

Facultad
de
Ciencias

**Aplicación de Machine Learning a
datos de la radiación del Fondo
Cósmico de Microondas**

Application of Machine Learning to Cosmic Microwave
Background radiation data

Trabajo de Fin de Grado
para acceder al

GRADO EN FÍSICA

Autor: Guillermo Barroso Vega

Director: Marcos Cruz Rodríguez

Codirector: Enrique Martínez González

Fecha: 13 de junio de 2025

Dedicatoria: Me gustaría agradecer este proyecto a mis directores del TFG Marcos y Enrique por transmitirme tantos conocimientos en estos pocos meses. A mis compañeros de despacho que me trataron como si estuviera en casa desde el primer día, especialmente a mi compañero de trabajo y de clase y amiga Sara, con la que he compartido toda la experiencia de realizar las prácticas externas y el TFG en el grupo de cosmología del IFCA, esta experiencia no hubiera sido lo mismo sin ti. Por último, dar las gracias a mi familia y amigos por brindarme el apoyo necesario para sacar la carrera y este TFG adelante.



Resumen

La sustracción de los *foregrounds* (o emisiones contaminantes) del cielo del Fondo Cósmico de Microondas tiene implicaciones importantes en la obtención del espectro de potencias tanto en temperatura como en polarización. En este trabajo nos centraremos en la polarización, donde el modo B primordial es de especial interés ya que su detección proporcionaría una prueba directa de la teoría de la Inflación Cósmica. En este proyecto, se pretende utilizar una aproximación basada en *machine learning* para realizar esta limpieza. Concretamente, se va a utilizar un modelo basado en redes neuronales convolucionales, recogido en el artículo [1]. Esta red se va a entrenar para tres de los experimentos más importantes del CMB tanto pasados como futuros: Planck, LiteBIRD y CMB-S4.

Este modelo entrenado permite obtener resultados muy prometedores en simulaciones que pueden ayudar en el futuro a realizar el proceso de limpieza. Este proceso, se puede realizar para un *foreground* variable sin necesidad de recalculer los parámetros como en los métodos paramétricos de separación de componentes. En el trabajo mostramos que el conocimiento del modelo de *foregrounds* puede tener gran importancia en el proceso de sustracción.

Palabras clave: CMB, *foregrounds*, cosmología, *machine learning*, CNN.



Abstract

The subtraction of foregrounds (or contaminant emissions) from the Cosmic Microwave Background sky have important implications in obtaining the power spectrum both in temperature and polarization. In this work we will focus on polarization, where the primordial B-mode is of special interest since its detection would provide direct evidence for the theory of Cosmic Inflation. In this project, it is intended to use an approach based on machine learning to perform this subtraction. Specifically, we will use a model based on convolutional neural networks, collected in the paper [1]. This network will be trained for three of the most important experiments of the CMB: Planck, LiteBIRD and CMB-S4.

This trained model allows us to obtain very promising results in realistic simulations that can help in the future to carry out the cleaning process. This process can be performed for a variable foreground without the need to recalculate the parameters as in the parametric component separation models. In this work we show that the knowledge of the foreground model can have great importance in the subtraction process.

Keywords: CMB, foregrounds, cosmology, machine learning, CNN.

Índice

1 El Fondo Cósmico Microondas	1
1.1 Espectro de potencias	2
1.2 Principales factores que afectan a la observación del CMB	4
1.2.1 <i>Foregrounds</i>	4
1.2.2 Ruido Instrumental	6
1.2.3 Efecto lente gravitacional	6
1.3 Métodos de separación de componentes	7
1.4 Experimentos	8
1.5 Polarización	9
1.6 La teoría de la Inflación Cósmica	11
2 Metodología	12
2.1 Simulaciones	12
2.1.1 Simulación del CMB con <i>foregrounds</i>	13
2.1.2 Simulación del ruido	16
2.1.3 Obtención de un mapa plano	16
2.2 Redes neuronales convolucionales	17
2.2.1 Fundamentos	17
2.2.2 Arquitectura	23
2.3 Cálculo del espectro de potencias	25
2.3.1 Comprobación experimental	26

3 Resultados	27
3.1 Test Inicial: CMB-S4	27
3.2 LiteBIRD	32
3.2.1 Utilización de los modelos s1, d1 y a2	33
3.2.2 Utilización de los modelos s5 y d10	39
3.3 Planck	41
4 Conclusiones	45

1

El Fondo Cósmico Microondas

Hace más de 13.800 millones de años, el universo primitivo se encontraba en un estado de densidad y temperatura extremadamente altas, lo que conocemos como el *Big Bang*. A partir de este estado inicial, el universo comenzó a expandirse y enfriarse hasta que, unos 380.000 años después, se dió la época de recombinación, momento en el cual los protones y electrones empezaron a formar átomos neutros dejando un camino libre a los fotones para propagarse hasta nuestros días.

El fondo cósmico microondas¹, son los vestigios de esta radiación que se produjo hace miles de millones de años. Aunque inicialmente se emitió en el ultravioleta por el plasma caliente que componía el universo en su etapa temprana, se ha ido corriendo a las microondas y enfriando debido a la expansión del universo. Este fondo se comporta como un cuerpo negro casi perfecto, homogéneo e isótropo con una temperatura actual de $T_0 = 2.725 \pm 0.001$ K[2].

Sin embargo, esta temperatura tiene pequeñas fluctuaciones del orden de una parte en 100.000, estas anisotropías reflejan variaciones en la densidad y el potencial gravitacional del universo temprano. Asimismo, esta señal está parcialmente polarizada debido a la dispersión de Thomson que genera una polarización lineal, en la época de recombinación y posteriormente en la de reionización. Aunque el CMB pueda estudiarse tanto a través de sus anisotropías en temperatura como en polarización, en este trabajo nos centraremos en la polarización, que proporciona información complementaria y es especialmente relevante en la detección de ondas gravitacionales primordiales.

Debido a que se trata de una radiación procedente del universo temprano, su estudio puede conllevar la obtención de características muy interesantes sobre la etapa temprana, como pueden ser la composición bariónica y de plasma del universo primitivo o probar si la teoría

¹CMB, por sus siglas en inglés.

de la Inflaci3n C3smica es correcta², que dejan su huella en la forma del espectro de potencias. A esto se le suma que el modelo cosmol3gico est3ndar, conocido como Λ CDM (Lambda Cold Dark Matter), basado en la relatividad general de Einstein, ha demostrado ser extraordinariamente exitoso en explicar m3ltiples observaciones cosmol3gicas. Este modelo, incluye dos componentes fundamentales adem3s de la materia ordinaria: la energ3a oscura, representada por la constante cosmol3gica Λ , responsable de la expansi3n acelerada del universo; y la materia oscura fr3a (Cold Dark Matter), de la que s3lo se conoce su interacci3n gravitatoria y que explica, entre otros fen3menos, la formaci3n de estructuras a gran escala en el universo.

1.1. Espectro de potencias

Como se ha comentado anteriormente, el CMB presenta anisotrop3as que tienen su origen en las fluctuaciones c3smicas primordiales que fueron amplificadas durante la 3poca de la Inflaci3n C3smica. Estas anisotrop3as se pueden formular como una perturbaci3n de la temperatura media del CMB debido su peque1o valor, es decir, se podr3a escribir que[3]:

$$T(\hat{n}) = T[1 + \Theta(\hat{n})] \quad (1.1)$$

Siendo Θ la perturbaci3n de temperatura, que solo depende de la direcci3n en el cielo en la que observamos, dada por el vector unitario \hat{n} .

Para analizar c3mo var3an las anisotrop3as seg3n la escala angular, resulta 3til descomponer la se1al en arm3nicos esf3ricos, que son los equivalentes a la transformada de Fourier pero sobre la esfera.

$$\Theta(\hat{n}) = \sum_{\ell=1}^{\ell_{max}} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\hat{n}) \quad (1.2)$$

$Y_{\ell m}$ son los arm3nicos esf3ricos y $a_{\ell m}$ sus coeficientes. Por otra parte, ℓ se refiere a la posici3n a una cierta distancia angular $\theta \approx \pi\ell^{-1}$. Es decir, los ℓ 's bajos representan grandes escalas angulares y, por el contrario, los ℓ altos, representan escalas angulares peque1as. Por 3ltimo, m representa la orientaci3n espacial.

Los coeficientes $a_{\ell m}$ de los arm3nicos esf3ricos se pueden obtener de la siguiente manera:

$$a_{\ell m} = \int Y_{\ell m}^* \Theta(\hat{n}) d\Omega \quad (1.3)$$

En esta ecuaci3n, se identifica a Ω como el 3ngulo s3lido. Esta integral se realiza sobre la esfera unidad, que representa el conjunto de direcciones \hat{n} desde las cuales se observa la radiaci3n procedente del CMB.

Si se aplican a estos $a_{\ell m}$ las propiedades de ortonormalidad de los arm3nicos esf3ricos se obtiene que $\langle a_{\ell m} a_{\ell' m'}^* \rangle = \delta_{\ell\ell'} \delta_{mm'} C_{\ell}$. Obteniendo el espectro angular de potencias en el CMB, C_{ℓ} .

²Esta teor3a se explicar3 con m3s detalle en las p3ginas siguientes.

El modelo cosmol3gico estandar predice que las anisotropias del CMB deben comportarse como un campo aleatorio gaussiano en la esfera celeste. Esta predicci3n es fundamental porque implica que toda la informaci3n estadística de las fluctuaciones primordiales est1 completamente caracterizada por el espectro de potencias C_ℓ . Las observaciones del CMB, especialmente las realizadas por las misiones WMAP y Planck, han confirmado esta hip3tesis de gaussianidad con gran precisi3n, proporcionando as1 un respaldo observacional robusto a las predicciones te3ricas del modelo estandar.

Bajo la hip3tesis de gaussianidad, cada C_ℓ sigue una distribuci3n chi-cuadrado (χ^2). Su desviaci3n estandar, conocida como varianza c3smica, representa el l1mite fundamental en la precisi3n con la que se pueden determinar estos par1metros y est1 dada por la siguiente expresi3n:

$$\Delta C_\ell = \sqrt{\frac{2}{2\ell + 1}} C_\ell \quad (1.4)$$

No obstante, en la pr1ctica, este l1mite no se va a alcanzar completamente, pues la gran emisi3n de *foregrounds* en el plano gal1ctico hace que sea imperativo enmascarar y evita que se alcancen este tipo de situaciones.

An1logamente a como se ha obtenido el espectro de potencias de temperatura, se pueden definir los espectros de potencias de los modos E y B de polarizaci3n que se definen en la secci3n 1.5. Un ejemplo del espectro de potencias que ha de ser obtenido para el CMB se encuentra en la Figura 1.1, en la que se representa el espectro de potencias del modo E.

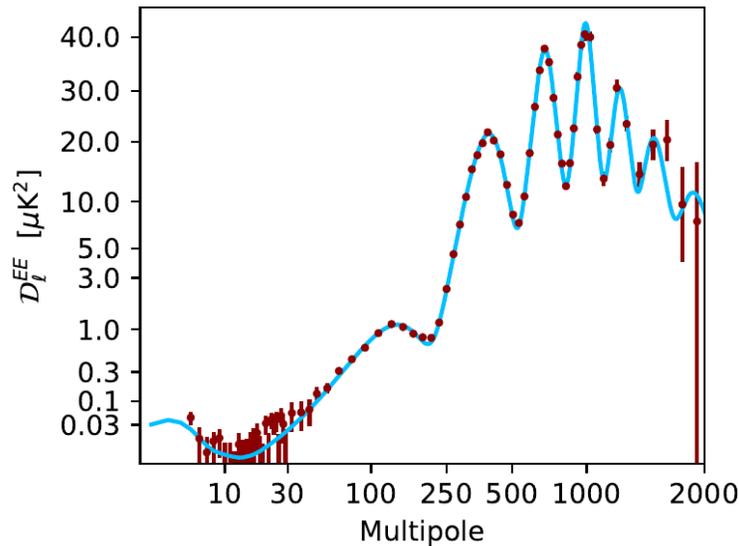


Figura 1.1: Representaci3n gr1fica del espectro de potencias del modo E del CMB observado por Planck. En el eje de ordenadas, se representa el espectro de potencias reescalado mediante la ecuaci3n $D_\ell = C_\ell \ell(\ell + 1)/2\pi$, con respecto al multipolo ℓ . La l1nea azul corresponde con el modelo que mejor se ajusta a los datos. Fuente: <https://www.cosmos.esa.int/web/planck/picture-gallery>.

1.2. Principales factores que afectan a la observación del CMB

Para analizar el CMB, sin embargo, hay que tener en cuenta que hay otras fuentes de emisión importantes de radiación de microondas procedentes de la esfera celeste o del propio instrumento. Además de este ruido sobre el mapa del CMB, existen efectos como el efecto lente gravitatoria que afecta a la geometría de la imagen, la distorsiona, además de convertir los modos E en modos B lo que dificulta la búsqueda de los modos B primordiales. Para obtener un correcto análisis del CMB es necesario intentar sustraer todos estos factores en la medida que sea posible.

1.2.1. *Foregrounds*

El primer factor importante que afecta al CMB es la contaminación por otras fuentes de radiación de microondas que se superponen a la señal primordial. Estas emisiones contaminantes, conocidas como *foregrounds*, provienen principalmente de procesos físicos dentro de nuestra propia galaxia (la Vía Láctea) y, en menor medida, de fuentes extragalácticas. Al observar el CMB, estas señales se suman a la radiación primordial, dificultando su análisis y requiriendo de técnicas especializadas para su separación. Las fuentes fundamentales de emisión de esta radiación microondas son las siguientes: sincrotrón, polvo térmico, emisión microondas anómala³, *free-free* y líneas de emisión CO⁴.

1. **La radiación sincrotrón**, esta radiación proviene de la aceleración de los electrones procedentes de los rayos cósmicos relativistas que pasan por los campos magnéticos galácticos. Constituye la fuente principal de foreground polarizado para frecuencias menores de 50 GHz, aunque entre los 50 GHz y 100 GHz es relevante.

$$Q_s(\hat{n}, \nu) = A_{Q,s\nu_0}(\hat{n}) \left(\frac{\nu}{\nu_0} \right)^{\beta_s(\hat{n})}, \quad U_s(\hat{n}, \nu) = A_{U,s\nu_0}(\hat{n}) \left(\frac{\nu}{\nu_0} \right)^{\beta_s(\hat{n})} \quad (1.5)$$

En esta ecuación, $Q_s(\hat{n}, \nu)$, $U_s(\hat{n}, \nu)$ representan las componentes de polarización sincrotrón de Stokes, $A_{Q,s\nu_0}$ y $A_{U,s\nu_0}$ sus amplitudes a la frecuencia pivote ν_0 y β_s el índice espectral.

2. **La radiación de polvo térmico**, procedente de los granos de polvo estelar que son calentados mediante absorción en el óptico, estos granos se enfrían subsecuentemente emitiendo en la región del infrarrojo cercano. Esta radiación domina a frecuencias mayores que 70 GHz. Sin embargo, entre los 50 GHz y 100 GHz también es relevante compitiendo con, entre otras, la radiación sincrotrón.

³AME, por sus siglas en inglés.

⁴Se dispone de información detallada de estos *foregrounds* en la documentación de PySM3: <https://pysm3.readthedocs.io/en/latest/>

$$Q_d(\hat{n}, \nu) = A_{Q,d\nu_0}(\hat{n}) \left(\frac{\nu}{\nu_0}\right)^{\beta_d(\hat{n})} B(\nu, T_d(\hat{n})), \quad U_d(\hat{n}, \nu) = A_{U,d\nu_0}(\hat{n}) \left(\frac{\nu}{\nu_0}\right)^{\beta_d(\hat{n})} B(\nu, T_d(\hat{n})) \quad (1.6)$$

Donde $Q_d(\hat{n}, \nu)$ y $U_d(\hat{n}, \nu)$ representan las componentes de polarización Stokes de polvo térmico. Las amplitudes a la frecuencia pivote ν_0 son $A_{Q,d\nu_0}$ y $A_{U,d\nu_0}$ mientras que β_d representa el índice de emisividad. Además, $B(\nu, T_d(\hat{n}))$ representa el espectro de cuerpo negro estándar con una temperatura $T_d(\hat{n})$.

3. **La emisión de microondas anómala**, aunque se trata de una hipótesis, se piensa que puede estar basada en la radiación de dipolo eléctrico emitida por pequeños granos de polvo giratorios. Esta radiación ha sido observada por instrumentos que operan en el rango de las microondas dentro del rango de frecuencias entre 10 GHz y 60 GHz.
4. **La emisión *free-free***, proviene del efecto Bremsstrahlung debido a colisiones ión-electrón. Esta radiación domina para un pequeño rango de frecuencias menor de 50 GHz.
5. **Las líneas de emisión CO**, puesto que el monóxido de carbono es una molécula abundante en nubes moleculares, esta radiación proveniente de las transiciones de energía entre niveles rotacionales es importante. Esta emisión domina en frecuencias muy específicas en las que se producen las transiciones.

En este caso, centrándose en la Figura 1.2 se puede observar que las únicas radiaciones interesantes a estudiar en polarización son la radiación sincrotrón y la emisión del polvo térmico. En este proyecto se incluirá también la emisión microondas anómala (AME) ya que, aunque su mecanismo físico exacto aún se encuentra bajo investigación, constituye una fuente significativa de contaminación en el rango de las frecuencias de interés. Estos *foregrounds* han de ser sustraídos y sus residuos tenidos en cuenta junto al ruido experimental con el fin de obtener la radiación del CMB buscada.

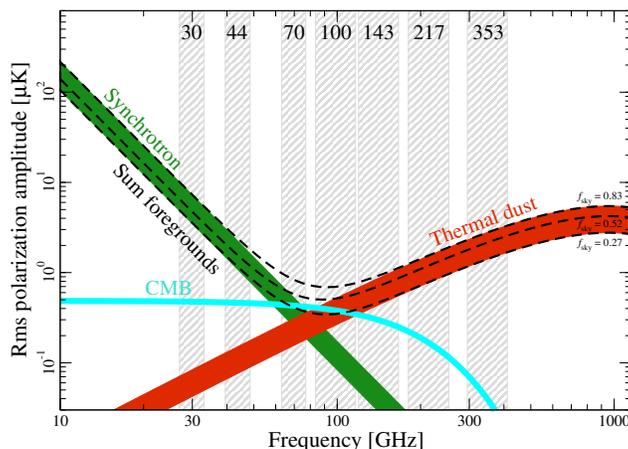


Figura 1.2: Representación gráfica de las contribuciones de los diferentes foregrounds a distintas frecuencias comparadas con la contribución del CMB. Fuente: <https://www.cosmos.esa.int/web/planck/picture-gallery>.

1.2.2. Ruido Instrumental

Los detectores utilizados en las observaciones del CMB no tienen un funcionamiento ideal. Las componentes electrónicas del sistema de detección introducen fluctuaciones aleatorias en la señal medida que no corresponden a la radiación astrofísica real. Este ruido instrumental[4] y [5] proviene de diversas fuentes como fluctuaciones térmicas en los detectores, ruido electrónico en los amplificadores, y variaciones en la respuesta del sistema. Este ruido en el CMB normalmente se modela como un ruido gaussiano blanco.

Este ruido se tendrá en cuenta a la hora de estimar el espectro de potencias del CMB mediante el cálculo del auto-espectro o el cros-espectro. En este proyecto, se utilizará el cros-espectro que tiene la ventaja de que el ruido no sesga los resultados. El cálculo de este espectro se realiza utilizando la aproximación de “*half-split map*”, cuyo funcionamiento se tratará más adelante.

1.2.3. Efecto lente gravitacional

El efecto lente gravitacional consiste en la deflexión de la luz procedente de cualquier fuente de radiación electromagnética (ya sean objetos puntuales como cuásares o radiación difusa como el CMB) debido a la curvatura del espacio-tiempo causada por estructuras masivas como cúmulos de galaxias que se interponen en su trayectoria hacia el observador. Este efecto puede manifestarse tanto a gran escala (por cúmulos de galaxia) como a pequeña escala mediante microlentes causadas por estructuras menores dentro de estos cúmulos[6] y [7]. Este fenómeno se puede observar esquemáticamente en la Figura 1.3.

A medida que estos objetos celestes se van separando de la alineación perfecta, comienzan a aparecer gradualmente dos imágenes del cuásar, situadas simétricamente a ambos lados del eje, como si se reflejaran en un espejo.

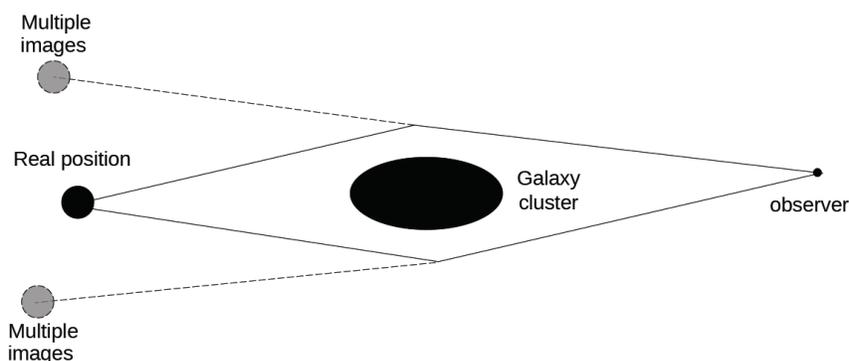


Figura 1.3: Imagen del efecto lente gravitatoria: la luz de un objeto es lensada por un cúmulo de galaxias en su camino a ser observada.

Para fuentes puntuales como cuásares, este efecto puede producir múltiples imágenes, ampliación del brillo, y en caso de alineación perfecta, la formación de anillos de Einstein. Sin embargo, para radiación difusa como el CMB, el efecto principal es una distorsión coherente

de los patrones de anisotropías a gran escala.

En el caso del CMB, este efecto proviene de la interacción de la radiación de microondas con las estructuras masivas del universo a lo largo de su trayectoria hacia el observador. Es decir, de la curvatura del espacio-tiempo generada. La existencia de este efecto tiene varias consecuencias importantes en los mapas del CMB observados[8].

En primer lugar, se produce, como ya se ha comentado con anterioridad, una deformación del mapa del CMB observado, lo que produce que no se tenga una reproducción exacta de las condiciones primigenias, sino un mapa algo distorsionado.

