

A novel approach to optimize clustering of parametric map-based component separation for upcoming CMB polarization satellites

Wassim Kabalan¹*, Wuhyun Sohn¹, Benjamin Beringue¹, Artem Basyrov¹,
 Pierre Chanial¹, Alexandre Boucaud¹, Josquin Errard¹†

¹Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We present a novel JAX-powered parametric component separation approach that optimizes the degrees of freedom in the component separation parametrization, achieving minimum variance after foreground cleaning. This work is the first to incorporate an algorithm for optimizing the clustering of sky regions where the frequency scaling laws are assumed constant. Our technique is particularly applied to the challenging large angular scales encountered in future CMB space missions—illustrated here with a LiteBIRD-like instrument targeting the measurement of primordial gravitational waves down to $r < 0.001$. Furthermore, our efficient and modular framework paves the way for the extension of traditional component separation techniques.

Key words: Cosmic Microwave Background – Component Separation – Parametric Methods – Clustering – Tensor-to-Scalar Ratio

1 INTRODUCTION

Detecting the B-mode polarization in the Cosmic Microwave Background (CMB) is a central objective in modern cosmology, as it offers direct observational evidence of primordial gravitational waves generated during the universe’s inflationary period (Kamionkowski et al. 1997; Seljak & Zaldarriaga 1997). The amplitude and shape of this B-mode signal are direct probes of inflationary energy scales, potentially reaching physics at $\sim 10^{16}$ GeV. Upcoming experiments, such as LiteBIRD (Hazumi 2022), aim to measure these faint signals with unprecedented sensitivity, targeting constraints on the tensor-to-scalar ratio r down to levels of $r < 0.001$.

Achieving these ambitious goals is complicated by the presence of astrophysical foregrounds, primarily arising from Galactic synchrotron emission and thermal dust (Planck Collaboration 2016b; Meisner & Finkbeiner 2015). These foregrounds dominate the CMB polarization signal across much of the sky and exhibit spatially varying spectral properties. Large angular scales (low multipoles) are both most sensitive to primordial B-modes and most affected by foregrounds and instrument systematics, making accurate separation especially critical in this regime.

Component separation methods can be broadly classified as parametric—assuming a model for the spectral energy distributions (SEDs) of foregrounds—or non-parametric, which rely on statistical independence or internal templates (Delabrouille 2003; Cardoso 2008). Key tools include COMMANDER (Eriksen 2008; Planck Collaboration 2016c), SMICA (Cardoso 2008; Planck Collaboration 2016a), and FGBUSTER (Stompor & Errard 2018; Errard 2021), each providing different approaches to the trade-off between flexibility and tractability.

Our method builds on the parametric class, enhancing its flexibility by introducing spatial variability in a systematic, data-driven manner. Although many traditional parametric approaches assume spatial uniformity in spectral parameters, recent observations indicate that the spectral energy distributions of foregrounds vary significantly across the sky (Planck Collaboration 2016b; Meisner & Finkbeiner 2015). Frameworks such as FGBUSTER (Stompor & Errard 2018; Errard 2021), as used in (Allys 2022), address this by supporting *clustered* parameter configurations—where the sky is divided into a finite number of regions (or *clusters*), each sharing a common set of spectral parameters. However, the computational cost of optimizing multiple such clustering choices in these frameworks is often prohibitive, limiting their practical flexibility.

In contrast, our framework is explicitly designed to efficiently explore a large space of clustering configurations. Our implementation, based on FURAX (Chanial & the CMBSciPol team 2024) (Chanial et al. In Prep), can be viewed as a JAX-native generalization of FG-BUSTER, built for end-to-end differentiability, GPU acceleration, and large-scale model selection. Leveraging the high-performance computing capabilities of JAX (Bradbury et al. 2018), FURAX enables scalable and reproducible component separation pipelines suited for next-generation satellite missions.

In the remainder of this paper, we detail our methodology (Section 2), illustrate our framework’s advantages over existing techniques (Section 5), and discuss implications for future observational missions (Section 6).

2 METHODOLOGY

2.1 Parametric Component Separation

We model multi-frequency sky observations using the standard parametric framework. The observed Stokes vector in each pixel is mod-

* Contact: wassim@apc.in2p3.fr

† Contact: josquin@apc.in2p3.fr

eled as a linear combination of astrophysical components with additive (Gaussian) noise:

$$\mathbf{d} = \mathbf{A}(\boldsymbol{\beta}) \mathbf{s} + \mathbf{n}, \quad (1)$$

where:

- $\mathbf{d} \in \mathbb{R}^{N_d}$: observed data vector (frequency \times polarization \times pixels),
- $\mathbf{s} \in \mathbb{R}^{N_s}$: sky components at a reference frequency (polarization \times pixels),
- $\mathbf{A}(\boldsymbol{\beta}) \in \mathbb{R}^{N_d \times N_s}$: mixing matrix encoding spectral dependencies,
- $\mathbf{n} \sim \mathcal{N}(0, \mathbf{N}) \in \mathbb{R}^{N_d}$: Gaussian noise with known covariance \mathbf{N} (frequency \times polarization \times pixels),

Each column of the mixing matrix $\mathbf{A}(\boldsymbol{\beta})$ models the spectral energy density of each component. In practice, this includes a modified blackbody (MBB) emission law for thermal dust and a power-law dependence for synchrotron radiation, with $\boldsymbol{\beta}$ denoting the set of corresponding spectral parameters ($\beta_{dust}, T_{dust}, \beta_{synchrotron}$).

Assuming Gaussian noise, the negative log-likelihood under this model is:

$$-2 \ln \mathcal{L}(\mathbf{s}, \boldsymbol{\beta}) = (\mathbf{d} - \mathbf{As})^\top \mathbf{N}^{-1} (\mathbf{d} - \mathbf{As}) + \text{const}, \quad (2)$$

To estimate the sky components \mathbf{s} , we solve:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} \Big|_{\mathbf{s}=\hat{\mathbf{s}}} = 0, \quad (3)$$

which yields the generalized least squares solution:

$$\hat{\mathbf{s}} = (\mathbf{A}^\top \mathbf{N}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{N}^{-1} \mathbf{d}, \quad (4)$$

Substituting this into the likelihood (2) eliminates dependence on \mathbf{s} , giving the spectral likelihood (Stompor et al. 2009):

$$\ln \mathcal{L}_{\text{spec}}(\boldsymbol{\beta}) \propto (\mathbf{A}^\top \mathbf{N}^{-1} \mathbf{d})^\top (\mathbf{A}^\top \mathbf{N}^{-1} \mathbf{A})^{-1} (\mathbf{A}^\top \mathbf{N}^{-1} \mathbf{d}), \quad (5)$$

which depends only on the spectral parameters $\boldsymbol{\beta}$, and is used to optimize their values.

2.2 Cluster-Based Generalization

To account for spatial variability in foreground spectral parameters, we generalize the model (1) by allowing $\boldsymbol{\beta}$ to vary across the sky. However, assigning one parameter set per pixel would increase the statistical uncertainty of the recovered components—particularly the CMB and derived cosmological parameters (Errard & Stompor 2015a). To mitigate this, we introduce a clustered model in which the sky is divided into regions (or *clusters*) that share common spectral parameters.

We use *spherical K-means* clustering to partition the sky into disjoint regions $\{C_k\}$, where:

$$\boldsymbol{\beta}(\hat{n}) = \boldsymbol{\beta}_k \quad \text{for all } \hat{n} \in C_k,$$

thus assigning a single set of spectral parameters to each region. The clustering configuration is parameterized solely by two quantities: the random seed used for centroid initialization and the total number of desired clusters.

Our implementation, available via `jax-healpy` (Chaniai, Biquard & Kabalan Chaniai et al.), is adapted specifically for spherical coordinates and inspired by the `kmeans_radec` package (Sheldon 2018). Clustering operates on right ascension and declination coordinates, minimizing angular distances across the celestial sphere. Although input points are in spherical coordinates, centroid updates during

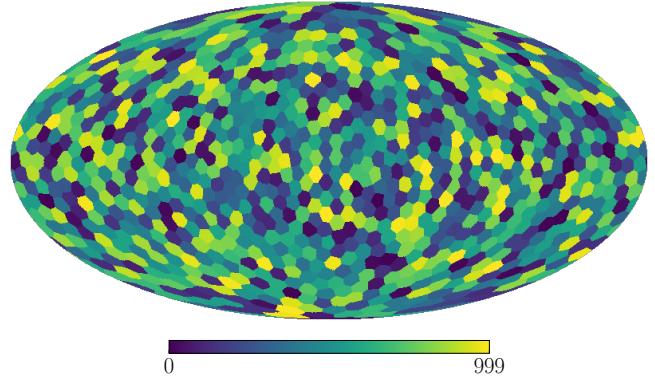


Figure 1. Example of spherical K-means clustering applied to a HEALPix-formatted dust map, showing regions sharing a common spectral parameter. Each color represents a distinct cluster.

K-means iterations are performed via 3D Cartesian averaging (over (x, y, z) coordinates mapped from (RA, DEC)) to ensure numerical stability and efficient convergence as shown in 1.

The entire clustering algorithm is implemented to be fully compatible with JAX’s functional programming paradigm and accelerator-based execution. It is fully vectorized, supports automatic differentiation (autodiff), and avoids explicit Python loops, making it highly scalable for large sky datasets and efficient within end-to-end optimization frameworks.

In practice, we initialize the clustering of sky pixels using pre-defined templates such as the Planck Galactic masks (e.g., GAL020, GAL040) (Planck Collaboration 2016d). These masks divide the sky into regions with different levels of Galactic foreground contamination—typically distinguishing low-, medium-, and high-foreground areas based on Galactic latitude or emission intensity thresholds. Each clustering configuration—defined by the number of patches used for parameters like β_d , T_d , and β_s —specifies a distinct model within the data framework of Equation (1). Rather than optimizing the clustering assignments directly, we evaluate a grid of such configurations by computing the spectral likelihood in Equation (5) for each mask, selecting the one that minimizes the variance of the recovered CMB.

