A Primer on Topological Data Analysis to Support Image Analysis Tasks in Environmental Science

LANDER VER HOEF,[a] HENRY ADAMS,[a] EMILY J. KING,[a] AND IMME EBERT-UPHOFF[b,c]

[a] Department of Mathematics, Colorado State University, Fort Collins, Colorado
[b] Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado
[c] Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado

(Manuscript received 20 May 2022, in final form 15 November 2022)

ABSTRACT: Topological data analysis (TDA) is a tool from data science and mathematics that is beginning to make waves in environmental science. In this work, we seek to provide an intuitive and understandable introduction to a tool from TDA that is particularly useful for the analysis of imagery, namely, persistent homology. We briefly discuss the theoretical background but focus primarily on understanding the output of this tool and discussing what information it can glean. To this end, we frame our discussion around a guiding example of classifying satellite images from the sugar, fish, flower, and gravel dataset produced for the study of mesoscale organization of clouds by Rasp et al. We demonstrate how persistent homology and its vectorization, persistence landscapes, can be used in a workflow with a simple machine learning algorithm to obtain good results, and we explore in detail how we can explain this behavior in terms of image-level features. One of the core strengths of persistent homology is how interpretable it can be, so throughout this paper we discuss not just the patterns we find but why those results are to be expected given what we know about the theory of persistent homology. Our goal is that readers of this paper will leave with a better understanding of TDA and persistent homology, will be able to identify problems and datasets of their own for which persistent homology could be helpful, and will gain an understanding of the results they obtain from applying the included GitHub example code.

SIGNIFICANCE STATEMENT: Information such as the geometric structure and texture of image data can greatly support the inference of the physical state of an observed Earth system, for example, in remote sensing to determine whether wildfires are active or to identify local climate zones. Persistent homology is a branch of topological data analysis that allows one to extract such information in an interpretable way—unlike black-box methods like deep neural networks. The purpose of this paper is to explain in an intuitive manner what persistent homology is and how researchers in environmental science can use it to create interpretable models. We demonstrate the approach to identify certain cloud patterns from satellite imagery and find that the resulting model is indeed interpretable.

KEYWORDS: Remote sensing; Satellite observations; Artificial intelligence; Classification; Model interpretation and visualization; Support vector machines

1. Introduction

Methods for image analysis have become an essential tool for many environmental science (ES) applications to automatically extract key information from satellite imagery or from gridded model output (Schultz et al. 2021; Zhu et al. 2017; Gagne et al. 2019; Ebert-Uphoff and Hilburn 2020). Machine learning (ML) methods such as convolutional neural networks (CNNs) are now the dominant technique for many such tasks, where they operate as black boxes (McGovern et al. 2019). This is undesirable for high stakes applications (Rudin 2019; McGovern et al. 2022). In this paper, we show how a tool that is beginning to be used in the community, namely, topological data analysis (TDA), can be combined with ML methods for interpretable image analysis. TDA is a mathematical discipline that can quantify geometric information from an image in a predictable and well-understood way. In section 4d, we give a novel example of how we can leverage this understanding to give a strong interpretation of ML results in terms of image features.

TDA has proven highly successful to aid in the analysis of data in a variety of applications, including neuroscience (Chung et al. 2009; Gardner et al. 2022), fluid dynamics (Kramár et al. 2016), and cancer histology (Lawson et al. 2019). In environmental science, TDA has recently shown potential to help identify atmospheric rivers (Muszynski et al. 2019), detect solar flares (Deshmukh et al. 2022; Sun et al. 2021), identify which wildfires are active (Kim and Vogel 2019), quantify the diurnal cycle in hurricanes (Tymochko et al. 2020), identify local climate zones (Sena et al. 2021), detect and visualize Rossby waves (Merritt 2021), and forecast COVID-19 spread using atmospheric data (Segovia-Dominguez et al. 2021). The purpose of this article is to provide an intuitive introduction to TDA for the environmental science community—using a meteorological application as a guiding example—and an understanding of where TDA might be applied. This article is accompanied by easy-to-follow sample code provided as a GitHub repository (https://github.com/zyjux/sffg_tda) that we hope will be used by the community in new applications.

Corresponding author: Lander Ver Hoef, lander.verhoef@gmail.com

DOI: 10.1175/AIES-D-22-0039.1 e220039

a. *Guiding application: Analyzing the mesoscale organization of clouds*

To provide a gentle introduction to TDA for the ES community, we illustrate its use for a practical example. We chose the application of classifying the mesoscale organization of clouds, specifically distinguishing four types of organization—sugar, gravel, fish, and flowers—identified by Stevens et al. (2020). This task provides an ideal case study for our exploration of topological data analysis for several reasons: 1) these four organization patterns are well known from the seminal paper by Stevens et al. (2020), and meteorological experts were able to reliably identify these patterns from satellite-visible imagery. 2) The task can be formulated as a classification of patches of single-image monochromatic imagery, for which a common TDA algorithm (persistent homology) is well suited. 3) TDA has never been applied for this application, so it is novel. 4) A well-developed benchmark dataset with reliable crowdsourced labels is publicly available for this task (Rasp et al. 2020).

Several ML approaches have already been developed with good success for this benchmark dataset to classify the four different types (Rasp et al. 2020). We emphasize that we are *not* seeking to match or exceed the performance of those ML approaches. Rather, we use this application to demonstrate TDA as an approach that can increase transparency, decrease computational effort, and be feasible even if few labeled data samples are available (see section 1c).

b. *Key TDA concepts discussed here*

In this paper, we focus on the TDA concept that is most appropriate for image analysis: persistent homology. We will provide a detailed introduction in section 3 but in this subsection give a short preview of key concepts to be discussed.

Homology is the classical study of connectivity and the presence of holes of various dimensions, giving large-scale geometric information. Persistent homology provides a descriptor with information on the texture of an image (how rough or smooth it is), which can be vectorized into a format useful for machine learning. It does this by scaling through all the intensity values in an image and recording at what intensities connected components and holes appear and disappear. Particularly on images, persistent homology and its vectorizations can be efficiently computed, so for image analysis (from models or satellites) the computational effort of implementing persistent homology is small.

The results of persistent homology computations can be displayed as either persistence diagrams or persistence barcodes. We focus here on barcodes in which each feature (connected component or hole) appears as a bar that starts at the intensity value at which the feature appears and ends at the intensity at which it disappears. The lengths of these bars indicate the persistence of each feature. The raw output of persistent homology is not suitable for most machine learning tasks as the output vector varies in length from sample to sample. While there are many proposed solutions to this, in this paper we use persistence landscapes, which translate a barcode into a mountain range, with the height of each mountain representing the persistence of the corresponding feature. The landscape is obtained from this mountain range by taking the n highest profiles as piecewise-linear functions, where n is a hyperparameter.

c. *Advantages of TDA for image analysis tasks in environmental science*

Persistent homology is a deterministic mathematical transformation (just as, say, the well-known Fourier transform). In the following, we first explore the advantages that persistent homology inherits from being a deterministic algorithm:

1) Transparency: All the internal steps of the algorithm are known and well-understood, and the method has a high degree of theoretical interpretation, giving it far more transparency than most ML methods. In section 4d, we use this theoretical background to understand what image features are driving differences in the output of persistent homology.
2) Known failure modes: No technique is perfect, and there will always be situations that cause errors and incorrect results. To use a method in practice, it is important to understand in what situations it struggles and what sorts of errors can result. Because persistent homology is a deterministic method, we can both theoretically predict these failure modes and interpret experimental results in terms of the original feature space.
3) No need for large, labeled datasets: As a deterministic algorithm, persistent homology does not require large, reliably labeled datasets. Instead, a small set of representative examples can be used to explore the different patterns that emerge in the transformed data. TDA is often used in combination with a simple ML model, and the number of labeled samples to obtain good performance is smaller than would be required to train a CNN or similar tool without TDA. This is a huge advantage for environmental science datasets, which are frequently large and detailed but almost entirely unlabeled.
4) Environmentally friendly: Many CNNs for image analysis tasks are known to have a surprisingly high carbon footprint due to the extensive computational resources required for model training (Schwartz et al. 2020; Xu et al. 2021). TDA is more in line with the Green AI movement (Schwartz et al. 2020; Xu et al. 2021), as it enables context-driven numerical results without the environmental impact inherent in training a deep neural network.

Next, we discuss the key abilities that persistent homology brings to image analysis tasks. These fall into three general categories: the incorporation of spatial context into a deterministic algorithm, the detection of texture and contrast, and invariance under certain transformations. The categories are discussed further below:

1) Incorporating spatial context: Many deterministic algorithms, as well as fully connected neural networks, struggle to incorporate the spatial context inherent in satellite data. Integrating this spatial context is precisely what motivated the development of CNNs, but CNNs are costly to train and challenging to make explainable. Persistent homology naturally incorporates spatial context, so patterns that are evident in this spatial context can be incorporated without resorting to

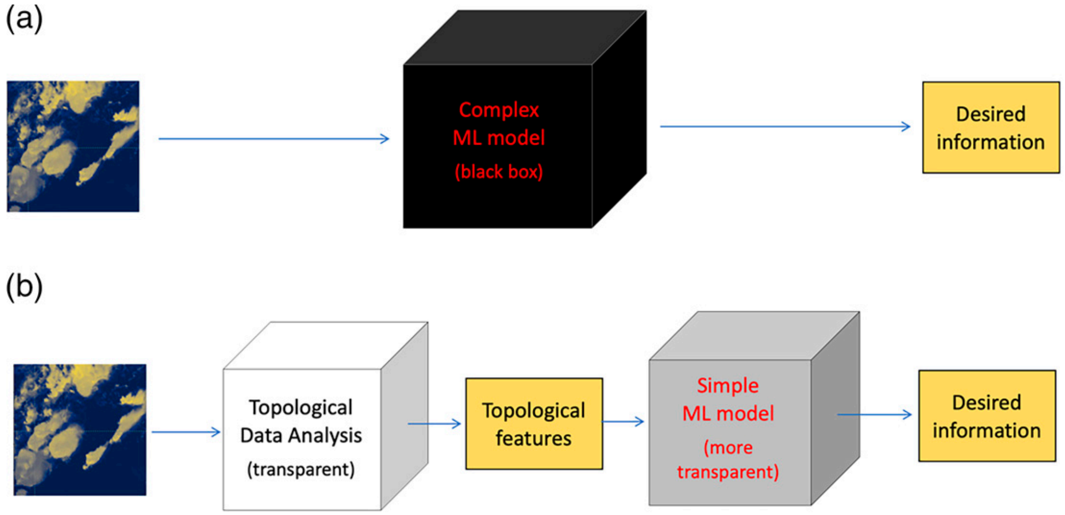

FIG. 1. Two different ways to extract desired information from imagery: (a) the pure ML approach, in which image information is extracted by using a complex ML model, typically a deep neural network, and (b) the TDA approach, in which image information is extracted by using TDA followed by a simpler ML model/method. The latter can lead to more transparent and computationally efficient approaches.

CNNs or other spatially informed neural network architectures.

2) Detection of texture and contrast: Persistent homology excels at detecting contrast differences—regions (small or large) that differ from the surrounding average, which gives a representation of the texture present in an image. This focus on texture is useful in analyzing satellite weather imagery, as texture is frequently a key distinguishing factor, even more than a cloud being a particular shape or size.

