In The Search of Cosmic Strings on Planck's Data with Image Processing, Statistical Tools and Machine Learning Methods

Haleh Hajizadeh

December 28, 2021

Abstract

Cosmic strings network is a topological defect that can cause anisotropies on Cosmic Microwave Background. To investigate the effects of cosmic strings that are conveniently encoded in their tension ($G\mu$), we use different simulations of the CMB temperature maps and simulations of the cosmic string network. Among CMB simulations, end-to-end (E2E) simulations are most similar to Planck's data. We add cosmic string simulation to CMB simulation by a coefficient of tension in the range $6 \times 10^{-9} \leq G\mu \leq 6 \times 10^{-6}$. Then we try to increase the edge effects of string on images by using different curvelet components and filters, which are considered image processing tools. To quantify the possible imprints, we apply standard deviation and Probably Density Function (PDF) for processed data. We use the machine-based method and represent results by confusion matrix and P-value method to predict minimum detectable tension. We used several algorithms and found that the LightGBM method is more accurate and faster than other methods. By comparing different results, we found that the constraints obtained from the matrix method are more accurate than others. We applied the LightGBM-trained model on these simulations on Planck's data and found the upper bound to cosmic strings tension to be $G\mu \leq 8.6 \times 10^{-7}$ with 3σ confidence level.

1 Introduction

According to theoretical cosmological models, topological defects in cosmic strings occur during the phase transition in the early universe [1]. Cosmic strings are significant because they can give us essential information about the early universe. The tension of cosmic string $(G\mu)$ is a dimensionless quantity that represents energy per unit per length of cosmic strings. Due to the lack of angels that cosmic strings create in space-time metric around them, they have imprints like edges on cosmic microwave background (CMB), that this effect called the Kaiser-Stebbins effect [2]. The induced

anisotropy has the form [3, 4]

$$\frac{\delta T}{T} \sim 8\pi G\mu v_{\rm s} \tag{1}$$

Where $G\mu$ is the main parameter characterizing the string network with μ being the string tension and $v_{\rm s}$ the transverse velocity of the string, we can use this effect to detect of cosmic strings network.

There have been a lot of studies to develop a model to find and simulate these imprints, and various statistical tools have been used for their research. We will point to some of them in the following.

[5] set the minimum detectability bound of $G\mu > 5.5 \times 10^{-8}$ with edge-detection algorithms For simulations of the South Pole Telescope, and [6] find a detection limit of $G\mu > 1.4 \times 10^{-7}$ using wavelets and curvelets. [7] also used various combinations of image processing tools and statistical measures, followed by tree-based learning algorithms [8] to forecast the detectability limit of different observational scenarios. [9] develop a framework for cosmic string inference by constructing Bayesian analysis in wavelet space on Planck-like maps that are sensitive to $G\mu \sim 5 \times 10^{-7}$ Nambu-Goto string. For string tension above this value, they recover accurate estimates of its posterior distribution, which can be used to provide precise point estimates of the $G\mu$ and associated error.

In this thesis, we answered these questions: First, what is the minimum value of $G\mu$ that can be measured on different CMB simulations? Second, which of the methods used to detect strings, such as various machine learning algorithms and P-value, sets a more vital constraint on string tension? And third, what is the limitation of our model on the tension of strings in Planck's data?

To answer these questions, we follow steps in the following. First, we will have a brief introduction to machine learning in section 2

2 Machine Learning

Machine learning is an application of artificial intelligence that gives computers to learn and improve from experience without being explicitly programmed. We give some data to a machine to train a model on that data, and then use the trained model to make predictions on a new dataset that The goal of the machine is to create a function between the input dataset and their targets or labels. In the following, we will explain the machine learning method with high performance in our work.

Gradient Boosting Method

Gradient boosting models are supervised algorithms based on decision trees and are commonly used in classification and regression problems. A decision tree has a tree-like structure with nodes,

branches, and leaves. The tree starts with a question corresponding to a node, and depending on the answer; the data is split into branches. The process continues until the algorithm reaches a leaf that predicts the label of the data. Gradient boosting combines many weak learners into a strong, weak learner and is based not on a single model but on gradually improving the model toward models with lower loss function and higher prediction power at each step. The performance of a model is evaluated by the loss or cost function by comparing predictions with fiducial values.

In this work, we use the two gradient boosting libraries of XGBoost and LightGBM, or LGBM for short. XGBoost is an advanced and powerful implementation of gradient boosting, and due to high accuracy, it is the popular boosting method for many learning problems. However, XGBoost can be very time-consuming for problems with massive datasets. In LightGBM, the tree grows leaf-wise compared to XGBoost and other tree-based algorithms where the tree grows depth-wise. The leaf-wise split surprisingly increases the speed and accuracy of the model. However, there is a high risk of overfitting due to increased complexity. This is usually avoided by setting a maximum depth to which splitting can occur. The leaf with a larger loss function is chosen to grow and further reduce loss. It is, therefore, faster, the reason it is called *light*.

