

TCN: Pioneering Topological-Based Convolutional Networks for Planetary Terrain Learning

Yuzhou Chen,^{1,5} Yuliya Marchetti,² Elena Sizikova,³ Yulia R. Gel^{4,5}

¹Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA

²Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA

³Center for Data Science, New York University, New York, NY 10011, USA

⁴Department of Mathematical Sciences, University of Texas at Dallas, Richardson, TX 75080, USA

⁵Energy Storage and Distributed Resources Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
yc0774@princeton.edu, yuliya.marchetti@jpl.nasa.gov, es5223@nyu.edu, ygl@utdallas.edu

Abstract

Implementations of artificial intelligence (AI) based on deep learning (DL) have proven to be highly successful in many domains, from biomedical imaging to natural language processing, but are still rarely applied in the space industry, particularly for onboard learning of planetary surfaces. In this project, we discuss the utility and limitations of DL, enhanced with topological footprints of the sensed objects, for multi-class classification of planetary surface patterns, in conjunction with tactile and embedded sensing in rover exploratory missions. We consider a Topological Convolutional Network (TCN) model with a persistence-based attention mechanism for supervised classification of various landforms. We study TCN’s performance on the Barefoot surface pattern dataset, a novel surface pressure dataset from a prototype tactile rover wheel, known as the Barefoot Rover tactile wheel. Multi-class pattern recognition in the Barefoot data has neither been ever tackled before with DL nor assessed with topological methods. We provide insights into advantages and restrictions of topological DL as the early-stage concept for onboard learning and planetary exploration.

Introduction

From seeking signs of past life on Mars to collecting samples of lunar rock, robotic rovers are planetary exploration vehicles with a high level of onboard intelligence that perform complex tasks in science data gathering. For example, NASA’s Mars Exploration Rovers (MER) is one of the longest deployments of robotic intelligence on remote planetary surfaces. Another example is the most recent rover Zhurong, part of China’s first Mars mission. An emerging instrumental concept of tactile wheels equipped with various in-situ sensors, such as tactile technology, directly on the wheel can add additional information to the planetary exploration rovers by providing them with a “sense of touch”. Sensing modality enabled by the tactile wheel has a premise to become the primary planetary mission driver, particularly in conjunction with mapping efforts at cruising, rather than crawling speeds for future Moon and Mars rover missions. Moreover, the tactile wheel has the potential to enhance mobility missions to Icy Moons due to the even more uncertain terrain properties and the need for increased autonomy. Learning such uncertain terrain properties requires development of novel machine learning (ML) techniques which

can not only capture sophisticated characteristics of planetary surfaces, but also demonstrate computational efficiency for onboard applications.

This paper is motivated by the challenging question of whether we can potentially address such needs in rover exploration missions using deep learning (DL) models, enhanced with topological footprints of terrain. In particular, we aim to explore the utility and limitations of topological DL as the primary tool for detecting various terrain classes in rover missions. We provide experiments on a novel surface pressure dataset from a prototype tactile rover wheel, named the Barefoot Rover tactile wheel, making a case for potential applications of topological DL in terrain exploration and autonomous driving. The embedded sensors, like the pressure sensor on a wheel, provide additional capabilities for development of the autonomous systems and could be an important component of future spacecraft exploring Earth, space and unknown environments. These new, under-explored applications for DL currently require higher levels of computational efficiency and memory utilization due to hardware and cost constraints. While DL, especially enhanced with topological information (Hofer, Kwitt, and Niethammer 2019), has proven to deliver superior accuracy, its computational costs is one of the primary roadblocks for onboard applications. Previous work (Chen, Marchetti, and Gel 2021) has shown that adding a topological layer to graph convolutional networks (GCNs) can yield noticeable gains for binary classification of planetary rock patterns collected by the similar tactile wheel. However, such topological GCNs are computationally prohibitive for exploratory missions.

Here we advance this analytic proof-of-concept method for rover exploration further into two directions. First, we consider a more realistic scenario for rover missions and learn more complex terrain types from pressure images of various landforms, such as sharp and smooth dunes, bedrock, gullies and pebbles. Second, we significantly reduce computational costs. To achieve these goals, in contrast to the previous study, we use an ensemble of topological summaries that are obtained from rotation augmentation. Such *topological meta-representation* gathers a richer knowledge on the hidden shape properties of planetary terrains, making it possible to utilize a simpler model (compared to GCN), and hence provide considerable computational gains with less training data necessary. In particular, to adaptively learn the importance weights for the resulting

embeddings in the topological meta-representation, we propose a *persistence-based attention mechanism*, making the first step toward the paradigm of *attention-based topological representation learning*. We then introduce *Topological Convolutional Networks (TCN)*, i.e., topologically-enhanced convolutional neural networks, that are able to efficiently uncover unseen local shape signatures of landforms, improve classification performance, and substantially reduce computational costs. Furthermore, some TCN model training can be done offline prior to deployment, with only incremental training required to update the model, allowing us to better balance accuracy and computational costs. This new approach does not require a priori knowledge of topological signals and noise, i.e., maximum signal is extracted from all available topological information; in a sense, we can say that we can quantify topological uncertainty. These innovations can make DL tools more attractive for the deployment onboard and for streaming applications, such in terrain exploration in space, as we show with the Barefoot dataset experiments.