3) Invariance to homeomorphisms: The notion of not wanting to be constrained by a particular geometry brings us to the final advantage: invariance under a common class of transformations called homeomorphisms (see section 3e).

d. *Combining TDA with simple ML algorithms*

For some image analysis tasks, TDA methods can be used as a stand-alone tool, but for the majority of tasks, one would first use TDA to extract topological features and then afterward add a simple machine learning algorithm, as shown in Fig. 1b. For example, the sample application in section 4 uses TDA followed by a support vector machine (SVM). TDA can thus be viewed as a transparent means to construct new, physically meaningful, and interpretable features that may reduce the need for black-box machine learning algorithms. Using TDA in this way can support the goals of creating ethical, responsible, and trustworthy artificial intelligence approaches for environmental science outlined in McGovern et al. (2022), since transparency is a key requirement for ML approaches to be used in tasks that affect life-and-death decision-making (Rudin 2019), such as severe weather forecasting.

e. *Objectives and organization of this article*

As mentioned before, we are not attempting to set a new benchmark for accuracy in classification nor are we declaring that this method renders existing techniques obsolete. Instead, we seek to raise awareness of a promising technique with significant potential for ES applications and provide the reader with a high-level understanding of how TDA works, what sorts of questions can be asked using TDA, and how the answers obtained can be interpreted and understood. The case study in section 4 provides examples of the sorts of questions TDA can help to address, including reports of negative examples, that is, situations in which persistent homology is *not* able to distinguish between classes, which are as informative as positive examples in order to understand the best use of TDA.

The remainder of this article is organized as follows: section 2 discusses in detail the sample application of classifying the mesoscale organization of clouds. Section 3 provides an introduction to the key concepts of topological data analysis. Section 4 illustrates the use of these TDA concepts for the sample application from section 2 in combination with a simple support vector machine. In particular, in section 4d, we provide a detailed and novel discussion of the characteristic image-level features that our combined TDA–SVM algorithm uses to classify. This highlights the ability to identify which learned patterns can be exposed and to discuss these in the original feature space, which is one of the greatest strengths of persistent homology and TDA. Section 5 provides an overview of advanced TDA concepts that are beyond the scope of this paper. Section 6 provides conclusions and suggests future work.

2. Guiding application: Classifying the mesoscale organization of clouds from satellite data

To illustrate the use of TDA we consider the task of identifying patterns of mesoscale (20–1000 km) organization of shallow clouds from satellite imagery, which has recently attracted much attention (Stevens et al. 2020; Rasp et al. 2020; Denby 2020). Climate models, because of their low spatial resolution, cannot model clouds at their natural scale (Gentine et al. 2018; Rasp et al. 2018). Since clouds play a major role in the radiation budget of Earth (L'Ecuyer et al. 2015), the limited representation of clouds in climate models causes significant uncertainty for climate prediction (Gentine et al. 2018). There has been progress in addressing this limitation from the climate modeling side, for example, using ML to better represent subgrid processes (Krasnopolsky et al. 2005; Rasp et al. 2018; Yuval and O'Gorman 2020; Brenowitz et al. 2020).

A different approach is to build a better understanding of cloud organization in satellite imagery (Stevens et al. 2020; Rasp et al. 2020; Denby 2020). One goal is to track the frequency of occurrence of certain cloud patterns across the globe, reaching back in time as far as satellite imagery allows, to better understand changes to the underlying meteorological conditions. To this end, in 2020 a group of scientists from an International Space Science Institute (ISSI) international team identified the primary types of mesoscale cloud patterns seen in Moderate Resolution Imaging Spectroradiometer (MODIS; Gumley et al. 2010) true color satellite imagery, focusing on boreal winter (December–February) over a trade wind region east of Barbados (Stevens et al. 2020). Using visual inspection they identified four primary mesoscale cloud patterns, namely, sugar, gravel, fish, and flowers (shown in Fig. 2).Subsequent study of these four cloud types using radar imagery (Stevens et al. 2020) and median vertical profiles of temperature, relative humidity, and vertical velocity (Rasp et al. 2020) indicate that the four cloud types occur in climatologically distinct environments and are thus a good indication of those environments.

While humans are fairly consistent at recognizing these four patterns after some training, it is difficult to describe them objectively so that a machine can be programmed to do the same. Deep learning offers a potential solution; however, most deep learning approaches require a large number of labeled images to learn from.

a. Approaches for dealing with lack of labeled samples

Rasp et al. (2020) solve the lack of labeled data for this application by a crowdsourcing campaign using a two-step process. First, they developed a crowdsourcing environment and recruited experts to label 10 000 images. Experts used a simple interface to mark rectangular boxes in the imagery and label them with one of the four patterns. The labeled dataset enabled the use of supervised learning algorithms as a second step, that is, the algorithms were supplied with pairs of input images and output labels and then trained to estimate output labels from given imagery. Two types of supervised deep learning algorithms were developed: one for object recognition and one for segmentation. Both algorithms performed well (Rasp et al. 2020).

In contrast, unsupervised learning approaches seek to develop models from unlabeled data samples. Clustering—which divides unlabeled input samples into groups that are similar in some way—is a classic unsupervised learning algorithm. For example, Denby (2020) trained an unsupervised deep learning algorithm, in combination with a hierarchical clustering algorithm, for a closely related application, namely, grouping image patches from Geostationary Operational Environmental Satellite (Schmit et al. 2017) imagery into clusters of similar cloud patterns. Their algorithm identified a hierarchy of clustered mesoscale cloud patterns, but since the classes of cloud patterns were generated by a black-box algorithm rather than by domain scientists, their meaning is less understood than the four patterns from Rasp et al. (2020). Indeed, a necessary step that comes after the unsupervised learning is to test whether the patterns identified by an algorithm correspond to climatologically distinct environments and if so which ones.

TDA is an alternative approach to address the lack of labels. With TDA we seek to match imagery to the original four classes identified by Stevens et al. (2020), yet only require a small number of labeled samples. We map patches of the MODIS imagery into topological space and then investigate whether there are significant differences in the topological properties that we can leverage to distinguish the patterns. TDA can thus be viewed as a means of sophisticated feature engineering, giving new, physically meaningful topological features. Our motivation is that this approach would allow us to identify the well-established patterns from Stevens et al. (2020) but with two key differences: 1) this approach does not require a large number of labels (less crowdsourcing required) and 2) this approach is more transparent than the supervised [such as Rasp et al. (2020)] and unsupervised [such as Denby (2020)] deep learning approaches, since topological properties can be understood intuitively.

We note that TDA can also be used in an unsupervised fashion similar to the approach of Denby (2020), only with more transparency and computational efficiency. On its own, TDA provides an embedding of the image data. However, rather than the embeddings being a learned property of a neural network whose properties can only be inferred after it is trained, and then only with difficulty, the TDA embedding is deterministically based on topological properties of the image. For this primer, however, we focus on the supervised task of identifying the previously established patterns of Stevens et al. (2020).

b. Dataset details and preprocessing

The dataset from Rasp et al. (2020) provides approximately 50 000 individual cloud-type "annotations" (where each annotation is a rectangle placed on an image surrounding a particular cloud type) on around 10 000 base images. To evaluate the quality of these crowdsourced annotations, Rasp et al. (2020) used a comparison of intersection-over-union (IOU) scores (also known as the Jaccard index; Jaccard 1901; Fletcher and Islam 2018) between annotators analyzing the same image, and their analysis indicated that these annotations were generally of high quality. See Fig. 2 for examples of these cloud types. In general, sugar-type clouds are small, relatively uniformly distributed clouds; gravel-type clouds are somewhat larger than sugar clouds and tend to show more organization; flowers-type clouds are yet larger clouds that clump together with areas of clear sky between; and fish-type clouds form distinctive mesoscale skeletal patterns. Each image in the dataset is a 14° × 21° (latitude–longitude) visible color MODIS image from the *Terra* or *Aqua* satellite. On these images, annotators

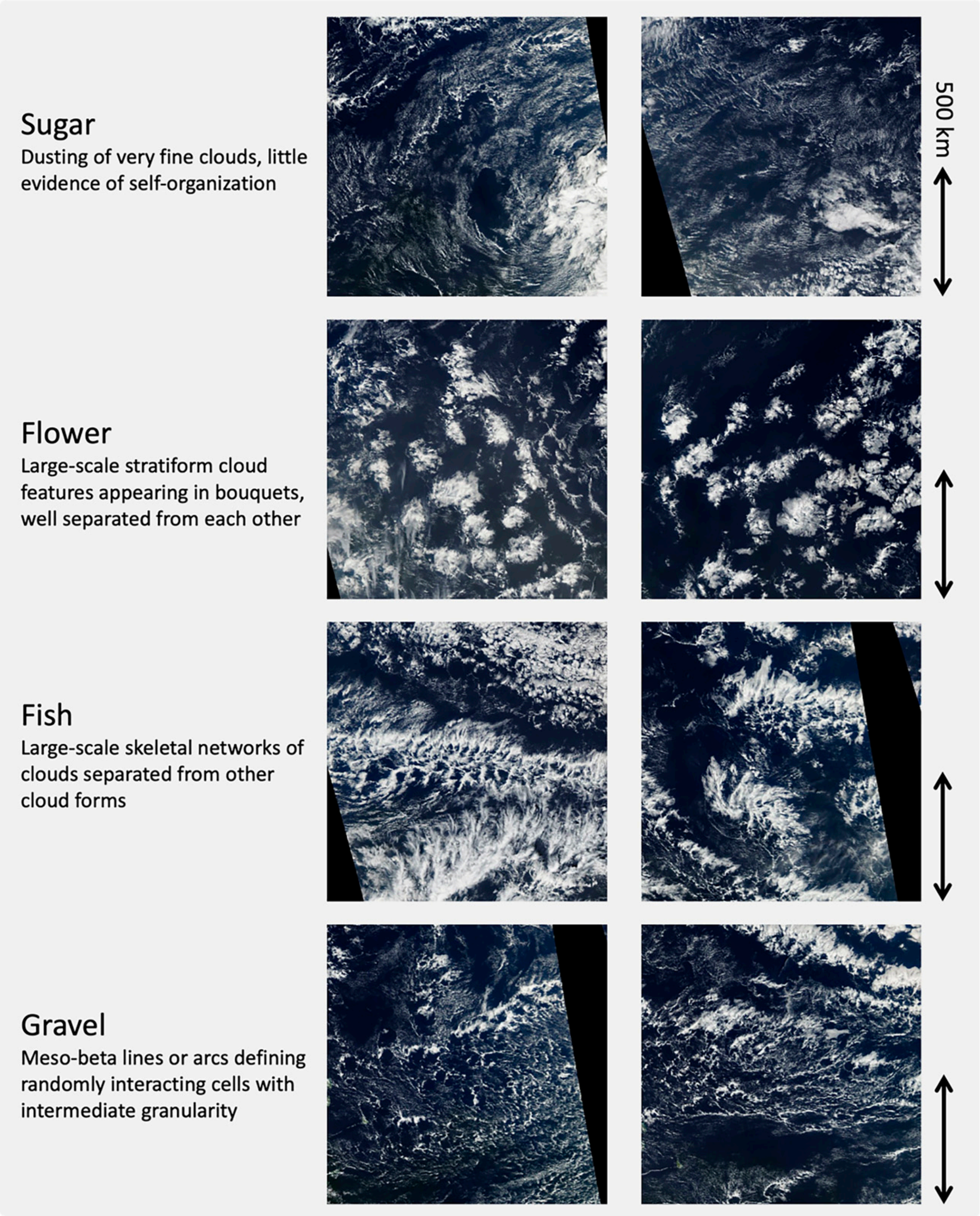
Sugar
Dusting of very fine clouds, little evidence of self-organization
500 km
Flower
Large-scale stratiform cloud features appearing in bouquets, well separated from each other
Fish
Large-scale skeletal networks of clouds separated from other cloud forms
Gravel
Meso-beta lines or arcs defining randomly interacting cells with intermediate granularity

FIG. 2. Examples of the four cloud types from the sugar, flowers, fish, and gravel dataset from Rasp et al. (2020). Note that Rasp et al. (2020) use the term “flower,” whereas we follow Stevens et al. (2020) in referring to this type as the plural “flowers.” Image credit: Figure 1 in Rasp et al. (2020), © American Meteorological Society. Used with permission.

could draw rectangular annotations encompassing a single cloud type and could apply as many annotations to each image as they desired, so long as each annotation encompassed at least 10% of the image.

