Detection of Cosmic Strings in CMB Polarization Maps Using a Convolutional Neural Network

William Laplante[1,*] and Robert Brandenberger[2,†]

[1]*Department of Physics, McGill University, Montréal. QC, H3A 2T8, Canada*
[2]*Department of Physics, McGill University, Montréal, QC, H3A 2T8, Canada*

(Dated: November 14, 2022)

We present the architecture and performance of a convolutional neural network (CNN) used to detect cosmic strings in cosmic microwave background (CMB) polarization maps. By training this model on simulated CMB polarization maps, with and without cosmic strings, the CNN can produce predictions on sky maps having a cosmic string network with string tension as low as $G\mu = 5 \times 10^{-8}$. Past this value of string tension, we find that the model is incapable of generalizing its predictive power to new data. Furthermore, we show that even though we train the CNN with maps of specific string tension, the CNN performs well with maps of any string tension within the model's limits. This showed that even if the model isn't trained on maps with the same tension as some cosmic string, it could still detect accurately its presence.

I. INTRODUCTION

Cosmic strings are linear topological defects appearing in the phase transition of matter in the very early Universe. In particular, they are lines of trapped energy density which are suspected to contribute to the power spectrum of curvature fluctuations giving rise to the large scale structures observable today. Their gravitational effects are parametrized by a dimensionless scalar, namely the string tension $G\mu$, where G is Newton's gravitational constant. We can relate the string tension μ of a cosmic string network to the energy scale η of the particle physics model by

$$\mu \sim \eta^2. \tag{1}$$

From equation I, we observe that the bigger the energy scale η, the more prominent the cosmic string's gravitational effects will be, because μ grows as η increases. Since greater string tensions make a cosmic string network easier to probe, studies of cosmic strings are great to test the larger effects predicted by particle physics models ($\eta \sim 10^{15}$ GeV), in contrast to accelerator experiments, which tend to probe lower values of η [8].

Cosmic strings are predicted to form in many inflationary models such as supergravity models and brane inflation models. In any theory admitting stable strings to form after inflation, a scaling solution inevitably results. That is, once a network of strings is laid down, it will persist through time and maintain its statistical distribution in space if scaled with the Hubble radius [3]. A network of cosmic strings thus leaves a scale-invariant footprint in the Universe, which implies that they can be detected at the present time through various observational windows. In the past years, the research done in cosmic string detection has been mainly focused on Cosmic Microwave Background (CMB) temperature and 21 cm intensity maps. These observational windows show promise because cosmic strings leave a distinctive mark in these maps. Indeed, when long cosmic strings move in space, they accumulate matter and create a wedge-shaped region behind the string with twice the background density, which corresponds to the cosmic string wake [6]. Since one dimension of the wake is in the string's direction, while the other is in the direction of motion, we end up with rectangular regions of over-density. These regions perturb CMB radiation and 21 cm lines so that the same rectangular shapes appear in 21 cm maps, thus making 21 cm redshift surveys useful for cosmic string detection [4].

In recent years, a new observational window, namely the CMB polarization, has given rise to an opening in the search of cosmic strings. Similarly to 21 cm maps, it is the ionization in the over-density regions of wakes that creates extra polarization in the form of scale-invariant rectangular patches in the sky [6]. This knowledge of the distinctive patterns laid down by cosmic string wakes in the polarization maps reduces the problem of string detection to the identification of rectangular regions of over-polarization. The main tools used in past studies to look for cosmic strings were edge and shape detection algorithms such as the canny algorithm [5], wavelets, and curvelets [8]. In this study, we instead put our focus on modern techniques of machine learning and image processing such as Convolutional Neural Networks (CNN) to perform cosmic string detection. In machine learning jargon, we tackle a supervised classification problem, which entails that we use labeled data to perform classification. Broadly speaking, we construct a database of simulated CMB polarization maps with and without a cosmic string signal, and use a machine learning algorithm to predict whether or not a given map contains a network of cosmic strings. Already, strong constraints have been produced on the tension of a cosmic string network : $G\mu \leq 1.7 \times 10^{-7}$