While the current computational costs of any DL tool, including the proposed TCN, are still high for onboard deployment, this project provides a suite of important messages toward eventually implementing DL within AI solutions for rover systems. In particular, our findings suggest that topological footprints, especially the topological meta-representation mechanisms, may bring highly valuable information about the hidden structural organization of multiple planetary landforms and assist in generalization of DL results. We believe that the direction toward topological transfer learning with extensive offline experiments and only limited onboard updates and matching of topological footprints may be one of the most promising AI solutions for planetary exploration missions.

Related Work

Research in DL methods for terrain classification utilizes natural images taken by orbiter cameras (Wagstaff et al. 2018; Rothrock et al. 2016; Kerner et al. 2019) and by autonomous vehicles (Schmidt and Cheein 2019). Terrain classification is studied in the context of robotic exploration and operational safety, particularly with the newly found applicability to tactile sensors (Zürn, Burgard, and Valada 2020; Chen, Rastogi, and Norris 2021). Recent efforts have been dedicated to extending both ML and DL methods for terrain image classification. For example, Wagstaff et al. (2018) use a neural network architecture based on an autoencoder to capture and explain novel features in multispectral images. Additionally, Marchetti et al. (2020) utilize tree-based Stochastic Gradient Boosting (SGB) to extract information from in-situ sensors and train models for terrain type classification and slip regression. Kerner et al. (2020) compare the performance of four detection methods and detect novel geology on multispectral images from planetary instrument datasets. To incorporate local and global information, Chen, Marchetti, and Gel (2021) propose a GCN-based model to capture geometric and topological features via a terrain image and its topological summary. Unlike aforementioned DL, our approach is based not on a single topological summary of a given image, but considers an ensemble

of topological summaries, allowing us to enhance topological knowledge representation and, by virtue of it, improve pattern recognition performance and associated computational efficiency.

Data Description

The Barefoot surface pressure dataset (Lightholder et al. 2021) is a collection of non-traditional images of terrain collected from experiments with a pressure sensor wrapped around a prototype tactile rover wheel. These images show imprints of various terrain types, e.g., rocks, dunes, gullies, as the wheel is rolling over the ground. An example of such an imprint and the wheel experiment are shown in Figures 1(a) and (b). For instance, Figure 1(b) shows an example of experiment setup where the Barefoot Rover mobility cart with the tactile wheel mounted on it sits in a metal trough over regolith with letters Jet Propulsion Laboratory (JPL) spelled in small rocks on the surface. Background and details for the Barefoot project, the dataset and data processing can be obtained in Marchetti et al. (2020) and Chen, Marchetti, and Gel (2021).

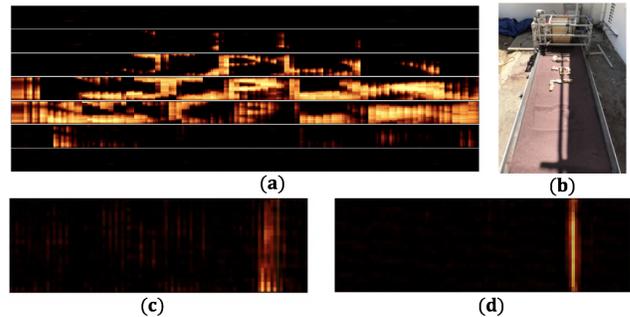


Figure 1: An example image from the Barefoot surface pressure dataset (Lightholder et al. 2021). A processed image of the pressure sensor (a) obtained for an experiment with a rolling wheel that forms letters in rocks (b). Calibrated pressure pad image, with the wheel slipping (c) and resting on a rock (d). The striping effect is produced by the grousers.

Figures 1(c) and (d) show an example of fully calibrated pressure sensor imprint for different pressure signatures, with a wheel resting on top of a rock and with a wheel on a sandy material, experiencing slippage. Most of the pressure images have calibrated pressure at around 0, since at any given time the wheel is only experiencing pressure over a very small area, while the rest of its circumference is not in contact with the ground. Marchetti et al. (2020) collected over a thousand experiments across different materials, terrain patterns and slippage values. We use a subset of the experiments (Lightholder et al. 2021) for surface patterns, including various rock types, to classify pressure sensor images into eight classes: bedrock, flat, gullies, pebbles, rock-above, rock-below, sharpdunes, smoodunes. Further pre-processing of the images for classification is covered in the Experiments section.

Methodology

Persistence Homology over Surface Pattern Image To

study the underlying shape of data, we invoke the machinery of persistent homology (PH), a rapidly emerging research subfield at the interface of data science, machine learning and algebraic topology (Chazal and Michel 2017; Otter et al. 2017; Wasserman 2018). Let \mathbb{X} be the observed data (in our application, \mathbb{X} is a 2D image). \mathbb{X} can also be a graph or point cloud lying in a Euclidean n -dimensional space \mathbb{E}^n or more generally, functional metric space M).

The main goal of PH is to retrieve structural properties about topology and geometry lost during sampling. To extract topological and geometric information in a systematic and efficient manner, we build abstract simplicial complexes, i.e., mathematical objects that are both topological and combinatorial, on top of the observed data \mathbb{X} . As mentioned in Chen et al. (2019), since images are made of pixels, ideally persistent homology on images shall account for representation of images as a pixel grid. As such, using cubical complexes over grid structures, instead of simplicial complexes over point clouds, appears as a more feasible choice to extract topological summaries of images.

