

Let's Unleash the Network Judgment: A Self-Supervised Approach for Cloud Image Analysis

DARIO DEMATTIES^{a,b}, BHUPENDRA A. RAUT^{a,c}, SEONGHA PARK^{a,b}, ROBERT C. JACKSON^{a,c}, SEAN SHAHKARAMI^b, YONGHO KIM^{a,b}, RAJESH SANKARAN^{a,b}, PETE BECKMAN^{a,b}, SCOTT M. COLLIS^{a,c}, AND NICOLA FERRIER^{a,b}

^a *Northwestern Argonne Institute of Science and Engineering, Northwestern University, Evanston, Illinois*

^b *Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, Illinois*

^c *Environmental Sciences Division, Argonne National Laboratory, Lemont, Illinois*

(Manuscript received 1 September 2022, in final form 13 February 2023, accepted 23 February 2023)

ABSTRACT: Accurate cloud-type identification and coverage analysis are crucial in understanding Earth's radiative budget. Traditional computer vision methods rely on low-level visual features of clouds for estimating cloud coverage or sky conditions. Several handcrafted approaches have been proposed; however, scope for improvement still exists. Newer deep neural networks (DNNs) have demonstrated superior performance for cloud segmentation and categorization. These methods, however, need expert engineering intervention in the preprocessing steps—in the traditional methods—or human assistance in assigning cloud or clear-sky labels to a pixel for training DNNs. Such human mediation imposes considerable time and labor costs. We present the application of a new self-supervised learning approach to autonomously extract relevant features from sky images captured by ground-based cameras, for the classification and segmentation of clouds. We evaluate a joint embedding architecture that uses self-knowledge distillation plus regularization. We use two datasets to demonstrate the network's ability to classify and segment sky images—one with ~85 000 images collected from our ground-based camera and another with 400 labeled images from the WSISEG-Database. We find that this approach can discriminate full-sky images based on cloud coverage, diurnal variation, and cloud-base height. Furthermore, it semantically segments the cloud areas without labels. The approach shows competitive performance in all tested tasks, suggesting a new alternative for cloud characterization.

SIGNIFICANCE STATEMENT: Cloud macrophysical properties such as cloud-base height and coverage determine the amount of incoming radiation, mostly solar, and outgoing radiation, partly reflected from the sun and partly emitted from the Earth system, including the atmosphere. When this radiative budget is out of balance, it can affect our climate. Reporting sky conditions or cloud coverage from ground-based sky-imaging equipment is crucial in understanding Earth's radiative budget. We present the application of a novel artificial intelligence approach to autonomously extract relevant features from sky images, for the characterization of atmospheric conditions. Unlike previous strategies, this novel approach requires reduced human intervention, suggesting a new path for cloud characterization.

KEYWORDS: Cloud cover; In situ atmospheric observations; Machine learning; Other artificial intelligence/machine learning

1. Introduction

Clouds are among the most dominant factors driving the global climate system (Boucher et al. 2013). They affect the energy balance of Earth's atmosphere, by absorption, reflection, and scattering of the incoming shortwave solar radiation and by absorbing and emitting outgoing longwave radiation. Different types of clouds are the manifestation of different atmospheric processes. Reporting the sky conditions, cloud type, and cloud cover from the ground is a conventional practice in the United States that is performed by Automated Surface Observing System (ASOS) (NOAA 1998). However, ASOS identifies cloud cover by looking at the temporal distribution of ceilometer measurements over an hour time period, which provides incomplete information about cloud cover and type, especially when multiple cloud layers are present. When skies are heterogeneous, errors result from the limited

areal coverage. Even bigger errors are caused by the limited vertical range. For instance, in conditions with a clear sky or few clouds, the instrument cannot detect the presence of upper-level clouds (Wagner and Kleiss 2016). Human observers have used laser-beam ceilometers for years to measure the height of clouds. Nevertheless, visual estimates are still needed to determine the amount of cloud. The challenge of automating the data from such sensors is not only to process the height accurately but also to provide a representative description of the amount of cloud coverage (NOAA 1998). Since the atmosphere is in motion, to get a representative observation similar to what a human observer delivers, one needs to process the ceilometer signal over a 30-min time period. To be sensitive to the latest changes in sky conditions, the most recent 10 min of the data have to be processed twice (double weighted) (NOAA 1998). Ceilometer observations are recorded only if the cloud base is 3660 m (12 kft) above ground level (AGL) or less. As a result, cirrus clouds are systematically ignored despite their significant impact on the radiative budget (Liou 1986; Wagner and Kleiss 2016).

Corresponding author: Dario Dematties, ddematties@anl.gov

DOI: 10.1175/AIES-D-22-0063.1 e220063

© 2023 American Meteorological Society. For information regarding reuse of this content and general copyright information, consult the [AMS Copyright Policy](#) (www.ametsoc.org/PUBSReuseLicenses).

