DroPS

Deriving r from Power Spectra

Zhiqi Huang

huangzhq25@mail.sysu.edu.cn

November 4, 2025

Contents

1 Introduction

1.1 Primordial power spectra

According to the standard cosmological model, the cosmic microwave background (CMB) and the large-scale structure of the universe originated from tiny, nearly Gaussian metric fluctuations in the primordial universe. At linear order, these primordial metric fluctuations can be decomposed into scalar, vector, and tensor modes. Vector perturbations decay rapidly and are therefore generally assumed to be negligible. The primordial scalar and tensor perturbations are respectively parameterized by the dimensionless scalar power spectrum $\mathcal{P}_S(k)$ and tensor power spectrum $\mathcal{P}_T(k)$, where k denotes the comoving wavenumber.

Guided by slow-roll inflation models, the primordial power spectra are commonly parameterized as

$$\mathcal{P}_S(k) = A_s \left(\frac{k}{k\text{pivot}} \right)^{n_s - 1}, \tag{1}$$

and

$$\mathcal{P}_T(k) = r A_s \left(\frac{k}{k\text{pivot}} \right)^{n_t}. \tag{2}$$

The scalar amplitude A_s and spectral tilt n_s have been tightly constrained by CMB experiments [1]. In most viable inflationary scenarios, the tensor tilt n_t is very small; it is typically either fixed to zero or set to the slow-roll prediction $n_t = -r/8$. The tensor-to-scalar ratio r at the pivot scale, however, remains poorly measured. As of this writing, the best 95% confidence-level (CL) upper limit is $r < 0.032$ [2]. Nevertheless, a broad range of inflationary models remains consistent with $0 < r < 0.032$.

1.2 CMB B-polarization and r

The search for the signature of primordial gravitational waves in CMB, i.e., measuring r from CMB is one of the most compelling pursuits in modern cosmology. This quest focuses on a unique observable: the B-mode polarization pattern.

The CMB photons became polarized when they scattered off free electrons in the early universe. This process imprinted a directional preference on the light, creating two distinct patterns: E-modes, which have a curl-free pattern like the electric field around charges, and B-modes, which have a curl-like pattern. While density fluctuations (scalar perturbations) in the primordial plasma can generate E-modes and a small amount of B-modes through gravitational lensing (so-called lensing B-modes), they cannot produce the specific, large-scale curl-like pattern of primordial B-modes.

This is where primordial gravitational waves come in. These waves, theorized to be generated during the inflationary epoch, are literally ripples in the fabric of spacetime.

As they propagated through the early universe, they periodically stretched and squeezed space, imparting a unique, curl-like distortion to the plasma. This gravitational tugging created a polarization pattern in the CMB that is fundamentally rotational in nature—the primordial B-mode polarization.

Therefore, the B-mode power spectrum at large angular scales acts as a direct tracer for these primordial gravitational waves. A confident detection of this primordial B-mode signal would be tantamount to detecting the gravitational waves themselves. Its amplitude is directly proportional to the energy scale of inflation, with the tensor-to-scalar ratio r quantifying the strength of the signal. Measuring this B-mode power spectrum thus provides a unique window into the physics of the universe's first moments and the grand unification of gravity and quantum mechanics.

1.3 Ground-based CMB experiments with small-aperture telescopes

Many ground-based CMB experiments with small-aperture telescopes (SAT), such as BICEP/Keck [3], AliCPT [4, 5, 6], Simons Observatory SAT (SO-SAT) [7, 8] and CMB Stage four SAT (CMB-S4-SAT) [9, 10][1] aim to measure the CMB B-mode polarization at degree scales and to constrain the primordial gravitaional waves. These telescopes typically measure the sky emission in the range between 30GHz and 300GHz. The raw signals measured by these telescopes are mixture of Galactic foreground, CMB, shot noise, instrumental noise, and contamination from the ground and atmoshpere. Extracting CMB B-mode polarization signal from the mixure of signals is a non-trivial problem and need to be dealt with by specialized softwares. DroPS is one of the software does this job, primarily designed for the ground-based small-aperture telescopes.

2 Software Documentation

DroPS is a very light software. All the examples shown in this documentation are run with my 7-years old laptop. To play with DroPS, what you need is just a personal computer with a few hundred GB free space.

2.1 Installing DroPS

The instruction here has been tested on Ubuntu-24.04.3LTS, and should be easily extendable to other linux platforms. A bit twists may need to be done if you are working with Windows or Mac-OS.

[1] At the time of writing, CMB-S4 is no longer financially supported.

2.1.1 Installing tools and libraries

Install the following packages and libraries with Synaptic Package Manager (or "sudo apt install"):

- git
- gcc
- gfortran
- cmake
- python3-pip
- python-is-python3
- python3-venv
- openmpi-dev
- libxcb-cursor0
- libcfitsio-dev
- libgsl-dev
- libfftw3-dev
- libfftw3-mpi-dev
- libhealpix-dev

2.1.2 Set up a python virtual environment

Create a directory for python virtual environment in your work path (hereafter denoted as YourWorkPath)

```
mkdir YourWorkPath/.work
```

Create the python virtual environment

```
python -m venv YourWorkPath/.work
```

Activate the virtual environment

```
source YourWorkPath/.work/bin/activate
```

On windows you may need to run

```
YourWorkPath/.work/Scripts/activate.bat
```

in cmd.exe or

```
YourWorkPath/.work/Scripts/activate.psl
```

in PowerShell.

When you are done with your work, exit the terminal or use

```
deactivate
```

to exit the virtual environment.

If you are not working with other python projects. You may want to activate the virtual environment automatically with the terminal

```
echo “source YourWorkPath/.work/bin/activate”  ∼/.bashrc
```

2.1.3 Install requirements

Activate the virtual environment either manually or automatically as described in the previous subsection.

Upgrade pip for the latest information of packages:

```
pip install --upgrade pip
```

Now enter your work path where you want to install DroPS

```
cd YourWorkPath
```

Get the DroPS repository

```
git clone https://github.com/zqhuang/DroPS
```

Now enter the DroPS directory

```
cd DroPS
```

Install all dependences

```
pip install -r requirements.txt
```

2.1.4 Hacking pysm3

Hacking a python package is probably against the basic idea of python, but we are doing it anyway to improve the efficiency of CMB simulations. If you only want to analyze maps, however, you can skip this "unpleasant" step.

Enter the DroPS directory

```
cd YourWorkPath/DroPS
```

Move the cmb.py file in the pysm3 package to somewhere else

```
mv PATH_TO_pysm3/models/cmb.py cmb_backup.py
```

and replace it with the cmb.py file that comes with DroPS

```
cp cmb.py PATH_TO_pysm3/models/
```

Here PATH_TO_pysm3 stands for the path where pysm3 was installed. On Ubuntu 24.04.3LTS, you may find PATH_TO_pysm3 to be

YourWorkPath/.work/lib/python3.12/site-packages/pysm3

If you are not using Ubuntu24.04.3LTS, the pysm3 path may be slightly different. You can find out the path by doing

```
sudo apt install plocate
```

and

```
locate pysm3
```

2.2 Base simulations

The base simulations can be understood as an analog to the "learning samples" in the machine-learning language[2]. They reflect the best understanding of the cosmological model, instruments, noise sources and foreground properties, but systematic offsets may still exist between the base simulations and the real observations.

2.2.1 Generate a TOD filtering model

A critical step in processing data from ground-based CMB experiments is the removal of contaminating ground and atmospheric signals from the time-ordered data (TOD). To simulate this filtering, one needs specific information about the experiment's site, which is not currently available. Fortunately, the overall effect of the filtering is understood: it suppresses large-scale (low multipole) power in the resulting maps and introduces non-Gaussianity by mixing different Fourier modes.