3 Simulations and Data

This section introduces the simulation maps and CMB data that we used in this paper. In this work, we use simulations to train, test, and verify the robustness of the proposed pipeline. The simulated maps contain contributions from inflationary Gaussian CMB anisotropies and include experimental effects and noises.

3.1 CMB Map Simulation

In this work, we used three sets of CMB simulations.

3.1.1 Healpix

The first set consists of HEALPix [1] maps. We use simulations of the two phases of CMB-S4, here referred to as CMB-S4-like(I) and (II). Different types of Healpix simulations differ by noise and beam.

[1] https://healpix.sourceforge.io

3.1.2 FFP10

FFP10 [2] (FFP10), Full Focal Plane, maps are the most realistic simulation of the CMB that the Planck detector observed. These maps incorporate astrophysical foregrounds, instrument noise, and the lensed scalar, tensor, and non-Gaussian CMB components. Moreover, our best estimates the detector's beam, bandpass, time-varying noise properties. Foregrounds in the FFP10 sky model include Galactic thermal dust, the cosmic infrared background, the cosmic infrared background, thermal and kinetic Sunyaev-Zeldovich effects from Galaxy clusters, and Galactic and extragalactic faint point sources. FFP10 maps are used for building the input sky for the E2E simulations [10].

3.1.3 Component-Separation FFP10 (E2E)

The end-to-end (E2E) CMB simulations [3] [10] are the FFP10 maps that they applied by the same component-separation pipelines as the Planck's observational data [11]. Because of this additional property E2E maps are the best choice to be compared to the data. 999 CMB and 300 noise realizations for each component-separation method are available. [11]

3.2 Cosmic String Simulation

As a cosmic string network, We use Nambo-Gutto simulations [12, 13] with tensions in the range $[6 \times 10^{-9}, 6 \times 10^{-6}]$, logarithmically spaced into ten classes, and one class with $G\mu = 0$ that we call it null class. We added string to map to gaussian maps in the following order:

$$T = B[G + G\mu \times S] + N \tag{2}$$

Where G, S, and N represent the primordial Gaussian CMB anisotropies, the template for cosmic string-induced anisotropies, and the instrumental noise. $G\mu$ parametrizes the level of contribution of cosmic strings to CMB fluctuations. The effect of the instrumental beam is encoded in the beam function B.

3.3 Planck's Data

Planck is a mission to observe the temperature and polarization anisotropies of the CMB Radiation Field over the whole sky. Planck is prepared for testing theories of the early universe and the origin of cosmic structure. To extract the CMB maps, the data have been produced using four different methods, COMMANDER, NILC, SEVEM, and SMICA. Each method's full-mission CMB intensity map

[2] pla.esac.esa.int → maps → simulations → cmb

[3] pla.esac.esa.int → advanced search and map operations → simulated maps search → comp-separation

provides the corresponding confidence mask and effective beam transfer function. All CMB products are provided at an approximate angular resolution of 5 arcmins FWHM. In this work, we focused on the SMICA data of Planck.

4 String Detection Pipeline

The approach that used in this section is based on the method that has been applied to the Healpix simulated maps in the paper [7] except that we use FFP10 maps and E2E in addition to Healpix maps. Furthermore, we use machine learning methods to train a model, and after pipeline preparation, we apply it on CMB maps observed by Planck to report the upper bound on the tension of cosmic strings. In this approach, certain features, chosen based on earlier studies, are extracted from the maps and fed as input to machines for classification.

We divide N simulation maps into 12 parts to prepare our data based on Healpix pixelization. In this case, we use the NESTED scheme that arranges the pixel indices in twelve tree structures, corresponding to base-resolution pixels [14]. Then we divide each part again into n patches. So, each map will obtain $N \times 12 \times n$ patches. The reason for dividing maps into smaller patches is to reduce the computational cost without losing much information on the Cosmic Strings network, as the strings mainly leave small-scale anisotropies.

We use ten maps for each simulation type in the intuitive search, E2E, noise-free and noisy version. E2E is similar to observational data more than other simulations. Therefore we will apply the trained models of these maps on observation data. In the LightGBM method, when the data size increases, the accuracy increases, so we use a hundred maps for each noise-free and noisy version of E2E while we use ten maps for other simulations to keep computation costs low.

This approach uses image processing tools, curvelet, and filter, to extract cosmic string properties from the background and then calculate statistical measures as input features of machine learning algorithms. We build different machine learning methods to set bounds of $G\mu$ on CMB data.

4.1 Image processing

In the first step, the sky patches go through two image processing steps of curvelet decomposition [15, 16, 17, 18, 19] and Canny filtering[20], to increase the detectability of the CS signature.

4.1.1 curvelet decomposition of maps

Curvelet transform is suitable for edge detection. This transform has four steps. First, it takes the original image and decomposes it into sub-bands, dividing an image into resolution layers. The

second step dissects each layer into partitions and makes the image smooth. Then renormalize it and in the last step apply Ridgelet analysis on image [21].

In this work, the prepared patches are decomposed into various curvelet components (C5, C6, C7) so that only components with the highest contribution from cosmic strings are kept for further analysis.