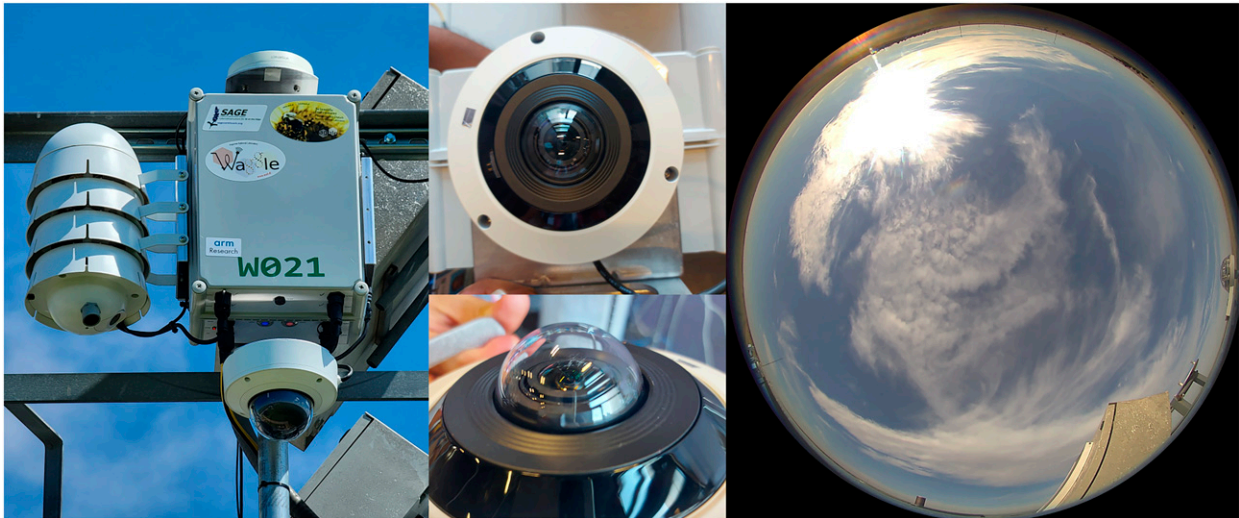


FIG. 1. Data collection device. (left) Sage node that includes cameras, microphones, and weather and air quality stations. (middle) Fish-eye sky-facing (top) camera. (right) A full-sky view high-resolution image.

Sky-facing cameras provide a much more complete view of the sky and cloud cover (Fig. 1).¹ Automated cloud cover detection is not routinely done from these cameras on an operational basis. However, much progress has been made in automating cloud classification from sky-facing cameras using traditional statistical, image-processing, and classical machine learning methods, such as edge and texture detection, naive Bayes, and k -nearest neighbor (KNN), and the recent breakthroughs in deep learning (DL) with convolutional neural networks (CNNs) (Wacker et al. 2015; Calbó et al. 2001; Zhang et al. 2018). The foundations of these methodologies rely on mimicking expert human observers either by using the specific and explicit choices of the feature-detection algorithms and thresholds, in the classical methods, or by using a large labeled dataset for supervised machine learning.

Regarding classical methods, cloud cover is commonly estimated from the images by using a ratio of red to blue channels with either a fixed threshold or varying thresholds depending on the lighting conditions or sector of the sky (Long et al. 2006; Huo and Lu 2009; Yang et al. 2012). A variable red-green difference feature using red and green channels was also proposed by Li et al. (2019) to tackle cloud detection for ground-based sky images. Image-processing algorithms are used to consider the visual appearance of clouds, such as the sharpness of cloud edges, texture, fibrousness, and size (Peura et al. 1996; Liu et al. 2011). For instance, many traditional image-processing methods utilize color to pinpoint clouds from the clear sky. Some of the methods establish fixed or variable thresholds in the ratio between blue and red channels (Kreuter et al. 2009; Long et al. 2006; Heinele et al. 2010; Souza-Echer et al. 2006). Combinations of statistical and machine learning (ML) models are also used for cloud cover

and cloud-type estimation (Tian et al. 1999; Calbó et al. 2001; Zhuo et al. 2014). KNN classifiers combine several features of images (i.e., the statistical color and texture, solar zenith angle, cloud coverage, the visible fraction of solar disk, radiative features, and the existence of raindrops in the images) to produce cloud cover and cloud-type classification from sky cameras (Wacker et al. 2015; Kazantzidis et al. 2012). Yet classical machine learning methods need human intervention for obtaining relevant features before classification. DL methods, on the other hand, automate the process of feature extraction by applying end-to-end optimization, typically through back-propagating errors. The features subsequently extracted by the network are not determined by using human engineering expertise because they are blindly fine-tuned by a loss function applied at the top of the system—that is, no human intervention is in the pipeline (LeCun et al. 2015).

Because of the demands of solar energy farms for more efficient and accurate estimation of cloud cover, many new DL approaches have been developed (Wang et al. 2020a; Xie et al. 2020; Park et al. 2021). For instance, Onishi and Sugiyama (2017) proposed a deep CNN approach for the estimation of cloud coverage. Additionally, Zhang et al. (2018) proposed a new CNN model for ground-based meteorological cloud classification. DL has also been used for cloud classification (Kurihana et al. 2019a) and cloud segmentation (Xie et al. 2020).

Currently, DL methods are divided into two main categories: *supervised* and *unsupervised* learning. Most CNN-based cloud detection and segmentation approaches are built on the supervised learning method, which requires a large number of pixelwise labels; this labeling of images is time consuming and labor intensive.

We note that supervised learning could bias the algorithms, limiting the capacity of the networks to learn more subtle features hidden in the nuances of input data (Cabrera et al. 2014; Sun et al. 2020; Karimi et al. 2020). Supervised learning restricts the network to predicting only oversimplified labels, such as cloud, sky, or sun, assigned by humans (Xie et al. 2020).

