020), it might be better to consider narrower frequency ranges in the case where the signal’s length-scale is changing with frequency, that way it avoids being ‘averaged out’ as much.

We test this in the case of including polarized foregrounds, and again find the best model to be Exponential (HI signal kernel, also describing indistinguishable noise) + RBF (smooth foreground kernel) + RBF (polarized foreground kernel) + indistinguishable noise assumed throughout. The only difference now is that when we

predict the foreground signal from our data using this model, we only consider half of the frequency range. When we split our data in half along frequency, we also compare both halves, to see if the redshift/frequency of observation makes a difference.

We again use nested sampling to find the best-fitting hyperparameters of our model, and note down the median and  $1\sigma$  of the distributions in Table 1. We then perform foreground removal using GPR, and plot the residual power spectra results for the low-frequency, high-redshift case in Fig. 11, and for the high-frequency, low-redshift case in Fig. 12.

For the low-frequency case, we show the  $N_{\text{FG}} = 5$  and  $N_{\text{FG}} = 4$  cases for PCA. For the spherical power spectra of Fig. 11, GPR tends to lie closer to the true residual power spectrum line, indicating it is performing better than PCA. In small scales, both PCA cases



**Figure 10.** Frequency covariance of our signals, as predicted by GPR and compared to the truth. We look at all foregrounds together (*top row*), the smooth foregrounds only (*second row*), the polarized foregrounds only (*third row*), and the HI + noise only (*bottom row*). *Left*: GPR prediction of what the signal frequency covariance looks like, given our chosen kernels and based on the relevant signal data only. *Centre*: GPR prediction of what the signal frequency covariance looks like, given our chosen kernels and based on our full data. *Right*: The true frequency covariance of our data. [URL](#)

overestimate the true residual power spectrum, while GPR slightly underestimates. GPR recovers the power spectrum within 10 per cent residuals down to a  $k$ -bin of  $k = 0.08 h \text{ Mpc}^{-1}$ , smaller than the full bandwidth case, indicating that it can now access larger scales. GPR also recovers the radial power spectrum nicely, but not the transverse power spectrum, as before.

For the high-frequency case, we show the PCA results for  $N_{\text{FG}} = 4$  and  $N_{\text{FG}} = 3$ . GPR does not outperform PCA in this case, but agrees nicely with the PCA  $N_{\text{FG}} = 4$  recovery throughout. It now recovers the spherically averaged power spectrum within 10 per cent residual down to  $k = 0.1 h \text{ Mpc}^{-1}$ , which is larger than the low-frequency case and full bandwidth case. GPR performs worse here than in the low-frequency case. As seen in Fig. 1, the amplitude of our foregrounds is higher in the lower frequency range than in the higher frequency range, and this could make a difference in our GPR foreground removal results.

In both cases, including the bias correction results in an overestimation of the spherically averaged and radial power spectra on large scales, but some discernible difference on medium to small scales that makes it closer to the truth. It is interesting that the low-frequency half experiences a more extreme bias correction.

Hothi et al. (2020) found that GPR performs worse in larger bandwidths. Here, this is not so clear – the low-frequency case can access larger scales than the full bandwidth case, but the high-frequency case performs worse than the full bandwidth one. Hothi et al. (2020) considers interferometric EoR simulations of HI signal, which has different systematics to our single-dish, low-

redshift IM simulations. This, and the fact that their noise level is much larger than ours could explain why we do not see the same effects.

#### 4.4 RFI effects

It is common in HI IM experiments to have to remove whole frequency channels due to them being contaminated with RFI. We test how GPR performs when frequency channels are missing, as compared to PCA, in the case of including polarized foregrounds. To do this, we use the method described in Carucci et al. (2020), where three different cases are considered in the context of the sparsity-based foreground removal algorithm generalized morphological component analysis (GMCA; Chapman et al. 2012):

(i) **Case 1:** The case of removing 40 per cent of the channels in one chunk in the middle of the frequency range, such that we have in order: 30 per cent good channels, 40 per cent bad (missing) channels, and 30 per cent good channels.

(ii) **Case 2:** The case of removing 40 per cent of the channels in one chunk at the beginning of the frequency range, such that we have in order: 10 per cent good channels, 40 per cent bad (missing) channels, and 50 per cent good channels.