4.1.2 Filtering the maps by extended canny algorithm

A canny edge detector is an operator that detects line-like edges in an image. This detector keeps valuable information of the picture to be processed. The filters we use in this work are based on the Canny algorithm. A filter is a 2D array that includes some coefficients and has a central point, and it replaces a pixel of the image with an output of the operator in a point that corresponds to the central point. In the second image processing step, curveleted patches passed through two filters, Scharr and Sobel, to produce gradient maps. Figure 1 shows full-sky that each part of that represents different simulations and examples of our image processing steps. Also, figure 2 shows various steps of image processing on gaussian, string, and combination of gaussian and string with different tensions.

4.2 Statistical Measures

The footprints of the cosmic string network are evident in the curvelet components and, more vividly, in the filtered maps. In this step, we calculate statistical measures of the processed patches to quantify the detectability of their cosmic string imprint. We use the statistical measures that have the best result in paper [7]. Standard deviation and one-point Probably Density function (hereafter, the PDF) performed best.

4.2.1 Standard deviation

We compute standard deviation for each of the processed patches; one number is obtained for each patch, it makes up distribution for each $G\mu$ class. Then baseline would be reduced from the calculations to make the effect of the cosmic string network on total patches more visible. The baseline is the mean of standard deviation on patches that $G\mu = 0$ for them, which means the pure Gaussian CMB simulation patches.

4.2.2 The one-point PDF

Moreover, the PDF is computed for each of the processed patches. PDF returns a distribution for each patch. To correspond one number to each patch, we integrate it on PDF. Like section **??**, baseline

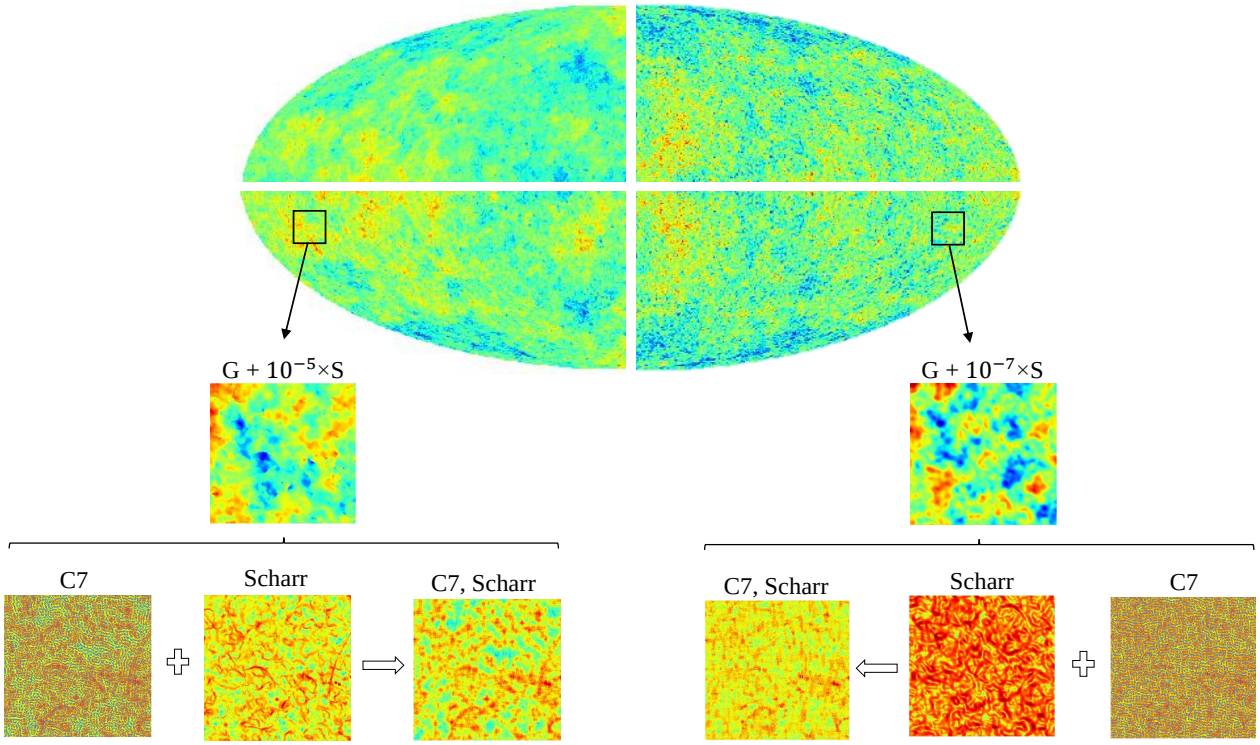

Figure 1: Up: The full-sky map, with $N_{\rm side} = 2048$, is divided into four patches to represent different simulations. The upper half corresponds to the simulations of E2E (right) and CS-induced (left). The lower parts show the sum of the E2E and string simulations with $G\mu = 10^{-5}$ (left) and $G\mu = 10^{-7}$ (right). Notice that all patches demonstrate the same patch of all maps that are rotated and flipped to make one full sky. It is evident that the E2E map with 10^{-5} portion of a string looks like the string map because its string part is significant. Also, the E2E map with 10^{-7} portion of the string is similar to the E2E map because its string fraction is small. Middle: The zoomed-in view of two 256×256 patches with different string contributions. Bottom: The same patches as the middle row, after having passed through parts of the image-processing steps of the proposed CS detection pipeline. Specifically, they correspond to the seventh curvelet component (labeled as C7), the Scharr-filtered image, and the combination of the two.