Este proceso también produce un desenfoque de las anisotropías causado por las desviaciones de los fotones de microondas al atravesar las inhomogeneidades en la densidad del universo. Esta concatenación de desviaciones produce un efecto de suavizado sobre las oscilaciones acústicas presentes en el espectro de potencias.

Por último, el efecto lente produce una conversión de parte de los modos E del CMB en modos B, debido a que las pequeñas desviaciones alteran la orientación de los vectores de polarización.

1.3. Métodos de separación de componentes

En este proyecto utilizaremos métodos de inteligencia artificial para sustraer los *foregrounds* del CMB con el fin de obtener una señal limpia. Sin embargo, es importante conocer y describir los métodos tradicionales de separación de componentes que se han desarrollado hasta la fecha[9] y [10], ya que proporcionan el contexto científico y las técnicas de referencia en el campo. Entre los métodos tradicionales más utilizados se encuentran:

1. **Commander**, se trata de la implementación bayesiana en el espacio de píxeles. Para ello, se ajusta un modelo paramétrico a la distribución de los datos a partir del muestreo de la distribución posterior de los parámetros del modelo. Para realizar esta implementación, sigue un procedimiento de dos pasos: primeramente, se encuentran las amplitudes e índices espectrales de *foregrounds* a baja resolución mediante procedimientos de muestreo como el Markov Chain Monte Carlo, para luego recalcular las amplitudes resolviendo un sistema de mínimos cuadrados generalizados para cada pixel, dejando los índices espectrales invariantes.
2. **ILC**, *Internar Linear Combination*, este método consiste en una combinación lineal de mapas del CMB a distintas frecuencias ponderadas con el fin de minimizar la varianza y así reducir la cantidad de *foregrounds* presentes.
3. **NILC**, este método consiste en aplicar el método ILC a cada mapa resultante de aplicar la *wavelet (needlet)* con distintas resoluciones. Esto permite que los pesos se adapten a la variación de las componentes en función de la escala. Este método se utiliza para extraer el CMB o cualquier componente conocida después de su descomposición en el espacio de las *needlets*.

4. **SMICA**, *Spectral Matching Independent Component Analysis*, se trata de m3todo que recompone la se1al del CMB como una combinaci3n lineal en el espacio arm3nico de N_{chan} mapas frecuenciales de entrada con pesos que dependen del multipolo. Este m3todo se basa en una separaci3n de componentes ciega, modelando las se1ales como una combinaci3n lineal de varias fuentes de radiaci3n de microondas.
5. **SEVEM**, para obtener el CMB limpio, se utiliza el m3todo de *template fitting*. Este m3todo crea *templates* de los *foregrounds* a partir de la diferencia de mapas a distintas frecuencias, es decir, crea mapas plantilla de los *foregrounds* para despu3s ajustarlos en cada frecuencia y sustraer su contribuci3n, con el fin de obtener un mapa limpio en la frecuencia correspondiente.

En la Figura 1.4, se puede observar la diferencia entre las limpiezas realizadas por los distintos m3todos de separaci3n de componentes presentados. En esta figura, C-R se refiere al m3todo de separaci3n Commander.

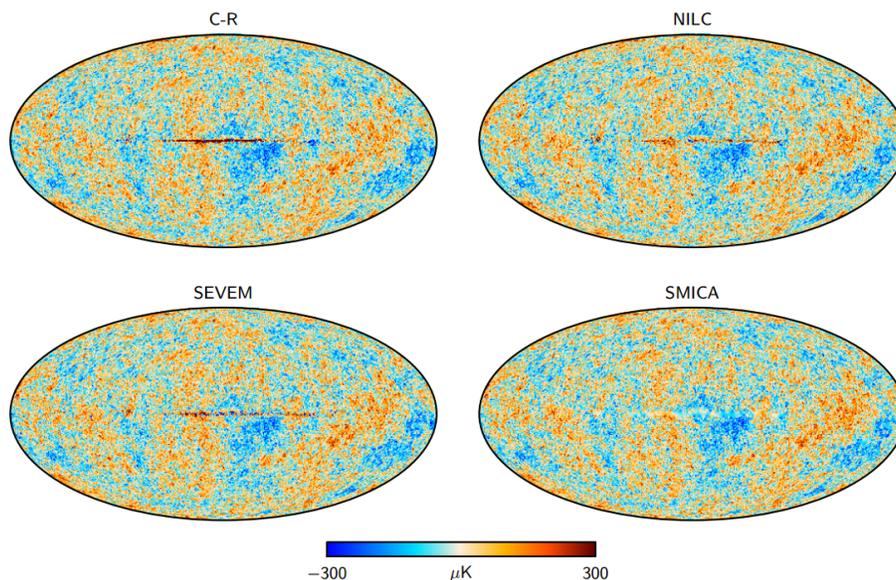


Figura 1.4: Representaci3n de un mapa del CMB al que se han aplicado diferentes modelos de sustracci3n de *foregrounds*. Este mapa corresponde con los resultados de Planck de 2013. Fuente: <https://sci.esa.int/web/services/multimedia-gallery>.

1.4. Experimentos

Diversos experimentos han medido o est3n proyectados para medir el CMB. **Planck**⁵ es un experimento espacial ya finalizado que ha proporcionado las mediciones m3s precisas del Fondo C3smico Microondas hasta la fecha, tanto en temperatura como en polarizaci3n, cubriendo toda la esfera celeste.

⁵<https://pla.esac.esa.int/#home>

Por otra parte, **LiteBIRD**⁶ (experimento espacial) y **CMB-S4**⁷ (experimento terrestre) son proyectos futuros que aún no han tomado datos. Para estos experimentos se realizan simulaciones y predicciones de lo que podrán medir en el futuro, con el objetivo de evaluar su capacidad para detectar señales específicas del CMB.

Al medir el CMB, tanto en temperatura como en polarización, se registran los parámetros de Stokes **T** (temperatura), **Q** y **U** (polarización lineal). Tanto en los datos reales como en las simulaciones, la información se organiza en tres columnas correspondientes a estos tres parámetros. Este proyecto se enfocará en el estudio de los parámetros de polarización **Q** y **U**, explicados con más detalle en la siguiente sección, con el fin de investigar las anisotropías del CMB y evaluar las predicciones de la teoría de la Inflación Cósmica.

En estos mapas, debido a que alrededor del plano galáctico hay un mayor nivel de contaminación, pues se tienen las emisiones en microondas de la Vía Láctea, entre otras cosas, se va a aplicar una máscara binaria que elimina zonas de emisión intensa, mayores que un umbral de $1\mu\text{K}$, con el fin de mejorar la precisión de la sustracción de las emisiones contaminantes en el resto del cielo.

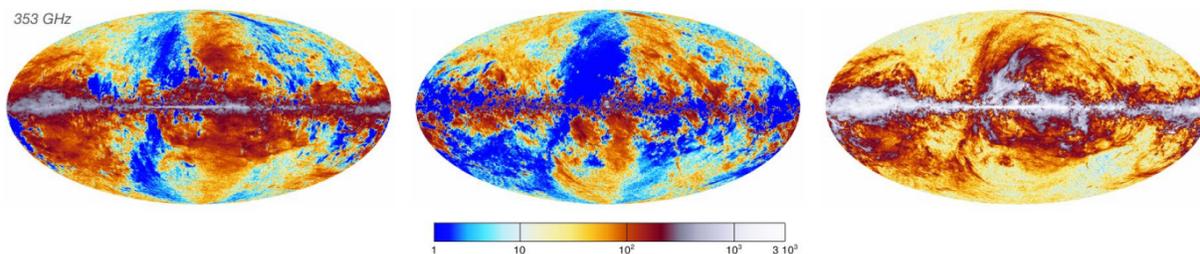


Figura 1.5: Mapas de polarización **Q**, **U** y **P** obtenidos de la misión Planck 2018. Fuente: <https://www.cosmos.esa.int/web/planck/picture-gallery>.

1.5. Polarización

Como se ha explicado anteriormente, se pretende trabajar con los parámetros de Stokes **Q** y **U**, que describen la polarización lineal de la radiación del CMB. Sin embargo, estos parámetros no son independientes del sistema de referencia. Así, se puede observar en la Figura 1.6 izquierda como, dependiendo del sistema de referencias en el que se trabaje, se van a tener distintos valores de **Q** y **U** dentro de la circunferencia. Por ello, aunque los parámetros de Stokes (**Q** y **U**) tengan sentido localmente, no es conveniente trabajar con éstos a nivel de mapa global, aunque sí sea técnicamente posible hacerlo.

Si consideramos el CMB desde la perspectiva de una onda electromagnética[11] propagándose en la dirección **n**, definiendo el vector de polarización como un sistema ortogonal *right-handed* con un campo eléctrico de la forma $\mathbf{E} = E_1\epsilon^{(1)} + E_2\epsilon^{(2)}$, entonces

⁶<https://litebird.isas.jaxa.jp/static/eng/>

⁷<https://cmb-s4.org>

$$Q = |E_1|^2 - |E_2|^2, \quad U = 2\text{Re}(E_1^* E_2) \quad (1.7)$$

Es sencillo observar que Q y U describen la simetría de la parte sin traza del tensor de polarización. Sin embargo, es necesario utilizar una componente que sea invariante, es decir, que no varíe ante rotaciones del sistema de coordenadas. Es de ahí de dónde surgen los modos de polarización E y B.

$$Q(x) = \int \frac{d^2\ell}{(2\pi)^2} (E_\ell \cos(2\phi_\ell) - B_\ell \sin(2\phi_\ell)) e^{i\ell x}$$

$$U(x) = \int \frac{d^2\ell}{(2\pi)^2} (E_\ell \sin(2\phi_\ell) + B_\ell \cos(2\phi_\ell)) e^{i\ell x} \quad (1.8)$$

El ángulo ϕ_ℓ se corresponde con el ángulo polar del multipolo, ℓ , en el plano de Fourier, $\phi_\ell = \tan^{-1}(\ell_y/\ell_x)$, mientras que E_ℓ y B_ℓ son los modos de polarización E y B en ℓ .

Si se utiliza un sistema de coordenadas local en el que uno de sus ejes está alineado con el vector de onda, se obtiene que $Q = E_\ell e^{i\ell x}$ y $U = B_\ell e^{i\ell x}$ con una representación similar a la de la Figura 1.6 derecha.

Con todo ello, es trivial obtener que el modo E tiene una paridad par, es decir, es invariante bajo inversiones espaciales, es de tipo gradiente. Por su parte, el modo B presenta una paridad impar, es decir, cambia de signo ante inversiones espaciales y se le asocia un patrón de tipo rotacional.

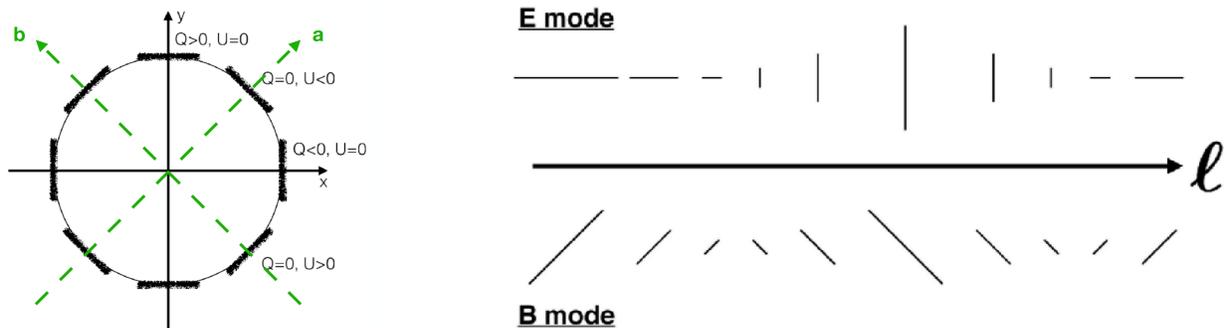


Figura 1.6: Imágenes correspondientes a los diferentes modos de polarización que se dan en el CMB. La primera imagen corresponde a una representación gráfica de los modos Q y U en diferentes sistemas de coordenadas; La segunda imagen representa los modos E y B a lo largo de un eje paralelo al vector de onda. Fuente: Blake Sherwin, 2022. Summer School of Cosmology: CMB talk 4. *ICTP*. Charla disponible en: <https://indico.ictp.it/event/9810/other-view?view=ictptimetable>.

1.6. La teoría de la Inflación Cósmica

La inflación cósmica, se considera una época de expansión acelerada que tuvo lugar después del *Bing Bang* durante un periodo de tiempo extremadamente breve (unos 10^{-32} segundos). Esta teoría se describe actualmente (y principalmente, entre otras muchas propuestas) mediante el deslizamiento lento de un campo escalar conocido como inflatón[3] Tal y como se puede observar en la Figura 1.7, el campo escalar se desliza lentamente hacia su verdadero estado fundamental, el término 'lentamente' es clave, ya que implica que la expansión no termina de manera abrupta.

Según esta teoría, la inflación cósmica es impulsada por un campo escalar cuyo potencial de energía es distinto de cero y prácticamente constante. En estas condiciones, la energía potencial domina rápidamente sobre la energía cinética del campo. Esto da lugar a un proceso de deslizamiento lento hacia el mínimo de potencial. Al alcanzar dicho mínimo, el campo comienza a oscilar y genera partículas más ligeras, finalizando así la fase de la inflación cósmica.

Durante el proceso inflacionario, el inflatón sufre fluctuaciones cuánticas que generan tanto perturbaciones escalares como tensoriales. Estas últimas se asocian al fondo de ondas gravitacionales primigenias, las cuales dejan su huella en los modos B de la polarización del CMB[12].

No obstante, no es tan sencillo como obtener el espectro de potencias del modo B del CMB, ya que, en su camino hacia nosotros, la radiación ha sufrido el efecto lente gravitatoria debido a la estructura a gran escala del universo lo que hace que se generen modos B secundarios a partir de los modos E.

En este proyecto, se va a realizar el primer paso para comprobar si la teoría de la Inflación Cósmica es correcta, obtener los mapas limpios del CMB con el fin de poder reconstruir, de la manera más exacta posible, el espectro de potencias del modo B, aunque en este proyecto no se tendrán en cuenta explícitamente los modos B primigenios.

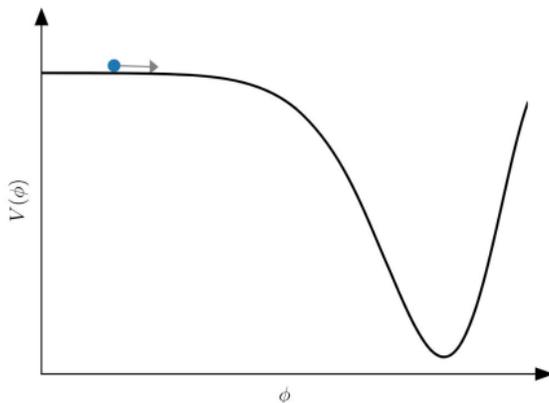


Figura 1.7: En esta gráfica se presenta un campo escalar que se desliza lentamente por un potencial $V(\phi)$. Esta época de inflación cósmica termina cuando el campo alcanza el mínimo de potencial. Fuente: [3].

2

Metodología

En esta sección, se va a presentar y explicar el algoritmo utilizado para llevar a cabo el proyecto. En la primera subsección, se explica cómo se han simulado los mapas que serán utilizados posteriormente. En la segunda subsección, por su parte, se presenta la red neuronal que se pretende entrenar para separar los *foregrounds* de la señal real del CMB. Finalmente, se describe el método para calcular el espectro de potencias del CMB limpio, incluyendo técnicas para tratar el ruido instrumental en el análisis.

Para llevar a cabo este proceso, se ha modificado el código de una red neuronal convolucional, llamado **CMBFSCNN**¹ para adaptarlo a las necesidades del proyecto. Este código tiene una estructura muy simple: simulación de los mapas con los *foregrounds* y ruido instrumental, entrenamiento de la red neuronal, extracción de los *foregrounds* (con sus correspondientes medidas de precisión), y cálculo de los espectros de potencias auto y cros. Para llevar a cabo todo este proceso, el código ha sido programado y modificado utilizando el software de Python².

2.1. Simulaciones

La forma más habitual de representar la radiación del Fondo Cósmico de Microondas es utilizando mapas esféricos a todo el cielo. Concretamente, el esquema de pixelización de HEALPix³, es el de mayor uso, por lo que ha sido el utilizado en este proyecto. Estos mapas consisten en una pixelización completa de la esfera con coordenadas conocidas. Para ello, se parte de una malla cuadrilateral geométrica y refinable construida sobre la esfera[13], como se muestra en la Figura 2.1.

¹<https://github.com/yanyepeng/CMBFSCNN>

²Fundamentos del software Python: <https://www.python.org/>

³<https://healpix.sourceforge.io/>

Esta malla posee una resolución inicial de 12 píxeles, los 12 parches de los que se trata a lo largo del proyecto. Estos 12 parches iniciales se pueden subdividir en píxeles más pequeños que permiten una mayor resolución. La resolución final está determinada por el parámetro N_{side} que es el número de divisiones a lo largo de un píxel base, de tal manera que la resolución angular final ha de ser $\theta_{pix} = \sqrt{\frac{4\pi}{12N_{side}^2}} = \sqrt{\frac{\pi}{3N_{side}^2}}$. El número total de píxeles también se determina mediante este parámetro, $N_{pix} = 12 \cdot N_{side}^2$.

Otra de las ventajas que tiene utilizar el formato HEALPix, es que existen librerías como `healpy` que tienen implementadas diferentes funciones y clases que permiten la visualización de mapas, manipulación de píxeles y transformación a armónicos esféricos, entre otras cosas. Estas funciones son muy interesantes a la hora de trabajar con señales del CMB porque pueden facilitarnos el trabajo.

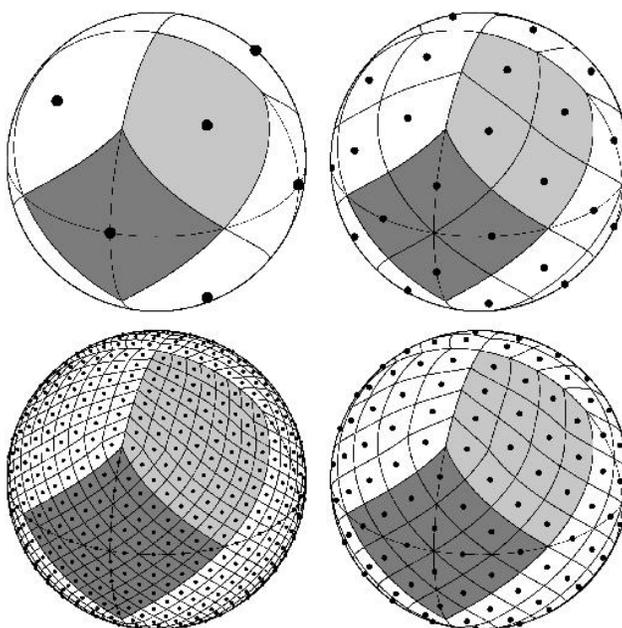


Figura 2.1: Pixelizaciones de la esfera realizadas por HEALPix correspondientes a $N_{side} = 1, 2, 4, 8$ (de izquierda a derecha, empezando por las figuras de arriba). Fuente: <https://healpix.sourceforge.io/>.

2.1.1. Simulación del CMB con *foregrounds*

Una de las principales ventajas de las simulaciones es la posibilidad de generar cada una de las componentes por separado para posteriormente combinarlas en un único mapa. Para el propósito de este proyecto es necesario realizar dos tipos de simulaciones: el fondo cósmico microondas y, por otra parte, esta misma radiación junto con el ruido y los *foregrounds*. Esto es debido al tipo de análisis que se quiere realizar con la red neuronal convolucional⁵.

⁵CNN, por sus siglas en inglés.

Para ello, en primer lugar, se generan realizaciones del CMB basadas en el modelo cosmológico Λ CDM[14], actualmente el modelo más preciso, utilizando los parámetros obtenidos por Planck en 2015[15] (recogidos en la tabla 3 del *paper*). Estas realizaciones se crean mediante el modelo *c2* de PySM (*Python Sky Model*)⁶, el cual genera mapas del CMB lensado.

Es importante destacar que cada realización del CMB generada será diferente debido a la naturaleza estadística de las fluctuaciones primordiales. Los coeficientes $a_{\ell m}$ se generan aleatoriamente siguiendo una distribución gaussiana con varianza determinada por el espectro de potencias teórico C_ℓ . De esta manera, cada mapa representa una posible realización del universo observable, manteniendo las propiedades estadísticas correctas pero con fluctuaciones específicas diferentes.

Una vez obtenido el mapa, es necesario simular los *foregrounds* utilizando uno de los modelos de *foreground* dados por PySM. Para esta simulación se van a utilizar dos casos distintos:

1. Un modelo de PySM sin variación en los parámetros, como puede ser el “s1” (modelo de sincrotrón con variación espacial de la amplitud y el índice espectral) o el “d1” (modelo de polvo con variación espacial de la amplitud, de la temperatura del polvo térmico y del índice de emisividad) en el que se tienen mapas deterministas, es decir, cada vez que se calcule el mapa del *foreground*, no cambia el mapa a todo cielo.
2. Un modelo de PySM con variación en los parámetros, en el que se añaden fluctuaciones aleatorias a los parámetros de amplitud, índice espectral y temperatura (en el caso del polvo térmico) de cada componente de *foreground*. Específicamente, a cada valor de pixel en los mapas de amplitud se suma un número aleatorio, idéntico para todos los píxeles, extraído de una distribución gaussiana con media 0 y desviación estándar de 0.1. Este número se multiplica previamente por el valor de pixel correspondiente. Tanto para el índice espectral y de emisividad y la temperatura del polvo térmico se realiza la misma operación, pero con una desviación estándar de 0.05. En el caso de la temperatura, al contrario que para los demás parámetros, se utiliza un valor distinto en cada píxel. Esto permite generar realizaciones distintas de *foregrounds* manteniendo las propiedades estadísticas correctas. Esta variabilidad se aplica a los parámetros mencionados de las ecuaciones 1.5 y 1.6, así como a las dos amplitudes de la componente de AME de las ecuaciones 9, 10 y 11 de [1].

La distinción entre estas dos formas de representar los *foregrounds* sirve para observar con qué tipo de entrenamiento se podría efectuar una limpieza más efectiva, ya que los mapas del cielo no están completamente caracterizados. Estos mapas de *foregrounds* se suman a los mapas del CMB anteriormente calculados.

