

COMAP Pathfinder – Season 2 results

I. Improved data selection and processing

J. G. S. Lunde^{1,*}, N.-O. Stutzer¹, P. C. Breyse^{6,7}, D. T. Chung^{3,4,5}, K. A. Cleary², D. A. Dunne², H. K. Eriksen¹, S. E. Harper¹⁰, H. T. Ihle¹, J. W. Lamb⁹, T. J. Pearson², L. Philip^{12,16}, I. K. Wehus¹, D. P. Woody⁹, J. R. Bond^{3,14,15}, S. E. Church¹⁷, T. Gaier¹², J. O. Gundersen¹⁸, A. I. Harris¹⁹, R. Hobbs⁹, J. Kim¹³, C. R. Lawrence¹², N. Murray^{3,14,15}, H. Padmanabhan⁸, A. C. S. Readhead², T. J. Rennie^{10,11}, and D. Tolgay^{3,14} (COMAP Collaboration)

¹ Institute of Theoretical Astrophysics, University of Oslo, PO Box 1029, Blindern N-0315, Oslo, Norway

² California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA

³ Canadian Institute for Theoretical Astrophysics, University of Toronto, 60 St. George Street, Toronto, ON M5S 3H8, Canada

⁴ Dunlap Institute for Astronomy and Astrophysics, University of Toronto, 50 St. George Street, Toronto ON M5S 3H4, Canada

⁵ Department of Astronomy, Cornell University, Ithaca, NY 14853, USA

⁶ Center for Cosmology and Particle Physics, Department of Physics, New York University, 726 Broadway, New York, NY 10003, USA

⁷ Department of Physics, Southern Methodist University, Dallas TX 75275, USA

⁸ Département de Physique Théorique, Université de Genève, 24 Quai Ernest-Ansermet, CH-1211 Genève 4, Switzerland

⁹ Owens Valley Radio Observatory, California Institute of Technology, Big Pine, CA 93513, USA

¹⁰ Jodrell Bank Centre for Astrophysics, Alan Turing Building, Department of Physics and Astronomy, School of Natural Sciences, The University of Manchester, Oxford Road, Manchester M13 9PL, UK

¹¹ Department of Physics and Astronomy, University of British Columbia, Vancouver, BC V6T 1Z1, Canada

¹² Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

¹³ Department of Physics, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

¹⁴ Department of Physics, University of Toronto, 60 St. George Street, Toronto, ON M5S 1A7, Canada

¹⁵ David A. Dunlap Department of Astronomy, University of Toronto, 50 St. George Street, Toronto, ON M5S 3H4, Canada

¹⁶ Brookhaven National Laboratory, Upton NY 11973-5000, USA

¹⁷ Kavli Institute for Particle Astrophysics and Cosmology and Physics Department, Stanford University, Stanford, CA 94305, USA

¹⁸ Department of Physics, University of Miami, 1320 Campo Sano Avenue, Coral Gables, FL 33146, USA

¹⁹ Department of Astronomy, University of Maryland, College Park, MD 20742, USA

Received 14 June 2024 / Accepted 1 October 2024

ABSTRACT

The CO Mapping Array Project (COMAP) Pathfinder is performing line intensity mapping of CO emission to trace the distribution of unresolved galaxies at redshift $z \sim 3$. We present an improved version of the COMAP data processing pipeline and apply it to the first two Seasons of observations. This analysis improves on the COMAP Early Science (ES) results in several key aspects. On the observational side, all second season scans were made in constant-elevation mode, after noting that the previous Lissajous scans were associated with increased systematic errors; those scans accounted for 50% of the total Season 1 data volume. In addition, all new observations were restricted to an elevation range of 35–65 degrees to minimize sidelobe ground pickup. On the data processing side, more effective data cleaning in both the time and map domain allowed us to eliminate all data-driven power spectrum-based cuts. This increases the overall data retention and reduces the risk of signal subtraction bias. However, due to the increased sensitivity, two new pointing-correlated systematic errors have emerged, and we introduced a new map-domain PCA filter to suppress these errors. Subtracting only five out of 256 PCA modes, we find that the standard deviation of the cleaned maps decreases by 67% on large angular scales, and after applying this filter, the maps appear consistent with instrumental noise. Combining all of these improvements, we find that each hour of raw Season 2 observations yields on average 3.2 times more cleaned data compared to the ES analysis. Combining this with the increase in raw observational hours, the effective amount of data available for high-level analysis is a factor of eight higher than in the ES analysis. The resulting maps have reached an uncertainty of 25–50 μ K per voxel, providing by far the strongest constraints on cosmological CO line emission published to date.

Key words. methods: data analysis – methods: observational – galaxies: high-redshift – diffuse radiation – radio lines: galaxies

1. Introduction

Line intensity mapping (LIM) is an emerging observational technique in which the integrated spectral line emission from many

unresolved galaxies is mapped in 3D as a tracer of the cosmological large-scale structure (e.g., Kovetz et al. 2017, 2019). It represents a promising and complementary cosmological probe to, say, galaxy surveys and cosmic microwave background (CMB) observations. In particular, LIM offers the potential to survey vast cosmological volumes at high redshift in a manner that is

* Corresponding author; j.g.s.lunde@astro.uio.no

sensitive to emission from the entire galaxy population, not just the brightest objects, as is the case for high-redshift galaxy surveys (Bernal & Kovetz 2022). The most widely studied emission line for LIM purposes is the 21-cm line of neutral hydrogen (Loeb & Zaldarriaga 2004; Bandura et al. 2014; Santos et al. 2017), which is the most abundant element in the universe, but other high-frequency emission lines also appear promising (Korngut et al. 2018; Pullen et al. 2023; Akeson et al. 2019; Crites et al. 2014; CCAT-Prime Collaboration 2022; Vieira et al. 2020; Karkare et al. 2022; Ade et al. 2020), in particular due to their different and complementary physical origin as well as lower levels of astrophysical confusion, Galactic foregrounds, and radio frequency interference.

The CO Mapping Array Project (COMAP) represents one example of such an alternative approach and uses CO as the tracer species (see, e.g., Lidz et al. 2011; Pullen et al. 2013; Breyse et al. 2014). The COMAP Pathfinder instrument consists of a 19-feed¹ focal plane array observing at 26–34 GHz (Lamb et al. 2022) deployed on a 10.4 m Cassegrain telescope. This frequency range corresponds to a redshift of $z \sim 2.4$ – 3.4 for the CO(1–0) line, a period during the Epoch of Galaxy Assembly (Li et al. 2016). The Pathfinder instrument started observing in 2019, and COMAP has previously published results from the first year of data, named Season 1, in the COMAP Early Science (ES) publications (Cleary et al. 2022; Lamb et al. 2022; Foss et al. 2022; Ihle et al. 2022; Chung et al. 2022; Rennie et al. 2022; Breyse et al. 2022; Dunne et al. 2024). These ES results provided the tightest constraints on the CO power spectrum in the clustering regime published to date. Since the release of the ES results, the COMAP Pathfinder instrument has continued to observe while also implementing many important lessons learned from Season 1, both in terms of observing strategy and data processing methodology. Combining the observations from both Seasons 1 and 2 and improving the data analysis procedure, the new results improve upon the ES analysis by almost an order of magnitude in terms of power spectrum sensitivity.

This paper is the first of the COMAP Season 2 paper series, and here we present the low-level data analysis pipeline and map-level results derived from the full COMAP dataset as of the end of Season 2 (November 2023). This work builds on the corresponding Season 1 effort as summarized by Ihle et al. (2022). The Season 2 power spectrum and null-test results are presented by Stutzer et al. (2024), while Chung et al. (2024) discuss their cosmological implications in terms of structure formation constraints. In parallel with the Season 2 CO observations, the COMAP Pathfinder continues to survey the Galactic plane, with the latest results focusing on the Lambda Orionis region (Harper et al. 2024).

The remainder of this paper is structured as follows: In Sect. 2 we summarize the changes made to the observational strategy in Season 2 and provide an overview of the current status of data collection and accumulated volume. In Sect. 3 we summarize our time-ordered data (TOD) pipeline with a focus on the changes since the ES analysis. In Sect. 4 we study the statistical properties of the spectral maps produced by this pipeline while paying particular attention to our new map-domain principal component analysis (PCA) filtering and the systematic errors that this filter is designed to mitigate. In Sect. 5 we present the current data selection methodology and discuss the resulting improvements in terms of data retention in the time, map, and power spectrum domains. In Sect. 6 we present updated trans-

fer function estimates and discuss their generality with respect to non-linear filtering. Finally, we summarize and conclude in Sect. 7.

2. Data collection and observing strategy

This section first briefly summarizes the COMAP instrument and low-level data collection, which is extensively explored in Lamb et al. (2022), before exploring the changes made to the telescope and observing strategy between Seasons 1 and 2.

2.1. Instrument overview

The COMAP Pathfinder consists of a 19-feed 26–34 GHz spectrometer focal plane array fielded on a 10.4 m Cassegrain telescope located at the Owens Valley Radio Observatory (OVRO). At the central observing frequency of 30 GHz, the telescope has a beam full width at half maximum (FWHM) of $4.5'$. The 8 GHz-wide RF signal is shifted to 2–10 GHz, and then split into two 4 GHz bands. The signals from each band are passed on to a separate “ROACH-2” field-programmable gate array spectrometer, which further separates the 4 GHz-wide bands into 2 GHz-wide sidebands. The spectrometer outputs 1024 frequency channels for each of the four sidebands, for a spectral resolution of ~ 2 MHz.

The telescope is also equipped with a vane of microwave-absorbing material, which is temporarily moved into the field of view of the entire feed array before and after each hour-long “observation” to provide absolute calibration of the observed signal in temperature units. Each observation is stored in a single HDF5 file containing both the spectrometer output and various housekeeping data, and these files are referred to as “Level 1” data. Each observation of one of the three target fields is divided into ten to fifteen numbered “scans,” during which the telescope oscillates in azimuth at constant elevation, repointing ahead of the field to start a new scan each time the field has drifted through the scanning pattern.

2.2. Status of observations

Table 1 shows the raw on-sky integration time per season. COMAP Season 1 included 5200 on-sky observation hours collected from May 2019 to August 2020, while the second season included 12 300 hours collected between November 2020 and November 2023. In these publications, we present results based on a total on-sky integration time of 17 500 hours, a 3.4-fold increase compared to the ES publications.

Several changes were made to the data collection and observing strategy before and during Season 2. Most of these changes came as direct responses to important lessons learned during the Season 1 data analysis and the aim was to increase the net mapping speed, although one was necessary due to mechanical telescope issues during Season 2. Overall, these changes were highly successful, and Season 2 has a much higher data retention than Season 1, which we discuss in Sect. 5. The most important changes in the Season 2 observing strategy are the following:

1. Observations were restricted to an elevation range of 35° – 65° in order to reduce the impact of ground pick-up via the telescope’s sidelobe response. As discussed by Ihle et al. (2022), the gradient of the ground pickup changes quickly at both lower and higher elevations, and the corresponding observations were therefore discarded in the Season 1 analysis; in Season 2 we avoid these problematic elevations altogether.

¹ We refer to a full detector chain as a “feed.”

Table 1. Overview of COMAP observation season definitions.

Name	Dates	Observing hours	Notes
Season 1	05/2019–08/2020	5200	Data published in Early Science. Contains 50% Lissajous, 50% CES
Season 2a	11/2020–04/2022	7900	100% CES from this point forward
Season 2b	05/2022–11/2023	4400	After azimuth drive slowdown and sampler frequency change

Notes. Season 2 was split into two sub-seasons, respectively denoted 2a and 2b, as the telescope scanning speed was significantly reduced in May 2022 for mechanical reasons.

- Similarly, Lissajous scans were abandoned in favor of solely using constant elevation scans (CES), since Foss et al. (2022) found elevation-dependent systematic errors associated with the former.
- The two frequency detector sub-bands, that previously covered disjoint ranges of 26–30 GHz and 30–34 GHz (Lamb et al. 2022), were widened slightly, such that they now overlap; this mitigates data loss due to aliasing near the band edges.
- The acceleration of the azimuth drive was halved to increase the longevity of the drive mechanism, which started to show evidence of mechanical wear.

The latter two changes were only implemented in the second half of Season 2, and they mark the beginning of what we refer to as Season 2b. These changes are now discussed in greater detail.

2.3. Restricting the elevation range

Sidelobe pickup of the ground is a potentially worrisome systematic error for COMAP, especially since it is likely to be pointing-correlated. Even though ground pickup is primarily correlated with pointing in alt-azimuthal coordinates, the daily repeating pointing pattern of COMAP means there will still be a strong correlation in equatorial coordinates. Analysis of Season 1 observations (Foss et al. 2022), that ranged from $\sim 30^\circ$ to $\sim 75^\circ$ elevation, showed pointing-correlated systematic errors at the highest and lowest elevations.

To study this effect in greater detail, we developed a set of antenna beam pattern simulations using GRASP² for the COMAP telescope (Lamb et al. 2022), and these showed the presence of four sidelobes resulting from the four secondary support legs (SSL), with each sidelobe offset by $\sim 65^\circ$ from the pointing center. These simulations were convolved with the horizon elevation profile at the telescope site, and the results from these calculations are shown in Fig. 1. This figure clearly shows that Fields 2 and 3 experience a sharp change in ground pickup around 65° – 70° elevation, as one SSL sidelobe transitions between ground and sky. At very low elevations the ground contribution also varies rapidly for all fields as two of the other SSL sidelobes approach the horizon. While the low-level TOD pipeline removes the absolute signal offset per scan, gradients in the sidelobe pickup over the duration of a scan still lead to signal contamination. We have therefore restricted our observations to the elevation range of 35° – 65° , where one SSL sidelobe remains pointed at the ground, and the other three SSL sidelobes are safely pointing at the sky, leaving us with a nearly constant ground pickup. This change incurred little loss in observational efficiency, as almost all allocated observational time outside the new range could be reallocated to other fields within the range.

2.4. Abandoning Lissajous scans

The first season of observations contained an even distribution of Lissajous and constant elevation scans (CES), with the aim of exploring the strengths and weaknesses of each. The main strength of the Lissajous scanning technique is that it provides excellent cross-linking by observing each pixel from many angles, which is useful for suppressing correlated noise with a destriper or maximum likelihood mapmaker. The main drawback of this observing mode is that the telescope elevation varies during a single scan, resulting in varying atmosphere and ground pick-up contributions. In contrast, the telescope elevation remains fixed during a CES, producing a simpler pick-up contribution although with somewhat worse cross-linking properties.

When analyzing the Season 1 power spectra resulting from each of the two observing modes, Ihle et al. (2022) find that the Lissajous data both produce a highly significant power spectrum, especially on larger scales and fail key null tests. The CES scans, on the other hand, produce a power spectrum consistent with zero, and pass the same null tests. We therefore conclude that the significance in the Lissajous power spectrum is due to residual systematic errors. Additionally, the main advantage of Lissajous scanning, namely better cross-linking, prove virtually irrelevant because of a particular feature of the COMAP instrument: because all frequency channels in a single COMAP sideband are processed through the same backend, the instrumental $1/f$ gain fluctuations are extremely tightly correlated across each sideband. As a result, the low-level TOD pipeline is capable of simultaneously removing virtually all correlated noise from both gain and atmosphere by common-mode subtraction (see Sect. 3.5). At our current sensitivity levels, we therefore find no need to employ a complex mapmaking algorithm that fully exploits cross-linking observations, but we can rather use a computationally faster binned mapmaker (Foss et al. 2022). After Season 1 we therefore concluded that there was no strong motivation to continue with Lissajous scans, and in Season 2 we employ solely CES.

2.5. Widening of frequency bands for aliasing mitigation

As discussed in detail by Lamb et al. (2022), the COMAP instrument exhibits a small but non-negligible level of signal aliasing near the edge of each sideband. In the Season 1 analysis, this was accounted for simply by excluding the channels with aliasing power from other channels suppressed by less than 15 dB. In total, 8% of the total COMAP bandwidth is lost due to this effect, and this leads to gaps in the middle of the COMAP frequency range. Both the origin of the problem and its ultimate solution were known before the Season 1 observations started (Lamb et al. 2022), but this took time to implement.

Band-pass filters applied after the first downconversion and low-pass filters applied after the second downconversion allow significant power above the Nyquist frequency into the sampler.

² <https://www.ticra.com/software/grasp/>

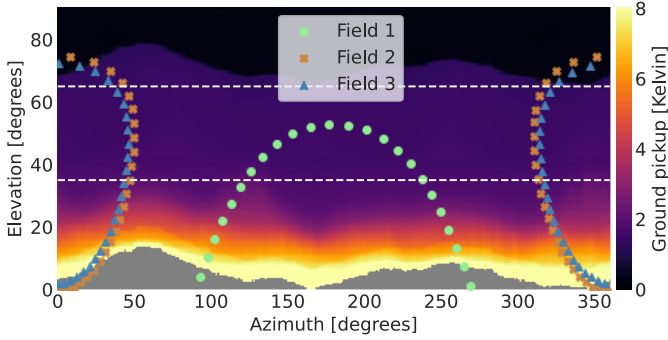


Fig. 1. Approximate sidelobe ground pickup as a function of az and el pointing, simulated by convolving a beam (simulated using GRASP) with the horizon profile (shown in gray) at the telescope site. The paths of the three fields across the sky, in half-hour intervals, are shown, as well as the Season 2 elevation limits at 35°–65°. These limits ensure minimal ground pickup gradient across the field paths.