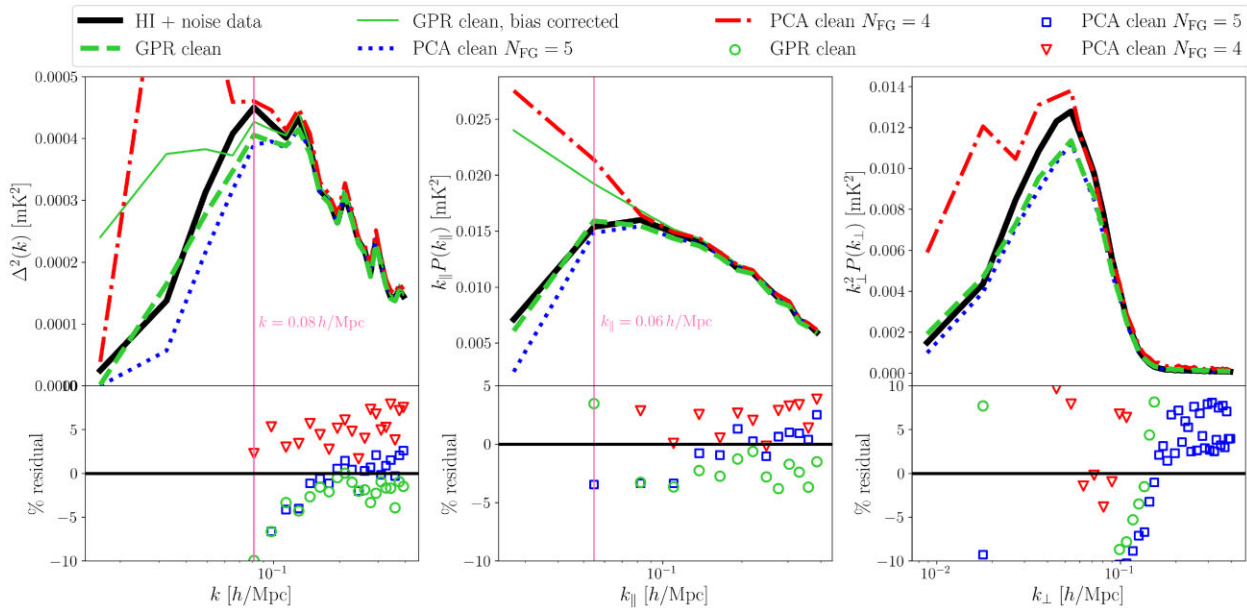
(iii) **Case 3:** The case of removing 40 per cent of the frequency channels, in a few different chunks throughout the frequency range, such that we have in order: 20 per cent good channels, 30 per cent bad (missing) channels, 20 per cent good channels, 10 per cent bad (missing) channels, and 20 per cent good channels.

Once again we used the kernel model: Exponential (HI signal kernel, also describing indistinguishable noise) + RBF (smooth foreground kernel) + RBF (polarized foreground kernel) + indistinguishable noise assumed throughout, and ran nested sampling to obtain the best-fitting kernel hyperparameters (see Table 1 for the best-fitting median and  $1\sigma$  values). The only difference from our full data case is that here we are ignoring certain channel when running GPR and performing foreground removal. We are not setting them to zero, but instead removing the channels entirely from the data, thus reducing the number of frequency channels  $N_{\nu}$ .

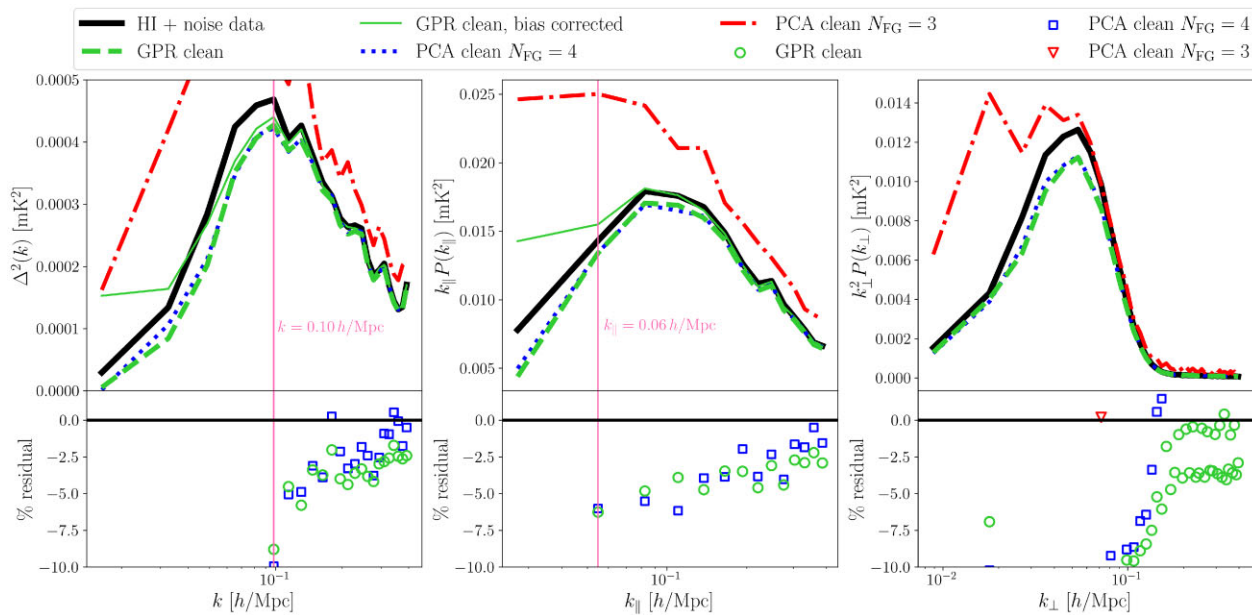
We plot our resulting power spectra for each of these different cases in Fig. 13. We show the cases of  $N_{\text{FG}} = 6$  and  $N_{\text{FG}} = 5$  for PCA. For the spherically averaged power spectra on the top row, GPR performs worse than it did in the no missing channels case, and either performs the same as or worse than the best PCA case. The same is true for the radial power spectrum in the middle row. The bias correction in this case does improve the spherically averaged and radial power spectra results, without leading to an overestimation on large scales as seen previously.

The foreground removal pipeline for GPR depends on first modelling the foreground signal as a function of frequency, so having gaps in the input data significantly decreases the continuity and quality of information GPR has to learn from. This in turn leads to worse foreground removal than in the no missing channels case, and worse foreground removal than PCA in many cases, since PCA does not have this strong dependency on modelling the foregrounds as a function of frequency.

