Inference of the optical depth to reionization τ from *Planck* CMB maps with convolutional neural networks

Kevin Wolz[1,2], Nicoletta Krachmalnicoff[1,2,3], and Luca Pagano[4,5]

[1] International School for Advanced Studies (SISSA), Via Bonomea 265, 34136 Trieste, Italy
e-mail: kwolz@sissa.it; nkrach@sissa.it
[2] National Institute for Nuclear Physics (INFN) – Sezione di Trieste, Via Valerio 2, 34127 Trieste, Italy
[3] Institute for Fundamental Physics of the Universe (IFPU), Via Beirut 2, 34151 Grignano (TS), Italy
[4] Dipartimento di Fisica e Scienze della Terra, Università degli Studi di Ferrara and INFN – Sezione di Ferrara, Via Saragat 1, 44122 Ferrara, Italy
e-mail: luca.pagano@unife.it
[5] Université Paris-Saclay, CNRS, Institut d'Astrophysique Spatiale, 91405 Orsay, France

ABSTRACT

The optical depth to reionization, τ, is the least constrained parameter of the cosmological Λ cold dark matter (ΛCDM) model. To date, its most precise value is inferred from large-scale polarized cosmic microwave background (CMB) power spectra from the High Frequency Instrument (HFI) aboard the *Planck* satellite. These maps are known to contain significant contamination by residual non-Gaussian systematic effects, which are hard to model analytically. Therefore, robust constraints on τ are currently obtained through an empirical cross-spectrum likelihood built from simulations. In this paper, we present a likelihood-free inference of τ from polarized *Planck* HFI maps which, for the first time, is fully based on neural networks (NNs). NNs have the advantage of not requiring an analytical description of the data and can be trained on state-of-the-art simulations, combining the information from multiple channels. By using Gaussian sky simulations and *Planck* SRoll2 simulations, including CMB, noise, and residual instrumental systematic effects, we trained, tested, and validated NN models considering different setups. We inferred the value of τ directly from Stokes Q and U maps at ~4° pixel resolution, without computing angular power spectra. On *Planck* data, we obtained $\tau_{\rm NN} = 0.0579 \pm 0.0082$, which is compatible with current EE cross-spectrum results but with a ~30% larger uncertainty, which can be assigned to the inherent nonoptimality of our estimator and to the retraining procedure applied to avoid biases. While this paper does not improve on current cosmological constraints on τ, our analysis represents a first robust application of NN-based inference on real data, and highlights its potential as a promising tool for complementary analysis of near-future CMB experiments, also in view of the ongoing challenge to achieve the first detection of primordial gravitational waves.

Key words. cosmic background radiation – cosmological parameters – dark ages, reionization, first stars – methods: data analysis – methods: statistical

1. Introduction

Cosmic reionization, the period in cosmic history that accompanies the ignition of the first stars, is of great interest to both astrophysics and cosmology. At recombination, about 380 000 years after the big bang, free electrons were bound in hydrogen atoms, causing the decoupling of matter from the photon field that we observe today as the cosmic microwave background (CMB). This is when the Universe entered the electrically neutral phase, called "cosmic dark ages." It is presumed that about 200 million years later, cold hydrogen gas had collapsed gravitationally in dark matter halos, forming the first stars. These earliest compact sources of UV radiation heated up the surrounding hydrogen gas, progressively ionizing the whole Universe via bubbles of expanding HII regions. The so-called Gunn-Petersen trough (Gunn & Peterson 1965; Scheuer 1965) in the absorption spectrum of high-redshift quasars indicates the presence of neutral hydrogen in the intergalactic medium (IGM) along the corresponding lines of sight. The detection of this feature in the spectra of some $z > 5.8$ quasars by the Sloan Digital Sky Survey provided the first spectroscopic evidence for reionization (Fan et al. 2000, 2001; Becker et al. 2001). Modern quasar measurements indicate that the epoch of reionization was completed by $z \approx 5.3$ (Qin et al. 2021; Villasenor et al. 2022; Bosman et al. 2022).

Reionization plays a crucial role in cosmology as well. Photons that are emitted during recombination have a finite probability of Compton scattering with free electrons released during reionization. For us as observers, this has two effects: firstly, CMB photons traversing the IGM cause a uniform damping of CMB anisotropies at scales below the cosmological horizon at the epoch of reionization ($\ell > 20$). Secondly, a statistically relevant fraction of CMB photons scatter into our line of sight, carrying a nonzero net polarization observable as secondary anisotropies in the CMB polarization. The first effect reduces the CMB power spectrum amplitude of both unpolarized and polarized components by a factor $e^{-2\tau}$, where τ is the optical depth to reionization, defined as

$$\tau = \int_{t(z_{\rm CMB})}^{t_0} n_{\rm e}\, \sigma_{\rm T}\, c\, {\rm d}t'. \tag{1}$$

Here, $z_{\rm CMB} \approx 1100$ is the time of last scattering between photons and baryons, $n_{\rm e}$ is the electron number density, $\sigma_{\rm T}$ is the Thomson scattering cross section, and c is the speed of light. The second effect introduces large-scale power in the polarized CMB proportional to τ^2, affecting scales larger than the horizon at the epoch of reionization ($\ell < 20$). Full-sky space missions such as WMAP and *Planck* have been able

to measure this "reionization bump" through pixel-based and power-spectrum-based analysis methods. The WMAP nine-year data release cites $\tau = 0.089 \pm 0.014$ (Hinshaw et al. 2013), a value that later turned out to be biased high due to Galactic dust emission (Planck Collaboration XI 2016; Natale et al. 2020). *Planck*'s low-frequency instrument (LFI) polarization data at 70 GHz contain less large-scale systematics than the High Frequency Instrument (HFI) data at 100 GHz and 143 GHz, motivating the Planck Collaboration to perform map-based analysis on LFI data and cross-spectrum analysis on HFI data. The *Planck* 2018 legacy release cites $\tau = 0.063 \pm 0.020$ as inferred from LFI data and 0.051 ± 0.009 from HFI data (Planck Collaboration V 2020). The cross-spectrum analysis method of *Planck* HFI data at 143 GHz and 100 GHz yields the tightest constraint to date, while avoiding the bias arising from uncorrelated noise in the individual frequency channels.

The *Planck* 2018 legacy polarization data products at large scales are known to be affected by residual contamination from instrumental systematic effects. As investigated in Planck Collaboration VI (2014) and Delouis et al. (2019), the most important effects at 143 GHz and 100 GHz are temperature-to-polarization (T-to-P) leakage due the analog-to-digital converter nonlinearity (ADCNL), uncertainties on the detectors' orientation and polarization efficiencies, T-to-P leakage due to bandpass mismatch and inaccurate Galactic foreground modeling, and a varying time constant associated with the heat transfer to the bolometers. In general, these systematic effects follow non-Gaussian statistical distributions and are expected to correlate among different channels, mainly because they are partially sourced by the temperature signal. Several updated mapmaking codes have been published that improve on the systematics cleaning, such as SRoll2 (Delouis et al. 2019), and NPIPE (Planck Collaboration Int. LVII 2020). The SRoll2 algorithm, an upgraded version of the *Planck* Collaboration's SRoll algorithm (Planck Collaboration Int. XLVI 2016), iteratively cleans systematics from *Planck*'s time-ordered data products. Major improvements in SRoll2 encompass a new gain calibration model that accounts for second-order ADCNL, updated foreground templates, and an internal marginalization over the polarization angles and efficiencies for each bolometer. The SRoll2 data products contain a significantly lower level of spurious systematic effects and a dipole residual power reduced by 50% with respect to the *Planck* 2018 legacy data, falling below the noise level. The SRoll2 EE cross-spectrum is dominated by the cosmological signal at all scales that were considered in the analysis ($2 < \ell < 30$).

In spite of the improved cleaning, a small residual contamination remains (mainly due to the second-order ADCNL effect), which may bias cosmological analyses. For their 100×143 GHz EE cross-spectrum analysis of the SRoll2 data products, Pagano et al. (2020) use an empirical likelihood built from realistic simulations (Planck Collaboration V 2020; Gerbino et al. 2020), motivating their choice by the expected non-Gaussianity of the maps and by the difficulty to model residual systematic effects analytically. They obtain $\tau = 0.0566^{+0.0053}_{-0.0062}$ (68% CL) from EE only and $\tau = 0.059 \pm 0.006$ when combining with TT data. Compared with the EE results from the *Planck* 2018 legacy release ($\tau = 0.051 \pm 0.009$), this reduces the uncertainty by ~40% and increases the best-fit τ value by up to 0.9σ. More recently, de Belsunce et al. (2021) applied various likelihood approximation schemes on EE cross-spectrum data from SRoll2 maps, finding results compatible with Pagano et al. (2020), though slightly larger by 0.3σ.

In recent years, neural network (NN)-based approaches to likelihood-free inference underwent a rapid development in cosmology, showing potential as an alternative tool for parameter estimation that does not require the existence of an analytical description of the data, but only relies on numerical simulations to train a regression model. In the general context of cosmology, a variety of machine learning (ML) techniques have been exploited and tested in recent years. Promising tools are being developed for many applications: from cosmic large-scale structure (LSS) simulations (Villaescusa-Navarro et al. 2022), to CMB lensing reconstruction (Caldeira et al. 2019), kinetic SZ detection (Tanimura et al. 2022), or modeling and cleaning of Galactic foregrounds (Jeffrey et al. 2022; Wang et al. 2022; Casas et al. 2022; Krachmalnicoff & Puglisi 2021). NN-based inference of cosmological parameters has seen significant progress in the context of observations of the LSS, where the complexity of the cosmological and astrophysical signals, together with the difficulty in the definition of optimal summary statistics, challenge analytical methods. Up to now, this approach has been tested on simulations (see e.g., Villaescusa-Navarro et al. 2022), with applications on real data still limited in number, although leading to promising results (e.g., Fluri et al. 2019). In this context, CMB data analysis could also benefit from the application of NN-based inference, helping overcome the limitations of traditional methods. This is relevant, for example, for the estimation of parameters affecting the large angular scales, such as the optical depth to reionization, which is critically hampered by the presence of spurious non-Gaussian signals, as outlined above.

