

DroPS

Deriving r from Power Spectra

Zhiqi Huang

huangzhq25@mail.sysu.edu.cn

November 2025

Contents

1	Introduction	3
1.1	Primordial power spectra	3
1.2	CMB B-polarization and r	3
1.3	Ground-based CMB experiments with small-aperture telescopes	4
2	Software Documentation	5
2.1	Installing DroPS	5
2.1.1	Installing tools and libraries	5
2.1.2	Set up a Python virtual environment	6
2.1.3	Install requirements	6
2.1.4	Hacking pysm3	7
2.2	Base simulations	8
2.2.1	Generate a TOD filtering model	8
2.2.2	Generating base simulations	9
2.3	Analyzing the sky maps	9
2.3.1	Analyzing one realization of sky	11
2.3.2	Analyzing multiple realizations of sky	11
2.3.3	How to apply on real data	12
2.3.4	Comparison with other pipelines	12
2.4	Map-level Operations	14
2.4.1	Likelihood-based component separation	14
2.4.2	Displaying the maps	16
3	Technical Details	18
3.1	TOD filtering and map making	19
3.2	Noise model	21
3.3	CMB model	21
3.3.1	Cosmology	21
3.3.2	Delensing	22
3.3.3	Cosmological parameter inference	23
3.4	Foreground model	24
3.4.1	Thermal dust emission and synchrotron radiation	24
3.4.2	Spatially varying foreground: moment expansion method	25
3.4.3	Frequency decorrelation: Talor expansion method	29
3.5	Covariance matrix of band powers	29
3.6	Component separation	32

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

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

and

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

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]. A broad range of inflationary models remains consistent with $0 < r < 0.032$ [3].

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

A key objective of many ground-based, small-aperture telescope (SAT) CMB experiments—including BICEP/Keck [4], AliCPT [5, 6, 7], the Simons Observatory SATs [8, 9], and the initially planned CMB-S4 SATs [10, 11]¹—is to measure the degree-scale CMB B-mode polarization, the unique signature of primordial gravitational waves. These telescopes observe the polarized sky emission across a wide frequency range (typically 30–300 GHz), where the raw data is a complex mixture of the CMB signal, pervasive Galactic foregrounds, atmospheric emission, and instrumental noise. Disentangling the faint primordial B-mode signal from these dominant contaminants presents a significant data challenge, necessitating sophisticated component-separation and analysis software. Our software, DroPS, is designed specifically to meet this need, providing a robust framework for inferring the tensor-to-scalar ratio r from the polarization data acquired by ground-based SATs.

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

2 Software Documentation

Designed for efficiency, DroPS has minimal hardware requirements. The analyses presented in this documentation were performed efficiently on a conventional laptop. A typical setup for running DroPS is a standard personal computer with a few hundred gigabytes of free storage.

2.1 Installing DroPS

While this guide has been tested on Ubuntu 24.04 LTS, the process is generally straightforward on other Linux distributions. For Windows and macOS users, some steps may require slight modifications, particularly regarding environment setup or package installation.

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.ps1
```

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" >> ${HOME}/.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 dependencies

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

2.1.4 Hacking pysm3

While hacking a Python package is generally not recommended, we have made a pragmatic choice to do so for significant gains in the efficiency of large-scale CMB simulations. Note that this step is entirely optional and is only required for simulation work; users who solely wish to analyze existing sky maps can safely skip it.

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

Before analyzing real or simulated sky maps, it is necessary to first generate the base simulations—a set of simulated CMB, noise, and foreground components. These simulations represent our best-informed model of reality, encapsulating our current understanding of the cosmology, instrument, noise, and foregrounds. As such, they serve a role analogous to “training data” in machine learning². Nevertheless, systematic offsets may persist between these simulations and the actual observations.

2.2.1 Generate a TOD filtering model

A critical step in processing data from ground-based CMB experiments is the mitigation of ground and atmospheric contamination in the time-ordered data (TOD). Accurately simulating this filtering process necessitates detailed knowledge of the survey strategy, which is often unavailable during the forecasting stage. Fortunately, the overall effect is well-understood: it suppresses large-scale power (low multipoles) in the final maps and introduces non-Gaussian correlations by mixing Fourier modes. For use cases that do not require a precise, survey-specific simulation, DroPS provides a “mock filtering” tool to generate an approximate filtering matrix.

To generate the TOD filtering matrix, run:

```
python mock_filtering.py
```

and follow the prompts. 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 ℓ_{cut} , f_{low} , f_{high} , α_{low} , θ_{ℓ} , θ_m that are defined in Sec. 3.1.

²Note that this is purely a conceptual analogy; DroPS does not employ ML techniques.

2.2.2 Generating base simulations

Now we are ready to generate the base simulations for given experiments. 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 [5], SO-SAT [8] and CMB-S4-SAT [11]). In this documentation we will mainly use the “AliCPT” experiment as examples. We will also use SO for comparison between different foreground modeling and between DroPS and other pipelines in the literature [8].

To begin with, you can simulate noise/cmb/foreground maps for the demo-experiment AliCPT.

```
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 describe 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. A robust r measurement pipeline must be insensitive to the finer details of the foreground modeling.

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

2.3 Analyzing the sky maps

In the previous section, we generated a set of base simulations. These serve as a reference library, constructed from a specific noise model, the foreground model [’d0’, ’s0’], and a set of fiducial r values within a Λ CDM cosmology. Now we can simulate an “observed sky” map. This mock observation uses the same noise model but can employ a different foreground model and, crucially, an r value that is independent of those in the base simulations. DroPS will determine the value of r in this “observed sky” by statistically comparing it against the base simulation library. Conceptually, this “observed sky”

Table 1: Key configurations of DroPS demo-experiments; ℓ_{knee} and α_{knee} are defined in Sec. 3.2

	frequency (GHz)	beam FWHM (arcmin)	noise-level σ_{white} (μK -arcmin)	ℓ_{knee}	α_{knee}
Test ($f_{\text{sky}} \approx 0.1$)	30	60	1	60	-1.7
	95	20	1	60	-1.7
	150	20	1	60	-3
	270	10	1	60	-3
AliCPT ($f_{\text{sky}} \approx 0.14$) (+Planck)	27	97	47	30	-2.4
	40	65	34	30	-2.4
	90	16.2	2.2	50	-2.5
	150	9.7	3.3	50	-3
	217	4.9	50	0 ⁺	-3
SO ($f_{\text{sky}} \approx 0.1$)	27	91	33	15	-2.4
	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 ($f_{\text{sky}} \approx 0.1$)	30	72.8	2.5	60	-1.7
	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

is analogous to a “test sample” in machine learning. We may also generate many realizations of the “observed sky” to assess the average offset (bias) between the reconstructed and input r values.

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`, a path that is also specified in the configuration file `AliCPT/AliCPT_ana_config.txt`.

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 ${i} dls1
    python mainpipe.py AliCPT/AliCPT_ana_config.txt maps/s${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 in Figure 1.

2.3.3 How to apply on real data

A real experiment comes with its own map making and noise simulation pipelines, and provides base simulations and 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 frequency maps.

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 [6]. Here we further compare DroPS with three pipelines that have been applied on simulation data of SO-SAT (Wolz et al. 2024 [8]). We adopt the optimistic configuration of SO-SAT. The input fiducial r is zero, and in the analyses unphysical negative r is allowed³. The results are shown in Figure 2. From top to bottom, foreground models ['d0', 's0'], ['d1', 's1'], and ['dm', 'sm'] are used, respectively⁴. From the left where the reconstructed means of r and the

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

⁴The model ['dm', 'sm'] does not come with pysm3 and needs to be prepared by the user. See [8] for its definition.