2.3 Grid Search over Clustering Models

Our model allows the clustering of different spectral parameters (e.g., dust and synchrotron) on independent sets of sky patches, without requiring them to align spatially. The number of clusters assigned to each spectral parameter is treated as a discrete modeling choice, assumed specifically for the purpose of component separation. For example, one configuration might use 100 patches for β_d , 20 patches for T_d , and 10 patches for β_s .

Letting \mathcal{G} denote the space of possible clustering configurations:

$$\mathcal{G} = \{K_{\beta_d}\} \times \{K_{T_d}\} \times \{K_{\beta_s}\},$$

we perform a structured grid search across \mathcal{G} , evaluating each configuration $\{C_k\} \in \mathcal{G}$ by fitting spectral parameters $\boldsymbol{\beta}$ and reconstructing the CMB component. Specifically, we maximize the spectral likelihood:

$$\mathcal{L}_{\text{spec}}(\boldsymbol{\beta}, \{C_k\}),$$

where the likelihood depends on both the parameter values and the spatial clustering structure.

Maximizing the spectral likelihood ensures a good fit to the observed frequency maps. However, it does not by itself prevent overfitting, especially when increasing the number of free parameters by introducing finer spatial patching. In low signal-to-noise regions, overly flexible models can fit noise realizations rather than true sky signals, increasing statistical residuals in the reconstructed CMB map.

To mitigate this, we define a secondary selection metric—based on the variance of the reconstructed CMB across noise realizations (described in Section 2.5). The CMB variance acts as a proxy for statistical residuals: minimizing it leads to reduced noise-induced contamination and more robust recovery of the underlying cosmological signal. Thus, the optimal clustering configuration is chosen to balance spectral fit quality with minimization of statistical uncertainty.

Each configuration is assessed over multiple noise realizations and sky regions, ensuring that the selected model is both statistically stable and robust against noise-driven biases.

Given the large size of the configuration space and the need to evaluate multiple noise realizations and sky regions, we developed a distributed, parallel evaluation framework, described next.

2.4 Distributed and Parallel Execution

To evaluate large grids of clustering configurations across multiple noise realizations and sky regions, we developed a distributed optimization engine: `jax-grid-search` (Wassim 2025). This JAX-native framework enables high-throughput exploration of parametric component separation models at scale, fully leveraging modern GPU architectures.

As defined previously, the grid \mathcal{G} spans combinations of patch counts across spectral parameters.

Each element in this grid represents a distinct clustering configuration. For every such configuration, we evaluate the component separation pipeline across multiple noise realizations and sky regions.

To handle this computational load efficiently, we combine two forms of parallelism:

- Intra-device parallelism via `jax.vmap`, enabling batched execution of component separation fits on a single GPU.
- Inter-device parallelism using MPI-style slicing: the global parameter grid is evenly partitioned across all processes, regardless of whether the number of combinations is divisible by the number of workers.

Formally, for P total processes, each process $I \in [0, P - 1]$ is assigned a contiguous chunk of the global grid:

$$\mathcal{G}_I = \mathcal{G} \left[\left\lfloor \frac{I \cdot N}{P} \right\rfloor : \left\lfloor \frac{(I + 1) \cdot N}{P} \right\rfloor \right], \quad N = |\mathcal{G}|$$

This ensures robust workload distribution, load balancing, and fault tolerance. It avoids assumptions about grid divisibility and gracefully handles incomplete or interrupted runs—making it well suited for large-scale HPC deployments.

Each batch is independently evaluated, with results checkpointed to disk for aggregation and analysis. This modular design supports recovery, progress tracking, and reproducibility—all essential for exploratory model selection pipelines.

2.5 Clustering Objective: Variance Minimization

Let $\{\mathcal{C}\}$ denote the set of all clustering configurations evaluated in our grid search, and let $C_k \in \{\mathcal{C}\}$ denote an individual configuration.

For each clustering configuration in our grid, we perform maximum likelihood estimation of the spectral parameters:

$$\forall C_k \in \{\mathcal{C}\} : \quad \boldsymbol{\beta}_k^* = \arg \max_{\boldsymbol{\beta}_k} \mathcal{L}_{\text{spec}}(\boldsymbol{\beta}_k, C_k) \quad (6)$$

Using these parameters, we reconstruct the sky components via:

$$\hat{\mathbf{s}} = \left(\mathbf{A}(\boldsymbol{\beta}_k^*)^\top \mathbf{N}^{-1} \mathbf{A}(\boldsymbol{\beta}_k^*) \right)^{-1} \mathbf{A}(\boldsymbol{\beta}_k^*)^\top \mathbf{N}^{-1} \mathbf{d} \quad (7)$$

$$\equiv W(\mathbf{d}) \quad (8)$$

From the reconstructed components, we extract the CMB signal \hat{s}_{CMB} . We compute its variance across noise realizations and average over sky pixels to obtain the scalar objective:

$$\sigma_{\text{CMB}}^2 = \left\langle \text{Var}_i \left[\hat{s}_{\text{CMB}}^{(i)} \right] \right\rangle_{\text{pixels}} \quad (9)$$

Although the spectral parameters $\boldsymbol{\beta}$ are defined in a clustered fashion—varying across spatial regions—the reconstructed components $\hat{\mathbf{s}}$, and hence \hat{s}_{CMB} , are evaluated as a full-sky solution.

This variance serves as the loss function for selecting the best clustering configuration. It acts as a proxy for residual foreground contamination and directly impacts the uncertainty on the inferred tensor-to-scalar ratio r .

Minimizing this variance reduces both statistical uncertainty and systematic leakage in the B -mode power spectrum. As shown in (Errard & Stompor 2015b), residuals from imperfect foreground modeling propagate into biases on C_ℓ^{BB} , making this metric critical for achieving robust and unbiased cosmological constraints.

2.6 The FURAX Framework

To support this method, we introduce FURAX (Chania & the CMB-SciPol team 2024)—a modular JAX-powered framework to express and optimize parametric component separation likelihoods.

FURAX is built around a central goal: to express symbolic algebraic expressions (e.g., likelihoods) directly in code using modular, composable operators. These operators abstract common structures like the mixing matrix, noise weighting, or parameter dispatching, and can be combined into scalable computation graphs.

Key features:

- Differentiable algebraic operators for the *mixing matrix* $\mathbf{A}(\boldsymbol{\beta})$, the *noise diagonal* operator \mathbf{N}^{-1} , and cluster-wise *SED parameter dispatch* using mask-based routing.
- Conjugate gradient solvers via the Lineax library (Rader et al. 2023) to solve inverse systems efficiently without explicitly forming matrices.
- Fully vectorized execution across simulation ensembles and clustering grids, leveraging JAX (Bradbury et al. 2018) for efficient large-scale computation.

All core computations in FURAX are implemented using memory-efficient linear operators, avoiding explicit dense matrices. This symbolic and modular approach allows FURAX to express and optimize large, realistic models without the memory bottlenecks associated with explicit matrix construction. Ongoing companion developments include forthcoming works on HWP modeling (Tsang King Sang et al.), the MegaTop pipeline (Beringue et al.), the ATOP map-maker

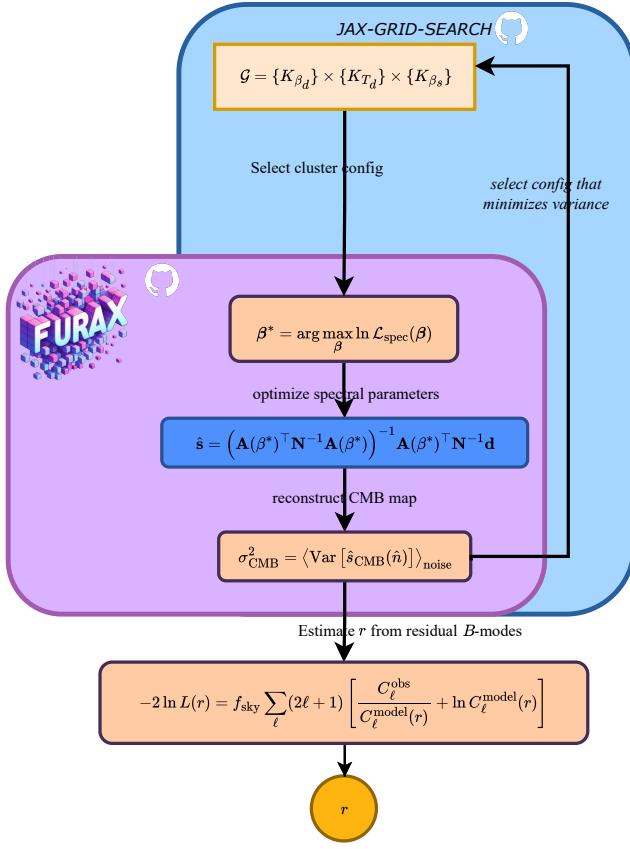


Figure 2. Overview of the FURAX pipeline for component separation and tensor-to-scalar ratio estimation. The process begins by selecting a clustering configuration from the discrete grid $\mathcal{G} = \{K_{\beta_d}\} \times \{K_{T_d}\} \times \{K_{\beta_s}\}$, evaluated in parallel using JAX-GRID-SEARCH. Each configuration undergoes spectral parameter optimization and map reconstruction using FURAX. The variance of the recovered CMB map is used as a selection metric to identify the best configuration. The final CMB map is then used to compute the B -mode power spectrum and infer the tensor-to-scalar ratio r .

(Sohn et al.), spherical harmonic tooling (Basyrrov), and extended forecast studies (Villarrubia-Aguilar et al.).

In the scope of this paper, we assume isotropic white noise and matched beam resolution. Although advanced treatments (e.g., beam convolution, correlated noise) exist in the literature, they are not our focus here. However, FURAX is designed to make such extensions straightforward in future developments, including applications to map-making, half-wave plate systematics, atmospheric effects, and beyond.