¹ Sage Cyberinfrastructure for AI at the Edge (<https://sagecontinuum.org/>).

In contrast, unsupervised learning needs only the input images to be properly trained (Guo et al. 2021). For example, an autoencoder method can extract cloud features from multiple satellite images and classify the features using hierarchical agglomerative clustering (HAC) without the use of labels (Kurihana et al. 2019b, 2021).

CNNs are the de facto choice in this regard. They use the typical encoder/decoder scheme and can be applied as an unsupervised strategy through a deep convolutional autoencoder (Kurihana et al. 2019a) or through a supervised strategy training the network to classify each pixel in the input image (U-Net) (Xie et al. 2020).

While deep neural networks generally require large training datasets, research has shown that U-Nets can be trained with a smaller dataset and still achieve comparable performance—in some cases with as few as 30 pairs of images. For tasks with small training datasets available, data augmentation is used on the training images. This allows the networks to learn invariant representations to such augmentations, without the need to see these transformations in the annotated image corpus (Ronneberger et al. 2015).

Several aspects in all the cited approaches can be improved. First, the task of the autoencoders is to reconstruct the input image as reliably as possible. Therefore, an autoencoder has to learn to capture as much information as possible rather than as much *relevant* information as possible. This means that if the most relevant information for a particular task is only a small fraction of the input, the autoencoder will possibly end up losing much of that information.

On the other hand, U-Net tends to present redundancy because the paired layers in the network extract similar-level features (Guo et al. 2020). As a response, Oktay et al. (2018) proposed *attention U-Net*, which implicitly learns to suppress irrelevant regions in an input image while highlighting salient features useful for improving the segmentation task.

In this work, we address all these issues. First, we address label scarcity by applying self-supervised learning (SSL) for cloud characterization. Second, instead of training the network to the heavy task of reconstructing the input as in autoencoders, we use a joint embedding architecture, applying acute augmentation strategies to gather the most relevant information behind the identity of clouds. Third, instead of integrating U-Net with attentional mechanisms, we directly use a transformer network, utilizing its attentional maps to generate self-supervised cloud segmentation.

An additional major motivation for this work is large-scale energy-efficient sensing of cloud conditions in regions that have limited network connectivity. It is hence important to be able to infer the conditions at the physical location, rather than transporting images to be processed elsewhere. Edge computing (Wang et al. 2020b; Chen and Ran 2019; Beckman et al. 2016; Cao et al. 2020) is a new processing paradigm that collocates powerful and efficient computing hardware and the sensors that provide the data. Therefore, edge computing is well suited for large-scale distributed cloud analysis, where pretrained ML models can be deployed at the edge for generating inferences. Unfortunately, supervised learning is not conducive to learning at the edge since the data collected by

edge devices have no labels. The self-supervised approach presented in this work can be trained on the images collected at the edge without having to transfer the bulky image data. The natural extension of this work is to deploy an application at the edge to learn in situ from the data collected.

2. Methods

a. Self-supervised learning

Self-supervised learning is a combination of supervised and unsupervised learning. It is like unsupervised learning in the sense that it does not need to use labels provided by human judgments in the dataset. On the other hand, it is like supervised learning in that it uses labels, but they are self-extracted from the dataset and not provided by humans. SSL has seen applications in cloud classification and characterization (Fabel et al. 2022; Kurihana et al. 2019a). These applications use either the U-Net architecture (Fabel et al. 2022) or an autoencoder (Kurihana et al. 2019a). In the case of U-Net for SSL, several pretext tasks are assigned to the learning process, such as filling cropped areas or increasing the image resolution (Fabel et al. 2022). In the autoencoder case, the pretext task is the reconstruction of the original image after reducing the dimensionality by encoding the input. The goal is to get the encoding to condense as much information as possible in order to reconstruct the input image (Fabel et al. 2022).

In the realm of computer vision (CV), SSL has shown tremendous advancements in reducing the amount of labeled data required to achieve state-of-the-art performance. In this area, we can find works such as a simple framework for contrastive learning (SimCLR; Chen et al. 2020b), swapping assignments between multiple views of the same image (SwAV; Caron et al. 2021a), bootstrap your own latent (BYOL; Grill et al. 2020), simple Siamese network (SimSiam; Chen and He 2020), and Distillation No labels (DINO; Caron et al. 2021b). The common approach in these methods is to present differently augmented versions of the same image to different branches of a joint embedding architecture. In some cases, such branches could share weights while others could just share the architecture, depending on the approach. Different architectures or even modalities could be used for different branches, too (Bardes et al. 2022). The main point in all these approaches is to train the networks to be invariant to the augmentations presented at the input by teaching them to ignore such augmentations. The hope is that by following such an invariance goal, the networks can acquire important features from the statistical structure of the data.

b. Data

1) SAGE CAMERA

A Sage node (Fig. 1), deployed at the Atmospheric Radiation Measurement (ARM) user facility's Southern Great Plains (SGP) atmospheric observatory (36.7°N, 97.5°W), hosts a sky-facing XNF-8010RV X series camera manufactured by Hanwha Techwin America. Using the 1.6-mm fish-eye lens for this study, the Sage node recorded 2048 × 2048 pixel full-color images every 30 s.