As we will discuss later, persistent homology takes as its input a space with an intensity value at each point, which in our case corresponds to a grayscale image. The MODIS images in the dataset from Rasp et al. (2020) were NASA Worldview true color images in red–green–blue (RGB; Gumley et al. 2010), which we converted to grayscale using the python package pillow, which uses the International Telecommunication Union Radiocommunication Sector (ITU-R) BT.601-7 luma transform (ITU-R 2011) for computing intensity from RGB input:

$$I = 0.299R + 0.587G + 0.114B.$$

This is a transform originally developed for television broadcasting and approximates the overall perceived brightness for each pixel, which is appropriate here as the NASA Worldview true color images are a close approximation of what a human observer in orbit would see. We note that this is a difference between our work and that of Rasp et al. (2020), because they used the RGB images throughout.

3. Introduction to topological data analysis

In this section, we provide a brief introduction to relevant mathematical topics. For more details, we refer readers to Carlsson (2009), Ghrist (2008), and Edelsbrunner and Harer (2010).

a. Topology

In the broadest sense, topology is the study of the fundamental shapes of abstract mathematical objects. When we speak of the "topology" of an object, we speak of properties that do not change under a smooth reshaping of the object, as if it is made of a soft rubber. Some example properties include how many connected components the object contains, how many holes or voids it contains, and in what ways the object loops back on itself. In this paper, we focus on the first two properties: connectivity and holes.

b. Homology

Homology is one of the tools from topology that focuses on connectivity and holes. The d-dimensional homology H_d (for $d \in \mathbb{Z}_{\geq 0}$) counts the number of d-dimensional holes (or voids) in that object. For $d = 0$, the 0-dimensional homology H_0 captures the number of connected components present in an object. For $d \geq 1$, the homology H_d captures holes: a one-dimensional hole is one that can be traced around with a one-dimensional loop (like a loop of string), while a two-dimensional hole is a void. As shown in Fig. 3, these holes and the surrounding surface need not be circular. Because homology is only interested in counting the presence of these features, it is invariant under any transformation of the space that does not create or destroy any holes or components. In our application of grayscale images, no holes of dimension two or larger can exist, as that would require a dataset that is at least three-dimensional.

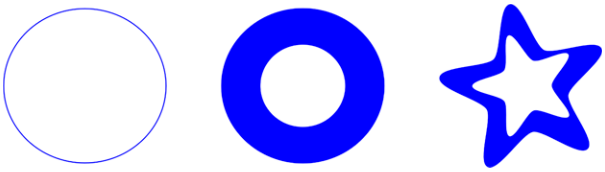

FIG. 3. Three shapes that each have the same homology—a single connected component, a single one-dimensional hole, and no higher-dimensional holes.

c. Persistent homology

Whereas homology focuses on global features of the space, there is an extension—known as "sublevelset/superlevelset persistent homology"—that captures more small-scale geometry (Edelsbrunner and Harer 2010). Superlevelset persistent homology is the primary tool from TDA that we use in this paper, because it gives the best descriptor of image texture.

For an example of superlevelset persistent homology being computed on a simple surface, see Fig. 4. The input to superlevelset persistent homology is a d-dimensional space plus an intensity value at every point. In our example, this is a grayscale image with two spatial dimensions ($d = 2$) with the pixel values as intensities. This input is then converted into superlevelsets: each superlevelset is a binary mask of the original space, in which only points that have an intensity value *greater* than a particular cutoff value have been included. As this cutoff value sweeps down from the maximum intensity, the homology of each superlevelset is computed and the cutoff values at which homological features (connected components, holes) appear and disappear are tracked.

For our example, this means that as the cutoff value decreases, more and more pixels with gradually decreasing intensities are included in the superlevelsets, and we track the connected components and holes that appear and disappear. Because we are using superlevelsets in which we start by including the highest intensities, we can view connected components appearing at high-intensity value as being analogous to cloud tops, which are typically brighter, and holes as darker regions within these bright clouds.

This added interpretability motivates our focus here on superlevelset persistent homology, which is a simple variation (reflection) of the more commonly used sublevelset persistent homology. Sublevelset persistent homology is computed the same way but instead of each set including all the pixels with intensities above the cutoff value, pixels with intensities below the cutoff value are included, and the cutoff value is viewed as sweeping from low intensities up to high intensities. Throughout this paper, we will frequently omit the prefix "superlevelset" in "superlevelset persistent homology" and simply use "persistent homology" to refer to this technique—this should not be confused with the persistent homology technique, which takes as its input a cloud of data points (Carlsson 2009).

In practice, it is not necessary to compute the homology for infinitely many superlevelsets; there are algorithms that discretize the data and then use linear algebra to implement this computation efficiently. These implementations are fast for

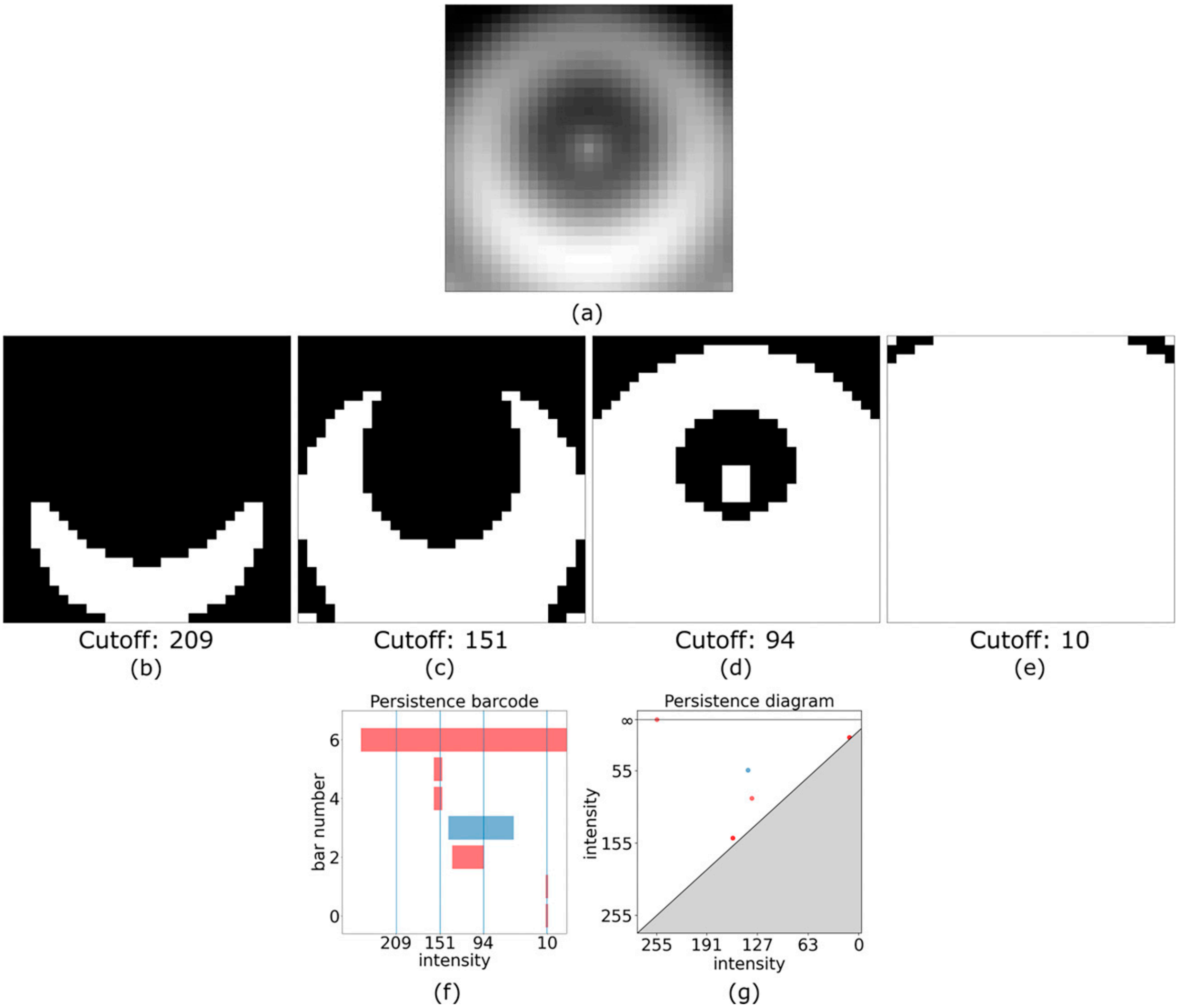

FIG. 4. (a) A grayscale image ranging between intensity 0 for black and 255 for white, and four superlevelsets from (a) at cutoff values (b) 209, (c) 151, (d) 94, and (e) 10. The pixels included in the superlevelsets are colored white. (f) The persistence barcode for (a), with vertical lines from left to right indicating the intensities corresponding to the four superlevelsets in (b)–(e). (g) The equivalent persistence diagram. In the barcode, diagram, and landscape, red elements (bars, points, and lines, respectively) indicate connected components (H_0 features) and blue elements indicate one-dimensional holes (H_1 features).

low-dimensional data (e.g., the two-dimensional grayscale images used in our guiding example) but become more resource intensive when the input space consists of higher-order tensors. In this work, all homological and other TDA computations were performed using the GUDHI software package in Python (Maria et al. 2014).

d. *Persistence barcodes and diagrams*

There are two main ways to display the output of persistent homology: persistence barcodes (Fig. 4f) and persistence diagrams (Fig. 4g).

In a barcode, each homological feature that appears is represented by a horizontal bar, which stretches from the cutoff value at which the corresponding feature first appears (is born) to the value at which it disappears (dies). Because we are using superlevelset persistent homology, our cutoff values are decreasing; thus, the intensity values on the x axis are decreasing from left to right. The persistence of each feature is the length of its bar. To distinguish between different homological classes, we color the bars depending on what dimension the homological feature is. We use red bars for zero-dimensional features (connected components) and blue bars for one-dimensional features (holes). The y axis of a persistence barcode counts the number of bars, typically ordered by birth value.

A persistence diagram contains the same information as a persistence barcode but represents each feature as a point rather than as a bar. In persistence diagrams, both the x axis and y axis represent intensity. The x coordinate of this point is given by the birth cutoff value, while the y coordinate is the death cutoff value. Because features always die at a higher cutoff value than they are born, all points lie above the diagonal line $y = x$. The persistence of a feature is represented by how far a persistence diagram point lies above the diagonal. We present persistence diagrams here to familiarize the reader with their use in, for example, Kim and Vogel (2019), Tymochko et al. (2020), and Sena et al. (2021), but for our case study we focus on barcodes and landscapes.