This paper represents the first map-level cosmological inference on CMB data that is entirely based on convolutional neural networks (CNNs). We use CNNs to infer the optical depth to reionization τ and its statistical uncertainty from *Planck* multifrequency maps on the 100 and 143 GHz channels at scales $\gtrsim 4°$, having trained and validated our findings on the SRoll2 simulations. Using moment networks (Jeffrey & Wandelt 2020), we infer τ and its statistical uncertainty $\sigma(\tau)$ from a single data set. In particular, we demonstrate:

1. When training the CNN on simulations with realistic, correlated Gaussian noise, we achieve unbiased estimates of τ from maps.
2. Our NN models can effectively combine multifrequency information, recognizing common features across channels, not only to reduce statistical uncertainty but also to diminish the impact of noise and systematic effects.
3. Training on non-Gaussian data is necessary to obtain unbiased results on the SRoll2 test simulations and *Planck* data. Limited by a low number of simulations that contain *Planck* systematics, we were forced to build a retrained model, which increased the error bar on τ by ~30% in exchange for unbiased results.

This paper is structured as follows. We present the simulations and data used in this work in Sect. 2, followed by the neural network inference method in Sect. 3. In order to validate this method, we apply it to a series of simulations and present the results in Sect. 4. We discuss the final results on the *Planck* SRoll2 maps follows in Sect. 5 and conclude in Sect. 6.

2. Simulations and data

The goal of our analysis is to build a NN model able to infer the value of the cosmological parameter τ from *Planck* low-resolution polarization input maps. In particular, in this work we used the SRoll2 maps at 100 and 143 GHz. To achieve this, we needed

a large number of simulations to perform NN training, testing, and validation. We generated simulated maps that include CMB emission, noise, and instrumental systematic effects, as well as possible spurious signals coming from our Galaxy. In this section, we describe the simulations, the data, and the sky masks needed to avoid the highly contaminated Galactic plane region.

2.1. Simulated CMB maps

Polarized CMB anisotropies, observed at the *Planck* noise levels, can be sufficiently well represented by a spin-2 field with Gaussian statistics (Planck Collaboration IX 2020). The TT, TE, and EE power spectra characterize the probability distribution of CMB temperature and polarization anisotropies and can be described by the six parameters of the Λ cold dark matter (ΛCDM) model. Analyses of small-scale temperature data from the *Planck* 2018 legacy release place a 0.5% constraint on the parameter combination $10^9 A_s e^{-2\tau} = (1.88 \pm 0.01;$ Planck Collaboration VI 2020). Varying the two parameters (A_s, τ) simultaneously conditioned on $10^9 A_s e^{-2\tau} = 1.884$, coherent with previous studies (Planck Collaboration Int. XLVI 2016; Planck Collaboration V 2020; Pagano et al. 2020; Planck Collaboration Int. LVII 2020), we used the Boltzmann solver CAMB[1] (Lewis et al. 2000) to generate a lookup table of EE power spectra computed with the ΛCDM model. To build the simulated CMB maps used to train and validate our NN models, we discretized $\tau \in [0.01, 0.13]$ with step size $\Delta\tau = 5 \times 10^{-4}$. Since the other ΛCDM parameters have no substantial impact on polarized CMB spectra at low multipoles, we fixed them to the *Planck* 2018 legacy best-fit values $H_0 = 67.32\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$, $\Omega_b h^2 = 0.02237$, $\Omega_c h^2 = 0.1201$, $n_s = 0.9651$, $m_\nu = 0.06$. From the tabulated power spectra, we uniformly drew 200 000 samples based on which we generated 200 000 pairs of full-sky Stokes Q and U maps using the HEALPix package (Górski et al. 2005). We fixed the Q and U maps' angular pixel resolution by choosing $N_{\mathrm{side}} = 16$ (or a pixel size of $\sim 4°$)[2] and smooth each map with a cosine beam window function (Benabed et al. 2009), in analogy with the procedure used to generate the *Planck* SRoll2 maps (see Sect. 2.4). These large scales retained in our maps correspond to multipoles $\ell \lesssim 50$, where the reionization bump leaves an observable imprint in the CMB EE spectrum.

2.2. Simulated Gaussian noise

Planck maps contain Gaussian instrumental noise which, in pixel space, is well described by the FFP8 covariance matrices (Planck Collaboration XII 2016). We drew samples from them for the *Planck* 100 and 143 GHz polarization channels (Planck Collaboration VI 2014; Planck Collaboration XIII 2016), obtaining 200 000 Gaussian noise maps at pixel resolution $N_{\mathrm{side}} = 16$ for both channels, respectively. We coadded the training maps of CMB and noise to obtain 200 000 *Planck*-like simulations on the full sky, out of which we selected 190 000 for training and 10 000 for validation. For the testing phase, we drew new noise samples in the same fashion as before, but coadded CMB simulations with fixed input values $\tau = 0.05$, 0.06, and 0.07 and different seeds than the ones used for training and validation. In this way, we obtained three sets of 10 000 independent Gaussian test simulations with the fixed input cosmologies.

---

[1] http://camb.info
[2] A HEALPix map has $N_{\mathrm{pix}} = 12 N_{\mathrm{side}}^2$ pixels of the same area $\Omega_{\mathrm{pix}} = \pi/(3 N_{\mathrm{side}}^2)$.

2.3. SRoll2 simulations

The SRoll2 simulations (Delouis et al. 2019) improve on the SRoll simulations published along with *Planck*'s third data release (Planck Collaboration Int. LVII 2020). They are the result of applying the SRoll2 cleaning algorithm to a set of 500 *Planck*-like realistic sky simulations containing modeled noise, foregrounds, and instrument systematics. We chose the SRoll2 simulations as our reference for systematic effects present in the SRoll2 *Planck* data. All simulated maps are cleaned from Galactic foregrounds through a template fitting procedure, as described in Pagano et al. (2020). In order to produce our training set, we started with 400 out of the 500 original SRoll2 simulations containing pairs of Q and U full-sky maps at pixel resolution $N_{\mathrm{side}} = 16$ and two channels corresponding to 100 GHz and 143 GHz. To augment our original SRoll2 simulation set, we randomly drew SRoll2 100 GHz and 143 GHz maps from the original 400 maps (with repetition), keeping corresponding Q and U maps together. This allowed us to assemble a total of 200 000 SRoll2 simulations. After coadding them with CMB simulations, we obtained a set of 200 000 polarized full-sky simulations, used for training and validation. For the testing phase, we made 3×100 copies of 100 unseen SRoll2 maps and coadded them with 10 000 CMB maps with fixed input $\tau = 0.05$, 0.06, and 0.07, respectively. In this way we obtained a set of $3 \times 10\,000$ full-sky SRoll2 test simulations.

2.4. Planck maps

The goal of this work is the analysis of the SRoll2 *Planck* polarization data products (Delouis et al. 2019). They consist of Stokes Q and U maps at the 100 GHz and 143 GHz HFI frequency channels, stored at pixel resolution $N_{\mathrm{side}} = 16$. The *Planck* maps are first smoothed with cosine beam window functions, and then cleaned from foreground contamination through a template fitting procedure (Pagano et al. 2020). Figure 1 shows the map products in Galactic coordinates. We note that close to the Galactic plane, Q and U on both channels are visibly contaminated by residual systematic effects, which we masked prior to the analysis in order to avoid bias. The arc-shaped features in the northern and southern Galactic hemisphere likely indicate residual gain variations caused by the ADCNL systematic effect. As shown by Delouis et al. (2019), these features show lower residual power than the CMB in the 100×143 GHz EE cross-spectrum but may still amount to a nonnegligible bias in cosmological analyses.

2.5. Masks

At low Galactic latitudes, the Milky Way emits polarized foreground radiation which dominates the CMB signal in intensity and polarization. Even after component separation, residuals of this emission need to be excluded from the analysis to avoid biasing cosmological analyses. We therefore applied masks to all maps described in the previous sections. We considered two of the binary polarization masks published in Pagano et al. (2020), retaining sky fractions of $f_{\mathrm{sky}} = \{50\%, 60\%\}$. We smoothed them with Gaussian beams of corresponding FWHM of $\{15°, 16°\}$, and apply a binary threshold, setting all pixels with a value larger than 0.5 to one and all others to zero. This procedure allows us to avoid fuzzy borders and mitigate groups of isolated masked pixels. The smoothed masks are shown in Fig. 2. Our baseline mask in this paper is the $f_{\mathrm{sky}} = 0.5$ smoothed mask, as it retains enough large-scale information to
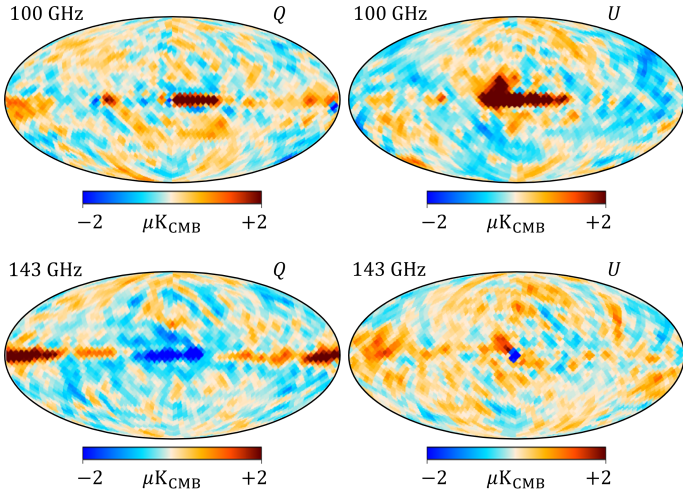
Fig. 1. SRoll2 data products of the *Planck Q* and *U* maps at frequencies 100 GHz and 143 GHz, after component separation, used in this work, displayed in Galactic coordinates.
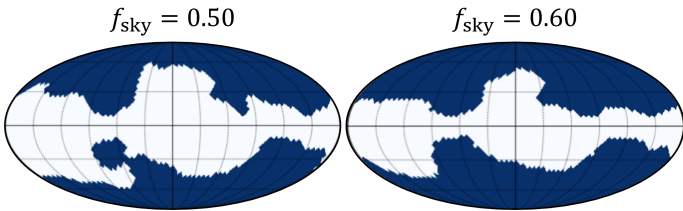

Fig. 2. Smoothed version of the SRoll2 sky masks at sky fractions 50% and 60% used in this paper, displayed in Galactic coordinates.

constrain τ but avoids excessive levels of foregrounds in the Galactic plane.

