A pixel-based Neural Network method for CMB component separation in polarization

Valentina Franco Velásquez[1,2] Biuse Casaponsa[3] R. Belén Barreiro[3] José Manuel Casas[1,2] Laura Bonavera[1,2]

[1] Universidad de Oviedo [2] Instituto de Ciencias y Tecnologías Espaciales de Asturias (ICTEA) [3] Instituto de Física de Cantabria (IFCA, UC - CSIC)

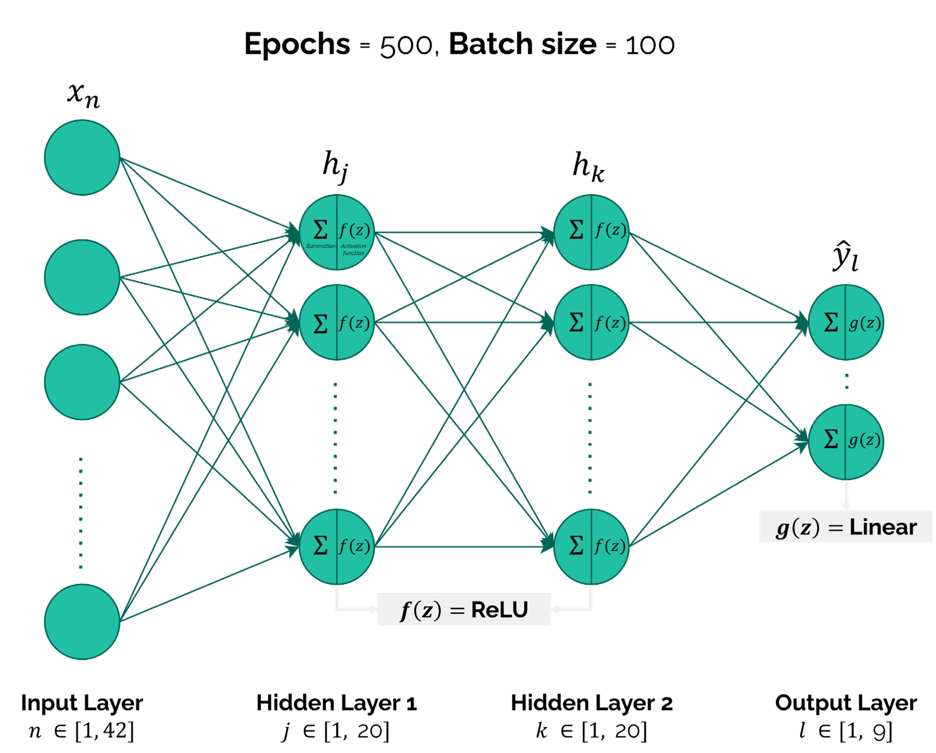
1. Introduction

Detecting **primordial B-mode** polarization in the **CMB** is a major goal in modern cosmology, as it would provide evidence for an inflationary period. This requires accurate component separation to disentangle the cosmological signal from astrophysical foregrounds and instrumental noise. In this work, we present a **pixel-based neural network** method to address this challenge, independent of the underlying sky geometry and thus suitable for direct application on spherical data.

3. The Neural Network (NN)

Our purpose: To employ a **Feedforward Neural Network** (FNN) for component separation using a pixel-based approach.

Basic architecture of the NN:



PICO satellite: The data simulated for training and validating the NN are based on the PICO satellite, a study funded by NASA featuring a highly sensitive imaging polarimeter designed to scan the sky for 5 years in 21 frequency bands. The key specifications used in this work are summarized in Table 1.

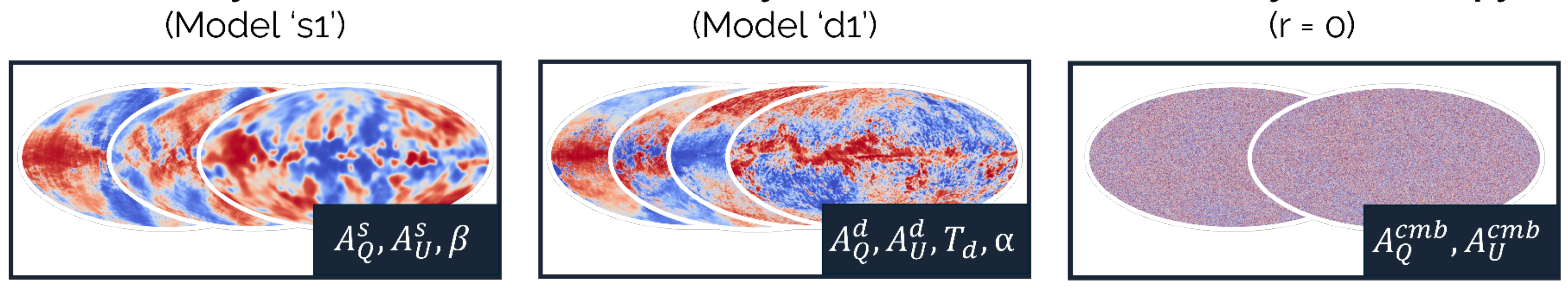
Labels

1.