*Electronic address: william.laplante2@mail.mcgill.ca
†Electronic address: rhb@physics.mcgill.ca

at 95% confidence [7]. Consequently, the statistical tool we construct must be able to detect the presence of cosmic strings at lower values of string tension than the presented limit if it is to be of any utility. In addition, we aim to make the algorithm generalize well to new data, else our efforts would be futile as it could not be applied on future CMB polarization surveys.

In this paper, we present the details of the model's architecture used for cosmic string detection along with the results of the model's performance. We first do a quick review on the underlying theory of cosmic strings in section II, and follow up with section III, where the process by which maps are simulated is introduced. Then, we explain the details of the learning algorithm in section IV and discuss the algorithm's performance in V. In section VI, we summarize our results and present the prospects for future work.

II. REVIEW OF COSMIC STRINGS

Cosmic strings form after a phase transition in the early universe during the spontaneous break of an internal symmetry of field space. To explain this phenomena, we consider a simple model with potential

$$V(\phi) = \frac{\lambda}{4}(|\phi|^2 - \eta^2)^2, \tag{2}$$

where ϕ is a complex scalar field, η is the vacuum expectation value of $|\phi|$ and λ is a coupling constant. In this example, the vacuum manifold $\mathcal{M}$, which consists of the set of field values that minimize the potential, is homotopically equivalent to the circle S^1. Moreover, in thermal equilibrium, $V(\phi)$ has finite temperature corrections, and more specifically there is an extra contribution of

$$\Delta V_T(\phi) \sim T^2|\phi|^2, \tag{3}$$

where T is the temperature. This implies that the field symmetry is unbroken above some critical temperature T_c, and so the average value of ϕ is 0 at each point in space. But as the Universe's temperature reduces below T_c, values of ϕ at each point in space will eventually take on values in $\mathcal{M}$ since the state becomes unstable. Some of the values in $\mathcal{M}$ taken by ϕ will be out of causal contact, implying there can be no correlation between them. We can thus infer that there is a probability of $\mathcal{O}(1)$ that field values will form an incontractible loop in $\mathcal{M}$ when evaluated for some loop $\mathcal{C}$ of radius t. We conclude that in a region of space bounded by $\mathcal{C}$, there exists a point where $\phi = 0$. Around any such points, there is trapped potential, and thus spatial gradient energy. Following the energy minimization argument, we conclude that together, these points a form line, and it is this line that we define as a cosmic string. Cosmic strings are thus lines of points with $\phi = 0$ surrounded by a tube of trapped energy, where the width of the tube w is given by

$$w = \lambda^{-1/2}\eta^{-1}. \tag{4}$$

In summary, in a field theory admitting cosmic string solutions, a network of lines, or cosmic strings, will form at the symmetry breaking phase transition occurring at some time $t = t_c$. We usually interpret the network of strings as a random walk with curvature radius $\zeta(t)$ bounded from above by the Hubble length t. A corollary of the causality argument used to infer the formation of strings is that the network of strings survives at all times $t > t_c$. Hence, this makes possible the search of cosmic strings in current cosmological observations [3].

III. CMB POLARIZATION MAPS SIMULATIONS

To produce CMB polarization maps, we start by deriving the polarization amplitude of CMB photons assuming a network of cosmic strings has been laid down. When unpolarized CMB light with quadrupolar anisotropy scatters off free electrons, the scattered radiation becomes polarized. We expect the magnitude of the polarization to be dependent on the Thomson cross section σ_T and on the integral of the number density of electrons along the radiation's geodesic. We thus express the magnitude of polarization of radiation coming from a direction $\boldsymbol{n}$, $P(\boldsymbol{n})$, as