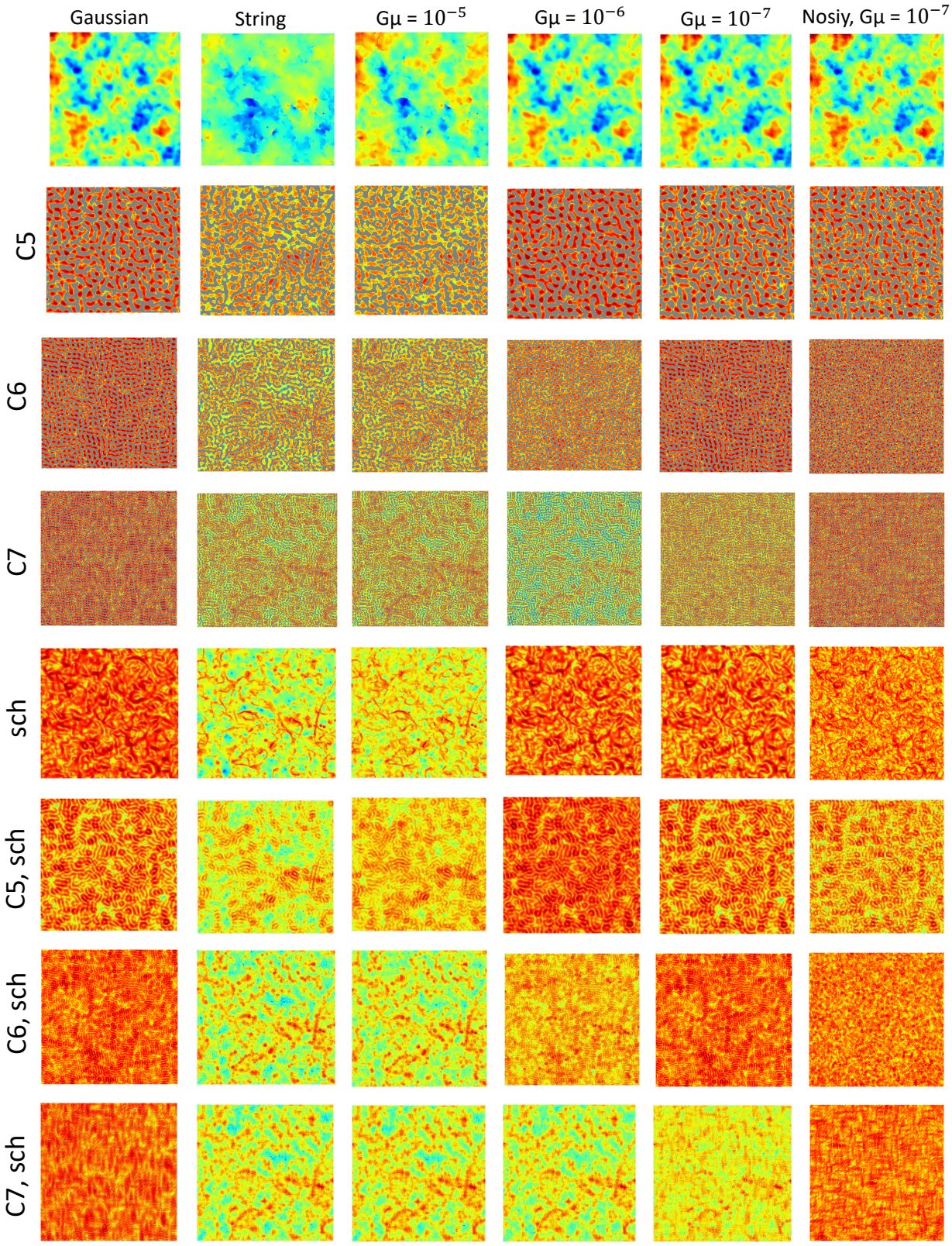

Figure 2: Image processing steps. Columns are patches of E2E gaussian map, string, combination of gaussian with string by tension $G\mu = 10^{-5}$, $G\mu = 10^{-6}$, and $G\mu = 10^{-7}$, and noisy patch of $G\mu = 10^{-7}$. The rows represent 5, 6, 7 components of curvelet decomposition, Scharr filter, and the sum of curvelet components with Sharr filter.

Parameter	Value	Description
learning_rate	0.9	Learning rate
num_leaves	50	Maximum number of tree leaves
max_depth	7	limit the max depth For tree model
min_child_samples	20	Minimum number of samples in a child (leaf)
max_bin	130	Max number of bins that feature values will be bucketed in

Table 1: The hyper-parameters used for the LightGBM classifier

value (PDF of patches without cosmic string) reduces from associated numbers to each patch to make cosmic string effect more visible.