This is then aliased into the 0–2.0 GHz observing baseband, requiring the contaminated channels to be excised. By increasing the sampling frequency from 4.000 GHz to 4.250 GHz, the Nyquist frequency is raised to 2.125 GHz, closer to the filter edges. Not only is the amount of aliased power reduced, it is also folded into frequencies above the nominal width of each 2 GHz observing band. The existing samplers were able to accommodate the higher clock speed, but the field-programmable gate array (FPGA) code had to be carefully tuned to reliably process the data. This was finally implemented from the start of Season 2b, and the aliased power is now shifted outside the nominal range of each band, such that the affected channels can be discarded without any loss in frequency coverage. The number of channels across the total frequency range is still 4096, so the “native” Season 2b channels are 2.075 MHz wide, up from 1.953 MHz.

2.6. Azimuth drive slowdown

It became clear during Season 2 that the performance of the telescope’s azimuth drives had degraded, owing to wear and tear on the drive mechanisms caused by the telescope’s high accelerations. In order to protect the drives from damage, the analog acceleration limit was reduced until the stress was judged by its audible signature to be acceptable. Though not carefully quantified, this was about an order of magnitude change, and the minimum time for a scan is therefore about a factor of three less. Additionally, the maximum velocity was reduced by a factor of two in the drive software.

Figure 2 illustrates the old (Season 2a) and new (Season 2b) pointing patterns, with the new pattern being slightly wider and around four times slower. The new realized pointing pattern is now also closer to sinusoidal since the drives are better able to ‘keep up’ with the sinusoidal pattern of the commanded position, due to the slower velocity.

With the new actually sinusoidal scanning pattern, the integration time is less uniform across each field in each observation, as the telescope spends more time pointing near the edges of the field than it previously did. However, co-adding across the observing season does smooth out the uneven integration time, based on the receiver field of view (each of the 19 feeds observes the sky at a position that is offset from the others) and field rotation (the telescope observes the fields at different angles as they move across the sky).

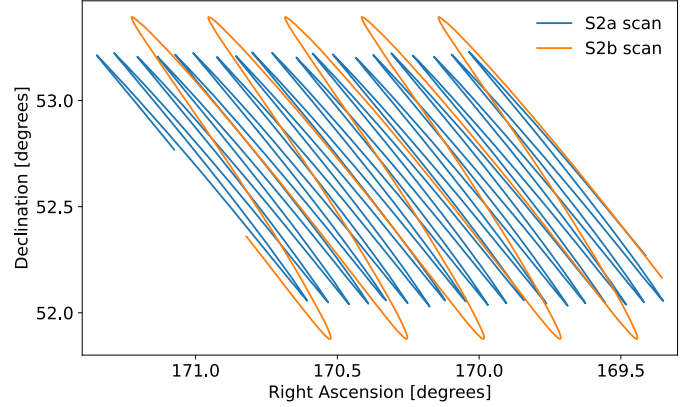


Fig. 2. Comparison of the faster pointing pattern from a Season 2a scan, and the slower pointing pattern of a Season 2b scan. Both patterns show a 5.5-minute constant elevation scan, as the field drifts across.

2.7. Data storage

With 19 feeds, 4096 native frequency channels, and a sampling rate of about 50 samples/s, COMAP collects 56 GB/hour of raw 24-bit integer data, stored losslessly as 32-bit floats. The full set of these raw data (combined with telescope housekeeping), named “Level 1”, currently span about 800 TB. These data are then filtered by our TOD pipeline into the so-called Level 2 data (Foss et al. 2022), in which a key step is to co-add the native frequency channels to 31.125 MHz width. These downsampled channels form the basis of the higher-level map-making and power spectrum algorithms. The total amount of Level 2 data is about 50 TB. Both Level 1 and Level 2 files are now losslessly compressed using the GZIP algorithm (Gailly & Adler 2023), that achieves average compression factors of 2.4 and 1.4, respectively, reducing the effective sizes of the two datasets to 330 TB and 35 TB. The lower compression factor of the filtered data is expected because the filtering leaves the data much closer to white noise, and therefore with a much higher entropy.

The files are stored in the HDF5 format (Koranne 2011), which allows seamless integration of compression. Both compression and decompression happen automatically when writing to and reading from each file. GZIP is also a relatively fast compression algorithm, taking roughly one hour to compress each hour of COMAP data on a typical single CPU core. Decompression takes a few minutes per hour of data, which is an insignificant proportion of the total pipeline runtime. HDF5 also allows for arbitrary chunking of datasets before compression. Chunking aids in optimizing performance since the Level 1 files consist of entire observations (1 hour), but the current TOD pipeline implementation (see Sect. 3.9) reads only individual scans of 5 minutes each. We partition the data into chunks of 1-minute intervals to minimize redundant decompression when accessing single scans. Other compression algorithms have been tested, and some, such as lzma³, achieve up to a 20% higher compression factor on our data. They are, however, also much slower at both compression and decompression, and they interface less easily with HDF5.

3. The COMAP TOD pipeline

This section lays out the COMAP time-domain pipeline, named 12gen, focusing on the changes from the first generation

³ <https://tukaani.org/xz/>

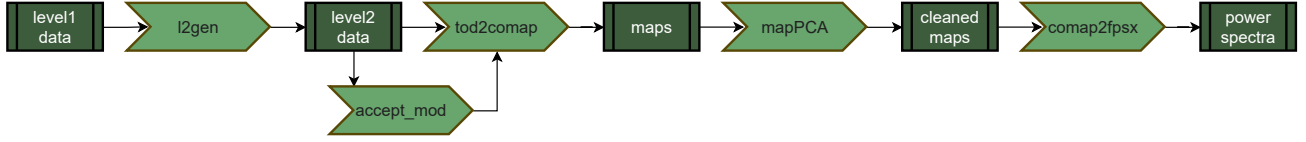


Fig. 3. Flowchart of the COMAP pipeline, from raw Level1 data to final power spectra. Data products are shown as darker boxes and pipeline code as lighter arrows. `l2gen` performs the time-domain filtering, turning raw data into cleaned Level 2 files. `accept_mod` performs scan-level data selection on cleaned data. `tod2comap` is a simple binned mapmaker. `mapPCA` performs a map-level PCA filtering. Finally, `comap2fpsx` calculates power spectra as described by Stutzer et al. (2024).

pipeline, which is described in detail by Foss et al. (2022). The pipeline has been entirely re-implemented (see Sect. 3.9) for performance and maintainability reasons but remains mathematically similar. Figure 3 shows a flowchart of the entire COMAP pipeline, of which `l2gen` is the first step.

The purpose of the TOD pipeline is to convert raw detector readout (Level 1 files) into calibrated time-domain data (Level 2 files) while performing two key operations: substantially reducing correlated noise and systematic errors, and calibrating to brightness temperature units. COMAP uses a filter-and-bin pipeline, meaning that we perform as much data cleaning as possible in the time domain, before binning the data into maps with naïve noise-weighting. This leaves us with a biased pseudo-power spectrum, that can be corrected by estimating the pipeline transfer function (Sect. 6).

The following sections explain the main filters in the TOD pipeline. The normalization (Sec. 3.3), $1/f$ filter (Sec. 3.5), calibration and downsampling (Sect. 3.8) steps remain unchanged from the ES pipeline, and are briefly summarized for completeness. We denote the data at different stages of the pipeline as $d_{\nu,t}^{\text{name}}$, where the ν, t subscript indicates data with both frequency and time dependence.

3.1. System temperature calculation

The first step of the pipeline is to calculate the system temperature T_{ν}^{sys} of each channel in the TOD. At the beginning and end of each observation, a calibration vane of known temperature is moved into the field of view of all feeds. The measured power from this “hot load”, P_{ν}^{hot} , and the temperature of the vane, T^{hot} , are interpolated between calibrations to the center of each scan. Power from a “cold load”, P_{ν}^{cold} , is calculated as the average power of individual sky scans. The system temperature is then calculated as (see Foss et al. 2022 for details)

$$T_{\nu}^{\text{sys}} = \frac{T^{\text{hot}} - T^{\text{CMB}}}{P_{\nu}^{\text{hot}}/P_{\nu}^{\text{cold}} - 1} \quad (1)$$

under the approximation that the ground, sky, and telescope share the same temperature.

3.2. Pre-pipeline masking

In ES, `l2gen` performed all frequency channel masking toward the end of the pipeline. While some masks are data-driven (specifically driven by the filtered data), others are not. We now apply the latter category of masks prior to the filters, to improve the filtering effectiveness. These are

- masking of channels that have consistently been found to be correlated with systematic errors, and have been manually flagged to always be masked;
- for data gathered before May 2022, masking of channels with significant aliased power, as outlined in Sect. 2.5;

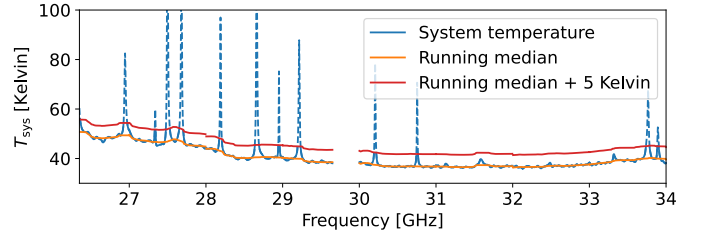


Fig. 4. Example of T_{ν}^{sys} spike masking by running median. The system temperature is shown in blue, the running median in orange, and the 5 K cut above the running median in red. The T_{ν}^{sys} values which are cut are shown as a dashed instead of solid line.

- masking of channels with system temperature spikes, as outlined below.

The system noise temperature, T_{ν}^{sys} , for each feed’s receiver chain has a series of spikes at specific frequencies, believed to result from an interaction between the polarizers and the corrugated feed horn (Lamb et al. 2022). The spikes are known to be associated with certain systematic errors, and the affected frequency channels are therefore masked out. The ES analysis used a static system temperature threshold of 85 K for masking, but the new version of `l2gen` applies a 400 channel-wide running median kernel to the data and masks all frequencies with a noise temperature of more than 5 K above the median. We repeat the running median fit and threshold operation once on the masked data, to reduce the impact of the spikes on the fit. The second iteration uses a kernel width of 150 channels. The final running median and threshold are illustrated in Fig. 4. As the system temperature can vary quite a lot across the 4 GHz range, this method fits the spikes themselves more tightly, while avoiding cutting away regions of elevated but spike-free system temperature.

The spike frequencies vary from feed to feed, and we are therefore not left with gaps in the redshift coverage of the final 3D maps. On average, we mask 6% of all frequency channels this way. However, because the affected channels are, by definition, more noisy, this only results in a loss of 3% of the sensitivity.

3.3. Normalization

The first filtering step in the TOD pipeline is to normalize the Level 1 data by dividing by a low-pass filtered version of the data and subtracting 1. The filter can be written as

$$d_{\nu,t}^{\text{norm}} = \frac{d_{\nu,t}^{\text{raw}}}{\langle d^{\text{raw}} \rangle_{\nu}} - 1, \quad \langle d^{\text{raw}} \rangle_{\nu} = \mathcal{F}^{-1}\{W \cdot \mathcal{F}\{d_{\nu,t}^{\text{raw}}\}\},$$

$$W = \left(1 + \left(\frac{f}{f_{\text{knee}}}\right)^{\alpha}\right)^{-1}, \quad (2)$$

where $\mathcal{F}$ is the Fourier transform and $\mathcal{F}^{-1}$ is the inverse Fourier transform, both performed on the time-dimension of the data,

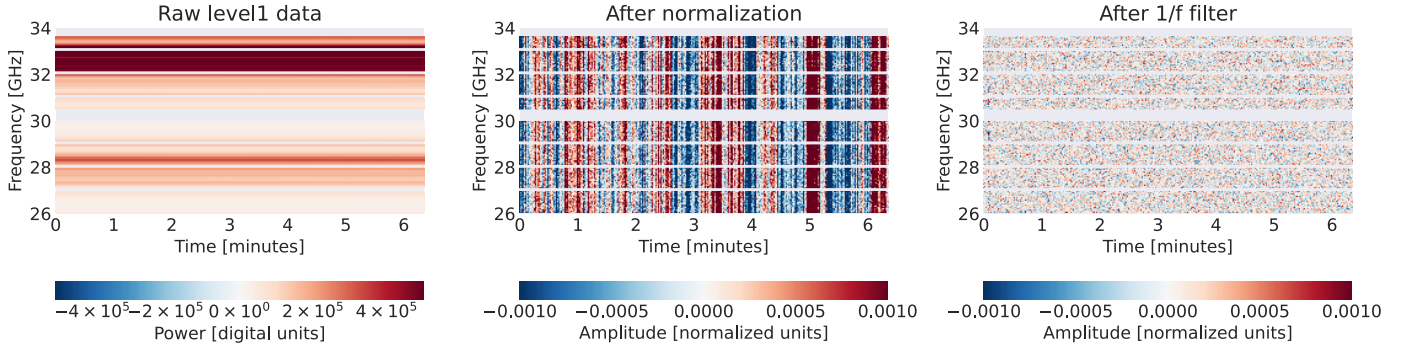


Fig. 5. Illustration of the effect of TOD normalization and $1/f$ filtering for a single feed and scan. The raw Level 1 data (left) are dominated by frequency-dependent gain variations, that correspond to the instrumental bandpass. After normalization (middle), the signal is dominated by common-mode gain fluctuations. Finally, after the $1/f$ filter is applied (right), the common-mode $1/f$ contribution has been suppressed, and the data are dominated by white noise. The horizontal gray stripes indicate channels that were masked by the pipeline. All three stages happen before absolute calibration, and the amplitudes are therefore given in arbitrary units.

and W is a low pass filter in the Fourier domain, with a spectral slope $\alpha = -4$, and a knee frequency $f_{\text{knee}} = 0.01$ Hz. This has the effect of removing all modes on 100 s timescales and longer. As the COMAP telescope crosses the entire field in 5–20 seconds, the normalization has minimal impact on the sky signal in the scanning direction, but heavily suppresses the signal perpendicular to the scanning direction, as the fields take 5–7 minutes to drift across.

The filter is performed per frequency channel, and the primary purpose of the normalization is to remove the channel-to-channel variations in the gain, making channels more comparable. After applying this filter, the white noise level in each channel becomes the same, and common-mode $1/f$ gain fluctuations become flat across frequency. As a secondary consideration, the normalization also removes the most slowly varying atmospheric and gain fluctuations, on timescales greater than 100 s.

The implications of the filter become more obvious by stating an explicit data model. The raw detector signal can be modeled by the radiometer equation as a product of gain and brightness temperature, $d_{v,t}^{\text{raw}} = g_{v,t} T_{v,t}$, where both terms can in theory vary freely. In practice, the LNA gain can be decomposed into a mean time-independent part, and a multiplicative fluctuation around this mean, such that we get $g_{v,t} = g_v(1 + \delta g_t)$, where δg_t is a small, frequency-independent term (often referred to as $1/f$ gain noise). Because all frequencies are processed by the same LNAs, these fluctuations become common-mode.

We can similarly decompose the brightness temperature into a mean and fluctuation part. However, we can make no assumption about the frequency-dependence of the fluctuation term, and therefore simply write $T_{v,t} = T_v + \delta T_{v,t}$. Here, T_v is now the system temperature, as described in Eq. (1), and $\delta T_{v,t}$ are comparatively small fluctuations around this temperature. Putting this all together we get the data model

$$\begin{aligned} d_{v,t}^{\text{raw}} &= g_{v,t} T_{v,t} = g_v(1 + \delta g_t)(T_v + \delta T_t) \\ &= g_v T_v(1 + \delta g_t + \delta T_{v,t}/T_v + \delta g_t \delta T_{v,t}/T_v) \\ &\approx g_v T_v(1 + \delta g_t + \delta T_{v,t}/T_v), \end{aligned} \quad (3)$$

where we have used the assumption that both fluctuations terms are small to approximate $\delta g_t \delta T_{v,t}/T_v \approx 0$.

Under this data model, the low-pass filtered data is assumed to take the form $\langle d_{v,t}^{\text{raw}} \rangle_v = g_v T_v$, such that the normalization filter has the effect of transforming the data into normalized fluctuations in gain and temperature around zero. Inserting Eq. (3) into

the filter as defined by Eq. (2), we get the normalized data

$$d_{v,t}^{\text{norm}} = \frac{g_v T_v(1 + \delta g_t + \delta T_{v,t}/T_v)}{g_v T_{v,t}} - 1 = \delta g_t + \delta T_{v,t}/T_v. \quad (4)$$

Technically, we defined T_v and g_v to have no time-dependence, but in the context of this filter, it makes sense to consider them the temperature and gain terms that fluctuate more slowly than ~ 100 s timescales, as these are the timescales subtracted by this filter.

The effect of the filter can be seen in the first two panels of Fig. 5, which shows the TOD of a single scan in 2D before and after the normalization. Before the normalization, the frequency-dependent gain dominates, and the time variations are invisible. After normalization, the data in each channel fluctuates around zero.