If you are not keen about simulating precise filtering effect for a specific experiment, you may use the "mock filtering" tool that comes with DroPS to generate a filtering matrix:

```
python mock_filtering.py
```

Follow the prompt and enter the healpix resolution (nside, 128 for testing, 256/512 for serious simulations) and the file name for the filtering matrix (e.g. filter_128.pickle). You may also refine the model by entering the parameters $\ell_{\rm cut}$, $f_{\rm low}$, $f_{\rm high}$, $\alpha_{\rm low}$, θ_ℓ, θ_m that are defined in Section 3.1.

2.2.2 Generating base simulations

In this section, we run "base simulations" to obtain the statistics of the sky. DroPS comes with four "demo-experiments": Test, AliCPT, SO, and CMBS4. The "Test" demo-experiment is just for testing purpose and does not have a counterpart in reality. The configurations (sky coverage, noise level etc., see Table 1) of the latter three demo-experiments (AliCPT, SO, CMBS4) are similar, but not identical to their counterparts in reality (AliCPT [4], SO-SAT [7] and CMB-S4-SAT [10]). In this documentation we will mainly use the "AliCPT" experiment as a benchmark. We will also use SO for comparison between DroPS and other pipelines that have been applied on SO-SAT simulations [7].

To begin with, you can simulate noise/cmb/foreground maps with AliCPT.

[2]This is just an anlog. DroPS does not use any machine-learning techniques.

Table 1: Key configurations of DroPS demo-experiments

	frequency (GHz)	beam FWHM (arcmin)	temperature noise (μK-arcmin)	$\ell_{\rm knee}$	$\alpha_{\rm knee}$
Test	30	60	1	60	-1.7
($f_{\rm sky} \approx 0.1$)	95	20	1	60	-1.7
	150	20	1	60	-3
	270	10	1	60	-3
AliCPT	27	97	47	30	-2.4
($f_{\rm sky} \approx 0.14$)	40	65	34	30	-2.4
	90	16.2	2.2	50	-2.5
	150	9.7	3.3	50	-3
(+Planck)	217	4.9	50	0^+	-3
SO	27	91	33	15	-2.4
($f_{\rm sky} \approx 0.1$)	39	63	22	15	-2.4
	93	30	2.5	25	-2.5
	145	17	2.8	25	-3
	225	11	5.5	35	-3
	280	9	14	40	-3
CMBS4	30	72.8	2.5	60	-1.7
($f_{\rm sky} \approx 0.1$)	40	72.8	3.15	60	-1.7
	85	25.5	0.622	60	-1.7
	95	22.7	0.552	60	-1.7
	145	25.5	0.87	60	-3
	155	22.7	0.948	60	-3
	220	13	2.46	60	-3
	270	13	4.22	60	-3

```
python simulate.py AliCPT/AliCPT_sim_config.txt
```

Compare the content of configuration file AliCPT/AliCPT_sim_config.txt with Table 1 to understand how the configurations are passed to the script. When no other command-line arguments are passed to simulate.py, it only produces base simulations, that are, a lot of realizations of noise and CMB maps based on the noise model and cosmology that are specified in the configuration file. The foreground maps are only produced once, as we do not yet have a reliable model to discribe the "cosmic variance" of foreground emissions. By default, the base simulations use the foreground model ['d0', 's0'] with fixed spectral indices ($\beta_d = 1.54$ for thermal dust, $\beta_s = -3$ for synchrotron). This ['d0', 's0'] foreground map only captures the gross feature of the Galactic emission. The "actual foreground" that we will analyze in the next section can be different from the one used in the base simulations.

To further understand how the foregound models are defined in pysm3, you may follow its documentation at https://pysm3.readthedocs.io/.

2.3 Analyzing the sky maps

In the last section, we run "base simulations" (learning samples) based on the best-known noise model, assumed foreground model (['d0', 's0']) and some assumed r values in a fiducial ΛCDM cosmology. In this section, we simulate the "observed sky" with the same noise model, optionally a different foreground model, and a r value that has nothing to do with the base simulations. DroPS will reconstruct r by comparing the "observed sky" with the base simulations. The simulations of the "observed sky" here can be understood as an analog to the "test samples" in the machine-learning lanuage.

2.3.1 Analyzing one realization of sky

Generate the "observed sky" with, e.g.,

```
python simulate.py AliCPT/AliCPT_sim_config.txt maps/AliCPT_ 0.01 999
```

You can replace maps/AliCPT_ with your preferred prefix for the output maps, 0.01 with your preferred fiducial r value, and 999 with your preferred random seed. To test whether DroPS can deal with a spatial variation of the foreground, you may also replace the ['d0', 's0'] foreground model with, e.g., ['d1', 's1'] in the configuration file AliCPT/AliCPT_sim_config.txt.

Now analyze the "observed sky" with

```
python mainpipe.py AliCPT/AliCPT_ana_config.txt maps/AliCPT_
```

Read the configuration file AliCPT/AliCPT_ana_config.txt to understand how to analyze the maps with different settings.

The summary statistics of r and other parameters will be shown on the screen when the analysis is done. More detailed results will be saved in AliCPT/results (this path is specifed in the configuration file AliCPT/AliCPT_ana_config.txt as well).

2.3.2 Analyzing multiple realizations of sky

In one simulation, the posterior mean value of r can be above or below the input r, and this depends on the random seed. To test whether the r-measurement pipeline is biased or not, we need to apply the pipeline on many simulations with different random seeds.

This time we choose a different foreground model ["d1", "s1"]. Now run the simulations and analyze them by running the following shell script (test_bias.sh in the repository)

```
./test_bias.sh
```

The content of test_bias.sh is shown below.

Listing 1: content of test_bias.sh

```
for i in `seq 100`
do
    python simulate.py AliCPT/AliCPT_sim_config.txt maps/s${i}_  0.01 ${i}  d1s1
    python mainpipe.py AliCPT/AliCPT_ana_config.txt maps/s${i}_ logfile.txt
done
```

(Here 0.01 is the fiducial r value; the random seed (bash variable ${i}) runs from 1 to 100; maps/s${i} is the root name (prefix for output files) for each simulation; logfile.txt is the file that saves the reconstructed means and standard deviations of r.

Now plot the saved means and standard deviations of r from logfile.txt, and compare them with the input $r = 0.01$.

```
python utils/plot_rs.py logfile.txt 0.01
```

If the bias of r-measurement is small, the mean of reconstructed mean values of r (dotted blue line) is supposed to be close to the fiducial value (solid orange line), as shown