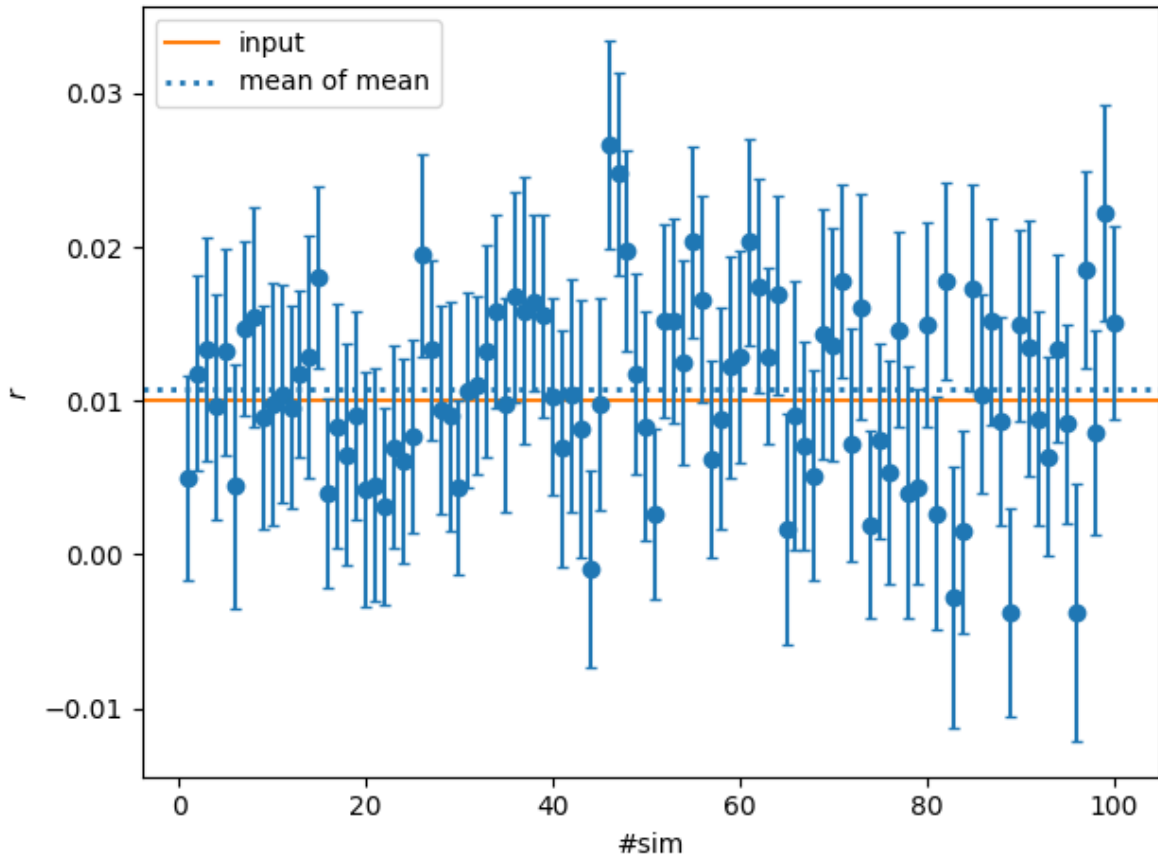


Figure 1: Reconstructed r for 100 realizations of sky with different random seeds. For each sky, the r value is reconstructed by comparing the sky with 300 base simulations. The 100 realizations of sky are simulated with foreground model ['d1', 's1'] (spatially varying SED of synchrotron and dust emission),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

bias (mean of the reconstructed means 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. Consequently, smoothing frequency maps to a common resolution inevitably alters their spectral energy distributions. This inherent limitation explains why, despite its popularity in simulations that neglect TOD filtering effects [8], ILC is often superseded by more computationally intensive, likelihood-based methods in real-world analyses, such as those conducted by BICEP/Keck [12].

DroPS also provides a map-level component separation code. To demonstrate how the 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 frequency maps with root name maps/r3AliCPT0_.

Now we do component separation

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

The code uses Stochastic Gradient Langevin Dynamics (SGLD) method to search the maximum of the likelihood described in Sec. 3.6. It typically takes a few days to run compsep.py for the first time, on a personal computer. When the gradient templates are built, the code runs much faster, typically takes only a few hours on a personal computer. Typically you need to run it multiple times to obtain a well converged CMB B-mode map. Figure 3 shows the result for the maps simulated with AliCPT/AliCPT_sim_config.txt.

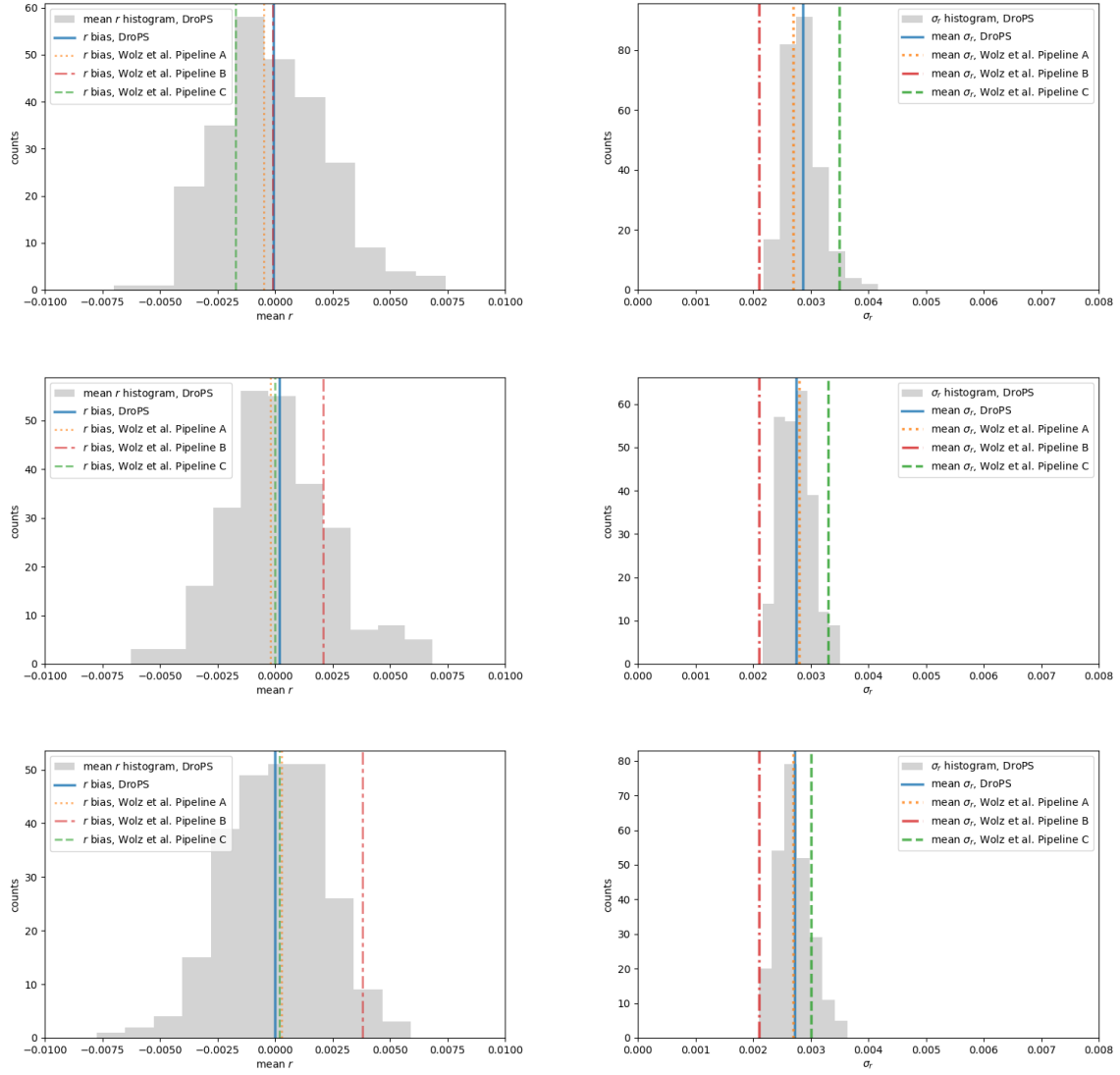


Figure 2: Comparing DroPS with three pipelines that are described in Wolf et al. [8]. From top to bottom, foreground models $['d0', 's0']$, $['d', 's1']$, and $['dm', 'sm']$ are used, respectively. Left and right columns are the distribution of reconstructed mean r and σ_r , respectively.

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 takes even longer time to build. Fortunately, the gradient template can be built in a parallel way. The template can be built for $\ell \in [\ell_{\min}, \ell_{\max}]$ by running

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

Thus, with parallel I/O that is commonly supported in modern architectures, the task can be decomposed into many sub-tasks with different l_{\min} and l_{\max} . You need $\sim 100\text{GB}$ disk space to store all the gradient templates.