3. NN inference

In this work, we use CNNs to build simulation-based empirical models to perform cosmological inference. In the following, we describe our CNN implementation and give details on the procedures applied to train and test our model on simulations.

3.1. CNN architecture for τ estimation

CNNs are the industry standard of pattern recognition in two-dimensional images, performing both classification (e.g., identifying families of objects) and regression tasks (e.g., estimating continuous parameters on maps). The success of CNNs in extracting low-dimensional information from visual input is due to a multilayer image filtering algorithm. This typically involves searching for distinct sets of local features in the original image (through convolution) and compressing the data (through so-called pooling layers), going to lower and lower resolution, before inferring the desired summary statistic.

In our case, we want to retrieve information from data projected on the sphere, requiring convolutions on the spherical domain. To this end, we made use of the NNhealpix[3] algorithm which allows to build deep spherical CNNs taking advantage of the HEALPix tessellation. In particular, NNhealpix performs convolution by looking at the first neighbors for each pixel on the map, and average pooling by downgrading the map
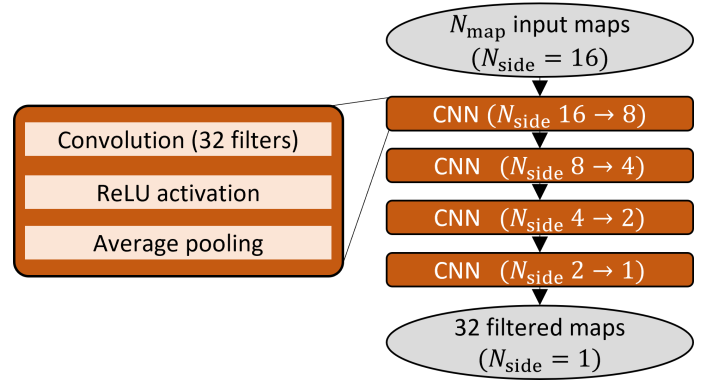

Fig. 3. Schematic of the convolutional layers of the neural network used in this paper. This represents the first part of the architecture, performing image filtering.

resolution (i.e., by going to lower N_{side} parameter). We refer to Krachmalnicoff & Tomasi (2019) for additional details on how the algorithm works, as well as its advantages and disadvantages. In this work, we used NNhealpix in combination with the public keras python package[4] to build our deep CNN architecture, and to perform training, validation, and testing.

The first part of our CNN, performing image filtering, consists of four CNN building blocks, as illustrated in Fig. 3. We accept N_{map} input maps, which in our case represent one or two frequency channels and Stokes Q and U maps, hence $N_{map} = 2$ or 4. Each convolutional layer introduces 32 filters with nine trainable pixel weights, respectively, and is followed by a Rectified Linear Unit (ReLU) activation layer. Mathematically, this means each image pixel p_i undergoes a linear transformation f followed by a nonlinear transformation g

$$p_i \mapsto p_i' = (f \circ g)(p_i), \tag{2}$$

$$f(p_i) = p_i w_0 + \sum_{j=1}^{N_{neigh}(i)} p_{k_j(i)} w_j, \tag{3}$$

$$g(x) \equiv \max(0, x), \tag{4}$$

where $k_j(i)$, $j = 1, \ldots, N_{neigh}(i)$ runs over the indices of all neighboring pixels in the HEALPix map (which can be either seven or eight, depending on the pixel location). Then, an "average pooling" degradation layer reduces the map resolution from N_{side} to $N_{side}/2$, assigning to every low-resolution pixel the average of its four children at the next higher resolution. Up to this point, the application of the four CNN building blocks transform the array of input maps at $N_{side} = 16$ (or $N_{pix} = 3072$ pixels) into an array of 32 filtered maps at $N_{side} = 1$ (or $N_{pix} = 12$ pixels). This represents the image filtering part, meaning the transformation of the original inputs into 32 maximally compressed feature maps that, ideally, retain all the desired (cosmological) information. We still need to "learn" the mapping from theses feature maps to the output numbers τ_{NN} and $\sigma_{NN}(\tau)$ described in the following section. Compression is achieved by two fully connected (or dense) layers.

A fully connected layer is a linear map from M-dimensional input feature space to N-dimensional output feature space and is commonly used for data compression (in which case $N < M$). A fully connected layer of output dimension N is said to contain N neurons associated to a vector of trainable weights that parameterize the layer. In each of its N neurons, a fully connected layer
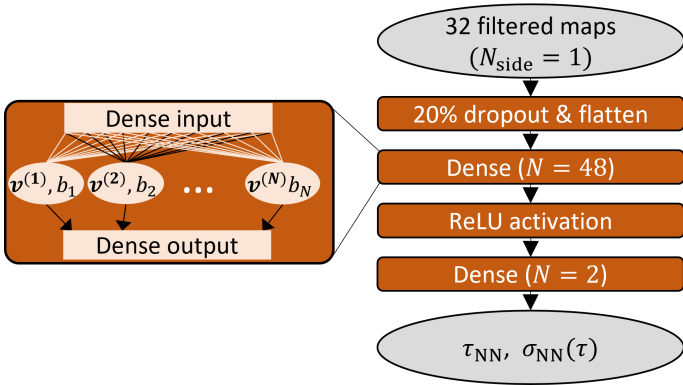

Fig. 4. Schematic of the fully connected layers of the neural network used in this paper. This represents the second part of the architecture, performing data compression.

linearly contracts the input vector x of length M to a number by means of a weights vector $v^{(i)}$,

$$x_i \mapsto x'_{i'} = \sum_{j=1}^{M} v_j^{(i')} x_j. \tag{5}$$

The second part of our CNN, the data compression block, is shown in Fig. 4 and contains a dropout and flattening layer, a fully connected layer with 48 neurons, a ReLU nonlinear activation layer, concluded by a final fully connected layer with two neurons that outputs τ_{NN} and $\sigma_{NN}(\tau)$ as described in the following section. The dropout layer acts as a selective off switch for parts of the following fully connected layer, deactivating at random 20% of its 48 neurons at a time, thus mitigating the overfitting problem common for neural networks (Srivastava et al. 2014). With the described architecture the total number of weights that need to be optimized during training is $N_w \approx 4.7 \times 10^4$.

3.2. Training

When we train a neural network, we effectively tune its many free parameters until the task at hand, such as estimating parameters from maps, would be optimally performed on the training data. In the following, we describe this procedure in detail.

At each training step we passed one batch of $N_{\text{batch}} = 32$ training simulations through the network, meaning we simultaneously considered the results from all simulations that belong to a single batch. Input maps need to be masked with the same mask that is used in the analysis. The output values of the two neurons of the final layer, representing the estimated parameters $\tau_{NNj}, \sigma_{NN}(\tau)_j$ ($j = 1, \ldots, N_{\text{batch}}$), as well as the truth values τ_j, are then inserted into the loss function (Jeffrey & Wandelt 2020)

$$\mathcal{L}\left[\tau, \left(\tau_{NNj}, \sigma_{NN}(\tau)_j\right)\right] =$$
$$\sum_{j=1}^{N_{\text{batch}}} \left[(\tau_j - \tau_{NNj})^2 + \left((\tau_j - \tau_{NNj})^2 - \sigma_{NN}(\tau)_j^2\right)^2\right]. \tag{6}$$

We then updated all N_w network parameters subject to minimizing this loss function. For doing so, we used the ADAM optimizer, a widely used stochastic gradient descent algorithm implemented in keras, for which we found an initial training rate of $LR = 10^{-3}$ and first- and second-moment exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to be appropriate. Repeating the described procedure for the entire training set of size

$N_{\text{train}} = 190\,000$ made up one training epoch[5]. We trained on a maximum of 45 epochs, using the keras callback function ReduceLROnPlateau to allow for learning rates to decrease by a factor of 0.1 if the loss of the validation set did not improve over the course of five epochs. Moreover, the callback function EarlyStopping allows for training to stop after a minimum number of epochs (in our case 20) without improvement in the validation loss. Using both of these callback functions allowed for a faster convergence and suppressed unwanted oscillations in the loss function during the training phase. Training on a 32-core Intel Xeon CPU node took about three hours, while training on eight NVIDIA Tesla V100 GPU cores took about 30 minutes.