Interestingly, the transverse power spectrum does not see a dramatic difference, and looks similar to the recovered transverse power spectra in the no missing channels case. This is not surprising,



**Figure 11.** Power spectra results for the low-frequency half-bandwidth case including polarization. *Top:* True HI + noise signal (black solid line), PCA foreground cleaned residuals (blue dotted line for  $N_{\text{FG}} = 5$  and red dash-dotted line for  $N_{\text{FG}} = 4$ ), and GPR foreground cleaned residuals without bias correction (green dashed line) and with (thin green solid line). *Bottom:* Percentage residual difference between the foreground removed residual power spectra and the true HI power spectra. *Left:* Spherically averaged power spectrum. *Centre:* Radial power spectrum. *Right:* Transverse power spectrum. [URL](#)



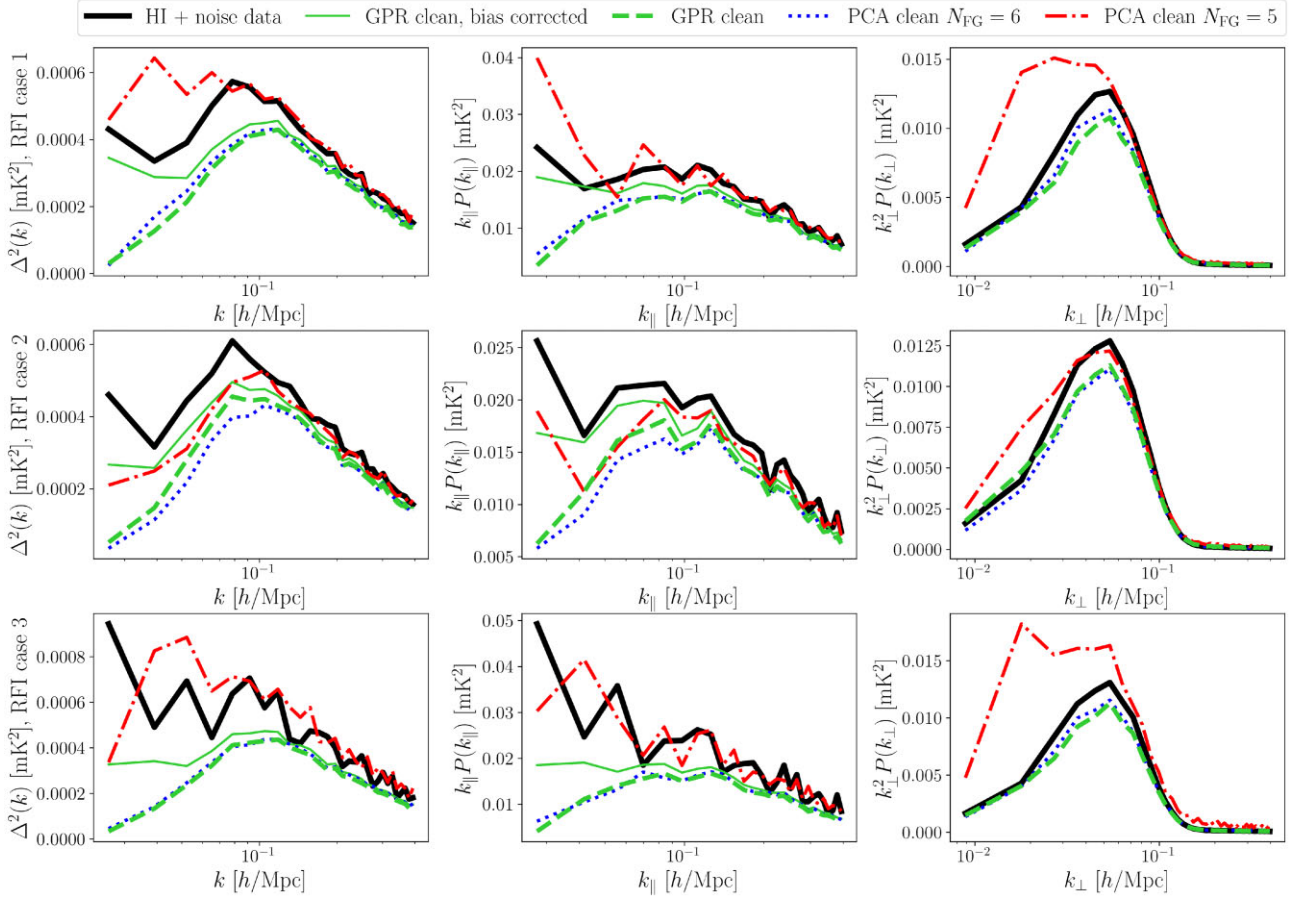
**Figure 12.** Power spectra results for the high-frequency half-bandwidth case including polarization. *Top:* True HI + noise signal (black solid line), PCA foreground cleaned residuals (blue dotted line for  $N_{\text{FG}} = 4$  and red dash-dotted line for  $N_{\text{FG}} = 3$ ), and GPR foreground cleaned residuals without bias correction (green dashed line) and with (thin green solid line). *Bottom:* Percentage residual difference between the foreground removed residual power spectra and the true HI power spectra. *Left:* Spherically averaged power spectrum. *Centre:* Radial power spectrum. *Right:* Transverse power spectrum. [URL](#)

since there is no gap of information in the spatial direction, only in the frequency direction.

While both GPR and PCA are negatively impacted by missing frequency channels, Carucci et al. (2020) found that GMCA is still able to recover the true radial power spectrum in the presence of these complications. Carucci et al. (2020) also tested this for the Independent Component Analysis algorithm FastICA and found that,

much like PCA and GPR, it is compromised by missing frequency channels.