2.4.2 Displaying the maps

DroPS provides a few tools to display the maps on the screen.

The script `utils/viewfits.py` displays the fits maps.

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

If you have a map `example_IQU.fits` with stokes I, Q, U components, you can run

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

with `mapind = 0, 1, 2` for I, Q, U components, respectively.

DroPS often saves partial-sky maps in `numpy` format. Now suppose the mask file is `mask.fits`. To display these masked-sky maps, use `utils/viewnumpy.py`:

```
python utils/viewnumpy.py mask.fits example_IQU.npy mapind
```

with `mapind = 0, 1, 2` for I, Q, U components, respectively.

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

```
python utils/viewEB.py mask.fits example_IQU.npy E/B
```

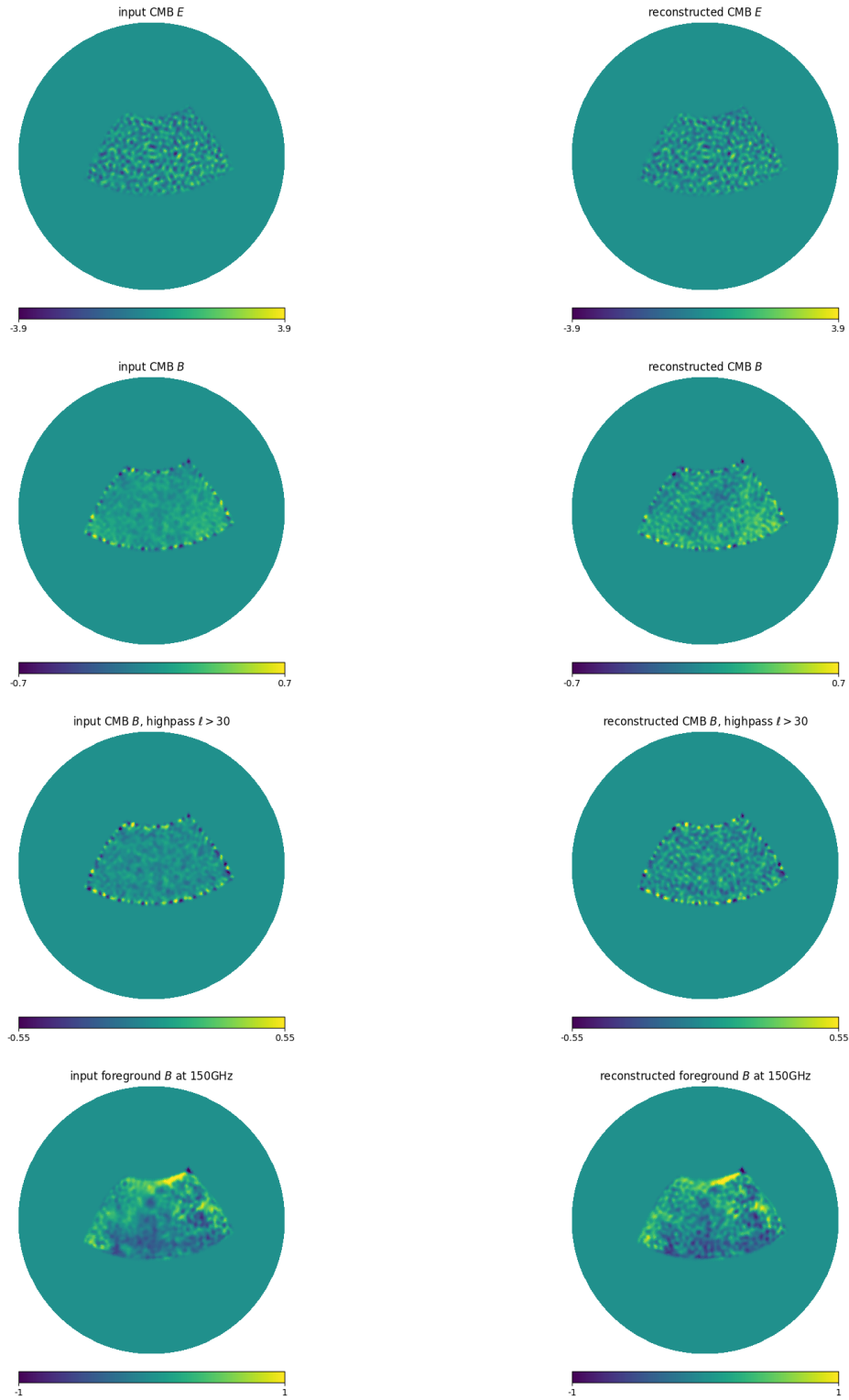



Figure 3: Component separation for maps simulated with AliCPT/iCPT_sim_config.txt. All maps are smoothed with top-hat low-pass filter at $\ell \leq 160$.

3 Technical Details

In this section I assume the reader has cosmological background and is familiar with Healpix [13].

DroPS analyzes the power spectrum by dividing the multipole range into a few bins and computing the band powers⁵ for masked maps using NaMaster [14]. For measuring the tensor-to-scalar ratio r , the analysis typically focuses on the range $20 \lesssim \ell \lesssim 200$ with a small number of bins. A resolution of $n_{\text{side}} = 256$ is sufficient for this purpose, while $n_{\text{side}} = 128$ may be adequate for quick estimates. However, caution is advised when performing operations such as changing the map resolution or rotating to a different coordinate system, as these can introduce significant E-to-B leakage in low-resolution maps. For any analysis involving these manipulations, we recommend using a higher resolution of $n_{\text{side}} = 512$.

The data vector for the likelihood analysis consists of the band powers derived from the filtered sky maps. These are constructed as follows: band powers between different frequency channels are computed directly as cross-power spectra, while for a single channel, they are obtained from the cross-power of pairs of independent season maps within that channel. This design ensures that the expectation value of the noise band power is zero in all cases, which effectively eliminates bias from imperfect noise modeling. This approach represents a standard in modern CMB analysis.

Let n_{bin} be the number of ℓ -bins, n_ν the number of frequency channels, and n_{fld} the number of fields (e.g., TT, TE, EE, BB). The data vector comprises the band powers $D_\ell^{XY}(\nu_1, \nu_2)$, indexed by:

- Multipole bin centered at ℓ ;
- Field XY ;
- All ranked frequency pairs (ν_1, ν_2) with $\nu_1 \leq \nu_2$.

As noted previously, for $\nu_1 = \nu_2$, the band power is computed by averaging over season-map cross-spectra. The total number of ranked frequency pairs is $n_\nu(n_\nu + 1)/2$, making the full data vector length $N_d = \frac{1}{2}n_{\text{fld}}n_{\text{bin}}n_\nu(n_\nu + 1)$. For r -measurement with limited computing or storage resources, we recommend using only the BB spectra ($n_{\text{fld}} = 1$), as they contain nearly all the information on r .

The fundamental approach is to evaluate the likelihood of the observed data vector $D_\ell^{XY}(\nu_1, \nu_2)$ (abstractly written as D^{obs}) given a model prediction (abstractly written as

⁵A band power represents the average power spectrum within a given ℓ bin.

D^{model}). Taking a Gaussian approximation, the likelihood is:

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

The model is constructed from the signal components:

$$\mathbf{D}^{\text{model}} = \mathbf{D}^{\text{CMB}} + \mathbf{D}^{\text{fg}}, \quad (4)$$

where D^{CMB} and D^{fg} are the expected band powers from the CMB and foregrounds. The noise expectation is zero due to the exclusive use of cross-correlations. The total covariance Cov encompasses all variance from noise, foregrounds, and CMB.

Prior to discussing the specific noise, CMB, and foreground models or the technical estimation of Cov, we will outline two key components: the map-making simulators for the base simulations, and how the TOD filtering is modeled throughout the pipeline.

3.1 TOD filtering and map making

A critical preprocessing step in ground-based CMB experiments is the TOD filtering, which isolates the faint CMB signal from dominant, structured ground noise. Observations from the ground are heavily contaminated by low-frequency noise from atmospheric emission and instrumental $1/f$ noise. TOD filtering operates in the time domain to identify and subtract these systematic contaminants, which often exhibit characteristic temporal or scan-synchronous patterns distinct from the CMB. By applying high-pass or more sophisticated modal filters, it suppresses large-scale noise while aiming to preserve the intrinsic CMB signal, enabling the production of high-fidelity sky maps.