Definition 1 Define an elementary cube C as a finite product of elementary intervals, i.e., $Q = I_1 \times I_2 \times \dots \times I_d \subseteq \mathbb{R}^d$, where an elementary interval is a subset $\mathcal{I} \in \mathbb{R}$ such that either $\mathcal{I} = [l, l + 1]$ or $\mathcal{I} = [l, l]$, $l \in \mathbb{Z}_{>0}$. Then, the cubical complex \mathcal{K} in \mathbb{R}^n is a collection of elementary cubes.

Armed with the notion of cubical complexes, we can now track how topological properties of an image evolve as we vary, for example, pixel intensity (of a grayscale image). Such analysis allows us to get a deeper understanding of hidden higher order properties of the image (Edelsbrunner and Harer 2010) which we cannot extract otherwise. Indeed, it is challenging, if not impossible, to distinguish images in the upper panel of Figure 2. However, we find the corresponding topological footprints of these images (see the lower panel of Figure 2) to be quite distinct. Formally, the idea is to consider a filtration of cubical complexes induced by some user-selected function on image pixels (i.e., vertices of \mathcal{K}). In particular, let f be a real-valued function which maps every simplex to the maximum function value of its vertices (e.g., the grayscale value as in our case) and let $\mathcal{K}_r = f^{-1}(-\infty, r]$, $r \in \mathbb{R}$. Then, we can set an increasing sequence of (dis)similarity thresholds r , i.e., $r_1 < r_2 < \dots < r_m$, and construct a nested sequence of cubical complexes $\mathcal{K}_{r_1} \subset \mathcal{K}_{r_2} \subset \dots \subset \mathcal{K}_{r_m}$ associated with evolving thresholds r . Such a nested sequence is called *lower-star filtered cubical complex* (Edelsbrunner and Morozov 2014).

As (dis)similarity threshold r changes, some topological features are born, while others disappear. Topological features which tend to span longer over $r_1 < r_2 < \dots < r_m$ are called *persistent*, while features with shorter lifespans are referred to as *topological noise*. The most popular topological summary under the PH framework is a *persistence diagram* (PD). PD is a multi-set of points in a 2D plane that records the birth-time and the death-time (as x - and y -coordinates, respectively) of each topological feature, e.g., the number of independent components or loops over filtration $\mathcal{K}_{r_1} \subset \mathcal{K}_{r_2} \subset \dots \subset \mathcal{K}_{r_m}$. Figure 2 shows PDs for four different surface pattern images.

Topological Convolutional Networks The architecture

of our proposed Topological Convolutional Networks (TCN) is shown in Figure 3. The key idea is that TCN is able to capture both local topological features and (visual) image information via learning two specific embeddings and combining them. Furthermore, in order to enhance the capability of learning the ring of algebraic information from PDs, we conduct transformation/rotation on target PD and employ the generalized locally periodic (GLP) kernel to kernelize rotated PDs. Besides, TCN utilizes an attention mechanism to adaptively fuse the topology-based embeddings with the learned importance weights. Lastly, we combine the learned embeddings from topological and image spaces for final classification tasks.

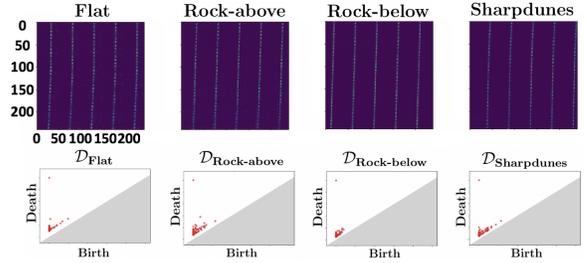


Figure 2: Examples of the image representation of each class and its corresponding topological footprint (i.e., persistence diagram) in the surface pattern dataset.

Topological Meta-Representation Given the input PD \mathcal{D} , we consider multiple transformed variants of \mathcal{D} for topological signature learning instead of only using the original \mathcal{D} , which enables the representation to be invariant to position and orientation of signatures in the PD. As a result, we no longer a-priori need to subjectively define which topological features contain signal and which ones are topological noise. The persistence diagram transformation (PDT) function is designed to extract the hidden prominent topological information from *all* features via the PD rotation. The PDT can be formulated as $\mathfrak{R}_\theta(x_i, y_i) = \{(\cos(\theta)x_i + \sin(\theta)y_i, \cos(\theta)y_i - \sin(\theta)x_i)\}$, where x_i and y_i denote the birth time and death time of the i -th persistence point in $\mathcal{D} = (p_1, \dots, p_i, \dots, p_n) \in \mathbb{R}^{n \times 2}$ (where n is the total number of persistence points), i.e., $p_i = (x_i, y_i) \in \mathbb{R}^2 \in \mathcal{D}$, $\theta \in (0, \pi)$ represents the rotation angle, and the $\mathfrak{R}_\theta(\cdot, \cdot)$ is the PDT function. We use $\mathcal{D}_\theta = (p_1^\theta, \dots, p_i^\theta, \dots, p_n^\theta) \in \mathbb{R}^{n \times 2}$ to denote the rotated PD, where $p_i^\theta = (x_i^\theta, y_i^\theta)$ is the i -th rotated persistence point. Such rotation augmentation also allows us to learn the ring of algebraic functions on PDs and to quantify uncertainty in the topological knowledge representation. To further model the long-range spatial relationships of topological features in the rotated PD, we apply the generalized locally periodic (GLP) kernel $\mathfrak{F}(\cdot, \cdot)$ on the rotated PD \mathcal{D}_θ . Specifically, for i -th rotated persistence point p_i^θ