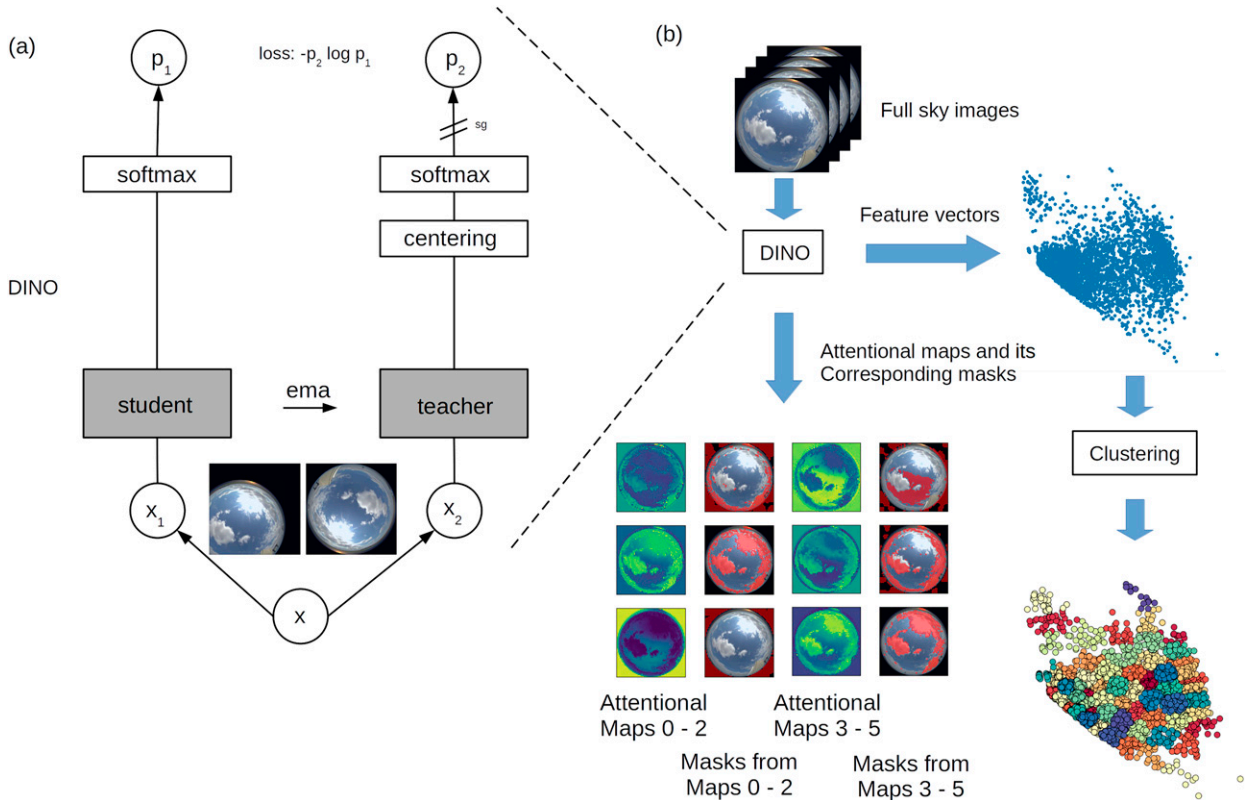


FIG. 2. Experimentation profile. (a) A joint embedding model architecture of DINO by Caron et al. (2021b). (b) Once trained, the feature vectors returned by DINO's inference process are clustered using different techniques, such as SOMs and k -means. Because DINO distributes feature vectors corresponding to different image properties in different zones of the outputs, different clusters will represent images with different visual properties. We use the attentional maps from the ViT in DINO to generate unsupervised segmentation of clouds.

2) CEILOMETER

Measurements of cloud-base height (CBH) from a collocated ceilometer (less than 300 m from the camera) over the SGP C1 site were available up to a 7.7-km altitude during October–December 2021 (Morris 2016). The ceilometer is an autonomous, ground-based active remote sensing instrument that detects multilayer clouds by transmitting near-infrared pulses of light and receiving the backscattered signal from cloud droplets that reflect a portion of the energy toward the ground (Morris 2016). The ceilometer uses multiple overlapping pulses to provide CBH estimations every 16 s. The CBH for the clear-sky condition is reported when the backscattered signal is not detected, because a low signal-to-noise ratio produces a missing signal for the sensor. During the study period, 74.4% of the readings corresponded to clear sky, 22.5% of the readings detected one cloud base, and 2.9% detected two cloud bases. In total, 25.6% of the readings detected cloud bases and passed all the quality checks.

c. Model architecture

As is the standard trend in the SSL for CV, we adopted a joint embedding architecture. A joint embedding is composed of two branches containing one network each. Such networks

can share their weights in the branches (Chen et al. 2020b), or they may have different weights but use the same architecture (Caron et al. 2021b). Alternatively, the branches can contain completely different architectures, even processing different modalities (Bardes et al. 2022).

As shown in Fig. 2a, the networks contrast different augmented versions of the same image (x_1 and x_2). In this work, we use the model developed by Caron et al. (2021b). We are particularly interested in the architecture because of its simplicity and its capacity to cluster images of different characteristics in different isles in the output feature space. The use of transformers as the main processing element in the branches allows us to use its attentional maps to generate masks for cloud segmentation. Another main feature of the Caron et al. (2021b) model is its capacity for generating semantic segmentation of images without labels (Fig. 2b).