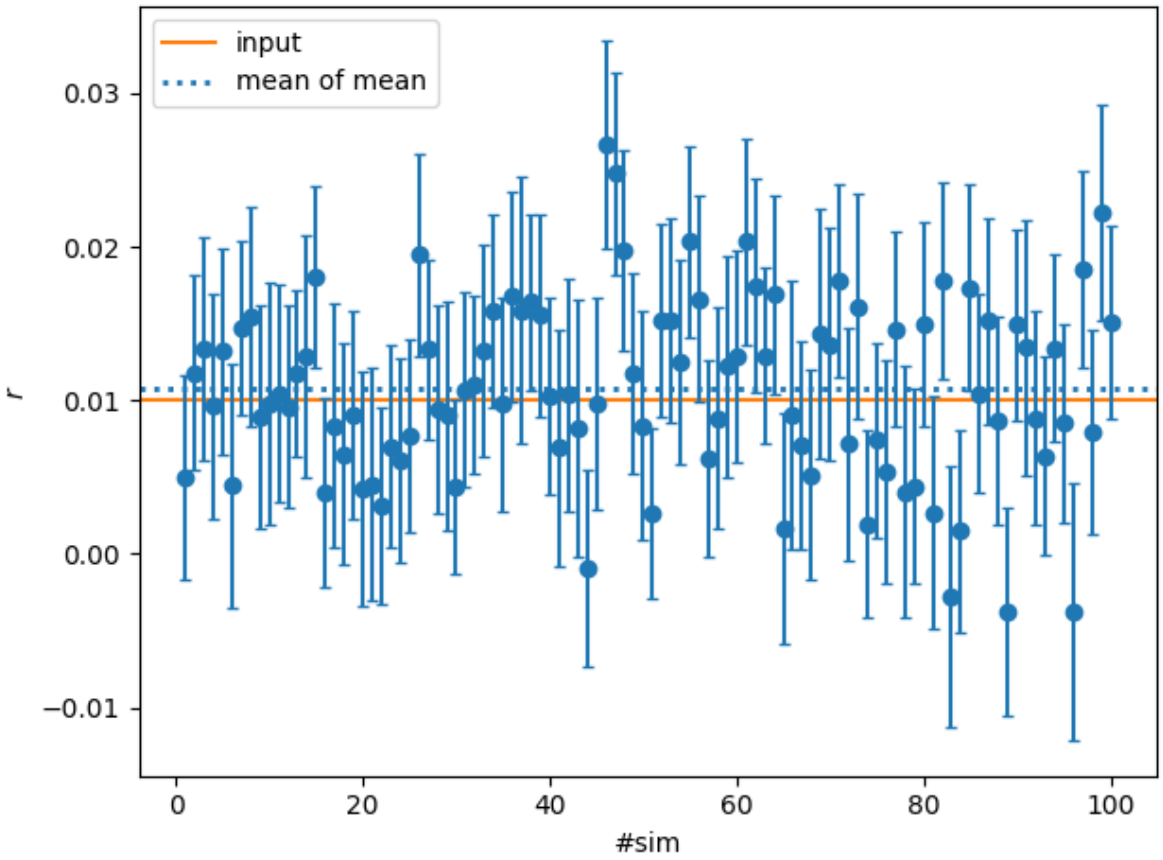

Figure 1: Reconstructed r for 100 realizations of sky with different random seeds. For each sky, the r value is reconstructed by comparing the sky with 300 base simulations. The 100 realizations of sky are simulated with foreground model ['d1','s1'] (spatially varying SED of synchrotron and dust emission), while ['d0', 's0'] (fixed SED) is used in the 300 base simulations.

in Figure 1.

2.3.3 How to apply on real data

A real experiment comes with its own pipeline of noise simulations and TOD filtering, and provides the actually observed sky maps. To analyze the real data, you can simply replace the base simulations with the maps from the simulation pipeline of the experiment, and replace the realization of sky with the actually observed ones.

2.3.4 Comparison with other pipelines

DroPS has been applied on AliCPT data challenge (simulated 14% sky, 95GHz and 150GHz) and achieved consistent results with other methods [5]. Here we further compare DroPS with three pipelines that have been applied on simulation data of SO-SAT (Wolz

et al. 2024 [7]). We adopt the optimistic configuration of SO-SAT. The input fiducial r is zero, and in the analyses unphysical negative r is allowed[3]. The results are shown in Figure 2. From top to bottom, foreground models ["d0", "s0"], ["d1", "s1"], and ["dm", "sm"] are used, respectively. From the left where the reconstructed mean values of r and the bias (mean of the mean values of r) are shown, we find DroPS has negligible biases for all cases and more stable than the three pipelines tested in Wolz et al. The right column shows σ_r, the statistical uncertainties in r. We again find good consistency between DroPS and the other pipelines.

2.4 Map-level Operations

2.4.1 Likelihood-based component separation

The most well known method of component separation is probably the internal linear combination (ILC) algorithm and its variations. The basic idea is to isolate a signal - whose frequency dependence is known - by taking linear combination of the frequency maps. To do that in pixel space, the frequency maps have to be smoothed to a common resolution.

For ground-based CMB experiments, however, a key challenge of ILC or ILC-like methods is that TOD filtering and beam convolution are non commutative operations. This prevents the frequency maps from being smoothed to a common resolution. Thus, while ILC remains popular in studies where the complexity of TOD filtering effect is ignored [7], likelihood-based method - which requires much more computing resources - has been used in real data analyses of BICEP/Keck [11].

To demonstrate how the component separation code works, we generate a sky realization from the base simulation #0:

```
python utils/combine_sim.py AliCPT/AliCPT_sim_config.txt maps/r3AliCPT0_ 0
```

The above script takes the base simulation #0 and adds the filtered foreground maps, filtered noise maps and filtered cmb maps into sky maps with root name maps/r3AliCPT0_.

Now we do component separation

```
python compsep.py AliCPT/AliCPT_sim_config.txt maps/r3AliCPT0_
```

[3]in DroPS this is realized by setting r_lowerbound negative in the configuration file