A estos mapas finales, se les aplica un suavizado para simular el efecto del *beam* instrumental del telescopio. El *beam* es la función de respuesta angular del instrumento que describe cómo la señal de una fuente puntual se distribuye espacialmente en el detector. Debido a limitaciones físicas como al difracción y el tamaño finito de la apertura, ningún telescopio

⁶Documentación de PySM: <https://pysm3.readthedocs.io/en/latest/>

puede alcanzar una resolución infinita. Este efecto se modela típicamente con una función gaussiana caracterizada por su anchura a media altura (FWHM), y se implementa mediante la función de *smoothing* de healpy aplicando el correspondiente filtro gaussiano.

Limitaciones de PySM

Tal y como se recoge en diversas fuentes[16], es importante darse cuenta de que la función *ud_grade* implementada en healpy⁷, no es la mejor forma de degradar un mapa, pues degrada promediando entre los píxeles. La razón de que no sea la forma apropiada de degradar reside en el hecho de que los píxeles iniciales pueden no cubrir los píxeles finales de manera densa, y de que el mapa inicial puede tener ya aplicado una ventana de pixel. Por ello, en lugar de generar los mapas al N_{side} buscado, es primordial para obtener un buen resultado seguir los siguientes pasos:

1. Se genera el mapa teniendo en cuenta las especificaciones que da la documentación de PySM acerca del modelo utilizado. Es decir, se genera el mapa al N_{side} especificado para que PySM no realice automáticamente un *ud_grade*.
2. Una vez generado el mapa, si no está al N_{side} buscado, se degrada manualmente pasando de mapa a $a_{\ell m}$ y volviendo a pasar, consiguientemente, de $a_{\ell m}$ a mapa.

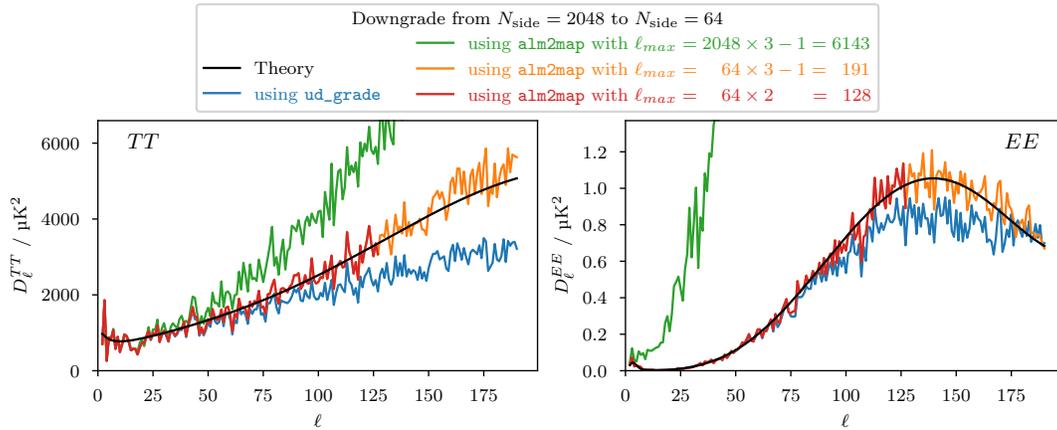


Figura 2.2: Representación gráfica de la dependencia del espectro de potencias modificado respecto al multipolo para diferentes modelos de degradado. Como se puede observar, el modelo *ud_grade* y el que tiene un l_{max} igual al del mapa entrante no corresponden con la teoría por lo que se han de desechar. Fuente: [16].

Este degradado manual se puede realizar de diversas maneras, como se puede observar en la Figura 2.2. En esta ocasión, se ha utilizado la opción representa en amarillo que abarca un mayor número multipolar l , para ello se ha utilizado la siguiente ecuación:

$$a_{\ell m}^{correct} = \frac{a_{\ell m} \cdot W_{\ell}^{out}}{W_{\ell}^{in}} \Big|_0^{\ell_{max}} \quad (2.1)$$

⁷Documentación de healpy: <https://healpy.readthedocs.io/en/latest/>

En esta ecuación se identifica a W_ℓ^{in} como la *pixel window function* del mapa generado y a W_ℓ^{out} como la del mapa que se quiere obtener (a un N_{side} determinado), mientras que $a_{\ell m}$ se corresponde con los coeficientes del mapa generado. En el caso de polarización, este proceso es aún más importante ya que el sistema de coordenadas es local y cambia con el pixel. Una vez realizado este proceso, se vuelve a pasar de $a_{\ell m}$ a mapa para obtener el mapa deseado.

2.1.2. Simulación del ruido

Una vez obtenido el mapa sin ruido del CMB+*foregrounds*, se simula el mapa del ruido utilizando un ruido gaussiano blanco de media cero. Este ruido (distinto en cada caso), se suma a los mapas generados con anterioridad, dando lugar al mapa total que se pretende analizar. Adicionalmente, en Yan et al. 2024, se añade una optimización del ruido. Nosotros no hemos tenido en cuenta esta optimización debido a que hemos trabajado con una resolución menor ($N_{side} = 128$) comparada con la usada en dicho trabajo ($N_{side} = 512$).

2.1.3. Obtención de un mapa plano

Las redes neuronales convolucionales clásicas, están creadas para trabajar con mapas planos, por ello, en múltiples artículos[17], se tiene que transformar el mapa esférico en una imagen en el plano antes de entrenar la red neuronal. Concretamente, en este proyecto, también es primordial para obtener resultados.

En ambos casos se sigue el mismo procedimiento para proyectar la imagen en el plano: se divide el mapa en 12 parches y se indexan en un mapa de mayor proporción, para ello se sigue un método de proyección personalizado. Es importante que el mapa sea de mayor tamaño que el mapa esférico original para evitar en su mayoría los efectos de borde. Esta mecánica de utilizar los 12 parches, es mejor que entrenar parche por parche debido a que permite a la red generalizar mejor, pues tiene una información global del mapa.

Una vez obtenida esta imagen en el plano, para el caso de un experimento con un cubrimiento parcial del cielo (en experimentos como el CMB-S4) se escoge el bloque con el que se va a trabajar. Con todo ello, se obtienen unos mapas similares a los de la Figura 2.3. Estos mapas, después de la extracción de los *foregrounds* han de ser devueltos a mapas esféricos.

Sin embargo, este tipo de representaciones tienen algunos inconvenientes. El hecho de que se proyecte de mapa esférico a imagen en el plano puede generar distorsiones en la representación, pues no es posible mantener la geometría intacta, por ejemplo, los píxeles de healpy no son todos de exactamente de la misma forma. Si se realiza además la proyección a mapa esférico estas distorsiones podrían afectar aún más a la calidad de los resultados.

A esto se añade el ensamblaje, en este código en concreto, en las zonas en las cuales se usen dos parches o más, es posible observar líneas de ensamblaje en el mapa esférico recuperado del CMB, haciendo que la recuperación no sea completamente perfecta.

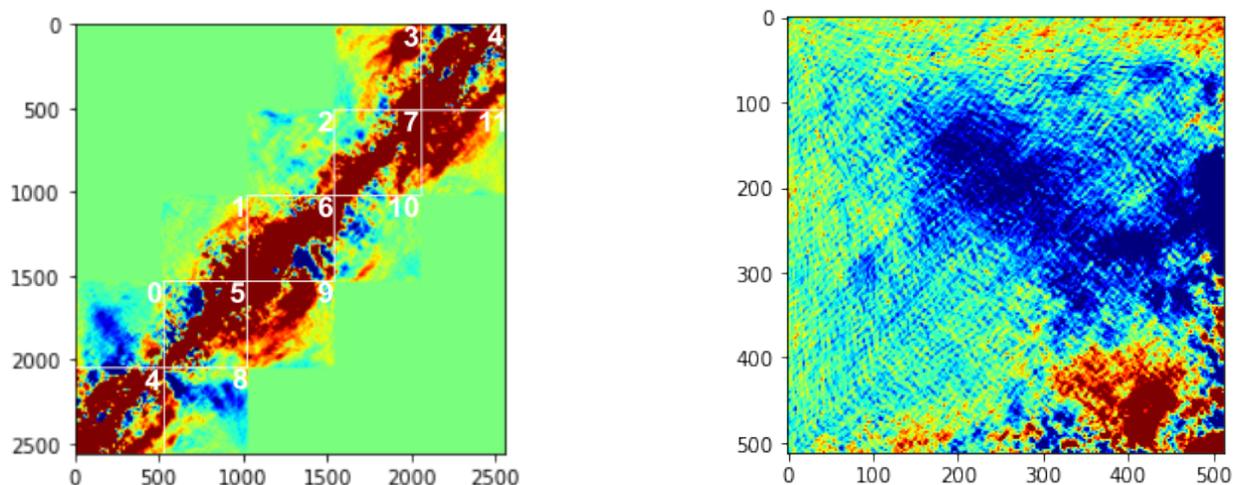


Figura 2.3: Representaciones gráficas de la transformación de mapas esféricos a imágenes en el plano. La figura de la izquierda corresponde con un mapa a todo el cielo en el que los números representan los diferentes parches, mientras que la figura de la derecha corresponde a un parche determinado.

2.2. Redes neuronales convolucionales

Las redes neuronales convolucionales se han convertido en las últimas décadas en una de las tecnologías principales para el procesamiento de imágenes y, en general, datos tipo *grid*. Estas redes utilizan operaciones de convolución para extraer características relevantes de los datos de entrada, como pueden ser los rasgos de una persona, si se está utilizando para *face recognition*. Por todo ello, estas redes han sido las responsables de grandes avances en procesos como la clasificación de objetos o el análisis de datos.

En este proyecto en concreto, constituye el esqueleto sobre el que se va a realizar el análisis. Para obtener la red neuronal utilizada, se ha utilizado la librería de Python de PyTorch⁸ junto con GPUs integradas de Apple para realizar el proceso de compilación más rápido.

2.2.1. Fundamentos

Las redes neuronales artificiales, como la utilizada en este proyecto, presentan unas bases muy sencillas. Estas redes relacionan datos de entrada con datos de salida, es decir, se puede decir que son “aproximadores universales” [18]. Estos datos en nuestro caso serán tensores. Sin embargo, en las redes neuronales más sencillas se relacionan escalares entre sí.

Para entender las redes neuronales, es necesario conocer su funcionamiento interno. Existen diversos tipos de redes neuronales con diferentes propósitos, divididos fundamentalmente en dos categorías: *supervised learning* y *unsupervised learning*. A la primera de ellas, se le tienen

⁸Documentación de PyTorch: <https://pytorch.org/docs/stable/index.html>

que pasar datos de referencia con etiquetas, se minimiza una función de pérdida basada en los errores, sirve para procesos como la clasificación (si una imagen es un gato, un perro o ninguna de las anteriores) o detección, entre otros; el *unsupervised learning*, por el contrario, no usa etiquetas y optimiza funciones de agrupación o reconstrucción, se utiliza para descubrir patrones ocultos a simple vista. En este proyecto, se ha optado por utilizar el *supervised learning*, pues se pretende limpiar un mapa según un mapa de referencia dado.

Dentro de estas redes neuronales, se encuentran distintas capas, dentro de estas capas a su vez encontramos diferentes unidades, las neuronas.

La estructura de una red neuronal, como se observa en la Figura 2.4, es una estructura a capas en la que la entrada de una neurona está conectada con la salida de una o más neuronas de la capa previa. Si se toma la k -ésima neurona de la capa de índice i , la entrada de esta capa está relacionada con la salida de las neuronas conectadas de la capa anterior $i - 1$, \vec{x} . De tal manera que la salida de esta neurona k -ésima ha de tener la siguiente forma:

$$y_k = g_i(\vec{w}_k \cdot \vec{x} + b_k) \quad (2.2)$$

En esta ecuación, \vec{w}_k es el vector de pesos asociado a la neurona k -ésima (de la misma dimensión que el vector de entrada \vec{x}), b_k el sesgo (escalar) y g_i la función de activación. Debido a que las operaciones lineales por sí solas no permiten que la red aprenda patrones complejos no lineales en los datos, es necesario introducir no-linealidad mediante las funciones de activación. Entre las funciones de activación clásicas, se encuentran:

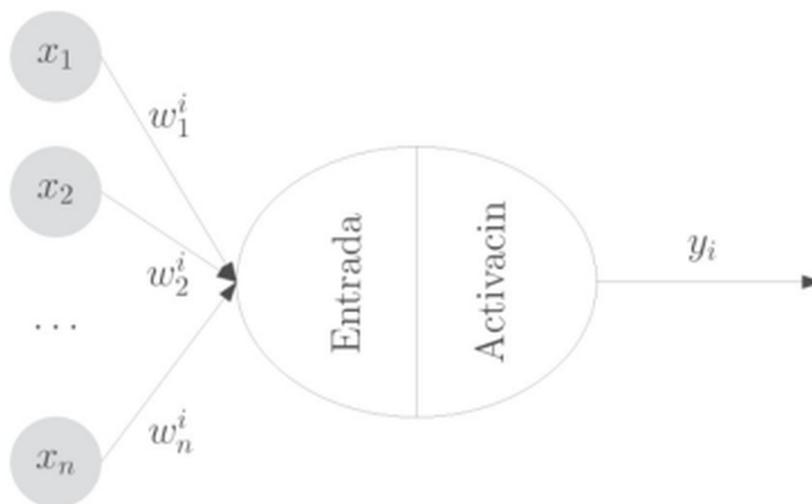


Figura 2.4: Representación de una red neuronal sencilla. En esta red, se encuentran las tres estructuras básicas de una red neuronal: el *input layer*, la primera capa de la red; los *hidden layers*, las capas intermedias con una función de activación entre capas; el *output layer*, la salida la red neuronal, en el caso más sencillo, simplemente un escalar. Fuente: [18].

- **La función escalón**, la cual separa la gráfica en 2, dividiendo en un valor umbral de

la función de activación, α .

$$y(x) = \begin{cases} 1 & \text{si } x \geq \alpha, \\ -1 & \text{si } x \leq \alpha, \end{cases}$$

Aunque también es habitual ver un escalón de tipo $\{0,1\}$ en lugar de $\{-1,1\}$.

- **La función lineal**, la cual genera combinaciones lineales de la entrada, con una ecuación de tipo $y(x) = \beta x$.
- **La función sigmoide**, una de las más utilizada, pues satisface los criterios de diferenciable y monotonía. Esta función varía más en el centro que en los bordes, con una forma parecida a la función escalón y de ecuación:

$$y(x) = \frac{1}{1 + e^{-\frac{x}{\rho}}}$$

- **La función tangente hiperbólica**, la cual satisface los criterios de ser diferenciable y monótona. Tiene una ecuación de la forma $y(x) = \tanh(x)$.
- **La función ReLU**, *rectified linear unit*, una de las funciones de activación más utilizadas en las redes neuronales, se trata de una función máximo: $y(x) = \max(0, x)$.

Las funciones diferenciables y monótonas son las más utilizadas debido a que se puede realizar un gradiente en todo punto del espacio y, además, ayuda a que el entrenamiento no cambie de signo, pues tiene una relación directa entre activaciones y signos.

En este proyecto, se utiliza una modificación de la función ReLU menos común, la función *Leaky ReLU*. Esta función no elimina los valores de x menores que cero en su activación, sino que disminuye su tamaño por un factor, β , en esta ocasión $\beta = 0.4$.

$$y(x) = \begin{cases} x & \text{si } x \geq 0, \\ 0.4 \cdot x & \text{si } x \leq 0, \end{cases} \quad (2.3)$$

Redes convolucionales

Las redes neuronales convolucionales, se diferencian de las redes clásicas debido a su operación de convolución. Así, en estas redes, se reemplaza una capa clásica por una capa convolucional, permitiendo la utilización de tensores, en lugar de escalares. En estas redes, los resultados procedentes de la capa anterior (es decir, el *input* de la capa) se hacen pasar por diferentes filtros (también llamados *kernels*). Es decir, cada capa convolucional está formada por un número determinado de filtros, a cada uno de los resultados de estos filtros se les llama canales.

La operación de convolución se puede expresar matemáticamente de la siguiente manera:

$$S(i, j) = (I * K)(i, j) \sum_m \sum_n I(i + m, j + n) K(m, n) \quad (2.4)$$

En esta ecuación, $S(i, j)$ representa el valor de salida en la posición (i, j) del mapa resultante e , donde i y j son los índices de fila y columna respectivamente. El término I representa la imagen de entrada (*input*) y K el *kernel* o filtro convolucional. Los índices m y n recorren las dimensiones del *kernel*. Normalmente, se utilizan filtros convolucionales de 2 o 3 dimensiones, aunque existen filtros de cualquier dimensión, incluso 1D.

A estos filtros básicos, se les pueden añadir diversos parámetros extra como el *padding*, que es la cantidad de borde adicional que se añade al tensor de entrada; el *stride*, controla la cantidad de píxeles que se mueve un filtro al aplicar la operación de convolución; la dilatación, consistente en insertar huecos entre los elementos de un filtro para ampliar el campo receptivo. Con este último parámetro, se crea la llamada convolución dilatada.

Otra diferencia importante respecto a las redes neuronales clásicas es que, en este tipo de redes, se usan los mismos pesos y sesgos para todas las neuronas de una misma capa (*layer*). Con todo ello, se consigue realizar un análisis de las imágenes del CMB no sólo a nivel local, sino también a nivel global.

Por último, es importante saber que existe la operación opuesta a la convolución, la convolución traspuesta o deconvolución que, en lugar de disminuir el tamaño del tensor (imagen) entrante, aumenta su tamaño, aumentando la resolución de la imagen. Esta operación se utiliza, entre otros procesos, en *autoencoders*.

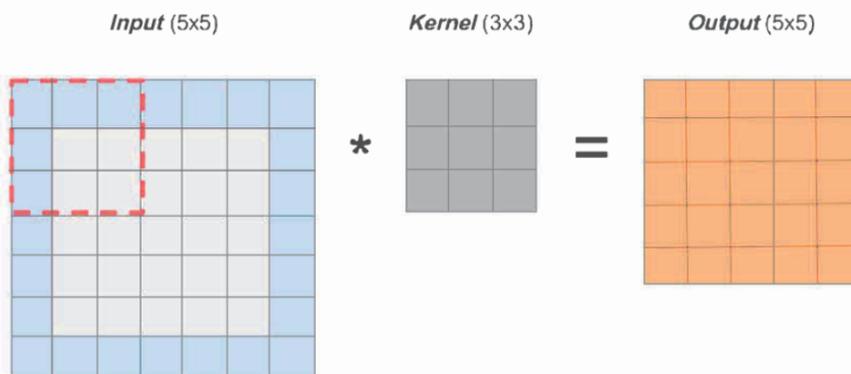


Figura 2.5: Ejemplo de una operación de convolución con *padding*, $p=1$. Como se puede observar, se multiplica el *kernel* por un tensor igual a su tamaño (que esté dentro del tensor entrante) dando lugar a un elemento del canal saliente. Fuente: [18].

Entrenamiento y *Loss Function*

Para hacer un uso correcto de la red neuronal, es necesario realizar un proceso de entrenamiento. Es decir, hay que enseñar a la red a adaptarse al conjunto de datos dado. Para ello, se ajustan los pesos correspondientes a las neuronas con el fin de disminuir la *loss function* (función de pérdida), la cual informa sobre la bondad de los resultados. Los datos utilizados para entrenar esta red se denominan conjunto de entrenamiento y consisten en pares de datos,

correspondientes al mapa que se pretende limpiar y su etiqueta (en este caso, un mapa limpio del CMB). Debido a que la red neuronal presenta un gran conjunto de parámetros, se necesita un conjunto de mapas suficientemente grande para poder llevar a cabo el entrenamiento de manera correcta.

La función de pérdida representa la precisión de las predicciones. En particular, cuanto menor sea la función de pérdida, más precisas las predicciones. Estas funciones de pérdida se pueden clasificar según el propósito que tenga la red neuronal. Entre los propósitos más comunes para los que se utilizan las redes en imágenes (datos 2D) destacan: clasificación binaria, decidir si una imagen es de un perro (1) o no (0), clasificación multiclase, segmentación, clasificando cada pixel de una imagen, o aprendizaje por refuerzo, entre otras.

Sin embargo, como se puede observar, la red neuronal utilizada no corresponde a ninguno de estos propósitos, pues lo que se quiere realizar es limpiar una imagen. Es decir, es necesario introducir un procedimiento más complejo llamada función de pérdida multitérmino, basada en dos funciones de pérdida[19].

$$\mathcal{L} = \mathcal{L}_{\text{MAE}} + \beta \mathcal{L}_{\text{FFT}} \quad (2.5)$$

En esta ecuación, β representa la contribución de \mathcal{L}_{FFT} a la función de pérdida total, durante todo este proyecto se establece $\beta = 1$. El término \mathcal{L}_{MAE} se trata de la función de pérdida del error absoluto medio, es decir, se pretende minimizar la diferencia entre el mapa etiqueta y el mapa del CMB resultante o con residuos. Esta función es denominada L1 en PyTorch.

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{WH} \sum_{w=1}^W \sum_{h=1}^H (|I_{w,h}^n - y_{w,h}^n|) \right] \quad (2.6)$$

Donde N representa el tamaño de lote (el número de muestras por iteración), W y H son el alto y ancho de la imagen en píxeles, $I = f(x)$ la imagen predicha e y la imagen etiqueta del CMB. Por su parte, \mathcal{L}_{FFT} es un término de espacio de Fourier que contribuye a preservar la coherencia espacial de la imagen a analizar.

$$\mathcal{L}_{\text{FFT}} = \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{WH} \sum_{w=1}^W \sum_{h=1}^H (|A_F(I_{w,h}^n) - A_F(y_{w,h}^n)|) \right] \quad (2.7)$$

Siendo A_F la amplitud de la transformada rápida de Fourier (FFT), calculada como al norma L2 de la parte real e imaginaria de dicha FFT.

Una vez definida la función de pérdida, es posible comenzar el entrenamiento de la red neuronal con la ayuda de nuestro conjunto de entrenamiento. Para ello, es necesario definir distintos hiperparámetros como son la tasa de aprendizaje (*learning rate*), el tamaño de lote y el número de épocas. La tasa de aprendizaje es uno de los más importantes, pues controla el paso que se da entre iteraciones y si el paso no es correcto puede tener consecuencias negativas en la red. Por ejemplo, en la Figura 2.6, izquierda, se observa que en ningún momento se llega a obtener el mínimo debido a que el paso es demasiado grande.