3.4. Azimuth filter

Next, we fit and subtract a linear function in azimuth, to reduce the impact of pointing-correlated systematic errors, first and foremost being ground pickup by the telescope sidelobes. This filter can be written as

$$d_{v,t}^{\text{az}} = d_{v,t}^{\text{norm}} - a_v \cdot az_t, \quad (5)$$

where a_v is fitted to the data per frequency, and az_t is the azimuth pointing of the telescope. Unlike in the ES pipeline, this filter is now fitted independently for when the telescope is moving eastward and westward, to mitigate some directional systematic effects we have seen.

In Season 1, we also employed Lissajous scans, meaning that an elevation term was also present in this equation. As we now only observe in constant elevation mode, this term falls away.

3.5. $1/f$ gain fluctuation filter

After normalization, the data are dominated primarily by gain, and secondarily by atmospheric fluctuations, and both are strongly correlated on longer timescales. Although the normalization suppresses power on all timescales longer than 100 seconds, we observe that common-mode noise still dominates the total noise budget down to ~ 1 s timescales.

To suppress this correlated noise, we apply a specific $1/f$ filter⁴ by exploiting the simple frequency behavior of the gain

⁴ The filter is referred to as the polynomial filter in our ES publications.

and atmosphere fluctuations. After we have normalized the data, the amplitude of the gain fluctuations is the same across all frequency channels, although fluctuating in time, as we see from the term δg_t in Eq. (4). The atmospheric fluctuations will enter the $\delta T_{v,t}$ term of Eq. (4). To model the atmosphere, we approximate both the atmosphere and the system temperature T_v as linear functions of frequency, such that the combined term $\delta T_{v,t}/T_v$ is itself linear. To jointly remove the gain and atmosphere fluctuations, we therefore fitted and subtracted a first-order polynomial across frequency for every time step:

$$d_{v,t}^{1/f} = d_{v,t}^{\text{point}} - (c_t^0 + c_t^1 \cdot \nu), \quad (6)$$

where c_t^0 and c_t^1 are coefficients fitted to the data each time step. To ensure that both the atmosphere and the system temperature are reasonably well approximated as linear in frequency, we fit a separate linear polynomial to the 1024 channels of each of the four 2 GHz sidebands.

This simple technique is remarkably efficient at removing $1/f$ noise, and we observe that the correlated noise is suppressed by several orders of magnitude, after which white noise dominates the uncertainty budget. This is illustrated in the last two panels of Fig. 5. After the normalization, the signal is completely dominated by common-mode $1/f$ noise. After the $1/f$ filter, the correlated noise is effectively suppressed, and we are left with almost pure white noise, as can be seen in the right panel, and further discussed in Sect. 3.10.

3.6. PCA filtering

Principal component analysis (PCA) is a common and powerful technique for dimensionality reduction (Pearson 1901). Given a data matrix $m_{v,t}$, PCA produces an ordered basis w_t^k for the columns of $m_{v,t}$, called the principal components of $m_{v,t}$. The component amplitude can then be calculated by re-projecting the components into the matrix, as $a_v^k = m_{v,t} \cdot w_t^k$. For our purposes, $m_{v,t}$ is the TOD, with frequencies as rows, and time-samples as columns. The ordering of the principal components w_t^k is such that the earlier components capture as much of the variance in the columns of $m_{v,t}$ as possible, and for any selected number of components N_{comp} , the following expression is minimized:

$$\sum_{v,t} \left(m_{v,t} - \sum_{k=1}^{N_{\text{comp}}} a_v^k w_t^k \right)^2. \quad (7)$$

In other words, PCA provides a compressed version of $m_{v,t}$, that approximates $m_{v,t}$ as the sum of an ordered set of outer products⁵

$$m_{v,t} \approx \sum_{k=1}^{N_{\text{comp}}} a_v^k w_t^k. \quad (8)$$

PCA is often employed on a dataset where the rows are interpreted as different observations, and the columns are the multi-dimensional features of these data. However, this is not a natural interpretation for our purposes, and it makes more sense to simply look at PCA as a way of compressing a 2D matrix as a sum of

⁵ PCA has several equivalent interpretations and ways of solving for the principal components. The principal components are, among other things, the eigenvectors of the covariance matrix of $m_{v,t}$. This is how we introduced the PCA in our ES publications. It is, however, both a slow way of solving for the PCA components in practice and not the best interpretation for our purposes.

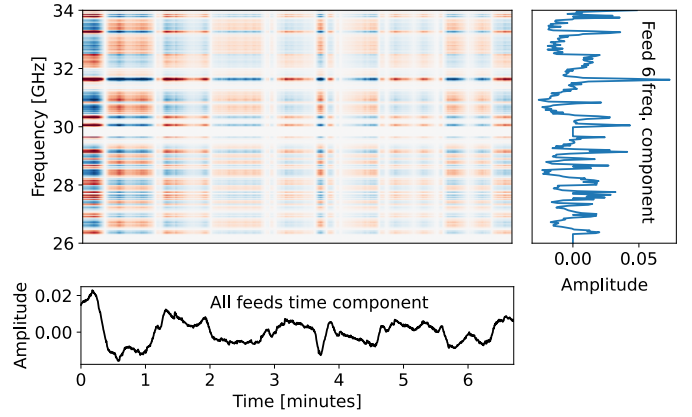


Fig. 6. Most significant component and amplitude of the all-feed PCA filter applied on scan 3354205. The bottom plot shows the first PCA component w_t^0 , which is common to all feeds. The right plot shows the corresponding amplitude a_v^0 for feed 6. The outer product of these two plots is shown in the central image and is the quantity subtracted from the TOD by the filter. As the filter is applied on normalized data, the amplitudes are all unitless, and the colorbar limits are $\pm 5 \times 10^{-4}$, half the range of those in Fig. 5.

outer products – we have no special distinction between columns and rows, and could equivalently have solved for the PCA of the transpose of $m_{v,t}$, which would swap a_v and w_t .

A PCA is often performed because one is interested in keeping the leading components, as these contain much of the information in the data. However, we subtract the leading components, because many systematic errors naturally decompose well into an outer product of a frequency vector and a time vector, while the CO signal does not (and is very weak in a single scan).

In practice, we solve for the leading principal components using a singular value decomposition algorithm (Halko et al. 2011), and then calculate the amplitudes as stated above. The N_{comp} leading components are then subtracted from the TODs, leaving us with the filtered data

$$d_{v,t}^{\text{PCA}} = m_{v,t} - \sum_{k=1}^{N_{\text{comp}}} a_v^k w_t^k. \quad (9)$$

The Season 2 COMAP pipeline employs two time-domain PCA filters, one of which was present in ES. In the following subsection, we introduce both filters and then explain how to decide the number of leading components, N_{comp} , to subtract from the data.

Figure 6 shows an example of a strong component w_t picked up by this filter in the bottom panel, and the corresponding component amplitude a_v for feed 6 in the right panel. The center image shows the resulting full frequency- and time-dependent outer product between the two functions, which is the quantity subtracted from the TOD. The time-dependent component is in this example strongly temporally correlated, and could be residual atmospheric fluctuations, that typically have $1/f$ -behavior with a steep frequency slope.

The process of calculating the principal components and subtracting them from the data constitutes a non-linear operation on the data. This has the advantage of being much more versatile against systematic errors that are difficult to model using linear filters, but the disadvantage is a more complicated impact on the CO signal itself. This is further discussed in Sect. 6.4, where our analysis shows that the PCA filter behaves linearly with respect to any sufficiently weak signal, and that, at the scan-level, all

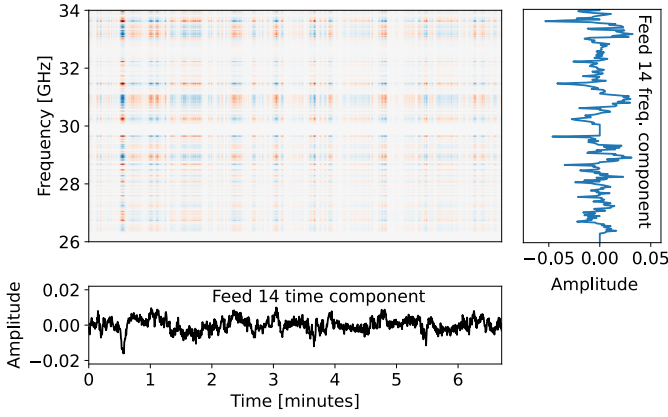


Fig. 7. Most significant component and amplitude of the per-feed PCA filter applied on scan 3354205. The bottom plot shows the first PCA component w_i^0 for feed 14, while the right plot shows the corresponding amplitude a_i^0 for feed 14. The other product of the two quantities are shown in the center plot, with colorbar limits of $\pm 5 \times 10^{-4}$.

plausible CO models (Chung et al. 2021) are sufficiently weak by several orders of magnitude.

3.6.1. The all-feed PCA filter

The all-feed PCA filter, which was also present in the ES pipeline, collapses the 19 feeds onto the frequency axis of the matrix, producing a data matrix $m_{v,t}$ of the $1/f$ -filtered data with a shape of $(N_{\text{feed}} N_{\text{freq}}, N_{\text{TOD}}) = (19 \times 4096, \sim 20\,000)$ for a scan with N_{feed} feeds, N_{freq} frequency channels, and N_{TOD} time samples. The PCA algorithm outlined above is then performed on this matrix. Combining the feed and frequency dimensions means that a feature in the data will primarily only be picked up by the filter if it is common (in the TOD) across all 19 feeds. This is primarily the case for any atmospheric contributions, and potentially standing waves that originate from the optics common to all feeds. It is, however, certainly not the case for the CO signal, which will be virtually unaffected by this filter.

3.6.2. The per-feed PCA filter

The new per-feed PCA filter has been implemented to combat systematic errors that vary from feed to feed. This filter employs the PCA algorithm outlined above on each individual feed and is performed on the output of the all-feed PCA filter. Additionally, we found that downsampling the data matrix (using inverse variance noise weighing) by a factor of 16 in the frequency direction before performing the PCA increased its ability to pick up structures in the data. The resulting data matrix $m_{v,t}$ gets the shape $(N_{\text{freq}}/16, N_{\text{TOD}}) = (256, \sim 20\,000)$ for each feed. The downsampling is only used when solving for the time-domain components w_i , and the full data matrix is used when calculating the frequency amplitudes, a_v .

Targeting each feed individually makes us more susceptible to CO signal loss, but the low signal-to-noise ratio (S/N), combined with the fact that the CO signal cannot be naturally decomposed into an outer product, makes the impact on the CO signal itself minimal. This filter appears to primarily remove components consistent with standing waves from the individual optics and electronics of each feed.

Figure 7 shows a typical strong component picked up by the per-feed PCA filter for feed 14, similar to what was shown for

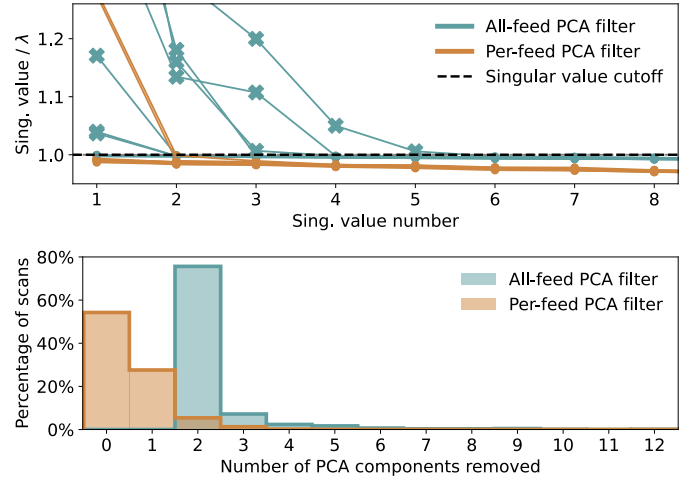


Fig. 8. *Top:* largest singular values of the all-feed and per-feed PCA filters, divided by λ , for a random selection of ten scans. PCA components with relative values above one are removed from the data and are marked with crosses in this plot. *Bottom:* number of PCA components subtracted across all scans. At least two components are always subtracted by the all-feed PCA filter.

the all-feed PCA filter in Fig. 6. The per-feed PCA filter typically picks up somewhat weaker features than the all-feed PCA filter, as it is applied after the all-feed PCA filter. The time-dependent component in Figure 7 is noticeably noisier than the equivalent component Fig. 6, also owing to the fact that it is fit with the data from 1 feed, as opposed to 19 for the all-feed filter. The physical origin of the feature shown is unknown.

3.6.3. Number of components

In the ES pipeline, the number of PCA components was fixed at four for the all-feed filter, and the per-feed filter did not exist. We now dynamically determine the required number of components for each filter, per scan. This allows us to use more components when needed, removing more systematic errors, and fewer when not needed, incurring a smaller loss of CO signal.

We subtract principal components until the components are indistinguishable from white noise, that can be inferred from the singular values of each component. Let λ be the expectation value of the largest singular value of a (N, P) Gaussian noise matrix (see Appendix A for how this value is derived). We subtract principal components until we reach a singular value below λ . However, for safe measure, we always subtract a minimum of 2 components for the all-feed PCA filter, as the signal impact of this filter is minimal.

Figure 8 shows typical singular values, relative to λ , for a random selection of scans, and a histogram of the number of components employed across all scans. The average number of PCA components subtracted is 2.3 and 0.5 for the all-feed and per-feed PCA, respectively; the most common number of components subtracted is the minimum allowed in each case: two and zero. The top part of the figure also demonstrates that there is a sharp transition between the singular values of the components that actually pick up meaningful features from the signal, and the remaining noise components. For reference, the quite significant components shown in Fig. 6 and Fig. 7 had singular value to λ ratios of 2.9 and 1.6, respectively.

3.7. Data-inferred frequency masking

After the PCA filters, we perform dynamic masking of frequency channels identified from the filtered TOD. This mainly consists of masking groups of channels that have substantially higher correlations between each other than expected from white noise, explained in more detail in Foss et al. (2022). This is assumed to be caused by substantial residuals of gain fluctuations or atmospheric signal. We did this by calculating the correlation matrix between the frequency channels and perform χ^2 tests on the white noise consistency of boxes and stripes of various sizes across this matrix. We also mask individual channels with a standard deviation significantly higher than expected from the radiometer equation.

After the frequency masking, the $1/f$ filter and PCA filters are reapplied to the data, to ensure that their performance was not degraded by misbehaving channels. The normalization and pointing filters do not need to be reapplied, as they work independently on each frequency channel.

3.8. Calibration and downsampling

The final step of the pipeline is to calibrate the data to temperature units and decrease the frequency resolution. After the normalization step, the data are in arbitrary normalized units. Using the system temperature calculated in Sect. 3.1 we calibrate each channel of the data,

$$d_{v,t}^{\text{cal}} = T_v^{\text{sys}} d_{v,t}^{\text{PCA}}. \quad (10)$$

Finally, we downsample the frequency channels from 4096 native channels to 256 science channels. The Seasons 1 and 2a frequency channels are 1.953 MHz wide, while they are 2.075 MHz wide in Season 2b, after the change in sampling frequency (Sect. 2.5). In both cases, the channels are downsampled to a grid of 31.25 MHz, that exactly corresponds to a factor 16 downsampling for the older data. For the newer data, either 15 or 16 native channels will contribute to each science channel, decided by their center frequency. The downsampling is performed with inverse-variance weighting, using the system temperature as uncertainty.

3.9. Implementation and performance

While the ES pipeline was written in Fortran, the Season 2 pipeline has been rewritten from scratch to run in Python. Performance-critical sections are either written in C++ and executed using the Ctypes package or employ optimized Python packages such as SciPy. Overall, the serial performance is similar to the ES pipeline, but the Season 2 pipeline employs a more fine-grained and optimal MPI+OpenMP parallelization, making it much faster on systems without a very large memory-to-core ratio.

The pipeline is run on a small local cluster of 16 E7-8870v3 CPUs, with a total of 288 cores, in about a week of wall time, totaling around 40 000 CPU hours for the full COMAP dataset. The time-domain processing dominates this runtime, with a typical scan taking around 20–25 minutes to process on a single CPU core.

3.10. Time-domain results

The Level 2 TODs outputted by 12gen are assumed to be almost completely uncorrelated in both time and frequency dimensions, such that the TOD are well approximated as white noise. To

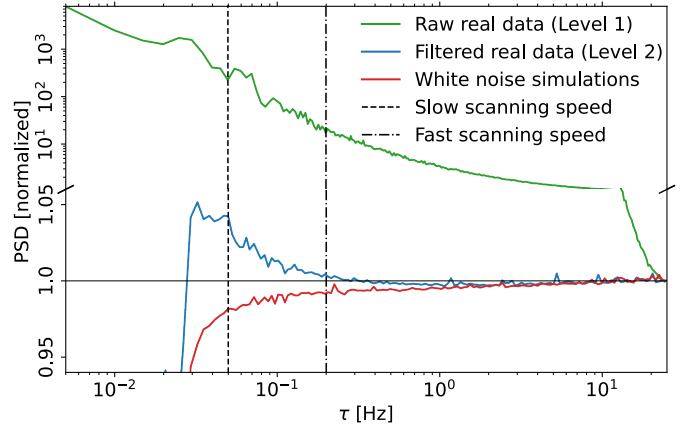


Fig. 9. Average temporal power spectrum of unfiltered (green) and filtered scans (blue), compared to correspondingly filtered white noise simulations (red). The y -axis is broken at 1.06 and is logarithmic above this. The data are averaged over scans, feeds, and frequencies, and are normalized with respect to the highest k -bin. For context, the old and new scanning frequencies (once across the field) are shown as vertical lines (dot-dashed and dashed, respectively).

quantify the correlations in the time domain we calculate the temporal power spectrum of each individual channel for all scans. Figure 9 shows this power spectrum averaged over both scans and frequencies, compared both to the equivalent power spectrum of un-filtered Level 1 data, and to that of a TOD obtained by injecting pure white noise in place of our real data into the TOD pipeline.