3.3. Testing

After training, we fix the neural network parameters, which in principle completes the model building. However, trained NNs may not perform well for two main reasons: firstly, the loss function may have not converged to its global minimum, causing model predictions to be biased. Secondly, the model may overfit the input, meaning that the network learns the training set's features with an excellent accuracy, but fails to make correct predictions on similar, independent test sets. One usually addresses both problems by testing the model's predictions on simulations that have not been fed into the network before. We used 2×3 test sets of 10 000 sky simulations with fixed input $\tau = \{0.05, 0.06, 0.07\}$, described in detail in Sects. 2.2 and 2.3.

We note that, by inferring only τ_{NN} and $\sigma_{NN}(\tau)$, we implicitly assumed Gaussian posteriors, which we exhaustively validated on simulations by checking for biases in the Gaussian mean and variance (see Sect. 4). If, instead, our algorithm had provided an entire, potentially non-Gaussian probability distribution function or higher statistical moments, we would have needed to perform more extensive sanity checks and indicate credible intervals instead of Gaussian standard deviations.

4. Results on simulations

Before arriving at the estimation of τ from the *Planck* SRoll2 data, we considered several setups to train our CNN model, increasing the complexity of the training simulations. This allowed us to get valuable insight into the learning process. In particular, we started by training the CNN on a set of simulations including CMB with Gaussian noise (see Sect. 2.2), either on a single frequency channel, or on two channels. We then moved to simulations including non-Gaussian systematic effects (i.e., SRoll2 simulations), trying different possible strategies to obtain unbiased τ estimates in the presence of complex residuals. Only once we achieved this, we applied our trained model to real *Planck* data. In all the cases presented in this section, we trained and tested the CNNs considering the $f_{\text{sky}} = 0.5$ mask as our reference (see Fig. 2). A summary of all analysis cases, along with their corresponding results tables and figures, can be found in Table 1.

4.1. Gaussian training

As aforementioned, we first tested the ability of our CNN to estimate the value of τ considering only Gaussian noise. These simulations have noise amplitudes and pixel-pixel correlations

---

[5] Among the total 200 000 simulations generated as described in Sect. 2, we actually used 190 000 to optimize the NN's parameters, while we used the remaining 10 000 as a validation set.

Table 1. References to results tables and figures in this paper.

	Gaussian test simulations	SRoll2 test simulations	*Planck* data
Gaussian NN (one channel)	Table 2; Fig. 5	Table 3; Fig. 5	
Gaussian NN (two channels)	Table 2; Figs. 5, 8	Table 3; Figs. 5, 8	Table 5; Figs. 9, 10
HL likelihood	Table 2	Table 3	
SRoll2 training		Table 4	
SRoll2 retraining		Table 4; Fig. 8	Table 5; Figs. 9, 10
Empirical likelihood			Table 5; Fig. 10

Table 2. τ predictions from 10 000 Gaussian CMB + noise simulations generated with three different, fixed fiducial τ values.

	Test on Gaussian simulations								
	143 GHz Gaussian training			143 + 100 GHz Gaussian training			143 × 100 GHz HL likelihood		
fiducial τ	$\overline{\tau_{\mathrm{NN}}}$	$\overline{\sigma_{\mathrm{NN}}(\tau)}$	$\sigma(\tau_{\mathrm{NN}})$	$\overline{\tau_{\mathrm{NN}}}$	$\overline{\sigma_{\mathrm{NN}}(\tau)}$	$\sigma(\tau_{\mathrm{NN}})$	$\overline{\tau_{\mathrm{HL}}}$	$\overline{\sigma_{\mathrm{HL}}(\tau)}$	$\sigma(\tau_{\mathrm{HL}})$
0.05	0.0508	0.0059	0.0066	0.0503	0.0054	0.0057	0.0496	0.0046	0.0047
0.06	0.0608	0.0065	0.0067	0.0600	0.0056	0.0059	0.0596	0.0048	0.0048
0.07	0.0712	0.0067	0.0070	0.0702	0.0057	0.0063	0.0697	0.0048	0.0049

Notes. The results correspond to the Gaussian NN training on one and two channels, and the Bayesian inference with a power spectrum likelihood. We show the posterior mean $\tau_{\mathrm{NN/HL}}$ and standard deviation $\sigma_{\mathrm{NN/HL}}(\tau)$ averaged over all simulations, as well as the scatter of $\tau_{\mathrm{NN/HL}}$ over all simulations.

directly estimated from *Planck* maps, and therefore serve as a good description of the Gaussian noise present in real data. At the same time, they lack for realism, since they do not include non-Gaussian residual systematic effects, contamination due to Galactic foregrounds, both known to be present on the *Planck* SRoll2 maps. We therefore expected these models (which we refer to as "Gaussian models") to induce a bias on τ when applied to the more realistic SRoll2 simulations, or to real *Planck* data.

4.1.1. Single channel

We began by training our CNN on Stokes Q and U maps with Gaussian *Planck*-like noise and CMB at 143 GHz only, thus feeding $N_{\mathrm{map}} = 2$ maps to the network. In the left-hand side of Table 2, we show the results of testing $N_{\mathrm{sims}} = 10\,000$ Gaussian simulations of CMB and noise generated with fiducial $\tau = 0.05$, 0.06, and 0.07, respectively. The average learned mean posterior values $\overline{\tau_{\mathrm{NN}}}$ are close to unbiased and deviate at the 0.2σ level. The average learned posterior standard deviations $\overline{\sigma_{\mathrm{NN}}(\tau)}$ are within 5% agreement with the sample scatter across simulations, $\sigma(\tau_{\mathrm{NN}})$.

To assess the performance of the Gaussian model also on non-Gaussian *Planck*-like maps, we tested this model on 10 000 SRoll2 simulations generated with fiducial $\tau = 0.06$ (see Sect. 2.3). As illustrated in the upper right panel of Fig. 5, this leads to a bias of more than 1σ on τ_{NN}. These tests on a single frequency channel leave us with two conclusions: on the one hand, CNNs are able to correctly retrieve τ and its statistical uncertainty from a single *Planck*-like simulation of the 143 GHz channel containing correlated Gaussian noise. On the other hand, systematic effects present in the *Planck* SRoll2 simulations bias the single-channel CNN inference, as expected. To improve our results, we added another frequency channel to the inference pipeline, seeking to mitigate this bias. We expected that combining two channels should lead to a lower error bar and a lower bias on the SRoll2 simulations, in a similar way as cross-spectra achieve lower noise bias than auto-spectra.
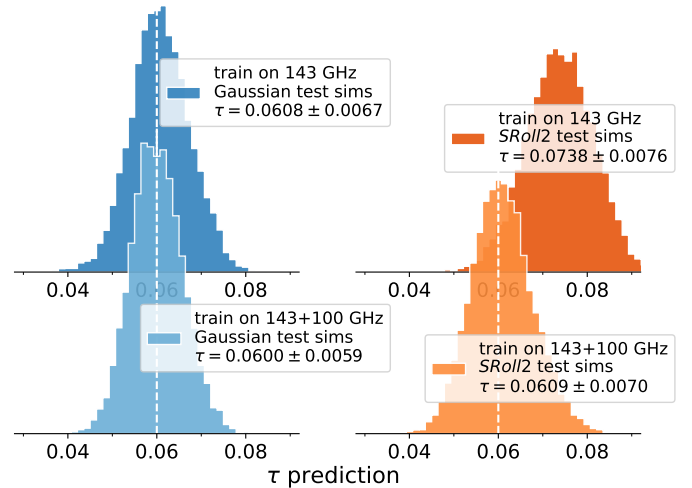

Fig. 5. Predictions of τ_{NN} from 10 000 simulations with input $\tau = 0.06$, containing either CMB with Gaussian noise (*left panels*) or CMB with SRoll2 noise + systematics (*right panels*). The two rows denote different CNN models trained on CMB with Gaussian noise on a single frequency channel (*top*), on two frequency channels (*bottom*).

4.1.2. Two channels

As a second test, we added the HFI 100 GHz channel in the training and testing procedures, simulated as CMB plus the corresponding Gaussian correlated noise, so that $N_{\mathrm{map}} = 4$ maps were fed into the neural network. The results from testing on Gaussian noise are shown in Table 2. We note two positive effects: firstly, the small bias observed for Gaussian noise on a single channel reduces to below 1% of a standard deviation. Secondly, the learned $\sigma_{\mathrm{NN}}(\tau)$ decreases by more than 10% when training on two frequency channels. Meanwhile, the prediction of the posterior standard deviation stays within 5% of the sample standard deviation of the inferred τ_{NN}. The same results are presented in Fig. 5 for fiducial $\tau = 0.06$, showing significant improvement of

Table 3. Same as Table 2 but testing on CMB and SRoll2 simulations instead of CMB and Gaussian noise simulations.

	Test on SRoll2 simulations								
	143 GHz Gaussian training			143 + 100 GHz Gaussian training			143 × 100 GHz HL likelihood		
fiducial τ	$\overline{\tau_{NN}}$	$\overline{\sigma_{NN}(\tau)}$	$\sigma(\tau_{NN})$	$\overline{\tau_{NN}}$	$\overline{\sigma_{NN}(\tau)}$	$\sigma(\tau_{NN})$	$\overline{\tau_{HL}}$	$\overline{\sigma_{HL}(\tau)}$	$\sigma(\tau_{HL})$
0.05	0.0669	0.0065	0.0074	0.0536	0.0055	0.0067	0.0478	0.0050	0.0079
0.06	0.0738	0.0067	0.0076	0.0609	0.0056	0.0070	0.0585	0.0050	0.0073
0.07	0.0813	0.0069	0.0074	0.0690	0.0057	0.0071	0.0688	0.0049	0.0069

the two-channel CNN inference in the lower panels with respect to the one-channel results (upper panels). We proceeded to test this two-channel Gaussian model on the SRoll2 simulations. As shown in the right panel of Fig. 5, for fiducial $\tau = 0.06$, the addition of a second channel allows for a significant reduction of the systematic-related bias in τ_{NN} and to a better statistical constraint. This led us to conclude that CNNs are able to recognize common features across channels, combining the information to reduce the statistical uncertainty and the bias due to uncorrelated systematic effects.