As can be seen in Fig. 2, the main strategy presents two branches with two networks using the same architecture [a vision transformer (ViT)]. One is called the *student*, while the other is the *teacher*. Both networks are randomly initialized, but only the student backpropagates its errors in an attempt to mimic the output activity of the teacher. The student tries to follow the frozen teacher adjusting its weights progressively while the teacher is just making inferences thanks to its stop

gradient (sg). Occasionally, the teacher updates its weights using the student weights, incorporating them into its architecture by an exponential moving average (EMA).

The aim of the task—imposed by the loss function—is to produce outputs as similar as possible in both branches, disregarding the augmentations present at the input. The idea is that the network learns to ignore changes in color, brightness, size, position, and so on imprinted in the augmented versions and acquires the main features of the original images to invariantly recognize the objects present in them.

Even though the reasoning behind this mechanism is appealing, inherent problems arise when training networks to use it. Collapse is a recurrent phenomenon in this kind of learning paradigm. To satisfy the demands imposed by the loss function, the networks end up ignoring their inputs, just producing an identical constant output in both branches. Under such conditions, the outputs perfectly satisfy the loss demands, but they do not carry any information about the inputs. This is a catastrophe from the perspective of the main goals of the analysis of the data.

One of the main elements that distinguish one method from others inside the SSL approaches is the strategy used to avoid collapse. They can use contrastive loss (Chen et al. 2020b), clustering constraints (Caron et al. 2021a), stop gradient (Grill et al. 2020; Chen and He 2020; Caron et al. 2021b), or different batch normalization techniques (Bardes et al. 2022). We used stop gradient plus centering and sharpening. Centering and sharpening are applied at the output of the teacher network Caron et al. (2021b). While centering prevents one dimension from dominating and encourages collapse to the uniform distribution, sharpening has the opposite effect. Centering, sharpening, and EMA help the networks avoid collapse in the learning process.

d. Training procedure

We trained the joint embedding using $\sim 85\,000$ sky images of size 2048×2048 pixels from October to December 2021. During training, the augmentations received by the teacher and student were composed of cropped versions of the original images. There were crops plus resized, Gaussian blur, solarization, flip, and color jitter. All augmentations were applied randomly. There were two different kinds of *random crops and resizes*: global and local crops. Global crops spanned from 40% to 100% of the original images, while local crops spanned from 5% to 40% of the original images. The teacher received only global crops, while the student processed all (global and local) crops. For each sample, we used 2 global crops plus 8 local crops. The idea is that by processing local crops, the student can grasp the semantic information of the objects in the scene while mimicking the teacher that processes only global information. We also used a small ViT with ~ 21 million parameters and a patch size of 16 pixels.

The network was trained for 200 epochs, during which the loss was reduced from 11.1 in epoch number 0 to 2.20 in epoch number 199. A copy of the checkpoint was saved every 20 epochs. Through a visual inspection of the attentional maps returned by the network, we discovered that even though the

loss was reduced epoch after epoch, the semantic maps described the clouds better for the first epochs than for the last ones. Consequently, we chose the checkpoint saved after epoch 0 for running our experiments.

This counterintuitive phenomenon of progressive loss reduction accompanied by less expressive attentional maps could have its origins in the adequacy of the pretext augmentation for the dataset applied in this research. In fact, the augmentations used to train the model here were similar to the ones used to train the original model on ImageNet. Nevertheless, these two datasets have completely different statistical distributions. Despite its evident importance, the exploration of better pretext tasks for the characterization of full-sky atmospheric conditions is outside the scope of this research.

e. Attention/feature maps

Transformers are the state of the art in most natural language processing (NLP) benchmarks (Devlin et al. 2019; Radford et al. 2019; Brown et al. 2020). In recent years, they have also become prominent in CV applications (Carion et al. 2020; Chen et al. 2020a; Dosovitskiy et al. 2020).

In transformers, attentional maps are matrices that indicate the level of influence that different parts of the input have on a portion of the input under analysis. For instance, in NLP, the model analyzes each word in a sentence by paying attention to the most influential words—for the word under analysis—in the same sentence. In CV, the input image is divided into nonoverlapping patches, and the analysis of each patch consists of paying attention to the surrounding patches in the same image, which could be more influential for predicting certain features in the patch under analysis. This architectural aspect of transformers allows us to conduct quantitative but also qualitative visual inspections of the behavior of the network.

Figure 2b shows attentional maps and their corresponding masks. The green squares are the attentional maps inside the network. Our transformer configuration has 6 heads. We take each attentional map as a probability and consider only a certain probability to compose the masks. The different heads pay attention to different aspects of the images. For instance, head 0 highlights clouds and the dark periphery of the image ignoring the sun, head 1 strongly highlights the sun and clouds, and head 2 highlights the dark border of the image and some portions of the sky. Head 3 highlights the border and the sky. Head 4 is similar to head 3, accurately ignoring the fish-eye perimeter. Head 5 highlights the sun and clouds, markedly ignoring the dark periphery. Thus, the mask can be produced for cloud, sun, and clear sky by using these attentional maps.