4.3 Analysis of the processed maps

At the end of the processing steps, we have 12 features for each of the total patches, obtained from three curvelet components, two filters, and two statistical measures. This work analyzes these data with machine learning methods and probability value (P-value).

4.3.1 Machine-based Search

We give the obtained features as input data to different machine learning algorithms and train a machine to search for cosmic strings and obtain upper bound on tension. The data of simulation maps is divided into two parts, 75 percent for training and 25 percent for testing.

We test different machine learning methods like Naive Bayes [22], Decision Tree [23], Random Forest [24], K-Nearest Neighbours [25], XGBoost [26] and LightGBM [27]. Comparing these methods, we find that gradient boosting methods, XGBoost and LightGBM, are better than other methods for our case. LightGBM performs better than XGBoost because It is faster. So, we report the results based on LightGBM. Some important hyper-parameters that used in LightGBM algorithm shown in Table 4.3.1. The performance of the model after training is summarized in the confusion matrix. The trained model will then be applied to observation data. Figure 3 shows the confusion matrix of the 11 $G\mu$ classes for the noisy E2E simulations. The columns and rows of this matrix represent the predicted and actual classes, respectively. We see that this matrix's upper right (5×5) diagonal block has a dominant diagonal element with diminishing values as one move to either side. This feature illustrates the low confusion of the machine in learning about the signal we are after in this regime and making relatively robust predictions about the $G\mu$ classes. On the other hand, the lower left (6×6) block of the matrix and the scattered distribution of the predictions for these classes show that the machine has not learned about these classes and the predictions are unreliable. We, therefore, consider the lowest class above this *confused block* as the minimum detectable $G\mu$ or $G\mu_{\rm min}$. However, it should be noted that detectability requires robust predictions based on powerful learning

by the machine and distinguishability from the null class with $G\mu = 0$. Therefore, for a (non-zero) $G\mu$ to be chosen as the detection limit, the majority (e.g., 95%) of the machine predictions for the patches are non-null. Figure 3 shows that $G\mu_{\rm min} = 8.6 \times 10^{-7}$ is the detection limit for the educated method in this work based on the confusion matrix. This is a conservative choice due to the discrete nature of the classes. One could decrease this limit by zooming into the neighborhood of this $G\mu_{\rm min}$ and further train the machine on fine-gridded classes.

4.4 Statistical Search

We also calculate and report the two-tail P-value statistics as a traditional measure to assess the performance of the algorithms and quantify their strength. With the P-value as the criterion, we define the detectability limit, $G\mu_{\rm min}$, as the minimum $G\mu$ with a distribution distinguishable from the null class with a maximum P-value of 0.05. Figure 4 shows the P-values for different levels of $G\mu$ contribution for E2E simulations. For the details of the quantification of the CS-induced deviation and the corresponding *P*-value calculation, see [7].

One could also follow a purely statistical path and calculate the P-value to compare the distribution of the above statistical measures (pdf and variance) with the null class, using $N_{\rm sim} \times N_{\rm patch}$ for different $G\mu$ classes and all observational cases of interest. Figure 4 shows the *P*-values of the $G\mu$s for the E2E case from this approach.

5 Results

The pipelines for cosmic string search were described in section 4. An overview of these algorithms is presented in Figure 5. The results of the analysis and training processes and the performance of the various paths in the pipeline for these different sets of simulations are presented in Table 5. After training the model on the training part of simulation maps with LightGBM, we applied the trained models to the test dataset to obtain these results. We valued the performance of the model with a confusion matrix. The middle column in Table 5 shows the minimum detectable $G\mu$ is the lowest $G\mu$ class that can be distinguished with equal or more than 95% from $G\mu = 0$ class. Also, in the obtained confusion matrix, we calculate *P*-values of predictions for each class with null class and take the minimum detectable $G\mu$ as a class that has maximum *P*-value among *P*-values which less than 0.05. The last column in the Table 5 represents *P*-value-based results. We see that the detectability limit from the confusion matrix is more conservative than the *P*-value. As was discussed in Section **??**, while a direct comparison between the distribution of predictions for different classes can distinguish the class with lower $G\mu$ imprints from the null case, the confusion matrix implies