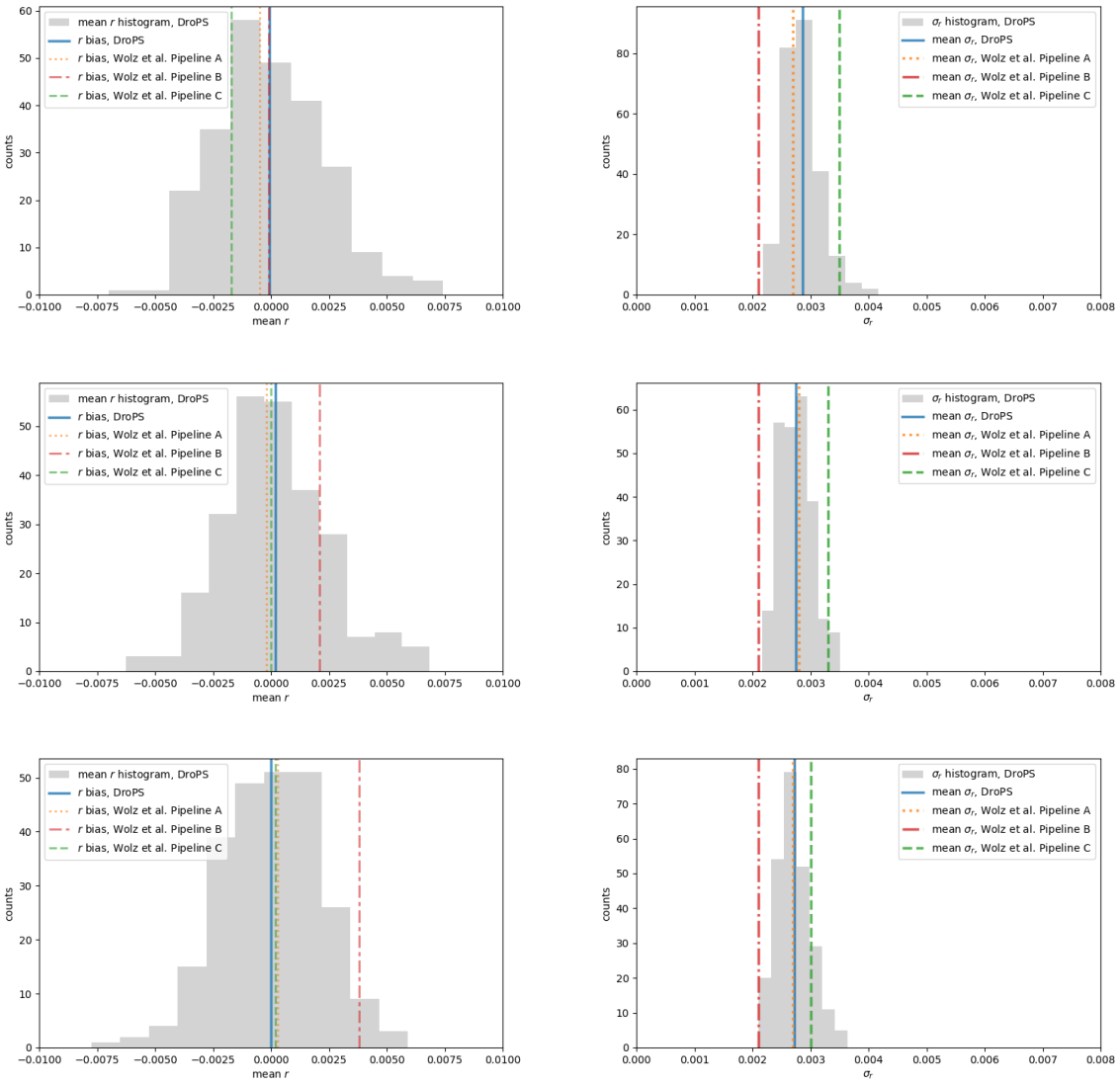

Figure 2: Comparing DroPS with three pipelines that are described in Wolf et al. [7]. From top to bottom, foreground models "d0, s0", "d1, s1", and "dm, sm" are used, respectively. Left and right columns are the distrubution of reconstructed mean r and σ_r, respectively.

The code uses Stochastic Gradient Langevin Dynamics (SGLD) method to search the maximum of the likelihood described in Section 3.5. It typically takes a few days to run compsep.py for the first time, on a personal laptop. When the gradient templates are built, the code runs much faster, typically takes only a few hours on a personal laptop. Typically you need to run it multiple times to obtain a well converged CMB B-mode map. Figure 3 shows the result for the 40GHz channel maps simulated with AliCPT/AliCPT_sim_config.txt. The E maps are recovered very well. While significant residual noises remain in the B maps.

For real data analyses, the calculation of gradient template involves map making and therefore can take months to built. Fortunately, the gradient template can be built in a parallel way. The template can be built for $\ell \in [\ell_{\rm min}, \ell_{\rm max}]$ by running

```
python compsep.py AliCPT/AliCPT_sim_config.txt maps/r3AliCPT0_ lmin lmax
```

Thus, the task can be decomposed into many subtasks with different lmin and lmax. You need ∼100GB disk space to store all the gradient templates.

2.4.2 Displaying the maps

DroPS provides a few tools to display the maps: utils/viewfits.py displays the fits map.

```
python utils/viewfits.py NHmask_G_128.fits
```

If you have a map example_IQU.fits with I, Q, U components, display the T map with

```
python utils/viewfits.py example_IQU.fits 0
```

and display the Q map with

```
python utils/viewfits.py example_IQU.fits 1
```

display the U map with

```
python utils/viewfits.py example_IQU.fits 2
```

To display maps in npy format, use utils/viewnpy.py.

To display E/B components, use utils/viewEB.py.

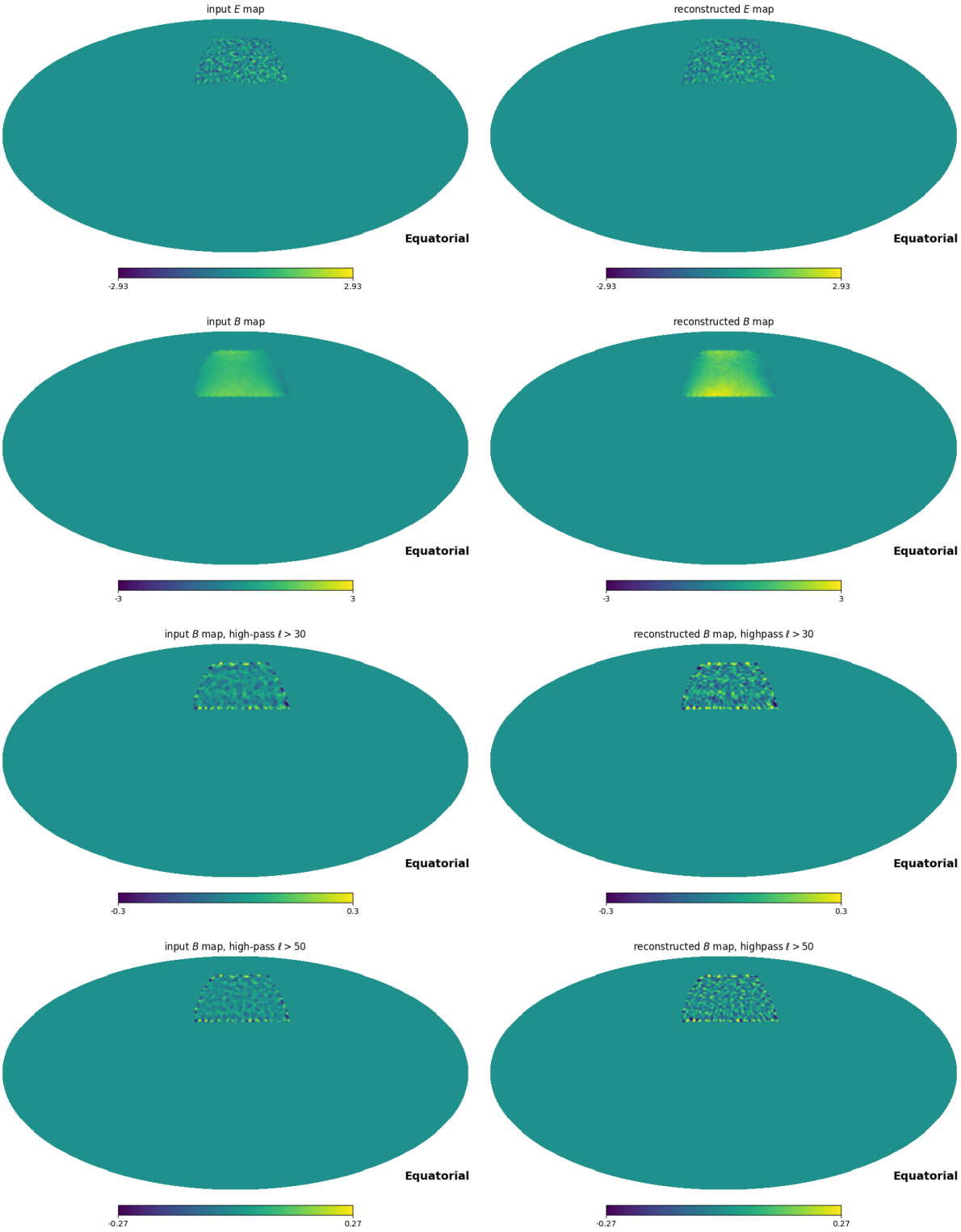

Figure 3: Component separation; Displayed are the 40GHz channel maps, simulated with AliCPT/AliCPT_sim_config.txt

3 Technical Details

In this section I assume the reader is familiar with Healpix and has the basic knowledge of cosmology and spherical harmonic transform.