In persistent homology, there are features that have infinite persistence—features that are born at a particular intensity but never die. The most common example of this is that the first connected component to appear will eventually become the only remaining connected component, as all other components eventually merge into it at high enough cutoff values. These infinite persistence points are represented as infinite bars (rays) in persistence barcodes, stretching out of the frame to the right and as infinite points appearing on a special "$+\infty$" line in persistence diagrams.

e. *How to read and interpret persistence barcodes*

To demonstrate how to read a persistence barcode we return to Fig. 4. The four vertical lines in the barcode in Fig. 4f correspond to the four superlevelsets in the middle row, where white pixels show regions included in the superlevelset. In Fig. 4b, we see the first connected component appear, corresponding to the top, infinite-length red bar in the barcode. In Fig. 4c, two small components in the lower corners appear, corresponding to the two short red bars in the barcode crossed by the second vertical line. The short length of these bars indicates that these components are short lived, and soon merge into the larger component. In Fig. 4d, we see both the central one-dimensional hole, which corresponds to the blue bar, and the connected component within that hole, which corresponds to the red bar that is about to end near the third vertical line in the barcode. In Fig. 4e, we see almost the entire image is in the superlevelset, because the cutoff value is very small. However, the upper two corners have just been included as two new components, which are even shorter lived than the lower-corner components, as indicated by their extremely short red bars.

The first thing to notice about this barcode is that there are relatively few red bars, apart from the infinite-length bar, and those bars are very short. This indicates that few connected components appear and disappear as we scale through intensity values and thus the base image is very smooth. There is one red bar of reasonable length, so we would expect there to be one somewhat significant "bump," a bright region surrounded by darker regions, which is precisely what we see in the middle of Fig. 4a. We also notice that there is one relatively long blue bar, which tells us that there is a hole (dark region surrounded by brighter regions), which persists for a relatively wide range of intensities.

Persistent homology is invariant under homeomorphisms of the input space, which are continuous deformations with continuous inverses (see Fig. 5). Examples include all the rigid motions of the plane (rotation and translation), affine transformations (scaling, skewing, etc.), as well as more radical reshapings, so long as no "ripping" occurs. Superlevelset-persistent homology is invariant over all such transformations. So, a cloud that has been reshaped, expanded, and moved, but which retains the same overall texture as in its original incarnation would have the same superlevelset persistence barcodes. See section 5 for some brief comments on versions of persistent homology that can distinguish between such different deformations of an image.

f. *Persistence landscapes*

Persistence barcodes and diagrams have a drawback: they are not always convenient inputs for use in machine learning tasks, as described by Bubenik (2015), Adams et al. (2017), and Mileyko et al. (2011), since they do not naturally live in a vector space. To deal with this, we use persistence landscapes to summarize and vectorize the persistence diagram (Bubenik 2015). A persistence landscape is a collection of piecewise-linear functions that capture the essence of the persistence diagram.

An example of a persistence landscape computed from a small persistence barcode is shown in Fig. 6. We separate out a particular homological dimension (e.g., H_0 or H_1) and remove any infinite bars and then create a new figure containing a collection of isosceles right triangles with hypotenuses along the x axis, one for each bar in the barcode. These triangles are scaled so that the triangle corresponding to a bar is the same width as that bar. We view this collection of triangles as a "mountain range" and begin to decompose it into landscape functions. The first landscape function is the piecewise-linear function that follows the uppermost edge of the union of these triangles, that is, it is the top silhouette of the mountain range. To compute the next landscape function, we delete the first landscape function from the mountain range and then find the piecewise-linear function that follows the uppermost edge of this new figure at every point, and so forth for the further landscape functions. This collection of piecewise-linear functions is the persistence landscape. The x axis still represents intensity, and the height of each peak is proportional to the length of the bar from which it came and is thus a measure of persistence.

This representation is stable—small changes to the input will only result in small changes in the persistence landscape (Bubenik 2015). While the entire persistence landscape determines a persistence barcode exactly, in our work we retain only the top several persistence landscape functions, which means that we obtain a descriptive summary capturing the information of high-persistence points but ignore some information about low-persistence points.

To compute landscapes, we once again use the GUDHI software package running in Python (Maria et al. 2014).

g. *How to read and interpret persistence landscapes*

We now look at a realistic example in Fig. 7. Reviewing the barcode in Fig. 7b, we first notice two red bars with high persistence: the infinite bar as well as another that stretches nearly all the way across the barcode. This indicates that there are two high-intensity regions that are separated by a dark region. Over middling intensities, there are few bars, indicating that outside the two bright regions we already identified, there is little going on. Finally, when we get to lower intensities (darker regions), there are many short bars, representing subtle variations in the

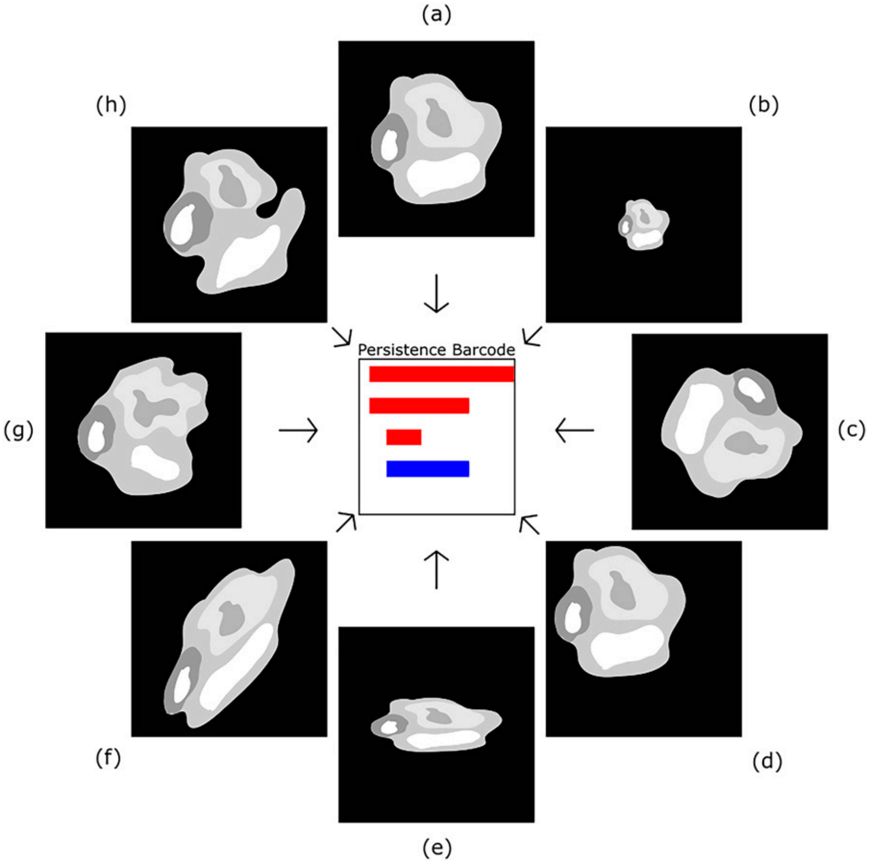

FIG. 5. Examples of deformations that all result in the same persistence barcode: (a) the original image and the transformations: (b) scaling, (c) rotation, (d) translation, (e) uneven scaling, (f) shearing, and (g),(h) general homeomorphisms.

dark regions. This information, however, is somewhat hard to read in a barcode with as many bars as this. Thus, we turn to landscapes as a way to summarize this information in a more readable format.

In the landscape in Fig. 7d, the single, tall, red mountain indicates that the original image (Fig. 7a) contains two connected components that persist over a large range of intensity levels (as the infinite-persistence component is implicit in the landscape). The only blue mountains in the landscape are much smaller than the red mountains and are mainly to the right of them. This indicates that while the original image contains numerous holes within the connected components, they only appear near the bottom end of the intensity range, that is, the holes do not appear until we have started including relatively dark regions. As an example, consider the single extremely dark pixel in the upper left-hand corner adjacent to the bright clouds and surrounded by a moderately dark region. This hole contributes a moderately tall blue peak far to the right in the landscape, as it does not appear until the relatively dark region surrounding it is included into the bright adjacent component, but it will not fill in until the nearly black pixel in the middle of the hole is included.

In general, high-persistence features (long bars, tall mountains) give information about large-scale features—the presence of two bright clouds in Fig. 7a, for example. On the other hand, low-persistence features (short bars, small mountains) give information about texture. In Fig. 7, the short bars and small mountains appearing at lower intensity values indicate that the background darkness in the image is relatively noisy rather than being uniformly black or smoothly graded. The few small blue mountains in Fig. 7d indicate that the bright clouds also contain some textural elements—regions of slightly darker cloud within brighter regions.

4. Environmental science satellite case study

Now that we have established the basic theory of TDA, we return to its application to classifying mesoscale clouds.

a. Adapting persistent homology to this dataset

We want to use persistent homology and landscapes to compare the clouds present in the rectangular annotations on images in the dataset of Rasp et al. (2020), so we need to find a consistent vector representation for these annotations. While the overall images in the dataset are of consistent size (1400 × 2100 pixels), the annotations are not. An annotated region covering more area has inherently more complexity, which would yield a barcode with more bars (and thus a

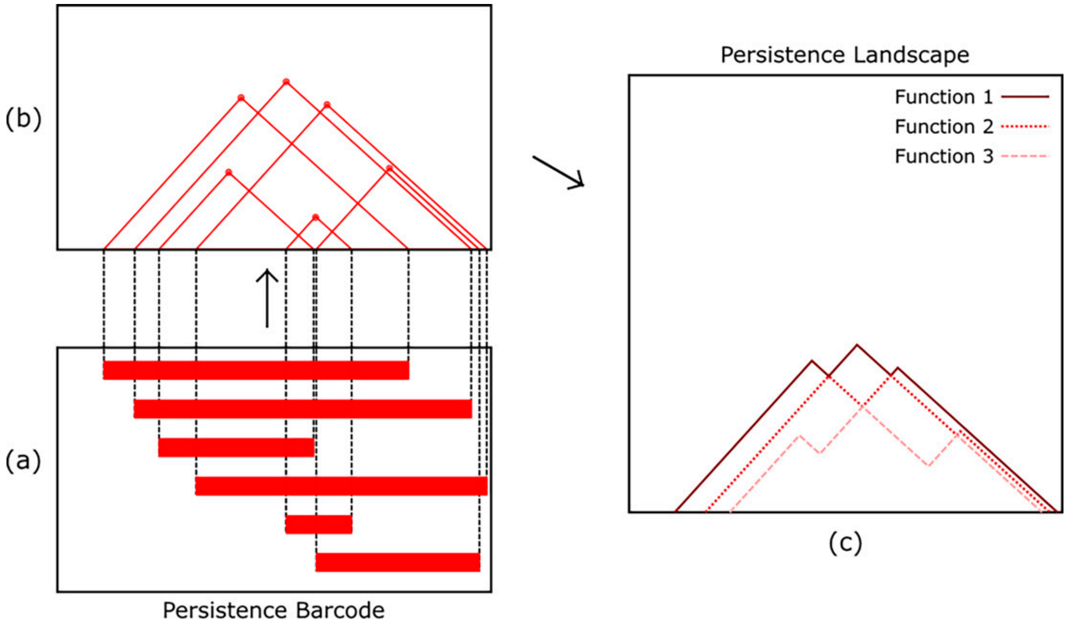

FIG. 6. The process of computing a persistence landscape from a barcode. Beginning with (a) the barcode (which already has the infinite bar removed), we (b) raise a "mountain" above each bar to obtain the "mountain range." Our first persistence landscape function is the piecewise-linear function that follows the highest edges of the mountain range in (b) [shown as the solid line in (c)], and (c) the next function is obtained by deleting in (b) the lines corresponding to this first function and then finding the piecewise-linear function that follows the highest edges of this modified mountain range in (b) (the dotted line). Further landscape functions are computed similarly by deleting previous functions in (b) and tracing along the highest remaining edges. In (c), only the first three landscape functions are shown.