Offringa et al. (2019) studied how missing frequency channels due to RFI affect GPR as a foreground removal technique in the context of EoR studies. They use an interpolation scheme to correct for the flagged RFI channels, which differs from our method of simply excluding the channels, and found that RFI flagging leads to excess



**Figure 13.** Power spectra results for the RFI-like missing channels cases including polarization. For all cases, 40 per cent of the frequency channels are missing, but each row represents a different distribution of missing channels in the data. We show the true HI + noise signal (black solid line), PCA foreground cleaned residuals (blue dotted line for  $N_{\text{FG}} = 6$  and red dash-dotted line for  $N_{\text{FG}} = 5$ ), and GPR foreground cleaned residuals without bias correction (green dashed line) and with (thin green solid line). *Left:* Spherically averaged power spectrum. *Centre:* Radial power spectrum. *Right:* Transverse power spectrum. [URL](#)

power in the final foreground removed power spectrum. We leave investigation of how interpolation methods for missing frequency channels would affect our results to future work.

#### 4.5 Foreground transfer function

Implementing a foreground removal transfer function is a common way of mitigating foreground removal effects that might be present in the residual power spectrum (Switzer et al. 2015). It is usually done by injecting HI IM mocks into the data, performing the foreground removal, and cross-correlating the result with the original mock, in order to obtain an estimate for how the signal has been biased (Masui et al. 2013; Switzer et al. 2013, 2015; Anderson et al. 2018; Cunnington et al. 2021; Wolz et al. 2021). This works well for blind foreground removal methods such as PCA, since the mocks should be as uncorrelated in frequency as the true HI signal. Therefore, both the mocks and the true HI signal will lose similar modes (the ones most degenerate with the foregrounds) in the foreground clean.

Here, we begin to investigate whether this transfer function technique could work for GPR. With GPR, we are modelling each component of our signal with a different kernel, and so we wanted to check if we would need to adapt our kernel model in the presence of mock data. We use a lognormal simulation of cosmological HI

signal as our mock. We do this in the case of including polarized foregrounds.

We generated a lognormal realization of our cosmological signal (a method first proposed by Coles & Jones 1991), in the same way as described in e.g. Cunnington et al. (2020b). We then smooth it by the same telescope beam as before, and add it to our data. Our data is now composed of the smooth and polarized foregrounds, our original HI signal, this lognormal HI realization, and noise.

We tested whether the model exponential (HI signal kernel, also describing indistinguishable noise) + RBF (smooth foreground kernel) + RBF (polarized foreground kernel) + indistinguishable noise assumed throughout would still work well with this data, or if we need an extra exponential kernel to describe the lognormal realization. By comparing the evidence in both cases, we find that the original model with only one exponential kernel is preferred, probably due to how difficult it is for GPR to pick out very similar signals, making it better to model them together (this is similar to how it is better to model the HI signal and noise together with one exponential kernel).

We ran nested sampling using our original model and this new data including a lognormal realization. The median and  $1\sigma$  distributions of our hyperparameters can be found in Table 1. The variance of the exponential kernel is larger than for the no-lognormal case (almost double), which makes sense since we have essentially doubled our

HI signal amplitude. The length-scale remains constant, because the signals are similar.

We compared how GPR foreground removal works in this case, to the no-lognormal case. We perform foreground removal on our data with the lognormal signal, using this best-fitting model, and find that the residual power spectrum looks extremely similar in shape to the no-lognormal case, only changing in amplitude since it includes the extra lognormal realization. Adding a lognormal realization of signal does not make the foreground removal any worse, or overall different. Most importantly, it does not require a new or different kernel function to describe it. This is good, since it means we can use the foreground transfer function technique in the context of GPR. Further work is required to determine whether it is viable to apply both a bias correction *and* a transfer function to GPR residuals.

## 5 CONCLUSIONS

The take-home message of this paper is that GPR can be used as a foreground removal technique for low redshift, single-dish HI intensity mapping. We presented a pipeline for performing GPR in this case, which uses the data to optimize and choose a kernel model. For PCA, we had to fine-tune the  $N_{\text{FG}}$  parameter for each case considered, and found that even just decreasing the bandwidth requires a different  $N_{\text{FG}}$  choice. This is a problem for real data, where we do not know the truth and have to make an educated guess for what the best  $N_{\text{FG}}$  choice is. GPR operates differently – it uses the data to determine the best choice for the kernel model, and then performs the foreground removal. We do not have to make an educated guess with GPR, as the Bayes factor analysis guides our final choice. This poses a significant advantage over blind foreground removal methods that require a fine-tuning parameter such as  $N_{\text{FG}}$  (e.g. PCA, FastICA, GMCA).

We presented the publicly available and user friendly `PYTHON` package `GPR4IM`,<sup>6</sup> which can be used to run GPR as a foreground removal technique in any real space HI intensity map, either simulated or real. Although we considered here the single-dish HI IM case, our code could also be used in the context of interferometric data, and higher redshift signal.

Our main findings can be summarized as follows:

(i) GPR may be used as a foreground removal technique in the single-dish, low redshift HI IM case. Without polarization leakage in the foregrounds, it recovers the true residual power spectrum within 10 per cent residual difference down to a  $k$ -bin of  $k = 0.07 h \text{ Mpc}^{-1}$ . With polarization leakage, it does so down to  $k = 0.09 h \text{ Mpc}^{-1}$ , since the polarized foregrounds make the removal more difficult;

(ii) In the no polarization case, GPR recovers the BAO scales better than PCA in the radial power spectrum, but worse in the transverse case. For the spherically averaged power spectrum, GPR recovers the small BAO scales best, but PCA does better on the larger BAO scales. The same is true for the case including polarization, except for the radial power spectrum, where GPR no longer outperforms PCA.