Although a linear process, TOD filtering is equivalent to applying a dense matrix that is difficult to model without detailed knowledge of the experiment’s scanning strategy. Consequently, DroPS does not include full map-making pipelines for specific experiments. Instead, it provides streamlined “map-making” simulators. These simulators approximate the net effect of the pipeline by combining a few linear operations in both pixel and harmonic space, guided by the survey mask and a handful of parameters that capture the essential features of TOD filtering.

Specifically, the simulator applies the following operations to an input map:

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

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

where

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

and G_{random} is a random Gaussian number. For each matrix element G_{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 [15].
3. PMultiply the input map pixel-wise by the smoothed mask.
4. Perform a spherical harmonic transform to obtain harmonic coefficients $a_{\ell m}$.
5. Apply the filtering matrix

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

6. Perform an inverse spherical harmonic transform back to a map.
7. Multiply the resulting map pixel-wise by the original (unapodized) mask.

When

python mock_filtering

is called. It generates a pickle file that saves the random numbers (G_{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).

The “map-making” simulator is designed for computational efficiency, requiring far fewer resources than a full pipeline that processes actual time-ordered data. It is a simplified model whose purpose is not to replicate every detail, but to capture the dominant large-scale effects of TOD filtering—namely, power suppression and mode mixing—for simulation-based studies.

In the analysis stage, DroPS operates without prior knowledge of the specific TOD filtering applied in the simulations. It phenomenologically models the net filtering effect using a coupling matrix $F_{\ell\ell'}$. This matrix is structured such that its main diagonal captures the power suppression in each multipole bin, while the first off-diagonals capture the mode coupling between adjacent bins; all other elements are set to zero. By default, this matrix is calibrated empirically by performing a least-squares fit that compares the band powers of filtered versus unfiltered noise maps from the base simulations.

An important caveat is that while TOD filtering is a linear operation on the map, its effect on the power spectrum is not strictly linear. The band-diagonal coupling matrix $F_{\ell\ell'}$ is therefore a simplified model. Consequently, the design of DroPS intentionally minimizes reliance on this approximate model to ensure robustness. (See Sec. 3.3 for more details.)

3.2 Noise model

DroPS adopts a noise model for its temperature power spectrum following the established approach in Refs. [16, 8]:

$$N_\ell = N_{\text{white}} \left[1 + \left(\frac{\ell}{\ell_{\text{knee}}} \right)^{\alpha_{\text{knee}}} \right]. \quad (5)$$

Here, the $1/f$ noise component is parameterized by the knee multipole ℓ_{knee} and the slope α_{knee} . The white noise level N_{white} is derived from the configuration parameter σ_{white} (see Table 1) as

$$N_{\text{white}} = \sigma_{\text{white}}^2, \quad (6)$$

where σ_{white} must be in units of $\mu\text{K-radian}$ ⁶.

For the demo-experiments, the polarization noise spectra are simplified to $N_\ell^{EE} = N_\ell^{BB} = 2N_\ell^{TT}$. While this relation is not exact for real instruments, it provides a reasonable approximation in many cases.

It is important to note that the analytic model in Eq. (5) is used exclusively for generating the base simulations. During the data analysis stage, DroPS bypasses this model and computes the noise covariance directly from the base simulations themselves. Furthermore, the expectation value of the noise power spectrum is always zero in the analysis because DroPS exclusively employs cross-power spectra between different frequency channels or observation seasons, thereby inherently deriving noise-debiased power spectra.

3.3 CMB model

3.3.1 Cosmology

DroPS assumes standard Lambda cold dark matter (ΛCDM) with an additional parameter r . For the demo experiments, the default cosmological parameters for base simulations are listed in Table 2.

Unlike the noise statistics, which can be empirically derived from the base simulations, the CMB statistics depend on underlying cosmological parameters—most notably the unknown tensor-to-scalar ratio r . This fundamental difference necessitates a more

⁶To convert from $\mu\text{K-arcmin}$, multiply by a factor of $\frac{\pi}{10800}$.

Table 2: Default cosmological parameters for base simulations

parameter	definition	default value
$\Omega_b h^2$	baryon density	0.02242
$\Omega_c h^2$	cold dark matter density	0.11933
θ_{MC}	angular extension of sound horizon	0.0104101
τ_{re}	reionization optical depth	0.0561
$\ln(10^{10} A_s)$	logarithm amplitude of primordial scalar power	3.047
n_s	spectral index of primordial scalar power	0.9665
r	tensor-to-scalar ratio	$r_0 = 0$ or $r_1 = 0.03$

sophisticated approach for modeling the CMB band powers and their contribution to the signal covariance matrix.

3.3.2 Delensing

Gravitational lensing by large-scale structure distorts the primordial CMB, smoothing its acoustic peaks and converting a portion of the E-mode polarization into a secondary B-mode signal. This lensing-induced B-mode acts as a dominant foreground on small scales, obscuring the fainter primordial B-modes from inflationary gravitational waves. While the expectation value of this lensing B-mode power can be theoretically predicted and subtracted, the actual lensing potential in our observed Universe randomly fluctuates around this mean—a source of cosmic variance. For measurements targeting an uncertainty of $\sigma_r \lesssim 10^{-3}$, this unpredictable fluctuation becomes a significant contaminant.

Delensing addresses this by statistically reconstructing the actual lensing distortion from high-resolution CMB data or external tracers and removing its specific imprint from the observed maps. This process cleans the cosmic foreground, sharpens the CMB’s original features, and suppresses the confounding lensing B-mode power, thereby dramatically enhancing the sensitivity to primordial signals.

In DroPS, the implementation assumes that an ideal lensing potential map is available from external sources. While such a map would, in principle, allow for complete delensing, practical limitations prevent perfect removal. To model this, DroPS introduces a delensing efficiency parameter ϵ_{delens} (`delens_fac` in the configuration file), which specifies the fraction of the lensing effect to be removed. Concretely, the process is performed by rescaling the input lensing potential map by a factor of $(1 - \epsilon_{\text{delens}})$ before applying the delensing operation.

3.3.3 Cosmological parameter inference

The primary goal of DroPS is to infer the tensor-to-scalar ratio r , which is by default the only free cosmological parameter. Users can choose to vary the full set of standard Λ CDM parameters by setting `vary_cosmology = True` in the configuration file. However, since the large-scale polarization data from ground-base SATs alone provide weak constraints on the background cosmology, DroPS applies a Gaussian prior based on Planck+BAO results [1] to ensure the parameters remain within physically reasonable ranges⁷.

The base simulations are generated with two fiducial values of the tensor-to-scalar ratio: $r = 0$ and $r = r_1 > 0$. By averaging over these simulations, we obtain the expected CMB band powers, $\langle D_\ell^{\text{CMB}}(r = 0) \rangle$ and $\langle D_\ell^{\text{CMB}}(r = r_1) \rangle$. The recommended value for r_1 is 0.03 (approximately the current 90% confidence level upper limit), which is user-configurable.

Under the standard cosmological model, and assuming other parameters θ are fixed to their fiducial values θ_{fid} , the CMB band powers exhibit an approximately linear response to r . DroPS leverages this by modeling the CMB band powers as a linear interpolation between the two simulated benchmarks, which inherently includes all instrumental effects:

$$D_\ell^{\text{CMB}} = (1 - w)\langle D_\ell^{\text{CMB}}(r = 0) \rangle + w\langle D_\ell^{\text{CMB}}(r = r_1) \rangle + \Delta_{\text{correction}}. \quad (7)$$

The interpolation weight w is defined as $w \equiv \max \left[\min \left(\frac{r}{r_1}, 1 \right), -1 \right]$, effectively clamping the interpolation for $|r| > r_1$.

The correction term $\Delta_{\text{correction}}$ applies the TOD filtering matrix $F_{\ell\ell'}$ only to the deviation from this fiducial interpolated spectrum:

$$\Delta_{\text{correction}} = \sum_{\ell'} F_{\ell\ell'} \left[D_{\ell'}^{\text{CMB}}(r, \theta) - (1 - w)D_{\ell'}^{\text{CMB}}(r = 0, \theta_{\text{fid}}) - wD_{\ell'}^{\text{CMB}}(r = r_1, \theta_{\text{fid}}) \right], \quad (8)$$

This design is highly efficient for the most likely scenario in next-generation experiments, where $|r| < r_1$ and cosmological parameters are well-constrained. In this case, the term inside the brackets is small, and the filtering matrix—whose model might be approximate—is applied only to a residual, thereby minimizing the impact of any inaccuracies in the filtering model on the final result.