3.1 TOD filtering and map making

In ground-based CMB experiments, TOD filtering is a crucial preprocessing step to isolate the faint CMB signal from overwhelming, structured ground-based noise. Since these telescopes observe from the ground, their data streams are heavily contaminated by low-frequency noise, primarily from atmospheric emission and instrumental $1/f$ noise. TOD filtering works in the time domain to identify and subtract these systematic contaminants, which often have characteristic temporal or scan-synchronous patterns distinct from the cosmological CMB fluctuations. By applying high-pass or more sophisticated modal filters, this process effectively suppresses large-scale noise while striving to preserve the intrinsic CMB signal, thereby enabling the recovery of high-fidelity sky maps for cosmological analysis. Despite being a linear transformation, TOD filtering is typically a dense matrix in both pixel space and harmonic space, which is difficult to model without more detailed knowledge (e.g. scanning strategy) of the experiments. Consequently, DroPS does not provide map-making pipelines for either of the demo-experiments. Instead, DroPS provides "map-making simulators" by joining a few linear operations in both pixel space and in harmonic space. These operations relies on the survey mask and a few input parameters that characterize the main features of TOD filtering. Specifically, these linear operations are listed below.

1. Generate a TOD filtering matrix (in harmonic space) with six parameters $\ell_{\rm cut}$, $f_{\rm low}$, $f_{\rm high}$, α, θ_ℓ and θ_m. Specifically, the matrix is given by

$$M_{\ell' m', \ell m} = \begin{cases} \sqrt{\left[1 - 2\left(\theta_\ell + \theta_m\right)\right] F_\ell}, & \text{if } \ell' = \ell, m' = m; \\ \sqrt{F_\ell \theta_\ell} G_{\rm random} & \text{if } |\ell' - \ell| = 1, m' = m \\ \sqrt{F_\ell \theta_m} G_{\rm random} & \text{if } \ell' = \ell, |m' - m| = 1 \\ 0, & \text{else} \end{cases}$$

   where

$$F_\ell \equiv f_{\rm high} \left\{ 1 - f_{\rm low} \exp\left[- \left(\frac{\ell}{\ell_{\rm cut}} \right)^\alpha \right] \right\}$$

   and $G_{\rm random}$ is a random Gaussian number. For each matrix element $G_{\rm random}$ is drawn independently.

2. Generate a smoothed mask by apodizing the mask edges using a C2-type kernel with an apodization scale of 2 degrees [12].

3. Pixel wisely multiply the map with the smoothed mask.

4. Do shperical harmonic transform.

5. In harmonic space, multiply the TOD filtering matrix with the map vector,

$$a_{\ell' m'} = \sum_{\ell,m} M_{\ell' m',\ell m} a_{\ell m}.$$

6. Do inverse spherical harmonic transform.

7. Pixel wisely multiply the map with the original mask.

Since each step of the above operations is linear, the combined map-makinig simulator is also a linear operator, too. The simulator takes much less computing resources than any real map making pipelines, as no actual TOD are generated. Nevertheless, the simulator captures the main features of TOD filtering (large-scale suppression and mode mixing).

When

```
python mock_filtering
```

is called. It generates a pickle file that saves the random numbers ($G_{\rm random}$) in operation 1.

When simulating maps, the pickle file is specified in the configuration file (e.g. AliCPT/AliCPT_sim_config.txt). DroPS loads the pickle file and performs "map making" (seven linear operations in sequence, as specified above).

In the data analyses, DroPS is blind to what has been used in the simulation. It models the filtering effect with a coupling matrix $F_{\ell\ell'}$, whose main diagonal elements describe the suppression effect, the first off-diagonal elements describe the coupling between adjacent bins, and all other elements are assumed to be zero. By default, DroPS estimate $F_{\ell\ell'}$ by comparing the band powers of filtered and unfiltered noise maps of the base simulations[4].

Note that, however, strictly speaking, TOD filtering is a linear operation on the map, but not on the power spectra. The linear modeling may not be very accurate. For this reason, DroPS tends to minimize the usage of the $F_{\ell\ell'}$ matrix, as described in Section 3.3.

3.2 Delensing

Gravitational lensing by large-scale structure subtly distorts the pristine CMB, smoothing its acoustic peaks and generating a spurious B-mode polarization signal that obscures

[4] A least square fit is done here.

the far weaker primordial signature. The expectation of CMB lensing B-mode power can be computed from theory and subtracted if the other cosmological parameters are known. However, for actual realization of the universe, the lensing BB power randomly deviate from its expectation value. Such random deviation, often called cosmic variance, which cannot be predicted from the theory may contaminate the measurement of r if we are targeting at $\sigma_r \lesssim 10^{-3}$. Delensing is the process of statistically reconstructing this actual lensing distortion (rather than the theoretical expectation) from high-resolution CMB data itself or from external tracers of the matter distribution, and then removing its effect from the observed CMB map. By cleaning this cosmic foreground, delensing sharpens the CMB's original features and significantly reduces the confounding lensing B-mode power, thereby dramatically improving the sensitivity of experiments to detect the faint imprints of inflation.

DroPS assumes that an ideal lensing potential map is obtained from external sources (high-resolution CMB experiments or large scale structure surveys). Although the ideal lensing potential map allows to remove all the lensing effect, we will not be able to do so in practice. By setting the delensing efficiency $\epsilon_{\rm delens}$ (delens_fac in configuration file, e.g., AliCPT/AliCPT_ana_config.txt), you can specify how much lensing effect you want to remove. More concretely, delensing is done by rescaling the lensing potential map by a factor of $(1 - \epsilon_{\rm delens})$.

3.3 Likelihood for band powers

DroPS evaluates band powers (average power spectra in ℓ bins) of masked maps with NaMaster [13]. For measurement of r, we typically use a few ℓ bins at $20 \lesssim \ell \lesssim 200$. Maps with nside = 256 are sufficient for such analyses, and nside = 128 are often good enough for quick estimations. However, if you are changing resolution of maps or rotating them to a different coordinate system, caution should be taken. These operations often leads to extra E-B leakage in low-resolution maps. We recommend nside = 512 for analyses that involve these operations.

The data vector is the band powers of filtered sky maps. The cross correlation between different frequency maps are computed directly, while the auto correlation of a frequency map is computed with pairs of season maps. In all cases, the expectation value of noise band powers are zero. DroPS avoids using auto-correlation of a same map, as the auto correlation of noise can be biased. This is a standard treatment in modern CMB data analyses.

We will assume that the number of ℓ-bins is $n_{\rm bin}$, the number of frequency channels is n_f, and the number of fields is n_{XY}. The data vector is a collection of band powers $D_\ell^{XY}(\nu_1, \nu_2)$, where XY ranges over the fields $(XY = TT, TE, TB, EE, EB, BB)$,

ℓ ranges over the central value of the $n_{\rm bin}$ ℓ-bins, and (ν_1, ν_2) ranges over all possible ranked ($\nu_1 \leq \nu_2$) pairs of the central frequencies of the n_f frequency channels. Note that when $\nu_1 = \nu_2$, the band powers $D_\ell^{XY}(\nu_1, \nu_2)$ are computed by averaging all band powers between season maps, as we mentioned earlier. The number of ranked frequency pairs is $n_f(n_f+1)/2$. Therefore, the length of the data vector is $N_d = \frac{1}{2} n_{XY} n_{\rm bin} n_f(n_f+1)$. For the purpose of measuring r and if computing/storage resources are limited, it is recommended to just use the BB band powers ($n_{XY} = 1$), which contain most of the information about r.