3. Results and discussion

a. Unlabeled images

1) OUTPUT FEATURE SPACE

In Fig. 3, we present a visualization of the behavior of the output feature vectors returned by the model for different input images. As can be seen in the figure, from simple pretext

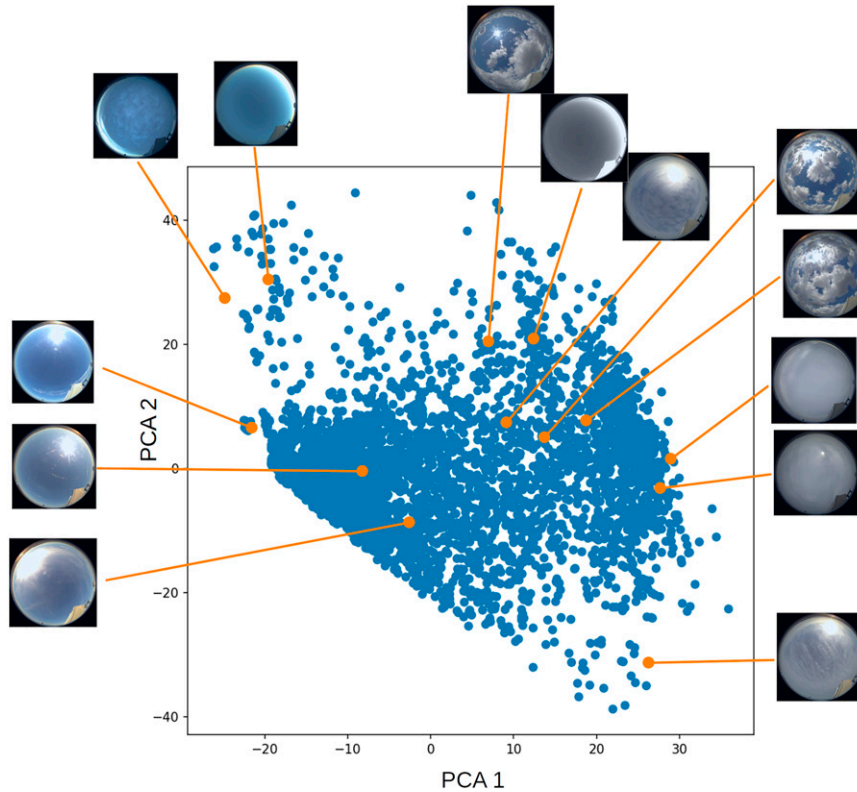


FIG. 3. Output feature map from DINO's inference process. For visualization purposes, we utilized PCA to bring the 384 dimensions of the output feature vectors from the model to the two principal components and plotted a subsample with 5000 points in this chart. From simple pretext tasks, the network is able to acquire feature vectors organized in the space by different properties.

tasks (like the ones explained in sections 2c,d) the model produces output vectors whose zones are highly correlated with the appearance of the full-sky images.

The model distributes images with different properties in different zones of the output feature space. For instance, on the left of the space, we can mostly see clear-sky images, while on the right, overcast sky images are prevalent. As we move up on the left of the space, the atmospheric conditions produce images with increased blueness. The limits of such a phenomenon can be found at the top of the space, where images are mostly crepuscular—darker blue—as can be deduced from the figure. As we move to the center of the space, clouds start to appear gradually in the sky. In the center are several zones with different properties. Some zones return sky images with clustered clouds, while others show altostratus clouds.² In the middle is a cluster that contains mostly the night-vision sky of our camera, as can be seen in the image. On the right side of the map, we can see mostly overcast skies. At the bottom, we have a delimited cluster of points with different altostratus clouds.

The training process of the network in itself promotes its invariance to certain modifications or different points of view of

the images. As can be seen in Fig. 4, even when the model is highly sensitive to certain aspects affecting the appearance of the sky images—for instance, clear-sky images on the left and overcast skies on the right—it is highly invariant to other aspects of the image appearance.

In Fig. 4, we can see how, independent of the rotation, position, or even the form of the fish-eye sky image, the location of the output vector corresponding to such an image in the feature space is stable. The figure shows that independent of the position of the sun and clouds in the scene, the model will end up categorizing each image by other aspects such as its cloud coverage and kind of clouds. This cloudy image represents a stable pattern for the model, and it is evidenced in the stable position of the output in the feature space regardless of the different modifications applied to the input.

2) CLUSTERING

We applied a clustering algorithm to the samples of the output feature space returned by the model. We processed the 384 component vectors from the model using principal component analysis (PCA) with maximum-likelihood estimation (MLE), which uses a Bayesian model selection to determine the true dimensionality of the data (Minka 2000). The dimensionality of the output was automatically reduced to 383

² Altostratus are large midlevel sheets of thin clouds.