The framework integrates natively with the JAX ecosystem and supports optimization and inference workflows through libraries such as Optax (DeepMind 2020) and NumPyro (Bingham et al. 2019).

3 SPECTRAL LIKELIHOOD MAXIMIZATION

The central task in parametric component separation is the estimation of spectral parameters β by maximizing the spectral likelihood $\mathcal{L}_{\text{spec}}(\beta)$ (5). This likelihood serves as the objective function that drives parameter inference across the model.

Thanks to JAX’s automatic differentiation, gradients of this objective with respect to all free parameters are computed efficiently. This

applies even in spatially varying models, where spectral parameters are defined per cluster and must be dispatched to the appropriate sky pixels. Crucially, this indexing and dispatching operation, performed via binary masks, are themselves implemented in a differentiable manner. As a result, the likelihood function is compatible with gradient-based optimization.

FURAX performs spectral likelihood maximization using gradient-based optimization from the Optax library (DeepMind 2020). The default method is L-BFGS, a limited-memory quasi-Newton optimizer that leverages curvature information from past gradients. Ideally, optimization would proceed via a full Newton step:

$$\theta_{n+1} = \theta_n - H^{-1} \nabla L(\theta_n), \quad (10)$$

where H is the Hessian of the loss. However, computing and inverting this matrix becomes intractable for large-scale clustered models with thousands of parameters. L-BFGS avoids this cost by constructing a low-rank approximation using a limited history of gradients and parameter updates, enabling efficient and memory-scalable updates.

In practice, we observe that L-BFGS in FURAX exhibits superior stability and convergence (see Figure 3) compared to earlier approaches such as FG Buster. This is primarily due to two factors: (i) we use exact gradients from automatic differentiation in JAX, ensuring consistent and smooth optimization dynamics; and (ii) we avoid the use of Singular Value Decomposition (SVD) (Golub & Reinsch 1970), which FG Buster employs as a performance shortcut at the cost of numerical noise in gradients. These design choices make FURAX robust to initialization and capable of converging reliably even in high-dimensional, highly clustered configurations.

We also experimented with TNC (Truncated Newton with Constraints) (Waltz et al. 2006), as used in FG Buster. In our framework, TNC is accessed through jaxopt (Blondel et al. 2021), which wraps `scipy.optimize.minimize` (Virtanen et al. 2020). However, this implementation runs exclusively on CPU and is incompatible with JAX’s `jit` compilation or `vmap` vectorization. More critically, TNC does not scale well with parameter count—as the number of clusters increases, the optimization problem becomes stiff and memory-bound, leading to significantly longer runtimes and poorer convergence behavior.

First-order optimizers such as Adam (Kingma & Ba 2017) were also tested, but proved to be ill-suited for this application. Adam applies a uniform learning rate schedule across all parameters, yet in component separation, the physical scales of spectral parameters differ substantially: for instance, β_d varies between 0.5 and 5, T_d between 10 and 50 K, and β_s between -6 and -1.

Moreover, the spectral likelihood typically reaches values on the order of 10^8 (for example, at $n_{\text{side}}=64$), making fixed or scheduled learning rates ineffective. In contrast, L-BFGS employs a line search strategy that dynamically adapts the step size to local curvature, enabling stable convergence even in high-dimensional, multi-scale optimization settings.

4 TENSOR-TO-SCALAR RATIO ESTIMATION

We estimate the tensor-to-scalar ratio r from the B -mode power spectrum of the reconstructed CMB maps. The process involves modeling residual contamination, combining it with signal templates, and performing likelihood-based inference of r (Stompor et al. 2016; Errard & Stompor 2019a).

To ensure statistical robustness, the full estimation pipeline is applied across multiple noise realizations and distinct sky regions. The resulting B -mode power spectra and corresponding likelihoods are

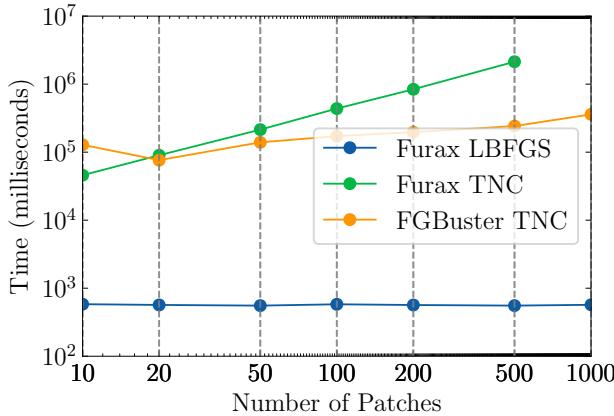


Figure 3. Runtime comparison of FGBuster (TNC), FURAX (TNC) using jaxopt, and FURAX (L-BFGS) using optax as a function of the number of clusters. L-BFGS scales robustly with parameter count and consistently outperforms TNC in clustered models.

then averaged over both axes, the noise and the sky region, to account for sample variance and foreground complexity.

4.0.0.1 BB Spectrum Convention. All angular power spectra are reported in the standard units:

$$\text{coeff} = \frac{\ell(\ell+1)}{2\pi},$$

4.0.0.2 Systematic and Statistical Residuals. To quantify contamination in the recovered CMB, we separate residuals into two components:

- Systematic residual power is obtained by rerunning the full component separation pipeline on foreground-only skies d_{ds} , which exclude both CMB and instrumental noise. This evaluates the imprint of foreground leakage that cannot be filtered out by the model:

$$C_{\ell}^{\text{syst}} = C_{\ell}(W(d_{\text{ds}})), \quad (11)$$

where W denotes the reconstruction operator as defined in (7). The resulting residuals capture systematic foreground contamination intrinsic to the modeling assumptions and patch structure.

- Total residual power is computed from the full reconstruction error across noise realizations:

$$C_{\ell}^{\text{res}} = \left\langle C_{\ell} \left(\hat{s}_{\text{CMB}}^{(n)} - s_{\text{CMB}}^{\text{true}} \right) \right\rangle_n, \quad (12)$$

where n indexes the noise realizations and $\hat{s}_{\text{CMB}}^{(n)} \equiv W(d^{(n)})$. This represents the total error power, including residual foregrounds and instrumental noise.

- Statistical residual power, due solely to instrumental noise, is extracted by subtraction:

$$C_{\ell}^{\text{stat}} = C_{\ell}^{\text{res}} - C_{\ell}^{\text{syst}}. \quad (13)$$

This decomposition isolates stochastic uncertainty from systematic bias, and C_{ℓ}^{stat} is used as the effective noise floor for likelihood estimation.

4.0.0.3 BB Spectrum Model. We model the theoretical B -mode spectrum as:

$$C_{\ell}^{\text{BB}}(r) = r \cdot C_{\ell}^{\text{tensor}} + C_{\ell}^{\text{lensing}} + C_{\ell}^{\text{stat}}, \quad (14)$$

Table 1. Validation setup used in the synthetic benchmark.

Parameter	Value / Range
Resolution	$N_{\text{side}} = 64$
Sky mask	GAL020
Noise realizations	100
Patch count grid	$\beta_d \in [10, 300], T_d, \beta_s \in [5, 20]$
True patch count	$\beta_d = 100, T_d = 20, \beta_s = 20$

where C_{ℓ}^{tensor} is a primordial tensor-mode template normalized to $r = 1$, $C_{\ell}^{\text{lensing}}$ is the fixed lensing B -mode contribution, and C_{ℓ}^{stat} accounts for statistical residuals from noise and imperfect foreground cleaning.

4.0.0.4 Likelihood and Inference. The tensor-to-scalar ratio r is inferred by maximizing a chi-squared likelihood function under the assumption of Gaussian-distributed residuals:

$$-2 \ln \mathcal{L}_c(r) = f_{\text{sky}} \sum_{\ell} (2\ell+1) \left[\frac{C_{\ell}^{\text{obs}}}{C_{\ell}^{\text{model}}(r)} + \ln C_{\ell}^{\text{model}}(r) \right] \quad (15)$$

We evaluate this function over a grid of r values and report the best-fit \hat{r} corresponding to the minimum of the negative log-likelihood. Uncertainties are estimated from the curvature of the likelihood around its maximum.

5 RESULTS

5.1 Validation of the Framework

To assess the performance of our clustered component separation framework, we construct a synthetic validation setup in which the true sky components and spectral parameters are fully known. This controlled environment enables precise measurement of reconstruction quality, residual contamination, and estimator consistency.

The simulation begins by defining a known sky signal s in Equation (1), consisting of Stokes Q/U maps for CMB, thermal dust, and synchrotron emission, all specified at a common reference frequency. This sky vector is then mixed using a known mixing matrix $\mathbf{A}(\beta)$, where β encodes the ground-truth spectral parameters for each component. Table 1 summarizes the simulation parameters.

The noise vector \mathbf{n} is sampled as Gaussian white noise, scaled to match LiteBIRD-like polarization depth across the frequency channels. Each clustering configuration is evaluated across 100 independent noise realizations to reduce statistical variance and assess estimator stability.

We perform a grid search over discrete cluster configurations (defined in Section 2.3), varying the number of patches assigned to each spectral parameter. For each setup, we run the full component separation pipeline by minimizing the spectral negative log-likelihood. The resulting CMB map \hat{s}_{CMB} is then compared with the input to assess reconstruction fidelity and the accuracy of tensor-to-scalar ratio estimation.

Figure 4 shows the grid search results. The configuration minimizing the CMB variance coincides with the configuration yielding the lowest spectral negative log-likelihood, indicating that improved physical reconstruction aligns with optimal statistical fit. Importantly, the recovered clustering matches the true simulation setup exactly, validating the robustness of the variance-based selection criterion against noise-driven overfitting.