$$\mathfrak{F}(x_i^\theta, y_i^\theta) = \sigma^2 e^{\left\{ -2 \sin^2 \left(\frac{\pi(x_i^\theta - \alpha_1)^2}{p_1} \right) - \frac{(x_i^\theta - \mu_1)^2}{2l_1^2} \right\}} \times e^{\left\{ -2 \sin^2 \left(\frac{\pi(y_i^\theta - \alpha_2)^2}{p_2} \right) - \frac{(y_i^\theta - \mu_2)^2}{2l_2^2} \right\}},$$

where $p_i, l_i, \mu_i, \alpha_i \in \mathbb{R}$, $i = 1, 2$ are hyperparameters of

GLP kernel and the output of GLP kernel for \mathcal{D}_θ is $\mathfrak{F}(\mathcal{D}_\theta) \in \mathbb{R}^{n \times 1}$.

Suppose we consider m different rotation angles $\Theta_{\mathfrak{R}} = (\theta_1, \dots, \theta_m)$, then we can obtain m kernelized topological signature representations $\{\mathfrak{F}(\mathcal{D}_{\theta_1}), \dots, \mathfrak{F}(\mathcal{D}_{\theta_m})\}$. To extract the topological information encoded in topology space, we utilize m multilayer perceptrons (MLPs) (Rosenblatt 1957) to extract topological signature embedding from a series of kernelized topological signature representations with m different rotation angles as $Z_k = f_{\psi_k}(\mathfrak{F}(\mathcal{D}_{\theta_k}))$, where $\mathfrak{F}(\mathcal{D}_{\theta_k})$ is the k -th kernelized topological signature representation ($k \in [1, m]$), f_{ψ_k} is a neural network with parameter set ψ_k , and the output embedding representation is $Z_k \in \mathbb{R}^{n \times d_c}$. (where d_c is the dimension of kernelized topological signatures.)

Persistence-based Attention Mechanism Now we have m kernelized topological signature embeddings $\{Z_1, \dots, Z_k, \dots, Z_m\}$. Considering the correlation between topological signature representations with different rotation angles, we use the attention mechanism to automatically learn the importance of weights (i.e., $\{\alpha_1, \dots, \alpha_k, \dots, \alpha_m\}$) for different kernelized topological signature embeddings. Inspired by the weighting function for PD (Kusano, Hiraoka, and Fukumizu 2016), we first calculate the persistence-weighted measures for all the persistence points in the PD. The weight $\omega(p_i^{\theta_k})$ represents the importance weight for each rotated persistence point $p_i^{\theta_k}$ in PD \mathcal{D}_{θ_k} , and is defined as $\omega(p_i^{\theta_k}) = \arctan(C(y_i^{\theta_k} - x_i^{\theta_k})^q)$, where \arctan is a bounded, continuous function, and C and q are the hyperparameters.

Hence, we can get the persistence-weighted measure $W(\mathcal{D}_{\theta_k}) = (\omega(p_1^{\theta_k}), \dots, \omega(p_i^{\theta_k}), \dots, \omega(p_n^{\theta_k})) \in \mathbb{R}^{n \times 1}$ for \mathcal{D}_{θ_k} . For the sake of notation, we denote W_k as the persistence-weighted measure of \mathcal{D}_{θ_k} . We next apply a linear transformation to W_k to get the corresponding attention vector $S_k = W_k \cdot \Theta_k$, where $\Theta_k \in \mathbb{R}^{1 \times 1}$ are trainable weights. By performing a dot-product, we can obtain the attention vector $S_k = (s_1, \dots, s_n) \in \mathbb{R}^{n \times 1}$, where s_i is the attention value. Similarly, we can get the attention vectors $\{S_1, \dots, S_{k-1}, S_{k+1}, \dots, S_m\}$ for rotated PDs $\{\mathcal{D}_{\theta_1}, \dots, \mathcal{D}_{\theta_{k-1}}, \mathcal{D}_{\theta_{k+1}}, \dots, \mathcal{D}_{\theta_m}\}$, respectively. Then, we normalize these attention vectors with a SoftMax function to get the final attention vectors $\alpha_k = \text{SoftMax}(S_k) = e^{S_k} / \sum_{k=1}^m e^{S_k} \in \mathbb{R}^{n \times 1}$. Finally, we combine m embeddings with attention vectors to obtain the final embedding Z , i.e., $Z = \alpha_1 \cdot Z_1 + \dots + \alpha_m \cdot Z_m = \sum_{k=1}^m \alpha_k \cdot Z_k$.