⁷This prior can be disabled by modifying `mainpipe.py`, but doing so may lead to poor convergence in the Monte Carlo sampling or numerical instability.

3.4 Foreground model

3.4.1 Thermal dust emission and synchrotron radiation

The primary diffuse foregrounds in CMB polarization—Galactic thermal dust and synchrotron radiation—possess spectral energy distributions (SEDs) that are distinctly different from the CMB’s blackbody spectrum. This spectral distinction is the basis for component separation, which necessitates a minimum of three frequency channels to disentangle the CMB, dust, and synchrotron signals.

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

$$W_d(\nu; \nu_{\text{ref}}, \beta_d) = \left(\frac{\nu}{\nu_{\text{ref}}} \right)^{\beta_d - 1} e^{\frac{h(\nu_{\text{ref}} - \nu)}{k_B T_{\text{CMB}}}} \left(\frac{e^{\frac{h\nu}{k_B T_{\text{CMB}}}} - 1}{e^{\frac{h\nu_{\text{ref}}}{k_B T_{\text{CMB}}}} - 1} \right)^2 \left(\frac{e^{\frac{h\nu_{\text{ref}}}{k_B T_{\text{MBB}}}} - 1}{e^{\frac{h\nu}{k_B T_{\text{MBB}}}} - 1} \right), \quad (9)$$

where $T_{\text{MBB}} \approx 20\text{ K}$ is the effective temperature of MBB spectrum, $T_{\text{CMB}} \approx 2.726\text{ K}$ is the CMB temperature, β_d is the frequency spectral index, and ν_{ref} is an arbitrarily chosen reference frequency. Because thermal dust emission dominates the high-frequency bands, we often choose a high reference frequency, e.g., $\nu_{\text{ref,dust}} \sim 300\text{ GHz}$.

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

$$W_s(\nu; \nu_{\text{ref}}, \beta_s) = \left(\frac{\nu}{\nu_{\text{ref}}} \right)^{\beta_s - 2} e^{\frac{h(\nu_{\text{ref}} - \nu)}{k_B T_{\text{CMB}}}} \left(\frac{e^{\frac{h\nu}{k_B T_{\text{CMB}}}} - 1}{e^{\frac{h\nu_{\text{ref}}}{k_B T_{\text{CMB}}}} - 1} \right)^2, \quad (10)$$

where β_s is the frequency spectral index, and ν_{ref} is an arbitrarily chosen reference frequency. Because thermal synchrotron radiation dominates the low-frequency bands, we often choose a low reference frequency, e.g., $\nu_{\text{ref, sync}} \sim 30\text{ GHz}$.

When there is no risk of confusion, we may write $W_d(\nu; \nu_{\text{ref}}, \beta_d)$ as $W_d(\nu)$, and $W_s(\nu; \nu_{\text{ref}}, \beta_s)$ as $W_s(\nu)$. When β_d (β_s) is known, we can convert a dust (synchrotron) map at frequency ν_1 to map at frequency ν_2 with a scaling factor $\frac{W_d(\nu_2)}{W_d(\nu_1)}$ ($\frac{W_s(\nu_2)}{W_s(\nu_1)}$). (Note that this ratio does not depend on the choice of reference frequency.) Finally, for a frequency band with (normalized) frequency distribution $f(\nu)$, $W_d(\nu)$ should be replaced with averaged weight $\int W_d(\nu) f(\nu) d\nu$.

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.

A blind parametrization of the dust-dust, dust-synchrotron, and synchrotron-synchrotron band powers involves $3n_{\text{fld}}n_{\text{bin}} + 2$ independent parameters. Here +2 corresponds to β_d and β_s . Even in the case where only $XY = BB$ is used, the number of foreground parameters is typically $\gtrsim 20$. This poses some challenge to the convergence of Monte Carlo Markov Chain (MCMC) simulations, which is used in DroPS for parameter inference. However, when the data contain a high-frequency channel and a low-frequency channel, which respectively give good “initial guesses” of the dust and synchrotron band powers, we empirically find the MCMC runs can reach good convergence.

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

3.4.2 Spatially varying foreground: moment expansion method

The above discussion is based on the assumption of constant β_d and β_s . For experiments that covers a significant portion of the sky, however, it is unlikely that β_d (β_s) will be strictly a constant across a large area of the sky [17]. DroPS deals with the spatial variation of foreground spectral indices with moment expansion approximation [18] or Taylor expansion methods, which can be switched on by setting `freq_decorr_model=“ME”` or `freq_decorr_model=“Taylor”` in the configuration file. Set `freq_decorr_model=None` if strictly constant β_d and β_s are wanted. By default, DroPS uses the moment expansion method, details of which are given below.

We take thermal dust emission B-mode map for example. At frequency ν , the map is

$$B(\nu, \mathbf{n}) = W_d(\nu; \nu_{\text{ref}}, \beta_d) B(\nu_{\text{ref}}, \mathbf{n}), \quad (11)$$

if there were no spatial variation of β_d . Here \mathbf{n} is the map pixel vector, and $B(\nu_{\text{ref}}, \mathbf{n})$ is the map at reference frequency (reference map). Now consider small spatial variation $\delta\beta_d(\mathbf{n})$ and expand to second order of $\delta\beta_d(\mathbf{n})$

$$B(\nu, \mathbf{n}) = \left\{ W_d(\nu; \nu_{\text{ref}}, \beta_d) + \frac{\partial W_d}{\partial \beta_d} \delta\beta_d(\mathbf{n}) + \frac{1}{2} \frac{\partial^2 W_d}{\partial \beta_d^2} [\delta\beta_d(\mathbf{n})]^2 \right\} B(\nu_{\text{ref}}, \mathbf{n}). \quad (12)$$

Since $W_d \propto \left(\frac{\nu}{\nu_{\text{ref}}}\right)^{\beta_d}$, Eq. (12) can be explicitly written as

$$B(\nu, \mathbf{n}) = W_d(\nu; \nu_{\text{ref}}, \beta_d) \left\{ 1 + \delta\beta_d(\mathbf{n}) \ln \frac{\nu}{\nu_{\text{ref}}} + \frac{1}{2} \left[\delta\beta_d(\mathbf{n}) \ln \frac{\nu}{\nu_{\text{ref}}} \right]^2 \right\} B(\nu_{\text{ref}}, \mathbf{n}). \quad (13)$$

We first consider the full-sky case and do spherical harmonic transform. Assuming $\delta\beta_d(\mathbf{n}) = \sum \beta_{\ell m} Y_{\ell m}(\mathbf{n})$ and $B(\nu, \mathbf{n}) = \sum a_{\ell m}(\nu) Y_{\ell m}(\mathbf{n})$.

$$\begin{aligned} a_{\ell m}(\nu) &= W_d(\nu; \nu_{\text{ref}}, \beta_d) \int Y_{\ell m}^*(\mathbf{n}) \left\{ 1 + \delta\beta_d(\mathbf{n}) \ln \frac{\nu}{\nu_{\text{ref}}} + \frac{1}{2} \left[\delta\beta_d(\mathbf{n}) \ln \frac{\nu}{\nu_{\text{ref}}} \right]^2 \right\} B(\nu_{\text{ref}}, \mathbf{n}) d^2\mathbf{n} \\ &= W_d(\nu; \nu_{\text{ref}}, \beta_d) \left[a_{\ell m}^{\text{ref}} + \ln \frac{\nu}{\nu_{\text{ref}}} g_{\ell m}^{\text{ref}} + \left(\ln \frac{\nu}{\nu_{\text{ref}}} \right)^2 h_{\ell m}^{\text{ref}} \right] \end{aligned} \quad (14)$$

where

$$a_{\ell m}^{\text{ref}} = a_{\ell m}(\nu_{\text{ref}}), \quad (15)$$

$$g_{\ell m}^{\text{ref}} = (-1)^m \sqrt{\frac{2\ell+1}{4\pi}} \sum_{\ell_1, m_1, \ell_2, m_2} a_{\ell_1, m_1}^{\text{ref}} \beta_{\ell_2, m_2} \sqrt{(2\ell_1+1)(2\ell_2+1)} \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & -m \end{pmatrix}, \quad (16)$$

and

$$\begin{aligned} h_{\ell m}^{\text{ref}} &= (-1)^m \frac{\sqrt{2\ell+1}}{8\pi} \sum_{\ell_1, m_1, \ell_2, m_2, \ell_3, m_3} a_{\ell_1, m_1}^{\text{ref}} \beta_{\ell_2, m_2} \beta_{\ell_3, m_3} \sqrt{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)} \\ &\times \sum_{\ell_4, m_4} (-1)^{m_4} (2\ell_4+1) \begin{pmatrix} \ell_2 & \ell_3 & \ell_4 \\ m_2 & m_3 & -m_4 \end{pmatrix} \begin{pmatrix} \ell_2 & \ell_3 & \ell_4 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell_4 & \ell_1 & \ell \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell_4 & \ell_1 & \ell \\ m_4 & m_1 & -m \end{pmatrix}. \end{aligned} \quad (17)$$