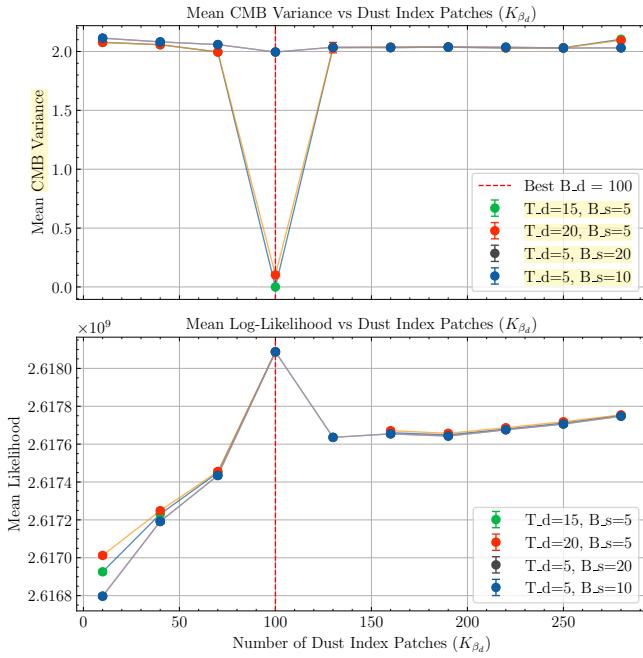


Figure 4. Grid search results for varying the number of dust spectral index patches K_{β_d} , while keeping other parameters fixed. Each curve corresponds to a different configuration of K_{T_d} and K_{B_s} . **Top:** Mean variance of the reconstructed CMB component across noise realizations, used as the primary selection metric. **Bottom:** Mean spectral negative log-likelihood, offset for readability. The vertical red line marks the true value used in the simulated sky ($K_{\beta_d} = 100$), and the dashed lines indicate the global extrema of each metric. The configuration $K_{\beta_d} = 100$, $K_{T_d} = 15$, and $K_{B_s} = 5$ was used to generate the reference clustered map. Minimizing the CMB variance successfully recovers the true clustering configuration.

5.2 Comparison Setup: Grid-Search K-means Clustering vs. Multi-Resolution

We compare two spatial modeling strategies for parametric component separation, both implemented within the FURAX framework:

- **K-means clustering**, which uses spherical K-means to learn spatial patches over which spectral parameters are shared (introduced in Section 2.2).
- **Multi-resolution grouping**, which defines parameter patches by downgrading HEALPix maps to coarser resolutions, following the strategy of (Ahuja 2022) (Errard & Stompor 2019b).

Although applied to the same sky and instrument setup, these approaches differ fundamentally in how they capture spatial variability. The K-means strategy allows a flexible, data-driven exploration of spatial granularity via grid search, while the multi-resolution method imposes predefined patch sizes. This comparison evaluates reconstruction fidelity, residual suppression, and consistency in inferred cosmological parameters.

Simulation Setup

Simulated skies are generated using PySM3 (Group et al. 2025; Zonca et al. 2021; Thorne et al. 2017), combining CMB, thermal dust, and synchrotron emission (c1d1s1 model). Observations are modeled after a LiteBIRD-like instrument, including polarization-sensitive Gaussian noise across multiple frequency bands.

Table 2. Simulation parameters used for both methods.

Parameter	Value
Resolution	$N_{\text{side}} = 64$
Sky model	c1d1s1
Sky mask	GAL020
Instrument	LiteBIRD-like sensitivity
Noise realizations	100 per region

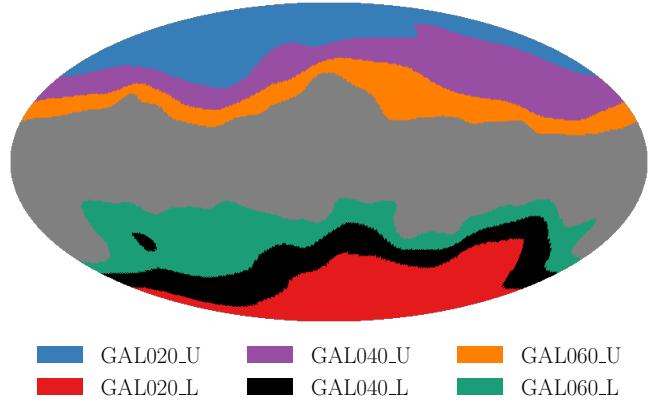


Figure 5. Sky regions used in the comparison study. Colored areas correspond to the six disjoint regions for adaptive clustering. These masks are based on the Planck Galactic masks (Planck Collaboration 2016d).

Table 3. Grid-search clustering configuration.

Parameter	Patch Count Range
β_d	100 to 5000 (in steps of 100)
T_d, β_s	{1, 5, 20, 30, 50, 60, 70, 80}
Sky regions	Six zones (GAL020_U, etc.)
Noise realizations	100 per region

Sky Region Partitioning

To probe performance across different foreground regimes, we define distinct sky zones using Planck-based Galactic masks:

• **K-means clustering** is applied to six disjoint subregions, obtained by splitting each mask into upper and lower hemispheric parts: GAL020_U, GAL020_L, GAL040_U, GAL040_L, GAL060_U, and GAL060_L.

• **Multi-resolution grouping** is applied separately to the full-sky masks: GAL020, GAL040, and GAL060.

These masks span from high Galactic latitudes (low foregrounds, GAL020) to regions closer to the Galactic plane (higher foregrounds, GAL060); see Figure 5.

Grid-Search Clustering

In the adaptive clustering approach, spectral parameters are assigned to pixel clusters defined via spherical K-means. A structured grid search over patch counts is performed, selecting the configuration that minimizes the CMB variance across 100 noise realizations. This fully data-driven search avoids assumptions about spatial scales and allows optimal patch sizes to emerge from the data itself.

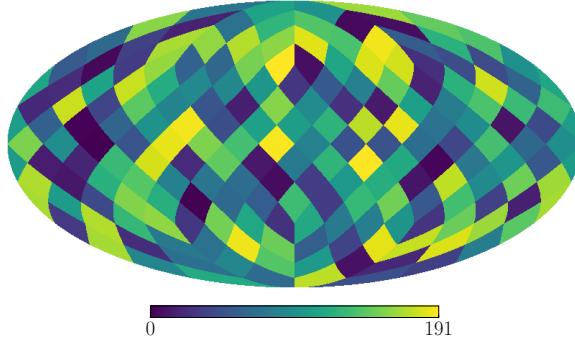


Figure 6. Multi-resolution mega-pixel structure. Example showing downgraded patch regions from $N_{\text{side}} = 64$ to $N_{\text{side}} = 8$.

Table 4. Multi-resolution configuration per mask region (used in the main experiments).

Region (Mask)	β_d	T_d	β_s
GAL060 (low lat.)	64	8	4
GAL040 (mid lat.)	64	4	2
GAL020 (high lat.)	64	0	2

Table 5. Alternative multi-resolution configurations used for validation.

Region (Mask)	β_d	T_d	β_s
GAL060 (low lat.)	64	8	8
GAL040 (mid lat.)	64	4	4
GAL020 (high lat.)	64	0	4

Multi-Resolution Grouping

The multi-resolution grouping method creates spatial patches by degrading the original HEALPix pixelization to coarser N_{side} values, separately for each spectral parameter. After patching, the maps are upsampled back to $N_{\text{side}} = 64$ to match the data grid.

The patching resolutions vary depending on the sky mask and spectral parameter, as summarized in Table 5. These configurations follow the multi-resolution grouping scheme used in (Allys 2022).

While computationally efficient, this fixed-resolution approach lacks the flexibility to adapt to localized foreground complexity, which explains the superior reconstruction performance achieved by adaptive clustering.

5.3 Runtime and Parallelization

The extensive clustering and component separation scans were performed on the *Jean Zay* supercomputer, utilizing NVIDIA A100 GPUs distributed across 4 compute nodes (32 GPUs total). Computations were executed using the `jax-grid-search` framework (Section 2.4), leveraging JAX’s native parallelization capabilities for high-throughput evaluation.

The full study encompassed:

- 3200 clustering configurations,
- each evaluated across 100 independent noise realizations,
- over 6 distinct sky regions (Figure 5),

leading to approximately 1.92 million independent component separation runs.

Despite the large computational volume, the full scan completed

in under 30 hours, consuming approximately 1000 GPU-hours. For comparison, earlier CPU-based frameworks such as FGBuster required around 40 minutes per configuration, making such exhaustive parameter scans practically infeasible.

This performance gain stems from two key design choices: (i) extensive batching and vectorization of likelihood evaluations within each GPU, and (ii) distributed slicing of the grid across multiple devices without assuming divisibility. The implementation supports checkpointing, failure recovery, and fully reproducible execution, ensuring robustness for large HPC deployments.

Thanks to this scalability, FURAX enables a new regime of data-driven model selection, where clustering hypotheses are tested systematically across noise realizations and sky regions, rather than being fixed a priori.

5.4 CMB Reconstruction Comparison

We assess the CMB reconstruction performance achieved by the two spatial modeling strategies introduced in Section 5.2: clustering-based patching via K-means and fixed-resolution multi-resolution grouping. Both methods are applied to the same simulated sky (CMB plus foregrounds) with LiteBIRD-like noise and angular resolution. Beam effects are not modeled in this comparison, as the analysis is restricted to low-resolution simulations with $N_{\text{side}} = 64$.

The analysis focuses on the accuracy of the reconstructed CMB polarization fields, examining both Stokes Q and U components, and on quantifying residual foreground leakage.

Patch Structures

Before presenting reconstruction results, we first illustrate the spatial patch layouts produced by each method. Figures 7 and 8 show the clustering for the three spectral parameters: β_d , T_d , and β_s .

The K-means clustering approach (Figure 7) produces flexible, equal-area patches that adapt to the local sky complexity by minimizing the CMB variance. In contrast, the multi-resolution method (Figure 8) relies on a fixed HEALPix downgrading scheme, imposing uniform patch sizes across large sky regions.