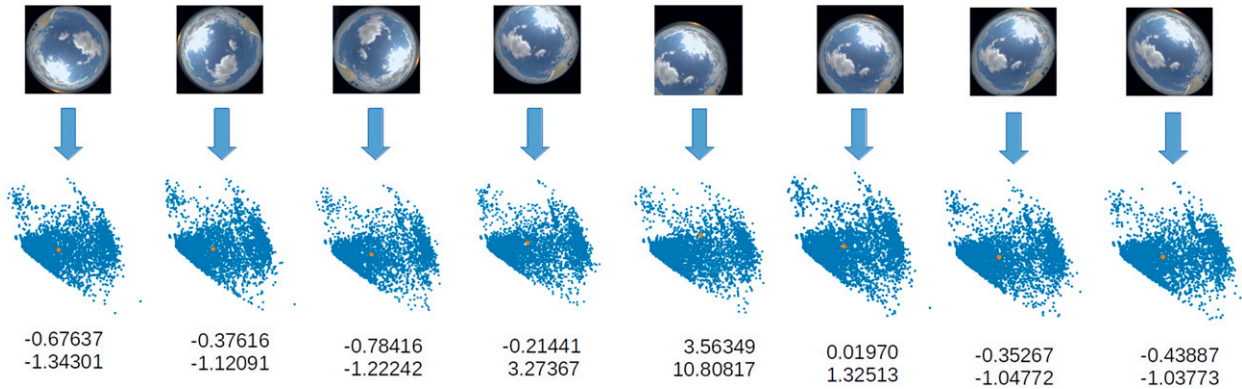


FIG. 4. Invariance of the algorithm to rotations, translations, and deformations of the same image. Each example is a specific augmentation from the same original image. For each augmentation example, there is an output feature space with the exact location of the output vector as an orange point. The coordinate values of the output vector are shown accordingly next to each example. Beyond differences in the rotations, translations, and deformations, the output vectors mostly keep the same position in the feature space.

components. We then pursued the clusterization of the samples. It was difficult to find appropriate parameters for density-based spatial clustering of applications with noise (DBSCAN) to get a sensible number of clusters; it turned out to be either one or more than 100 clusters, which did not represent the underlying atmospheric conditions in the dataset. After several experimental repeats conducting clusterization and visual inspection of the atmospheric conditions of the sky images in the different clusters, we concluded that 20 clusters had the best representation of the different atmospheric conditions. In this regard, the self-organizing map (SOM) presented a more acute division of the clusters in correlation with the visual appearance of sky images than did *k*-means. This conclusion was guided by a global visual inspection of the images in the different clusters as well. We validated our selection with additional quantitative experimentation (Figs. 6–8).

We used the new space and trained a two-dimensional SOM grid of 5 nodes \times 4 nodes. We chose this dimensionality because it has a better distribution of nodes in a two-dimensional grid than does the alternative combination, such as 10 \times 2. Then we classified each image in the dataset by applying the KNN algorithm using the prediction inference from the SOM nodes. In Fig. 5, we can see a set of 20 fish-eye full-sky images organized in a grid structure of 5 rows and 4 columns. Each image is a representative sample extracted from 20 clusters obtained by applying SOM clustering to the output feature vector returned by the model.

As can be seen in this figure, the properties of the output feature map in Fig. 3 are reflected by the grid. First, in row 4, we can see completely overcast images, while in row 0, we see mostly clear-sky ones. Again, the blueness of the sky increases as we move from D to A in row 0. Row 0 column A holds the *crepuscular* images also shown in Fig. 3 at the top of the map. As in Fig. 3, we have a smooth transition from a clear sky in row 0 to an overcast sky in row 4. In the middle rows, we observe different partially cloudy images.

In Figs. 6–8, we present a statistical analysis of the clusters shown in Fig. 5. In the analysis, we support the visual inspection

conducted over the images gathered by the different clusters in the SOM shown in Fig. 5.

First, we processed the same dataset using an already trained U-Net (Park et al. 2021). We used the segmentation prediction from U-Net and computed the cloud coverage for each image in the dataset.

In Fig. 6 we can see the mean cloud coverage values for each cluster in Fig. 5. Figure 6 presents a coordinate system

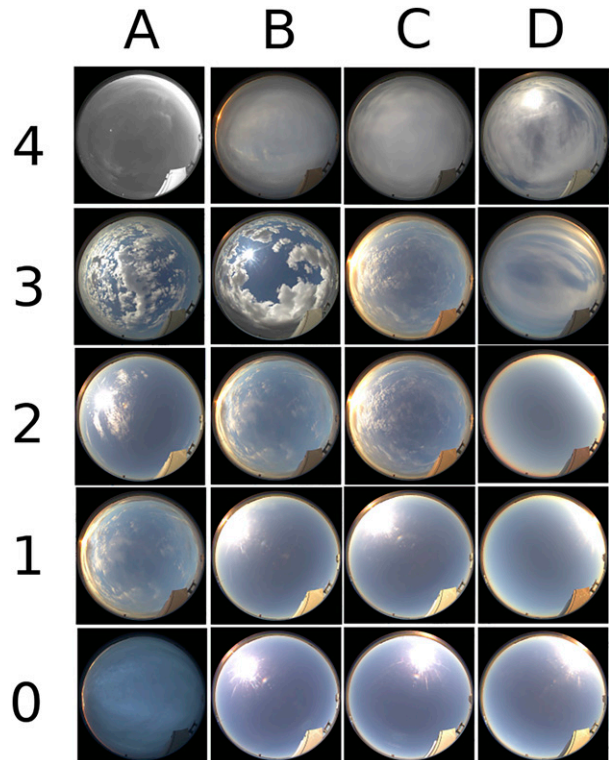


FIG. 5. A 5 \times 4 grid from a SOM of 20 nodes when using dimensionality reduction by PCA with MLE to 383 dimensions. Each sky image is a representative sample from each cluster in the SOM.