Assuming $\delta\beta_d(\mathbf{n})$ is a random Gaussian field that is uncorrelated with the reference map $B(\nu_{\text{ref}}, \mathbf{n})$ and keeping only up to second order of $\delta\beta_d$, we have

$$\langle (a_{\ell m}^{\text{ref}})^* g_{\ell m}^{\text{ref}} \rangle = 0, \quad (18)$$

$$\langle (h_{\ell m}^{\text{ref}})^* g_{\ell m}^{\text{ref}} \rangle = 0, \quad (19)$$

and

$$\langle (h_{\ell m}^{\text{ref}})^* h_{\ell m}^{\text{ref}} \rangle = 0. \quad (20)$$

Here $\langle \cdot \rangle$ represents cosmic expectation value.

Further approximately treating $B(\nu_{\text{ref}}, \mathbf{n})$ a random Gaussian field, we may also use the power spectra $C_\ell^{\text{ref}} \equiv \langle (a_{\ell m}^{\text{ref}})^* a_{\ell m}^{\text{ref}} \rangle$ and $S_\ell \equiv \langle (\beta_{\ell m})^* \beta_{\ell m} \rangle$ to express some non-zero expectation values. Specifically, they are

$$\langle (g_{\ell m}^{\text{ref}})^* g_{\ell m}^{\text{ref}} \rangle = \frac{1}{4\pi} \sum_{\ell_1, \ell_2} C_{\ell_1}^{\text{ref}} S_{\ell_2} (2\ell_1+1)(2\ell_2+1) \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ 0 & 0 & 0 \end{pmatrix}^2, \quad (21)$$

and

$$\langle (a_{\ell m}^{\text{ref}})^* h_{\ell m}^{\text{ref}} \rangle = \frac{1}{8\pi} C_{\ell}^{\text{ref}} \sum_{\ell_2} (2\ell_2 + 1) S_{\ell_2}. \quad (22)$$

To derive Eq. (21), we have used the identity

$$\sum_{m_1, m_2} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} \ell_1 & \ell_2 & \ell'_3 \\ m_1 & m_2 & m'_3 \end{pmatrix} = \begin{cases} \frac{1}{2\ell_3+1}, & \text{if } |\ell_1 - \ell_2| \leq \ell_3 \leq \ell_1 + \ell_2, \ell_3 = \ell'_3, m_3 = m'_3 \\ 0, & \text{otherwise} \end{cases}, \quad (23)$$

namely the orthogonal property of 3- j symbol. To derive Eq. (22), we have used the identity

$$\sum_m (-1)^{\ell-m} \begin{pmatrix} \ell & \ell & n \\ m & -m & 0 \end{pmatrix} = \begin{cases} \sqrt{2\ell+1}, & \text{if } n = 0 \\ 0, & \text{else} \end{cases}. \quad (24)$$

The power spectrum between two frequency maps $B(\nu, \mathbf{n})$ and $B(\nu', \mathbf{n})$ is then

$$\begin{aligned} C_{\ell}(\nu, \nu') &= \langle a_{\ell m}(\nu) a_{\ell m}(\nu') \rangle \\ &= W_d(\nu; \nu_{\text{ref}}, \beta_d) W_d(\nu'; \nu_{\text{ref}}, \beta_d) \\ &\quad \times \left\{ C_{\ell}^{\text{ref}} + \langle (g_{\ell m}^{\text{ref}})^* g_{\ell m}^{\text{ref}} \rangle \ln \frac{\nu}{\nu_{\text{ref}}} \ln \frac{\nu'}{\nu_{\text{ref}}} + \langle (a_{\ell m}^{\text{ref}})^* h_{\ell m}^{\text{ref}} \rangle \left[\left(\ln \frac{\nu}{\nu_{\text{ref}}} \right)^2 + \left(\ln \frac{\nu'}{\nu_{\text{ref}}} \right)^2 \right] \right\} \\ &= W_d(\nu; \nu_{\text{ref}}, \beta_d) W_d(\nu'; \nu_{\text{ref}}, \beta_d) \\ &\quad \times \left\{ C_{\ell}^{\text{ref}} \left[1 + \frac{1}{8\pi} \left(\ln^2 \frac{\nu}{\nu_{\text{ref}}} + \ln^2 \frac{\nu'}{\nu_{\text{ref}}} \right) \sum_{\ell_2} (2\ell_2 + 1) S_{\ell_2} \right] \right. \\ &\quad \left. + \frac{1}{4\pi} \ln \frac{\nu}{\nu_{\text{ref}}} \ln \frac{\nu'}{\nu_{\text{ref}}} \sum_{\ell_1, \ell_2} C_{\ell_1}^{\text{ref}} S_{\ell_2} (2\ell_1 + 1)(2\ell_2 + 1) \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ 0 & 0 & 0 \end{pmatrix}^2 \right\}, \quad (25) \end{aligned}$$

The above result can be written in a more compact way

$$C_{\ell}(\nu, \nu') = W_d(\nu) W_d(\nu') \left\{ C_{\ell}^{\text{ref}} \left[1 + \mu \left(\ln^2 \frac{\nu}{\nu_{\text{ref}}} + \ln^2 \frac{\nu'}{\nu_{\text{ref}}} \right) \right] + \varepsilon_{\ell} \ln \frac{\nu}{\nu_{\text{ref}}} \ln \frac{\nu'}{\nu_{\text{ref}}} \right\}, \quad (26)$$

where

$$\mu \equiv \frac{1}{8\pi} \sum_{\ell_2} (2\ell_2 + 1) S_{\ell_2} \quad (27)$$

is proportional to the variance of $\delta\beta_d$, and

$$\varepsilon_{\ell} \equiv \frac{1}{4\pi} \sum_{\ell_1, \ell_2} C_{\ell_1}^{\text{ref}} S_{\ell_2} (2\ell_1 + 1)(2\ell_2 + 1) \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ 0 & 0 & 0 \end{pmatrix}^2 \quad (28)$$

is proportional to the convolution between β_d and the reference map.

The minimal parametrization then involve two unknown spectra C_{ℓ}^{ref} and S_{ℓ} , while μ is a derived parameter that relies on S_{ℓ} .

In general we deal with maps in a masked fractional sky. If the variance of $\delta\beta_d$ is dominated by small-scale fluctuations, we may still use Eqs. (26) as an approximation and

replace S_{ℓ_2} with $S_{\ell_2} f_{\text{fsky}}$ in Eq. (27) and Eq. (28), where f_{fsky} is the sky fraction. However, this approximation may not work very well, for the following reasons.

1. Large-scale variation of β_d has been observed [17], while small-scale variation of β_d remains a conjecture. Therefore, it is possible that the variance of $\delta\beta_d$ may be dominated by large-scale modes, whose information is partially missing in a masked sky.
2. Similarly, because for masked sky we are dealing with band powers with limited scale coverage, we cannot accurately evaluate the convolution ε_ℓ .
3. The mask and non-Gaussianity of the dust map may induce coupling between different modes. This will lead to extra scale-dependent contribution to μ and ε_ℓ .