$$P(\boldsymbol{n}) \approx \frac{1}{10}(\frac{3}{4\pi})^{1/2}\tau_T Q, \tag{5}$$

where Q is the temperature quadrupole, and

$$\tau_T = \sigma_T \int n_e(\chi) d\chi, \tag{6}$$

with the integral being along the null geodesic in terms of conformal time. Now, we assume that a cosmic string wake was laid down at some time t_i, and has grown up until some time t, which marks the time at which a CMB photon passes through the wake. We then estimate the integral 6 by

$$\tau_T \approx 2\sigma_T n_e(t, t_i)(z(t) + 1)h(t, t_i), \tag{7}$$

and substitute equation 7 back into equation 5. Using the results of [6] for the height $h(t, t_i)$ and number density of free electrons $n_e(t, t_i)$, we obtain the following expression for the polarization signal :

$$\frac{P}{Q} \approx \frac{24\pi}{25}(\frac{3}{4\pi})^{1/2}\sigma_T f G\mu v_s\gamma_s \times \Omega_B \rho_c(t_0) m_p^{-1} t_0 (z(t)+1)^{5/2}, \tag{8}$$

where v_s is the string's velocity, γ_s is the relativistic gamma factor, f is the ionization fraction, Ω_B is the baryon fraction, m_p^{-1} is the inverse proton mass, t_0 is the

current time, and ρ_B is the energy density in baryons. From equation 8, we can then create the polarization maps. This is done by drawing rectangles of various sizes with pixel intensity determined by equation 8. The rectangles simulate the wakes passing by, and thus they are drawn so that they fade away in some direction. The angular dimensions of the rectangles correspond to the co-moving size

$$c_1 t_i(z(t_i)+1) \times v_s \gamma_s t_i(z(t_i)+1). \tag{9}$$

Once the first wakes are laid down, we perform a double sum over all wakes crossed by a photon over all times steps. The result of this computation yields the cosmic string signal. To obtain the final simulated map, there remains to add a layer of Gaussian noise coming from primordial perturbations on top of the cosmic string signal. To produce this noise, we follow the method used in [5]. In short, we construct a Fourier space from the E-mode polarization power spectrum using the CAMB module, and convert to position space by Fourier transform. We end up with a map of fluctuations for E-mode polarization which simulates the density variations of the early Universe. Together, the noise and the cosmic string signal form a CMB polarization map, as shown in figure 1. In these simulated maps, the string tension can be adjusted as desired. The lower the string tension, the weaker the cosmic string signal is in the map. In this research, it is this parameter that we try to reduce as much as possible when developing the algorithm used to detect the presence of cosmic strings.

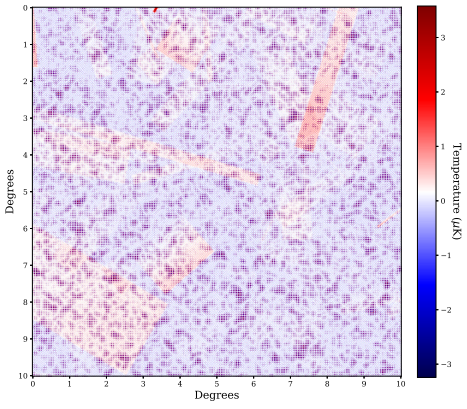

FIG. 1: *Simulated CMB polarization map.* Here, the base string tension used to produce the cosmic string signal is of $G\mu = 10^{-5}$ to ensure proper visualization of the signal.