The corresponding quantitative results, for all the three τ values used during testing, are listed in Table 3: adding a second channel in the Gaussian training model leads to improved results on the SRoll2 test simulations for all considered values of τ. However, a residual bias is still present, especially for $\tau = 0.05$, when the CMB signal is smallest.

Moreover, we noticed that, when applied to the SRoll2 test maps, the models trained on Gaussian simulations returned values of $\sigma_{NN}(\tau)$ that disagreed with the actual spread of estimates $\sigma(\tau_{NN})$, with the latter being up to ~25% larger. This implies that the learned error was not accurate in this case, hence could not be used to describe the uncertainties of our inferred τ values on SRoll2 maps. We address this issue in Sect. 4.4.

4.2. Comparison with Bayesian inference from cross-QML power spectrum estimates

In this section we compare NN inference results with results coming from a standard Bayesian approach applied to *E*-mode power spectra. In particular, we considered quadratic Maximum Likelihood (QML) estimates (see, e.g., Tegmark & de Oliveira-Costa 2001) of the 100×143 GHz *EE* cross-spectrum and drew posterior samples using the well-known power spectrum likelihood approximation introduced by Hamimeche & Lewis (2008; in the following HL likelihood). The HL likelihood provides a good approximation to the non-Gaussian distribution of the exact power spectrum likelihood, which markedly differs from Gaussianity at the low multipoles $2 \leq \ell \lesssim 30$ that are most relevant for constraining τ. Evaluating the HL likelihood requires a power spectrum covariance matrix, which we obtained directly from simulations of Gaussian noise and CMB realized with the same τ values used for generating the test simulations (Sect. 2). For the HL likelihood we assumed a theoretical model of the CMB *E*-modes, computed with CAMB, considering the multipole range $2 \leq \ell \leq 30$, and sampling only for the τ parameter, keeping $10^9 A_s e^{-2\tau} = 1.884$ fixed. Our final results are the best-fit value τ_{HL}, the standard deviation $\sigma_{HL}(\tau)$ of the posterior, and the scatter $\sigma(\tau_{HL})$ computed from the set of test simulations.

We ran the HL likelihood on $3 \times 10\,000$ Gaussian sky simulations with input $\tau = 0.05, 0.06,$ and 0.07. As shown in the

last three columns of Table 2, we find unbiased best-fit results with average posterior standard deviation $\overline{\sigma_{HL}(\tau)}$ and best-fit parameter scatter $\sigma(\tau_{HL})$ of ~0.0048. We note that the uncertainties derived from sampling the HL likelihood are ~20% smaller than the ones from NN estimates. Part of the scatter of τ_{NN} comes from the intrinsic stochastic nature of the training process. We could reduce this scatter by taking the average over multiple NN models (as discussed in Sect. 4.4). Nevertheless, these results reveal that although we were able to retrieve unbiased τ values with NNs from Gaussian simulations, our estimator does not achieve minimum variance. Further development of the method, including an optimization of the convolution algorithm on the sphere, the NN architecture, and the training procedure, are required and will be explored in future work in the light of improving the estimator's variance.

In addition to Gaussian simulations, we applied the cross-spectrum inference pipeline on $3 \times 10\,000$ SRoll2 simulations and show the corresponding results in the last three columns of Table 3. We stress that the HL likelihood contains the same covariance matrix as before, calculated from Gaussian simulations. This is done in analogy with the case of Gaussian NN training applied to SRoll2 simulations, therefore neglecting the presence of systematic effects. We retrieve biased estimates on τ, confirming our expectation that the power spectrum model implemented in the likelihood is an inaccurate representation of the SRoll2 simulations, which include spurious non-Gaussian signals. Interestingly, this affects the NN and HL estimates in different ways, leading to biases in opposite directions for $\tau = 0.05$ and 0.06. To study the relative behavior of the two estimators, it is instructive to look at a one-by-one comparison of the NN and HL results on the same 10 000 test simulations, as presented in Fig. 6 for $\tau = 0.06$. The scatter plot of the estimated τ_{NN} and τ_{HL} on Gaussian simulations and on SRoll2 simulations are shown in bright red and dark green, respectively. In the Gaussian case the correlation of the estimated τ values is at a level of ~76%, while for SRoll2 it is at 63%. We conclude that map-level systematic effects, which are partially unaccounted for in the estimates, decrease the correlation and increase the differences between τ_{HL} and τ_{NN} when changing from Gaussian to SRoll2 test simulations. This indicates that spurious non-Gaussian signals impact the two estimators in different ways.

4.3. Training including systematic effects

As previously seen, the two-channel Gaussian training allowed to improve our τ estimates on SRoll2 simulations. However, the continued occurrence of bias, even though small, motivated us to evolve the training setup by including systematic effects in the training set. Our goal was to achieve fully unbiased results before applying our NN models to real *Planck* maps. In this section we explore two possible ways of including systematics: training
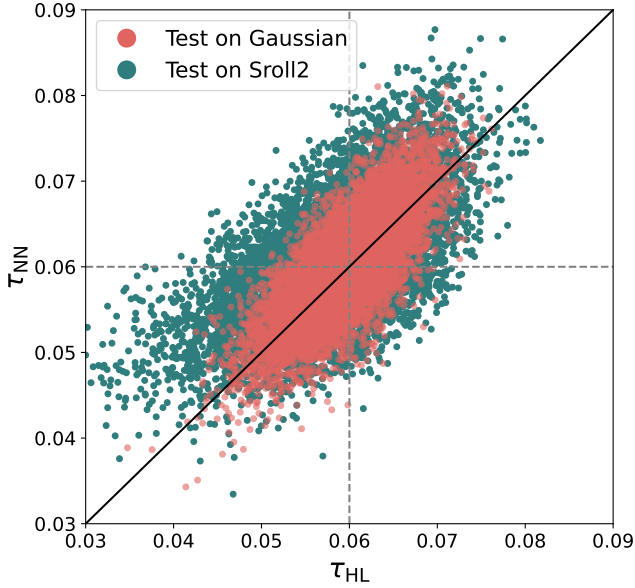
Fig. 6. Per-simulation comparison between the HL likelihood estimate τ and the NN estimate τ_{NN} for a test set of 10 000 simulations realized with $\tau = 0.06$. Gaussian simulations are shown in bright red, `SRoll2` simulations in dark green. The correlation coefficients between both estimators are 76% (Gaussian) and 63% (`SRoll2`).

on `SRoll2` simulations from the very beginning and performing a `SRoll2` retraining update on previously trained Gaussian networks.

4.3.1. Training on SRoll2 simulations

The `SRoll2` simulations ([Delouis et al. 2019](#)) are designed to accurately describe *Planck*'s Gaussian noise component and non-Gaussian polarization systematics. Motivated by this, we trained a CNN from the start on the 200 000 `SRoll2` training simulations described in Sect. [2.3](#). As usual, we used 190 000 simulations to perform weight optimization, and 10 000 for validation. We trained on *Planck*'s 143 GHz and 100 GHz channels simultaneously and used the same hyperparameter values as for the Gaussian training, described in Sect. [3.2](#). We stress that even though artificially augmented by forming new channel pair combinations, the `SRoll2` training set was essentially built from 400 sampled skies only. We tested on $3 \times 10\,000$ `SRoll2` simulations with fixed $\tau = 0.05$, 0.06, and 0.07, generated from the remaining 100 independent realizations that the CNN did not "see" during training.

Table [4](#) shows the results obtained with this approach. For the three input τ values we find a positive bias of $\sim 0.4\sigma$. For $\tau = 0.06$, the average learned error $\overline{\sigma_{NN}(\tau)} = 0.0062$ is slightly larger than for the two-channel Gaussian training but smaller than the scatter $\sigma(\tau_{NN}) = 0.0070$. We see similar results both for the Gaussian CNN and the HL likelihood inference (see Table [3](#)). As in the case of Gaussian NN training, the learned error does not agree with the `SRoll2` simulation scatter, therefore it cannot be used to infer the statistical uncertainty on τ_{NN}.

We ascribe the main reason for the bias on τ to overfitting. Figure [7](#) illustrates this problem. We compared the τ predictions on a set of 10 000 test simulations with the ones coming from 10 000 training simulations. The results show a bias and standard deviation of $\Delta\tau = 0.0023 \pm 0.0069$ for the test set, while the training set is unbiased, with $\Delta\tau = 0.0001 \pm 0.0068$. This is
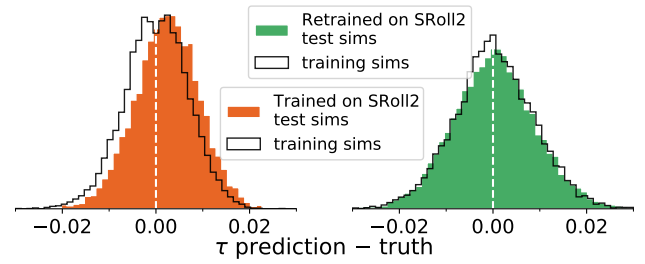
Fig. 7. Neural network accuracy in predicting the true τ input from 10 000 simulations. Step-filled histograms show the results on unseen test simulations, black outlines show the results on a subset of the actual training simulations. We compare a network exclusively trained on `SRoll2` simulations (*left panel*) with a Gaussian network retrained on `SRoll2` simulations (*right panel*).

clear evidence for overfitting: while the model performs well on the 400 `SRoll2` simulations that the training set is built from, these are not enough to generalize to the remaining 100 `SRoll2` simulations used to build the test set, leading to the observed bias on τ in the latter case.