El tamaño del lote, por su parte, influye a la hora de realizar el entrenamiento. Este entrenamiento consiste en dos pasos: la evaluación del conjunto de entrenamiento y el posterior cálculo de los pesos de la red. Es en este segundo paso donde se utilizan los lotes, así, se evalúa un lote en vez del conjunto de entrenamiento. Esto tiene diversas ventajas, debido a que disminuye el ruido en la función de pérdida porque promedia más ejemplos, lo que hace los resultados más precisos. El cálculo de los pesos de la red y de la función de pérdida se realizan con el optimizador Adam[20], que calcula el gradiente de la función de pérdida con respecto a los pesos y actualiza estos en la dirección del descenso del gradiente. Además, este optimizador actualiza algunos hiperparámetros como la tasa de aprendizaje para evitar los problemas anteriormente expuestos.

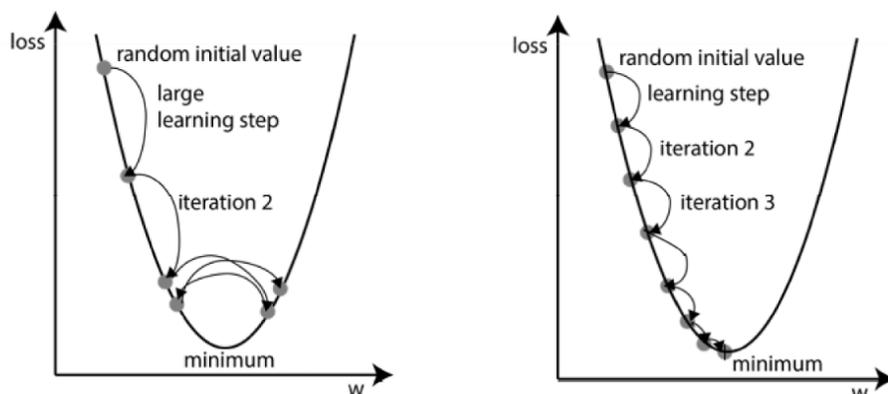


Figura 2.6: Representación gráfica del proceso de descenso de gradiente, como se puede observar si se elige un *learning rate* demasiado alto, es posible que no se llegue al mínimo en el proceso de entrenamiento. Fuente: [21].

Este proceso, se realiza en un número de épocas dado, que coincide con el número de iteraciones que se realizan en el proceso de entrenamiento, es decir, el número de veces que se actualizan los pesos en la red. Entre cada iteración, se congelan los pesos de la red y se calcula la función de pérdida tanto para el conjunto de entrenamiento como de validación. Este segundo grupo, suele ser de aproximadamente el 5% en el caso de tener una gran cantidad de datos. Este conjunto, al estar fuera del conjunto de entrenamiento, es decir, no se ha entrenado la red con estos datos, puede ayudarnos a observar si la red generaliza bien a datos reales.

Comúnmente, la función de pérdida del conjunto validación es algo mayor que la función correspondiente al conjunto de entrenamiento, sin embargo, hay que observar si se produce el proceso denominado *overfitting*. Este proceso se da cuando la red además de aprender los datos reales del entrenamiento, aprende otros patrones que no interesan como puede ser el ruido. Esto hace que generalice mal en los datos fuera del entrenamiento. Para evitar que esto suceda se utilizan diferentes estrategias como reducir el número de épocas o mejorar la red neuronal. Un ejemplo de *overfitting* es el observado en la Figura 2.7. En ésta se puede observar como, a medida que avanza el número de iteraciones, la función de pérdida del conjunto de validación alcanza un mínimo de valor mucho mayor que el mínimo correspondiente al conjunto de entrenamiento.

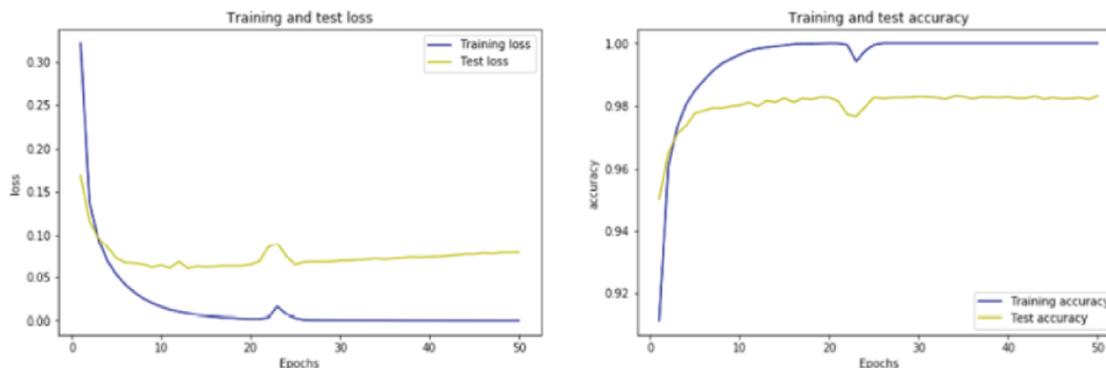


Figura 2.7: Representación gráfica del *overfitting* en el entrenamiento de una red. Normalmente los términos *test loss* y *test accuracy* se sustituirán por *validation*, como es el caso de este proyecto. En esta ocasión se han representando tanto la función de pérdida como la de precisión. Fuente: [21].

Valoración de la precisión de la red

Una vez entrenada la red y visto que la función de pérdida del conjunto de validación es correcta, previniendo el *overfitting* y el *underfitting*, se tiene que observar si se recuperan los mapas del conjunto de test de manera correcta.

Por ello, en lugar de observar la diferencia entre mapas, que puede ser muy tedioso, pues en el CMB es difícil observar a simple vista la diferencia entre dos cielos, se ha elegido estudiar un parámetro. Este parámetro es la desviación media absoluta, MAD, por sus siglas en inglés. La MAD, informa de cuanto se desvía cada predicción del valor real, realizando un promedio.

$$\text{MAD} = \frac{1}{N} \sum_i^N |X_i - Y_i| \quad (2.8)$$

En esta ecuación, se identifica a N como el número total de píxeles de un mapa, a X como el mapa predicho por la red y a Y como el mapa real simulado. Este valor corresponde únicamente a un mapa. Para obtener la MAD para el número de mapas totales, se hace la desviación estándar y el valor cuadrático medio, para obtener la incertidumbre y el valor de la MAD a todos los mapas, respectivamente. Esta cantidad tiene unidades de temperatura.

2.2.2. Arquitectura

En esta sección, se va a analizar la red neuronal convolucional a estudiar. De esta red (Figura 2.8), destacan los siguientes componentes:

- **Conv:** se trata de una operación de convolución.
- **Dilated Conv:** se trata de una operación de convolución dilatada.
- **Deconv:** es una operación de deconvolución.

- **Cat**: es una concatenación. Es decir, una unión de dos tensores a lo largo de una de las dimensiones de estos tensores. En este caso, necesariamente todas las dimensiones tienen que ser las mismas, excepto la dimensión a lo largo de la cual se pretende concatenar.
- **BN**: significa que se hace una normalización por lotes (*batch normalization*[22]). Se realiza la normalización del *output* de una capa dada, en cada uno de sus canales. Para ello, se calcula la media y desviación estándar en cada canal, de manera que:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}} \quad (2.9)$$

- **ReLU**: significa que se aplica una función de activación de tipo *Leaky ReLU*, con una pendiente negativa establecida en 0.4.

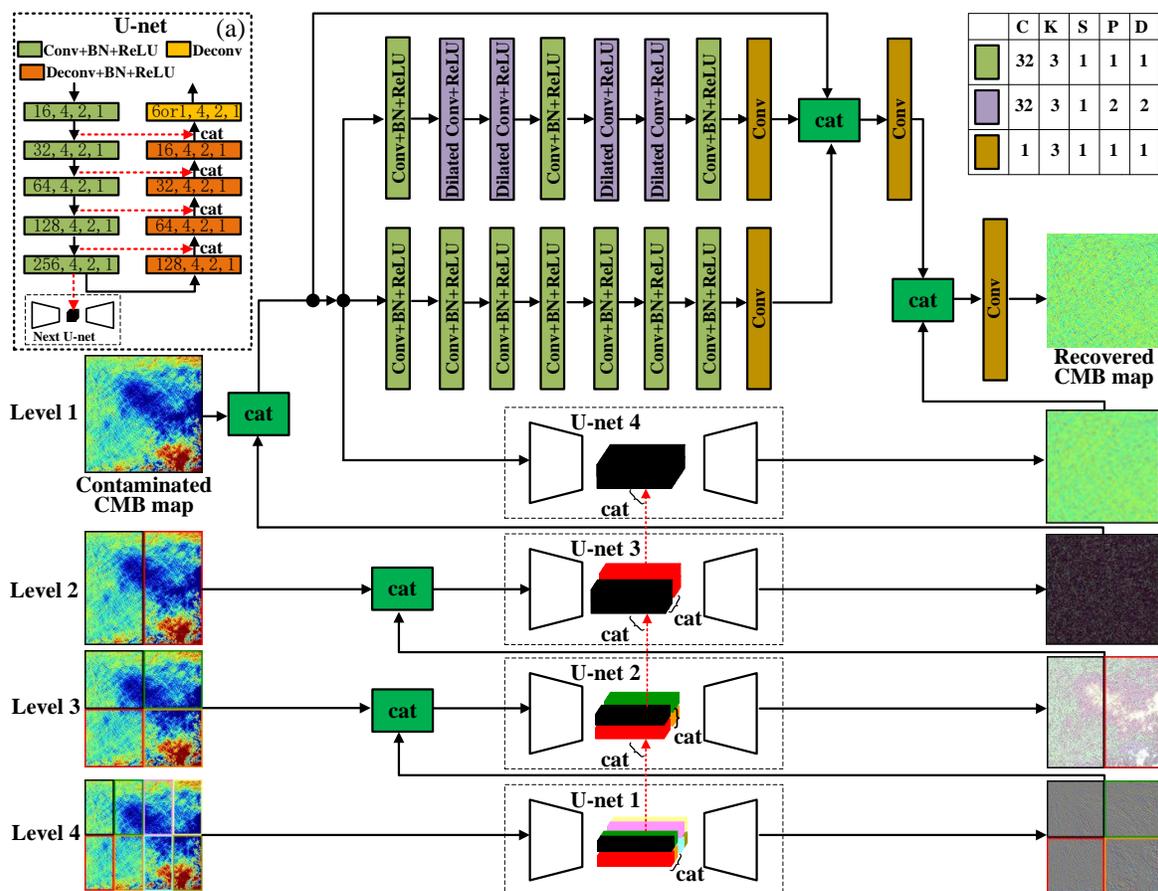


Figura 2.8: Ejemplo de red neuronal convolucional utilizada en el código. En este proyecto, debido al tiempo de compilación que lleva utilizar este tipo de red neuronal se ha utilizado una un poco más simple, de la misma forma pero utilizando 3 niveles en lugar de 4. Fuente: [19].

A continuación, se va a desglosar la red neuronal utilizada. Para ello, primeramente, es necesario dividir el mapa de entrada (de tipo Q o U) en 2 (*level 2*) y cada uno de estos mapas

a su vez en 2 (*level 3*).

Cada uno de estos niveles pasa por una red neuronal tipo U-net[23] modificada, pues se utiliza una función de activación de tipo *Leaky* ReLU en lugar de la función ReLU clásica. Es también importante tener en cuenta que en cada uno de los niveles, el *input* de la U-net consiste en el mapa cortado a ese nivel concatenado con el resultado del mapa del nivel anterior. Además, dentro de la red U-net, después de aplicar la codificación, los resultados se concatenan entre ellos y con los resultados de la codificación del siguiente nivel.

El nivel 1, es el más complejo de todos debido a que realiza múltiples operaciones, después de haber concatenado el mapa del nivel 1 con los resultados del nivel anterior:

1. Hace pasar estos datos por una última U-net que nuevamente se comunica con la anterior entre el proceso de codificación y decodificación.
2. Hace pasar estos datos por múltiples operaciones de convolución seguidas por función de activación *Leaky* ReLU hasta llegar a una operación de convolución final.
3. Hace pasar los datos por una repetición de operaciones de convolución seguidas por una función de activación *Leaky* ReLU junto con dos operaciones de convolución dilatada seguida de una función de activación *Leaky* ReLU, para cada una de ellas (este proceso se repite dos veces), para finalizar con una nueva operación de convolución con una función de activación *Leaky* ReLU y una última convolución.

El resultado de los procesos 2 y 3 se concatena con el *input* para después realizar una nueva convolución. El resultado de esta convolución se concatena con los resultados del proceso 1 para finalizar con una última convolución que nos va a dar el *output*, un mapa limpio del CMB.

En la tabla superior derecha de la Figura 2.8, se observan los parámetros asociados a las diferentes operaciones de convolución, “C”, “K”, “S”, “P” y “U”, representan el número de canales de salida, tamaño de filtro, paso, *padding* y el factor de dilatación de parche convolucional (si lo hubiera) respectivamente.

2.3. Cálculo del espectro de potencias

Una vez obtenido el CMB limpio, sin *foregrounds* ni ruido instrumental, es necesario calcular el espectro de potencias para observar si se recupera correctamente la señal de los modos B secundarios. En el análisis se calculan dos tipos de espectros de potencias: el auto-espectro (calculado a partir de un único mapa) y el espectro de potencias cruzado utilizando la aproximación de “*half-split map*”. Para este último, se divide el conjunto de datos en dos mitades correspondientes a dos instantes de tiempo distintos, y al calcular el espectro de potencias cruzado entre estas dos mitades, el ruido no correlacionado se cancela estadísticamente, mejorando la relación señal-ruido. Ambos resultados se presentarán y compararán en la sección de resultados.

Para ello, en primer lugar, se aplica la máscara binaria tratada con anterioridad. Esto es debido a que, para el cálculo del espectro de potencias, es importante no tener emisiones intensas, normalmente procedente del plano galáctico, pues pueden sesgar este espectro a multipolos bajos. Luego, se pasa de mapa a a_{lm} con healpy. Una vez obtenidos estos a_{lm} , es posible calcular el espectro de potencias usando la librería de NaMaster¹⁰. NaMaster (Narita's Master) es una librería especializada que calcula el espectro de potencias angular utilizando el formalismo pseudo- C_ℓ . y, crucialmente, corrige los efectos sistemáticos introducidos por la aplicación de máscaras, como el acoplamiento de modos y la pérdida de ortogonalidad de los armónicos esféricos.

En esta ocasión, dentro del cálculo del espectro de potencias, es necesario distinguir entre dos casos: el espectro de potencias para spin 1, es decir, para QQ y UU; el espectro de potencias para spin 2, es decir, EE y BB. El cálculo de estos espectros de potencias se realiza para distintos modelos de *foregrounds* y experimentos.

Todos los espectros de potencias obtenidos en las diferentes figuras de este proyecto, corresponden a la media aritmética de los espectros de potencias de múltiples mapas, a diferencia de lo mostrado a nivel de mapa, como se hace en las figuras correspondientes a los mapas esféricos y planos, donde únicamente se presenta un mapa representativo. Asimismo, para el espectro de potencias recuperado, se ha calculado la desviación estándar de todo el conjunto de mapas, observada en las barras de error. Esta forma de representación es adoptada porque resulta más interesante realizar la media debido a que es más informativo, resume el comportamiento estadístico global de la red.

2.3.1. Comprobación experimental

Para comprobar la bondad de los resultados de los espectros de potencias, se ha decidido calcular la diferencia entre el espectro real que se espera obtener, y el resultado.

$$\sigma_\ell = \sqrt{\frac{\sum_{i=1}^N (D_\ell^{i,pred} - D_\ell^{i,esp})^2}{N}} \quad (2.10)$$

En esta ecuación, $D_\ell^{i,pred}$ es el espectro de potencias predicho y $D_\ell^{i,esp}$, el esperado y N , el número de simulaciones. Si esta diferencia, es pequeña en comparación con las amplitudes esperadas en cada uno de los casos, se puede hablar de un resultado aceptable.

En el *paper*[1], se ha utilizando un *binning* a la hora de representar los espectros de potencias. Sin embargo, en este proyecto no se observa este *binning* debido a que se ha decidido representar el espectro de potencias completo, pues presenta mucha información interesante sobre la eficiencia de nuestra red.

¹⁰Documentación de NaMaster: <https://namaster.readthedocs.io/en/latest/>

3

Resultados

En esta sección, se van a presentar los distintos resultados obtenidos en este proyecto trabajando con la red neuronal anteriormente propuesta.

En primer lugar, antes de realizar cualquier análisis, es importante destacar que los valores de FWHM del *beam* instrumental mostrados en la Tabla 3.1 corresponden a las especificaciones nominales de cada experimento. Sin embargo, cuando se trabaja con simulaciones a una resolución diferente (distinto N_{side}), es necesario ajustar el tamaño del *beam* instrumental θ_b para que sea consistente con la pixelización utilizada. Esto se debe a que el *beam* debe ser mayor que el tamaño angular del pixel para evitar efectos de *aliasing* (fenómeno que ocurre cuando estructuras más pequeñas que el tamaño del pixel aparecen incorrectamente como estructuras de mayor escala). El nuevo tamaño del *beam* se calcula mediante la ecuación:

$$\theta_b = \frac{2.4\theta_{pix}}{\pi} \cdot 60 \cdot 180[\text{arcmin}] \quad (3.1)$$

Donde θ_{pix} es el tamaño angular típico de un pixel, calculado como: $\theta_{pix} = \sqrt{4\pi/N_{pix}}$. El factor 2.4 proviene de la relación empírica para evitar *aliasing* en la pixelización de HEALPix, y los factores numéricos convierten de radianes a minutos de arco. Es importante mencionar que, aunque el entrenamiento y test se realiza utilizando todas las frecuencias disponibles de cada experimento, los resultados finales se presentan a una frecuencia representativa: $\nu_{S4} = 220$ GHz para CMB-S4, $\nu_{LB} = 166$ GHz para LiteBIRD y $\nu_P = 143$ GHz para Planck.

3.1. Test Inicial: CMB-S4

Antes de realizar cualquier análisis, se ha optado por realizar un test inicial para comprobar si se recuperan los resultados del *paper*[1], concretamente los resultados correspondientes al experimento CMB-S4. Para ello, se siguen las indicaciones del *paper* sobre cómo entrenar a la red neuronal. Concretamente se pasan como hiperparámetros: $\alpha = 0.01$, $iterations =$

$3 \cdot 10^4$, *batch size* = 12, la tasa de aprendizaje, el número de épocas y el tamaño de lote respectivamente. Además, se han decidido crear 1000 mapas de entrenamiento, a los que se añaden 300 mapas distintos del ruido. Para el conjunto de validación y de test, se han creado 300 mapas del cielo y otros 300 de ruido, para cada uno de los casos.

Tabla 3.1: Datos de los distintos experimentos utilizados en este proyecto. En estas tablas se recogen las principales variables que se van a utilizar como son los distintos canales de frecuencia, la FWHM del *beam* instrumental y el nivel de ruido blanco de polarización en estos canales. Fuentes: [24], [25] y [2].

CMB-S4		
Frecuencia (GHz)	FWHM del <i>beam</i> (arcmin)	Nivel de ruido blanco ($\mu\text{K}\cdot\text{arcmin}$)
30	72.8	3.53
40	72.8	4.46
85	25.5	0.88
95	22.7	0.78
145	25.5	1.23
155	22.7	1.34
220	13.0	3.48
270	13.0	5.97

LiteBIRD		
Frecuencia (GHz)	FWHM del <i>beam</i> (arcmin)	Nivel de ruido blanco ($\mu\text{K}\cdot\text{arcmin}$)
50	56.8	32.78
78	36.9	18.59
100	30.2	12.93
119	26.3	9.79
140	23.7	9.55
166	28.9	5.81
195	28.0	7.12
235	24.7	15.16
280	22.5	17.98
337	20.9	24.99

Planck		
Frecuencia (GHz)	FWHM del <i>beam</i> (arcmin)	Nivel de ruido blanco ($\mu\text{K}\cdot\text{arcmin}$)
30	32.29	210.0
44	27.94	240.0
70	13.08	300.0
100	9.66	117.6
143	7.22	70.2
217	4.90	105.0
353	4.92	438.6
545	4.67	42.0
857	4.22	43.2

Estos mapas han sido generados a $N_{side} = 512$. Además, para la simulación, se han utilizado los modelos “s1”, “d1” y “a2” de PySM para las componentes de *foreground*, así como el modelo “c2” para el CMB.

Con todo ello, se ha entrenado a la red neuronal con una duración del entrenamiento de unos 3 días utilizando una GPU Apple M2 Ultra con 60 núcleos GPU y 128 GB de memoria unificada, lo que permitió acelerar significativamente el proceso de entrenamiento mediante el uso de *Metal Performance Shaders* (MPS) como backend de PyTorch y una desviación media absoluta para Q y U de $MAD = (0.015 \pm 0.001) \mu\text{K}$ y $MAD = (0.020 \pm 0.001) \mu\text{K}$, respectivamente.

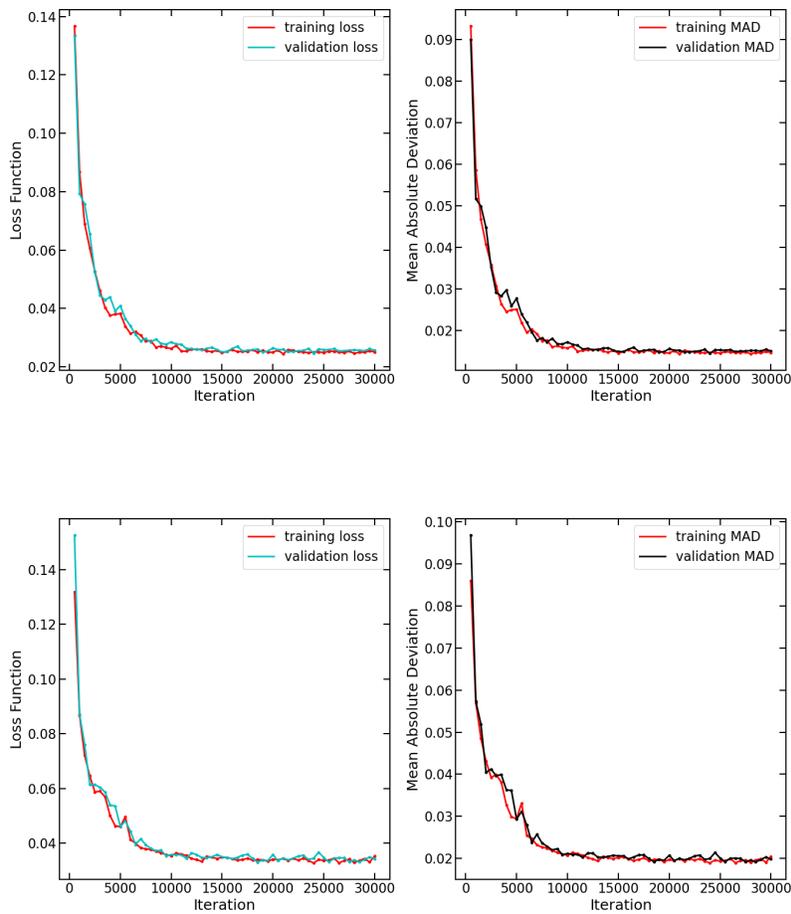


Figura 3.1: Representación gráfica de las funciones de pérdida y la MAD para el experimento CMB-S4. Las gráficas superiores corresponden con el modo Q y las inferiores con el modo U de polarización.