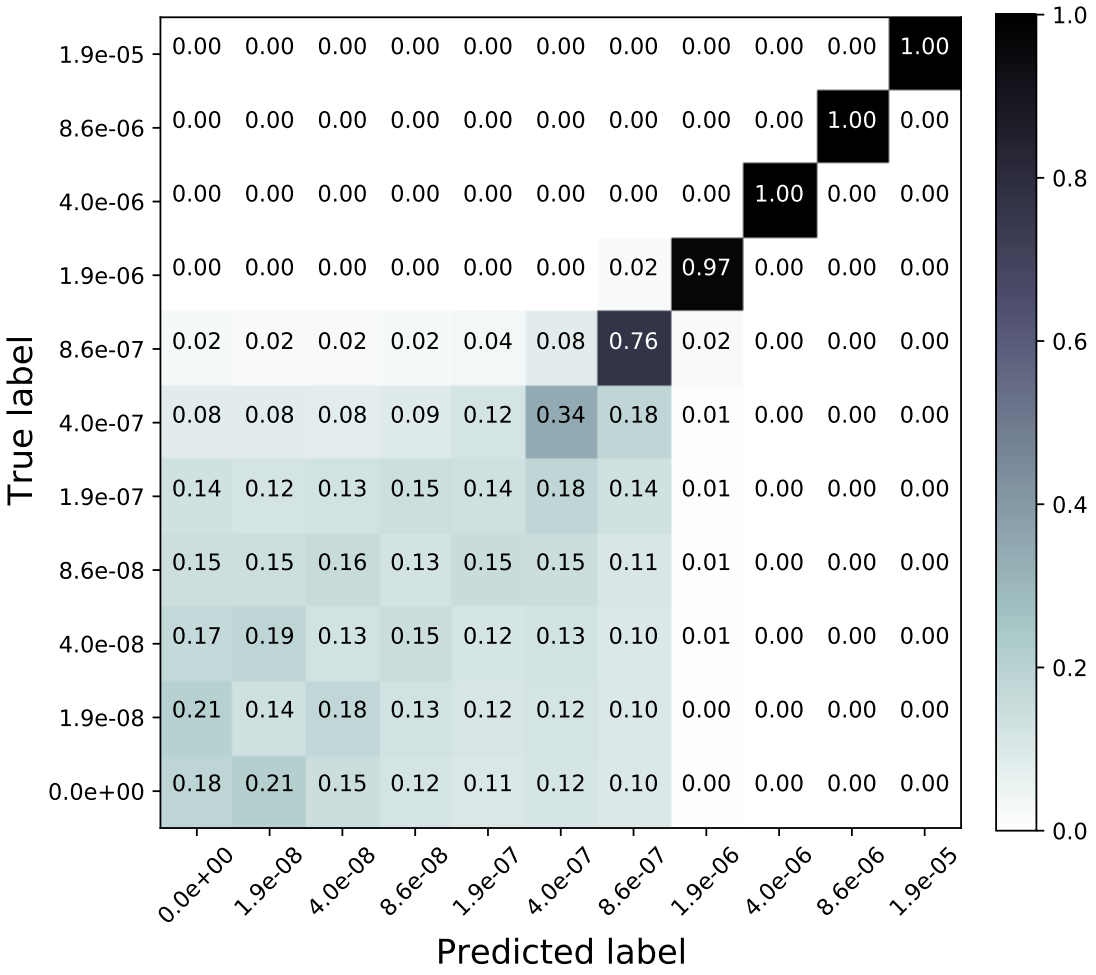

Figure 3: The confusion matrix summarizes the results of the classification of the LightGBM algorithm on E2E simulations. The diagonal elements are the fractions of correct predictions, while the off-diagonals represent the mislabelled. We see the model has not learned much about the first six classes, and the predictions in the 6×6 block in the lower left are quite scattered and hardly centered around fiducial values. We, conservatively, report the first class above this confused block as the minimum detectable class by the model that can be distinguished from the null class, i.e., $G\mu_{\rm min} = 8.6 \times 10^{-7}$.

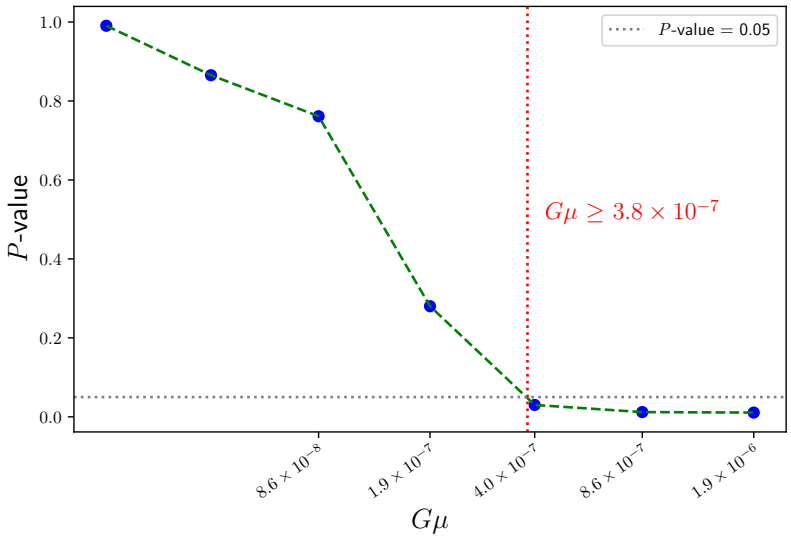

Figure 4: The *P*-values for the $G\mu$ classes, measured by the LightGBM method, quantify the difference between the distributions of the predictions for each class from the null case for the E2E simulations. The vertical line marks the minimum detectable $G\mu$ ($G\mu_{\rm min}$) where the *P*-value crosses the detectability threshold (taken to be 0.05 here) and is found by interpolation.