IV. LEARNING ALGORITHM

We start by generating a database of simulated maps. One half of the maps contain a network of cosmic strings with some pre-determined string tension, while the other half are strictly made of noise. Each map is labeled in a binary fashion, so that it either contains a cosmic string signal, or it doesn't. Determining the presence of a cosmic string signal in a map thus falls into the category of a supervised binary classification problem. That is, we have a dataset $D = \{(\boldsymbol{x_i}, \boldsymbol{y_i})|i = 1...N\}$, where $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ belong to some input space $\boldsymbol{A}$ and some label space $\boldsymbol{B}$ respectively, and the goal is to find a mapping $f : \boldsymbol{A} \longrightarrow \boldsymbol{B}$ which assigns binary labels to data points. In the present case, the $\boldsymbol{x_i}$'s are 400×400 arrays of pixels representing the CMB polarization maps, while the $\boldsymbol{y_i}$'s are binary labels determining the presence of a cosmic string network in the input map. Due to the structure of the polarization maps (image-like), and to the the geometric features the algorithm is expected to learn from the input (rectangular shape of wakes), the learning algorithm we choose is the convolutional neural network (CNN). In short, a CNN chains together multiple instances of a convolution layer, a non-linear layer and a pooling layer, and feeds the output of these operations to a fully connected neural network which performs the classification [2]. The convolution step convolves a set of kernels (matrices of weights) with a map in order to locate specific features in the input, and the non-linear layer applies an activation function on the result of the convolution operation in order to introduce non-linearity in the model. Once the convolution and activation steps are completed, the resulting output, namely the feature maps, go through a pooling layer in order to down sample. The idea of down sampling is to create a lower resolution version of an input that still contains the important structural elements, without the fine details that might be irrelevant for the task at hand [1]. This allows a simplification of the model, since the predictive function's number of parameters greatly reduces when performing down sampling. The specific architecture of the model presented in this paper is shown in table I. We see that the model chains together three convolution and pooling steps, and finishes with a single dense layer. This structure leads to a total of 13,793 trainable parameters. An important feature of the model is the average pooling step. Introducing it made the model improve by an order of magnitude on string tension instantaneously. This might have been because the average pooling layer smooths out the image [11], and thus helped the algorithm deal with the noisy background in the CMB polarization maps.

To evaluate and estimate the accuracy of the model, we use stratified k-fold cross-validation. This method consists of randomly splitting the data into k different folds, so that each fold is an appropriate representation of the data, and then train the model on $k-1$ fold while keeping the kth fold for testing the accuracy [9]. This is repeated k-times, so that the model has the chance to train on all the data. In order to obtain an estimate of the model's accuracy on some dataset, we take the average and standard deviation of the k accuracy values obtained when performing stratified k-fold. This gives us a single

Layer	Output Shape	Param #
Convolution 2D	(None, 398, 398, 32)	320
Average Pooling 2D	(None, 132, 132, 32)	0
Convolution 2D	(None, 130, 130, 16)	4624
Average Pooling 2D	(None, 65, 65, 16)	0
Convolution 2D	(None, 63, 63, 8)	1160
Average Pooling 2D	(None, 31, 31, 8)	0
Flatten	(None, 7688)	0
Dense	(None, 1)	7689

TABLE I: *Specifications of the CNN.* The convolution layers have 32, 16, and 8 filters respectively, all with kernel size of (3,3). The average pooling layers all have a pooling size of (2,2). we use the hyperbolic tangent as our activation function, and for the dense layer we use the sigmoid function. The remaining parameters are set to default. To compile the CNN, we use binary cross entropy as our loss function, and the adam algorithm to optimize our model. The epochs and batch size vary from one fitting to another in order to ensure the algorithm converges properly. Note that the CNN was implemented using TensorFlow's keras module.

metric for both feedback on the model's performance and for presenting the model's ability to classify the CMB polarization maps.

V. RESULTS AND DISCUSSION

We first train the CNN and test its accuracy on CMB polarization maps generated with different values of string tension. The results are illustrated in figure 2. We observe that the model's accuracy abruptly drops when approaching a string tension of $G\mu = 10^{-8}$. More precisely, the model has trouble generalizing to new data, as shown by the discrepancy between the accuracy of the train set and test set. Note that as the string tension decreases, we expect the model to converge to an accuracy of 50%, meaning it has no predictive power. This is due to the binary nature of the classification problem at hand. Although we weren't able to cross the regime of $G\mu = 10^{-8}$, we still managed to predict well up until $G\mu = 5 \times 10^{-8}$, since at this value of string tension, the accuracy result is reasonable and the model generalizes well. This implies the objective was accomplished, since we were able to make predictions on simulated maps with a string tension smaller than the theoretical upper bound of $G\mu = 1.7 \times 10^{-7}$.