Since the pipeline filtering removes more data on longer timescales, the white noise simulation (red) gradually deviates from a flat power spectrum on longer timescales, until it falls rapidly at timescales below ~ 0.03 Hz due to the high-pass normalization performed in the pipeline. The power spectrum for the filtered real data (blue) also follows the same trend on short timescales, but then increases on timescales around $k = 0.2$ Hz, due to small residual $1/f$ gain fluctuations and atmosphere remaining after the processing. Below ~ 0.03 Hz, this power spectrum again falls rapidly due to the high-pass filter. The power spectrum obtained from raw data that have not gone through any filtering (green), simply increases on longer timescales as expected, due to $1/f$ gain and atmospheric fluctuations.

The difference between the blue and red spectra shows the residual correlated noise left in the data. While there is some residual correlated noise, it is an insignificant fraction of our final noise budget. We find that our real filtered data have a standard deviation only 1.7% higher than that of the filtered white noise. Compared to the amount of power we see in the unfiltered data, we see that our pipeline is very efficient at suppressing correlated noise.

4. Mapmaking and map domain filtering

4.1. The COMAP mapmaker

COMAP employs a simple binned inverse-variance noise-weighted mapmaker, identical to the one in ES (Foss et al. 2022). This can be written as

$$m_{v,p} = \frac{\sum_{t \in p} d_{v,p} / \sigma_{v,t}^2}{\sum_{t \in p} 1 / \sigma_{v,t}^2}, \quad (11)$$

where $m_{v,p}$ is an individual map voxel⁶, $d_{v,t}$ represents the time-domain data over time samples, $\sigma_{v,t}^2$ is the time-domain white noise uncertainty, and $t \in p$ means all time-samples t that observes pixel p . We assume that the white noise uncertainty $\sigma_{v,t}$ is constant (for a single feed and frequency) over the duration of a scan, and calculate it per scan as

$$\sigma_v = \sqrt{\frac{\text{Var}(d_{v,t} - d_{v,t-1})}{2}}, \quad (12)$$

then let $\sigma_{v,t} = \sigma_v$ for all time-samples within the scan. This value is also binned into maps, and is used as the uncertainty estimate of the maps throughout the rest of the analysis:

$$\sigma_{v,p}^2 = \frac{1}{\sum_{t \in p} 1/\sigma_t^2}. \quad (13)$$

In practice, we calculate per-feed maps, both because the map-PCA filter (introduced in the next section) is performed on per-feed maps, and because the cross-spectrum algorithm utilizes groups of feed-maps.

The reason for not using more sophisticated mapmaking schemes, such as destriping (Keihänen et al. 2010) or maximum likelihood mapmaking (Tegmark 1997), is partially of necessity – the COMAP TOD dataset is many hundreds of TB, making iterative algorithms difficult. However, the TOD pipeline has also proven remarkably capable of cleaning most unwanted systematic errors from the data, especially correlated $1/f$ noise, as we saw from Fig. 9. As the main purpose of more sophisticated mapmaking techniques is dealing with correlated noise, COMAP is served well with a simple binned mapmaker.

The mapmaking algorithm is identical to the one used for the ES analysis. However, as with 12gen the actual implementation has been rewritten from scratch in Python and C++, with a focus on optimal parallelization and utilization of both MPI and OpenMP.

4.2. Map-domain PCA filtering

The pipeline now employs a PCA filtering step also in the map domain, in addition to the one we apply at the TOD level. This technique is almost entirely analogous to the PCA foreground subtraction often employed in 21 cm LIM experiments (Chang et al. 2010; Masui et al. 2013; Anderson et al. 2018), although we did not employ it to subtract foregrounds. The primary purpose of this filter is to mitigate a couple of pointing-correlated systematic errors (see the next subsection) which proved challenging to remove entirely in the time domain. The method is similar to the TOD PCA algorithm from Sect. 3.6, but instead of having the TOD data matrix $d_{v,t}$, we have a map $m_{v,p}$ with one frequency and one (flattened) pixel dimension. The data matrix then gets the shape $(N_v, N_p) = (256, 14\,400)$, with $N_v = 256$ being the number of frequency channels in the map and $N_p = 120 \cdot 120 = 14\,400$ the number of pixels in each frequency slice (although many of the pixels in each individual feed-map are never observed).

The technique we employ here is technically a slight generalization of the PCA problem, as we want to weigh individual voxels by their uncertainty when solving for the components and amplitudes⁷. This is not possible in the regular PCA frame-

work without also morphing the modes one is trying to fit, as we explain in Appendix B. As shown in Sect. 3.6, the first principal component w_p^0 and its amplitude a_v^0 are the vectors that minimize the value of the expression

$$\sum_{v,p} (m_{v,p} - a_v^0 w_p^0)^2. \quad (14)$$

In other words, they are the two vectors such that their outer product explains as much of the variance in $m_{v,p}$ as possible. This formulation of the PCA makes it obvious how to generalize the problem to include weighting for individual matrix elements: we can simply minimize the following sum,

$$\sum_{v,p} \frac{(m_{v,p} - a_v^0 w_p^0)^2}{\sigma_{v,p}^2}, \quad (15)$$

where $\sigma_{v,p}$ is the uncertainty in each voxel. Minimizing Eq. (15) gives us the vectors a_v^0 and w_p^0 for which the outer product $a_v^0 w_p^0$ explains as much of the variance in $d_{v,p}$ as possible, weighted by $\sigma_{v,p}$. The resulting map $d_{v,p} - a_v^0 w_p^0$ represents the filtered map. The process can then be repeated any number of times, solving for and subtracting a new set of vectors. We minimize the expression in Eq. (15) iteratively with an algorithm outlined in Appendix B, where we also explain why this is not equivalent to simply performing the PCA on a noise-weighted map $d_{v,p}/\sigma_{v,p}$. Due to the large similarity of our technique to a regular PCA, we simply refer to this filter as a PCA filter.

As for the TOD PCA filter, a selected number of components are subtracted from the data maps

$$m_{v,p}^{\text{mPCA}} = m_{v,p} - \sum_{i=1}^{N_{\text{comp}}} a_v^i w_p^i, \quad (16)$$

This filtering is performed per feed, as the systematic errors outlined in the next subsection manifest differently in different feeds. We have chosen $N_{\text{comp}} = 5$, which is further explained in Sect. 4.3.3. Because the COMAP scanning strategy stayed the same throughout Seasons 1 and 2a but changed with the azimuth slowdown of Season 2b, we apply the map-PCA separately to the former and latter, as the pointing-correlated effects we are trying to remove might also be different.

4.3. Newly discovered systematic effects

The two most prominent new systematic errors discovered in the second season of observations have been dubbed the “turn-around” and “start-of-scan” effects. They have in common that they are difficult to model in the time domain, subtle in individual scans, but strongly pointing-correlated, and they therefore show up as large-scale features in the final maps. Additionally, they are present to varying extents in all feeds, have similar quantitative behavior in the map-domain, and can both be removed effectively with the map-PCA. The effects are discussed in the subsections below, with further analysis shown in Appendix C.

4.3.1. The turn-around effect

The so-called “turn-around” effect can be observed as strongly coherent excess power near the edges of the scan pattern, where the telescope reverses direction in azimuth. Illustrations of this effect can be seen in the first row, and partially in the third row, of Fig. 10. The feature manifests at the top and bottom of the

⁶ A voxel here is the 3D equivalent of a pixel, with two angular dimensions and a redshift (frequency) dimension. Here we separate the voxel dimensions into frequencies v and pixels p .

⁷ For the time-domain PCA, it was enough to weight individual channels, with all time-samples in that channel sharing the same weight. This can be done with a normal PCA, as we show in Appendix B.

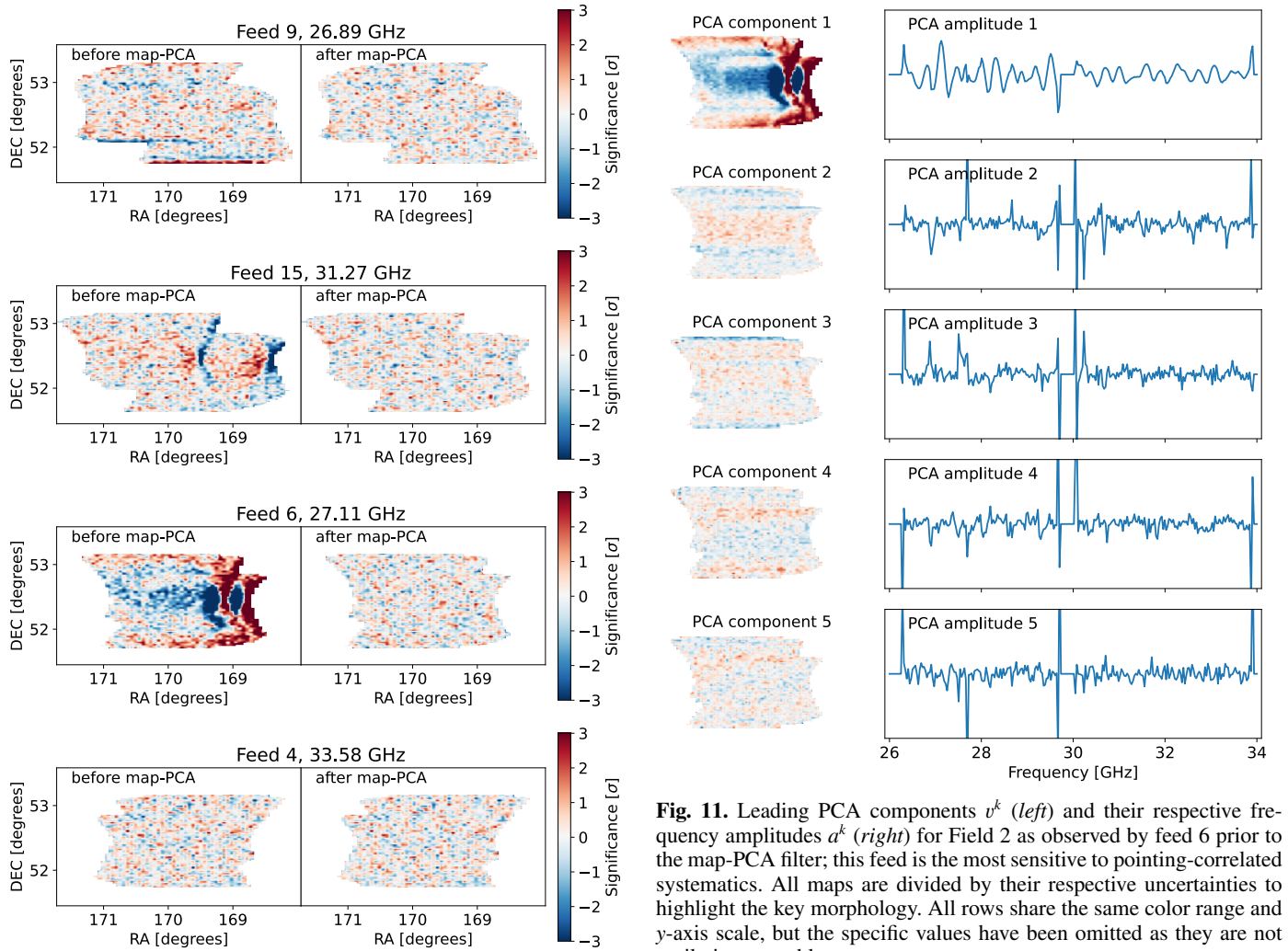


Fig. 10. Selection of individual frequency maps from Field 2 before (*left*) and after (*right*) the map-domain PCA filter. Because of the uneven sensitivity among the pixels, and to emphasize the relevant systematic effects, all maps have been divided by their white noise uncertainty. From top to bottom, each row shows maps that are dominated by (1) the turn-around effect; (2) the start-of-scan effect; (3) both effects simultaneously; and (4) neither effect. Both effects appear to manifest twice, on two slightly offset maps. This offset effect originates from the physical placement of the feeds, as they observe the fields as they are both rising and setting on the sky. In the equatorial coordinate system of these maps, the telescope scans vertically, and the field drifts from left to right.

maps, as this is where the telescope turn-around happens for Field 2 in equatorial coordinates. The feature oscillates slowly across the frequency domain, and the leading theory of its origin is some standing wave oscillation induced by mechanical vibrations. The effect is somewhat less pronounced in Season 2b, after the reduction in telescope pointing speed and acceleration, but it is still present.

4.3.2. The start-of-scan effect

A related effect is called the “start-of-scan” effect, that is a wave-like feature in frequency that occurs at the beginning of every scan and decays exponentially with a mean lifetime of around 19 s. As the telescope always starts each scan at the same side of each sky field, this systematic effect shows up in the map domain

Fig. 11. Leading PCA components v^k (*left*) and their respective frequency amplitudes a^k (*right*) for Field 2 as observed by feed 6 prior to the map-PCA filter; this feed is the most sensitive to pointing-correlated systematics. All maps are divided by their respective uncertainties to highlight the key morphology. All rows share the same color range and y-axis scale, but the specific values have been omitted as they are not easily interpretable.

as a strong feature on the Eastern edge of the map, as can be seen in the second and third rows of Fig. 10. Next to the strong positive or negative signal (this varies by frequency) at the very edge of the map, the opposite power, at lower amplitude, can be observed as we move Westward across the map. This opposite power is simply a ringing feature from the normalization performed during the pipeline (see Appendix C for details).

The exact origins of the “start-of-scan” effect are unknown, but the fact that it only happens at the beginning of scans (that are separated by a repointing to catch up with the field), and disappears during constant elevation scanning, suggests that a potential candidate is mechanical vibrations induced by the repointing. We also observe the effect to be mostly associated with one of the four DCM1s (first downconversion module), namely DCM1-2, relating to feeds 6, 14, 15, 16, and 17. The effect’s strong correlation with DCM1-2 points to a possible source in the local oscillator cable, shared by the channels in a DCM1 module; imperfect isolation of the mixer would cause a weak common-mode resonance to manifest.

An important detail in this analysis is that, because of the normalization and $1/f$ filter in the TOD pipeline, any standing wave signal with a constant resonant cavity wavelength over time will be filtered away. For a standing wave to survive the filtering, it must have a changing wavelength. Prime suspects for the origin of this effect are therefore optical cavities that could expand or contract in size, or cables that could be stretched.

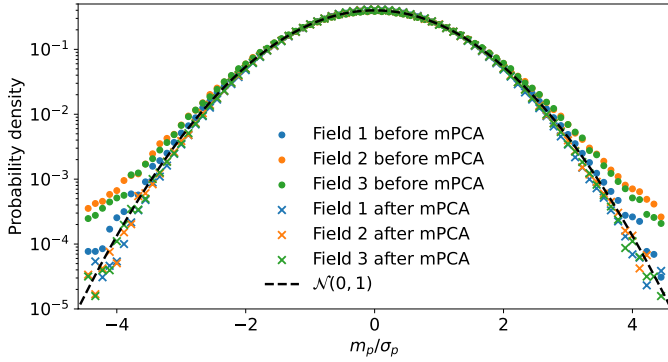


Fig. 12. Histogram of all map pixel temperature values across all feeds and frequencies, divided by their white noise uncertainty. The three fields are shown separately, before and after application of the map PCA filter. A normal distribution is shown in black; a completely white noise map will trace this distribution. All three fields show excess high-significance pixels before the map PCA. After the filter, all three fields fall slightly below the normal distribution on the wings, because of the slight over-subtraction of noise at various stages in the pipeline.

4.3.3. Effects of map-domain filtering

The map-domain PCA filtering was implemented in an attempt to mitigate these systematic effects and it has proved to be effective at this task. The first PCA component alone subtracts both effects to a level where they are not visible in the maps by eye. This shows that both effects are well modeled as the outer product of a pixel vector and a frequency vector.

Visually inspecting the PCA components, we usually see some structure for the first 3–5 modes. Figure 11 shows an example of the five leading PCA components and their amplitudes. Here we can see that the first component has very clear structure in both the map and frequency domain. The remaining modes seem to absorb some residuals after this first mode, especially on specific channels close to the edges of each of the two Bands that divide the frequency range in two.