that the predictions for this low $G\mu$ the region is not fully reliable. We, therefore, conservatively use the bound from the confusion matrix as the main limit. We apply the pipelines developed for the E2E *Planck* simulations to the `SMICA` temperature map. We divide the full (except for the mask) sky *Planck* map into smaller patches, as was done for the simulations, and compare the histogram of predictions for both pipelines with the histograms of the individual $G\mu$ classes from simulations through calculating the *P*-value for each case. The measured *P*-values for all cases are below 0.0027, which corresponds to the 3σ frequentist level. We, therefore, conclude that $G\mu$ is below the minimum detectability limit of the network with 3σ significance. The 3σ upper bound on the $G\mu$ as estimated from *Planck* sky is $G\mu \leq 8.6 \times 10^{-7}$.

We also explored binary classifications with two classes of $G\mu = 0$ as class zero, and $G\mu G\mu_{\rm min}$ as class one, to see whether the upper bounds can be improved. However, we found no significant improvement over the results of Table 5 for the E2E case and therefore the $G\mu \leq 8.6 \times 10^{-7}$ bound is our final result.

6 Summary and discussion

In this work, we developed a pipeline to estimate or put tight upper bounds on the level of cosmic string contribution to the observed CMB anisotropies by *Planck*18 data. We also made forecasts for the $G\mu$ detectability by next-generation CMB experiments. Table 5 summarizes the performance of the pipeline for these various observational scenarios using two different criteria, here labeled by *P*-

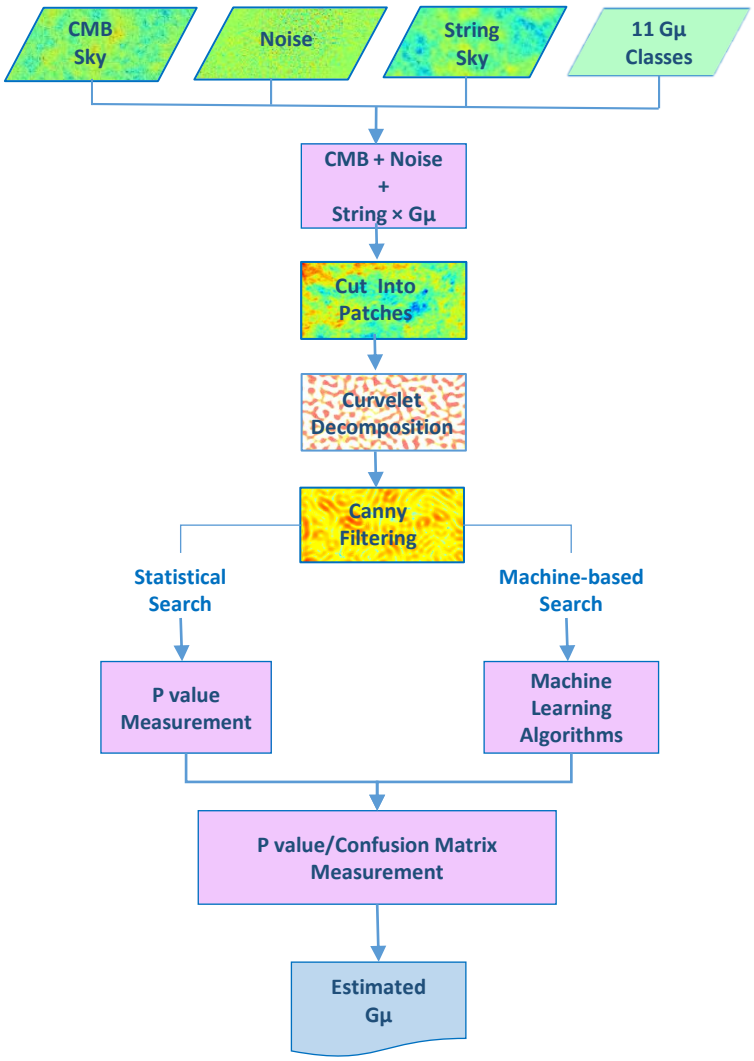

Figure 5: Overview of the data preparation and training pipelines of this work.

Experiment	Confusion Matirx	P-value
CMB-S4-like(II)	4.0	0.5
CMB-S4-like(I)	4.0	0.8
noise-free FFP10	4.0	0.3
FFP10	8.6	1.7
noise-free E2E	8.6	2.1
E2E	8.6	2.7

Table 2: The minimum detectable $G\mu_{\rm min}(\times 10^{-7})$ from LightGBM pipeline based on the confusion matrix and *P*-value measurements for different sets of simulations.

value and confusion matrix. The limits from the confusion matrix are more conservative by ensuring that confused decisions do not drive the predictions of the trained models. Overall, the detectability limits based on the *P*-value and confusion matrix of Table 5 are respectively compared with, and slightly higher than the previous machine-based forecasts of [8] who finds the measurement limit of $G\mu = 1.2 \times 10^{-7}$ in the case of CMB-S4-like simulations. In particular, the LightGBM pipeline put tighter bounds on string tension in the case of noisy *Planck* simulations.