The net effect is then the relation between μ , C_ℓ^{ref} (in form of band powers) and S_ℓ (also in form of band powers) becomes vague in the masked-sky case, and μ essentially becomes scale-dependent if mode mixing is not negligible. The final model of band powers used in DroPS is then

$$D_\ell(\nu, \nu') = W_d(\nu)W_d(\nu') \left\{ D_\ell^{\text{ref}} \left[1 + \mu_\ell \left(\ln^2 \frac{\nu}{\nu_{\text{ref}}} + \ln^2 \frac{\nu'}{\nu_{\text{ref}}} \right) \right] + \varepsilon_\ell \ln \frac{\nu}{\nu_{\text{ref}}} \ln \frac{\nu'}{\nu_{\text{ref}}} \right\}, \quad (29)$$

where D_ℓ^{ref} , μ_ℓ and ε_ℓ (both in form of band powers) are directly taken as independent parameters. To avoid overfitting and consistently keep the μ_ℓ and ε_ℓ terms perturbative, we assume an upperbound for $|\mu_\ell|$ and $|\varepsilon_\ell|$ (default: 0.05).

At the time of writing, there is no observational confirmation or rejection of spatial variation of β_d on small scales. Eqs. (27,28) imply that ε_ℓ and μ_ℓ are positive parameters, if the extra contribution from mode mixing is negligible. However, applying this physical prior can bias the r measurement if there is actually no spatial variation of β_d , and we have no clue whether the mode mixing can be safely neglected. Therefore, by default DroPS does not apply the positiveness prior. If wanted, the positiveness prior which can be turned on by setting `ME_is_positive = True` in the configuration file.

The treatment of synchrotron is exactly the same, as the dependence of W_d on β_d and W_s on β_s is identical ($W_d \propto (\nu/\nu_{\text{ref}})^{\beta_d}$ and $W_s \propto (\nu/\nu_{\text{ref}})^{\beta_s}$).

In summary, we have generalized the moment expansion method to masked sky, but still have ignored many complexities that may emerge in reality, such as possible correlation between $\delta\beta_d$, $\delta\beta_s$ and the reference map. Although some of the complexities may be absorbed into the free parameters μ_ℓ and ε_ℓ , we do not expect Eq. (29) covers all possible degrees of freedom. Nevertheless, at least for small perturbations around the base (constant-SED) model, moment expansion method is a good tool to account for the foreground complexity.

3.4.3 Frequency decorrelation: Talor expansion method

For the base (constant-SED) model of dust (synchrotron), the band powers satisfy $[D_\ell(\nu, \nu')]^2 = D_\ell(\nu, \nu)D_\ell(\nu', \nu')$. Complexity of foreground such as spatial variation of SED in general weakens the correlation between different frequency maps, that is,

$$[D_\ell(\nu, \nu')]^2 \leq D_\ell(\nu, \nu)D_\ell(\nu', \nu'), \quad (30)$$

with equality if and only if $\nu = \nu'$.

Note that, however, the moment expansion formula Eq. (29) (or the full-sky version Eq. (26)) does not explicitly follow this physically motivated inequality. Motivated by the inequality (30), DroPS provides a phenomenological model, which can be turned on by setting `fg_decorr_model = "Taylor"` in the configuration file. The idea of this model is to Taylor expand the spectral index β_d in the ν, ℓ space. Specifically, the band powers of dust in is parameterized as

$$D_\ell(\nu, \nu') = W_d(\nu; \nu_{\text{ref}}, \beta_d + \delta\beta_{d,\ell} + \gamma_d \frac{\nu}{\nu_{\text{ref}}}) W_d(\nu'; \nu_{\text{ref}}, \beta_d + \delta\beta_{d,\ell} + \gamma_d \frac{\nu'}{\nu_{\text{ref}}}) \exp \left[-\frac{\ln^2 \frac{\nu}{\nu'}}{Q_\ell} \right] D_\ell^{\text{ref}}, \quad (31)$$

where β_d and γ_d are the average SED and the frequency running of SED; D_ℓ^{ref} is the reference band power; $\delta\beta_{d,\ell}$ describes the variation of β_d in ℓ space; $Q_\ell > 0$ describes the logarithm spacing of frequency decorrelation. (Frequency decorrelation is significant if $\ln \frac{\nu}{\nu'} \gtrsim Q_\ell$.)

The moment expansion and Taylor expansion methods typically induce shifts of r on the order of a few $\times 10^{-4}$. We use the more precise demo-experiment SO to highlight the impact of foreground modeling. Figure 4 compares the reconstructed r using the base model (constant SED), the moment expansion, and the Taylor expansion methods. Each data point and error bar represent the mean and standard deviation, respectively, averaged over 250 random realizations of the "observed sky." Our analysis shows that the moment expansion method remains nearly unbiased across all tested scenarios, albeit at the cost of a modest increase in uncertainty. The Taylor expansion method also performs robustly in the presence of foreground frequency decorrelation. However, in the absence of such decorrelation, it tends to systematically underestimate the value of r .

3.5 Covariance matrix of band powers

While the covariance matrix of the band powers could, in principle, be estimated directly from simulations, this approach faces a practical challenge: the covariance matrix typically contains over a thousand free elements, which cannot be robustly constrained by the few hundred simulations typically available. To overcome this, DroPS implements

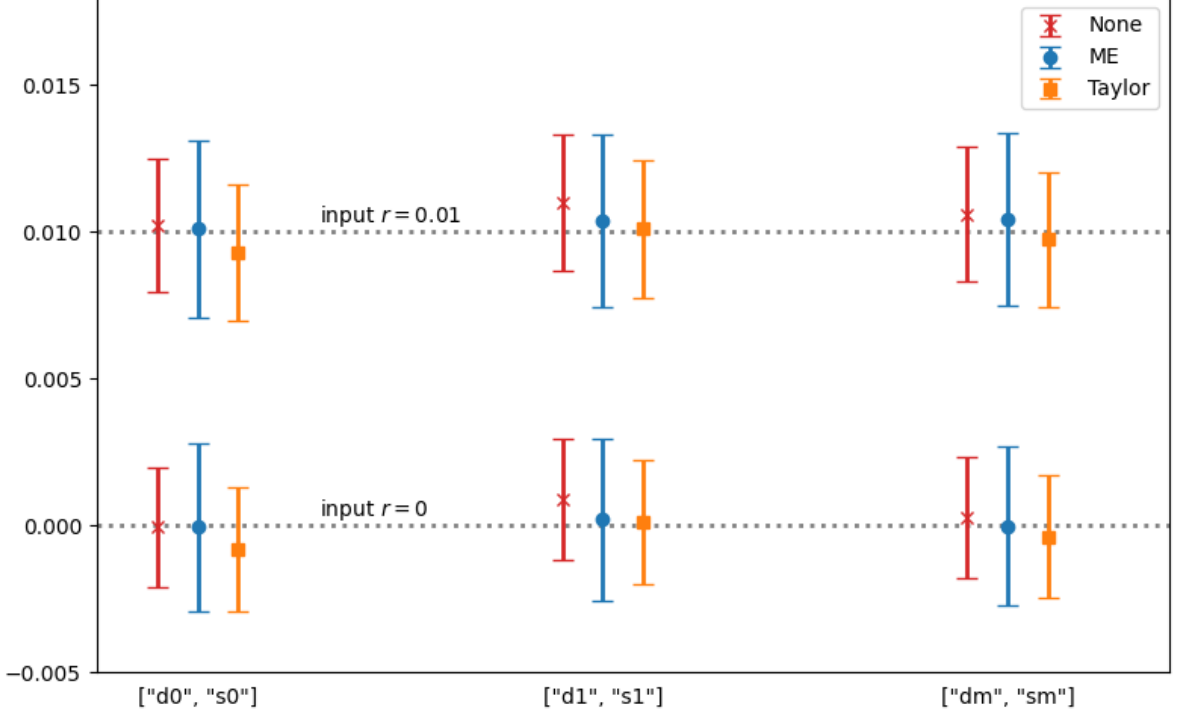


Figure 4: Comparison between base model (None), moment expansion method (ME) and Taylor expansion method (Taylor).

a two-step stabilization procedure: it applies “noise cleaning” by neglecting noise correlations across different channels or observation seasons, and enforces “scale separation” by ignoring correlations between widely separated multipole bins.