To improve the model's performance, common practice would suggest that we use more data to train the CNN [10]. In figure 3, we illustrate how the model's accuracy varies when increasing the total amount of data used to train and test the model. We show this result for 4 different string tension values. First, we notice the expected tendency that more data leads to more predictive power. This holds true except for $G\mu = 2.5 \times 10^{-8}$, which suggests that to improve the model for smaller values of string tension, we would have to change its architecture rather than feed the model more data. Secondly, an important effect of using more maps to train and test the CNN is that it reduces the variance of the model's predictions. In the figure, we observe that as more data is used, the smaller the error bars on the accuracy become. Exceptions occur when approaching string tensions of 5×10^{-8}, but this is mainly due to the fact that this regime of string tensions corresponds to the model's limit, that is where the CNN becomes unable to make any substantial predictions.

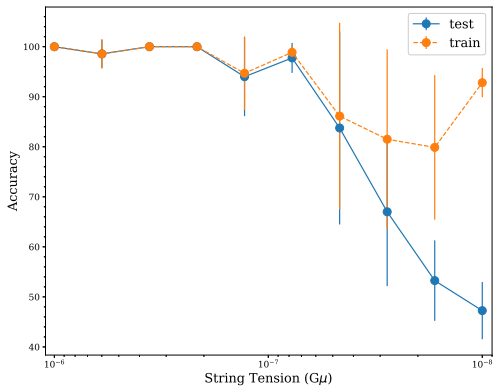

FIG. 2: *Accuracy plotted against string tension for test and train set.* In this figure, the accuracy represents the number of maps classified properly out of all the maps in some data set. The accuracy is tested both on the train set, which is the data on which the model was trained, and on the test set, which represents new data to the model. The accuracy metric and its error are determined by averaging over the accuracy results from the k folds of the cross validation step, and by taking the standard deviation.

In figure 2, the model was re-trained for each data point. The advantage of doing so is that we can push the boundaries of our model in terms of how low we can get

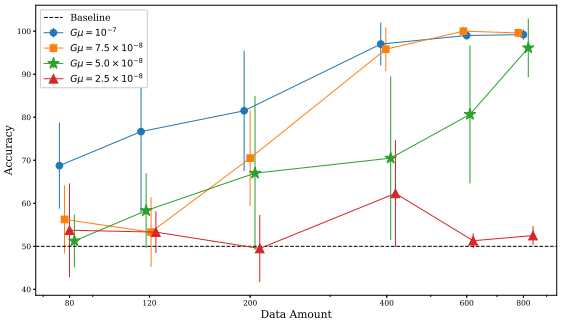

FIG. 3: Accuracy of CNN plotted against amount of data used to train it.

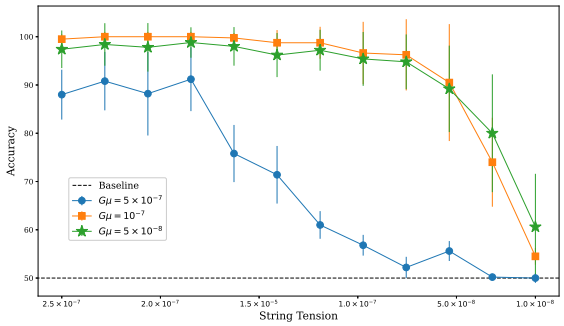

FIG. 4: Accuracy plotted against string tension when keeping the string tension fixed.