From the results shown in Figure 7, we found that the optimal patch count for our setup was 1500 for both the synchrotron spectral index β_s and the thermal dust temperature T_{dust} , compared to 343 and 144, respectively, in the multi-resolution scheme (Figure 8). The dust spectral index β_d was modeled as a per-pixel parameter in both methods.

These structural differences in patching strongly influence the quality of component separation, as discussed in the following sections.

5.5 Residuals and CMB Reconstruction

We assess the quality of CMB reconstruction achieved by the three spatial modeling strategies: single-patch, multi-resolution, and K-means clustering. The evaluation focuses separately on:

- Residual CMB polarization maps (Q and U)
- Residual B -mode power spectra

We first examine the residual maps, defined as the difference between the reconstructed and true CMB polarization fields for Stokes Q and U .

Figure 9 compares the results across methods.

The multi-resolution approach achieves the visually cleanest maps,

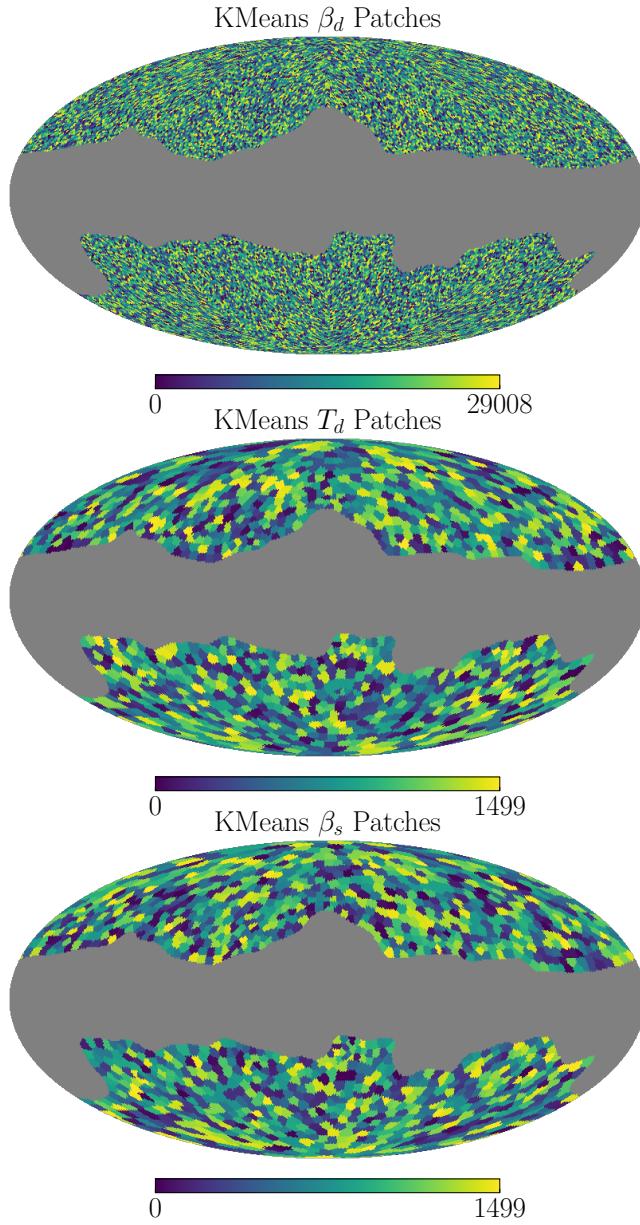


Figure 7. Final FURAX clustering configuration. Top to bottom: patches for β_d , T_d , and β_s , obtained by grid search minimizing the CMB reconstruction variance. The clusters are irregular and data-driven, adapting to spatial variations in the foreground complexity.

with minimal foreground residuals across the sky. K-means clustering yields slightly larger residuals but still substantially improves over the single-patch baseline, which shows significant large-scale errors. These results illustrate the benefits of spatially adaptive modeling; however, as we will discuss in the next section, lower visual residuals do not necessarily imply better statistical performance at the power spectrum level.

Residual B-Mode Power Spectra

We now decompose the residuals into systematic and statistical contributions at the level of the B -mode angular power spectrum.

Figure 10 shows the residual spectra across all multipoles. To contextualize the scale of residual contamination, we also plot the

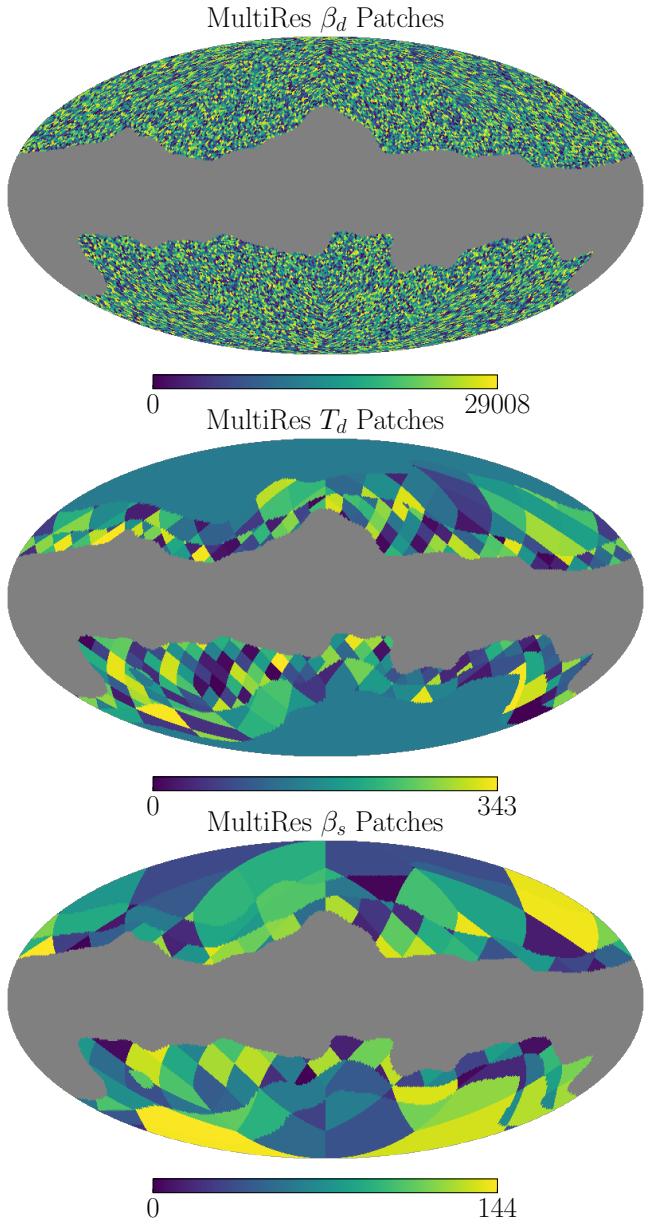


Figure 8. Multi-resolution patch structure. Top to bottom: groupings for β_d , T_d , and β_s , generated by downgrading HEALPix maps to fixed N_{side} resolutions. The patches are regular and uniform, independent of the true sky variability. This corresponds to table 5.

theoretical B -mode power spectra from lensing and primordial tensor modes ($r = 1$), computed using the CAMB python package (Lewis et al. 2000).

This confirms that minimizing the CMB variance through adaptive clustering results in a globally improved reconstruction, even if the residual maps appear slightly noisier compared to multi-resolution grouping.

5.6 Tensor-to-Scalar Ratio Estimation

We now assess the impact of residual foreground contamination on the estimation of the tensor-to-scalar ratio r . For each spatial modeling strategy—K-means clustering, multi-resolution grouping, and