Además, se han añadido variaciones a los *foregrounds* debido a que los modelos no están bien definidos. Estas variaciones son de: $\sigma_\beta = 5\%$, $\sigma_A = 10\%$ el error del índice espectral y de la amplitud del modelo s1 de sincrotrón, respectivamente; $\sigma_\beta = 5\%$, $\sigma_A = 10\%$, $\sigma_T = 5\%$, el

error del índice de emisividad, de la amplitud y de la temperatura del modelo d1 del polvo, respectivamente ; $\sigma_A = 10\%$, el error de la amplitud del modelo a2 de AME. Con la suma de un valor constante para cada pixel excepto en el caso de la temperatura, a la que se le suma un valor distinto en cada pixel.

Por su parte, en la Figura 3.1, se presentan las curvas de entrenamiento de la red neuronal. En éstas, se puede observar como, a medida que se realizan las iteraciones, la función de pérdida y la desviación media absoluta disminuyen hasta converger en un valor constante. Las curvas se mantiene estables aún aumentando el número de iteraciones, lo que indica que el entrenamiento ha alcanzado la convergencia. Además, es importante destacar que no existe sobreajuste (*overfitting*) y que el modelo generaliza correctamente. Este comportamiento, junto con el valor aceptable alcanzado, confirma la calidad del entrenamiento.

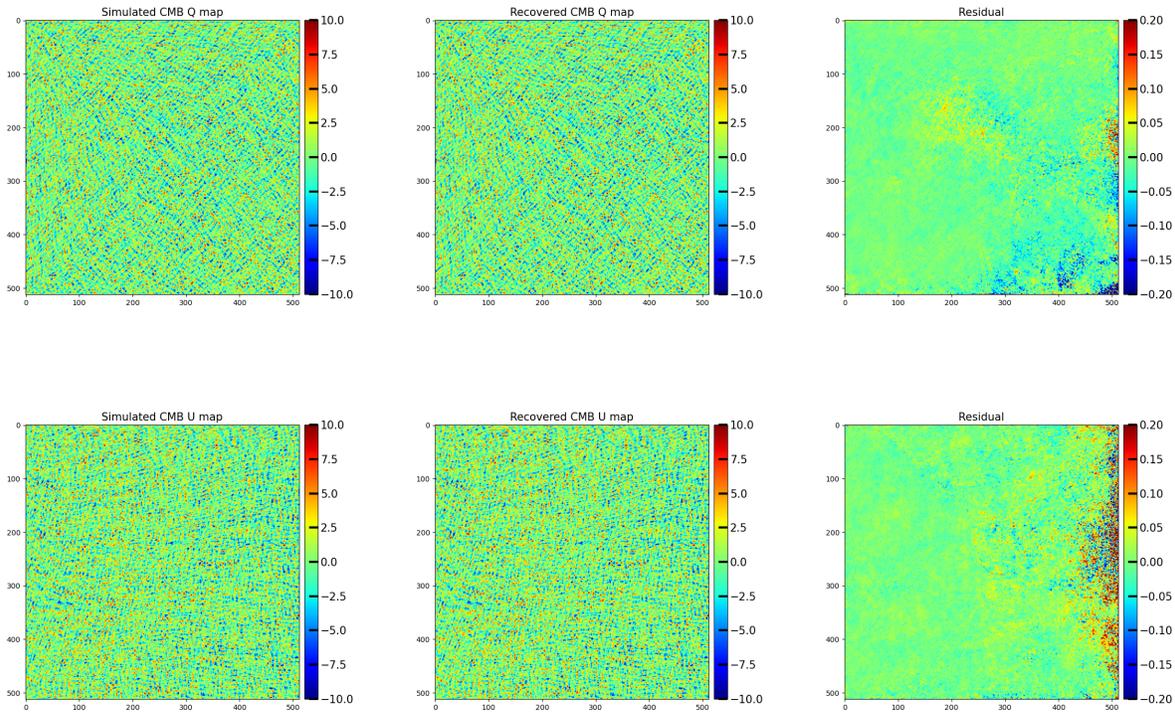


Figura 3.2: Representaciones gráficas de los mapas planos del CMB simulado y recuperado, así como el mapa residual correspondiente a la comparación entre los mapas previos para el caso del CMB-S4. Las figuras superiores corresponden con los mapas planos de Q, mientras que las inferiores se trata de los mapas planos de U.

Los resultados obtenidos para la desviación media absoluta en el experimento de CMB-S4 muestran valores de $MAD = (0.0015 \pm 0.001) \mu\text{K}$ y $MAD = (0.0020 \pm 0.001) \mu\text{K}$ para los modos Q y U respectivamente. Estos resultados son compatibles con los valores reportados en el *paper* original: $MAD = (0.0016 \pm 0.008) \mu\text{K}$ y $MAD = (0.0021 \pm 0.002) \mu\text{K}$, lo que confirma la reproducibilidad del método propuesto utilizando los mismos hiperparámetros y

el mismo número de mapas.

Por otro lado, para el caso de un parche del cielo del experimento CMB-S4 representado en el plano, Figura 3.2, es importante señalar que el entrenamiento es más eficiente en el centro de nuestro parche que en los bordes, específicamente, en el borde derecho donde se observa más contribución de mapa residual. Esto es debido a que este borde está más cerca de la zona de emisión de *foreground* galáctico. El resto de mapas planos se pueden observar en el Apéndice B.

Sin embargo, al realizar la transformación de mapa plano a mapa esférico, Figura 3.3, este efecto de borde se intensifica, haciendo que se presente un ruido residual bajo en la mayor parte del mapa esférico. Esto es debido a lo que se comentó en anteriores apartados, al realizar el cambio de mapa esférico a mapa plano y nuevamente a mapa esférico, aparecen distorsiones disminuyendo la calidad de los resultados.

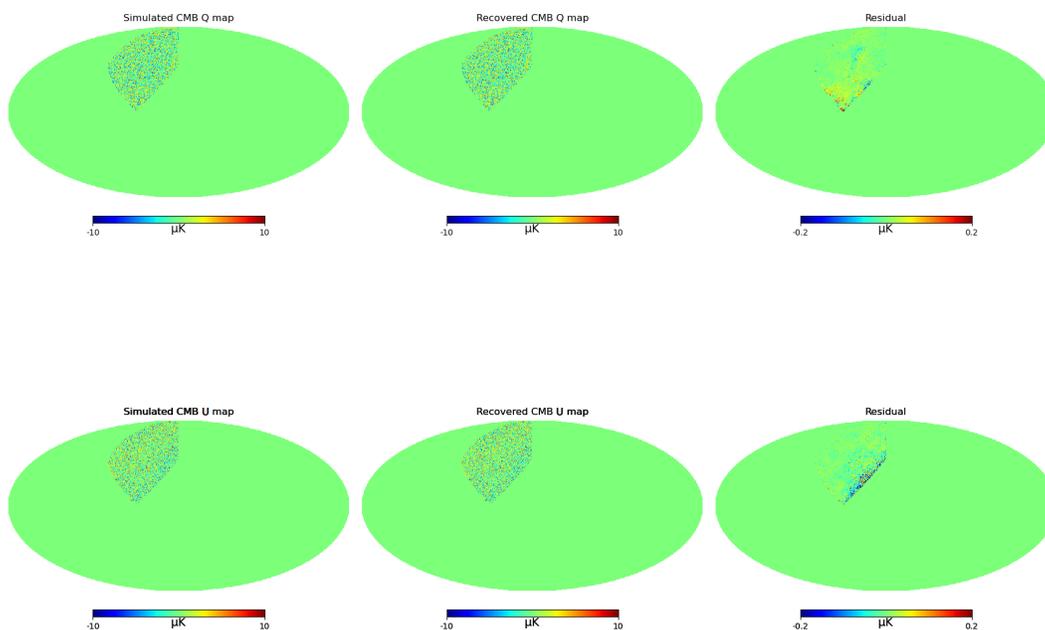


Figura 3.3: Representaciones gráficas de los mapas simulado, recuperado y residual del CMB para el experimento CMB-S4. La figura superior corresponde con el mapa de Q, mientras que la inferior corresponde con el mapa de U.

En cuanto al análisis de los espectros de potencias obtenidos, Figura 3.4, se representan tanto los autoespectros como los espectros cruzados para los modos E y B de polarización. Para el experimento CMB-S4, se observa una excelente recuperación del modo E, tanto en el autoespectro como en el espectro cruzado. Únicamente a ℓ bajos se aprecia un ligero incremento en las barras de error, lo cual puede atribuirse a que CMB-S4 no observa todo el cielo, sino

únicamente un parche limitado.

En el modo B se observa un comportamiento similar a ℓ bajos, aunque adicionalmente, a ℓ altos, las barras de error se incrementan considerablemente debido al aumento del ruido instrumental a escalas angulares pequeñas. No obstante, en términos generales, los resultados obtenidos son muy satisfactorios y demuestran la eficacia del método de separación de componentes implementado.

Por último, es de comentar que todos estos resultados obtenidos se corresponden con los provistos en el *paper*[1] para el experimento del CMB-S4. Esto quiere decir que los resultados obtenidos se corresponden con una utilización correcta del código. Por tanto, se pueden llevar a cabo los siguientes pasos del proyecto.

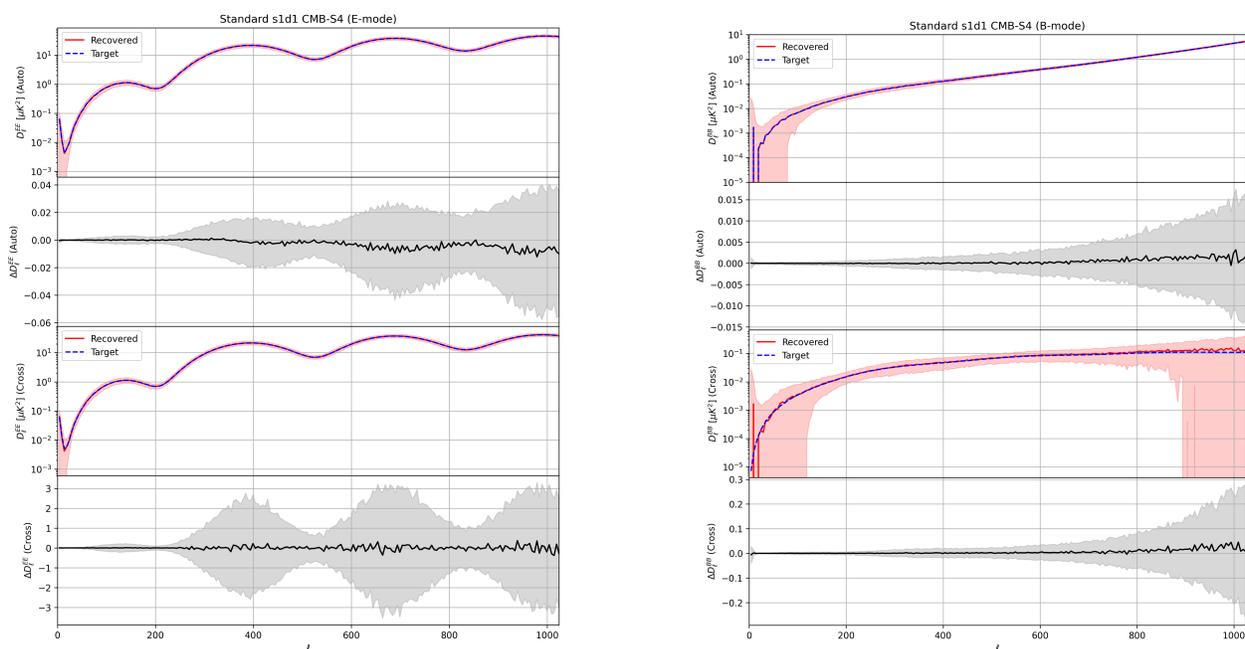


Figura 3.4: Espectro de potencias del modo E y B del CMB. El panel superior de cada figura corresponde con el espectro de potencias del CMB, mientras que el panel inferior muestra la estimación del error asociada. Por otra parte, es preciso señalar que las figuras inferiores muestran el espectro de potencias obtenido mediante cross-correlación de dos conjuntos de datos independientes.

3.2. LiteBIRD

Además de los resultados del CMB-S4 obtenidos en el *paper*, también se pretende obtener resultados nuevos. Por ello, se ha procedido a simular el experimento de LiteBIRD a $N_{side} = 128$ para comprobar la eficiencia de la red. En todos estos casos, se ha utilizado la misma tasa de aprendizaje, $\alpha = 0.01$, y el mismo tamaño de lote, $batch\ size = 1$. Asimismo, se ha

empleado el mismo número de mapas: 900 mapas del cielo y 300 mapas del ruido para el conjunto de entrenamiento; 300 mapas del cielo y 300 mapas del ruido tanto para el conjunto de validación como para el conjunto de test. Cabe destacar que estas redes han sido más complicadas de entrenar debido a que se han tenido que hacer múltiples pruebas hasta dar con los hiperparámetros correctos.

Para llevar acabo este análisis se ha disminuido de manera considerable la resolución, inicialmente de $N_{side} = 512$, debido a que al tener un mapa a todo el cielo el proceso de compilación sería demasiado lento. Sin embargo, en lugar de ser un sesgo en nuestro proyecto nos sirve para comprobar el funcionamiento de la red a esta resolución, siendo posible observar también los modos B primordiales.

3.2.1. Utilización de los modelos s1, d1 y a2

En esta subsección, se procede a trabajar con mapas del CMB con *foregrounds* para distintas situaciones: *foregrounds* con variación introducida; *foregrounds* sin variación; entrenamiento sin variación de *foregrounds* y conjunto de test con variación de los *foregrounds*. Para todas estas situaciones, sólo será necesario entrenar dos redes neuronales que, consiguientemente, se utilizarán para otros fines. Ambas redes tiene el mismo número de épocas, $iterations = 4 \cdot 10^4$.

Con variación de los *foregrounds*

En esta ocasión, se ha entrenado una red neuronal en la cual se han añadido variaciones a los *foregrounds* debido a que, actualmente, no están bien definidos. Por ello, creemos importante obtener una red neuronal que pueda limpiar un mapa del cielo que tenga un cierto error, es decir, cuyos *foregrounds* puedan tener una cierta desviación estándar. En este caso, para “s1”: $\sigma_\beta = 5\%$, $\sigma_A = 10\%$; para “d1”: $\sigma_\beta = 5\%$, $\sigma_A = 10\%$, $\sigma_T = 5\%$; para “a2”: $\sigma_A = 10\%$. Con la suma de un valor constante para cada pixel excepto en el caso de la temperatura, a la que se le suma un valor distinto en cada pixel.

En la Figura 3.5, se puede observar el entrenamiento de esta red neuronal. En este caso, el nivel de ruido en el entrenamiento es significativamente mayor que en el caso del CMB-S4. Esto es debido a que se ha cogido un tamaño de lote muy pequeño. No obstante, debido al excesivo tiempo de cómputo que llevaría aumentar el tamaño de lote (unas 200 horas), se ha decidido tomar este resultado. Por lo tanto, a pesar del ruido presente, el resultado obtenido es satisfactorio, ya que la variación observada es mínima. El resto de curvas de entrenamiento obtenidas en este proyecto se recogen en el Apéndice A.

La desviación media absoluta para el conjunto de test en esta ocasión es de $MAD = (0,011 \pm 0,002) \mu K$ y $MAD = (0,011 \pm 0,002) \mu K$, para Q y U respectivamente.

Como se puede observar en las dos subfiguras superiores de la Figura 3.6, esta red es capaz de recuperar con bastante exactitud el cielo excepto en la zona del plano galáctico, es decir, a latitud galáctica cero. Esto es debido a que en estas zonas domina la emisión de *foregrounds* frente a la del CMB lo que hace más difícil eliminar esta emisión. A esto se le añade, que en

este plano se tienen estructuras más complejas que en otras zonas del cielo, lo que dificulta aún más la limpieza.

En cuanto al espectro de potencias, Figura 3.8 primera columna, se puede observar que se recuperan con bastante exactitud todos los espectros de potencias excepto el del modo B del CMB obtenido mediante cross-correlación. El primer factor que hace que este espectro no se recupere del todo bien es algo que se puede observar también en el auto-espectro del modo B, a multipolos bajos el espectro recuperado es mayor que el espectro real.

Este hecho, está relacionado con el mapa residual que se puede observar del CMB. Al haberse recuperado peor el mapa del CMB en zonas cercanas al plano galáctico, es decir, a multipolos bajos, se produce una sobreestimación del espectro de potencias en estas zonas que origina que, a ℓ aproximadamente por debajo de 75, el espectro recuperado y simulado se separen.

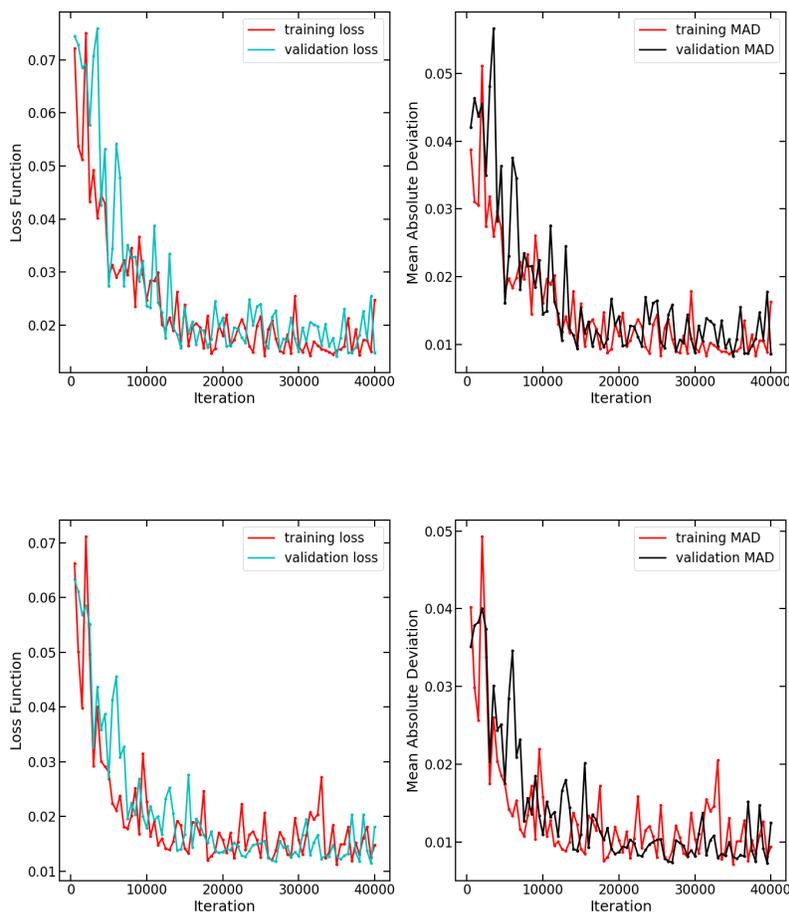


Figura 3.5: Representación gráfica de las funciones de pérdida y la MAD para el experimento LiteBIRD con *foreground* variable. Las gráficas superiores corresponden con el modo Q y las inferiores con el modo U de polarización.

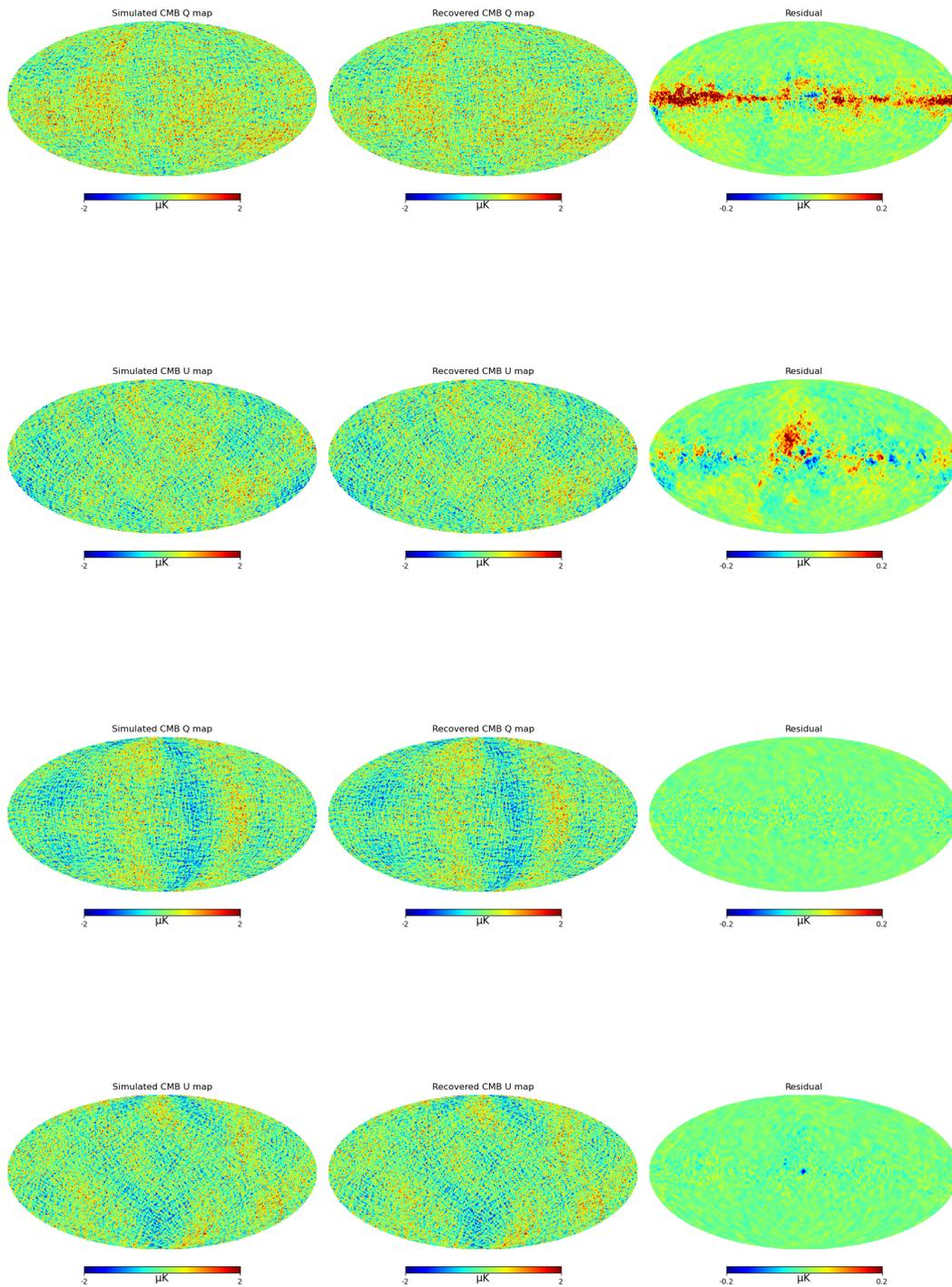


Figura 3.6: Representaciones gráficas de los mapas simulados, recuperado y residual del CMB para el experimento de LiteBIRD. Las dos primeras subfiguras corresponden al experimento de LiteBIRD con variación de *foregrounds*. Las otras dos subfiguras corresponden al experimento de LiteBIRD sin variación.