different vector representation) than a smaller annotation. To account for this, we implemented a subsampling routine, which is illustrated in Fig. 8. For each annotation, we randomly chose six 96 × 96 pixel regions (the subsamples) and computed their persistence landscapes individually. This is shown in Figs. 8a–c. Each landscape consisted of 10 piecewise-linear functions: 5 giving information on connected components (plotted in red) and 5 giving information on one-dimensional holes (plotted in blue). The height of each function was recorded at 200 evenly spaced locations along the intensity axis, giving a vector of length 200 representing that function. These 10 vectors of length 200 were concatenated to yield a vector of length 2000 (i.e., a point in $\mathbb{R}^{2000}$), representing the persistence landscape of that particular subsample. This vectorization is shown in Fig. 9. At that point, the persistent homology of each annotation was represented by a small-point cloud of six points in $\mathbb{R}^{2000}$, one for each of the subsamples. To obtain a single point to represent the annotation, we took the geometric vector average of the six points in the point cloud, giving us the single vector that we can use to compare and analyze our annotations. Additionally, we can display and interpret this average vector as a landscape, in that it can be displayed as 10 functions, 5 representing the persistence and intensity ranges of connected components and 5 representing the same for one-dimensional holes. This averaged landscape is visualized in Fig. 8d. In particular, we can view (for instance) the first 200 values of the average landscape vector as the heights of the first average landscape function at the 200 evenly spaced intensity values where we sampled the piecewise landscape functions. We can view these values as coming from two equivalent formulations: first, as described above, as coming from a pointwise vector average, or second, by taking the average height of the first piecewise landscape function at each of the 200 evenly spaced intensity values across the six subsamples.

b. *Dimensionality reduction and adding a simple machine learning model to build a classifier*

Once we obtained vectorized representations of each annotation, we sought to visualize the dataset. Because $\mathbb{R}^{2000}$ is not visualizable, we applied a dimensionality reduction algorithm to yield a representation that we can plot. We used principal component analysis (PCA), as it is a widely used and relatively simple technique, which in our case produced quite good results. We found that patterns in the data were visible upon projecting down onto the first three principal components, which captured over 90% of the variation in the high-dimensional data. We also note that the principal component vectors from repeated random samplings were extremely consistent, indicating that our projections were quite stable.

Once the data were projected down to three dimensions, we could visualize the data as a point cloud, with points colored according to which cloud pattern they represent. As a note, the PCA algorithm was entirely unsupervised with regard to these cloud pattern labels; it used only the vectorized representation of the persistence diagram.

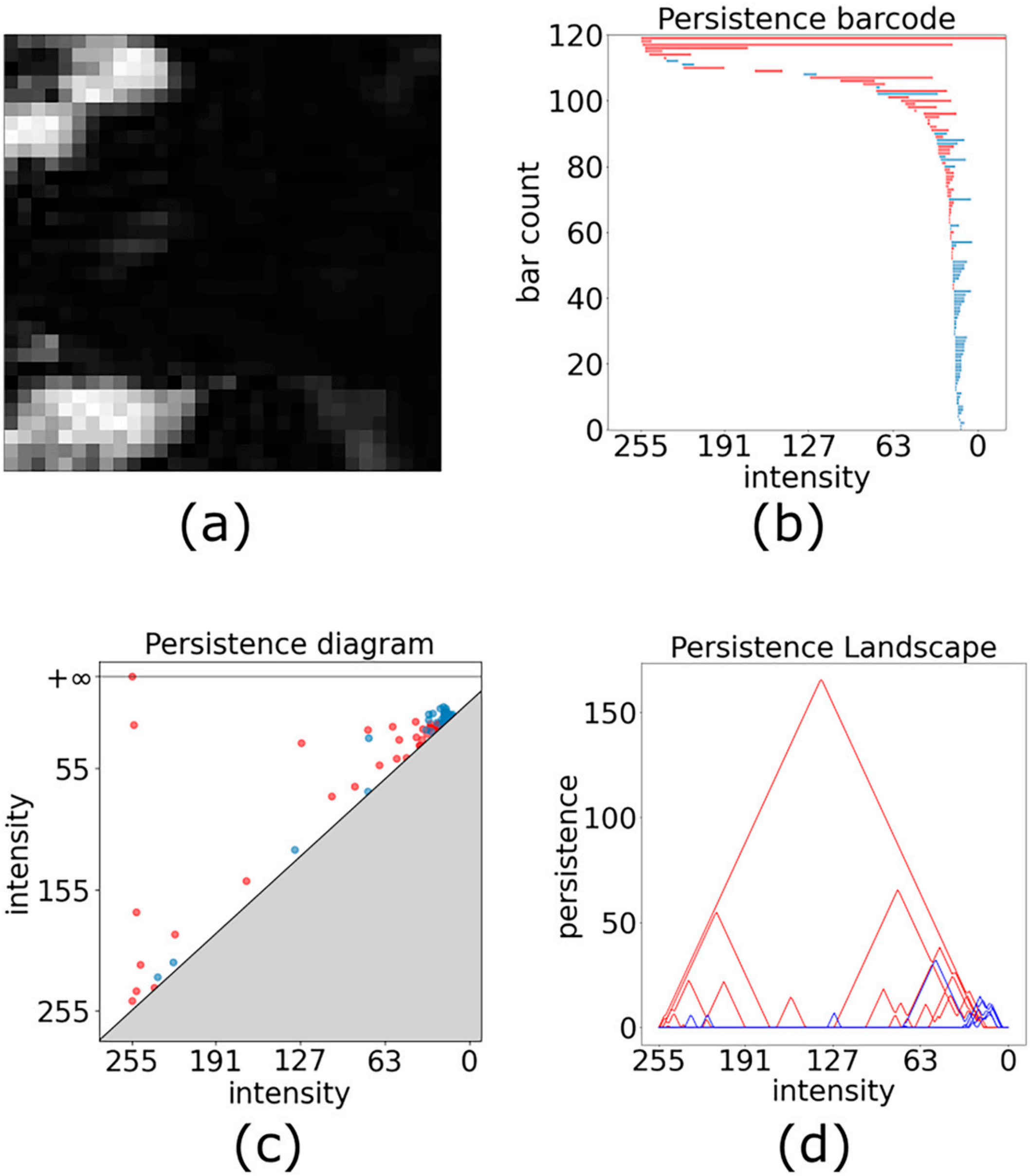

FIG. 7. (a) A 32 × 32 sample image from the grayscale MODIS imagery used in the sugar, flowers, fish, and gravel dataset, along with its (b) persistence barcode, (c) persistence diagram, and (d) persistence landscape with the first five piecewise-linear functions for each of the H_0 and H_1 classes.

We analyzed each of the six pairs of classes separately by training an SVM (Boser et al. 1992) to find the plane that best separates the two classes of projected data in three dimensions. We considered running the SVM on the high-dimensional data, but initial testing indicated that this tended to overfit the data and actually resulted in reduced classification performance. The SVM was trained on a random sample of 350 annotations of each class, then performance metrics were computed for a test set of 200 random annotations of each class. For visualizations of the test data and SVM separating plane as well as performance metrics for each of the six pairwise comparisons, see Fig. 10.

c. *Results: What initial patterns emerged from applying persistent homology?*

Overall, the projected points separated well, with one notable exception. We began our investigation by comparing the sugar versus flowers patterns, because these are visually the most dissimilar (Fig. 2). As expected, these classes had the most distinct separation out of the six possible pairs, as seen in Fig. 10a. While the separation was not perfect, most of the error comes in the form of some intermingling near the separating hyperplane. The performance of the algorithm in separating these classes was striking, given that only a small,

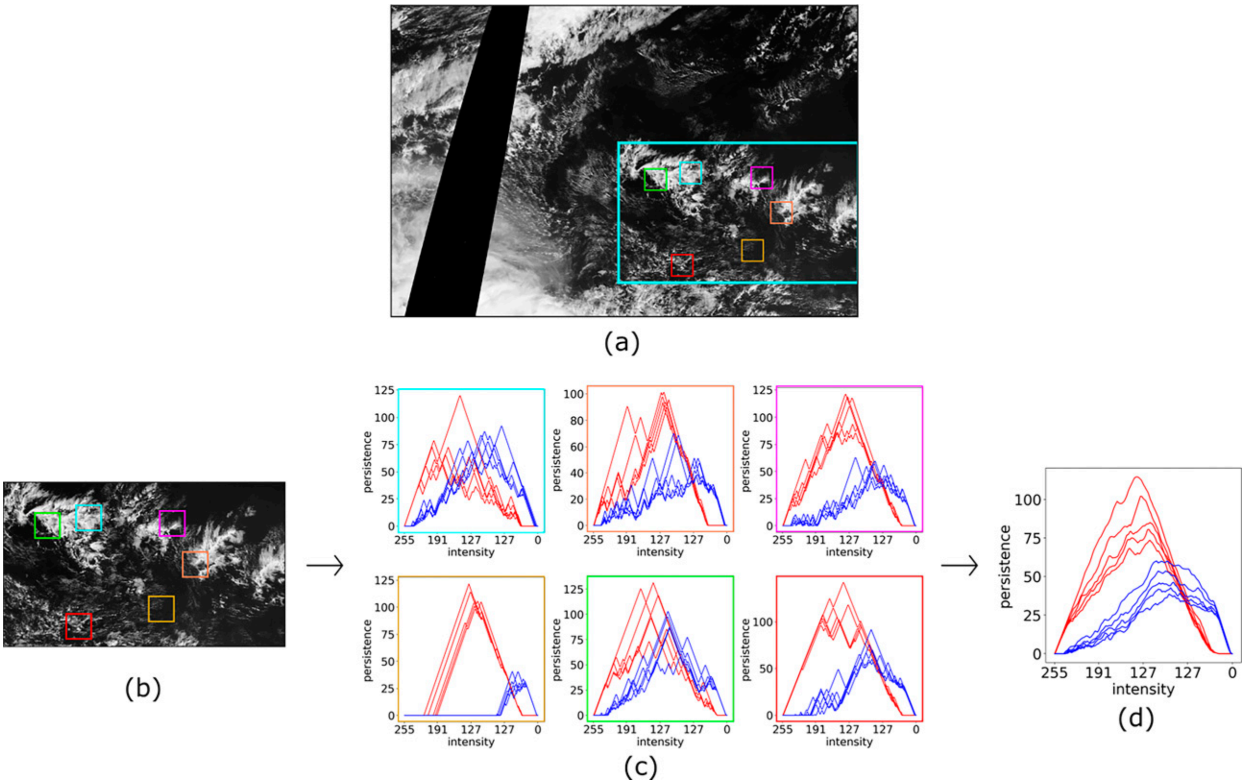

FIG. 8. An illustration of the annotation and subsampling process: (a) a full 14° × 21° image, with an example fish annotation outlined in blue, with the subsample boxes outlined in various colors inside, (b) an enlarged view of this annotated region, including the same color-coded subsamples, (c) the corresponding landscapes for each subsample, and (d) the resulting averaged landscape.

random subset of each annotation was included and that the PCA projection algorithm was entirely blind to the data labels. This example is a clear indicator of the potential that persistent homology has to usefully extract textural and shape differences in satellite imagery.

While not quite as exceptional, the flowers and gravel patterns also separate well, as seen in Fig. 10c. There is again a degree of intermingling near the separating plane and in this case that intermingling extends a bit farther to either side. This is what we would expect, based on the visual presentation of the cloud regimes; the gravel class falls somewhere between sugar and flowers in terms of cloud size and organization.