(iii) GPR outperforms PCA at recovering the true residual power spectrum on small scales, and both fail on large scales. This is no longer true when the bandwidth is halved of frequency channels are missing.

(iv) GPR is very good at recovering the true residual radial power spectrum, more so than PCA in the full bandwidth case. Both GPR

and PCA are worse at recovering the transverse power spectrum, due to how they mainly take into account frequency information;

(v) GPR prefers lower frequencies. When splitting the bandwidth of our data in half, GPR performs much better in the low-frequency (high redshift) domain, where the foregrounds are brighter, and recovered the power spectrum within 10 per cent residual down to a  $k = 0.08 h \text{ Mpc}^{-1}$ . In the high-frequency case, this went up to  $k = 0.1 h \text{ Mpc}^{-1}$ ;

(vi) In the presence of missing channels due to RFI contamination, both PCA and GPR suffer, and GPR either performs the same as or worse than PCA on all scales. This is clear for the power spectrum and radial power spectrum. The transverse power spectrum is not much affected in any case.

(vii) The Mertens et al. (2020) bias correction causes an over-estimation of the spherically averaged and radial power spectrum on large scales. However, in the case when frequency channels are missing, the bias correction helps us recover the truth better. When halving the bandwidth, the bias correction improves results on medium to small scales.

(viii) Adding a lognormal realization of HI cosmological signal does not require an additional kernel, and does not change how GPR performs as a foreground removal technique. As such, it should be possible to apply a foreground removal transfer function with GPR.

We hope that these results are useful for implementing GPR as a foreground removal technique in the single-dish, low redshift HI IM case going forward. Future plans include performing a foreground removal method comparison study like this one on real data to understand how GPR performs in the context of real HI IM data.

It would also be interesting to try different power spectrum bias correction techniques on the residual power spectrum, such as the one described in Kern & Liu (2020). It could also be beneficial to try to incorporate the error in the hyperparameter posterior distributions obtained with nested sampling into the residual power-spectrum estimation.

Since the smooth foreground signal is actually composed of three different signals (Galactic synchrotron, free-free and point source emission), it would also be interesting to check whether three kernels instead of one are best for describing the smooth foregrounds.

Finally, we reiterate that our kernel models used in this paper are overly optimistic since our simulations are idealized, and a full nested sampling model selection procedure should be performed when choosing the best-fitting kernels for real data.

## ACKNOWLEDGEMENTS

We are grateful to the reviewer for their extremely helpful comments that improved the quality of this manuscript. We thank Ian Hothi, Phil Bull, Florent Mertens, Isabella Carucci, Mario Santos, José Fonseca for helpful feedback and Andrei Mesinger, Fraser Kennedy and Ioannis Patras for useful discussions. PS is supported by the Science and Technology Facilities Council (grant number ST/P006760/1) through the DISCnet Centre for Doctoral Training. CW's research for this project was supported by a UK Research and Innovation Future Leaders Fellowship, grant number MR/S016066/1. SC is supported by STFC grant ST/S000437/1. AP is a UK Research and Innovation Future Leaders Fellow, grant MR/S016066/1, and also acknowledges support by STFC grant ST/S000437/1. This research utilized Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT <http://doi.org/10.5281/zenodo.438045>. We acknowledge the use of open source software (Jones et al. 2001; Hunter 2007; McKinney 2010; van der Walt, Colbert & Varoquaux 2011; Lewis, Challinor &

<sup>6</sup>[github.com/paulassoares/gpr4im](https://github.com/paulassoares/gpr4im)



Lasenby 2000; Lewis 2019). Some of the results in this paper have been derived using the `HEALPY` and `HEALPIX` package. We thank New Mexico State University (USA) and Instituto de Astrofísica de Andalucía CSIC (Spain) for hosting the Skies & Universes site for cosmological simulation products.

## DATA AVAILABILITY

The data and code underlying this article can be found [here](#).