We have chosen to remove just five out of 256 PCA components in the map-PCA filter, as no structure was visible by eye in the worst-affected cases after this number of components was removed, and the removal of more components did not significantly affect the results of any subsequent analysis (such as the power spectrum). With only five components being removed, we also limit the potential for CO signal loss. We could have employed a similar approach to the TOD PCA, with a dynamic amount of components, but the noise properties of the maps are more complicated than the TODs, and we have chosen to keep a static number of components, postponing more fine-tuning to future analysis. The filter is applied to the individual feed maps, and to individual splits – both the elevation split used for the cross power spectrum, and the individual map splits for the null tests.

With the application of the map-PCA filter, we observe that the start-of-scan and turn-around effects are suppressed well below the white noise of the maps, as can be seen in Fig. 10. We have also designed a null test to specifically target the turn-around systematic effect, by splitting the maps at the TOD level into east- and west-moving azimuthal directions. The turn-around effect manifests very differently in each half of this split, making it the basis for a sensitive null test. We find that the Season 2 data passes this test after the map-PCA filter has been applied (Stutzer et al. 2024).

The standard deviation of the maps only falls by 2% after applying the filter, as the noise still dominates the overall amplitude. However, smoothing the maps slightly will enhance large-scale correlations, while suppressing uncorrelated noise. Smoothing both the filtered and unfiltered 3D maps using a Gaussian with $\sigma = 3$ voxels, the standard deviation is 67% lower in the map-PCA filtered map. We can similarly observe that the average correlation between neighboring pixels (of the unsmoothed map), a good indication of the level of larger scale structure, falls from 6.3% to -0.4% after applying the map-PCA.

The magnitude of these systematic effects is different between feeds and frequencies, as seen in Figure 10. Performing a χ^2 white noise consistency test on the individual frequency channel maps of each feed, we find that for the worst feeds, namely those associated with DCM1-2 (feeds 6, 14, 15, 16 and 17), around 50% of their channels fail this test at $>5\sigma$. The best-behaving feeds are 4, 5, 10, and 12, all with fewer than 5% of channels failing at $>3\sigma$. However, we want to emphasize both that no feed is completely without these effects before the map-PCA, and that after the map-PCA, there is no longer a quantitative difference between the “good” and “bad” feeds, with all feeds passing χ^2 -tests at expected levels.

The PCA filtering (both in the time and map domain) constitutes the only non-linear processing in the pipeline. Non-linear filtering makes it more difficult to estimate the resulting signal bias and transfer function. In Sect. 6.4 we demonstrate that a PCA filter applied on a noisy matrix behaves linearly with respect to a very weak signal, and we find that any CO signal in the data is well within this safe regime.

4.4. Final maps

Figure 12 shows the distribution of map voxel values for all three fields, in units of significance, before and after the map-PCA. Before the map-PCA, the distribution shows a clear excess on the tails, while the cleaned maps are very consistent with white noise. This is expected and desired, as the CO signal is so weak that individual frequency maps are still very much dominated by the system temperature. The noise level in the maps is, actually, about 2.5% lower than expected from the white noise uncertainty, due to the filtering in the pipeline. This effect can be seen in Fig. 12, with the histograms falling slightly below the normal distribution.

Figure 13 shows the distribution of voxel uncertainties over the three fields for this work and our ES maps. Each voxel has an approximate size of 2×2 arcmin, that, together with the frequency direction, corresponds to a comoving cosmological volume of $\sim 3.7 \times 3.7 \times 4.1 \text{ Mpc}^3$. For Fields 2 and 3, the high sensitivity $<50 \mu\text{K}$ region corresponds to a comoving cosmological cube of around $150 \times 150 \times 1000 \text{ Mpc}^3$ per field. Combining all three fields, Season 2 has one million voxels with an uncertainty $<50 \mu\text{K}$, compared to one million voxels below $<125 \mu\text{K}$ for Season 1. The footprint of the final maps have increased slightly in size because of the wider scan pattern of Season 2b.

The sensitivity increase per field over Season 1 is 2.0, 2.6, and 2.7, for Fields 1, 2, and 3, respectively. Fields 2 and 3 are now the highest sensitivity fields, while Field 1 is noticeably worse, from larger losses to data selection, especially Moon and Sun sidelobe pickup. The uncertainties are estimated from Eq. (13), and correspond well to the noise level observed in the map, as we saw from Fig. 12. A figure showing the uncertainties across the fields on the sky can be found in Appendix E, together with a subset of the final maps.

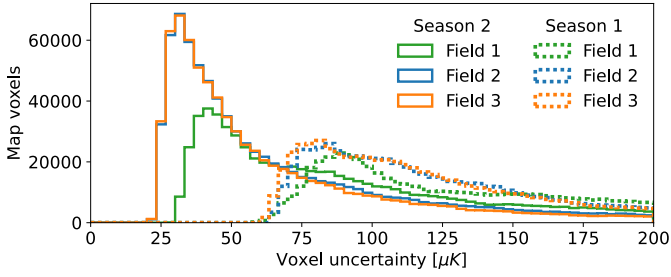


Fig. 13. Histograms comparing the map voxel uncertainties of this work (Season 2), and the ES publications (Season 1). The voxel values are for feed-coadded maps.

5. Data selection

In addition to a three-fold increase in observational hours, the second season also features a similar increase in data retention compared to the ES results. Table 2 compares the data loss in the ES results and this work. The table is split into three parts, namely 1) observational losses, 2) time- and map-domain losses, and 3) power spectrum domain losses. The Season 2 column only relates to data taken during Season 2, but we have also reprocessed Season 1 data with the new pipeline for the final results.

5.1. Observational data retention

The first three rows of Table 2 are observational inefficiencies that have been corrected since the first season. E_{scan} constitutes the fraction of scans that were performed in constant elevation mode, as opposed to Lissajous scans that were cut due to large systematic effects (see Sect. 2.4). E_{feed} is the fraction of functioning feeds, and E_{el} is the fraction of data taken at elevation 35° – 65° (see Sect. 2.3). Since Season 1, we no longer observe in Lissajous mode, all feeds are functional and the observing strategy has been optimized to maximize E_{el} . As a result, the total data retention from these three cuts, which was 32% in the first season, is now 100%.

5.2. Time and map domain data selection

In the next section in Table 2, E_{freq} refers to the frequency channel masking performed in 12gen, as discussed in Sect. 3.7. The masking algorithm itself is virtually identical to in ES, with a few changes. The shifting of aliased power into channels outside the nominal frequency range (Sect. 2.5) means that, from Season 2b onward, we recover the 8% of channels masked in Season 1 and 2a. The inclusion of the new per-feed PCA filter in 12gen results in slightly fewer data being masked by data-driven tests. However, we have also increased the number of manually flagged channels that seem to be performing sub-optimally, leaving us with a E_{freq} data retention only slightly higher than for ES.

Next, E_{stats} constitutes the cuts performed in the `accept_mod` script, that discards scans based on different housekeeping data and summary statistics of the scans. There are over 50 such cuts in total, most of them removing a small number of outlier scans. Upon the completion of the Season 2 null test framework (Stutzer et al. 2024), null tests failed on five scan-level parameters. Cuts on these parameters were tightened or implemented in `accept_mod`, and the null tests now pass. The five new or tightened cuts are: 1) any rain during the scan; 2) wind speeds above 9 m/s; 3) high average amplitude of the

Table 2. Data retention overview.

	Season 1	Season 2	Explanation
E_{scan}	50.0%	100.0%	Retained scans (CESs)
E_{feed}	84.2%	100.0%	Functional feeds
E_{el}	75.6%	100.0%	Inside good elevation range
E_{obs}	31.8%	100.0%	Observational data retention
E_{freq}	72.8%	74.3%	Frequency masking in 12gen.
E_{stats}	57.4%	33.6%	Cuts on accept-mod statistics
$E_{\chi^2_{\text{P}(k)}}$	72.2%	100.0%	Per-scan auto-PS χ^2 -test
E_{cuts}	30.1%	24.9%	Map-level data retention
$E_{\chi^2_{\text{C}(k)}}$	52.4%	100.0%	Cross-spectrum χ^2 -test
$E_{\text{C}(k)}$	94.7%	75.0%	Cross-spectrum auto combinations
E_{PS}	49.6%	75.0%	Retained data at PS-level
S_{tot}	6.8%	21.6%	Final PS-domain sensitivity, calculated as $S_{\text{tot}} = \sqrt{E_{\text{obs}}^2 E_{\text{cuts}}^2 E_{\text{PS}}}$

Notes. Surviving fraction of data for different filtering steps of the pipeline. The left column shows the values used for the ES analysis, and the right column shows this work. The first 3 rows show individual data losses to observational constraints, that are combined in the gray row below. The three next rows show data retention to time and map domain cuts, again combined below. Finally, the next two rows show the losses in the power spectrum domain, also combined. The last row, S_{tot} shows the final fraction of theoretical power spectrum sensitivity from the combined data retention (see the text for details). The losses are multiplicative, such that multiplying E for all the individual losses gives the retained data fractions shown in gray.

fitted TOD PCA components; and 4–5) outliers in the f_{knee} of the 0th and 1st order $1/f$ filter components⁸. Additionally, all other `accept_mod` cuts from ES are continued, and the surviving data fraction has therefore fallen noticeably, from 57.4% to 33.6%. No attempt has yet been made to tune these cuts, presenting us with future potential for increased data retention.

Finally, $E_{\chi^2_{\text{P}(k)}}$ is the last scan-level cut. Each scan is binned to a very low-resolution 3D map, and a series of χ^2 -tests are performed on different 2D and 3D auto power spectra calculated from these maps. In the Season 2 pipeline, this cut is removed entirely, for two reasons. Firstly, we found little evidence that it helped us pass null tests or remove dangerous systematic errors from the final data. Secondly, we found it difficult to calculate robust pipeline transfer functions for each individual power spectrum, as individual scans might vary a lot in sky footprint and pointing pattern. We therefore saw little reason to keep this cut in the pipeline.

5.3. Power spectrum level data selection

The last section of Table 2 shows the fraction of data retained after cuts in the power spectrum domain; details on the power spectrum methodology are described by Stutzer et al. (2024). In summary, we calculate pseudo cross spectra between different groups of feeds and across pointing elevations and then average these spectra to get the CO power spectrum. However, some of the cross-spectra are discarded before averaging, and this is the loss discussed in this section. This loss in the power spectrum

⁸ The $1/f$ filter fits the time-dependent components c_t^0 and c_t^1 , primarily picking up correlated noise and changes in the atmosphere. We perform a $1/f$ fit to the components as functions of time, and cut when the f_{knee} falls outside the typical range of values

domain has to be tracked separately from data loss in the map and TOD domain, as the losses cannot naively be added together. Since map values are squared when calculating the power spectrum, so is the map-domain data volume when calculating the power spectrum sensitivity. The total power spectrum sensitivity is therefore calculated as $S_{\text{tot}} = \sqrt{E_{\text{obs}}^2 E_{\text{cuts}}^2 E_{\text{PS}}}$.

In this table section, $E_{\chi_{C(k)}^2}$ constitutes a χ^2 -test on the individual feed-feed cross-spectra and cuts away any cross-spectrum with an average significance above 5σ . In ES, we lost around half the data to this cut. The cut has now been entirely removed, for several reasons. First of all, we now have a much more rigorous null test framework, and find that we pass all null tests without these cuts. Secondly, we strongly prefer moving all data-inferred cuts to a point as early in the pipeline as possible, to reduce any potential biasing effects. It is therefore a considerable pipeline improvement over ES that we now perform no data-inferred cuts in the power spectrum domain.

Finally, $E_{C(k)}$ is the fraction of the cross power spectra that are not auto-combinations between the same feeds. In ES we performed cross-spectra between all 19 feeds, which resulted in a loss of 19 out of 19×19 cross-spectra, or 5.3%. We now calculate cross-spectra between four groups of feeds, for better mitigation of systematic effects and improved overlap, resulting in a loss of 4 out of 4×4 feeds, or 25%. This is a theoretical approximation of the sensitivity, as the varying degrees of overlap between different feeds will interplay with the sensitivity.

5.4. Future prospects for data selection

Combining the retained map-level data with the retained PS-level data we keep 21.6% of the theoretical sensitivity, compared to 6.8% for ES, a more than three-fold increase. Most of this increase comes from much higher observational data retention E_{obs} , and the removal of the $\chi_{P(k)}^2$ and $\chi_{C(k)}^2$ cuts. We now have no data-driven cuts in the power spectrum domain, where the signal is the strongest, leaving us less susceptible to signal bias.

In order to pass null tests and allow for the removal of other cuts, the data retention after cuts on `accept_mod` statistics, E_{stats} , has decreased quite substantially. We erred on the side of caution when introducing the new cuts, and once the data passed all the null tests we made no attempt at reclaiming any data from these cuts. In future analysis, we are therefore confident that better tuning of these parameters, assisted by an even better understanding and filtering of systematic effects, will allow us to substantially increase the amount of data retained at this step.

The numbers in Table 2 are averages across fields, feeds, and scans, and the combined data retentions, E_{map} , E_{PS} and S_{tot} are for simplicity calculated by naively multiplying together the individual retentions. This ignores certain complications, such as correlations between the cuts, and the actual sensitivity might therefore differ slightly. It should also be noted that the right column constitutes the efficiency of Season 2 data in combination with the Season 2 pipeline, and re-analysis of Season 1 does not reach a S_{tot} of 21.6%, as the losses to E_{obs} will still apply even with the improvements to the pipeline.

6. Pipeline signal impact and updated transfer functions

The final maps are biased measurements of the CO signal, due to signal loss incurred in observation and data processing, leading to a biased power spectrum. This effect can be reversed by esti-

imating a so-called transfer function $T(k_{\parallel}, k_{\perp})$, that quantifies this signal loss at different scales. We separate the angular modes $k_{\perp}$ and the frequency (redshift) modes $k_{\parallel}$, as the impact on the CO signal is usually very different in these two dimensions. In this section, we present updated versions of the three relevant transfer functions:

- The pipeline transfer function $T_p(k_{\parallel}, k_{\perp})$: The time- and map-domain processing will inevitably remove some CO signal.
- The beam transfer function $T_b(k_{\perp})$: The size and shape of the beam will suppress signal on smaller scales in the angular dimensions.
- The voxel window transfer function $T_v(k_{\parallel}, k_{\perp})$: The finite resolution of the voxels suppresses signal on both angular and redshift scales close to the size of the voxels.

6.1. Updated beam and voxel window transfer functions

In the ES analysis, the beam and voxel window transfer functions were estimated using simulations (Ihle et al. 2022). However, because the voxel grid and beam of the COMAP instrument are well understood we can also compute $T_b(k_{\perp})$ and $T_v(k_{\parallel}, k_{\perp})$ analytically. As the COMAP mapmaker simply uses nearest neighbor binning of the TOD into equispaced voxels the map is smoothed by a $\text{sinc}^2(x)$ function along each map axis. Specifically, the voxel window can be expressed as⁹

$$T_v(k_{\parallel}, k_{\perp}) = T_{\text{freq}}(k_{\parallel})T_{\text{pix}}(k_{\perp}) = \text{sinc}^2\left(\frac{\Delta x_{\parallel} k_{\parallel}}{2\pi}\right) \text{sinc}^2\left(\frac{\Delta x_{\perp} k_{\perp}}{2\pi}\right), \quad (17)$$

where $\Delta x_{\perp}$ and $\Delta x_{\parallel}$ are the voxel sizes in angular and frequency directions. Specifically, we have voxel resolutions of $\Delta x_{\perp} \approx 3.7$ Mpc and $\Delta x_{\parallel} \approx 4.1$ Mpc. We note that since the angular pixel window is approximately radially symmetric we have approximated $T_{\perp}(k_{\perp}) \approx T_{\text{RA}}(k_{\text{RA}}) \approx T_{\text{Dec}}(k_{\text{Dec}})$. Both the perpendicular and parallel voxel transfer functions can be seen in Figs. 14 and 15.

In principle, we could reduce the voxel window signal impact on smaller scales in both the angular and frequency dimensions by binning the maps into higher resolution voxels, shifting the decline of $T_{\text{freq}}(k_{\parallel})$ and $T_{\text{pix}}(k_{\perp})$ to higher k -values. In practice, however, the angular voxel window applies at a scale where the beam transfer function already suppresses the signal beyond recovery. Similarly, line broadening is expected to heavily attenuate the CO signal above ~ 1 Mpc⁻¹ (Chung et al. 2021), although the exact extent of line-broadening depends on galaxy properties that are not yet well constrained. Additionally, it would be more computationally costly to perform the analysis in higher resolution.