Image representation learning To learn the features of an input surface pattern image, we can use any CNN-based model f_{cnn} . Given the input surface pattern image X of resolution $r \times r$, we can obtain the corresponding feature maps $f_{\text{cnn}}(X)$. After that, we employ the MLP f_ϕ to obtain image-level feature representation $Q = f_\phi(f_{\text{cnn}}(X))$.

Topological-based Convolutional Neural Networks Finally, we combine the topological-based embedding representation Z and the image-level feature representation Q to obtain a joint embedding $H = \pi_1 \times Z + \pi_2 \times Q$, where π_1, π_2 are hyperparameters encoding importance of the two factors.

Experiments

We now evaluate performance of our TCN model on the Barefoot surface pattern dataset (Lightholder et al. 2021), a challenging collection of surface pattern terrain images collected onboard a real rover. We first introduce the baselines and parameter settings for all experiments, and then present the quantitative classification results.

The surface pattern image dataset consists of 328 objects $X_{\text{pg}} = \{X_{\text{pg}_1}, \dots, X_{\text{pg}_u}, \dots, X_{\text{pg}_{328}}\}$ and each object $X_{\text{pg}_u} = \{X_{\text{pg}_u}^{(1)}, \dots, X_{\text{pg}_u}^{(\tau)}, \dots, X_{\text{pg}_u}^{(\mathcal{N}_u)}\}$ contains \mathcal{N}_u timestamps, where $u \in [1, 328]$, $\tau \in [1, \mathcal{N}_u]$, and $\mathcal{N}_u \in [199, 860]$. For the object X_{pg_u} , each timestamp $X_{\text{pg}_u}^{(\tau)}$ consists of 1,920 pixel values. Figure 4(a) shows a sample object in the surface pattern image dataset. The shapes of the object u and each timestamp observation $X_{\text{pg}_u}^{(\tau)}$ are $(\mathcal{N}_u, 1920)$ and $(1, 1920)$, respectively. Since each object (e.g., X_{pg_u}) has its own class c_ℓ , each observation $X_{\text{pg}_u}^{(\tau)}$ in the object at timestamp τ has the same class c_ℓ . To incorporate more inputs into the model, each object is aggregated into 30-unit windows. We can obtain $\lfloor \mathcal{N}_u / 30 \rfloor$ observations per object with class c_ℓ . Overall, we generate $\sum_{u=1}^{328} \lfloor \mathcal{N}_u / 30 \rfloor$ observations from the original 328 objects, where the shape of each observation X_i is $(240, 240)$ (i.e., reshaped from $(30, 1920)$ to $(240, 240)$); see Figure 4 (b). To reduce the computational complexity and memory cost, we convert the higher resolution image to a lower resolution of $(60, 60)$ and, hence, the shape of the final image fed into the model is $\tilde{X}_i \in \mathbb{R}^{60 \times 60}$ (see the bounding box in the green rectangle in Figure 4(c)). Moreover, here we consider two multi-label classification tasks: (i) rock-flat classification (i.e., covering three classes $\{0 : \text{“flat”}; 1 : \text{“rock-above”}; 2 : \text{“rock-below”}\}$) and (ii) pattern classification (i.e., covering eight classes $\{0 : \text{“bedrock”}; 1 : \text{“flat”}; 2 : \text{“gullies”}; 3 : \text{“pebbles”}; 4 : \text{“rock-above”}; 5 : \text{“rock-below”}; 6 : \text{“sharpdunes”}; 7 : \text{“smoodunes”}\}$). Thus, rock-flat and pattern datasets contain 3,297 and 5,754 observations, respectively.

Baselines and Implementation Details We use the following four methods as baselines: (i) MLP (Rosenblatt 1957); (ii) LeNet (LeCun et al. 2015); (iii) AlexNet (Krizhevsky, Sutskever, and Hinton 2012); and (iv) TOPO-GCN (Chen, Marchetti, and Gel 2021). We implement our TCN with Pytorch framework on a single NVIDIA GeForce RTX 3090 GPU. Further, for the dataset, TCN is trained by the Adadelta optimizer with an L1 loss function. For our model, TCN consists of 3 layers whose hidden dimensions are 1, 32, 64 respectively. The learning rate is 0.95 and the batch size is set as 32. In addition, the dropout rate is 0.1. We split the surface pattern image dataset into training and testing sets and the split ratio is 8:2. We use accuracy to evaluate performances of models. We run 5 times with the same partition and report the average results with standard deviations. The data and codes are available at <https://github.com/TopoCN/TCN.git>.

Surface Pattern Classification Table 1 compares TCN against four baselines on surface pattern image datasets with different numbers of classes (rock-flat detection and pattern recognition). TCN achieves the top performance on both rock-flat detection and pattern recognition, and outperforms

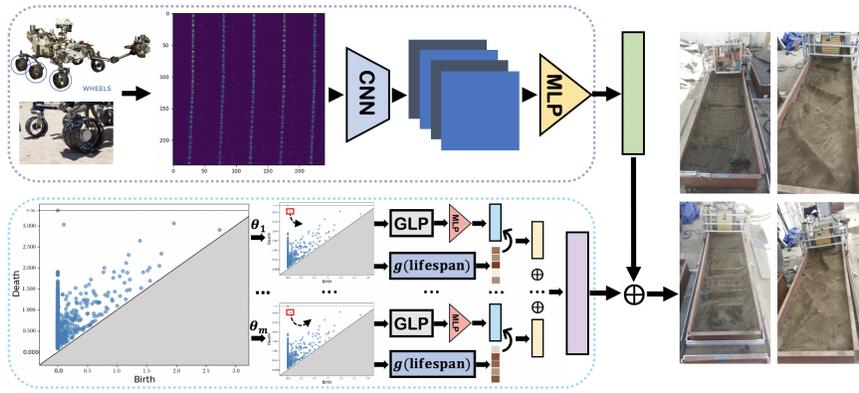


Figure 3: Overview of our TCN framework. The upper part is the diagram of image representation learning. The lower part is the diagram of topological representation learning.