on string tension. On the other hand, it is instructive to explore the model's sensitivity by testing its performance on maps with different string tensions than the one it was trained with. In figure 4, we illustrate the model's performance on a range of string tension values when trained on maps with $G\mu = 5 \times 10^{-7}$, $G\mu = 10^{-7}$ and $G\mu = 5 \times 10^{-8}$. We see that the accuracy monotonically decreases when string tension decreases, except for the case where we train our model on maps with high values of string tension ($G\mu = 5 \times 10^{-7}$). This indicates that the CNN is picking up the cosmic string signal's features in the maps, since it doesn't customize to maps with any particular string tension and is capable of making predictions on maps with lower string tensions than the one used during the training step. This result is promising since it shows that in future surveys of CMB polarization maps, a CNN would be able to pickup the presence of a cosmic string network even though it wasn't trained on the same string tension as the network's.

VI. CONCLUSION

In this research work we built a CNN capable of predicting the presence of cosmic strings in simulated CMB polarization maps. The point of doing so was to ultimately be able to detect cosmic strings in future surveys. We were able to make substantial predictions on maps with string tensions of $G\mu = 5 \times 10^{-8}$. At lower values of string tension, the main problem was that the model wasn't capable of generalizing, meaning it tested well on the data on which it was trained, but performed poorly on new data. Future research work on the topic should explore two main avenues. The first would consist of increasing the complexity of the CMB polarization maps. Doing so would improve the chances of cosmic string detection, since the models would be trained on maps replicating better future surveys. The second avenue would consist of repeating the work done in this paper, but instead use B-mode polarization to produce the maps. We expect that B-mode polarization would yield better results in terms of string tension since the Gaussian noise contribution is much smaller due to primordial fluctuations producing only E-mode polarization to linear order.

Acknowledgement

I'd like to thank professor Robert Brandenberger for supervising me on this research.

[1] N. Akhtar and U. Ragavendran, *Interpretation of intelligence in cnn-pooling processes: a methodological survey*, Neural computing and applications, 32 (2020), pp. 879–898.

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, *Understanding of a convolutional neural network*, in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.

[3] R. H. Brandenberger, *Searching for cosmic strings in new observational windows*, Nuclear Physics B - Proceedings Supplements, 246-247 (2014), p. 45–57.

[4] R. Ciuca, O. F. Hernández, and M. Wolman, *A convolutional neural network for cosmic string detection in cmb temperature maps*, Monthly Notices of the Royal Astronomical Society, 485 (2019), p. 1377–1383.

[5] R. J. DANOS and R. H. BRANDENBERGER, *Canny algorithm, cosmic strings and the cosmic microwave background*, International Journal of Modern Physics D, 19 (2010), p. 183–217.

[6] R. J. Danos, R. H. Brandenberger, and G. Holder, *Signature of cosmic string wakes in the cmb polarization*, Physical Review D, 82 (2010).

[7] C. Dvorkin, M. Wyman, and W. Hu, *Cosmic string constraints from wmap and the south pole telescope data*, Physical Review D, 84 (2011).

[8] L. Hergt, A. Amara, R. Brandenberger, T. Kacprzak, and A. Réfrégier, *Searching for cosmic strings in cmb anisotropy maps using wavelets and curvelets*, Journal of Cosmology and Astroparticle Physics, 2017 (2017), p. 004–004.

[9] S. Purushotham and B. Tripathy, *Evaluation of classifier models using stratified tenfold cross validation techniques*, in International Conference on Computing and Communication Systems, Springer, 2011, pp. 680–690.

[10] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, *A review of machine learning for the optimization of production processes*, The International Journal of Advanced Manufacturing Technology, 104 (2019), pp. 1889–1902.

[11] D. Yu, H. Wang, P. Chen, and Z. Wei, *Mixed pooling for convolutional neural networks*, in Rough Sets and Knowledge Technology, Cham, 2014, Springer International Publishing, pp. 364–375.