Next, given the radial beam profile $B(r)$ (see Fig. 2 of Ihle et al. 2022) and the convolution theorem we can obtain the beam transfer function as

$$T_b(k_{\perp}) = |\mathcal{F}\{B(r)\}|^2, \quad (18)$$

where r is the radius from the beam center, and $\mathcal{F}$ is the (2D) Fourier transform. As we assume the telescope beam to be radially symmetric the resulting beam smoothing transfer function will be a function of just $k_{\perp} = \sqrt{k_{\text{RA}}^2 + k_{\text{Dec}}^2}$ giving $T_b(k_{\perp})$. The main-beam efficiency is taken into account in the same manner as Ihle et al. (2022) prior to computing the Fourier transform of the beam. As we can see in Figs. 14 and 15 the beam is by far the

⁹ We note that we use the convention where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

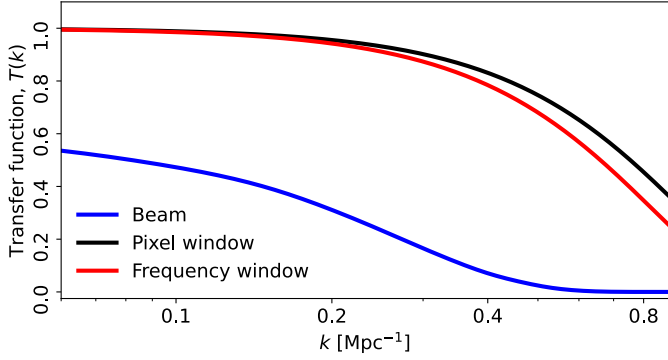


Fig. 14. Effective one-dimensional transfer functions for the instrumental beam (blue curve), pixel window (black curve), and frequency window (red curve) resulting from spherical averaging over the corresponding two-dimensional transfer functions shown in Fig. 15.

most dominant effect limiting our ability to recover the signal at smaller scales.

6.2. The signal injection pipeline

The last transfer function is that of the pipeline, that will inevitably remove some CO signal from the data. We estimate this impact by a signal injection pipeline, similar to what was done in ES (Foss et al. 2022).

We inject a simulated CO signal into the real Level 1 data before any filtering. The data are then propagated through the entire pipeline as usual, and the resulting mock observations are then compared to the known, unfiltered input signal to estimate the pipeline transfer function. We chose to inject the simulations into the actual data, instead of simulating the entire observation, in order to mimic the real systematic error and noise properties as closely as possible.

For the simulations we use approximate cosmological dark-matter-only simulations using the peak-patch method of Bond & Myers (1996) with updates by Stein et al. (2019). These are subsequently populated with CO emission using the COMAP fiducial model derived in Chung et al. (2022) (‘UM+COLDz+COPSS’), that describes CO luminosities as a function of dark matter halo masses, $L_{\text{CO}}(M_{\text{halo}})$. These simulated mock maps are then boosted by a factor of 20, to recover a less noisy transfer function. This is counter-weighted by splitting the full dataset in 10 subsets, and passing each of these separately through the pipeline. We denote the maps as $s_{\nu,\theta}^{\text{mock}}$, where ν and θ are the frequency (redshift) and angular (pixel) dimensions, respectively. The TOD pipeline is quantitatively unaffected by the injection of this weak CO signal – even after a factor 20 boost, the brightest CO pixel in the simulation is still more than four orders of magnitude below the system temperature. In the map-domain, the S/N is much higher, and the implications of injecting a boosted signal are discussed in Sect. 6.4, where we conclude that the map-PCA filter also behaves predictably for our chosen boost strength.

Using the real telescope pointing $\mathbf{P}$, and estimated gain $\mathbf{G}$ and beam $\mathbf{B}$, we get the signal-injected Level 1 data as

$$d_{t,\nu}^{\text{mock}} = \mathbf{GPB}s_{\theta,\nu}^{\text{mock}} + n_{t,\nu}, \quad (19)$$

where $n_{\nu,t}$ represents the actual Level 1 data, that acts as the noise term with respect to the injected mock CO signal. In order to mimic the observed CO signal as closely as possible, we also beam-smooth the maps used for the signal injection. The mock

data $d_{\nu,t}^{\text{mock}}$ are then filtered by the pipeline to produce a mock map:

$$\begin{aligned} m_{\theta,\nu}^{\text{mock}} &= f_{\text{map}} \left[M \left(f_{\text{TOD}} \left[d_{\nu,t}^{\text{mock}} \right] \right) \right] \\ &= f_{\text{map}} \left[M \left(f_{\text{TOD}} \left[\mathbf{GPB}s_{\nu,t}^{\text{mock}} + n_{\nu,t} \right] \right) \right], \end{aligned} \quad (20)$$

where we let f_{TOD} represent all time-domain filtering, M represents a noise-weighted binned map-maker, as described in Sect. 4.1, and f_{map} represents the map-domain PCA filter. Examples of resulting maps are shown in Appendix D.

To make sure that the reference CO simulation $s_{\nu,\theta}^{\text{mock}}$ is directly comparable to the filtered maps, we also perform some of the same treatment on it: we beam-smooth it, read it into a TOD with the real telescope pointing, and bin the TOD back into a map with the same resolution as the real maps. The difference, however, is that this is done completely without noise, and we do not apply any of the filters. We can write this as

$$\hat{s}_{\nu,\theta}^{\text{mock}} = M \left(\mathbf{GPB}s_{\nu,\theta}^{\text{mock}} \right). \quad (21)$$

Doing it this way means that we isolate the filter transfer function, and the effect of the beam and pixelation are not included. This is intentional, as we already estimated these impacts analytically in Sect. 6.1.

From these maps, we can now write the filter transfer function as

$$T = \frac{C(\mathbf{m}^{\text{mock}}, \hat{s}^{\text{mock}})}{P(\hat{s}^{\text{mock}})}, \quad (22)$$

where the cross-spectrum, C in the numerator between the filtered mock data, $\mathbf{m}^{\text{mock}}$, and the unfiltered mock signal, $\hat{s}^{\text{mock}}$, picks up all common signal modes after filtering while canceling residual systematic effects and noise in the mock data. The cross-spectrum is divided by the unfiltered signal auto spectrum, $P_k(\hat{s}^{\text{mock}})$, to obtain a filter transfer function T .

Equation (22) represents a more robust estimator of the transfer function than the one employed in ES (Foss et al. 2022 Eq. (34)), both because it does not require an accurate estimation of the noise power spectrum, and because using the cross-spectrum estimator as opposed to the auto-spectrum estimator makes it less susceptible to picking up signal in the data that does not originate from the injected CO (e.g. from systematic effects).

For the signal injection, we use all scans from Seasons 1 and 2a for Field 2. Preliminary analysis of the transfer functions of Fields 1 and 3, and the slower pointing scans of Season 2b, show that they are very similar, especially in the k -regime included in this work. As mentioned in the beginning of the section, we divide the scans into 10 random and equally large parts. Different dark matter simulations are injected into each part. This both reduces the impact of sample variance in the simulations and allows us to boost the signal a bit more without having to worry about PCA non-linearity (see the next section). We then average over the 10 resulting transfer functions, to get the final transfer function estimate.

6.3. Updated filter transfer function

The left-most panel of Fig. 15 shows the full COMAP pipeline transfer function, as described in the previous section, in parallel and perpendicular directions (i.e., redshift and angular scale, respectively). The Season 2 publications (Stutzer et al. 2024; Chung et al. 2024) exclude some of the larger angular scales

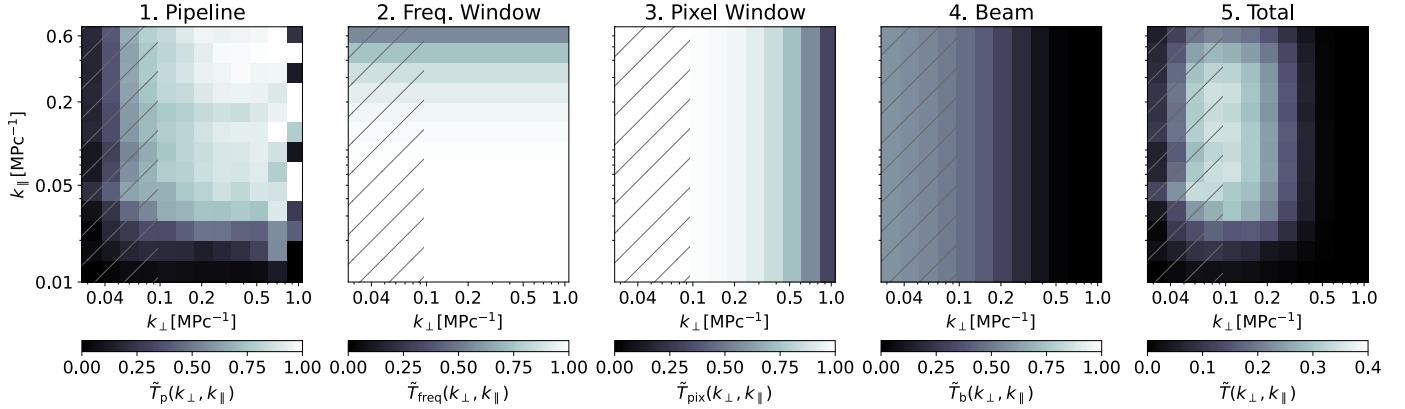


Fig. 15. Effective transfer functions used in the COMAP pipeline. From left to right, the five panels show 1) the filter transfer function, $T_f(\mathbf{k})$, quantifying signal loss due to pipeline filters; 2) the pixel window transfer function $T_{\text{pix}}(\mathbf{k})$ resulting from binning the TOD into a pixel grid; 3) the frequency window transfer function, $T_{\text{freq}}(\mathbf{k})$, resulting from data down-sampling in frequency; 4) the beam smoothing transfer function, $T_b(\mathbf{k})$; and 5) the full combined transfer function, $T(\mathbf{k})$, corresponding to the product of the four individual transfer functions. The striped region to the left is not used for our final analysis but is shown for completeness. We note that the leftmost panel has a colorbar that saturates at 0.4, unlike the other four.

accessible in the maps due to concerns about mode mixing and unconstrained modes due to poor overlap. For reference, the cut-off value for $k_{\perp}$ at 0.93 Mpc^{-1} corresponds to angular scales of 36.4 arcmin.

Figure 16 shows the individual contributions of each filter to the full transfer function. We are somewhat limited on large angular scales by the normalization and pointing filters, and very limited on large redshift scales by the $1/f$ filter. The right-most column is noisy because the beam suppresses most of the signal at these small scales.

We also note that a small issue was discovered with the transfer function analysis published in [Ihle et al. \(2022\)](#), relating to how the mock signal was interpolated when injected into the TOD. The effect was that the transfer functions from ES was slightly underestimated, and the transfer function from [Ihle et al. \(2022\)](#) peaked at around 0.8. This issue has now been solved, and the new transfer function peaks correctly at almost 1.0.

6.4. Linearity of PCA filtering

All filters except the various PCAs constitute linear operations on the data. Linearity makes transfer function estimation much simpler, as neither the choice of CO signal model nor its level with respect to the noise impacts the resulting transfer function. For the PCA filters, both these factors could in principle impact the shape of the transfer function.

To quantify the sensitivity of the transfer function to such factors, we constructed a simplified version of the signal-injection pipeline. In order to be able to run many simulations, we made the following alterations to the pipeline:

- We bypass the time domain, and perform the signal injection in the map domain. This is the domain where the CO S/N is the strongest, and the map-domain processing is much more computationally efficient than that for the time-domain. The map mocks are the same as in Sect. 6.2, with some boost factor b .
- Instead of the real map, we use white noise simulations, drawn from the white noise uncertainty of Field 2.
- The resulting transfer function is calculated using Eq. (22), and then averaged across the simulations and feeds.

This process is repeated for 40 000 noise realizations for each of the 19 feeds, with boosts between 0.3 and 300 relative to the fiducial CO model of [Chung et al. \(2022\)](#). The variation of the resulting average transfer functions with boost strength can be seen in Fig. 17. The figure shows two distinct regimes. In the low S/N regime to around $S/N=0.02$ (boost 10), the PCA filter behaves linearly with respect to the CO signal. This is demonstrated by the independence of the transfer function to the S/N in this regime. Additionally, all the k -points lie on top of each other around a value of 0.96, meaning that all scales are suppressed at the same level because the PCA is simply fitting and subtracting random white noise. In the second regime, at high S/N, the transfer function is strongly scale-dependent but flattens out as the signal dominates the noise.

In conclusion, for a noisy matrix with an accompanying signal, a PCA filter behaves linearly with respect to the signal for a sufficiently weak signal. For us this means that when estimating the transfer functions, we need to use a sufficiently low boost value to avoid biasing our estimate of the transfer function. The individual data chunks used to estimate the transfer function are well within this linear regime, at an S/N of 0.004. The actual CO signal is of course of unknown amplitude, but assuming the fiducial model of [Chung et al. \(2022\)](#) results in an S/N of 0.002. If the CO signal were even close to the unsafe regime of $S/N > 0.02$, we would already have made a strong detection of it, as this would correspond to a 100 times brighter power spectrum than the fiducial model. We also note that in the future, as the experiment's sensitivity increases, we can simply perform the map PCA on sub-divisions of the data to keep the S/N low, as we already do on Season 1 + 2a and Season 2b, due to their differing pointing strategy.

This analysis was performed on the map-domain PCA filter, as the map domain is where the CO S/N is the strongest. Equivalent analysis has been performed for the TOD PCA filter, but the CO signal is so weak at the per-scan level that a boost factor of 2000 or greater is required to make it behave non-linearly with respect to the CO signal. The final transfer function is currently estimated jointly for all filters, but in future work we intend to estimate the transfer function for the mPCA separately from the other filters, gaining higher sensitivity on the linear parts of the pipeline that do not require as low a boost.

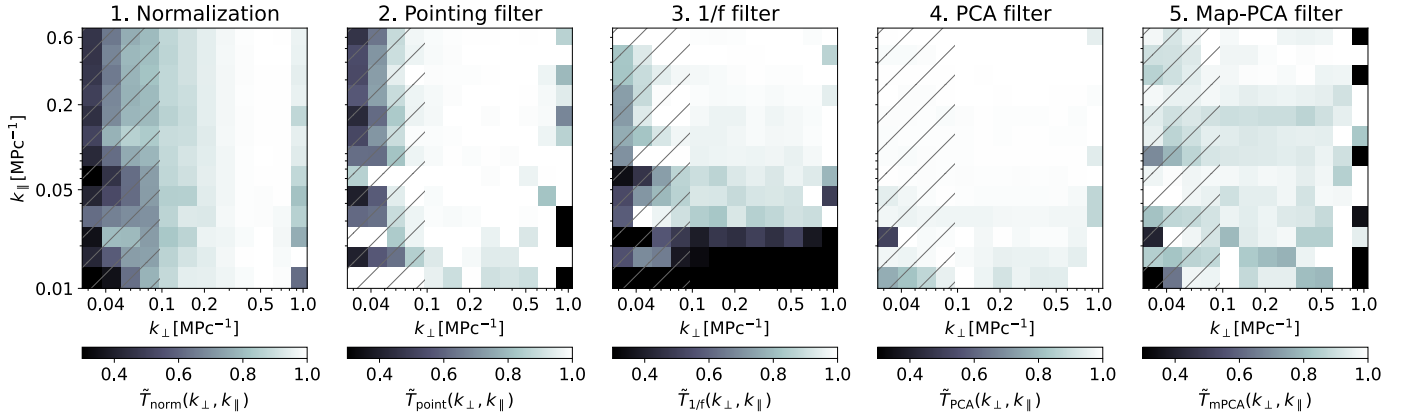


Fig. 16. Transfer functions for each of the five individual filters used in the pipeline. The normalization and pointing filters suppress large angular scales, while the $1/f$ filter almost entirely eliminates parallel modes larger than $k = 0.02 \text{ Mpc}^{-1}$. The TOD PCA filter has almost no impact on the signal, primarily because so few modes are subtracted. The map PCA has a more noticeable signal loss, but it remains relatively scale-independent because the signal is still weak enough (see Sect. 6.4). The total pipeline transfer function shown in the leftmost panel of Fig. 15 is the product of these. The striped region to the left is not used in Season 2 results.

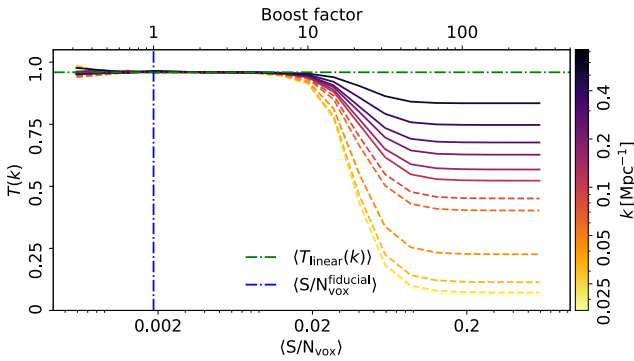


Fig. 17. Map PCA transfer function $T(k)$ as a function of the voxel S/N, with the different k -scales shown as differently colored lines. The equivalent boost used to the fiducial CO model is shown as the top x -axis. Modes with k -values that fall within the analysis bounds are shown as solid, while modes outside the scope of COMAP Season 2 are shown as striped. The horizontal green line shows the average transfer function in the low-S/N regime, which is 0.96. The vertical blue line shows the S/N that the maps would have if the fiducial model of Chung et al. (2022) perfectly described the true CO signal.

7. Summary and conclusions

We have presented the improvements in data analysis, filtering, and data selection that have enabled us to increase the power spectrum sensitivity to 21.6% of the theoretical maximum, up from 6.8% in the ES publications. Combined with an increased integration time, the two most sensitive Season 2 fields both have a voxel uncertainty of $<50 \mu\text{K}$ across $\sim 1.5^\circ \times 1.5^\circ$ patches on the sky. Across the three fields, this corresponds to a 2.5 times decrease in the total map uncertainty.