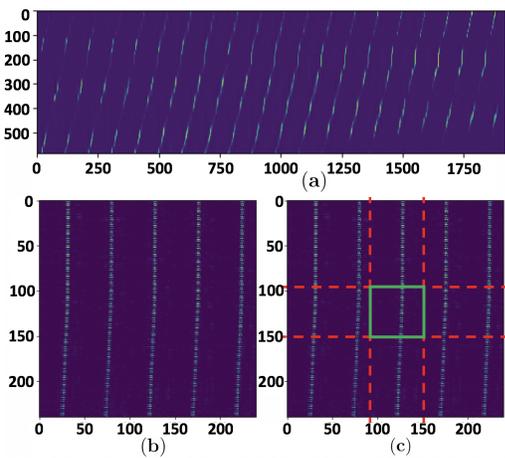


Figure 4: Examples of (a) one pattern object in surface pattern image dataset, (b) the observation $X_i \in \mathbb{R}^{240 \times 240}$ in above pattern object, and (c) its corresponding low-resolution representation $\tilde{X}_i \in \mathbb{R}^{60 \times 60}$.

the runner-ups (LeNet and TOPO-GCN) by 1.25% for rock-flat detection and 4.87% for pattern recognition. Moreover, Figures 5 and 6 present confusion matrices of TCN and top three competitors separately for each landform class. As Figure 5 shows, results of TCN are consistently better than the runner-up across all three classes (flat, rock-above, and rock-below), indicating the advantage of using topological information on rock types of landforms. Figure 6 suggests that TCN outperforms the runner-up over bedrock, flat, gullies, pebbles, rock-above, and smoodunes classes. Specifically, TCN yields relative gains of 31.23%, 13.87%, and 7.28% compared to the runner-up across pebbles, rock-above, and smoodunes respectively. On rock-below and sharpdunes, the runner-up outperforms TCN. For rock-below, this might be due to the limited ability of CNN to capture below surface information. However, we currently cannot find interpretation for results on sharpdunes. Overall, the results of our TCN model across different tasks consistently demonstrate that the combination of information from image and topological spaces exhibit better performance than image-based

neural networks.

Method	Rock-Flat Detection	Pattern Recognition
MLP	62.50 ± 1.75	44.05 ± 1.13
LeNet	72.10 ± 1.59	46.67 ± 1.96
AlexNet	71.62 ± 2.63	56.25 ± 1.78
TOPO-GCN	71.17 ± 1.89	56.68 ± 2.00
TCN (ours)	73.01 ± 1.35	59.58 ± 1.15

Table 1: Classification accuracy (%) with standard deviations. Best results are in **bold**.

		TCN (ours)			Runner-up (LeNet)		
True	(I)	80.84%	9.42%	9.74%	79.17%	10.90%	9.94%
	(II)	12.50%	64.47%	23.03%	15.97%	63.19%	20.83%
	(III)	13.41%	14.53%	72.07%	14.67%	16.85%	68.48%
		Predicted (a)			Predicted (b)		
		AlexNet			TOPO-GCN		
True	(I)	76.70%	9.39%	13.92%	80.14%	6.97%	12.89%
	(II)	15.89%	62.91%	21.19%	15.51%	58.29%	26.20%
	(III)	8.89%	21.67%	69.44%	10.99%	18.68%	70.33%
		Predicted (c)			Predicted (d)		

Figure 5: Confusion matrices of (a) TCN (accuracy: 74.49%), (b) runner-up (LeNet) (accuracy: 72.50%), (c) AlexNet (accuracy: 71.41%), and (d) TOPO-GCN (accuracy: 71/19%) for rock-flat detection task. Labels: (I) Flat, (II) Rock-above, and (III) Rock-below.

Computational Costs Table 2 shows the running times, i.e., per-epoch training times of our TCN model and baselines across both rock-flat detection and pattern recognition datasets. Note that in the topological representation learning of our TCN model, we directly use PD to learn the local topological information; as an alternative, although TOPO-GCN attains comparable results, it utilizes persistence image

TCN (ours)