A multipolos altos, por otra parte, lo que sucede es que la señal-ruido (SNR) es muy pequeña debido al ruido instrumental presente en el instrumento. En esta ocasión, el efecto de la señal-ruido va a afectar a ℓ a partir de aproximadamente N_{side} debido a que el ruido de LiteBIRD es mayor que el del CMB-S4, es decir, será comparable con la señal a multipolos relativamente bajos. Es más difícil extraer información de la deconvolución a escalas menores lo que hace que se observe el gran aumento del error en el cross-espectro recuperado que presenta el modo B.

Sin variación de los *foregrounds*

En esta ocasión, se ha entrenado una red neuronal con los modelos s1 y d1 fijos, con el fin de observar la eficiencia de una red que recibe los mismos mapas de *foregrounds* todo el tiempo, es decir, una red que puede aprender mejor a limpiar un caso de *foreground* concreto.

Para este caso, una vez entrenada la red neuronal y pasado el conjunto de teste por ella, se obtienen unas desviaciones medias absolutas de $MAD = (0.0052 \pm 0.0001) \mu\text{K}$ y $MAD = (0.0055 \pm 0.0004) \mu\text{K}$, para Q y U respectivamente.

En este caso, se puede observar como los mapas recuperados, es decir, las dos subfiguras inferiores de la Figura 3.6 se recuperan de manera aproximadamente uniforme. En esta ocasión, no se tiene el efecto de los *foregrounds* procedentes del plano galáctico excepto en el punto central del mapa de U. Es decir, la red neuronal para el caso de sin variación de los *foregrounds* realiza una limpieza bastante eficaz.

En cuanto al espectro de potencias, Figura 3.8 columna 4, se recuperan exactamente los modos E y B, excepto el modo B de cross-correlación. En éste, al igual que en el resto de resultados anteriores, existe un efecto de baja señal-ruido a multipolos altos que hace que no se pueda recuperar correctamente la información del espectro de potencias y aumente el error enormemente.

Comparación de los resultados del entrenamiento con y sin variación del *foreground*

Lo primero que llama la atención de estos dos procesos de aprendizaje de la red neuronal convolucional son los mapas esféricos obtenidos, especialmente, los mapas residuales.

Mientras que en la red entrenada con variación de los *foregrounds* se puede observar que no se recupera correctamente el CMB cerca del plano galáctico, esta recuperación en la red entrenada con *foregrounds* sin variación es casi perfecta. Es decir, el hecho de haber entrenado la red con un *foreground* fijo hace posible a la red limpiar eficazmente el *foreground* galáctico, lo cual es una ventaja importante de este tipo de entrenamiento.

En cuanto al cros-espectro de potencias del modo B, se puede observar en la Figura 3.8 como, a multipolos bajos, la sobreestimación del espectro de potencias recuperado debido a los residuos es mucho menor en el caso del entrenamiento sin variación de *foregrounds*, es decir,

para un mapa fijo se recuperan mejor los multipolos bajos.

Por otra parte, se puede observar que a multipolos altos, el entrenamiento sin variación de *foregrounds* es capaz de recuperar una mayor cantidad de señal, siendo el error de la diferencia mucho menor para el caso de los modelos de *foregrounds* sin variación.

Es decir, el entrenamiento de la red neuronal convolucional con *foregrounds* sin variabilidad funciona más eficazmente en la limpieza de mapas sin variabilidad, pudiendo llegar a una resolución mayor. Ahora bien, el *foreground* no es una cantidad que esté bien definida sino por lo que sería interesante comprobar esta segunda red neuronal con mapas con una cierta variabilidad.

Por todo ello, para finalizar el análisis, es importante también, comparar la eficiencia de estas redes en casos para los que no han sido entrenadas. Es decir, utilizar la red entrenada para el apartado «sin variación» para limpiar los mapas «con variación» en los *foregrounds*.

Entrenamiento sin variación, test con variación

Debido a lo que se ha comentado anteriormente, es necesario comprobar la eficiencia del caso comentado. Es decir, se procede a comprobar cómo funciona una red entrenada para limpiar cielos con un modelo exacto de *foregrounds* para *foregrounds* realistas que presentan incertidumbres en los parámetros.

Con esta red entrenada anteriormente, se pasa el conjunto de test por la red con el fin de obtener la desviación media absoluta en este caso de $MAD = (0.036 \pm 0.015) \mu K$ y $MAD = (0.024 \pm 0.010) \mu K$ para Q y U respectivamente.

Para este caso, se puede observar como la red no limpia correctamente el mapa del CMB, como se observa en la Figura 3.7, dejando una gran región del cielo con *foreground* residual. Es decir, la red neuronal entrenada sin variabilidad al introducirle un mapa con *foregrounds* variables, no sabe cómo limpiarlo correctamente.

En cuanto al espectro de potencias, figura 3.8 columna 3, se comienza a ver una pequeña diferencia entre el modo E real y el esperado debido a la mala limpieza del mapa. El modo B, por su parte, en el espectro de potencias de auto-correlación presenta una gran separación a multipolos bajos hasta $\ell \approx 175$, es decir, un amplio intervalo de multipolos. Esto es debido nuevamente a que el espectro de potencias de la mayor parte del cielo está sobreestimado cómo se observa en el mapa residual. En cuanto al modo B de cross-correlación, se puede observar como la recuperación es inexistente, siendo el espectro de potencias recuperado aproximadamente un orden de magnitud mayor que el esperado. Nuevamente, se presenta el aumento del error característico, por falta de información, a $\ell \approx 225$.

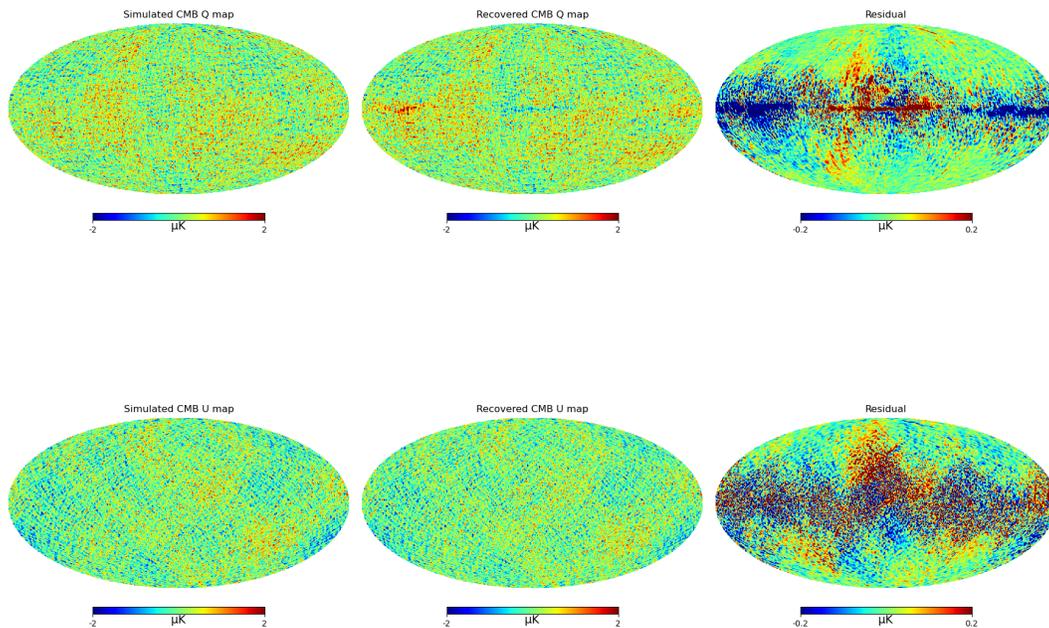


Figura 3.7: Representaciones gráficas de los mapas simulado, recuperado y residual del CMB para el experimento de LiteBIRD. Estas figuras corresponden al experimento entrenado sin variación y testado sin variación.

Comparación de los resultados de entrenamiento sin variación y test con variación con los resultados de los modelos con y sin variación de los *foregrounds*

Estos resultados nos permiten obtener información esencial para obtener cuál es el mejor modelo de entrenamiento de red neuronal que se puede realizar con esta red.

Tratando los resultados a nivel de mapa, se puede observar que la limpieza de los mapas de Q y U en el caso del entrenamiento sin variabilidad cuando se realiza el test con un conjunto con variación de los *foregrounds* es mucho peor. En estos mapas, la emisión de *foregrounds* residual ocupa un área muy grande en comparación con el entrenamiento con variación y test con variación, y el entrenamiento sin variación y test sin variación. En estos últimos casos, los residuos están confinados principalmente en el plano galáctico.

Si ahora se observa el espectro de potencias, se puede ver cómo los desplazamientos tanto a multipolos altos como a multipolos bajos en el modo B y E son mucho mayores en el entrenamiento sin variación y test con variación. Esto sobre todo se observa en el modo B auto que tiene un desplazamiento aproximadamente 10 veces mayor en su punto final. Además, el modo B cross del caso del entrenamiento sin variación y test con variación no se consigue recuperar.

Es decir, aunque inicialmente parecía que la eficiencia del modelo sin variación de *foreground* era mejor que en el caso de los *foregrounds* con variabilidad, al someter este entrenamiento a un test en una situación más realista se observa que no es el entrenamiento óptimo. Esto es debido a que, como se ha comentado anteriormente, actualmente, los modelos de *foreground* no están bien definidos. Cabe la posibilidad de que, al realizar una medida en LiteBIRD, este *foreground* fuera ligeramente distinto al implementado en el entrenamiento lo que haría que se obtuvieran resultados que distan mucho de la realidad.

Por tanto, la utilización de un modelo de *foreground* fijo para el entrenamiento sólo sería posible realizarla en contextos futuros dónde se conozca bien el *foregorund* al que se enfrenta la red o en simulaciones, en las que tiene sentido usar un *foreground* fijo, siendo difícilmente trasladable a la vida real en el contexto actual.

3.2.2. Utilización de los modelos s5 y d10

En esta ocasión, se procede a comprobar modelos más sofisticados de *foreground* del CMB, los modelos “s5” y “d10”. Para este caso, es necesario modificar el código inicial del *paper*[1], pues utiliza una modificación de PySM2, la cual no presenta los modelos “s5” y “d10”. Es decir, es necesario modificar el código para introducir PySM3 que contiene modelos más realistas de *foregrounds*. Esto se ha realizado implementando PySM3 en el código original, modificando las partes necesarias. Las variaciones de los *foregrounds* implementadas serán las mismas que para “s1” y “d1”.

En este caso, se entrena la red neuronal con un número de épocas de: $iterations = 3 \cdot 10^4$. Una vez entrenada la red neuronal para realizar el proceso de limpieza, se procede a introducir el conjunto de test en la red para obtener los mapas del CMB limpios. Estos resultados dan una desviación media absoluta de $MAD = (0.014 \pm 0.002) \mu K$ y $MAD = (0.011 \pm 0.002) \mu K$ para Q y U correspondientemente.

Observando la comparación de los distintos mapas en la Figura 3.9, se puede observar cómo, al igual que para los modelos “s1”, “d1” y “a2” con variabilidad, la red neuronal entrenada no es capaz de eliminar correctamente la emisión de *foreground* galáctico en su totalidad. Fuera del plano galáctico, se recupera eficazmente cuando se realiza el entrenamiento y el test de manera coherente.

En lo que respecta al espectro de potencias, Figura 3.8 columna 2, se ha conseguido recuperar de manera muy adecuada tanto el modo E como el B, a excepción del modo B cros-correlacionado. En este modo B, se puede observar cómo, a multipolos bajos, vuelve a aparecer el efecto del *foreground* residual que lleva a una sobreestimación del espectro de potencias recuperado. A multipolos altos, por su parte, se puede observar, como en los resultados anteriores, un aumento brusco del error correspondiente al espectro de potencias recuperado debido a la pérdida de información, pues la señal es comparable al ruido. En este espectro, este aumento se da a $\ell \approx 200$.

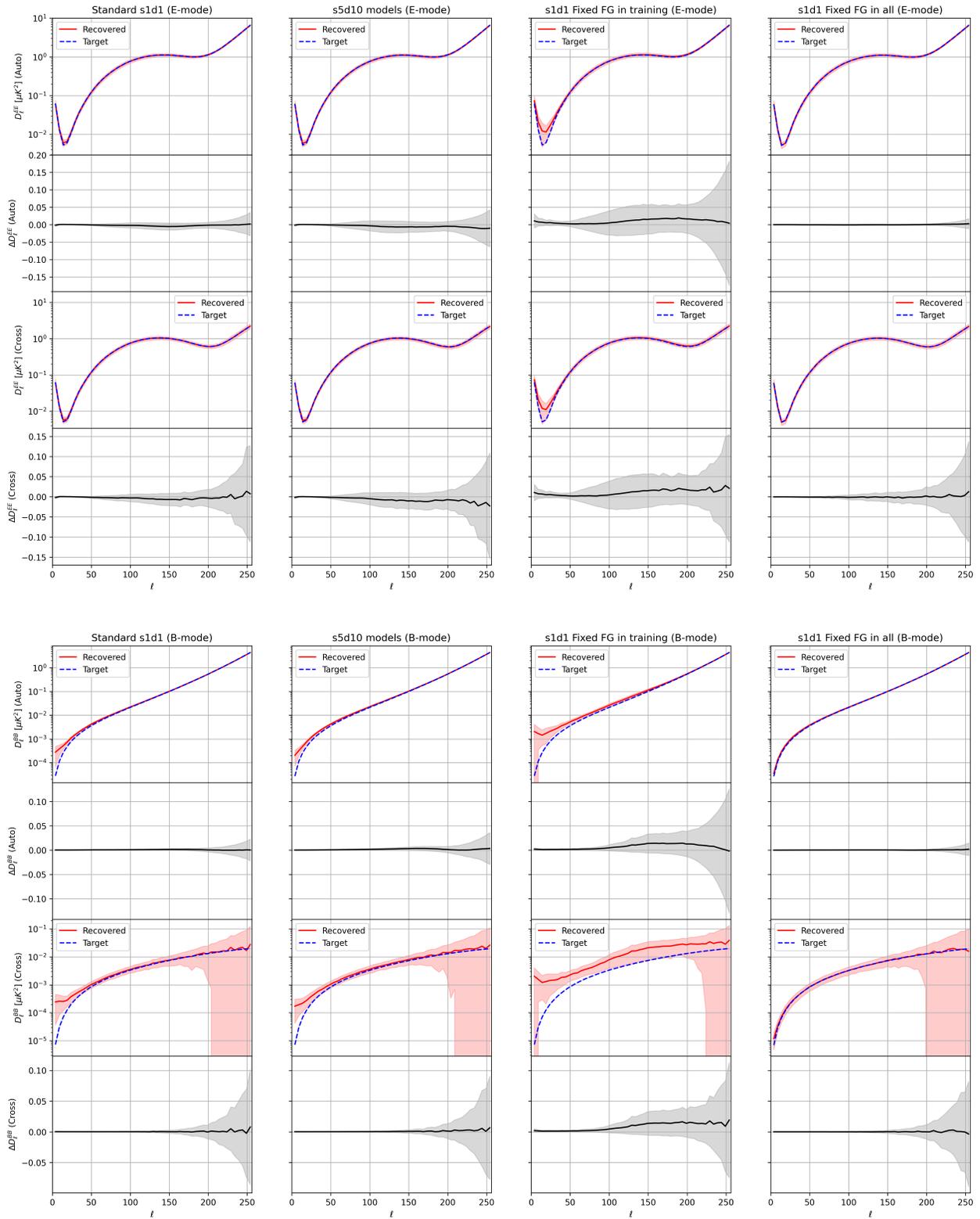


Figura 3.8: Espectro de potencias del modo E y B del CMB. El panel superior de cada figura corresponde con el espectro de potencias del CMB, mientras que el panel inferior muestra la estimación del error asociada.

Es de comentar que, aunque se recupere de una manera adecuada el espectro de potencias, se puede observar cómo al haber utilizado modelos de *foreground* más complejos, se ha disminuido también el número de multipolos que se han podido recuperar antes de la caída.

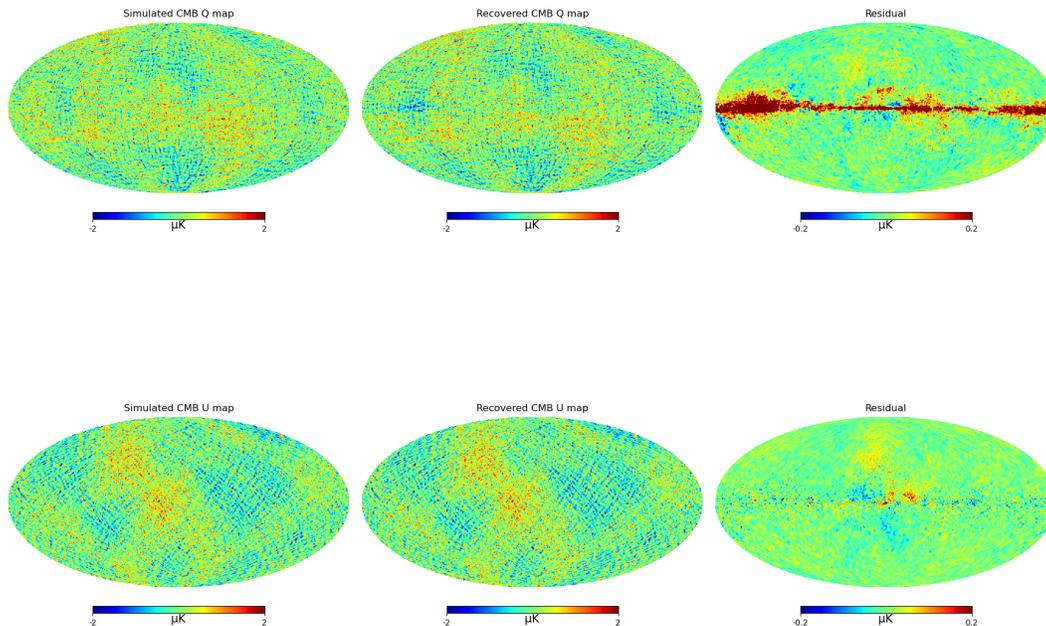


Figura 3.9: Representaciones gráficas de los mapas simulado, recuperado y residual del CMB para el experimento LiteBIRD utilizando los modelos “s5” y “d10” de PySM3. La figura superior corresponde con el mapa de Q, mientras que la inferior corresponde con el mapa de U.

3.3. Planck

Para finalizar con los resultados, se ha considerado importante realizar simulaciones de un experimento del que ya se tienen resultados, como es el experimento de Planck. En esta ocasión, se ha decidido simular Planck a $N_{side} = 128$. Para entrenar a la red neuronal ha sido necesario utilizar los siguientes hiperparámetros: $\alpha = 0.01$, $batch\ size = 1$ e $iterations = 3 \cdot 10^4$. Además, el número de mapas utilizado para entrenar la red de manera correcta ha sido de: 1200 mapas de cielo y 400 de ruido para el conjunto de entrenamiento; 400 mapas de cielo y 400 mapas de ruido tanto para el conjunto de validación como para el de test. Como se puede observar, este ha sido el cielo más difícil de limpiar, pues se ha necesitado un mayor número de mapas. Debido a que se trata del cielo más ruidoso como se puede observar en la Tabla 3.1.

Similar a como ocurre en LiteBIRD, para llevar a cabo este análisis se ha disminuido de manera considerable la resolución, inicialmente de $N_{side} = 2048$, debido nuevamente a que al

tener un mapa a todo el cielo el proceso de compilación sería demasiado lento. Además, en lugar de utilizar datos reales, se han utilizado simulaciones y se han obtenido los modos E y B correlados, pues en el *paper*[19] se supuso que no se iba a obtener ningún tipo de información relevante.

Para simular el *foreground* de Planck, se han utilizado los modelos “s1”, “d1” y “a2” con la misma variación que en LiteBIRD. Una vez simulado el cielo a estudiar y entrenada la red, se ha comprobado la eficiencia de la CNN utilizando el conjunto de test. Para esta red, se ha obtenido una desviación media absoluta de $MAD = (0.030 \pm 0.007) \mu\text{K}$ y $MAD = (0.018 \pm 0.005) \mu\text{K}$, para Q y U respectivamente.