The largest increase in data retention comes from improvements in observational strategy. We now solely observe using CES, whereas in Season 1 50% of observations used Lissajous scans that proved prone to systematic effects and were not included in our ES analysis. Additionally, we now observe within elevation boundaries of 35° – 65° , a region with minimal gradients in ground sidelobe pickup. In Season 1, 25% of scans fell outside this range and were discarded.

Additional increases in data retention have come from the removal of data cuts. In Season 1, power spectrum χ^2 tests

were performed both on individual scan maps and on individual feed-feed cross-spectra before averaging them. These cuts removed (respectively) 28% of scans and 48% of cross-spectra, and are now no longer applied. This also reduces the possibility of signal bias owing to data-inferred cuts late in the pipeline.

In order to pass null tests and allow for the removal of other cuts, scan-level data cuts were implemented on six new house-keeping statistics, increasing the data lost to such cuts from 43% to 66%. However, no attempt was made to reclaim data after the null tests had passed, and the necessity of the cuts carried over from the ES pipeline is largely untested. We are confident this number can be greatly reduced in future work.

The removal of data cuts was also made possible by better mitigation of systematic errors. The time-domain pipeline has numerous smaller improvements, such as a new per-feed PCA filter, dynamically determined number of PCA components, better masking of the T_{sys} spikes, and more manual masking of consistently problematic channels. Most impactful, however, was the introduction of a map-level PCA filter, that proved essential to dealing with a couple of pointing-correlated systematic errors that emerged due to increased sensitivity. We show that the map PCA filter suppresses these effects to below the noise level, and decreases the standard deviation of slightly smoothed maps by 67%, to a level consistent with the expected white noise.

Although the PCA filters constitute non-linear filtering, we have shown that the PCA filters behave linearly with respect to any sufficiently weak signal. We find that the expected CO signal falls well within this regime, substantially simplifying transfer function estimation. We repeat the signal-injection pipeline transfer function estimation of the ES publications and ensure that the injected signal is also weak enough to maintain the PCA filters in their linear range. We have also replaced the simulation-based estimates of the beam and voxel windows transfer functions with more robust analytic expressions, improving their reliability at small scales.

COMAP has thus greatly increased its integration speed both through observational improvements, better processing, and reduced data cuts. Our final sensitivity retention of 21.6% of the theoretical maximum still leaves significant room for improvement, and we aim to increase this further as the Pathfinder continues to observe.

Acknowledgements. We acknowledge support from the Research Council of Norway through grants 251328 and 274990, and from the European Research Council (ERC) under the Horizon 2020 Research and Innovation Program (Grant agreement No. 772253 bits2cosmology and 819478 Cosmoglobe) This material is based upon work supported by the National Science Foundation under Grant Nos. 1517108, 1517288, 1517598, 1518282, 1910999, and 2206834, as well as by the Keck Institute for Space Studies under “The First Billion Years: A Technical Development Program for Spectral Line Observations”. Parts of the work were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. HP acknowledges support from the Swiss National Science Foundation via Ambizione Grant PZ00P2_179934. SEH and CD acknowledge funding from an STFC Consolidated Grant (ST/P000649/1) and a UKSA grant (ST/Y005945/1) funding LiteBIRD foreground activities. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00340759). PCB was supported by the James Arthur Postdoctoral Fellowship. DTC was supported by a CITA/Dunlap Institute postdoctoral fellowship for much of this work. The Dunlap Institute is funded through an endowment established by the David Dunlap family and the University of Toronto. Research in Canada is supported by NSERC and CIFAR. JGSL and NOS thank Sigurd K. Næss for all the fruitful discussions in the office, and while biking through nature, during the last three years. This work was first presented at the Line Intensity Mapping 2024 conference held in Urbana, Illinois; we thank Joaquin Vieira and the other organizers for their hospitality and the participants for useful discussions. JGSL and NOS thanks the 2023 Stockholm Beam Mode workshop for discussions on beam convolution. *Software acknowledgments.* Matplotlib for plotting (Hunter 2007); NumPy (Harris et al. 2020) and SciPy (Virtanen et al. 2020) for efficient numerics and array handling in Python; Astropy a community made core Python package for astronomy (Astropy Collaboration 2013, 2018, 2022); Multi-node parallelization with MPI for Python (Dalcín et al. 2005, 2008; Dalcín et al. 2011; Dalcín & Fang 2021); Pixell (<https://github.com/simonsobs/pixell>) for handling sky maps in rectangular pixelization.

References

- Ade, P., Aravena, M., Barria, E., et al. 2020, *A&A*, **642**, A60
- Akeson, R., Armus, L., Bachelet, E., et al. 2019, arXiv e-prints [arXiv:1902.05569]
- Anderson, C. J., Luciw, N. J., Li, Y. C., et al. 2018, *MNRAS*, **476**, 3382
- Astropy Collaboration (Robitaille, T. P., et al.) 2013, *A&A*, **558**, A33
- Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, *AJ*, **156**, 123
- Astropy Collaboration (Price-Whelan, A. M., et al.) 2022, *ApJ*, **935**, 167
- Bandura, K., Addison, G. E., Amiri, M., et al. 2014, *SPIE*, **9145**
- Bernal, J. L., & Kovetz, E. D. 2022, *A&A Rv*, **30**, 5
- Bond, J. R., & Myers, S. T. 1996, *ApJS*, **103**, 1
- Breysse, P. C., Kovetz, E. D., & Kamionkowski, M. 2014, *MNRAS*, **443**, 3506
- Breysse, P. C., Chung, D. T., Cleary, K. A., et al. 2022, *ApJ*, **933**, 188
- CCAT-Prime Collaboration (Aravena, M., et al.) 2022, *ApJS*, **264**, 7
- Chang, T.-C., Pen, U.-L., Bandura, K., & Peterson, J. B. 2010, *Nature*, **466**, 463
- Chung, D. T., Breysse, P. C., Ihle, H. T., et al. 2021, *ApJ*, **923**, 188
- Chung, D. T., Breysse, P. C., Cleary, K. A., et al. 2022, *ApJ*, **933**, 186
- Chung, D. T., Breysse, P. C., Cleary, K. A., et al. 2024, *A&A*, **691**, A337
- Cleary, K. A., Borowska, J., Breysse, P. C., et al. 2022, *ApJ*, **933**, 182
- Crites, A., Bock, J., Bradford, C., et al. 2014, *SPIE*, **9153**, 613
- Dalcín, L., & Fang, Y.-L. L. 2021, *Comput. Sci. Eng.*, **23**, 47
- Dalcín, L. D., Paz, R. R., Kler, P. A., & Cosimo, A. 2011, *Adv. Water Resources*, **34**, 1124
- Dalcín, L., Paz, R., & Storti, M. 2005, *J. Parallel Distrib. Comput.*, **65**, 1108
- Dalcín, L., Paz, R., Storti, M., & D’Elia, J. 2008, *J. Parallel Distrib. Comput.*, **68**, 655
- Dunne, D. A., Cleary, K. A., Breysse, P. C., et al. 2024, *ApJ*, **965**, 7
- Foss, M. K., Ihle, H. T., Borowska, J., et al. 2022, *ApJ*, **933**, 184
- Gabriel, K. R., & Zamir, S. 1979, *Technometrics*, **21**, 489
- Gailly, J. L., & Adler, M. 2023, <https://www.gnu.org/software/gzip/>
- Geman, S. 1980, *Ann. Probab.*, **8**, 252
- Halko, N., Martinsson, P. G., & Tropp, J. A. 2011, *SIAM Rev.*, **53**, 217
- Harper, S. E., Dickinson, C., Cleary, K. A., et al. 2024, *MNRAS*, submitted [arXiv:2405.04383]
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, **585**, 357
- Hunter, J. D. 2007, *Comput. Sci. Eng.*, **9**, 90
- Ihle, H. T., Borowska, J., Cleary, K. A., et al. 2022, *ApJ*, **933**, 185
- Karkare, K. S., Anderson, A. J., Barry, P. S., et al. 2022, *J. Low Temp. Phys.*, **209**, 758
- Keihänen, E., Keskitalo, R., Kurki-Suonio, H., Poutanen, T., & Sirviö, A.-S. 2010, *A&A*, **510**, A57
- Koranne, S. 2011, *Handbook of Open Source Tools* (Springer), 191
- Korngut, P. M., Bock, J. J., Akeson, R., et al. 2018, *SPIE*, **10698**, 576
- Kovetz, E. D., Viero, M. P., Lidz, A., et al. 2017, ArXiv e-prints [arXiv:1709.09066]
- Kovetz, E., Breysse, P. C., Lidz, A., et al. 2019, *BAAS*, **51**, 101
- Lamb, J. W., Cleary, K. A., Woody, D. P., et al. 2022, *ApJ*, **933**, 183
- Li, T. Y., Wechsler, R. H., Devaraj, K., & Church, S. E. 2016, *ApJ*, **817**, 169
- Lidz, A., Furlanetto, S. R., Oh, S. P., et al. 2011, *ApJ*, **741**, 70
- Loeb, A., & Zaldarriaga, M. 2004, *Phys. Rev. Lett.*, **92**
- Masui, K. W., Switzer, E. R., Banavar, N., et al. 2013, *ApJ*, **763**, L20
- Pearson, K. 1901, *Phil. Mag. J. Sci.*, **2**, 559
- Pullen, A. R., Chang, T.-C., Doré, O., & Lidz, A. 2013, *ApJ*, **768**, 15
- Pullen, A. R., Breysse, P. C., Oxholm, T., et al. 2023, *MNRAS*, **521**, 6124
- Rennie, T. J., Harper, S. E., Dickinson, C., et al. 2022, *ApJ*, **933**, 187
- Rudelson, M., & Vershynin, R. 2010, *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II-IV: Invited Lectures* (World Scientific), 1576
- Santos, M. G., Cluver, M., Hilton, M., et al. 2017, ArXiv e-prints [arXiv:1709.06099]
- Stein, G., Alvarez, M. A., & Bond, J. R. 2019, *MNRAS*, **483**, 2236
- Stutzer, N. O., Lunde, J. G. S., Breysse, P. C., et al. 2024, *A&A*, **691**, A336
- Tamuz, O., Mazeh, T., & Zucker, S. 2005, *MNRAS*, **356**, 1466
- Tegmark, M. 1997, *Phys. Rev. D*, **56**, 4514
- Vershynin, R. 2012, in *Introduction to the Non-asymptotic Analysis of Random Matrices*, eds. Y. C. Eldar, & G. Kutyniok (Cambridge: Cambridge University Press), 210
- Vieira, J., Aguirre, J., Bradford, C. M., et al. 2020, ArXiv e-prints [arXiv:2009.14340]
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *Nat. Methods*, **17**, 261

Appendix A: Dynamic PCA threshold

It is known that the largest singular value of a Gaussian $P \times N$ matrix with variance σ^2 can be approximated as $\lambda \approx C(P, N) \cdot \sigma(\sqrt{P} + \sqrt{N})$ for large matrices (Geman 1980; Rudelson & Vershynin 2010; Vershynin et al. 2012), where $C(P, N) \approx 1$ is some correction-factor for which no reliable theoretical model exists. We, therefore, simulated 50,000 noise matrices and empirically solved for $C(P, N)$ within the relevant regime of N from 5,000 to 30,000, and P from 100 to 80,000, that captures all matrix sizes our pipeline will encounter. We find that the correction factor can, in our size range, be well-modeled as

$$C(P, N) = 1.00476 - 0.00396 \cdot \log(P) \cdot \log(N) + 0.0000876 \cdot \log(P)^2 \cdot \log(N)^2. \quad (\text{A.1})$$

We have no doubt that this factor is unlikely to extrapolate sensibly beyond the range we tested it in, but that is of no concern. The relative error to the mean of our 50,000 simulations is less than 0.1% within our defined bounds.

Appendix B: The map-domain PCA filter

B.1. The effect of noise-weighting on PCA

This section expands on the PCA discussion of Sect. 4.2, and we keep the same variable and index names, for easier comparison. The first principal component $\mathbf{w}^1$ and its amplitude $\mathbf{a}^1$ of a PCA¹⁰ for a data-matrix $\mathbf{m}$ are the vectors that minimize

$$f(\mathbf{a}, \mathbf{v}) = \sum_v \sum_p (m_{v,p} - a_v w_p)^2. \quad (\text{B.1})$$

We might, however, want to find these vectors while weighting the elements of $\mathbf{m}$, for example, if it is non-uniformly noisy. If the weights themselves can be separated into an outer product, such that we have row and column uncertainties σ^{row} and σ^{col} , this can trivially be done by inverse-variance weighting the matrix elements of the expression above. We now minimize

$$f(\mathbf{a}, \mathbf{w}) = \sum_v \sum_p \frac{(m_{v,p} - a_v w_p)^2}{\sigma_v^{\text{row}} \sigma_p^{\text{col}}}, \quad (\text{B.2})$$

which can be expanded to

$$f(\mathbf{a}, \mathbf{w}) = \sum_v \sum_p \left(\frac{m_{v,p}}{\sqrt{\sigma_v^{\text{row}} \sigma_p^{\text{col}}}} - \frac{a_v}{\sqrt{\sigma_v^{\text{row}}}} \frac{w_p}{\sqrt{\sigma_p^{\text{col}}}} \right)^2. \quad (\text{B.3})$$

This is still a valid PCA problem, on the same form as Eq. (B.1), with $\mathbf{a}' = \mathbf{a}/\sigma^{\text{row}}$ and $\mathbf{w}' = \mathbf{w}/\sigma^{\text{col}}$ now being the amplitude and component we fit for. If $\mathbf{m}$ contains a feature that can be decomposed into an outer product, this will be recovered by $\mathbf{w} = \mathbf{w}'\sigma^{\text{col}}$ and $\mathbf{a} = \mathbf{a}'\sigma^{\text{row}}$.

However, if we want to weight every element of $\mathbf{m}$ with arbitrary uncertainty $\sigma_{v,p}$, this no longer holds. We can still write the problem simply as

$$f(\mathbf{a}, \mathbf{w}) = \sum_v \sum_p \frac{(m_{v,p} - a_v w_p)^2}{\sigma_{v,p}^2}, \quad (\text{B.4})$$

¹⁰ In the regular PCA formalism, the eigenvectors $\mathbf{w}$ are typically unit vectors of length 1, which is not automatically the case throughout this section. However, $\mathbf{w}$ can be normalized to 1 at any point by inversely adjusting $\mathbf{a}$. In a regular PCA, a (scalar) singular value is present in the solution. In the formalism presented here, there is no explicit singular value, and it can be absorbed into the amplitudes $\mathbf{a}$.

which can be expanded to

$$f(\mathbf{a}, \mathbf{w}) = \sum_v \sum_p \left(\frac{m_{v,p}}{\sigma_{v,p}} - \frac{a_v}{\sqrt{\sigma_{v,p}}} \frac{w_p}{\sqrt{\sigma_{v,p}}} \right)^2, \quad (\text{B.5})$$

but because $a_v/\sigma_{v,p}$ and $w_p/\sigma_{v,p}$ are now matrices and not vectors, this minimization problem is no longer a PCA. We therefore have no way of recovering the desired $\mathbf{a}$ and $\mathbf{w}$ corresponding to the matrix $\mathbf{m}$ if we perform the regular PCA algorithm on the matrix $m_{v,p}/\sigma_{v,p}$, for a general $\sigma_{v,p}$. We must therefore find a different way of minimizing B.4, which is discussed in the following section.

B.2. Generalization of the PCA algorithm

With the generalization of Eq. (B.4), we can no longer utilize the usual methods of solving a PCA problem, such as the SVD. Instead we employ the technique suggested by Tamuz et al. (2005) and Gabriel & Zamir (1979), where we iteratively make improved guesses at $\mathbf{w}$ and $\mathbf{a}$:

1. Make an initial guess at $\mathbf{a}$ and $\mathbf{w}$, either completely random or informed by some knowledge of the data.
2. Solve for the optimal $\mathbf{a}$ while holding the current $\mathbf{w}$ constant by differentiating Eq. (B.4), holding $\frac{df(\mathbf{a}, \mathbf{w})}{d\mathbf{a}} = 0$, and solving for $\mathbf{a}$ as

$$a_v = \frac{\sum_p \frac{m_{v,p} w_p}{\sigma_{v,p}^2}}{\sum_p \frac{w_p^2}{\sigma_{v,p}^2}}. \quad (\text{B.6})$$

3. Given the new $\mathbf{a}$, calculate $\frac{df(\mathbf{a}, \mathbf{w})}{d\mathbf{w}} = 0$, and solve for the new optimal $\mathbf{w}_p$ as

$$w_p = \frac{\sum_v \frac{m_{v,p} a_v}{\sigma_{v,p}^2}}{\sum_v \frac{a_v^2}{\sigma_{v,p}^2}}. \quad (\text{B.7})$$

4. Repeat 2. and 3. until the incremental changes in $\mathbf{a}$ and $\mathbf{w}$ are below some chosen threshold ϵ .

Although we cannot prove that this is a convex problem and that the optimal solution is guaranteed, we have never seen it converge to an unreasonable solution. Additional robustness can be achieved by repeating the fit with different initial guesses, and confirming that they converge to the same solution. The algorithm will converge on the same solution as the regular PCA in the case of uniform weights $\sigma_{v,p} = 1$, or where the weights can be perfectly decomposed into an outer product of weights in rows and columns, as in Eq. (B.2).