True	(I)	-46.97%	6.06%	4.55%	0.00%	19.70%	7.58%	4.55%	10.61%
	(II)	-0.00%	80.34%	3.10%	5.17%	2.76%	3.79%	0.00%	4.83%
	(III)	-7.27%	40.00%	25.45%	0.00%	12.73%	5.45%	1.82%	7.27%
	(IV)	-0.00%	9.64%	0.00%	66.27%	1.20%	15.66%	3.61%	3.61%
	(V)	-2.82%	9.04%	3.39%	2.26%	61.58%	10.73%	3.39%	6.78%
	(VI)	-4.52%	10.17%	0.56%	11.86%	17.51%	44.63%	2.82%	7.91%
	(VII)	-2.13%	3.19%	2.13%	1.06%	7.45%	4.26%	46.81%	32.98%
	(VIII)	-3.37%	10.11%	4.49%	2.81%	6.74%	2.25%	9.55%	60.67%
		Predicted							

Figure 6: Confusion matrices of TCN (accuracy: 60.09%) for pattern recognition task. Labels: (I) Bedrock, (II) Flat, (III) Gullies, (IV) Pebbles, (V) Rock-above, (VI) Rock-below, (VII) Sharpdunes, and (VIII) Smoodunes.

(PI) via vectorizing PD, which suffers high computational time and memory cost. Similarly, compared with AlexNet (CNN architecture with many parameters), our TCN is more efficient and faster. Besides, although LeNet has a faster training time, it performs significantly worse on the pattern recognition task. Note that the input image of our TCN model only involves 6.25% information, TCN can still exhibit superior accuracy on rock-flat detection and pattern recognition, and this means that TCN still has much room for the improvement of multi-label classification accuracy. Finally, we can conclude that TCN is simple to implement and provides significant gains in run-time and memory efficiency.

Method	Rock-Flat Detection	Pattern Recognition
MLP	0.49 s	0.76 s
LeNet	1.80 s	2.65 s
AlexNet	8.19 s	14.01 s
TOPO-GCN	9.15 s	15.99 s
TCN (ours)	4.39 s	8.29 s

Table 2: Running time (training time per epoch).

Lessons Learned

The current findings on surface pattern classification with DL provide a variety of lessons for further integration of AI to onboard learning and exploratory missions:

- DL tools, such as CNNs and GCNs, show promising results for learning complex planetary terrain classes. However, further enhancement of such DL models with topological footprints can boost not only model accuracy, but also reduce variability and computational costs.
- Local topological information of surface patterns is an important signal for terrain classification and could be useful in not only fetching rich information about image topology, but also in discovering higher-order connectivity patterns in the collected heterogeneous data and matching them to previous records. As such, comparative analysis of these topological footprints can help to better understand

geologic history of the planet, including higher order interactions among its geophysical properties and potential for life. In turn, the proposed topological attention mechanism might be particularly valuable for accurate classification of objects, collected in heterogeneous sensing scenarios, including nonrigid shapes.

- Although TCN yields a promising multi-label classification accuracy (especially in rock depth detection), it shows limited ability to detect gullies. The reason may be due to incorrect estimation of topological properties of a gully (i.e., V-shape with sloping heads and sides). To remedy this issue, we may need to consider not one filtered cubical complex, but filtrations along multiple geometric dimensions.
- Compared to rock-flat detection, we observe a decline in performance for pattern recognition, which may be due to insufficient representation of certain types of patterns, e.g., bedrock and gullies. This issue can be tackled by further topological data augmentation and topological subsampling within the meta-representation mechanism.

Path to Deployment

Development of DL tools for onboard applications in earth and spaces sciences is one of the primary interests for NASA (see, for example, the most recent August 2021 call for early-stage concepts from NASA’s Advanced Information Systems Technology Program (NASA 2021)). Indeed, the DL methodology enables for accurate modeling of sophisticated nonlinear spatial and spatio-temporal patterns, including, but not limited to terrain classification and exploration, and exhibits better performance compared to hand-crafted methods. As a result, DL has a higher potential for feature detection and tracking in the data gathered from heterogeneous sensing. In turn, topological descriptors of planetary surfaces can bring an invaluable insight on local terrain characteristics and their similarities at multiple resolutions, which are otherwise inaccessible with standard Euclidean-based approaches. Equipping DL models with topological footprints of terrain has the potential to improve model generalizability and transferrability, which are of particular importance in onboard applications.

Finally, the application of DL and topological methods in onboard exploration tasks is currently largely obstructed by prohibitive computational costs. Our project presents an early-concept approach which aims to make a step in bridging the power of DL with onboard exploration of planetary terrain, by substantially reducing computational and storage costs, while maintaining the high classification accuracy. In particular, compared to Marchetti et al. (2020), in our model, we use the low resolution surface pattern image for image representation learning and representation learning based on local topological information, i.e., using only 6.25% image information for multi-label classification tasks. Another benefit of TCN is that the topological signatures can be computed offline from images, which substantially reduces online computation time necessary. It is worth mentioning that TCN requires a much lower number of parameters compared to GCNs (i.e., TOPO-GCN) (Chen, Marchetti, and Gel 2021), e.g., TOPO-GCN (with #4,293,959) has 40 times more than TCN (with #104,468 parameters). More-

over, computational complexity of graph convolutions is a major limiting factor, with complexity of $O(\mathcal{N}^3)$, where \mathcal{N} denotes number of nodes in the graph.

TCN is not limited to analyzing terrain characteristics on-line, and can also readily be extended to images of terrain from the orbit. Images from current and future missions, e.g., Mars Reconnaissance Orbiter or Europa Clipper, would be an interesting application of TCN for better accuracy of identification of topological features on surfaces from remote sensing. Thus, deployment can also be extended to analysis of terrain images on the ground.