4.3.2. Retraining update with SRoll2 simulations

We recognized the bias described above as a critical problem that needed to be addressed. The obvious option, training on a considerably larger simulation set, was unavailable to us due to the limited number of `SRoll2` realizations. Therefore, we applied a transfer learning technique to inform our previously trained Gaussian networks on the `SRoll2` systematics. As shown in the previous sections, our Gaussian NN model is not affected by overfitting issues and, if trained on two channels, performs reasonably well even on `SRoll2` simulations. This motivated us to leverage the existing results on Gaussian networks to solve the overfitting issue with as little changes as possible. To this end, we chose the approach of retraining the two-channel Gaussian model on the full set of `SRoll2` training simulations, while targeting two specific goals:

(i) The retrained CNN should learn to extract information on the systematic effects present in the `SRoll2` simulations and update its CNN weights just enough to achieve fully unbiased results on the `SRoll2` training set.

(ii) At the same time, we wanted to ensure that the information already learned was not destroyed during the new training phase (an issue sometimes referred to as "catastrophic forgetting", see e.g., [Kirkpatrick et al. 2017](#); [Ramasesh et al. 2021](#)), avoiding going back to the overfitting situation described in the previous section.

We achieved this by performing what we call "minimal retraining": we chose the hyperparameters of the NN such that we obtained unbiased results on the `SRoll2` test simulations while making minimal changes to the original network. We found an optimal setup with five retraining epochs, a learning rate of $LR = 10^{-7}$, and no additional changes to the original network architecture.

The right panel of Fig. [7](#), in analogy to the left panel, compares the distribution of $\Delta\tau$ from the `SRoll2`-retrained model on training simulations (black contours), or test simulations (green filled histogram). We find both histograms to be in good agreement, indicating that unlike the `SRoll2`-trained model, the retrained model does not suffer from overfitting, thus achieving our goal (ii) defined above. Table [4](#) on the right-hand side lists the results of the `SRoll2`-retrained model on `SRoll2` test

Table 4. τ predictions from 10 000 CMB + SRoll2 test simulations generated with three different fiducial τ values.

	Test on SRoll2 simulations					
	143 + 100 GHz SRoll2 training			143 + 100 GHz SRoll2 retraining		
fiducial τ	$\overline{\tau_{\mathrm{NN}}}$	$\overline{\sigma_{\mathrm{NN}}(\tau)}$	$\sigma(\tau_{\mathrm{NN}})$	$\overline{\tau_{\mathrm{NN}}}$	$\overline{\sigma_{\mathrm{NN}}(\tau)}$	$\sigma(\tau_{\mathrm{NN}})$
0.05	0.0526	0.0059	0.0066	0.0508	0.0077	0.0091
0.06	0.0622	0.0062	0.0070	0.0606	0.0079	0.0088
0.07	0.0722	0.0064	0.0070	0.0707	0.0081	0.0087

Notes. These results correspond to two frequency channels, either training on SRoll2 from the start or retraining on SRoll2 maps. Displayed are the average posterior mean, average predicted standard deviation $\overline{\sigma_{\mathrm{NN}}(\tau)}$, and the scatter $\sigma(\tau_{\mathrm{NN}})$ calculated across the test simulations.
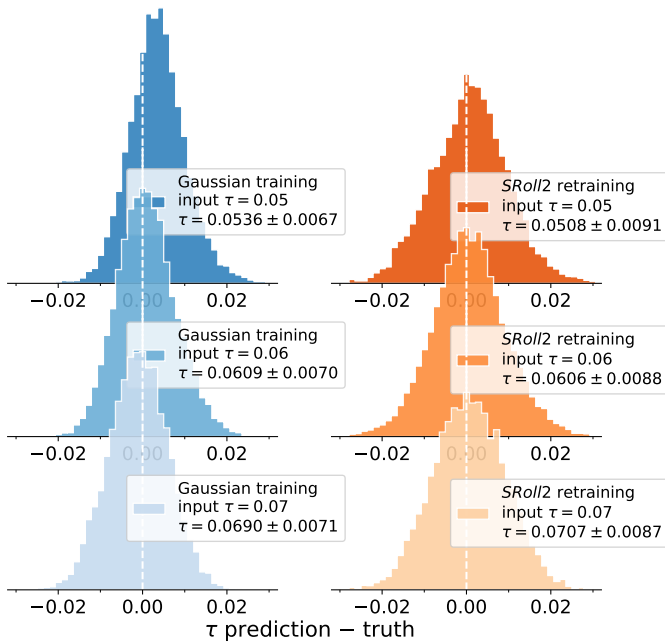

Fig. 8. Predictions of τ_{NN} on 10 000 SRoll2 simulations with input $\tau = 0.05$, 0.06, and 0.07 (first, second, third row, respectively). The two columns display two different NN models trained on two channels of Gaussian simulations (*left panels*) and retrained on two channels of SRoll2 simulations (*right panels*). All results correspond to $f_{\mathrm{sky}} = 0.5$.

simulations. We find $\overline{\tau_{\mathrm{NN}}} = 0.0508$, 0.0606, and 0.0707 for the respective input values of $\tau = 0.05$, 0.06, and 0.07. This amounts to a bias below $\Delta\tau = 8 \times 10^{-4}$, or $\lesssim 0.1\sigma$. In Fig. 8, we show a comparison of the results on SRoll2 test sets obtained by Gaussian and SRoll2-retrained CNNs. The reduction of the bias is evident, in particular for $\tau = 0.05$. Therefore, we chose the retrained approach as our baseline model to estimate τ on real *Planck* data. At the same time, this approach brings an increase in $\sigma(\tau_{\mathrm{NN}})$, an effect not seen with the SRoll2 training procedure described in Sect. 4.3.1[6]. This could be the consequence of the typical variance-bias trade-off observed between statistical estimators: with minimal retraining we are able to achieve unbiased estimates (goal (i) above) at the cost of a larger $\sigma(\tau_{\mathrm{NN}})$. In addition to that, we are still unable to retrieve values of the learned $\sigma_{\mathrm{NN}}(\tau)$ that agree with $\sigma(\tau_{\mathrm{NN}})$ for SRoll2 simulations (and therefore also for *Planck* data). We conclude that, except for case in which we test the Gaussian model on Gaussian sim-

---

[6] Compare the fourth column in Table 4 with the seventh column in Table 3.

ulations, we cannot use the learned error as an estimate of the uncertainty of the inferred τ_{NN}.

4.4. NN errors

The loss function in Eq. (6) provides an estimate for the posterior standard deviation $\sigma_{\mathrm{NN}}(\tau)$. However, as seen in the previous sections, the learned $\sigma_{\mathrm{NN}}(\tau)$ tends to underestimate the actual spread of the inferred values of τ_{NN} on test set maps, especially in the case of SRoll2 maps. We therefore proceeded to empirically estimate our errors from simulations.

In doing so, we needed to make an additional consideration: training a NN is an intrinsically stochastic procedure that relies upon the use of a stochastic optimizer, randomly initialized NN weights and random realizations of the maps in the training set. This results in the fact that each NN prediction can be described as the sum of two random variables: $\tau_{\mathrm{NN}} = \tau + \Delta_{\mathrm{NN}}$, and therefore

$$\sigma^2(\tau_{\mathrm{NN}}) = \sigma^2(\tau) + \sigma^2(\Delta_{\mathrm{NN}}) + 2\,\mathrm{Cov}(\tau, \Delta_{\mathrm{NN}}), \qquad (7)$$

where the first source of uncertainty, $\sigma(\tau)$, is due to noise and cosmic variance of test simulations or observed data, while the second, $\sigma(\Delta_{\mathrm{NN}})$, represents the stochasticity of the NN estimator. These two terms are sometime referred to as aleatory and epistemic error, respectively (Hüllermeier & Waegeman 2021).

We can measure the uncertainty related to the NN stochasticity by training an ensemble of models, all based on the same architecture and hyperparameters, but with different initial weights and training set realizations. Our estimate of $\sigma(\Delta_{\mathrm{NN}})$ is given by the standard deviation of the models' τ predictions when tested on a single test map. In practice, we defined the "model ID" of a trained NN as the fixed random seed controlling the initialization of network weights. We generated a new training set of simulations whose specific realizations (of CMB, noise, and potentially systematics) was fully determined by the model ID. Following this recipe, we created 100 independent Gaussian training sets and used them to train 100 Gaussian networks. Repeating this procedure with 100 SRoll2 training sets, we retrained the set of 100 Gaussian networks to obtain 100 SRoll2-retrained networks. Using a single test map with input $\tau = 0.06$, we find $\sigma(\Delta_{\mathrm{NN}}) \simeq 0.0024$ for Gaussian NN models tested on Gaussian maps, and $\sigma(\Delta_{\mathrm{NN}}) \simeq 0.0034$ for minimally retrained NN models tested on SRoll2. In both cases this represents about 40% of the corresponding value of $\sigma(\tau_{\mathrm{NN}})$ reported in Tables 2 and 4, respectively.