The basic idea is then to compare the data vector with the theory, which include cosmological, foreground, and noise models. The likelihood can be abstractly written as

$$\mathcal{L} = \frac{1}{(2\pi)^{N_d/2}\sqrt{\det \mathrm{Cov}}} \exp\left[-\frac{1}{2}(D^{\rm obs} - D^{\rm model})^T \mathrm{Cov}^{-1}(D^{\rm obs} - D^{\rm model})\right], \tag{3}$$

where $D^{\rm obs}$ is the data vector (band powers of observed maps). The model band powers $D^{\rm model}$ can be written as

$$D^{\rm model} = D^{\rm CMB} + D^{\rm fg}, \tag{4}$$

where $D^{\rm CMB}$ and $D^{\rm fg}$ are expectation values of band powers of CMB maps and foreground maps, respectively. (The expectation value of noise band powers vanishes as DroPS only takes cross correlations between different frequency bands or between different season maps in a same frequency band.) The covariance Cov contains the contribution from noise, foreground and CMB.

3.3.1 CMB band powers

The base simulations provide band powers for (filtered and delensed) CMB maps with fiducial $r = 0$ and $r = r_1 > 0$, respectively. The recommended value of r_1 is 0.03 (roughly current 90%CL upperbound), but you can change it in the configuration file, e.g., AliCPT/AliCPT_sim_config.txt. Averaging over the base simulations, we obtain the expectation values of CMB band powers $\langle D_\ell^{\rm CMB}(r=0)\rangle$ and $\langle D_\ell^{\rm CMB}(r=r_1)\rangle$.

In the standard cosmological scenario, the band powers has a linear response to r, if the other cosmological parameters (collectively denoted as θ) remain fixed to their fiducial values ($\theta_{\rm fid}$). In DroPS, the CMB band powers are modeled as

$$\begin{aligned} D_\ell^{\rm CMB} &= (1-w)\langle D_\ell^{\rm CMB}(r=0)\rangle + w\langle D_\ell^{\rm CMB}(r=r_1)\rangle & (5)\\ &+ \sum_{\ell'} F_{\ell\ell'}\left[D_{\ell'}^{\rm CMB}(r,\theta) - (1-w)D_{\ell'}^{\rm CMB}(r=0,\theta_{\rm fid}) - wD_{\ell'}^{\rm CMB}(r=r_1,\theta_{\rm fid})\right], & (6)\end{aligned}$$

where

$$w \equiv \max\left[\min\left(\frac{r}{r_1}, 1\right), -1\right]. \tag{7}$$

In other words, DroPS takes the linear interpolation from simulations, which contains all observational effects (e.g. E-to-B leakage), and corrects it if $|r| > r_1$ or other cosmological parameters vary. For next generation CMB experiments, the most likely scenario is that $|r| \lesssim r_1$ (as BICEP/Keck data already suggest) and the other cosmological parameters are well constrained. Therefore, the TOD filtering matrix is only applied on a small quantity. This approach minimizes the impact of possibly inaccurate modeling of filtering effect.

3.3.2 Covariance matrix

The covariance matrix of band powers (of filtered sky maps) can be, in principle, blindly estimated from simulations. In practice, however, the covariance typically contains $\gtrsim 10^3$ free elements, which do not converge well with a few hundred simulations that we typically have. DroPS solves this problem by doing "noise cleaning" and ignoring correlations between ℓ bins that are mutually far away.

DroPS decomposes the covariance matrix into three components: the noise covariance, the signal covariance, and the signal×noise covariance. The advantage of doing so is that noise covariance can be "cleaned" with the piror knowledge that noise in different frequency channels are uncorrelated. DroPS also assumes that the noise T, E, B are mutually uncorrelated[5]. Consequently, the noise covariance between $D_\ell^{XY}(\nu_1, \nu_2)$ and $D_{\ell'}^{X'Y'}(\nu_1', \nu_2')$ is nonzero only if $XY = X'Y'$ and $(\nu_1, \nu_2) = (\nu_1', \nu_2')$. This substantially reduces the degrees of freedom and improves the accuracy of noise covariance estimation.

By default DroPS ignores correlation between different ℓ bins (ell_cross_range = 0 in the configuration file). Optionally DroPS can can include *some* correlations between adjacent or next-to-adjacent ℓ bins (ell_cross_range = 1, 2 in the configurationfile). We emphasize "some" here, because keeping only diagonal and the first off-diagonal blocks (or also the second off-diagonals if required) of a positive definite matrix may introduce unphysical ghost modes (negative eigen values introduced by this numeric approximation) that lead to catastrophic overfitting. Therefore, DroPS trades part of the information in the correlations between adjacent ℓ bins (and next-to-adjacent ℓ bins if they are included) with numeric stabilit. Technically, this is realized by multiplying the non-diagonal blocks with some suppression factors. Table 2 lists the optimal suppression factors, which are either analytically derived (if the number of ℓ bins is small) or numerically evaluated (if the number of ℓ bins is big). When only correlations of adjacent bins are included (ell_cross_range=1), the first off-diagonal blocks are multiplied by f_1 in the second column of Table 2. While correlations of the next-to-adjacent bins are included (ell_cross_range = 2), the first off-diagonal blocks and the second off-diagonal blocks are multiplied by f_1 and f_2 in the third column of Table 2, respectively.

[5]In the default settings where BB band powers are used, this assumption is unecessary.

Table 2: suppression factors (f_1 for adjacent and f_2 for next-to-adjacent bins)

	ell_cross_range = 1	ell_cross_range = 2
# ℓ bins	f_1	f_1, f_2
2	1	-
3	$\frac{1}{\sqrt{2}}$	1, 1
4	$\frac{\sqrt{5}-1}{2}$	$\frac{\sqrt{3}}{2}$, $\frac{1}{2}$
5	$\frac{1}{\sqrt{3}}$	$\sqrt{2-\sqrt{\frac{28}{3}}\cos\frac{\pi+\arccos\sqrt{\frac{27}{28}}}{3}}$, $\sqrt{\frac{28}{3}}\cos\frac{\pi+\arccos\sqrt{\frac{27}{28}}}{3}-1$
6	$\frac{2}{3}-\frac{\sqrt{28}}{3}\cos\frac{\pi+\arccos\frac{1}{\sqrt{28}}}{3}$	$\sqrt{\frac{3}{2}-\frac{4\sqrt{7}}{9}\cos\frac{\pi-\arccos\frac{17}{7\sqrt{7}}}{3}+\frac{7}{9}\cos\frac{2\pi-2\arccos\frac{17}{7\sqrt{7}}}{3}}$, $\frac{5}{6}-\frac{\sqrt{7}}{3}\cos\frac{\pi-\arccos\frac{17}{7\sqrt{7}}}{3}$
7	$\frac{\sqrt{2-\sqrt{2}}}{2}$	0.74222720 , 0.25777280
8	0.53208889	0.74535599 , $\frac{1}{3}$
9	0.52573111	0.73967857 , 0.29430138
10	0.52110856	0.73205081 , 0.26794919
11	0.51763809	0.72458926 , 0.24925315
12	0.51496392	0.72430917 , 0.29789426
13	0.51285843	0.72360680 , 0.27639320
14	0.51117030	0.72095982 , 0.25989153
15	0.50979558	0.71754959 , 0.24694521
16	0.50866092	0.71688142 , 0.28311858
17	0.50771331	0.71707895 , 0.26835006
18	0.50691364	0.71592096 , 0.25627141
19	0.50623256	0.71404546 , 0.24626867
20	0.50564767	0.71340854 , 0.27515189
∞	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$, $\frac{1}{4}$