## REFERENCES

- Alonso D., Ferreira P. G., Santos M. G., 2014a, *MNRAS*, 444, 3183
- Alonso D., Bull P., Ferreira P. G., Santos M. G., 2014b, *MNRAS*, 447, 400
- Anderson C. J. et al., 2018, *MNRAS*, 476, 3382
- Ansari R., Campagne J.-E., Colom P., Magneville C., Martin J.-M., Moniez M., Rich J., Yeche C., 2012, *Comptes Rendus Physique*, 13, 46
- Battye R. A., Davies R. D., Weller J., 2004, *MNRAS*, 355, 1339
- Battye R. A., Browne I. W. A., Dickinson C., Heron G., Maffei B., Pourtsidou A., 2013, *MNRAS*, 434, 1239
- Bharadwaj S., Nath B. B., Sethi S. K., 2001, *J. Astrophys. Astron.*, 22, 21
- Bigot-Sazy M.-A. et al., 2015, *MNRAS*, 454, 3240
- Blake C., Carter P., Koda J., 2018, *MNRAS*, 479, 5168
- Buchner J. et al., 2014, *A&A*, 564, A125
- Bull P., Ferreira P. G., Patel P., Santos M. G., 2015, *ApJ*, 803, 21
- Carucci I. P., Irfan M. O., Bobin J., 2020, *MNRAS*, 499, 304
- Castorina E., White M., 2018, *MNRAS*, 476, 4403
- Chang T.-C., Pen U.-L., Peterson J. B., McDonald P., 2008, *Phys. Rev. Lett.*, 100
- Chang T.-C., Pen U.-L., Bandura K., Peterson J., 2010, *Nature*, 466, 463
- Chapman E. et al., 2012, *MNRAS*, 429, 165
- Coles P., Jones B., 1991, *MNRAS*, 248, 1
- Croton D. J. et al., 2016, *ApJS*, 222, 22
- Cunnington S., Pourtsidou A., Soares P. S., Blake C., Bacon D., 2020a, *MNRAS*, 496, 415
- Cunnington S., Camera S., Pourtsidou A., 2020b, *MNRAS*, 499, 4054
- Cunnington S., Irfan M. O., Carucci I. P., Pourtsidou A., Bobin J., 2021, *MNRAS*, 504, 208
- Dickinson C., Davies R. D., Davis R. J., 2003, *MNRAS*, 341, 369
- Feroz F., Hobson M. P., Bridges M., 2009, *MNRAS*, 398, 1601
- Gehlot B. K. et al., 2019, *MNRAS*, 488, 4271
- Ghosh A. et al., 2020, *MNRAS*, 495, 2813
- GPY since, 2012, GPY: A Gaussian Process Framework in Python, GitHub, <http://github.com/SheffieldML/GPY>
- Hothi I. et al., 2020, *MNRAS*, 500, 2264
- Hunter J. D., 2007, *Comput. Sci. Eng.*, 9, 90
- Jelic V. et al., 2008, *MNRAS*, 389, 1319
- Jelic V., Zaroubi S., Labropoulos P., Bernardi G., de Bruyn A. G., Koopmans L. V. E., 2010, *MNRAS*, 409, 1647
- Jones E. et al., 2001, SciPy: Open Source Scientific Tools for Python. <http://www.scipy.org/>
- Kennedy F., Bull P., 2021, *MNRAS*, 506, 2638
- Kern N., Liu A., 2020, *MNRAS*, 501, 1463
- Klypin A., Yepes G., Gottlober S., Prada F., Hess S., 2016, *MNRAS*, 457, 4340
- Knebe A. et al., 2018, *MNRAS*, 474, 5206
- Kovetz E. D. et al., 2017, Line-Intensity Mapping: 2017 Status Report. preprint ([arXiv:1709.09066](https://arxiv.org/abs/1709.09066))
- Lewis A., 2019, Getdist: A Python Package for Analysing Monte Carlo Samples. preprint ([arXiv:1910.13970](https://arxiv.org/abs/1910.13970))
- Lewis A., Challinor A., Lasenby A., 2000, *ApJ*, 538, 473
- Li Y., Santos M. G., Grainge K., Harper S., Wang J., 2020, *MNRAS*, 501, 4344
- Liao Y.-W., Chang T.-C., Kuo C.-Y., Masui K. W., Oppermann N., Pen U.-L., Peterson J. B., 2016, *ApJ*, 833, 289
- Liu A., Shaw J. R., 2020, *PASP*, 132, 062001
- Liu A., Tegmark M., 2011, *Phys. Rev. D*, 83, 103006
- Luger R., Foreman-Mackey D., Hedges C., 2021, *AJ*, 162, 124
- Masui K. W. et al., 2013, *ApJ*, 763, L20
- Matshawule S. D., Spinelli M., Santos M. G., Ngobese S., 2021, *MNRAS*, 506, 5075
- McKinney W., 2010, in van der Walt S., Millman J., eds, Proceedings of the 9th Python in Science Conference, Data Structures for Statistical Computing in Python. Austin, Texas, p. 56
- Mertens F. G., Ghosh A., Koopmans L. V. E., 2018, *MNRAS*, 478, 3640
- Mertens F. G. et al., 2020, *MNRAS*, 493, 1662
- Miville-Deschenes M.-A., Ysard N., Lavabre A., Ponthieu N., Mac'ias-Pérez J., Aumont J., Bernard J., 2008, *A&A*, 490, 1093
- Moore D. F., Aguirre J. E., Parsons A. R., Jacobs D. C., Pober J. C., 2013, *ApJ*, 769, 154
- Offringa A. R., Mertens F., Koopmans L. V. E., 2019, *MNRAS*, 484, 2866
- Olivari L. C., Remazeilles M., Dickinson C., 2015, *MNRAS*, 456, 2749
- Olivari L. C., Dickinson C., Battye R. A., Ma Y.-Z., Costa A. A., Remazeilles M., Harper S., 2017, *MNRAS*, 473, 4242
- Peterson J. B. et al., 2009, 21 cm Intensity Mapping. preprint ([arXiv:0902.3091](https://arxiv.org/abs/0902.3091))
- Planck Collaboration et al., 2016, *A&A*, 594, A13
- Pourtsidou A., Bacon D., Crittenden R., 2017, *MNRAS*, 470, 4251
- Rasmussen C. E., Williams C. K. I., 2006, Gaussian Processes for Machine Learning. The MIT Press, Cambridge, MA
- Remazeilles M., Dickinson C., Banday A. J., Bigot-Sazy M. A., Ghosh T., 2015, *MNRAS*, 451, 4311
- Santos M. G., Cooray A., Knox L., 2005, *ApJ*, 625, 575
- Santos M. G. et al., 2015, Cosmology with a SKA Hi Intensity Mapping Survey. preprint ([arXiv:1501.03989](https://arxiv.org/abs/1501.03989))
- Santos M. G. et al., 2017, MeerKAT Large Area Synoptic Survey. preprint ([arXiv:1709.06099](https://arxiv.org/abs/1709.06099))
- Seo H.-J., Dodelson S., Marriner J., McGinnis D., Stebbins A., Stoughton C., Vallinotto A., 2010, *ApJ*, 721, 164
- Simpson F., Lalchand V., Rasmussen C. E., 2020, Marginalised Gaussian Processes with Nested Sampling. preprint ([arXiv:2010.16344](https://arxiv.org/abs/2010.16344))
- Soares P. S., Cunnington S., Pourtsidou A., Blake C., 2021, *MNRAS*, 502, 2549
- Switzer E. R. et al., 2013, *MNRAS*, 434, L46
- Switzer E. R., Chang T.-C., Masui K. W., Pen U.-L., Voytek T. C., 2015, *ApJ*, 815, 51
- van der Walt S., Colbert S. C., Varoquaux G., 2011, *Comput. Sci. Eng.*, 13, 22
- Villaescusa-Navarro F., Alonso D., Viel M., 2016, *MNRAS*, 466, 2736
- Villaescusa-Navarro F. et al., 2018, *ApJ*, 866, 135
- Wang J. et al., 2020, *MNRAS*, 505, 3698
- Wolz L. et al., 2015, Foreground Subtraction in Intensity Mapping with the SKA. preprint ([arXiv:1501.03823](https://arxiv.org/abs/1501.03823))
- Wolz L. et al., 2016, *MNRAS*, 464, 4938
- Wolz L. et al., 2021, HI Constraints from the Cross-Correlation of eBOSS Galaxies and Green Bank Telescope Intensity Maps. preprint ([arXiv:2102.04946](https://arxiv.org/abs/2102.04946))