We can reduce the impact of the NN stochasticity by taking, for each test map, the ensemble average of the τ estimates over the 100 trained NNs. By doing so, for the case with $f_{\mathrm{sky}} = 0.5$ and input $\tau = 0.06$, we find $\sigma(\tau_{\mathrm{NN}}) \simeq 0.0054$ for Gaussian
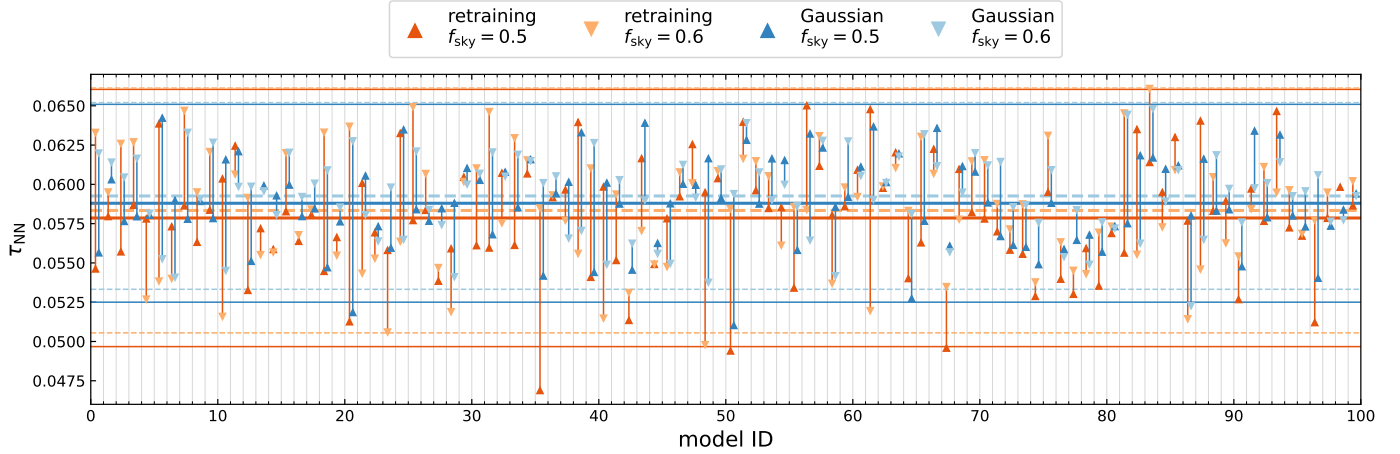
Fig. 9. NN predictions of τ from *Planck* 100+143 GHz data, resulting from training 100 equivalent models with different random initial weights and random seeds for training data, considering Gaussian two-channel training (blue tones) versus SRoll2 retraining (orange tones), and $f_{sky} = 0.5$ (downward triangles) versus $f_{sky} = 0.6$ (upward triangles). Colored triangle markers show the best-fit values for the single models and horizontal lines in the corresponding colors show the ensemble average of τ (middle) ± the 68% confidence interval (upper and lower lines).
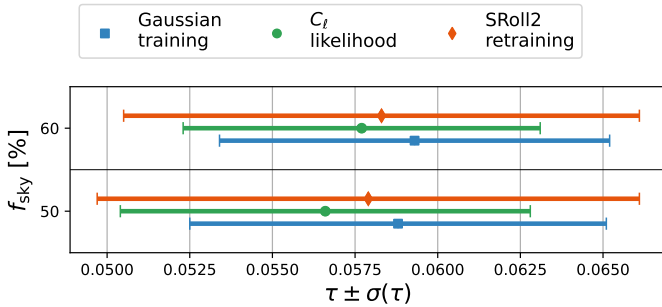
Fig. 10. Results on τ obtained from *Planck* SRoll2 data. The values in this plot are shown in Table 5.

models applied to Gaussian maps and $\sigma(\tau_{NN}) \simeq 0.0083$ for retrained models applied to SRoll2 simulations.

We also evaluated the correlation coefficient between the predictions of pairs of models (j, k), tested on the same 10 000 simulations, for both Gaussian and SRoll2 training and testing, respectively. In both cases, we find $\rho_{jk} \simeq 0.84$, in agreement with what is expected if Eq. (7) holds and the models' epistemic errors are uncorrelated, $\text{Cov}(\Delta_{NN}^j, \Delta_{NN}^k) = \delta_{jk}^{\mathcal{K}} \sigma^2(\Delta_{NN})$. In the following section we describe how we applied our CNN models to *Planck* maps to infer the value of τ from data, estimating its uncertainty from simulations and using the ensemble average over 100 trained models to reduce the impact of the NN stochasticity.

5. Results on *Planck* data

As shown in Sects. 4.3.2 and 4.4, by retraining on the SRoll2 simulations, we are able to obtain a CNN-based model that yields unbiased results on unseen SRoll2 test simulations generated with fixed $\tau \in \{0.05, 0.06, 0.07\}$. Having thus confirmed the robustness of our method, we moved to real *Planck* data and proceeded to predict τ from the 100 and 143 GHz SRoll2 HFI maps.

We obtained our baseline τ estimate by taking the average of the inferred values from the 100 minimally retrained NNs applied to *Planck* data for a sky mask with $f_{sky} = 0.5$, resulting in a mean estimate of $\tau_{NN} \simeq 0.0058$. Figure 9 shows the obtained

τ values for each of these NN models. Following the conclusions of the previous sections, since the learned $\sigma_{NN}(\tau)$ is inadequate as an error prediction, we estimated the uncertainty from simulations. In practice, we generated a set of 10 000 SRoll2 simulations realized with $\tau = 0.058$ and average the τ_{NN} estimates over 100 networks. Afterwards, we computed the standard deviation over 10 000 simulations. Our final inference on *Planck* maps in this baseline case results in:

$$\tau_{NN} = 0.0579 \pm 0.0082 \quad (Planck\ 100 + 143\ \text{GHz}). \quad (8)$$

This value is in very good agreement with the τ estimates obtained with an empirical likelihood based on cross-QML power spectra, presented in Pagano et al. (2020; hereafter P2020) applied to the same *Planck* maps and constructed from the same SRoll2 simulations that we use in this work. In particular, P2020 obtained $\tau = 0.0566^{+0.0053}_{-0.0062}$ on the $f_{sky} = 0.5$ sky mask. We note that the uncertainty from our NN method is ~30% larger. As previously described, this is due to the fact that our NN estimator does not reach minimum variance and that we relied on the retraining strategy leading to larger errors. However, the fact that we obtain a τ value in agreement with the literature while using an inherently different inference approach that is, for the first time, fully based on NNs, represents a remarkable result of this work.

We also applied the Gaussian NN model to *Planck* data, deriving the best-fit parameter value and error bars analogously. We note that, although the Gaussian model leads to results that are mildly biased by up to ~0.5σ when applied to SRoll2 maps with low CMB input signal ($\tau = 0.05$), the bias is below 0.15σ when $\tau = 0.06$, as displayed in the fifth column of Table 3. In this case, using the same $f_{sky} = 0.5$ mask, we obtained $\tau_{NN} = 0.0588 \pm 0.0063$. The statistical uncertainty is lower for this second method, as we omitted retraining on systematics, and similar to the one obtained from the empirical likelihood presented in P2020.

Lastly, as a robustness test, we applied these same methods to a second sky mask, with a larger sky coverage of $f_{sky} = 0.6$. The parameter estimated remained stable for both the retrained and the Gaussian model, with lower uncertainties. The NN predictions of the single models on $f_{sky} = 0.6$ are displayed in Fig. 9. A summary of our results on *Planck* maps is shown in Fig. 10 and Table 5.

Table 5. Results from *Planck* data on two different sky masks, using Gaussian NNs, SRoll2-retrained NN models, and the empirical C_ℓ-based likelihood presented in Pagano et al. (2020).

| | Predictions on *Planck* SRoll2 data | | | | | |
| | 143 + 100 GHz Gaussian training | | 143 + 100 GHz SRoll2 retraining | | 143 × 100 GHz C_ℓ likelihood | |
$f_{\rm sky}$	$\tau_{\rm NN}$	$\sigma(\tau_{\rm NN})$	$\tau_{\rm NN}$	$\sigma(\tau_{\rm NN})$	τ	$\sigma(\tau)$
50%	0.0588	0.0063	0.0579	0.0082	0.0566	0.0062
60%	0.0593	0.0059	0.0583	0.0078	0.0577	0.0054

Notes. The NN results are averaged over 100 models, and $\sigma(\tau_{\rm NN})$ is computed from 10 000 simulations with input $\tau = 0.058$.

6. Conclusions

In this paper, we present the first cosmological parameter inference on *Planck*'s CMB polarization maps that is performed entirely by neural networks. We estimated the optical depth to reionization, τ, from the SRoll2 low resolution polarization maps of *Planck*-HFI at 100 and 143 GHz. These maps are known to contain a significant level of residual systematic effects at large angular scales that, if ignored, would bias cosmological results. These spurious signals are non-Gaussian and hard to model in an analytical way. For this reason, in the literature (P2020), the estimation of τ from these maps is obtained by sampling an empirical *EE* cross-spectrum likelihood (Planck Collaboration V 2020; Gerbino et al. 2020), built from a set of realistic SRoll2 simulations (Delouis et al. 2019).

In this work, we approached this problem through NN-based inference applied directly on the map domain. One of the benefits of this method is that it does not require an analytical model of the data but, instead, relies solely on simulations to train a regression model. In particular, we used the NNhealpix algorithm to build our NN models, allowing the application of convolutional layers on the sphere. We considered several setups to train and validate CNNs on multiple sets of simulations, before applying them to *Planck* data. We adopted the moments loss function (Jeffrey & Wandelt 2020) to learn the mean and standard deviation of the marginal posterior on τ inferred from Stokes Q and U maps pixelized on a grid at $N_{\rm side} = 16$ (~4°). To find the best training method, we started from simulations of a single frequency channel of CMB with coadded Gaussian correlated noise and, step by step, moved to more complex setups that involved two frequency channels containing CMB, noise, and systematic effects. We compared the results obtained with NNs with the ones from a standard Bayesian method that applies the HL likelihood to *EE* cross-spectra. Our main results and conclusions from the analysis applied to simulations are the following:

1. When trained and applied to Gaussian simulations, the NN models are able to retrieve unbiased values of τ directly from maps. Additionally, by using the moments loss function reported in Eq. (6), the models can also learn and return an error estimate that is consistent with the spread of the best-fit values on the test set.