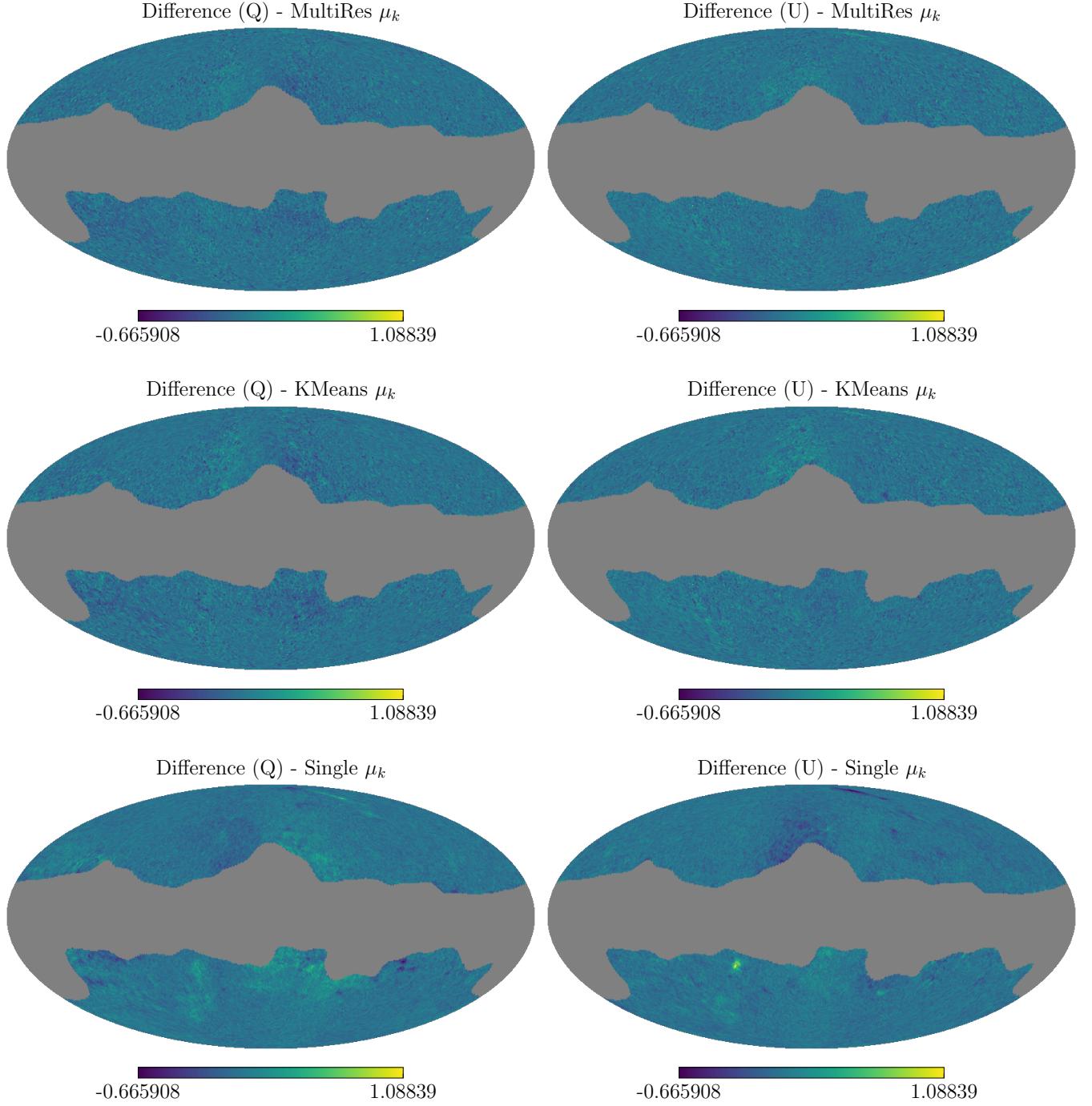


Figure 9. Residual CMB polarization maps (Stokes Q and U) for single-patch, multi-resolution 5, and K-means clustering methods. While multi-resolution yields the lowest mean squared error (MSE)—with $MSE_Q = 6.7 \times 10^{-4} \mu_k$, $MSE_U = 4.6 \times 10^{-4} \mu_k$, K-means performs only slightly worse ($MSE_Q = 8.0 \times 10^{-4} \mu_k$, $MSE_U = 5.5 \times 10^{-4} \mu_k$), but substantially outperforms the single-patch model. The small sacrifice in pixel-level reconstruction accuracy is offset by improved control of residual foregrounds—consequently giving a less biased estimation oftensor-to-scalar ratio r .

single global patch—we compute the r -likelihood using the reconstructed B -mode power spectra.

Figure 12 shows the resulting normalized likelihood distributions aggregated over sky regions and noise realizations.

The K-means clustering approach yields the most accurate and precise measurement, with a likelihood distribution narrowly centered around the true r value. The multi-resolution method exhibits a slight positive bias and broader dispersion, consistent with increased

statistical noise in the reconstruction. In contrast, the single-patch model produces a deceptively tight constraint but a significant bias, strongly overestimating r due to residual systematic contamination. This behavior reflects the dominance of coherent residuals, which shift the likelihood peak while artificially reducing the inferred uncertainty. These trends mirror those observed in the residual B -mode spectra and further demonstrate the importance of flexible spatial modeling to achieve unbiased and robust cosmological inference.

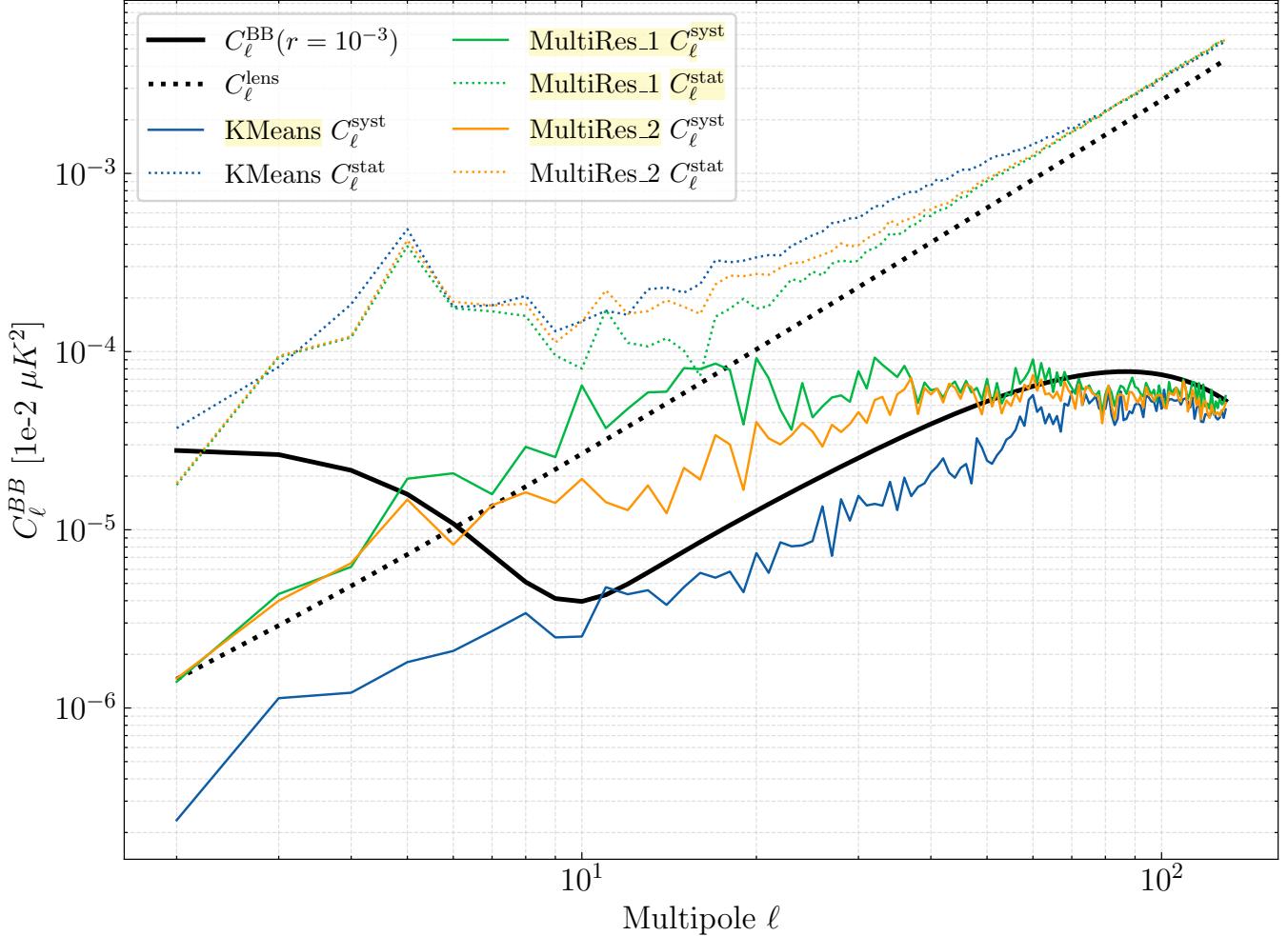


Figure 10. Residual B -mode power spectra for single-patch, multi-resolution, and K-means clustering methods. The MultiRes-1 case (*cf.* Table 5, Fig. 8) illustrates the limited adaptability of pixel downgrading: moving from $N_{\text{side}} = 4$ to $N_{\text{side}} = 8$ for example entails a large, discrete increase in patch count, preventing intermediate configurations. This rigidity leads to overshooting, especially when increasing the patch granularity for T_d and β_s MultiRes-2 (Fig. 1). In contrast, K-means clustering (Fig. 7) enables fine-grained, data-driven tuning of spatial complexity, yielding lower residuals by balancing systematic leakage and statistical variance.

5.6.0.1 Summary of Results.

- **K-means clustering** achieves the tightest and most unbiased constraints on r .
- **Multi-resolution grouping** shows slight bias and wider uncertainty due to statistical noise amplification.
- **Single global patch** yields a biased estimate of r despite a narrow likelihood, driven by uncorrected systematic residuals.

5.7 Impact of Noise on Clustered Component Separation

An important advantage of our approach lies in its adaptability to noise realization variability. In a multi-resolution approach, the spatial patch structure—defined by a fixed HEALPix downgrading—is predetermined and remains constant across noise realizations. This rigidity prevents the model from adapting to noise-induced fluctuations in the data.

In contrast, our K-means clustering strategy combined with a grid search allows the patch configuration to be dynamically re-optimized for each noise realization. For any given realization, the clustering grid search evaluates multiple spatial partitionings and selects the one

that minimizes the variance of the recovered CMB map. This flexibility enables the model to better mitigate noise-driven distortions in the spectral parameter estimates, particularly in low signal-to-noise regions.

As a result, the adaptive clustering approach can reduce residual foreground contamination and enhance the fidelity of component separation by effectively tailoring the patch structure to each specific noise scenario. This noise-aware reconfiguration leads to lower residual power and improves robustness in downstream cosmological inference.



6 DISCUSSION

This work introduces a high-resolution, grid-based clustering approach for spatially varying parametric component separation, implemented in the modular and scalable FURAX framework. Our objective was to evaluate whether learned spectral patch structures can offer advantages over fixed-resolution strategies, using an exhaustive scan over clustering configurations compared against a baseline multi-resolution method.