Conclusion and Future Work

Multi-class classification of surface patterns is a very challenging problem, compared, e.g., to a simpler task of binary classification of rock. Good classification performance is challenging to achieve with conventional ML techniques due to the sophisticated structure of landforms. Thus, advanced DL tools such as TCN are needed to handle more complex terrain patterns exhibited in planetary exploration missions. Our work underscores the importance of DL with topological footprints such as TCN for terrain recognition, especially given that the diversity and complexity of the terrain patterns is much more present in real life situations. The ability to adaptively re-train onboard classifiers given new information from a sensor would also be critical since new classes need to be discovered and existing classes need to be updated. In the future, we will extend TCN with capabilities to re-train on demand, hence, making its integration with onboard and embedded systems more competitive. We will also explore applicability of TCN to other types of data, such as 3D modeling or video analysis, which also have prohibitively high computational costs.

Acknowledgements

This work is supported by NSF grants ECCS 2039701, INTERN supplement for ECCS 1824716, and DMS 1925346.

References

Chazal, F.; and Michel, B. 2017. An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *arXiv:1710.04019*.

Chen, Y.; Marchetti, Y.; and Gel, Y. R. 2021. Deepening the Sense of Touch in Planetary Exploration with Geometric and Topological Deep Learning. In *IAAI*, volume 35, 15278–15285.

Chen, Y.; Rastogi, C.; and Norris, W. R. 2021. A CNN Based Vision-Proprioception Fusion Method for Robust UGV Terrain Classification. *Robotics and Automation Letters*, 6: 7965–7972.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019. Multi-label image recognition with graph convolutional networks. In *CVPR*, 5177–5186.

Edelsbrunner, H.; and Harer, J. 2010. *Computational topology: an introduction*. American Mathematical Soc.

Edelsbrunner, H.; and Morozov, D. 2014. Persistent Homology: Theory and Practice. In *European Congress of Mathematics*, 31–50.

Hofer, C. D.; Kwitt, R.; and Niethammer, M. 2019. Learning Representations of Persistence Barcodes. *JMLR*, 20: 1–45.

Kerner, H. R.; Wagstaff, K. L.; Bue, B. D.; Wellington, D. F.; Jacob, S.; Horton, P.; Bell, J. F.; Kwan, C.; and Amor, H. B. 2020. Comparison of novelty detection methods for multispectral images in rover-based planetary exploration missions. *DMKD*, 34: 1642–1675.

Kerner, H. R.; Wellington, D. F.; Wagstaff, K. L.; Bell, J. F.; Kwan, C.; and Amor, H. B. 2019. Novelty detection for multispectral images with application to planetary exploration. In *IAAI*, volume 33, 9484–9491.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, volume 25, 1097–1105.

Kusano, G.; Hiraoka, Y.; and Fukumizu, K. 2016. Persistence weighted Gaussian kernel for topological data analysis. In *ICML*, 2004–2013.

LeCun, Y.; et al. 2015. LeNet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, 20(5): 14.

Lightholder, J.; Marchetti, Y.; Mandrake, L.; Junkins, E.; Ma, R.; Schibler, T.; Springer, P.; Cross, M.; Tavallali, P.; Yates, D.; Young, J.; Kennedy, B.; Moreland, S.; Thoesen, A.; Green, M.; Martia, J.; Chang, J.; Villalobos, C.; Salah, M.; McBryan, T.; Mick, D.; Horton, P.; Doerksen, K.; Pascual, A.; Southwell, B.; and Bao, R. 2021. Barefoot Rover MMINTR Material Data. Zenodo.

Marchetti, Y.; Lightholder, J.; Junkins, E.; Cross, M.; Mandrake, L.; and Fraeman, A. 2020. Barefoot rover: a sensor-embedded rover wheel demonstrating in-situ engineering and science extractions using machine learning. In *ICRA*, 6000–6006.

NASA. 2021. Advanced Information Systems Technology. <https://esto.nasa.gov/aist/>.

Otter, N.; Porter, M. A.; Tillmann, U.; Grindrod, P.; and Harrington, H. A. 2017. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6: 17.

Rosenblatt, F. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Rothrock, B.; Kennedy, R.; Cunningham, C.; Papon, J.; Heverly, M.; and Ono, M. 2016. Spoc: Deep learning-based terrain classification for mars rover missions. In *AIAA SPACE*, 5539.

Schmidt, J. R.; and Cheein, F. A. 2019. Prognosis of the energy and instantaneous power consumption in electric vehicles enhanced by visual terrain classification. *Computers & Electrical Engineering*, 78: 120–131.

Wagstaff, K. L.; Lu, Y.; Stanboli, A.; Grimes, K.; Gowda, T.; and Padams, J. 2018. Deep Mars: CNN classification of mars imagery for the PDS imaging atlas. In *IAAI*.

Wasserman, L. 2018. Topological Data Analysis. *Annual Review of Statistics and Its Application*, 5: 1–42.

Zürn, J.; Burgard, W.; and Valada, A. 2020. Self-supervised visual terrain classification from unsupervised acoustic feature learning. *Transactions on Robotics*, 37(2): 466–481.