Observando los mapas esféricos recuperados en la Figura 3.10, se puede observar como limpia correctamente el cielo, excepto en las zonas correspondientes al plano galáctico y entorno al centro galáctico. Hecho que sucede de manera similar al recuperar los mapas de LiteBIRD. Debido a que en estas zonas la emisión de *foreground* es comparable a la del CMB y se presentan estructuras más complejas de *foreground*.

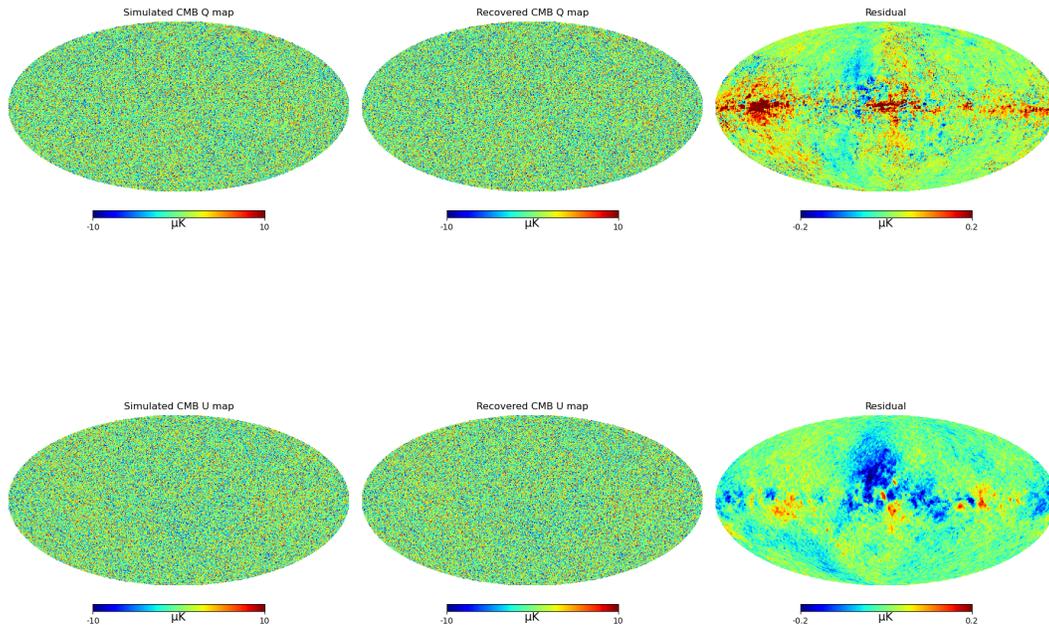


Figura 3.10: Representaciones gráficas de los mapas simulado, recuperado y residual del CMB para el experimento Planck. La figura superior corresponde con el mapa de Q, mientras que la inferior corresponde con el mapa de U.

En cuanto al espectro de potencias, se puede observar en la Figura 3.11 que para el modo B y E auto se recupera este espectro de manera adecuada. Para el modo E cros, se puede observar que se obtiene un espectro adecuado hasta un cierto multipolo, a $\ell \approx 175$. A medida que

se disminuye la escala angular, multipolos mayores, se puede observar como la discrepancia se hace mayor debido a que la señal comienza a disminuir y el ruido a dominar. Ya no hay información suficiente debido al efecto de la relación señal-ruido y el espectro de potencias cae bruscamente.

El modo B cros, por su parte, no se recupera correctamente en ningún punto multipolo. Esto es debido a que el ruido en todos los puntos es mucho más grande que la señal surgiendo una pérdida de información constante. La señal que se observa, seguramente sea debida a fluctuaciones estadísticas en las dos mitades del mapa. Así, se puede observar en el espectro de potencias que todos los errores cruzan el cero. Estos resultados tienen sentido en el contexto del experimento de Planck, el cual no era capaz de detectar los modos B debido al gran ruido instrumental presente y detectaba el modo E hasta un cierto multipolo, menor que para LiteBIRD o CMB-S4, experimentos más modernos con un ruido instrumental mucho menor.

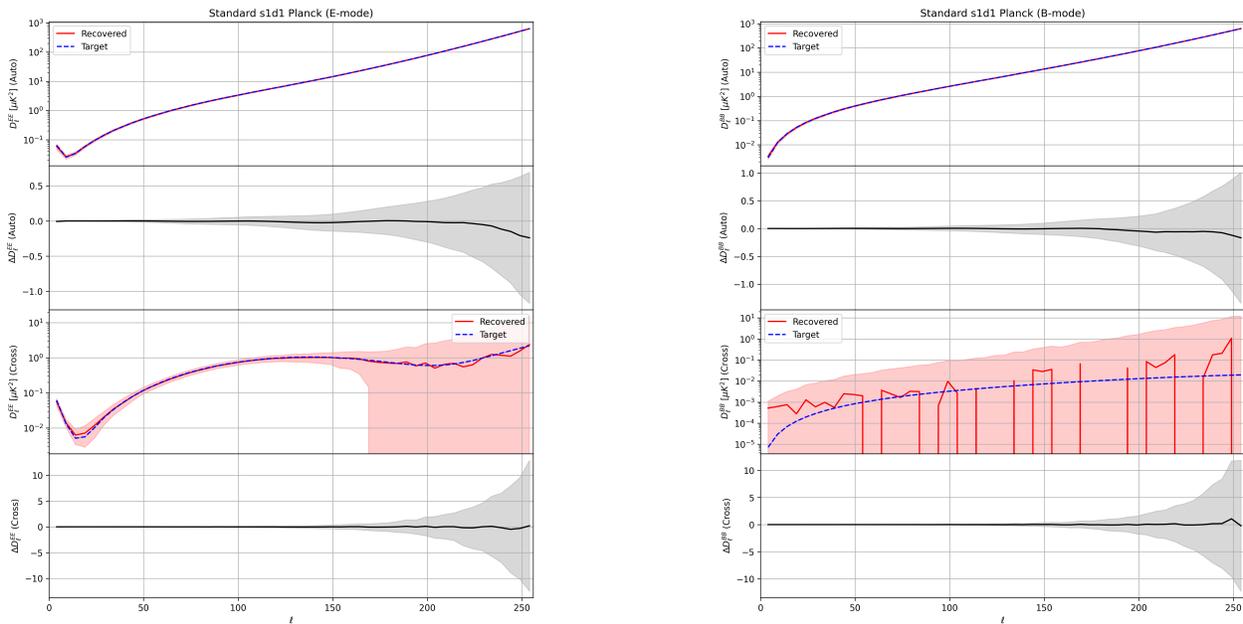


Figura 3.11: Espectro de potencias del modo E y B del CMB en el experimento Planck. El panel superior de cada figura corresponde con el espectro de potencias del CMB, mientras que el panel inferior muestra la estimación del error asociada. Por otra parte, es preciso señalar que las figuras inferiores muestran el espectro de potencias cross-correlacionado.

Aunque sea complicado comparar los espectros de potencias de LiteBIRD y Planck con el del experimento CMB-S4, pues no se ha utilizado el mismo N_{side} , hay algunas características a nivel de mapa que es importante destacar. En primer lugar, se puede observar que, debido al número de parches escogido, el tiempo de compilación varía mucho entre mapas. Así, mientras que en LiteBIRD y Planck este tiempo se hace muy grande (del orden de una semana, en algunas ocasiones) al aumentar el tamaño de lote, en el experimento CMB-S4, al sólo escoger un parche, no se hace tan elevado.

Por otra parte, se puede observar en los resultados obtenidos a nivel de mapa esférico como es mucho más sencillo limpiar un parche del cielo (como en el caso del CMB-S4) que todo el cielo, habiendo una menor cantidad de mapa residual. También es importante comentar que todos los resultados obtenidos son compatibles con el mapa *input* simulado debido a que la diferencia es compatible con cero, está dentro de 2σ

Por último, es también de comentar la importancia del ruido blanco instrumental, como se puede observar a lo largo de los resultados, cuanto mayor ruido se tenga en la medida, peor se va a recuperar el modo B cruz del espectro de potencias. Así, mientras que en el experimento CMB-S4 se puede obtener el modo B hasta $2N_{side}$, en el caso del experimento Planck, el ruido instrumental es tan grande que hace imposible obtener cualquier información sobre este modo B.

Es crucial, además, darse cuenta de que si se comparan los resultados de LiteBIRD con los de PLInck, usando en ambos modelos de PySM con variación de los *foregrounds*, los primeros son mucho mejores, a pesar de que el entrenamiento para Planck tiene muchas más simulaciones.

Esto es debido a que la sensibilidad de LiteBIRD es mucho mayor, pues es un experimento más preciso en el que se quieren recuperar los modos B primordiales del CMB y, por tanto, se necesita un menor ruido. Este ruido, a su vez está relacionado con los multipolos, ℓ , debido a que a mayor multipolo, se necesita una mayor resolución angular, puesto que se corresponden con escalas angulares menores.

Por ello, tiene sentido que, como se observa en las Figuras 3.8, izquierda y 3.11, la recuperación de Planck a escalas angulares pequeñas sea menor que la de LiteBIRD, pues si se pretende llegar a escalas angulares menores se necesita un menor ruido, como en el caso de LiteBIRD.

4

Conclusiones

A lo largo de todo este proyecto, se ha podido observar como la utilización de CNN da lugar a unos resultados precisos y fiables de mapas limpios del Fondo Cósmico Microondas. Esto se ha conseguido utilizando simulaciones realistas de datos del CMB con *foregrounds* con distintos modelos de sincrotrón y polvo térmico de diferentes grados de complejidad.

Es decir, esta red neuronal es capaz de limpiar eficazmente mapas de diferentes experimentos, utilizando parches o mapas a todo el cielo con una generalización adecuada para diferentes modelos. Incluso con la presencia de un ruido blanco gaussiano en estos mapas.

Estas redes, además son capaces de limpiar cielos con una cierta variabilidad en los *foregrounds*. Es importante destacar que con los métodos convencionales de separación de componentes, hasta la fecha no se había tenido en cuenta de manera sistemática las incertidumbres en los parámetros de los *foregrounds*. En éstos, si se cambiaba el cielo había que volver a ajustar el modelo. En cambio, con redes neuronales simplemente hay que pasar el mapa a una red neuronal convolucional con entrenamiento variable. Es decir, se ha obtenido un método más eficiente para obtener un mapa del CMB con buena precisión, aunque todavía hace falta demostrar si se puede recuperar el modo B primordial.

Asimismo, la utilización de esta red neuronal también presenta algunos inconvenientes. En primer lugar, es de comentar que la utilización de una CNN para el análisis de mapas esféricos es un problema en sí. Esto es debido a que este tipo de redes trabaja con mapas planos lo que hace que se tenga que pasar obligatoriamente de mapas esféricos a mapas planos lo que genera distorsiones que pueden afectar a la recuperación del mapa además de residuos en los bordes de los parches, como se comentó en la sección 2.1.3. Esto además ha sido comentado en relación al experimento CMB-S4.

Para mejorar estos resultados, se podría aplicar una *Spherical CNN*[26], red neuronal con-

volucional que trabaja con mapas esféricos, como su propio nombre indica. Una solución alternativa, sería realizar giros de unos 45 grados (la mitad de la escala angular de un parche de $N_{side} = 1$) a lo largo del entrenamiento y promediar en los bordes, lo que disminuiría los efectos de borde presentes en el mapa.

Por otra parte, NaMaster no es la mejor opción en Python para obtener el espectro de potencias a multipolos bajos porque, a pesar de ser insesgado, presenta una alta varianza a estos multipolos, en particular, por debajo de $\ell = 15$. En estos casos, se podrían implementar los estimadores de *Quadratic Maximum Likelihood*[27] en futuros análisis para obtener resultados más precisos a gran escala.

Por último, es de comentar la dificultad de verificar si esta red neuronal se puede generalizar también a resultados reales de los diferentes experimentos, pues es necesario tener un buen control de los efectos sistemáticos. En particular, el ruido podría no comportarse como un ruido blanco. Esto puede hacer necesario modificar la red neuronal convolucional para limpiar de manera correcta la emisión de *foregrounds*.

En conclusión, la utilización de las redes neuronales convolucionales en el contexto de cosmología y, especialmente, en el contexto de la limpieza del cielo para obtener la emisión del CMB, tiene un futuro muy prometedor. Sin embargo, es necesario seguir progresando para crear redes neuronales que puedan limpiar esta emisión de manera aún más eficaz y así, en un futuro, poder verificar si la teoría de la Inflación Cósmica es correcta.

Referencias

- [1] Y.-P. Yan, S-Y. Li, G.-J. Wang, Z. Zhang, and J.-Q. Xia. CMBFSCNN: Cosmic microwave background polarization foreground subtraction with convolutional neural network, 2024. Disponible en: <https://arxiv.org/abs/2406.17685>.
- [2] The Planck Collaboration. Planck 2018 results: I. overview and the cosmological legacy of planck. *Astronomy and Astrophysics*, 641:A1, September 2020. DOI: 10.1051/0004-6361/201833880. Disponible en: <https://doi.org/10.1051/0004-6361/201833880>.
- [3] Scott Dodelson and Fabian Schmidt. *Modern Cosmology*. Academic Press, 2 edition, 2020.
- [4] X. Dupac and M Giard. Map-making methods for cosmic microwave background experiments. *Monthly Notices of the Royal Astronomical Society*, 330(3):497–505, March 2002. DOI: 10.1046/j.1365-8711.2002.05057.x. Disponible en: <https://doi.org/10.1046/j.1365-8711.2002.05057.x>.
- [5] D. Shen, E. Schaan, and S. Ferraro. CMB lensing power spectrum without noise bias. *Physical Review D*, 110(4), August 2024. DOI: 10.1103/physrevd.110.043523. Disponible en: <http://dx.doi.org/10.1103/PhysRevD.110.043523>.
- [6] Peter Schneider, C. S. Kochanek, Joachim Wambsganss, Swiss Society of Astronomy, and Astrophysics. Advanced Course (Les Diablerets (Suiza)) 33^o. *Gravitational lensing: strong, weak and micro : Saas-Fee Advanced Course 33*, volume 33. Springer, Berlin, 2006.
- [7] J. M. Diego. Constraining the abundance of primordial black holes with gravitational lensing of gravitational waves at ligo frequencies. *Physical Review D*, 101(12), June 2020.
- [8] Daniel Baumann. *Cosmology*. Cambridge University Press, Cambridge, UK, 2022.

- [9] The Planck Collaboration. Planck 2013 results. XII. diffuse component separation. *Astronomy and Astrophysics*, 571:A12, October 2014. DOI: 10.1051/0004-6361/201321580. Disponible en: <http://dx.doi.org/10.1051/0004-6361/201321580>.
- [10] The Planck Collaboration. Planck 2018 results: IV. diffuse component separation. *Astronomy and Astrophysics*, 641:A4, September 2020. DOI: 10.1051/0004-6361/201833881. Disponible en: <http://dx.doi.org/10.1051/0004-6361/201833881>.
- [11] Ruth Durrer. *The Cosmic Microwave Background*. Cambridge University Press, Cambridge, 2 edition, 2020.
- [12] T. L. Smith, E. Pierpaoli, and M. Kamionkowski. New cosmic microwave background constraint to primordial gravitational waves. *Physical Review Letters*, 97(2), July 2006. DOI: 10.1103/physrevlett.97.021301. Disponible en: <http://dx.doi.org/10.1103/PhysRevLett.97.021301>.
- [13] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2):759, April 2005. DOI: 10.1086/427976. Disponible en: <https://dx.doi.org/10.1086/427976>.
- [14] P. J. E. Peebles and B. Ratra. The cosmological constant and dark energy. *Reviews of Modern Physics*, 75(2):559–606, April 2003. DOI: 10.1103/revmodphys.75.559. Disponible en: <http://dx.doi.org/10.1103/RevModPhys.75.559>.
- [15] The Planck Collaboration. Planck 2015 results - XIII. cosmological parameters. *AA*, 594:A13, 2016. DOI: 10.1051/0004-6361/201525830. Disponible en: <https://doi.org/10.1051/0004-6361/201525830>.
- [16] R. M. Sullivan, L. T. Hergt, and D. Scott. Methods for CMB map analysis, 2024. Disponible en: <https://arxiv.org/abs/2410.12951>.
- [17] J. Caldeira et al. DeepCMB: Lensing reconstruction of the cosmic microwave background with deep neural networks. *Astronomy and Computing*, 28:100307, July 2019. DOI: 10.1016/j.ascom.2019.100307. Disponible en: <http://dx.doi.org/10.1016/j.ascom.2019.100307>.
- [18] Anna Bosch Rué, Jordi Casas Roma, and Toni Lozano Bagén. *Deep Learning: Principios y fundamentos*. Editorial UOC, Barcelona, España, 2020.
- [19] Y.-P. Yan, G.-J. Wang, S.-Y. Li, and J.-Q. Xia. Recovering cosmic microwave background polarization signals with machine learning. *The Astrophysical Journal*, 947(1):29, April 2023. DOI: 10.3847/1538-4357/acbf4. Disponible en: <http://dx.doi.org/10.3847/1538-4357/acbf4>.
- [20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. Disponible en: <https://arxiv.org/abs/1412.6980>.

- [21] Jesús Bobadilla Sancho. *Machine Learning y Deep Learning*. RA-MA S.A. Editorial y Publicaciones, Madrid, España, 2020.
- [22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. Disponible en: <https://arxiv.org/abs/1502.03167>.
- [23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. Disponibl en: <https://arxiv.org/abs/1505.04597>.
- [24] The CMB-S4 Collaboration. CMB-S4 forecasts for constraints on f_{nl} through μ -distortion anisotropy. *Phys. Rev. D*, 108:103536, November 2023. DOI: 10.1103/PhysRevD.108.103536. Disponible en: <https://link.aps.org/doi/10.1103/PhysRevD.108.103536>.
- [25] The LiteBIRD collaboration. Probing cosmic inflation with the LiteBIRD cosmic microwave background polarization survey. *Progress of Theoretical and Experimental Physics*, 2023(4), November 2022. DOI: 10.1093/ptep/ptac150. Disponible en: <http://dx.doi.org/10.1093/ptep/ptac150>.
- [26] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling. Spherical CNNs, 2018. Disponible en: <https://arxiv.org/abs/1801.10130>.
- [27] M. Tegmark. How to measure cmb power spectra without losing information. *Physical Review D*, 55(10):5895–5907, May 1997.
- [28] Barbara Ryden. *Introduction to Cosmology*. Cambridge University Press, Cambridge, UK, 2 edition, 2017.
- [29] S. M. Leach et al. Component separation methods for the PLANCK mission. *Astronomy amp; Astrophysics*, 491(2):597–615, October 2008. DOI: 10.1051/0004-6361:200810116. Disponible en: <http://dx.doi.org/10.1051/0004-6361:200810116>.
- [30] The Planck Collaboration. Planck 2018 results: V. CMB power spectra and likelihoods. *Astronomy amp; Astrophysics*, 641:A5, September 2020. DOI: 10.1051/0004-6361/201936386. Disponible en: <http://dx.doi.org/10.1051/0004-6361/201936386>.
- [31] D. Scott and G. F. Smoot. Cosmic microwave background mini-review. *Particle Data Group, Physics Letters B*, 667:1–1340, 2012. Disponible en: <https://pdg.lbl.gov/2022/reviews/rpp2022-rev-cosmic-microwave-background.pdf>.
- [32] The Planck Collaboration. Planck 2015 results: I. overview of products and scientific results. *Astronomy amp; Astrophysics*, 594:A1, September 2016. DOI: 10.1051/0004-6361/201527101. Disponible en: <http://dx.doi.org/10.1051/0004-6361/201527101>.
- [33] R. K. Sachs and A. M. Wolfe. Perturbations of a Cosmological Model and Angular Variations of the Microwave Background. *The Astrophysical Journal*, 147:73, January 1967. DOI: 10.1086/148982. Disponible en: <https://ui.adsabs.harvard.edu/abs/1967ApJ...147...73S>.

- [34] David Balbás. A deep learning approach for the detection of point sources in the cosmic microwave background, September 2019. Disponible en: <http://hdl.handle.net/10902/17861>.
- [35] M. Rahimi and C. L. Reichardt. Polarization of the cosmic microwave background, 2024. Disponible en: <https://arxiv.org/abs/2412.04099>.
- [36] G.-J. Wang, H.-L. Shi, Y.-P. Yang, J.-Q. Xia, Y.-Y. Zhao, S.-Y. Li, and J.-F. Li. Recovering the cmb signal with machine learning. *The Astrophysical Journal Supplement Series*, 260(1):13, May 2022. DOI: 10.3847/1538-4365/ac5f4a. Disponible en: <http://dx.doi.org/10.3847/1538-4365/ac5f4a>.
- [37] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, Cambridge, MA, 2022.
- [38] The LiteBIRD collaboration. In-flight polarization angle calibration for LiteBIRD: blind challenge and cosmological implications. *Journal of Cosmology and Astroparticle Physics*, 2022(01):039, January 2022. DOI: 10.1088/1475-7516/2022/01/039. Disponible en: <https://dx.doi.org/10.1088/1475-7516/2022/01/039>.
- [39] P. Paykari and A. H. Jaffe. Optimal binning of the primordial power spectrum. *The Astrophysical Journal*, 711(1):1–12, February 2010. DOI: 10.1088/0004-637x/711/1/1. Disponible en: <http://dx.doi.org/10.1088/0004-637x/711/1/1>.
- [40] H. Liu, J. Creswell, S. von Hausegger, and P. Naselsky. Methods for pixel domain correction of eb leakage. *Phys. Rev. D*, 100:023538, July 2019. DOI: 10.1103/PhysRevD.100.023538. Disponible en: <https://link.aps.org/doi/10.1103/PhysRevD.100.023538>.

A

Curvas de entrenamiento

En este apéndice, se pueden observar las distintas curvas de entrenamiento obtenidas para las simulaciones realizadas en este proyecto.