PySM (Model 's1') — A_Q^s, A_U^s, β

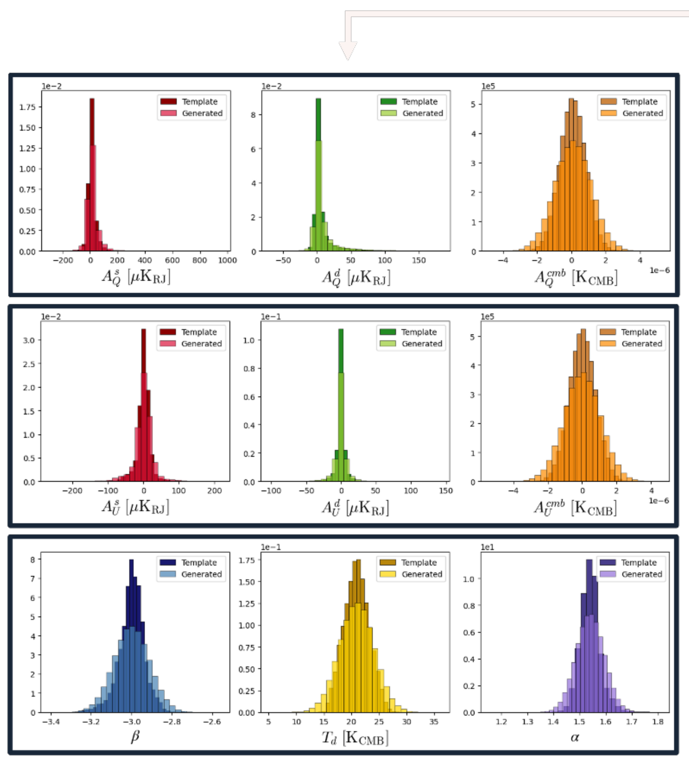
PySM (Model 'd1') — $A_Q^d, A_U^d, T_d, \alpha$

CAMB + Synfast, Healpy (r = 0) — A_Q^{cmb}, A_U^{cmb}



2. Smoothing by **38.4** arcmin (Worst PICO resolution)

3.



786432 (NSIDE = 256) pixels for each free parameter

Generated using numpy **random.choice** method

The **ranges** were increased a **40%** in relation to the smoothed templates

Generated using numpy **random.normal** method

Final Label

$$\hat{y}_{m,l} = \begin{bmatrix} \hat{y}_{1,1} & \cdots & \hat{y}_{1,l} \\ \vdots & \ddots & \vdots \\ \hat{y}_{m,1} & \cdots & \hat{y}_{m,l} \end{bmatrix}$$

m = 786432 (*Total pixels*)
l = 9 (*Free parameters*)

Training set

Equations [1] + **Labels**

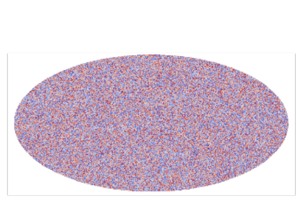
$$S_{pixel,\nu}^T = S_{pixel,\nu}^s + S_{pixel,\nu}^d + S_{pixel,\nu}^{cmb} + S_{pixel,\nu}^n =$$

n features, **m examples**

$$\begin{bmatrix} S_{1,1}^T & \cdots & S_{1,n}^T \\ \vdots & \ddots & \vdots \\ S_{m,1}^T & \cdots & S_{m,n}^T \end{bmatrix}$$

m = 786432 (*Total pixels*)
n = 42 (*21 PICO frequencies in Q and U*)

Final Training set



Gaussian Noise of the instrument

2. The Microwave Sky Model

The diffuse polarized microwave sky signal (S_ν^T) at frequency ν can be modeled as the sum of three components: synchrotron (S_ν^s), thermal dust (S_ν^d), and CMB (S_ν^{cmb}), whose frequency scalings are:

$$S_\nu^s = A_S^s\left(\frac{\nu}{\nu_s}\right)^\beta; \quad S_\nu^d = A_S^d\left(\frac{\nu}{\nu_d}\right)^{\alpha+1}\left(\frac{e^{h\nu_d/(k_BT_d)}-1}{e^{h\nu/(k_BT_d)}-1}\right); \quad S_\nu^{cmb} = A_S^{cmb}\left(\frac{x^2e^x}{(e^x-1)^2}\right). \quad \text{[Eq. 1]}$$