2. When trained using maps from two frequency channels that share the same cosmological signal, the NNs are able to effectively combine the information from both maps. This leads to improved accuracy in the τ estimates and smaller uncertainties. This ability to combine information from different channels is a key advantage of the NN approach as, in the future, it would allow for a straightforward combination of all available data sets without the need for a joint model, thus reducing the impact of noise and systematics.

3. A comparison of the NN estimates with the ones obtained from the HL cross-spectrum method applied to Gaussian simulations shows that the NN approach leads to higher uncertainties by about 20%. This implies that the NN estimator, although unbiased, does not reach the minimum variance. In order to further improve the performance of the estimator, future work should focus on optimizing the spherical convolution algorithm, the model architecture, and the training procedure. This will help to minimize the uncertainties and reach the best possible performance.

4. The application of the Gaussian two-channel model to the SRoll2 simulations, which include systematic effects, leads to inaccurate estimates on τ, as does the use of the HL likelihood. Although expected, this observed bias is much smaller (nearly unbiased for $\tau \sim 0.06$) than that seen for the single-channel model, demonstrating that the neural network is able to identify common features in the maps, efficiently ignoring the uncorrelated signal between different channels.

5. To recover fully unbiased results on SRoll2 maps, as a prerequisite to apply our NN model to *Planck* data, we needed to train NNs on maps that incorporate instrumental systematic effects. Due to the limited number of available SRoll2 simulations, we adopted a minimal retraining approach, building on the good results already obtained with the Gaussian models. This approach helps to minimize the issue of overfitting, but it also leads to slightly larger errors in the recovered τ values.

6. In more complex scenarios, when we applied the NN models to the SRoll2 maps, we found that the error estimate learned by the NN, $\sigma_{\rm NN}(\tau)$, underestimated the spread evaluated on the empirical distribution of the test maps, $\sigma(\tau_{\rm NN})$. This suggests that the NN model did not capture the full range of uncertainty in the data. To overcome this issue, we proceeded by evaluating the final error on τ through simulations, by taking the ensemble average of 100 NN models. This helped to reduce the impact of the epistemic uncertainty caused by the intrinsic stochasticity of the NN estimator.

After evaluating and validating the performance of the NNs on simulations, we applied our trained models to *Planck* SRoll2 maps at 100 and 143 GHz. For the minimally retrained model, which is the one that leads to fully unbiased results on the SRoll2 simulations, we obtain $\tau_{\rm NN} = 0.0579 \pm 0.0082$ on our fiducial $f_{\rm sky} = 0.5$ mask. This value is in very good agreement with the one obtained from the empirical likelihood based on cross power spectra reported in P2020, which relies on the same set of simulations. We consider this a remarkable result of our work, given the fact that the two estimators are intrinsically different. However, we note that our final uncertainty on the $\tau_{\rm NN}$ estimate, which we evaluated from simulations and the ensemble average of 100 NN models, is about 30% larger than the one

obtained in P2020. This is because our NN estimator does not reach minimum variance and, moreover, we could rely only on a limited number of SRoll2 simulations to inform the NN about systematic effects. The minimal retraining approach allows us to achieve unbiased results, but at the cost of an increased variance.

An effective robustness test against systematics-induced "unknown unknowns" is to predict τ from SRoll2 simulations using the Gaussian network (as described in Sect. 4.1), which can be considered agnostic to the strong non-Gaussian map features characteristic for SRoll2 maps. Given its good performance on SRoll2 simulations for $\tau \sim 0.06$, we applied the Gaussian model to the *Planck* data. In this second case we obtain $\tau_{NN} = 0.0588 \pm 0.0063$, in agreement with the estimate reported in the literature, and with a similar level of uncertainty.

As an additional robustness test of the NN approach, we considered a second mask that retains a larger sky fraction of $f_{sky} = 0.6$ and find consistent results. The summary of our results is reported in Table 5 and Fig. 10, showing full stability of the retrieved τ_{NN} estimations.

Concluding, in this work we present a first thorough application of NN-based inference to real CMB maps. It is important to stress that obtaining reliable results on real data required a significant effort to validate and test our models on different setups and to develop training strategies that can effectively cope with systematic effects. This highlights the fact that NN models developed to perform well on simplified simulations cannot always be straightforwardly applied to real data and need careful consideration of the training and validation procedures. Nonetheless, the consistent and robust results we obtain demonstrate that NNs represent a promising tool that could complement standard statistical data analysis techniques for CMB observations, especially in cases where the Gaussian CMB signal is contaminated by spurious effects that cannot be analytically described in a likelihood model. This is particularly relevant for the ongoing search for primordial gravitational waves, constrained by large-scale *B*-modes which are targeted by a number of near-future experiments such as the *Simons* Observatory (Ade et al. 2019), LiteBIRD (LiteBIRD Collaboration 2022) and CMB-S4 (Abazajian et al. 2019). However, additional optimization and validation of this approach must be developed before tackling this challenge.

References

Abazajian, K., Addison, G., Adshead, P., et al. 2019, arXiv e-prints [arXiv:1907.04473]
Ade, P., Aguirre, J., Ahmed, Z., et al. 2019, JCAP, 02, 056
Becker, R. H., Fan, X., White, R. L., et al. 2001, AJ, 122, 2850
Benabed, K., Cardoso, J. F., Prunet, S., & Hivon, E. 2009, MNRAS, 400, 219
Bosman, S. E. I., Davies, F. B., Becker, G. D., et al. 2022, MNRAS, 514, 55
Caldeira, J. A., Wu, W. L. K., Nord, B., et al. 2019, Astron. Comput., 28, 100307
Casas, J. M., Bonavera, L., González-Nuevo, J., et al. 2022, A&A, 666, A89
Chollet, F., et al. 2015, Astrophysics Source Code Library, [record ascl:1806.022]
de Belsunce, R., Gratton, S., Coulton, W., & Efstathiou, G. 2021, MNRAS, 507, 1072
Delouis, J. M., Pagano, L., Mottet, S., Puget, J. L., & Vibert, L. 2019, A&A, 629, A38
Fan, X., White, R. L., Davis, M., et al. 2000, AJ, 120, 1167
Fan, X., Narayanan, V. K., Lupton, R. H., et al. 2001, AJ, 122, 2833
Fluri, J., Kacprzak, T., Lucchi, A., et al. 2019, Phys. Rev. D, 100, 063514
Gerbino, M., Lattanzi, M., Migliaccio, M., et al. 2020, Front. Phys., 8, 15
Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, ApJ, 622, 759
Gunn, J. E., & Peterson, B. A. 1965, ApJ, 142, 1633
Hamimeche, S., & Lewis, A. 2008, Phys. Rev. D, 77, 103013
Harris, C. R., Jarrod Millman, K., van der Walt, S. J., et al. 2020, Nature, 585, 357
Hinshaw, G., Larson, D., Komatsu, E., et al. 2013, ApJS, 208, 19
Hüllermeier, E., & Waegeman, W. 2021, Mach. Learn., 110, 457
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Jeffrey, N., & Wandelt, B. D. 2020, in 34th Conference on Neural Information Processing Systems
Jeffrey, N., Boulanger, F., Wandelt, B. D., et al. 2022, MNRAS, 510, L1
Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. 2017, Proc. Nat. Acad. Sci., 114, 3521
Krachmalnicoff, N., & Puglisi, G. 2021, ApJ, 911, 42
Krachmalnicoff, N., & Tomasi, M. 2019, A&A, 628, A129
Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473
LiteBIRD Collaboration (Allys, E., et al.) 2022, Prog. Theor. Exp. Phys., 2023, 042F01
Natale, U., Pagano, L., Lattanzi, M., et al. 2020, A&A, 644, A32
Pagano, L., Delouis, J. M., Mottet, S., Puget, J. L., & Vibert, L. 2020, A&A, 635, A99
Planck Collaboration VI. 2014, A&A, 571, A6
Planck Collaboration XI. 2016, A&A, 594, A11
Planck Collaboration XII. 2016, A&A, 594, A12
Planck Collaboration XIII. 2016, A&A, 594, A13
Planck Collaboration V. 2020, A&A, 641, A5
Planck Collaboration VI. 2020, A&A, 641, A6
Planck Collaboration IX. 2020, A&A, 641, A9
Planck Collaboration Int. XLVI. 2016, A&A, 596, A107
Planck Collaboration Int. LVII 2020, A&A, 643, A42
Qin, Y., Mesinger, A., Bosman, S. E. I., & Viel, M. 2021, MNRAS, 506, 2390
Ramasesh, V. V., Lewkowycz, A., & Dyer, E. 2021, in International Conference on Learning Representations
Scheuer, P. A. G. 1965, Nature, 207, 963
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, J. Mach. Learn. Res., 15, 1929
Tanimura, H., Aghanim, N., Bonjean, V., & Zaroubi, S. 2022, A&A, 662, A48
Tegmark, M., & de Oliveira-Costa, A. 2001, Phys. Rev. D, 64, 063001
Villaescusa-Navarro, F., Genel, S., Anglés-Alcázar, D., et al. 2022, ApJS, 259, 61
Villasenor, B., Robertson, B., Madau, P., & Schneider, E. 2022, ApJ, 933, 59
Wang, G.-J., Shi, H.-L., Yan, Y.-P., et al. 2022, ApJS, 260, 13
Zonca, A., Singer, L., Lenz, D., et al. 2019, J. Open Source Software, 4, 1298