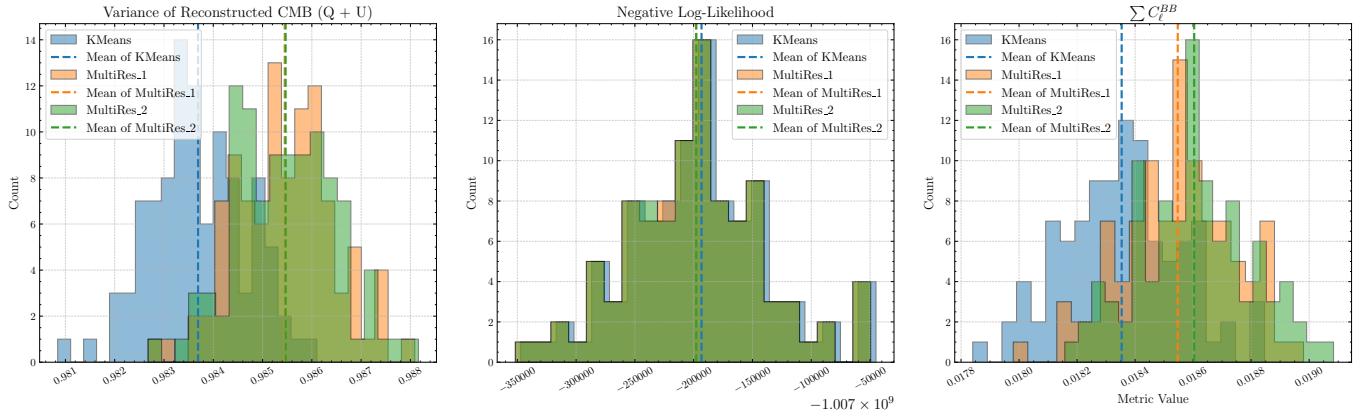


Figure 11. Distribution of evaluation metrics for three spatial modeling strategies: K-means clustering (see Fig. 7), MultiRes-1 (Table 5, Fig. 8), and MultiRes-2 (Table 1, shown in green). Metrics shown are: W variance of the reconstructed CMB ($Q + U$), (center) negative log-likelihood, and (right) total C_ℓ^{BB} power. Dashed vertical lines indicate the mean of each distribution. K-means clustering yields the lowest residual variance and highest likelihood. Multi-resolution approaches show broader distributions, reflecting higher statistical variability.

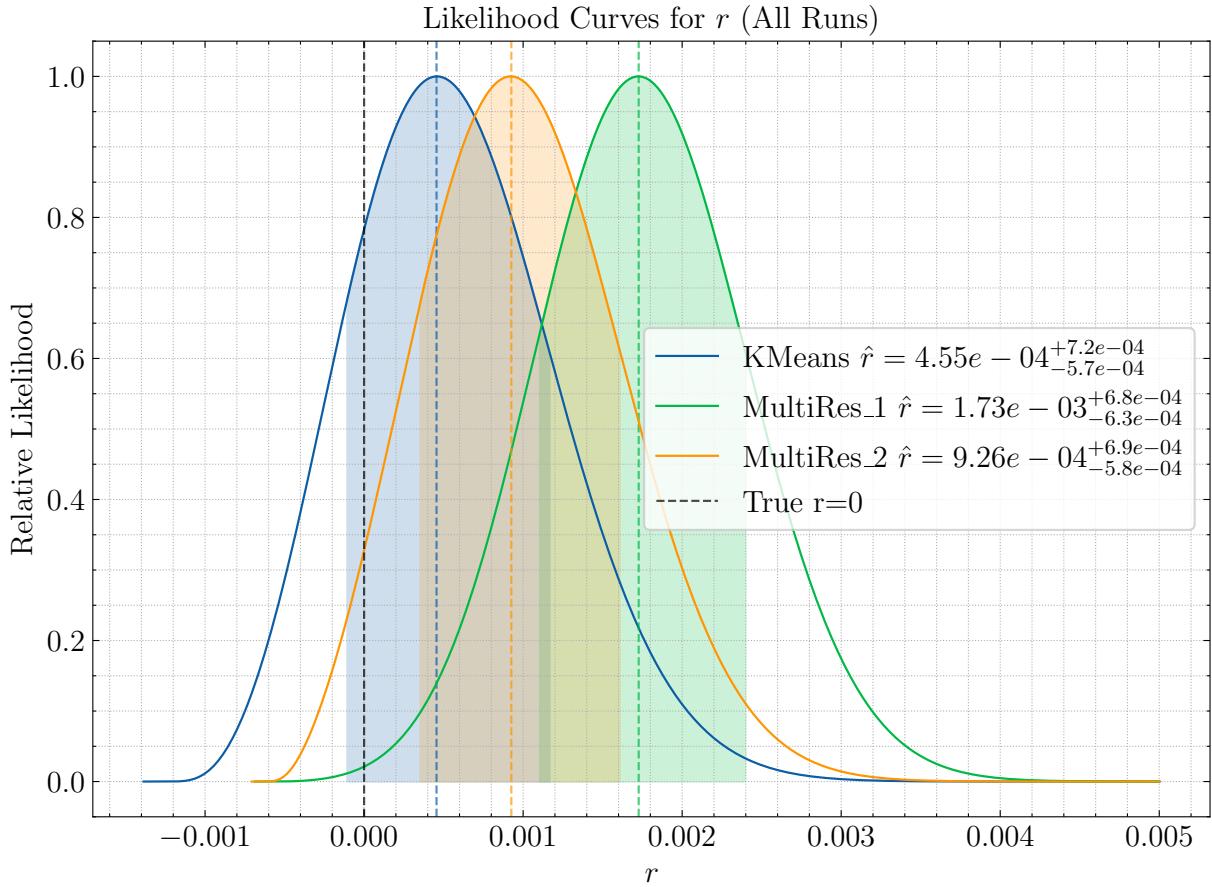


Figure 12. Estimated tensor-to-scalar ratio likelihoods for K-means clustering (blue; see Fig. 7), multi-resolution grouping with configurations MultiRes-1 (green; Table 5, Fig. 8) and MultiRes-2 (orange; Table 1), compared to the true value $r = 0$. K-means clustering yields the lowest bias ($\hat{r} = 4.55 \times 10^{-4}$) and tightest credible interval, while MultiRes-1 and MultiRes-2 exhibit increasing bias and broader uncertainty with finer granularity.

Foreground Modeling at Scale

Rather than fixing patch scales by hand, as was done in (Allys 2022), we perform an exhaustive grid search over the number of patches per spectral parameter. This enables data-driven selection of spatial granularity, allowing each sky region to adopt the level of modeling

complexity most appropriate to its foreground structure, as quantified by the variance in the recovered CMB map.

The resulting clusters, generated using spherical K-means, are highly flexible: patches are irregular, non-contiguous, and vary in size, adapting to the statistical structure of the input data. This avoids

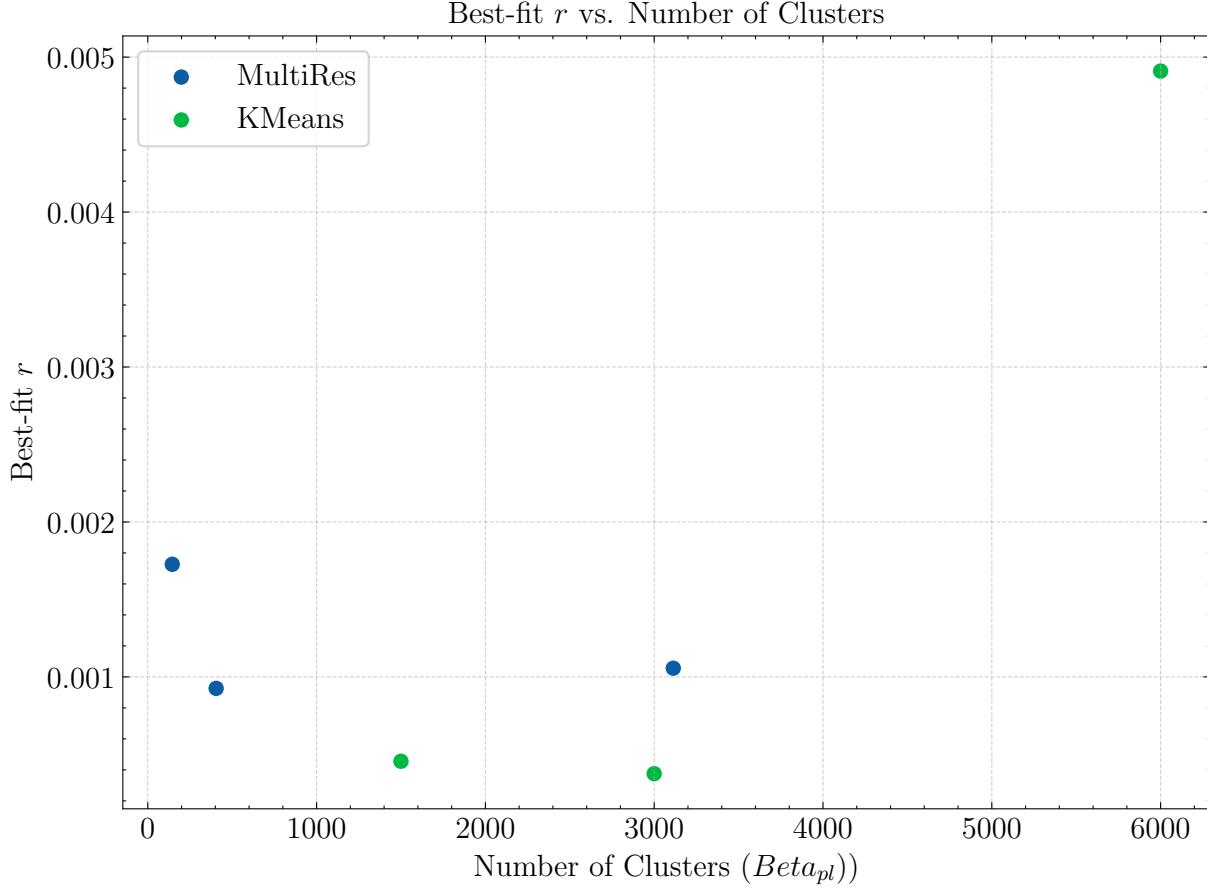


Figure 13. Best-fit tensor-to-scalar ratio \hat{r} as a function of the number of clusters used for modeling the dust spectral index β_d , comparing K-means clustering (green) and multi-resolution grouping (blue). K-means supports finer-grained control over spatial complexity, enabling exploration of a broader and more continuous range of cluster counts. In contrast, the multi-resolution approach is constrained by discrete HEALPix resolutions, resulting in coarser sampling and larger gaps between viable configurations. This flexibility allows K-means to achieve lower bias in \hat{r} , particularly at intermediate clustering scales.

arbitrary binning and allows for more precise foreground modeling, particularly in regions with sharp spectral gradients or complex emission.