Here S is Stokes Q or U. The amplitudes $A_S^{x=s,d}$, in brightness temperature, are defined at reference frequencies ν_s = 23 GHz and ν_d = 353 GHz, and A_S^{cmb} is in thermodynamic temperature. The spectral indices β and α are assumed to be equal for Q and U, T_d is the dust temperature, and $x = \frac{h\nu}{K_BT_{cmb}}$, with T_{cmb} = 2.7255 K. Overall, the model contains 9 free parameters:

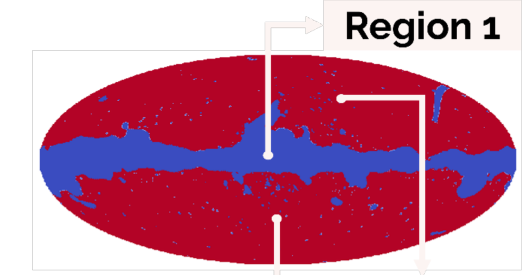
$$A_Q^s, A_Q^d, A_Q^{cmb}, \beta, T_d, \alpha, A_U^s, A_U^d, A_U^{cmb}$$

Table 1. PICO specifications

Frequency [GHz]	21	25	30	36	43	52	62	75	90	108	129	155	186	223	268	321	385	462	555	666	799
FWHM [arcmin]	38.4	32.0	28.3	23.6	22.2	18.4	12.8	10.7	9.5	7.9	7.4	6.2	4.3	3.6	3.2	2.6	2.5	2.1	1.5	1.3	1.1
Instrumental noise [μK_{CMB} · arcmin]	23.9	18.4	12.4	7.9	7.9	5.7	5.4	4.2	2.8	2.3	2.1	1.8	4.0	4.5	3.1	4.2	4.5	9.1	45.8	177	1050

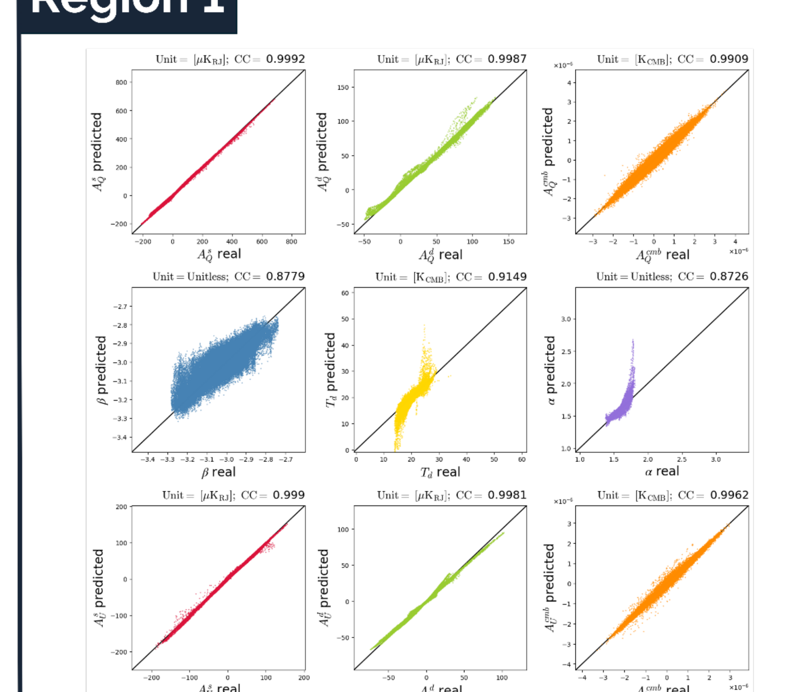
4. Results

To evaluate the **performance** of the NN, we use as input the matrix $S_{pixel,\nu}^T$, constructed from the smoothed original templates, and divide the analisis into the **two regions** of the mask shown.