## APPENDIX: SIMULATIONS

### A1 Smooth foregrounds

Here we describe how we generate the smooth foregrounds in our simulation. The most abundant smooth foregrounds that pose problems in HI IM experiments are synchrotron emission, free-free emission and point sources. The synchrotron emission comes from our Galaxy's magnetic field accelerating cosmic ray electrons, causing them to emit radiation. The free-free emission comes mainly from ions in our Galaxy causing free electrons to scatter, but some extragalactic contribution to this signal is present. The point sources represent extragalactic objects which emit radio signals, such as Active Galactic Nuclei.

For the Galactic synchrotron and Galactic free–free emission, we use the Planck Legacy Archive<sup>7</sup> FFP10 simulations. For more detail on these simulations, please see C21.

### A1.1 Synchrotron emission

To generate maps of Galactic synchrotron emission, we take the FFP10 maps at frequencies of 217 and 353 GHz. These maps are derived from the source-subtracted and destriped 408 MHz all-sky map, which is limited by a resolution of 56 arcmin but this  $N_{\text{side}} = 2048$  version uses Gaussian random field realizations to fill in the higher resolution details (Remazeilles et al. 2015).

The synchrotron spectral index map can be determined from the 217 and 353 GHz synchrotron maps, which use the ‘Model 4’ synchrotron spectral index map (Miville-Deschenes et al. 2008). This has a resolution of approximately 5 degrees, so we must fill in the higher resolution details. To do this, we use a Gaussian random field, using the synchrotron scaling relation in Santos, Cooray & Knox (2005). We then use this spectral index map to interpolate the synchrotron emission at our frequency range.

### A1.2 Free-free emission

For simulating the Galactic free–free emission, we once again make use of the FFP10 maps, specifically the 217 GHz free–free simulation which has  $N_{\text{side}} = 2048$  (Miville-Deschenes et al. 2008). This map is made using the free–free template from Dickinson, Davies & Davis (2003) as well as the ones from WMAP MEM. Taking the free–free amplitude ( $a_{\text{free}}$ ) from this map, we then simulate the free–free emission using a power law

$$T_{\text{free}} = a_{\text{free}} \left( \frac{\nu}{\nu_0} \right)^{\beta_{\text{free}}}, \quad (\text{A1})$$

where  $\beta_{\text{free}}$  is the free–free spectral index, which is constant for all pixels.

### A1.3 Point sources

The point sources are generated by fitting a polynomial to radio sources at 1.4 GHz, using the model described in Battye et al. (2013). We then use the method described in Olivari et al. (2017) to scale this signal down to our frequencies. This method uses a power law, where the spectral index is described by a Gaussian distribution with a mean of  $-2.7$  and standard deviation of  $0.2$ . Our source extraction has an upper bound of 100 mJy.

## A2 Polarization leakage

Polarization leakage arises from the magnetic fields present in our Galaxy’s interstellar medium. When light interacts with these fields, Faraday rotation can change its polarization angle. This happens for our Galactic synchrotron emission, and some of its emission will be leaked from Stokes Q and Stokes U to Stokes I. This leakage will change with frequency because Faraday rotation is frequency dependent (Jelic et al. 2010; Moore et al. 2013). Any frequency-dependent variation to the foreground signal will make foreground removal more difficult, specifically for methods that rely on foregrounds being smooth in frequency.

To simulate this effect, we use CRIME,<sup>8</sup> a software which simulates Stokes Q emission maps for a given frequency. We generate this for our frequency range, and choose the polarization leakage level of the Stokes Q signal to be 0.5 per cent. For more details on CRIME, see Alonso, Ferreira & Santos (2014a).