A.1. LiteBIRD

A.1.1. Modelos s1, d1 y a2

Con *foreground* fijo

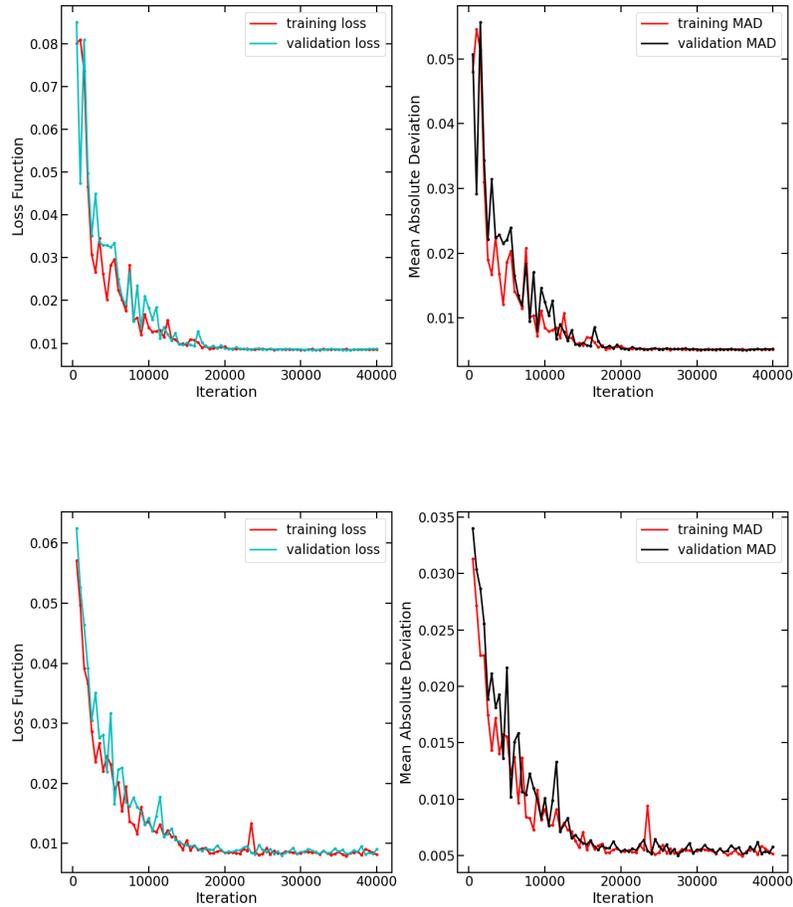


Figura A.1: Representación gráfica de las funciones de pérdida y la MAD para el experimento LiteBIRD con los modelos de *foreground* “s1”, “d1” y “a2”, y con *foreground* fijo. Las gráficas superiores corresponden con el modo Q y las inferiores con el modo U de polarización.

A.1.2. Modelos s5 y d10

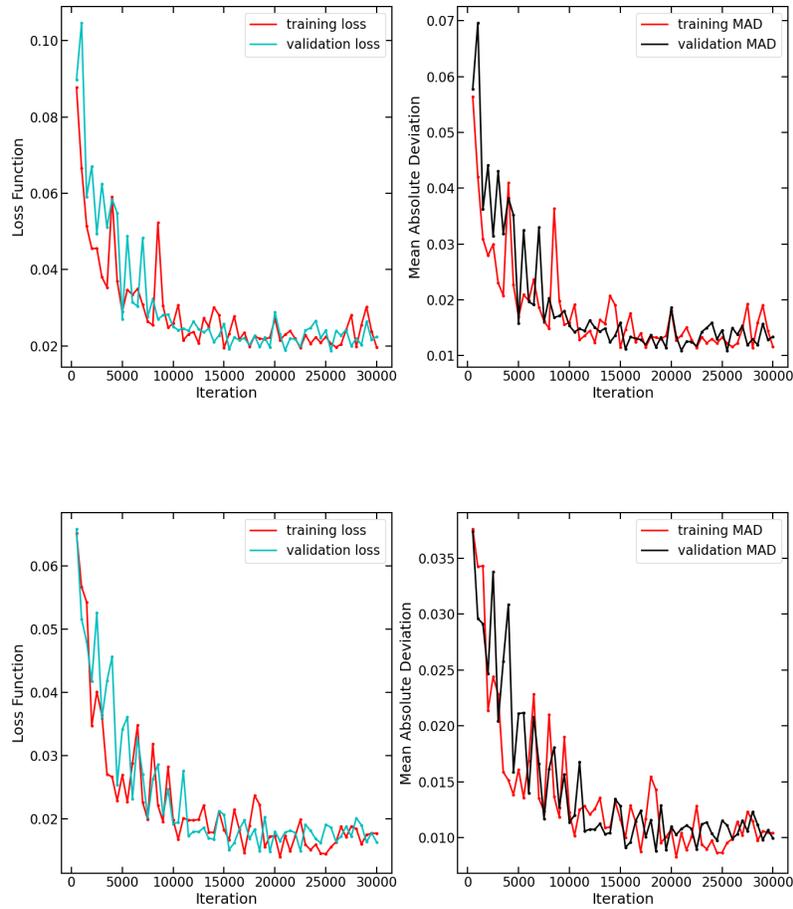


Figura A.2: Representación gráfica de las funciones de pérdida y la MAD para el experimento LiteBIRD con los modelos de *foreground* “s5” y “d10”. Las gráficas superiores corresponden con el modo Q y las inferiores con el modo U de polarización.

A.2. Planck

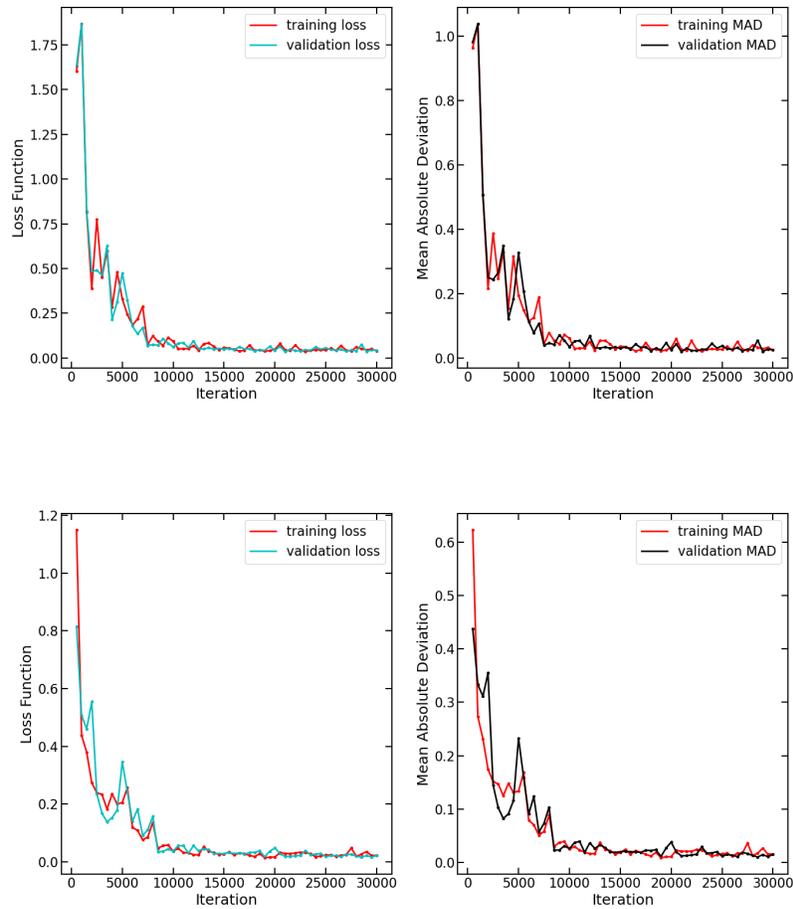


Figura A.3: Representación gráfica de las funciones de pérdida y la MAD para el experimento Planck con los modelos “s1” y “d1”. Las gráficas superiores corresponden con el modo Q y las inferiores con el modo U de polarización.

B

Mapas planos

En este apéndice, se recogen los mapas planos obtenidos en las diferentes situaciones a analizar.

B.1. LiteBIRD

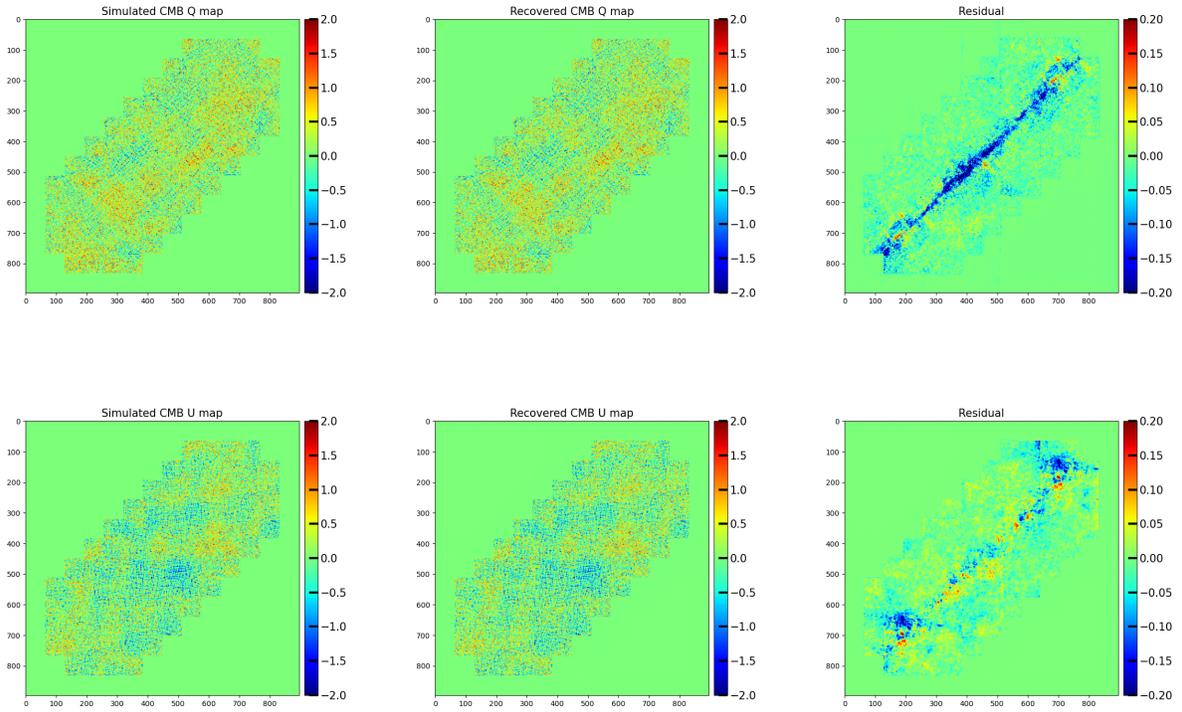


Figura B.1: Representaciones gráficas de los mapas planos del CMB simulado, recuperado y residual para el experimento LiteBIRD con variación de los *foregrounds*. Las figuras superiores consisten en los mapas planos de Q, mientras que las inferiores se corresponden con los mapas planos de U.

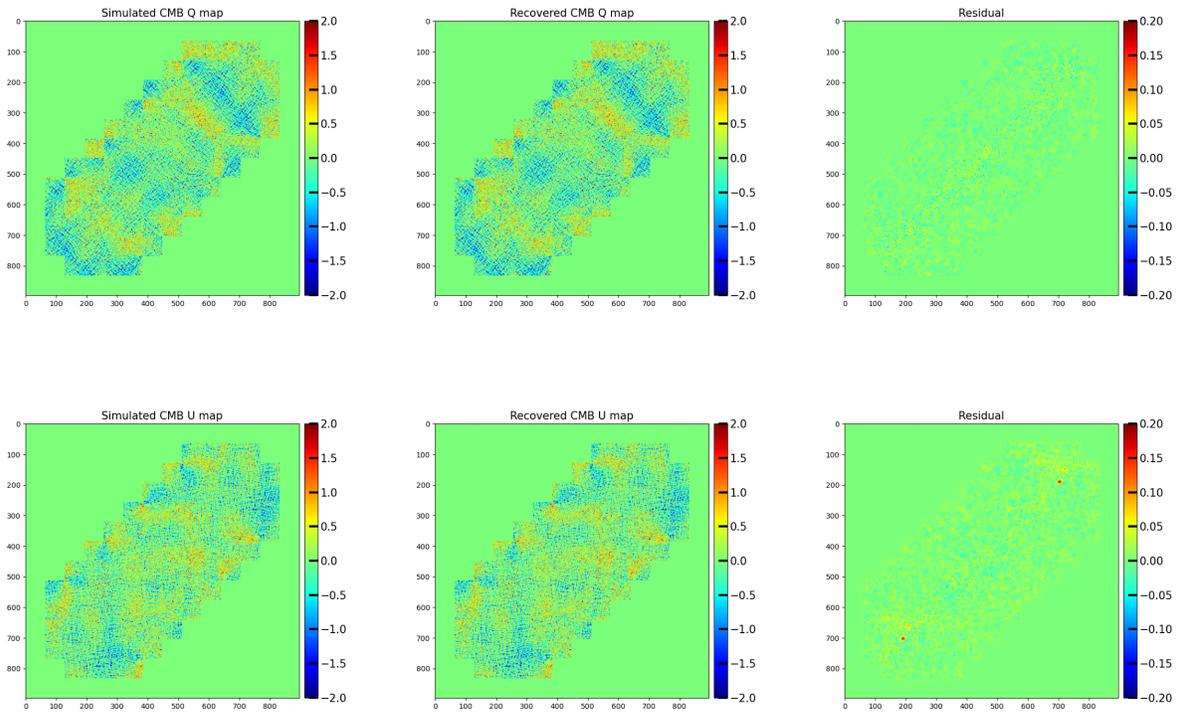


Figura B.2: Representaciones gráficas de los mapas planos del CMB simulado, recuperado y residual para el experimento LiteBIRD con entrenamiento sin variación de los *foregrounds*. Las figuras superiores consisten en los mapas planos de Q, mientras que las inferiores se corresponden con los mapas planos de U.

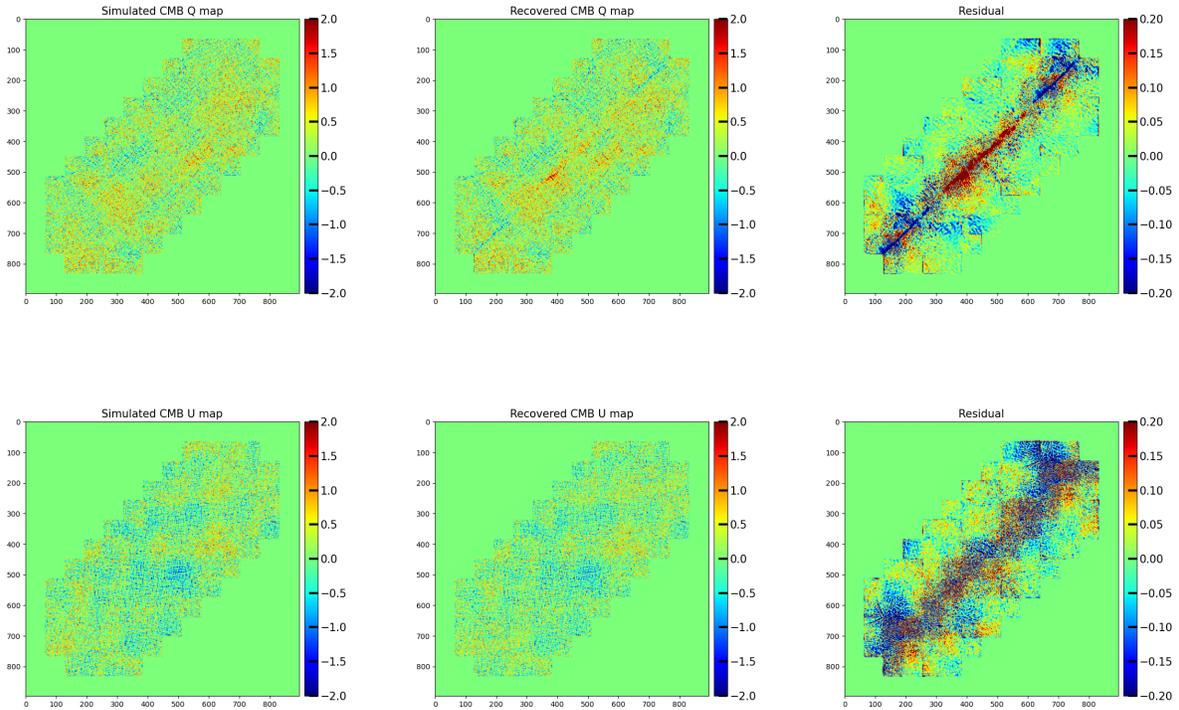


Figura B.3: Representaciones gráficas de los mapas planos del CMB simulado, recuperado y residual para el experimento LiteBIRD con entrenamiento sin variación de los *foregrounds* y testeado con variación. Las figuras superiores corresponden con los mapas planos de Q, mientras que las inferiores se trata de los mapas planos de U.

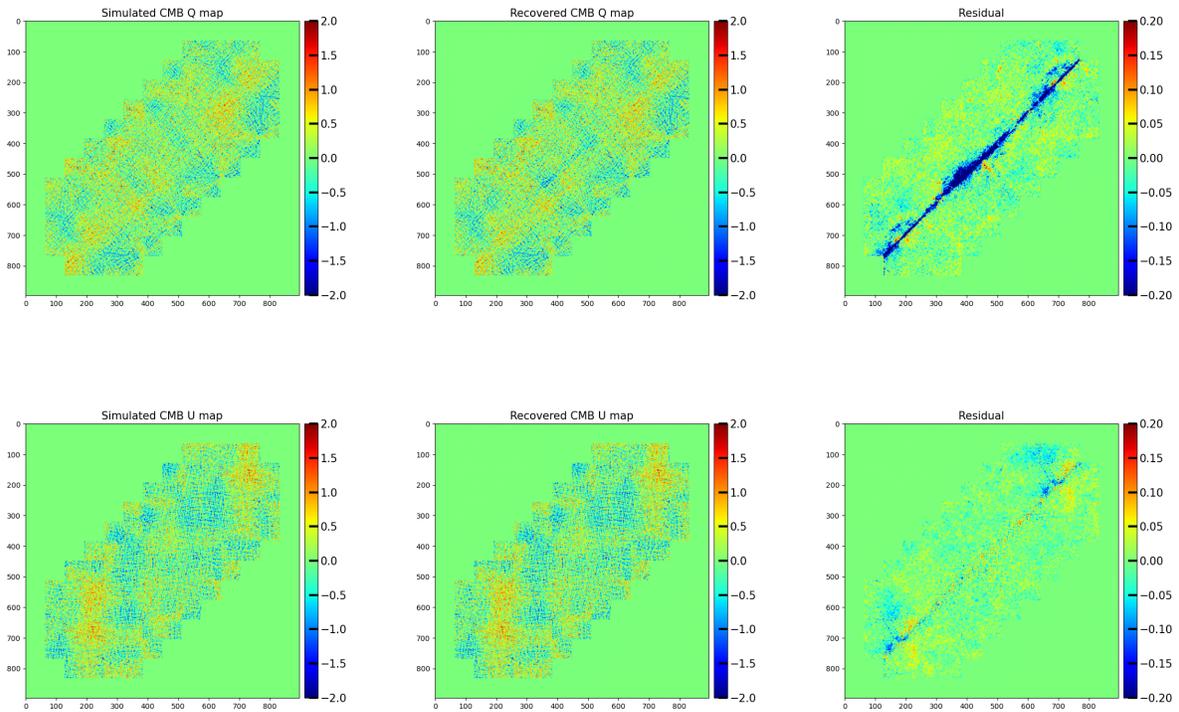


Figura B.4: Representaciones gráficas de los mapas planos del CMB simulado, recuperado y residual para el experimento LiteBIRD utilizando los modelos de *foreground* “s5” y “d10”. Las figuras superiores corresponden con los mapas planos de Q, mientras que las inferiores se trata de los mapas planos de U.

B.2. Planck

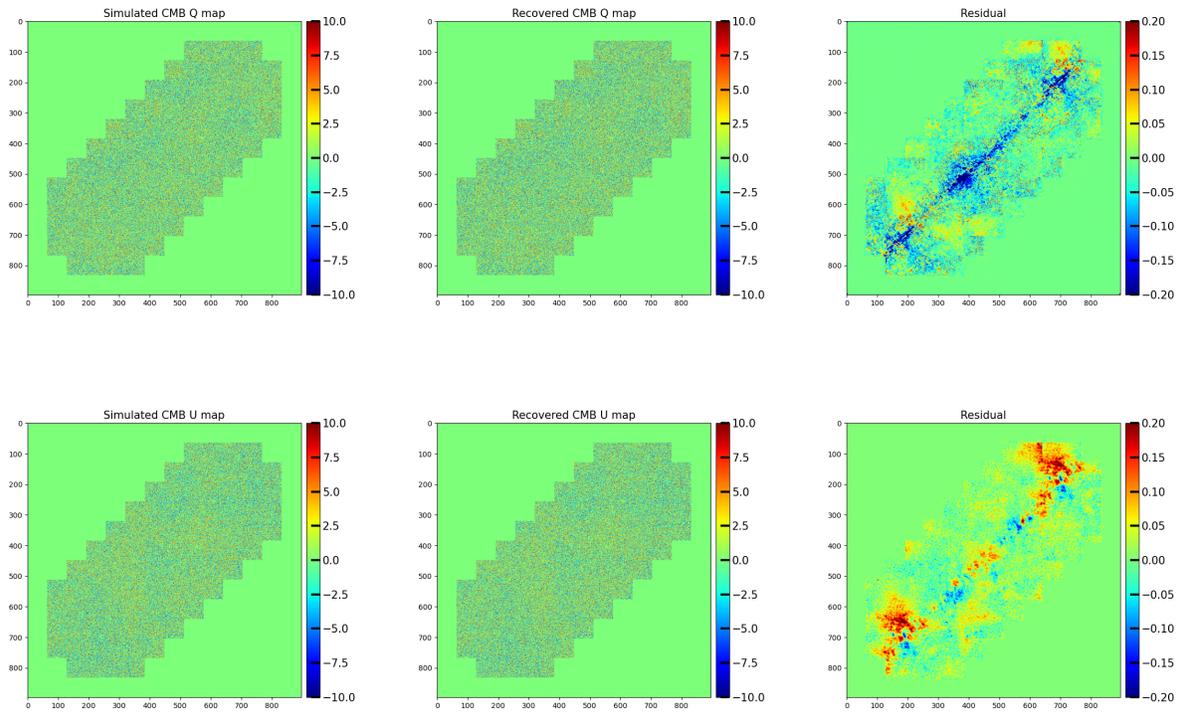


Figura B.5: Representaciones gráficas de los mapas planos del CMB simulado, recuperado y residual para el experimento Planck utilizando los modelos de *foreground* “s1” y “d1”. Las figuras superiores corresponden con los mapas planos de Q, mientras que las inferiores se trata de los mapas planos de U.