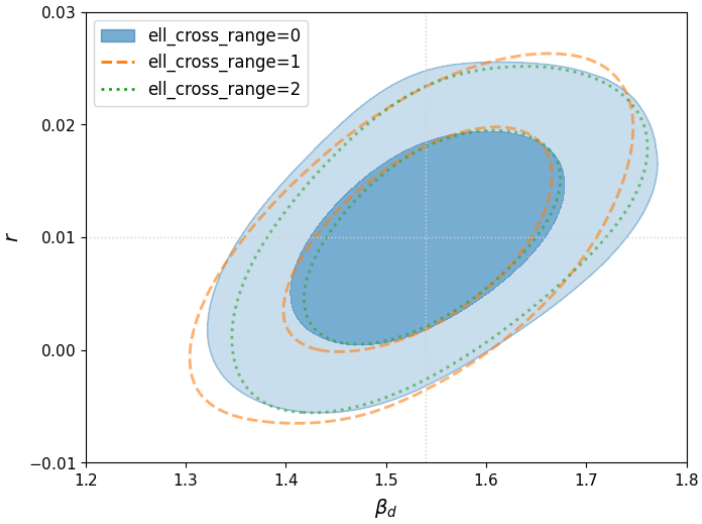

Figure 4: Impact of the correlations between adjacent ℓ bins and next-to-adjacent ℓ bins

Emprical tests show that the correlations between adjacent ℓ bins and next-to-adjacent ℓ bins usually do not have a significant impact on the constraint on r and foreground parameters. Figure 4 shows an exmaple. The simulation is done with AliCPT/AliCPT_sim_config.txt with fiducial $r = 0.01$ and random seed 5; the analyses are done with AliCPT/AliCPT_ana_config.txt (with ell_cross_range = 0, 1, 2 respectively). The input fiducial $r = 0.01$ and $\beta_d = 1.54$ are marked with lightgray dotted lines in the figure. The contours are plotted with the script utils/plot_cont.py that comes with DroPS.

The signal and signal×noise covariance matrices relies on the fiducial cosmology. DroPS evaluates them at $r = 0$ and $r = r_1$. For $0 < r < r_1$, because the signal variation linearly scales with r, the signal covariance is interpolated linearly in r^2, and the signal×noise covariance is interpolated linearly in r. For $r > r_1$, DroPS uses the covariance evaluated at r_1. We do not do extrapolation, which typically introduce unphysical ghost modes that lead to catastrophic overfitting.

3.4 Foreground model

By default DroPS treats the band powers of *filtered* dust and synchrotron maps as free parameters. The advantage of doing so is two folds. First, we do not need to worry about the accuracy of $F_{\ell\ell'}$ modeling, as we are directly modeling the band powers of filtered maps. Second, although the average dust/synchrotron power spectra is likely to be smooth, their power spectra in a small sky area may have random "cosmic variance", which is taken into account in the free-bandpower parametrization. The disadvantage of abandoning the smoothness assumption on foreground band powers is that it degrades the

constraining power on r. However, this extra uncertainty is physical, as it correctly reflects that the foreground band powers in a small patch of sky can have random fluctuations around a smooth model.

If the number of available frequency channel is small (less than five) or the noise is large (target $\sigma_r \gtrsim 0.01$), however, smooth parameterization of foreground power is recommended. The reason is that the "cosmic variance" of foreground becomes a tiny effect compared to noise, filter modeling uncertainty, and uncertainties due to parameter degeneracy. The smooth model of foreground band powers - which is modeled as a qudratic function of ℓ - can be turned on in the configuration file by setting analytic_fg = True.

3.4.1 Thermal dust emission

The frequency dependence of dust temperature fluctuation is modeled as a modified black-body spectrum, which converted to the CMB μK unit is

$$W_d(\nu) = \left(\frac{\nu}{\nu_{\rm ref}}\right)^{\beta_d-1} e^{\frac{h(\nu_{\rm ref}-\nu)}{k_B T_{\rm CMB}}} \left(\frac{e^{\frac{h\nu}{k_B T_{\rm CMB}}}-1}{e^{\frac{h\nu_{\rm ref}}{k_B T_{\rm CMB}}}-1}\right)^2 \left(\frac{e^{\frac{h\nu_{\rm ref}}{k_B T_{\rm MBB}}}-1}{e^{\frac{h\nu}{k_B T_{\rm MBB}}}-1}\right). \tag{8}$$

To convert a dust map (in CMB μK unit) of frequency ν_1 to dust map at frequency ν_2, a scaling factor $\frac{W_d(\nu_2)}{W_d(\nu_1)}$ is applied.

3.4.2 Synchrotron emission

The frequency dependence of synchrotron temperature fluctuation is modeled as a power-law spectrum, which converted to the CMB μK unit is

$$W_s(\nu) = \left(\frac{\nu}{\nu_{\rm ref}}\right)^{\beta_s-2} e^{\frac{h(\nu_{\rm ref}-\nu)}{k_B T_{\rm CMB}}} \left(\frac{e^{\frac{h\nu}{k_B T_{\rm CMB}}}-1}{e^{\frac{h\nu_{\rm ref}}{k_B T_{\rm CMB}}}-1}\right)^2. \tag{9}$$

To convert a synchrotron map (in CMB μK unit) of frequency ν_1 to dust map at frequency ν_2, a scaling factor $\frac{W_s(\nu_2)}{W_s(\nu_1)}$ is applied. Consequently, a blind parametrization of the dust band powers involves $n_{XY} n_{\rm bin} + 1$ independent parameters. (Here +1 corresponds to the dust frequency spectral index β_d.)

3.4.3 Correlation between thermal dust and synchrotron

Synchrotron emission dominates the low-frequency bands, while thermal-dust emission dominates the high-frequency bands.

3.4.4 Spatially-varying SED: moment expansion method

While a constant SED (β_d for dust and β_s for synchrotron) may be a good approximation for a small patch of sky, the spatial variation SEDs may not be neglected

for experiments targeting at larger fraction of sky (e.g., AliCPT, SO-SAT). By default DroPS deals with spatial variation of foreground with moment expansion method [14] (freq_decorr_model="ME" in the configuration file). This feature can be turned off, though, by setting freq_decorr_model=None.

Another option to deal with spatially-varying SED is to use phenomenological Taylor expansion of the SED on frequency and on decorrelation (freq_decorr_model = "Taylor").

3.5 Likelihood for map-level component separation

The likelihood $\mathcal{L} \propto e^{-\chi^2/2}$ is based on a Gaussian noise model

$$\chi^2 = v_{\rm noise}^T N^{-1} v_{\rm noise}, \tag{10}$$

where $v_{\rm noise}$ is the noise frequency maps inferred from a sky model and the observed frequency maps, and N is the covariance of $v_{\rm noise}$. In pixel space, the size of each vector is $n_{\rm pix} n_f$, i.e., product of the number of pixels and the number of frequency channels. For AliCPT or SO-SAT and maps with degree-scale resolution, we typically have $n_{\rm pixel} n_{\rm frequency} \gtrsim 10^4$. The number of elements in the covariance matrix N is $\gtrsim 10^8$, which is difficult to estimate with simulations and also difficult to invert. The BICEP/Keck analysis [11] only considers pixel auto-correlation, that is, the diagonal approximation of N in pixel space. A better approximation, which will be used in DroPS, is a diagonal form of N in harmonic space.

We label a pixel with index j ($j = 1, 2, \ldots, n_{\rm pix}$), and frequency channels with index k ($k = 1, 2, \ldots, n_f$). The CMB maps are identical in all frequency channels, and are modeled as

$$(Q \pm {\rm i}U)_{\rm CMB,j,k} = -\sum_{\ell=0}^{\ell_{\rm max}} \sum_{m=-\ell}^{\ell} (c^E_{\ell m} \pm {\rm i} c^B_{\ell m})\ {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \tag{11}$$

where $\mathbf{n}_j$ is the directional vector of the j-th pixel. Note that here $c^E_{\ell m}$ and $c^B_{\ell m}$ are harmonic coefficients of the full-sky CMB map. Degeneracy between these parameters are expected for partial sky observations.