## A3 HI cosmological signal

To generate the HI cosmological signal, we use the MULTIDARK-PLANCK (MDPL2) (Klypin et al. 2016) dark matter N-body simulation, a cosmological simulation which follows  $3840^3$  dark matter particles evolving in a box with side lengths of  $1 \text{ Gpc } h^{-1}$ . The cosmology assumed in the simulation is consistent with PLANCK15, with  $\{\Omega_{\text{M}}, \Omega_{\text{b}}, \Omega_{\Lambda}, \sigma_8, n_s, h\} = \{0.307, 0.048, 0.693, 0.823, 0.96, 0.678\}$  (Planck Collaboration et al. 2016). This simulation has been made into the MULTIDARK-GALAXIES (Knebe et al. 2018) galaxy catalogue, and the MULTIDARK-SAGE catalogue was also produced by applying the semi-analytical model SAGE (Croton et al. 2016). These are publicly available in the Skies & Universe<sup>9</sup> web page. We choose to work with the MULTIDARK-SAGE catalogue.

This simulation is available in the form of redshift snapshots, where each snapshot shows the state of the cosmological density field and galaxies to a different redshift evolution. We choose to use the  $z = 0.39$  snapshot, and assume a redshift range of  $0.2 < z < 0.58$  and an effective redshift of  $z_{\text{eff}} = 0.39$ . This is not entirely realistic, as it does not include the evolution of the density field and galaxies within this redshift range, but it is sufficient for our purposes. We grid our snapshot of galaxies into voxels using Nearest Grid Point (NGP) assignment.

As first described in Cunnington et al. (2020a), the method we use for generating HI intensity mapping simulations is as follows: Each galaxy in the MULTIDARK-SAGE catalogue has an associated cold gas mass, which we use to compute a HI mass. For each voxel, we take the HI mass of every galaxy falling into it, and bin them together. We then convert this binned HI mass into a HI brightness temperature  $T_{\text{HI}}(\mathbf{x})$  for that voxel. Doing this for each voxel yields the desired HI intensity map for our simulation.

One of the limitations of this method is that the MULTIDARK-SAGE catalogue does not include haloes with mass lower than  $\lesssim 10^{10} h^{-1} M_{\odot}$ . In reality, these haloes would be present and contain HI, contributing to the total HI brightness of each voxel. To account for these missing low mass haloes, we rescale the mean HI temperature of our simulation to a more realistic value. We calculate this realistic value based on the HI abundance measurement made by the GBT-WiggleZ cross-correlation analysis,  $\Omega_{\text{HI}} b_{\text{HI}} r = [4.3 \pm 1.1] \times 10^{-4}$  (Masui et al. 2013). We assume a cross-correlation coefficient of  $r = 1$ , and take the HI bias to be  $b_{\text{HI}}(z_{\text{eff}}) = 1.105$ , based on the HI bias fit from Villaescusa-Navarro et al. (2018).

For each redshift slide, this HI IM simulation can be then transformed into an overtemperature field by subtracting the mean temperature. This overtemperature field traces the underlying dark matter overdensity field,  $\delta_{\text{M}}(z)$ :

$$\delta T_{\text{HI}}(z) = T_{\text{HI}}(z) - \langle T_{\text{HI}} \rangle = \langle T_{\text{HI}} \rangle b_{\text{HI}}(z) \delta_{\text{M}}(z). \quad (\text{A2})$$

<sup>7</sup><http://pla.esac.esa.int/pla>

<sup>8</sup><http://intensitymapping.physics.ox.ac.uk/CRIME.html>  
<sup>9</sup>[skiesanduniverses.org](http://skiesanduniverses.org)

**A4 Noise**

We wish to simulate what the instrumental noise for a MeerKLASS-like survey would look like in our case. The noise is uncorrelated in frequency, and it is Gaussian distributed with a standard deviation of (Alonso et al. 2014a):

$$\sigma(\nu) = T_{\text{sys}}(\nu) \left( \delta_\nu t_{\text{tot}} \frac{\Omega_p}{\Omega_a N_{\text{dish}}} \right)^{-1/2}, \quad (\text{A3})$$

where  $\delta_\nu$  is the frequency resolution of the data (1 MHz in our case),  $t_{\text{tot}}$  is the total observing time (which we assume to be 1000 h),  $N_{\text{dish}}$  is the number of dishes (64, consistent with MeerKAT),  $\Omega_p = 1.13\theta_{\text{FWHM}}^2$  is the pixel solid angle, and  $\Omega_a$  is the survey solid angle:

$$\Omega_a = 4\pi \frac{A_{\text{sky}}}{41253}, \quad (\text{A4})$$

where  $A_{\text{sky}}$  is the observed sky area (2927 deg<sup>2</sup>). The frequency dependent system temperature  $T_{\text{sys}}(\nu)$  is made up of the receiver noise temperature ( $T_{\text{rec}}$ ), and the sky temperature ( $T_{\text{sky}}(\nu)$ ) (Santos et al. 2015):

$$T_{\text{sys}}(\nu) = T_{\text{sky}}(\nu) + T_{\text{rec}}, \quad (\text{A5})$$

where  $T_{\text{rec}} = 25$  K and  $T_{\text{sky}}(\nu)$  is given by

$$T_{\text{sky}}(\nu) = 1.1 \times 60 \left( \frac{300}{\nu[\text{MHz}]} \right)^{2.55}. \quad (\text{A6})$$

This paper has been typeset from a  $\text{\TeX}/\text{\LaTeX}$  file prepared by the author.