We begin to see the algorithm struggle a bit more when we attempt to separate the sugar regime from the gravel regime. As we can see in Fig. 10e, the intermingling of data points stretches throughout the point cloud, although there is still a difference in densities between the classes on either side of the separating hyperplane.

However, there are two classes that are effectively indistinguishable by this algorithm: the flowers and fish patterns. The plot of these points can be seen in Fig. 10f, and it is apparent that there is no effective linear separator between these classes. While there is a "separating" hyperplane plotted, it is much less relevant in this case than in the others; the data points are remarkably evenly mixed. This proved to be the case even when more principal components were included. A potential explanation for why the algorithm struggles so much with this task is that the distinguishing features of fish versus flowers are simply too large scale for the subsampling technique to pick up on. The fish pattern is characterized by its mesoscale skeletal structure, particularly in its difference from the flowers regime, which is more randomly distributed. This mesoscale organization is simply not visible to the subsamples, as the 96 × 96 patches are too small to detect that skeletal structure. Future analyses could include using larger patches to better capture these features and perhaps distinguish between these classes more effectively. We also note that in Rasp et al. (2020), the fish pattern was the most controversial among the expert labelers, so it is perhaps not surprising that our algorithm also struggles.

When we look at fish versus sugar and fish versus gravel in Figs. 10b and 10d, respectively, we can see how similar these plots appear to those in Figs. 10a and 10c, in which the flowers pattern was compared with sugar and gravel. This similarity is made even more remarkable by the fact that the sugar samples in these plots were drawn separately rather than being reused for the pairwise comparisons (and similarly for the gravel samples). While the algorithm is not doing well at distinguishing between fish and flowers, we can at least see that its behavior is consistent: fish and flowers are projected similarly into the 3D embedding space, so they compare similarly to the other classes.

Overall, this case study suggests that it is possible to use persistent homology to quantify and understand the shape and texture

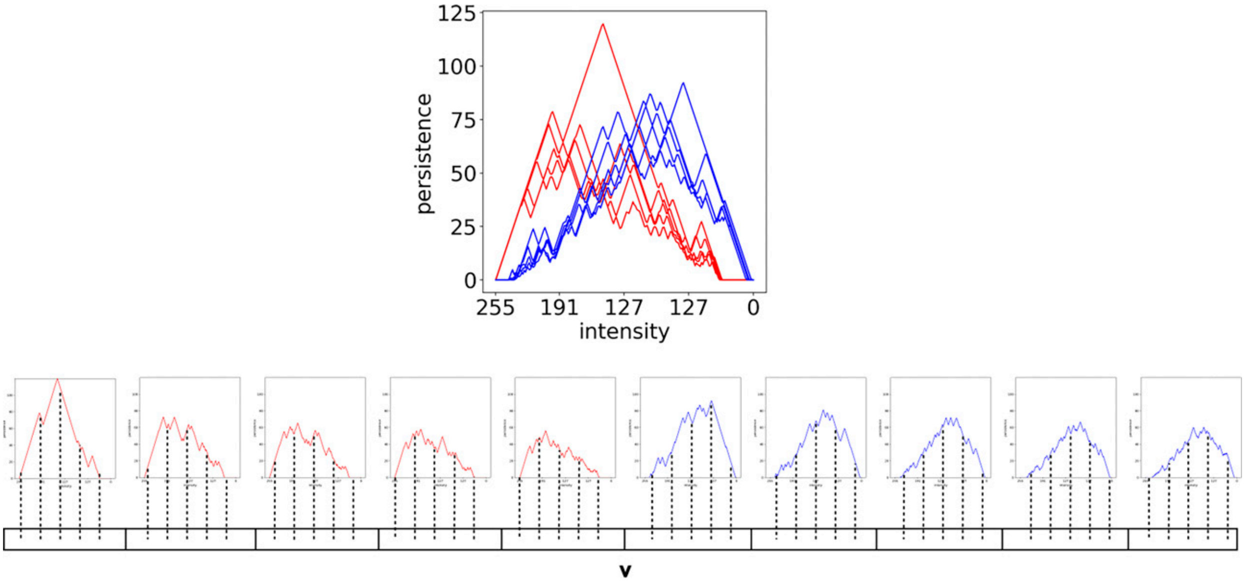

FIG. 9. A cartoon showing the sampling and concatenation of a persistence landscape into a vector. In this figure, we see the landscape, each landscape function individually, and the process of sampling each landscape function evenly and then concatenating the results into a single vector **v**.

of satellite cloud data. While there are cases where the algorithm struggles, these are understandable in terms of the visual task being requested and are internally consistent from sample to sample. Moreover, in the cases where the algorithm does well, it does so consistently across repeated samples and suggests that when these tools are appropriately applied, excellent results can be obtained from very limited sample sizes.

d. *A novel interpretation method: Deriving interpretations in terms of weather and homology*

As an example of how this separation can be interpreted, we examine the case of sugar versus flowers. Recall that in Fig. 10a, we saw that this pair of classes had the strongest separation in the dataset.

To begin, we explore what can be learned just from the summarized data, without looking at examples. To discover what the separating plane between sugar points and flowers points represents, we create "virtual" landscapes. We first lift the separating plane in $\mathbb{R}^3$ to the hyperplane in $\mathbb{R}^{2000}$ consisting of all the points that project (under PCA) into the separating plane in $\mathbb{R}^3$. Next, we find the line normal to this hyperplane that passes through geometric center of the data. Finally, we choose points on this line that fall at the outer extent of the data point cloud. These points live in the landscape embedding space $\mathbb{R}^{2000}$ but are not sampled data points. However, by applying the inverse landscape embedding, we can visualize the landscape-like set of curves that would give this embedded point. Virtual landscapes for sugar and flowers can be seen in Figs. 11c and 11f, respectively.

An advantage of this approach is that it synthesizes trends from the real data into a readable, controllable format that demonstrates how SVM is separating these classes. When we compare these virtual landscapes with the actual landscapes farthest from the separating hyperplane (seen in Figs. 11b,e), we can see that the virtual landscapes are smoother but that the overall shapes are remarkably similar.

We can also interpret the shapes of these landscapes in terms of the features present in the images. Let us examine the images and corresponding landscapes in Fig. 11. The most prominent feature in the two landscapes is the tall red peak in the sugar landscape, shown in Fig. 11b. Recall that the red lines denote zero-dimensional homology (connected components), while the blue lines denote one-dimensional homology (holes). This red peak represents the presence of strongly persistent connected components, that is, separated regions of bright cloud strongly contrasting against a much darker background. The sharpness of this peak also indicates that these features are similar in both the intensity of the cloud top and the intensity of the surrounding background. The comparatively low blue curves with only a small peak at the end indicate a lack of one-dimensional homological features (holes), and thus the texture within connected components is relatively uniform. Looking at the image in Fig. 11a, we see these observations borne out: there are numerous small clouds of similar brightnesses, which stand in stark contrast to the overall uniformly dark background, matching the tall H_0 peak. Because these clouds are relatively small, there is little discernible texture within each cloud, which corresponds to the relative absence of H_1 features.

On the other hand, the flowers landscape in Fig. 11e displays a lower red peak, with more separated curves. This lack of concentration indicates that there is more variation in the intensity values at which connected components appear (the brightest part of the component) and at which they merge together (the intensity of the bridge connecting that component to another), while the lower height indicates that these features are overall less persistent—they merge into one another more rapidly.

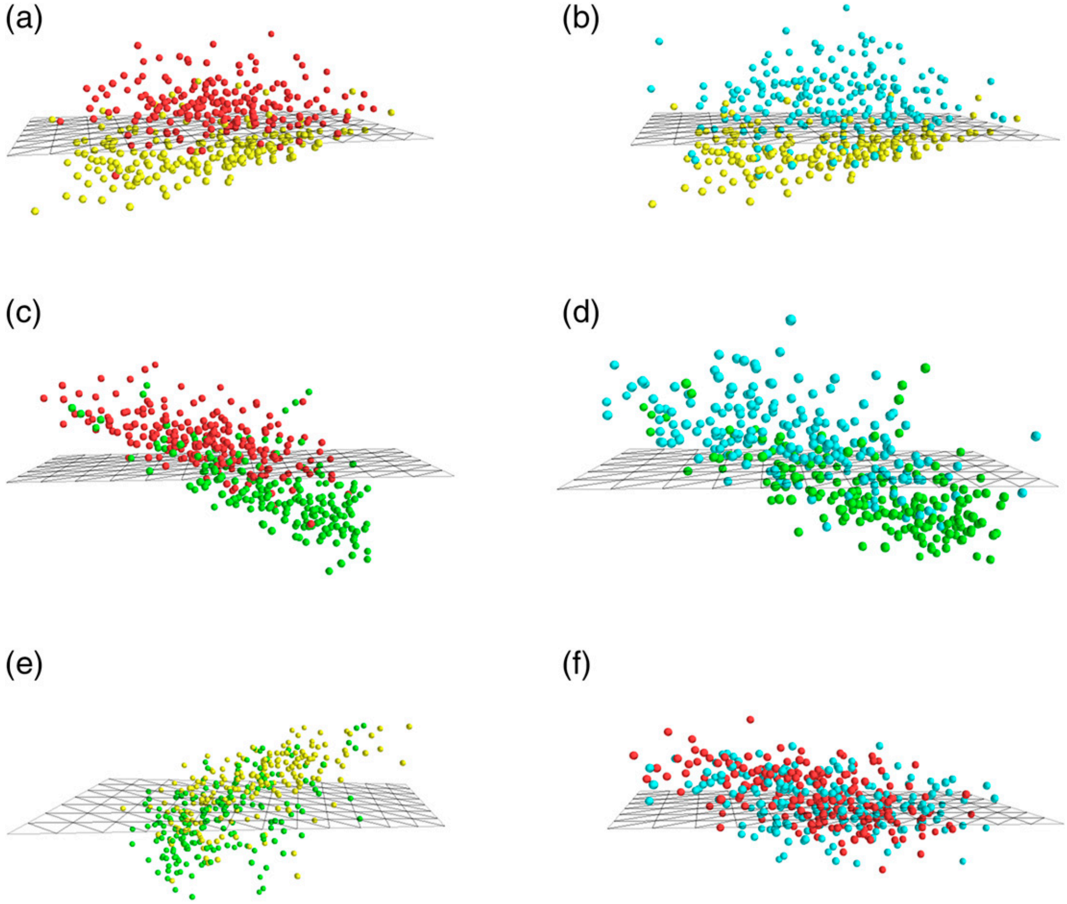

FIG. 10. Plots of the embedded landscape test points for each pairwise combination: (a) sugar vs flowers (89.25%), (b) fish vs sugar (86%), (c) flowers vs gravel (81%), (d) fish vs gravel (79.5%), (e) sugar vs gravel (71.25%), and (f) fish vs flowers (57%). Flower points are red, sugar points are yellow, gravel points are green, and fish points are blue, with color representing the class assigned in the crowdsourced dataset in Rasp et al. (2020). The SVM separating plane for each pair is shown in wireframe, and the percentage of correctly classified points for each combination (test performance) is reported in parentheses earlier in the caption.

Additionally, there is a much stronger H_1 signal in this case than in the sugar landscape, meaning that the connected components have more internal texture, with numerous holes appearing and disappearing over a wide range of intensity values. These observations match with what we see in Fig. 11d. The clouds in this image are much larger and cover more of the frame, with varying intensities within and between the clouds, leading to the more varied H_0 landscape. This image also shows much more internal texture to the clouds, with far more of a dimpling effect than in the sugar example.