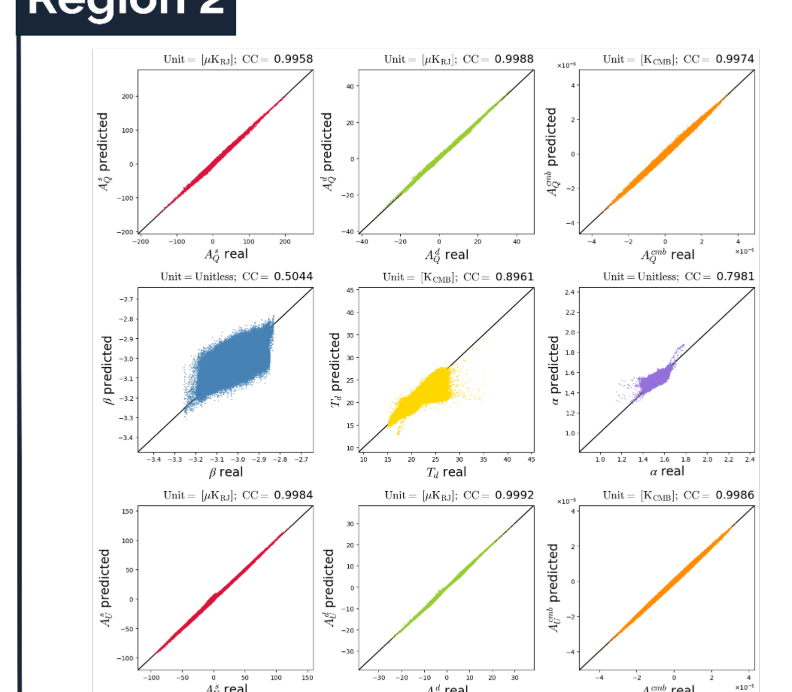


Correlation plots

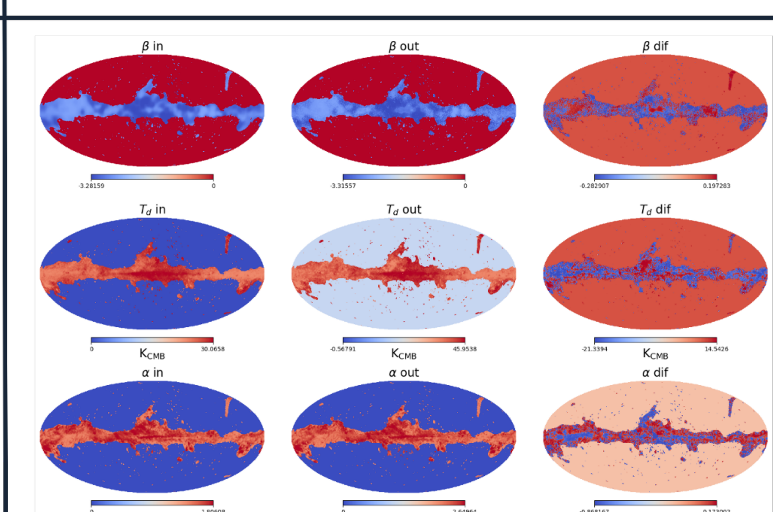
Region 1

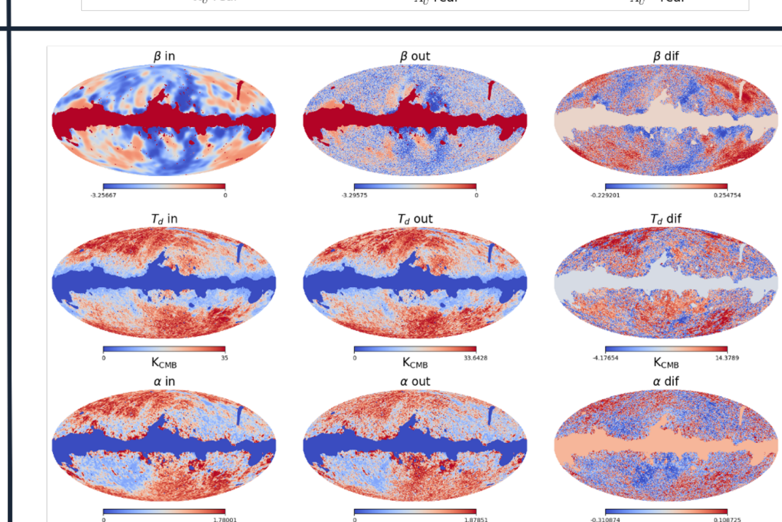


Region 2



Input, output and difference parameters





Statistical Measures

Region 1:

Map	A_Q^s	A_Q^d	A_Q^{cmb}	β	T_d	α	A_U^s	A_U^d	A_U^{cmb}
Error [%]	4,20	5,19	13,53	1,44	4,76	1,67	4,50	6,23	8,71

Region 2:

Map	A_Q^s	A_Q^d	A_Q^{cmb}	β	T_d	α	A_U^s	A_U^d	A_U^{cmb}
Error [%]	9,13	4,92	7,25	1,64	4,05	1,28	5,65	4,59	5,32

5. Conclusions

- The results obtained are very **promising**, given the simplicity of the current NN. It will be interesting to test its performance with **additional layers** and/or neurons.
- Further **improvements** may be obtained by including information from **low-frequency surveys**.
- As next steps, we will **train** the NN separately in **regions 1 and 2**, since their signal behavior can differ significantly.
- **Spatial information** could be included in the algorithm by providing the NN with pixel values at multiple resolutions.
- Finally, we plan to assess the **robustness** of our method by testing it with other foreground models.

Ph.D. Student: Valentina Franco Velásquez; Email: francovalentina@uniovi.es

Refs