B.2.1. Multiple components

Using this decomposition, we can fit multiple components, similar to the ordered set of principal components in the PCA, by simply subtracting the previous components from the data. We then simply define $a_i^{(1)}$ and $m_j^{(1)}$ as the results from the previous section, and let

$$m_{v,p}^{(1)} = m_{v,p} - a_v^1 w_p^1. \quad (\text{B.8})$$

We can then find $a_v^{(2)}$ and $w_p^{(2)}$ by performing the procedure from the previous section on $m_{v,p}^{(1)}$. This is again, in the case of $\sigma_{v,p} = 1$, entirely equivalent to finding the largest principal components of $m_{v,p}$.

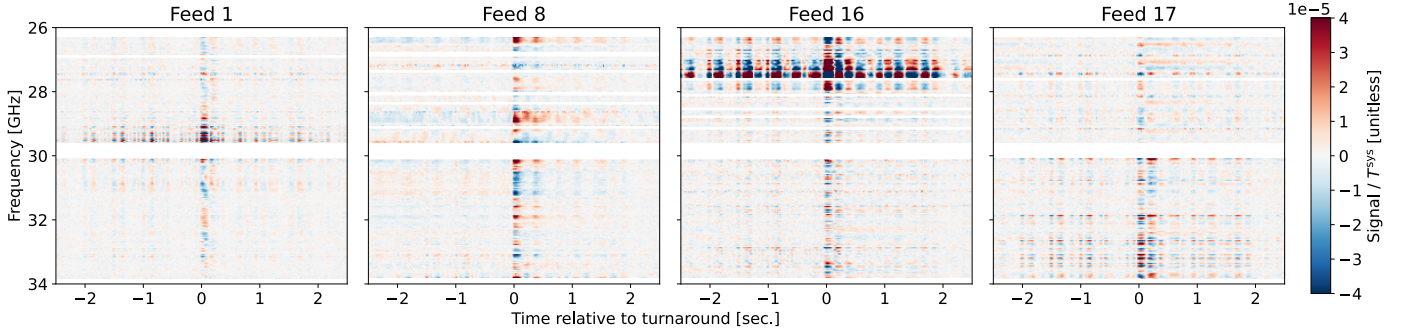


Fig. C.1. Filtered time-domain data stacked on the turn-around of the telescope, to emphasize the turn-around systematic effect, for four selected feeds. The data is an average across all turn-arounds of all available Level 2 scans. The data is divided by the system temperature of the data in each channel, and the values therefore represent the signal strength of the effect relative to the average white noise level in the scans.

Appendix C: Start-of-scan and turn-around effects

As outlined in Sect. 4.3, our increased sensitivity has revealed two systematic effects in our maps that were not discovered in our ES publications. We here explore these effects in more detail, especially in the time-domain, where these effects are easier to understand than in the map-domain. We note that the data shown in this section has not been filtered by the map-PCA filter, and the systematic effects demonstrated are (to the best of our analysis) not present in our final maps.

C.1. The turn-around effect

The turn-around effect is a sharp feature located around the azimuth edges of the scanning pattern, where the telescope turns. In Fields 2 and 3, which rise and set almost vertically across the sky (see Figure 1) and are therefore observed at almost the same angle at all times, this effect manifests as sharp edges on the top and bottom (i.e., the highest and lowest declinations) of the equatorial coordinate maps. This can be seen in the first and third rows of Fig. 10.

To better understand this systematic effect, we have extracted the data around the telescope turn-arounds for all our scans, and stacked the result on the turn-around time. Figure C.1 shows the result for four selected feeds. For all four feeds, we see a feature that peaks around the turn but is also present both leading up to and after the turn. In the frequency direction the feature has a slow wave-like feature. The feature manifests differently in different feeds and frequencies but has in common that it is wave-like both in frequency and time and peaks in power around the turn-around. As the telescope turn-arounds are the regions with the highest acceleration, a likely origin of this effect is some standing wave induced by the mechanical vibrations of the azimuth drive. Some feeds show significantly stronger manifestations of the turn-around effect than others, but all feeds are affected to some extent, and no explanation has yet been found as to why feeds are affected differently.

Attempts have been made to model this effect in the time-domain. This has proven difficult, among other reasons because the effect is actually very weak compared to the noise level in a single scan. As seen from Fig. C.1, the effect peaks at more than four orders of magnitude below the noise temperature of the telescope. The effect is only visible in the final maps because it seems strongly coherent across different scans. However, because we have to combine thousands of scans in order to observe the effect, it is also difficult to assess if the effect is indeed perfectly coherent across all scans, or if we are simply observing the average impact of this effect. The efforts of model-

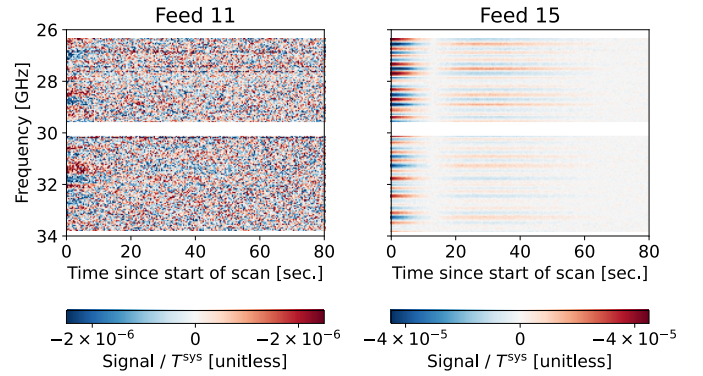


Fig. C.2. Same setup as Figure C.1, but stacked on the beginning of each scan, to emphasize the start-of-scan systematic effect. Notice the difference in the colorbar limits.

ing the effect in the time domain was also made somewhat moot by how effective the map-PCA was at removing the effect in the map-domain.

C.2. The start-of-scan effect

The start-of-scan effect is similar to the turn-around effect in that it is also weak in individual scans, but coherently adds as we add scans. Figure C.2 shows a plot similar to what was presented in the previous section, but that stacks all available Level 2 scans on the beginning of each scan. For Feed 15, we see a very strong wave across frequency, that falls to zero around 17 seconds after the start of the scan, and then switches from negative to positive, or positive to negative power. This is simply an artifact of the low-pass normalization we perform during TOD processing, and the strong wave at the very beginning of each scan is the real start-of-scan feature. This artifact explains why we, in the second and third rows of Figure 10 (where the start-of-scan effect can be seen), observe a similar switching of power as we move from the right edge of the map and toward the center.

Looking at Feed 11 in Figure C.2 a much weaker, but similar, start-of-scan feature can be seen. Generally, all feeds show very similar behavior to either Feed 11 or 15: all Feeds associated with DCM1-2 (Feeds 6, 14, 15, 16, and 17) have very similar behavior, and all remaining Feeds show only a weak start-of-scan effect, as appears in Feed 11. It is unclear why this clear divide exists, and how it relates to DCM1-2.

The exact origin of the start-of-scan effect is unknown, but a standing wave induced by mechanical vibration is also a strong

candidate for this systematic effect. The re-pointing that is performed in between scans is currently the only time in the scanning strategy the elevation drive is utilized, as our scans are performed in constant elevation mode.

As with the turn-around effect, some effort was made to model the start-of-scan effect in the time domain. This was fairly successful, and fitting a decaying exponential function to the beginning of each scan appeared to remove more than 90% of the signal induced by this effect. However, the map-PCA proved much more effective than the time-domain efforts, and they are therefore not employed.

Appendix D: Signal injection example maps

Figure D.1 illustrates the signal injection pipeline. The first figure shows the CO simulation itself, while the subsequent three panels show the results of injecting this simulation into the real data with different boost strengths. We note that the simulation is injected into the TOD of the Level 1 data, and the maps shown have gone through the entire pipeline.

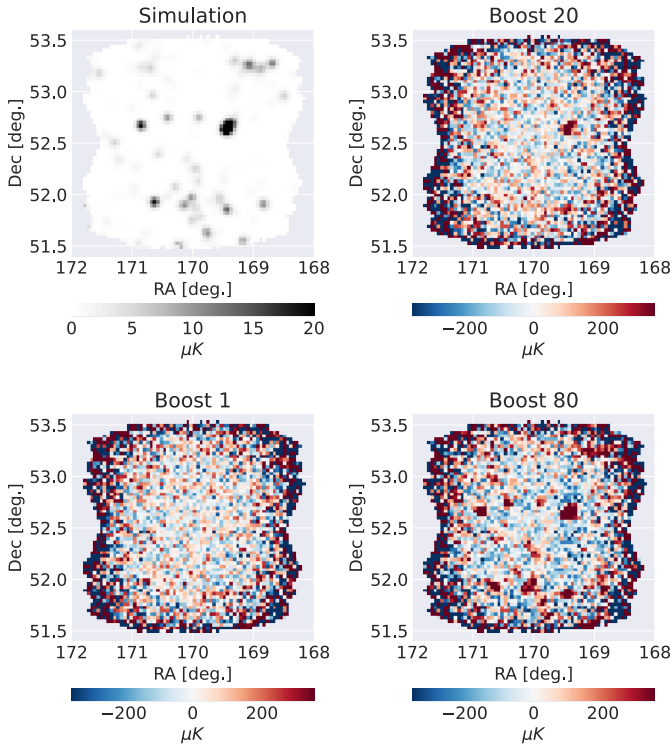


Fig. D.1. Illustration of the signal-injection method for transfer function analysis for different signal boost strengths. The top left plot shows the signal-only CO simulation over the relevant patch. The remaining plots show the resulting maps of the simulation injected into the real TOD with different boost strengths and passed through the entire COMAP pipeline. The boost is relative to the fiducial model of [Chung et al. \(2022\)](#), used in the simulations. All four plots show the same frequency slice centered at 26.953GHz .

Appendix E: Uncertainty and frequency maps

Figure E.1 compares the uncertainties of the Season 2 maps to the Season 1 maps. The values are averages across all frequencies, calculated by inverse-variance co-addition of the uncertain-

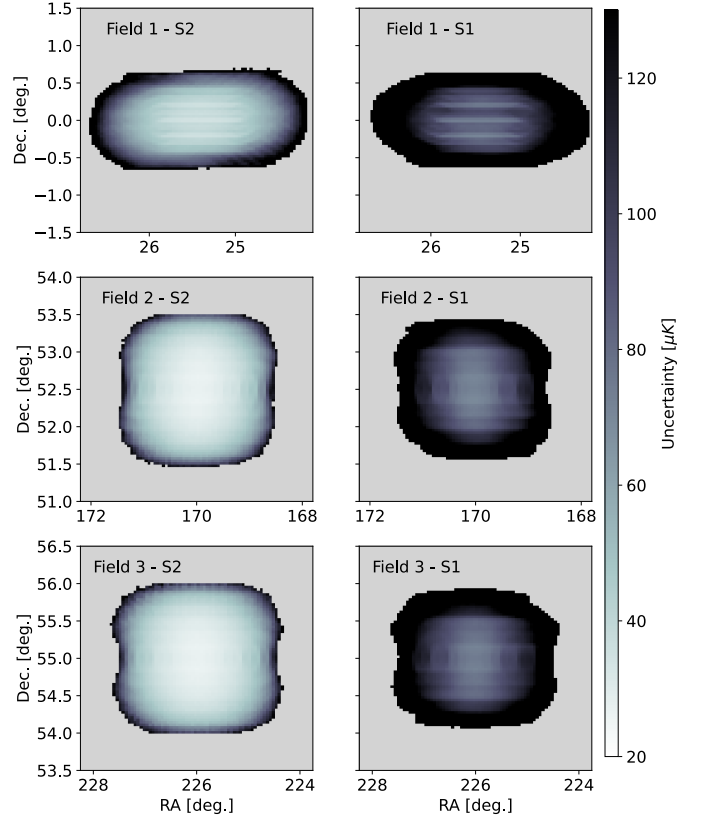


Fig. E.1. Uncertainties across the three fields for Season 2 (S2) maps published in this work (left), and the Season 1 (S1) maps published in ES (right).

ties, as

$$\sigma_{\text{mean}} = \sqrt{\frac{1}{\langle 1/\sigma_v^2 \rangle}}. \quad (\text{E.1})$$

All frequencies have relatively similar uncertainties, with some exceptions close to the Band edges. The center of the maps have a uncertainties of around $25\mu\text{K}$, while the high-sensitivity $\sim 1.5^\circ \times 1.5^\circ$ regions have an uncertainty of $< 50\mu\text{K}$ for Fields 2 and 3, with the Field 1 region being slightly smaller.

Figure E.2 shows feed-coadded individual frequency maps for Field 2 across $32\text{GHz} - 34\text{GHz}$ (1/4th of all channels) for all the data of Season 1 and 2 combined, processed with the Season 2 pipeline. All maps are noise dominated after the map-level PCA filtering. The noise increases toward the highest frequencies because of aliasing cuts on older data (see Sect. 2.5).

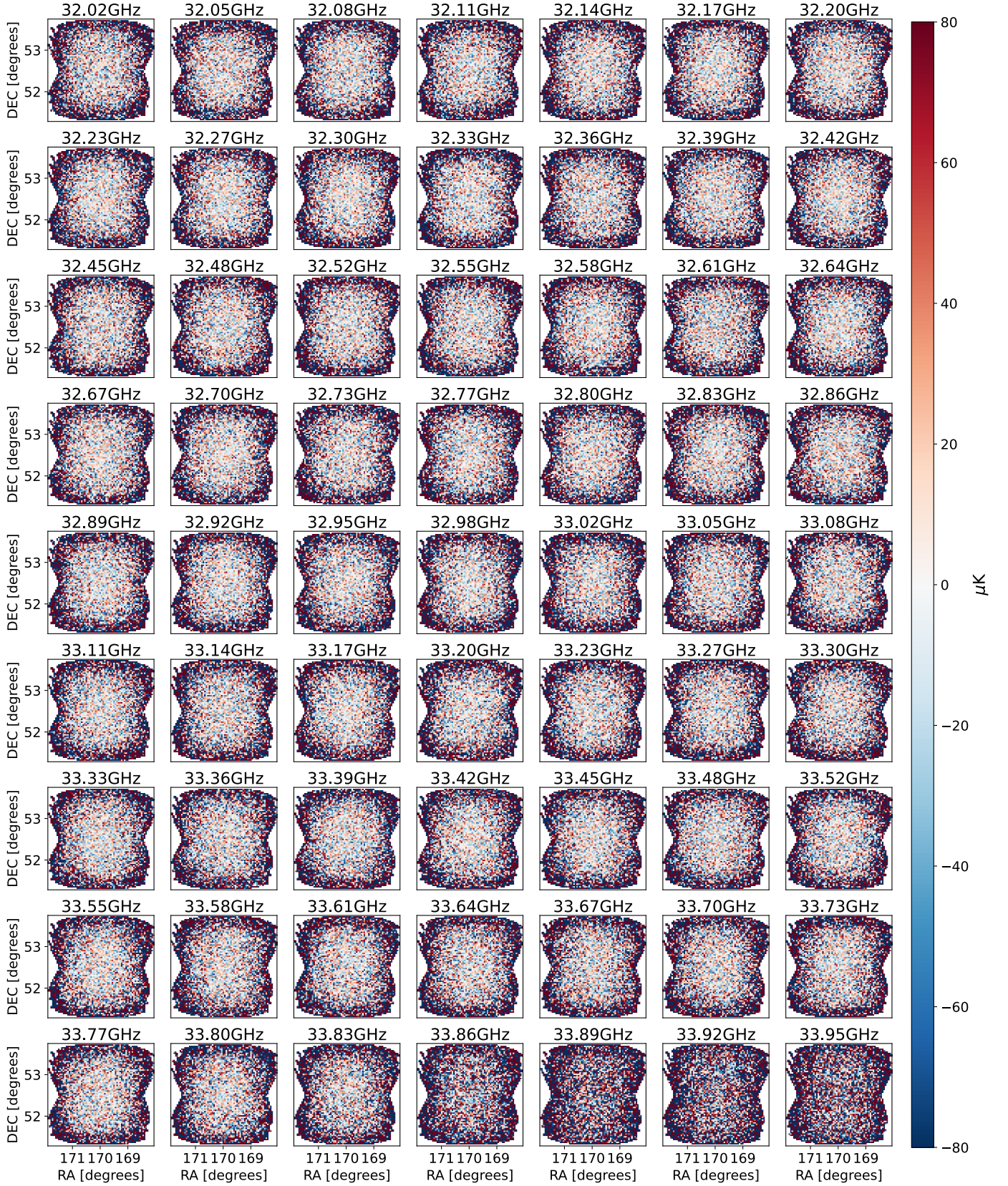


Fig. E.2. Final 31.25 MHz wide frequency maps for sideband A:USB and Field 2. The titles of each sub-plot indicate the center frequencies of each frequency map. The maps are shown for completeness, and no interesting features beyond the noise can be seen in the maps.