In summary, this example shows how the patterns learned by the TDA–SVM algorithm can be translated back to homological features which in turn correspond to weather-relevant features in the original image. This is made possible by the fact that the SVM model can be represented by a single separating plane, which can be translated back into the space of persistence landscapes and then interpreted, yielding a highly interpretable approach to the pairwise classification problem.

e. *Comparison of this classifier with those in Rasp et al. (2020)*

1) ACCURACY

The accuracy of our approach cannot be directly compared with the deep learning algorithms in Rasp et al. (2020) because they address different tasks. The task considered here is to choose a single class (out of two) for an annotation assumed to consist of a single cloud type based on several small patches (96 × 96 pixels). In contrast, the task considered in Rasp et al. (2020) is much more complex, namely, to assign one or more labels for an annotation based on a very large image. We choose the simpler task for our TDA approach in order to expose the properties of a TDA algorithm, and trying to implement a multi-label assignment (e.g., by using a sliding window approach) likely would have made this exploration more complicated without providing new insights. However, even without a direct comparison, it is obvious from the results that this first TDA–SVM

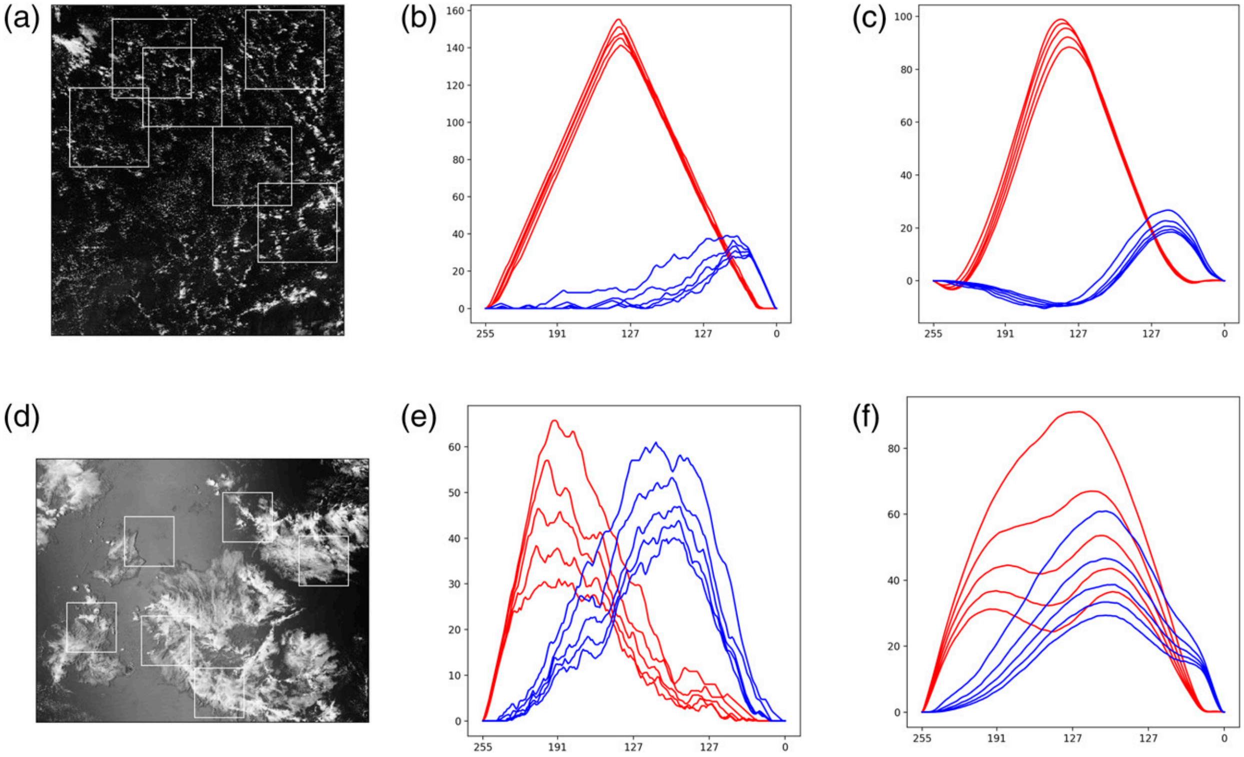

FIG. 11. The (a) sugar and (d) flowers samples farthest from the separating hyperplane (most extreme example) and (b),(e) their respective landscapes. Also shown are the "virtual" (c) sugar and (f) flowers landscapes obtained by traveling along the line normal to the separating hyperplane.

approach cannot nearly achieve the accuracy of the deep learning approaches.

2) REQUIRED DATA SAMPLES

Our approach only requires a few hundred labeled data samples to develop a classifier. This reduces the required labeling effort by two orders of magnitude relative to the tens of thousands of labeled samples in Rasp et al. (2020).

3) COMPUTATIONAL EFFORT

Computations were performed on a Surface Pro 6 computer with an Intel Core i5-8250U CPU. The computational bottleneck in this case was computing the persistent homology: for 800 samples (and therefore 4800 subsamples to compute persistent homology for), approximately 45 min of wall-clock time was required. This is already much less computational time than is generally required to train a deep network, and it is likely that this could be significantly improved by parallelizing, because each sample can be processed entirely separately.

4) INTERPRETABILITY AND FAILURE MODES

Our approach yields a highly interpretable model that provides an intuitive explanation of how the algorithm distinguishes different classes, while the deep learning methods do not. Furthermore, the interpretation of the separation plane in our model makes it easy to provide insights into failure modes, that is, which types of mesoscale patterns can be easily or not so easily be distinguished by their topological features, and thus by this approach.

5. Advanced TDA concepts

In this section, we briefly discuss and provide references for some advanced TDA concepts that are beyond the scope of this article, along with motivations for when and why readers might find them useful.

Figure 5 shows that while persistent homology measures some spatial aspects of the intensity function, it is also invariant under nice deformations ("homeomorphisms") of the domain. However, there is another (very popular) type of persistent homology, constructed using growing offsets of a shape, or unions of growing balls, that distinguishes between different deformations of the domain (Carlsson 2009; Ghrist 2008). We expect that this variant of persistent homology will also find applications in atmospheric science, and we refer the reader to Tymochko et al. (2020) for such an example.

We are particularly interested in exploring the use of TDA to analyze cloud properties from satellite imagery, for example, to detect convection. While the example here looked at large-scale organization of clouds, to analyze properties like convection we would zoom far into a single cloud and analyze its texture, for example, seek to identify whether there is a "bubbling" texture apparent in a considered area of the cloud.

Preliminary analysis leads us to believe that it might be necessary to use more sophisticated TDA tools for this purpose than discussed here, such as vineyards (Cohen-Steiner et al. 2006) or "contour realization of computed *k*-dimensional hole evolution in the Rips complex (CROCKER)" plots (Topaz et al. 2015), which incorporate temporal context by analyzing the topological properties of sequences of images rather than individual images.

We have considered persistent homology that varies over a single parameter—the intensity of the satellite image. However, one frequently encounters situations in which two or more parameters naturally arise. For example, one can perform superlevelset persistent homology on a two-channel image, containing the intensities with respect to two frequencies, with respect to the parameter from either the first channel or the second. In these contexts, multiparameter persistence (Carlsson et al. 2009; Carlsson and Zomorodian 2009; Cerri et al. 2013; RIVET Developers 2020) allows one to consider both parameters at once, even though the underlying mathematics is more subtle, and computations are more difficult. A version of multiparameter persistence was applied recently to the atmospheric domain in Strommen et al. (2023).

Persistence barcodes and diagrams are not ideal as inputs into machine learning algorithms because they are not vectors residing in a linear space. This is evidenced by the fact that averages of persistence diagrams need not be unique (Mileyko et al. 2011). There is a wide array of options for transforming persistence diagrams for use in machine learning, including not only persistence landscapes (Bubenik 2015) but also persistence images (Adams et al. 2017) and stable kernels (Reininghaus et al. 2015), for example. TDA has been gaining traction in machine learning tasks as more tools become available to integrate it into existing workflows in both neural network layers (Moroni and Pascali 2021) and loss functions (Clough et al. 2020). As an example application, TDA has recently been used to compare models with differing grids and resolutions (Ofori-Boateng et al. 2021).

There is a variant of superlevelset persistent homology called extended persistent homology (Edelsbrunner and Harer 2010, their section VII.3), which performs two sweeps (instead of just one) over the range of intensity values. Extended superlevelset persistent homology detects all of the features measured by superlevelset persistent homology, plus more. It may be the case that one can extract more discriminative information from a satellite image by instead computing the extended persistence diagram.

6. Conclusions and future work

The primary contributions of this paper are as follows: 1) It presents, to the best of our knowledge, the first attempt to provide a comprehensive, easy-to-understand introduction to popular TDA concepts customized for the environmental science community. In particular, we seek to provide readers with an intuitive understanding of the topological properties that can be extracted using TDA by translating cloud imagery into persistence landscapes, interpreting the landscapes, then highlighting the topological properties in the original images. 2) In a case study, we demonstrate step by step the process of applying TDA, combined with a simple machine learning model, to extract information from real-world meteorological imagery. The case study focuses on how to use TDA to classify mesoscale organization of clouds from satellite imagery, which has never been addressed by TDA before. 3) The most novel contribution is the interpretation procedure we developed that projects the class separation planes identified by the SVM algorithm back into topological space. This, in turn, allows us to fully understand the strategy used by the classifier in meteorological image space, thus providing a fully interpretable classifier.

In future work we seek to explore several of the advanced methods outlined in section 5. We believe that there are many applications to be explored with TDA, including the applications suggested by Rasp et al. (2020) for their methods, namely, "detecting atmospheric rivers and tropical cyclones in satellite and model output, classifying ice and snow particles images obtained from cloud probe imagery, or even large-scale weather regimes" (Rasp et al. 2020). Furthermore, as discussed in section 1, TDA has already been shown to be useful to identify certain properties of atmospheric rivers, wildfires, and hurricanes, and we expect TDA to find additional use in those areas as well. Our group is particularly interested in using TDA to detect convection in clouds and to distinguish blowing dust from, say, blowing snow in satellite imagery.

We have only scratched the surface here of exploring how TDA can support image analysis tasks in environmental science, but we hope that this primer will accelerate the use of TDA for this purpose.

Acknowledgments. The work by authors Ver Hoef and Ebert-Uphoff was supported in part by the National Science Foundation under Grant OAC-1934668 and under Grant ICER-2019758, which funds the NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography. Author Adams thanks the Institute of Science and Technology Austria for hosting him during this research. We thank Charles H. White at CIRA (CSU) for constructive feedback and excellent suggestions for this paper. The authors also thank Rasp et al. (2020) for making their dataset publicly available.

Data availability statement. The underlying crowdsourced data from Rasp et al. (2020) are available online (https://github.com/raspstephan/sugar-flower-fish-or-gravel). The code used in our analysis is available as a GitHub repository (https://github.com/zyjux/sffg_tda).

REFERENCES

Adams, H., and Coauthors, 2017: Persistence images: A vector representation of persistent homology. *J. Mach. Learn. Res.*, **18** (8), 1–35, https://www.jmlr.org/papers/volume18/16-337/16-337.pdf; Corrigendum, https://jmlr.org/papers/volume18/16-337/pi_erratum.pdf.

Boser, B. E., I. M. Guyon, and V. N. Vapnik, 1992: A training algorithm for optimal margin classifiers. *Proc. Fifth Annual Workshop* *on Computational Learning Theory: COLT'92*, Pittsburgh, PA, ACM, 144–152, https://doi.org/10.1145/130385.130401.