Due to limited frequency resolution, BICEP/Keck model $v_{\rm FG}$ as a single component (thermal dust). DroPS takes a more accurate two-component model with emission from Galactic synchrotron and thermal dust. The dust maps are modeled as

$$(Q \pm {\rm i}U)_{\rm dust,j,k} = -\int W_d(\nu) f_k(\nu)\ {\rm d}\nu \sum_{\ell=0}^{\ell_{\rm max}} \sum_{m=-\ell}^{\ell} (d^E_{\ell m} \pm {\rm i} d^B_{\ell m})\ {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \tag{12}$$

where the frequency dependence frunction W_d is defined in Eq. (8) and $f_k(\nu)$ is the frequency distribution of the k-th frequency channel. Synchrotron maps are modeled

similarly,

$$(Q \pm \mathrm{i}U)_{\mathrm{sync,j,k}} = -\int W_s(\nu) f_k(\nu)\, \mathrm{d}\nu \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} (s^E_{\ell m} \pm \mathrm{i} s^B_{\ell m})\, {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \tag{13}$$

where W_s is given in Eq. (9).

The sky model $(Q + iU)_{\mathrm{sky}}$ is the sum of Eqs. (11, 12, 13). Passing the sky model through the TOD filtering and map making process, we obtain the filtered sky maps. Substracting the filtered sky maps from the observed (filtered) maps, we obtain filtered noise maps in pixel space. To compute the likelihood, we decompose the noise maps into harmonic space,

$$M_j (Q \pm \mathrm{i}U)_{\mathrm{noise,filtered,j,k}} = -\sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} (\tilde{n}^E_{\ell m} \pm \mathrm{i}\tilde{n}^B_{\ell m})\, {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j). \tag{14}$$

Here M_j is pixel value of a smoothed mask, whose edges are apodized using a C2-type kernel with an apodization scale of 2 degrees [12]. The purpose of introducing such a smoothed mask is to suppress unphysical modes due to sharp-edge and to reduce E-to-B leakage.

On the other hand, we can calculate $\tilde{n}^E_{\ell m}$ and $\tilde{n}^B_{\ell m}$ from the filtered noise maps in the base simulations, and compute their diagonal covariance $\tilde{N}^E_{\ell m} \equiv \langle |\tilde{n}^E_{\ell m}|^2 \rangle$ and $\tilde{N}^B_{\ell m} \equiv \langle |\tilde{n}^B_{\ell m}|^2 \rangle$. For most mask shapes, the m-dependences of $\tilde{N}^E_{\ell m} \equiv \langle |\tilde{n}^E_{\ell m}|^2$ and $\tilde{N}^B_{\ell m} \equiv \langle |\tilde{n}^E_{\ell m}|^2$ are very weak, and mostly reflect the statistic fluctuations due to finite sampling. Therefore, DroPS uses the m-averaged pseudo power spectra $\tilde{N}^E_\ell$ and $\tilde{N}^B_\ell$ to construct the likelihood in harmonic space. The likelihood $\propto e^{-\chi^2/2}$ is given by

$$\chi^2 = f_{\mathrm{sky}} \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} \frac{|\tilde{n}_{\ell m}|^2}{N_\ell}. \tag{15}$$

References

[1] Nabila Aghanim, Yashar Akrami, Mark Ashdown, et al. Planck 2018 results-vi. cosmological parameters. *Astronomy & Astrophysics*, 641:A6, September 2020.

[2] M. Tristram, A. J. Banday, K. M. Górski, R. Keskitalo, C. R. Lawrence, K. J. Andersen, R. B. Barreiro, J. Borrill, L. P. L. Colombo, H. K. Eriksen, R. Fernandez-Cobos, T. S. Kisner, E. Martínez-González, B. Partridge, D. Scott, T. L. Svalheim, and I. K. Wehus. Improved limits on the tensor-to-scalar ratio using BICEP and Planck data. *Physical Review D*, 105(8):083524, April 2022.

[3] P. A. R. Ade, Z. Ahmed, M. Amiri, et al. BICEP/Keck XV: The BICEP3 Cosmic Microwave Background Polarimeter and the First Three-year Data Set. *Astrophysical Journal*, 927(1):77, March 2022.

[4] Hong Li, Si-Yu Li, Yang Liu, Yong-Ping Li, and Xinmin Zhang. Tibet's window on primordial gravitational waves. *Nature Astronomy*, 2:104–106, February 2018.

[5] Junzhou Zhang, Shamik Ghosh, Jiazheng Dou, Yang Liu, Siyu Li, Jiming Chen, Jiaxin Wang, Zhaoxuan Zhang, Jacques Delabrouille, Mathieu Remazeilles, Chang Feng, Bin Hu, Hao Liu, Larissa Santos, Pengjie Zhang, Wen Zhao, Le Zhang, Zhi-Qi Huang, Hong Li, and Xinmin Zhang. Forecast of Foreground Cleaning Strategies for AliCPT-1. *Astrophysical Journal Supplement Series*, 274(2):26, October 2024.

[6] Yang Liu, Lei Ming, Marco Drewes, and Hong Li. AliCPT Sensitivity to Cosmic Reheating. *arXiv e-prints*, page arXiv:2503.21207, March 2025.

[7] Kevin Wolz, Susanna Azzoni, Carlos Hervías-Caimapo, Josquin Errard, Nicoletta Krachmalnicoff, David Alonso, Carlo Baccigalupi, Antón Baleato Lizancos, Michael L. Brown, Erminia Calabrese, Jens Chluba, Jo Dunkley, Giulio Fabbian, Nicholas Galitzki, Baptiste Jost, Magdy Morshed, and Federico Nati. The Simons Observatory: Pipeline comparison and validation for large-scale B-modes. *Astronomy & Astrophysics*, 686:A16, June 2024.

[8] Emilie Hertig, Kevin Wolz, Toshiya Namikawa, et al. The Simons Observatory: Combining cross-spectral foreground cleaning with multitracer B-mode delensing for improved constraints on inflation. *Physical Review D*, 110(4):043532, August 2024.

[9] Kevork Abazajian, Graeme E. Addison, Peter Adshead, et al. CMB-S4: Forecasting Constraints on Primordial Gravitational Waves. volume 926, page 54, February 2022.

[10] Sebastian Belkner, Julien Carron, Louis Legrand, Caterina Umiltà, Clem Pryke, Colin Bischoff, and CMB-S4 Collaboration. CMB-S4: Iterative Internal Delensing and r Constraints. *Astrophysical Journal*, 964(2):148, April 2024.

[11] BICEP/Keck Collaboration, P. A. R. Ade, Z. Ahmed, M. Amiri, D. Barkats, R. Basu Thakur, et al. BICEP/Keck XX: Component-separated maps of polarized CMB and thermal dust emission using Planck and BICEP/Keck Observations through the 2018 Observing Season. *arXiv e-prints*, page arXiv:2509.21648, September 2025.

[12] J. Grain, M. Tristram, and R. Stompor. Polarized CMB power spectrum estimation using the pure pseudo-cross-spectrum approach. *Physical Review D*, 79(12):123515, June 2009.

[13] David Alonso, Javier Sanchez, Anže Slosar, and LSST Dark Energy Science Collaboration. A unified pseudo-C_ℓ framework. *MNRAS*, 484(3):4127–4151, April 2019.

[14] Jens Chluba, James Colin Hill, and Maximilian H. Abitbol. Rethinking CMB foregrounds: systematic extension of foreground parametrizations. *MNRAS*, 472(1):1195–1213, November 2017.