In contrast, the multi-resolution approach uses HEALPix downgrading to define fixed NSIDE patch maps. While fast and easy to implement, it lacks the capacity to adapt to regional sky complexity and yields higher residuals in zones with rapidly varying spectra. This difference highlights the value of learned patch structures in a data-driven way.



Computational Scalability and Pipeline Design

To support this large-scale scan—over 2 million component separation evaluations across 6 sky zones and 100 noise realizations per region—we leveraged distributed GPU parallelism using `jax-grid-search`. The entire sweep was completed in under 30 hours on 32 A100 GPUs across 4 nodes on the Jean Zay supercomputer.

Previous pipelines, including those used in [Allys \(2022\)](#), could not scale to this volume: a single multi-patch likelihood fit could take up to 40 minutes, making exhaustive searches prohibitive. FURAX brings this class of analysis within reach, enabling principled exploration of spatial model structures at scale.

Reproducibility and Open Tools

All results in this paper were produced using open-source software developed or co-developed by the author, including the FURAX framework and `jax-grid-search`. The full codebase used to generate the simulations, run the experiments, and produce the plots is publicly available at:

<https://github.com/ASKabalan/furax-compsep-paper>

This ensures transparent and reproducible research, allowing others to easily re-run, extend, or adapt the pipeline for future use cases.

Limitations and Future Work

While effective, our current strategy is based on a brute-force grid scan. This could be made more efficient using modern hyperparameter optimization techniques such as Tree-structured Parzen Estimators or Bayesian search.

Other directions for future work include:

- Scaling to higher resolutions (e.g., $N_{\text{side}} = 128$ or 256)
- Adding priors on patch structure (e.g., smoothness or contiguity)
- Exploring disjoint and irregular clustering with parameter sharing
 - Integrating additional systematic effects such as beam asymmetries, gain drift, or correlated noise

- Investigating alternative spatial modeling strategies beyond K-means, such as adaptive averaging of spectrally similar patches

These extensions are naturally supported by the modular FURAX architecture and will be investigated in future work.

Limitations of Variance-Based Selection Criteria

Although the variance of the reconstructed CMB and the total $\sum C_\ell^{BB}$ power are effective proxies for assessing residual contamination in the output CMB maps, these metrics can be sensitive to statistical residuals. As a result, they may sometimes overlook systematic residuals.

In our analysis, we observed that relying solely on variance-based selection occasionally favored configurations with slightly elevated systematic residuals. To address this, we performed an additional manual inspection step: among the top-ranked configurations (with lowest variance), we selected the one that also satisfied a strict threshold on systematic residuals. This refinement ensures robust foreground cleaning and is reflected in the results shown in Figures 11, 10, and 12.

Advanced Search Strategies for Cluster Configuration

The choice of cluster configuration in our framework is better understood as part of a model selection problem, rather than a standard parameter estimation task. It determines the structure of the likelihood model itself—how spectral parameters are spatially grouped and constrained.

We experimented with more advanced search techniques to optimize cluster configuration, including the Optuna (Akiba et al. 2019) framework and discrete sampling strategies like Gibbs sampling. However, these methods showed limited improvements over exhaustive scanning, likely due to the discrete and highly non-smooth nature of the objective landscape.

Future efforts could explore:

- Structured search methods that exploit correlations across sky zones
- Relaxed parameterizations that allow for differentiable optimization

For this work, the grid-based distributed search provided a robust and parallelizable solution, allowing exhaustive exploration of the model space in a tractable amount of time.

7 CONCLUSIONS

This work presents a new implementation of the parametric component separation approach, based on JAX. The new environment, called FURAX provides a modular environment to generalize the formalism to more complex data models, easing the explicit inclusion of e.g. instrumental response. Compared to previous implementation of similar ideas, for example fgbuster, we show that the new implementation is typically a factor X faster on CPUs and a factor Y faster on GPUs.

In addition, the new implementation offers a scalable, flexible framework for a spatially varying characterization of CMB foregrounds, demonstrating the advantages of data-driven spectral patch selection through exhaustive clustering scans.

Compared to fixed-resolution baselines which were previously

fine-tuned by hand, looking directly at the (in principle inaccessible) residuals, our approach yields:

- Lower residual contamination in the reconstructed CMB maps
- Reduced bias and tighter constraints on the tensor-to-scalar ratio r
- Greater adaptability to complex Galactic foreground structures

By optimizing spectral patching using a physical criterion—the variance of the recovered CMB—we show that adaptive clustering improves both reconstruction fidelity and cosmological parameter estimation.

The FURAX framework, fully open-source and scalable to millions of likelihood evaluations, provides a reproducible foundation for future high-fidelity CMB experiments targeting primordial B -mode polarization.

ACKNOWLEDGEMENTS

The authors would like to thank Tran Hoang Viet, Arianna Rizzieri, Clément Leloup and Radek Stompor for useful exchanges during the development of this work.

This work was supported by the SciPol project (scipol.in2p3.fr), funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Principal Investigator: Josquin Errard, Grant agreement No. 101044073). Some of the authors also benefited from the European Union’s Horizon 2020 research and innovation program under grant agreement no. 101007633 CMB-Inflate.

The computations were performed using HPC resources provided by the *Jean Zay* supercomputer at IDRIS under allocation 2024-AD010414161R2 granted by GENCI.

REFERENCES

- Akiba T., Sano S., Yanase T., Ohta T., Koyama M., 2019, in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2623–2631, doi:[10.1145/3292500.3330701](https://doi.org/10.1145/3292500.3330701)
- Allys e. a., 2022, *Progress of Theoretical and Experimental Physics*, 2023
- Bingham E., Chen J. P., Jankowiak M., Obermeyer F., Pradhan N., Karaletsos T., Kingma D. P., Tran D., 2019, NumPyro: A Lightweight Library for Probabilistic Programming in JAX, <https://github.com/pyro-ppl/numpyro>
- Blondel R., Blondel M., et al., 2021, jaxopt: optimization library in JAX, <https://github.com/google/jaxopt>
- Bradbury J., et al., 2018, JAX: composable transformations of Python+NumPy programs
- Cardoso J.-F. e. a., 2008, *IEEE J. Sel. Top. Signal Process.*, 2, 735
- Chania P., the CMBSciPol team 2024, FURAX: Flexible Unified Regression Analysis for Component Separation, <https://github.com/CMBSciPol/furax>
- Chania P., Biquard S., Kabalan W., , jax-healpy: Implementation of HEALPix related functions and extensions in JAX, <https://github.com/CMBSciPol/healpy>, <https://github.com/CMBSciPol/jax-healpy>
- DeepMind 2020, Optax: Gradient Processing and Optimization Library in JAX, <https://github.com/deepmind/optax>
- Delabrouille J. e. a., 2003, *MNRAS*, 346, 1089
- Eriksen H. K. e. a., 2008, *ApJ*, 676, 10
- Errard J., 2021, FGIBuster: A Flexible Parametric Component Separation Tool
- Errard J., Stompor R., 2015a, *Physical Review D*, 92, 063005
- Errard J., Stompor R., 2015b, *Phys. Rev. D*, 92, 063005
- Errard J., Stompor R., 2019a, *Physical Review D*, 99
- Errard J., Stompor R., 2019b, *Physical Review D*, 99

- Golub G. H., Reinsch C., 1970, Singular Value Decomposition and Least Squares Solutions, doi:10.1007/BF02163027
- Group P.-E. G. S., et al., 2025
- Hazumi M. e. a., 2022, *Progress of Theoretical and Experimental Physics*, 2022, 04A102
- Kamionkowski M., Kosowsky A., Stebbins A., 1997, *Phys. Rev. Lett.*, 78, 2058
- Kingma D. P., Ba J., 2017, Adam: A Method for Stochastic Optimization (arXiv:1412.6980), <https://arxiv.org/abs/1412.6980>
- Lewis A., Challinor A., Lasenby A., 2000, *The Astrophysical Journal*, 538, 473–476
- Meisner A., Finkbeiner D., 2015, *Astrophys. J.*, 798, 88
- Planck Collaboration 2016a, *Astron. Astrophys.*, 594, A9
- Planck Collaboration 2016b, *Astron. Astrophys.*, 594, A10
- Planck Collaboration 2016c, *A&A*, 594, A10
- Planck Collaboration 2016d, *Astronomy & Astrophysics*, 594, A12
- Rader J., Lyons T., Kidger P., 2023, Lineax: unified linear solves and linear least-squares in JAX and Equinox (arXiv:2311.17283), <https://arxiv.org/abs/2311.17283>
- Seljak U., Zaldarriaga M., 1997, *Phys. Rev. Lett.*, 78, 2054
- Sheldon E., 2018, kmeans_radec: KMeans clustering on the sphere, https://github.com/esheldon/kmeans_radec
- Stompor R., Errard J., 2018, *Phys. Rev. D*, 97, 083006
- Stompor R., Leach S., Stivoli F., Baccigalupi C., 2009, *Monthly Notices of the Royal Astronomical Society*, 392, 216–232
- Stompor R., Errard J., Poletti D., 2016, *Physical Review D*, 94
- Thorne B., Dunkley J., Alonso D., Næss S., 2017, *Monthly Notices of the Royal Astronomical Society*, 469, 2821–2833
- Virtanen P., Gommers R., Oliphant T. E., et al., 2020, *Nature Methods*, 17, 261
- Waltz R. A., Morales J. L., Nocedal J., Orban D., 2006, An Interior Algorithm for Nonlinear Optimization that Combines Line Search and Trust Region Steps, Technical Report, Northwestern University
- Wassim K., 2025, JAX Distributed Grid Search for Hyperparameter Tuning, <https://github.com/ASKabalan/jax-grid-search>
- Zonca A., Thorne B., Krachmalnicoff N., Borrill J., 2021, *Journal of Open Source Software*, 6, 3783