DroPS decomposes the covariance matrix into three components: the noise covariance, the signal covariance, and the signal \times noise covariance. The advantage of doing so is that noise covariance can be “cleaned” with the prior knowledge that noise in different frequency channels (or observation seasons) are uncorrelated. DroPS also assumes that the noise T , E , B are mutually uncorrelated⁸. 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 assumes no correlation between different multipole bins (`ell_cross_range` = 0 in the configuration file). Optionally, it can model correlations between adjacent (`ell_cross_range` = 1) or next-to-adjacent bins (`ell_cross_range` = 2). However, simply truncating the covariance matrix to these limited off-diagonal blocks can violate its positive-definite property, introducing unphysical “ghost modes” with negative eigenvalues that lead to catastrophic overfitting. To balance information retention with numerical

⁸In the default settings where BB band powers are used, this assumption is not used.

Table 3: 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
∞	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}, \frac{1}{4}$

stability, DroPS applies suppression factors to the off-diagonal ℓ blocks. The optimal factors, listed in Table 3, are derived analytically for a small number of bins or numerically for a larger set. Specifically:

- For ell_cross_range = 1, the first off-diagonal blocks are multiplied by a factor f_1 .
- For ell_cross_range = 2, the first and second off-diagonal blocks are multiplied by factors f_1 and f_2 , respectively.

Empirical 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 5 shows an example. The simulation is done with AliCPT/AliCPT_sim_config.txt with fiducial $r = 0.01$ and random seed 5; the analyses

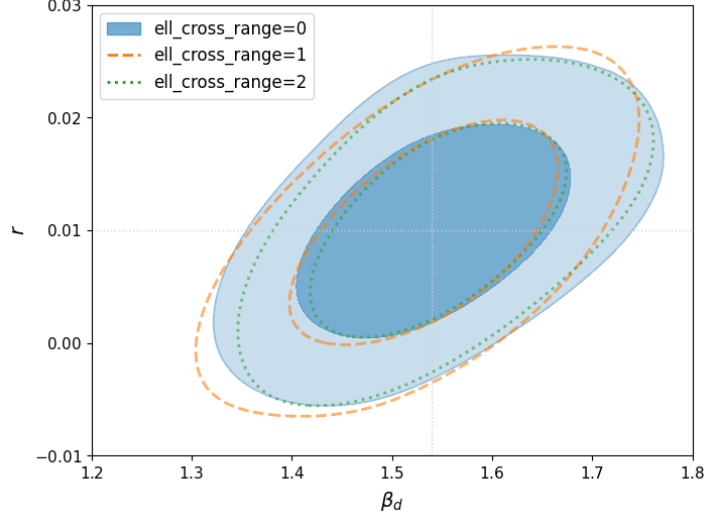


Figure 5: Impact of the correlations between adjacent ℓ bins and next-to-adjacent ℓ bins

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 \times noise covariance matrices are cosmology-dependent. DroPS pre-computes them at the two fiducial points, $r = 0$ and $r = r_1$, from the base simulations. Within the interval $0 < r < r_1$, the covariances are interpolated according to their known scaling with r : the signal covariance scales with r^2 and is thus interpolated linearly in r^2 , while the signal \times noise covariance scales with r and is interpolated linearly in r . For values $|r| > r_1$, DroPS uses the covariance evaluated at $r = r_1$, deliberately avoiding any extrapolation to prevent the introduction of unphysical modes and catastrophic overfitting.

3.6 Component separation

For map-level component separation, the likelihood $\mathcal{L} \propto e^{-\chi^2/2}$ is constructed under a Gaussian noise model, where

$$\chi^2 = v_{\text{noise}}^T N^{-1} v_{\text{noise}}, \quad (32)$$

Here, v_{noise} represents the noise frequency maps, and N is their covariance matrix. In pixel space, the dimension of v_{noise} is $n_{\text{pix}} n_\nu$, which for experiments like AliCPT or SO-SAT with $\gtrsim 10\%$ sky coverage typically exceeds 10^4 . This makes the covariance matrix N , with 10^8 elements, prohibitively large to estimate robustly from simulations and to invert efficiently.

To overcome this, the BICEP/Keck analysis [12] adopts a diagonal approximation of N in pixel space, considering only pixel auto-correlations. DroPS employs a more accurate approximation by assuming N is diagonal in harmonic space. This approach better captures the correlations inherent in the data while remaining computationally tractable.

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

$$(Q \pm iU)_{\text{CMB},j,k} = - \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} (c_{\ell m}^E \pm i c_{\ell m}^B) {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \quad (33)$$

where \mathbf{n}_j is the directional vector of the j -th pixel. The expansion bases ${}_{\pm 2}Y_{\ell m}$ are spin-weighted spherical harmonics. Note that here $c_{\ell m}^E$ and $c_{\ell m}^B$ 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 analysis models the foreground maps (v_{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

$$(Q \pm iU)_{\text{dust},j,k} = - \int W_d(\nu) f_k(\nu) d\nu \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} (d_{\ell m}^E \pm i d_{\ell m}^B) {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \quad (34)$$

where the frequency dependence function W_d is defined in Eq. (9) and $f_k(\nu)$ is the frequency distribution of the k -th frequency channel. Synchrotron maps are modeled similarly,

$$(Q \pm iU)_{\text{sync},j,k} = - \int W_s(\nu) f_k(\nu) d\nu \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} (s_{\ell m}^E \pm i s_{\ell m}^B) {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j), \quad (35)$$

where W_s is given in Eq. (10).

The sky model $(Q + iU)_{\text{sky}}$ is the sum of Eqs. (33, 34, 35). Passing the sky model through the TOD filtering and map making process, we obtain the filtered sky maps. Subtracting 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,

$$M_j(Q \pm iU)_{\text{noise,filtered},j,k} = - \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} (\tilde{n}_{\ell m}^E \pm i \tilde{n}_{\ell m}^B) {}_{\pm 2}Y_{\ell m}(\mathbf{n}_j). \quad (36)$$

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 [15]. 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 $\tilde{n}_{\ell m}^E$ and $\tilde{n}_{\ell m}^B$ from the filtered noise maps in the base simulations, and compute their diagonal covariance $\tilde{N}_{\ell m}^E \equiv \langle |\tilde{n}_{\ell m}^E|^2 \rangle$ and $\tilde{N}_{\ell m}^B \equiv \langle |\tilde{n}_{\ell m}^B|^2 \rangle$. For most mask shapes, the m -dependencies of $\tilde{N}_{\ell m}^E \equiv \langle |\tilde{n}_{\ell m}^E|^2 \rangle$ and $\tilde{N}_{\ell m}^B \equiv \langle |\tilde{n}_{\ell m}^B|^2 \rangle$ are very weak, and mostly reflect the statistic fluctuations due to finite sampling. Therefore, DroPS uses the m -averaged pseudo power spectra \tilde{N}_ℓ^E and \tilde{N}_ℓ^B to construct the likelihood in harmonic space. The likelihood $\propto e^{-\chi^2/2}$ is given by

$$\chi^2 = f_{\text{sky}} \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} \frac{|\tilde{n}_{\ell m}|^2}{N_\ell}. \quad (37)$$

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] John Ellis and David Wands. Inflation (2023). *arXiv e-prints*, page arXiv:2312.13238, December 2023.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] Yang Liu, Lei Ming, Marco Drewes, and Hong Li. AliCPT Sensitivity to Cosmic Reheating. *arXiv e-prints*, page arXiv:2503.21207, March 2025.

- [8] 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.
- [9] 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.
- [10] Kevork Abazajian, Graeme E. Addison, Peter Adshead, et al. CMB-S4: Forecasting Constraints on Primordial Gravitational Waves. volume 926, page 54, February 2022.
- [11] 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.
- [12] 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.
- [13] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *Astrophysical Journal*, 622(2):759–771, April 2005.
- [14] 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.
- [15] 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.
- [16] Peter Ade, James Aguirre, Zeeshan Ahmed, et al. The Simons Observatory: science goals and forecasts. *JCAP*, 2019(2):056, February 2019.
- [17] P. A. R. Ade, M. Amiri, S. J. Benton, et al. Analysis of Polarized Dust Emission Using Data from the First Flight of SPIDER. *Astrophysical Journal*, 978(2):130, January 2025.

- [18] S. Azzoni, M. H. Abitbol, D. Alonso, A. Gough, N. Katayama, and T. Matsumura. A minimal power-spectrum-based moment expansion for CMB B-mode searches. *JCAP*, 05:047, 2021.