Brenowitz, N. D., T. Beucler, M. Pritchard, and C. S. Bretherton, 2020: Interpreting and stabilizing machine-learning parametrizations of convection. *J. Atmos. Sci.*, **77**, 4357–4375, https://doi.org/10.1175/JAS-D-20-0082.1.

Bubenik, P., 2015: Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, **16**, 77–102.

Carlsson, G., 2009: Topology and data. *Bull. Amer. Math. Soc.*, **46**, 255–308, https://doi.org/10.1090/S0273-0979-09-01249-X.

——, and A. Zomorodian, 2009: The theory of multidimensional persistence. *Discrete Comput. Geom.*, **42**, 71–93, https://doi.org/10.1007/s00454-009-9176-0.

——, G. Singh, and A. Zomorodian, 2009: Computing multidimensional persistence. *International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, Vol. 5878, Springer, 730–739, https://doi.org/10.1007/978-3-642-10631-6_74.

Cerri, A., B. D. Fabio, M. Ferri, P. Frosini, and C. Landi, 2013: Betti numbers in multidimensional persistent homology are stable functions. *Math. Methods Appl. Sci.*, **36**, 1543–1557, https://doi.org/10.1002/mma.2704.

Chung, M. K., P. Bubenik, and P. T. Kim, 2009: Persistence diagrams of cortical surface data. *International Conference on Information Processing in Medical Imaging*, Lecture Notes in Computer Science, Vol. 5636, Springer, 386–397, https://doi.org/10.1007/978-3-642-02498-6_32.

Clough, J., N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. King, 2020: A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Trans. Pattern Anal. Mach. Intell.*, **44**, 8766–8778, https://doi.org/10.1109/TPAMI.2020.3013679.

Cohen-Steiner, D., H. Edelsbrunner, and D. Morozov, 2006: Vines and vineyards by updating persistence in linear time. *Proc. 22nd Annual Symp. on Computational Geometry*, Sedona, AZ, ACM, 119–126, https://doi.org/10.1145/1137856.1137877.

Denby, L., 2020: Discovering the importance of mesoscale cloud organization through unsupervised classification. *Geophys. Res. Lett.*, **47**, e2019GL085190, https://doi.org/10.1029/2019GL085190.

Deshmukh, V., S. Baskar, E. Bradley, T. Berger, and J. D. Meiss, 2022: Machine learning approaches to solar-flare forecasting: Is complex better? arXiv, 2202.08776v1, https://doi.org/10.48550/arXiv.2202.08776.

Ebert-Uphoff, I., and K. Hilburn, 2020: Evaluation, tuning, and interpretation of neural networks for working with images in meteorological applications. *Bull. Amer. Meteor. Soc.*, **101**, E2149–E2170, https://doi.org/10.1175/BAMS-D-20-0097.1.

Edelsbrunner, H., and J. L. Harer, 2010: *Computational Topology: An Introduction*. American Mathematical Society, 294 pp.

Fletcher, S., and M. Z. Islam, 2018: Comparing sets of patterns with the Jaccard index. *Australas. J. Inf. Syst.*, **22**, https://doi.org/10.3127/ajis.v22i0.1538.

Gagne, D. J., II, S. E. Haupt, D. W. Nychka, and G. Thompson, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Wea. Rev.*, **147**, 2827–2845, https://doi.org/10.1175/MWR-D-18-0316.1.

Gardner, R. J., E. Hermansen, M. Pachitariu, Y. Burak, N. A. Baas, B. A. Dunn, M.-B. Moser, and E. I. Moser, 2022: Toroidal topology of population activity in grid cells. *Nature*, **602**, 123–128, https://doi.org/10.1038/s41586-021-04268-7.

Gentine, P., M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis, 2018: Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.*, **45**, 5742–5751, https://doi.org/10.1029/2018GL078202.

Ghrist, R., 2008: Barcodes: The persistent topology of data. *Bull. Amer. Math. Soc.*, **45**, 61–76, https://doi.org/10.1090/S0273-0979-07-01191-3.

Gumley, L., J. Descloitres, and J. Schmaltz, 2010: Creating reprojected true color MODIS images: A tutorial. Space Science and Engineering Center Tech. Rep., 17 pp., https://cdn.earthdata.nasa.gov/conduit/upload/946/MODIS_True_Color.pdf.

ITU-R, 2011: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios. International Telecommunication Union Tech. Rep. BT.601, 20 pp., https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-E.pdf.

Jaccard, P., 1901: Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bull. Soc. Vaud. Sci. Nat.*, **37**, 547–579, https://doi.org/10.5169/seals-266450.

Kim, H., and C. Vogel, 2019: Deciphering active wildfires in the southwestern USA using topological data analysis. *Climate*, **7**, 135, https://doi.org/10.3390/cli7120135.

Kramár, M., R. Levanger, J. Tithof, B. Suri, M. Xu, M. Paul, M. F. Schatz, and K. Mischaikow, 2016: Analysis of Kolmogorov flow and Rayleigh–Bénard convection using persistent homology. *Physica D*, **334**, 82–98, https://doi.org/10.1016/j.physd.2016.02.003.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov, 2005: New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Mon. Wea. Rev.*, **133**, 1370–1383, https://doi.org/10.1175/MWR2923.1.

Lawson, P., A. B. Sholl, J. Brown, B. T. Fasy, and C. Wenk, 2019: Persistent homology for the quantitative evaluation of architectural features in prostate cancer histology. *Sci. Rep.*, **9**, 1139, https://doi.org/10.1038/s41598-018-36798-y.

L'Ecuyer, T. S., and Coauthors, 2015: The observed state of the energy budget in the early twenty-first century. *J. Climate*, **28**, 8319–8346, https://doi.org/10.1175/JCLI-D-14-00556.1.

Maria, C., J.-D. Boissonnat, M. Glisse, and M. Yvinec, 2014: The Gudhi library: Simplicial complexes and persistent homology. *Mathematical Software—ICMS 2014*, Lecture Notes in Computer Science, Vol. 8592, Springer, 167–174.

McGovern, A., R. Lagerquist, D. J. Gagne, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, https://doi.org/10.1175/BAMS-D-18-0195.1.

——, I. Ebert-Uphoff, D. J. Gagne, and A. Bostrom, 2022: Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environ. Data Sci.*, **1**, e6, https://doi.org/10.1017/eds.2022.5.

Merritt, R. B., 2021: Visualizing planetary Rossby waves with topological data analysis. M.S. thesis, Dept. of Statistics, University of Georgia, 53 pp., https://esploro.libs.uga.edu/esploro/outputs/9949375354302959.

Mileyko, Y., S. Mukherjee, and J. Harer, 2011: Probability measures on the space of persistence diagrams. *Inverse Probl.*, **27**, 124007, https://doi.org/10.1088/0266-5611/27/12/124007.

Moroni, D., and M. A. Pascali, 2021: Learning topology: Bridging computational topology and machine learning. *Pattern Recognit. Image Anal.*, **31**, 443–453, https://doi.org/10.1134/S1054661821030184.

Muszynski, G., K. Kashinath, V. Kurlin, M. Wehner, and Prabhat, 2019: Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets. *Geosci. Model Dev.*, **12**, 613–628, https://doi.org/10.5194/gmd-12-613-2019.

Ofori-Boateng, D., H. Lee, K. M. Gorski, M. J. Garay, and Y. R. Gel, 2021: Application of topological data analysis to multi-resolution matching of aerosol optical depth maps. *Front. Environ. Sci.*, **9**, 684716, https://doi.org/10.3389/fenvs.2021.684716.

Rasp, S., M. S. Pritchard, and P. Gentine, 2018: Deep learning to represent subgrid processes in climate models. *Proc. Natl. Acad. Sci. USA*, **115**, 9684–9689, https://doi.org/10.1073/pnas.1810286115.

——, H. Schulz, S. Bony, and B. Stevens, 2020: Combining crowdsourcing and deep learning to explore the mesoscale organization of shallow convection. *Bull. Amer. Meteor. Soc.*, **101**, E1980–E1995, https://doi.org/10.1175/BAMS-D-19-0324.1.

Reininghaus, J., S. Huber, U. Bauer, and R. Kwitt, 2015: A stable multi-scale kernel for topological machine learning. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, Institute of Electrical and Electronics Engineers, 4741–4748, https://doi.org/10.1109/CVPR.2015.7299106.

RIVET Developers, 2020: RIVET. RIVET Developers, https://github.com/rivetTDA/rivet/.

Rudin, C., 2019: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, **1**, 206–215, https://doi.org/10.1038/s42256-019-0048-x.

Schmit, T. J., P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman, and W. J. Lebair, 2017: A closer look at the ABI on the GOES-R series. *Bull. Amer. Meteor. Soc.*, **98**, 681–698, https://doi.org/10.1175/BAMS-D-15-00230.1.

Schultz, M., C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. Leufen, A. Mozaffari, and S. Stadtler, 2021: Can deep learning beat numerical weather prediction? *Philos. Trans. Roy. Soc.*, **A379**, 20200097, https://doi.org/10.1098/rsta.2020.0097.

Schwartz, R., J. Dodge, N. A. Smith, and O. Etzioni, 2020: Green AI. *Commun. ACM*, **63**, 54–63, https://doi.org/10.1145/3381831.

Segovia-Dominguez, I., Z. Zhen, R. Wagh, H. Lee, and Y. R. Gel, 2021: TLife-LSTM: Forecasting future COVID-19 progression with topological signatures of atmospheric conditions. *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, Vol. 12712, Springer, 201–212.

Sena, C. Á. P., J. A. R. da Paixão, and J. R. de Almeida França, 2021: A topological data analysis approach for retrieving local climate zones patterns in satellite data. *Environ. Challenges*, **5**, 100359, https://doi.org/10.1016/j.envc.2021.100359.

Stevens, B., and Coauthors, 2020: Sugar, gravel, fish and flowers: Mesoscale cloud patterns in the trade winds. *Quart. J. Roy. Meteor. Soc.*, **146**, 141–152, https://doi.org/10.1002/qj.3662.

Strommen, K., M. Chantry, J. Dorrington, and N. Otter, 2023: A topological perspective on weather regimes. *Climate Dyn.*, https://doi.org/10.1007/s00382-022-06395-x, in press.

Sun, H., W. Manchester, and Y. Chen, 2021: Improved and interpretable solar flare predictions with spatial and topological features of the polarity inversion line masked magnetograms. *Space Wea.*, **19**, e2021SW002837, https://doi.org/10.1029/2021SW002837.

Topaz, C. M., L. Ziegelmeier, and T. Halverson, 2015: Topological data analysis of biological aggregation models. *PLOS ONE*, **10**, e0126383, https://doi.org/10.1371/journal.pone.0126383.

Tymochko, S., E. Munch, J. Dunion, K. Corbosiero, and R. Torn, 2020: Using persistent homology to quantify a diurnal cycle in hurricanes. *Pattern Recognit. Lett.*, **133**, 137–143, https://doi.org/10.1016/j.patrec.2020.02.022.

Xu, J., W. Zhou, Z. Fu, H. Zhou, and L. Li, 2021: A survey on green deep learning. arXiv, 2111.05193v2, https://doi.org/10.48550/arXiv.2111.05193.

Yuval, J., and P. A. O'Gorman, 2020: Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nat. Commun.*, **11**, 3295, https://doi.org/10.1038/s41467-020-17142-3.

Zhu, X. X., D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, 2017: Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.*, **5**, 8–36, https://doi.org/10.1109/MGRS.2017.2762307.