[5] used edge-detection algorithms to set $G\mu = 5.5 \times 10^{-8}$ as a minimum detectability bound. Furthermore, [6] used wavelets and curvelets to provide a limit of $G\mu = 1.4 \times 10^{-7}$ on SPT simulations. In our study, The results obtained from educated search for CMB-S4-like simulations, according to *P*-value and LGBM algorithm, are comparable with these results. Using Bayesian analysis in wavelet space on Planck-like simulations, [9] develop a pipeline that is sensitive to 5×10^{-7}.

We used the machines trained with E2E *Planck* simulations to search for the CS imprints on the observed `SMICA` map. LightGBM model found no observable trace and yielded $G\mu 8.6 \times 10^{-7}$ with 3σ confidence level. It should be noted that this bound corresponds to the discrete class labels in the confusion matrix and are therefore conservative. The actual upper bound could be in between the current bound and the class label right below it, which needs to be investigated with finer grids in the vicinity of the current upper bound.

References

[1] T. W. B. Kibble. Topology of Cosmic Domains and Strings. *J. Phys.*, A9:1387–1398, 1976.

[2] Nick Kaiser and A. Stebbins. Microwave Anisotropy Due to Cosmic Strings. *Nature*, 310:391–393, 1984.

[3] Mark Hindmarsh. Small scale microwave background fluctuations from cosmic strings. *Astrophys. J.*, 431:534–542, 1994.

[4] Albert Stebbins and Shoba Veeraraghavan. Beyond the small-angle approximation for mbr anisotropy from seeds. *Physical Review D*, 51(4):1465, 1995.

[5] Andrew Stewart and Robert Brandenberger. Edge Detection, Cosmic Strings and the South Pole Telescope. *JCAP*, 0902:009, 2009.

[6] Lukas Hergt, Adam Amara, Robert Brandenberger, Tomasz Kacprzak, and Alexandre Réfrégier. Searching for cosmic strings in cmb anisotropy maps using wavelets and curvelets. *Journal of Cosmology and Astroparticle Physics*, 2017(06):004, 2017.

[7] A Vafaei Sadr, SMS Movahed, M Farhang, C Ringeval, and FR Bouchet. A multiscale pipeline for the search of string-induced cmb anisotropies. *Monthly Notices of the Royal Astronomical Society*, 475(1):1010–1022, 2017.

[8] A Vafaei Sadr, M Farhang, SMS Movahed, B Bassett, and M Kunz. Cosmic string detection with tree-based machine learning. *Monthly Notices of the Royal Astronomical Society*, 478(1):1132–1140, 2018.

[9] Jason D McEwen, SM Feeney, Hiranya V Peiris, Yves Wiaux, C Ringeval, and FR Bouchet. Wavelet-bayesian inference of cosmic strings embedded in the cosmic microwave background. *Monthly Notices of the Royal Astronomical Society*, 472(4):4081–4098, 2017.

[10] Peter AR Ade, N Aghanim, M Arnaud, M Ashdown, J Aumont, C Baccigalupi, AJ Banday, RB Barreiro, JG Bartlett, N Bartolo, et al. Planck 2015 results-xiii. cosmological parameters. *Astronomy & Astrophysics*, 594:A13, 2016.

[11] Y Akrami, M Ashdown, J Aumont, C Baccigalupi, M Ballardini, AJ Banday, RB Barreiro, N Bartolo, S Basak, K Benabed, et al. Planck 2018 results. vii. isotropy and statistics of the cmb. *arXiv preprint arXiv:1906.02552*, 2019.

[12] David P Bennett and Francois R Bouchet. High-resolution simulations of cosmic-string evolution. i. network evolution. *Physical Review D*, 41(8):2408, 1990.

[13] Christophe Ringeval, Mairi Sakellariadou, and François R Bouchet. Cosmological evolution of cosmic string loops. *Journal of Cosmology and Astroparticle Physics*, 2007(02):023, 2007.

[14] Krzysztof M Gorski, Benjamin D Wandelt, Frode K Hansen, Eric Hivon, and Anthony J Banday. The healpix primer. *arXiv preprint astro-ph/9905275*, 1999.

[15] David L Donoho and Mark R Duncan. Digital curvelet transform: strategy, implementation, and experiments. In *AeroSense 2000*, pages 12–30. International Society for Optics and Photonics, 2000.

[16] Emmanuel J Candes and David L Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document, 2000.

[17] Emmanuel J Candès and David L Donoho. Curvelets and curvilinear integrals. *Journal of Approximation Theory*, 113(1):59–90, 2001.

[18] Emmanuel J Candès and Franck Guo. New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002.

[19] Emmanuel Candes, Laurent Demanet, David Donoho, and Lexing Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006.

[20] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[21] Emmanuel J Candes and David L Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document, 2000.

[22] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[23] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[24] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[25] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[26] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[27] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.