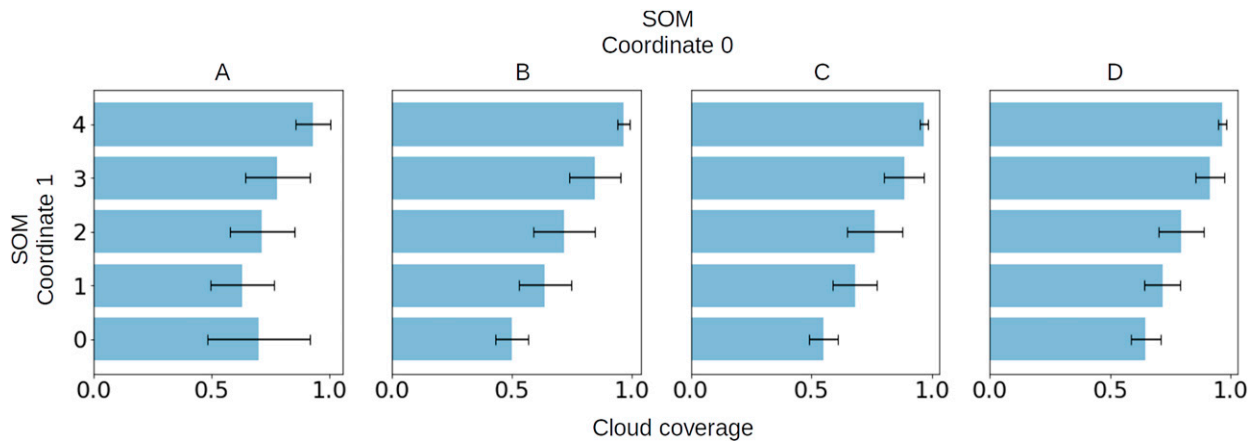


FIG. 6. Mean cloud cover and standard deviation for each SOM cluster using U-Net.

coherent with Fig. 5. Again, we have a 5×4 grid in which each bar represents the mean cloud coverage computed by U-Net in such a cluster together with the corresponding standard deviation (STD) in Fig. 5. The bottom horizontal axis in Fig. 6 shows the cloud coverage. Each bar in the chart corresponds to a different cluster in the SOM. Coordinate 1 in the SOM is the vertical axis. Coordinate 2 in the SOM is the horizontal axis. One of the SOM grid coordinates shows a clear orientation related to the mean cloud coverage returned by U-Net. U-Net agrees with the cloud coverage percentage presented by Fig. 5, where a smoothly increasing cloud coverage is produced as we move from row 0 to row 4 in both figures.

Consequently, one of the dimensions in the SOM is aligned with cloud coverage. This is the vertical axis in Fig. 5. Since in Fig. 6 we show only information related to the mean cloud coverage computed by U-Net, we have no way of knowing what could be indicating the other dimension—the horizontal one—in the SOM grid. It is also hard to judge from the images in the grid in Fig. 5 what the horizontal dimension is indicating. Furthermore, there might be additional features along the vertical axis beyond cloud coverage. Indeed, our representative images in the clusters in Fig. 5 show great variability for the different clusters, which might well be indicating different atmospheric conditions.

To address this issue, in Fig. 7 we show the diurnal hourly frequency of the sky images collected by different nodes in the SOM. In this figure, we can observe that, except for D4, column D is highly influenced by dawn images of the sky.

From Fig. 7, A0 is a node that concentrates mostly dawn or dusk images. A visual inspection of such a node allows us to observe that this effectively captures most of the dark blue crepuscular images. They could be cloudy or clear sky, but most of them are dark blue crepuscular since they present sunrise or sunset situations (Byrd 2014).

A similar scenario is shown by node A4, which captures most of the night-vision images from our camera. Figure 5 shows a night-vision sky image fully covered by clouds in cluster A4, while Fig. 7 shows a sharp peak at 0700 local time. In those cases, A4 is collecting early-morning night-vision images from the camera.

A global inspection of Fig. 7 allows us to see that, beyond certain exceptions in each column, the model poses a *diurnal ordering* in the horizontal dimension of the SOM shown in Fig. 5. Generally, the tendency changes from morning to evening as we move from column D to column A. This effect is evident in Fig. 5, but it also can be seen in Fig. 7, where column D is highly populated with morning skies. Then, in column C, this tendency changes to mostly afternoon skies but with a more uniform diurnal distribution. Column B reflects a more acute inclination toward evening, and in column A such an inclination is even more acute.

Such a diurnal variation represents a clear tendency in the model to sort sky images by the diurnal cycle. Yet there are remarkable cases in which the different kinds of clouds are playing a major role. In some cases, the atmospheric conditions given by different kinds of clouds are taken into account by the algorithm when distributing images in the hyperdimensional output space. In this regard, nodes A3 and B3 gather most of the complex configurations of cumulus clouds. Node A4 gathers most of the night-vision sky images, while node A0 has most of the very early or late (sunrise/sunset) crepuscular images (Byrd 2014). Likewise, node B0's main feature is that it has the darkest blue clear skies.

Figure 8 shows the distribution of CBH corresponding to the sky images shown in Fig. 5 using a collocated ceilometer. We sampled CBH measurements for periods of at least 10 min of consecutive occurrences of the SOM node. The frequency of sampled periods (f) and percentages missing CBH measurements (NA) for each node are shown at the top of Fig. 8. Missing measurements correspond to the clear-sky conditions and low signal-to-noise ratio.

As expected, clear-sky nodes are the ones with the most missing measurements. Nodes C0 and D0 are the ones with fewer cloud episodes, both with 99% of missing measurements. We also have nodes B0 with 97%, D2 with 95%, A1 with 90%, C1 with 88%, and C2 with 87% of missing measurements. These nodes concentrate most of the clear-sky images in the dataset.

In agreement with Fig. 6, there is a general tendency of clear-sky measurements to decrease as we move from 0 to 4

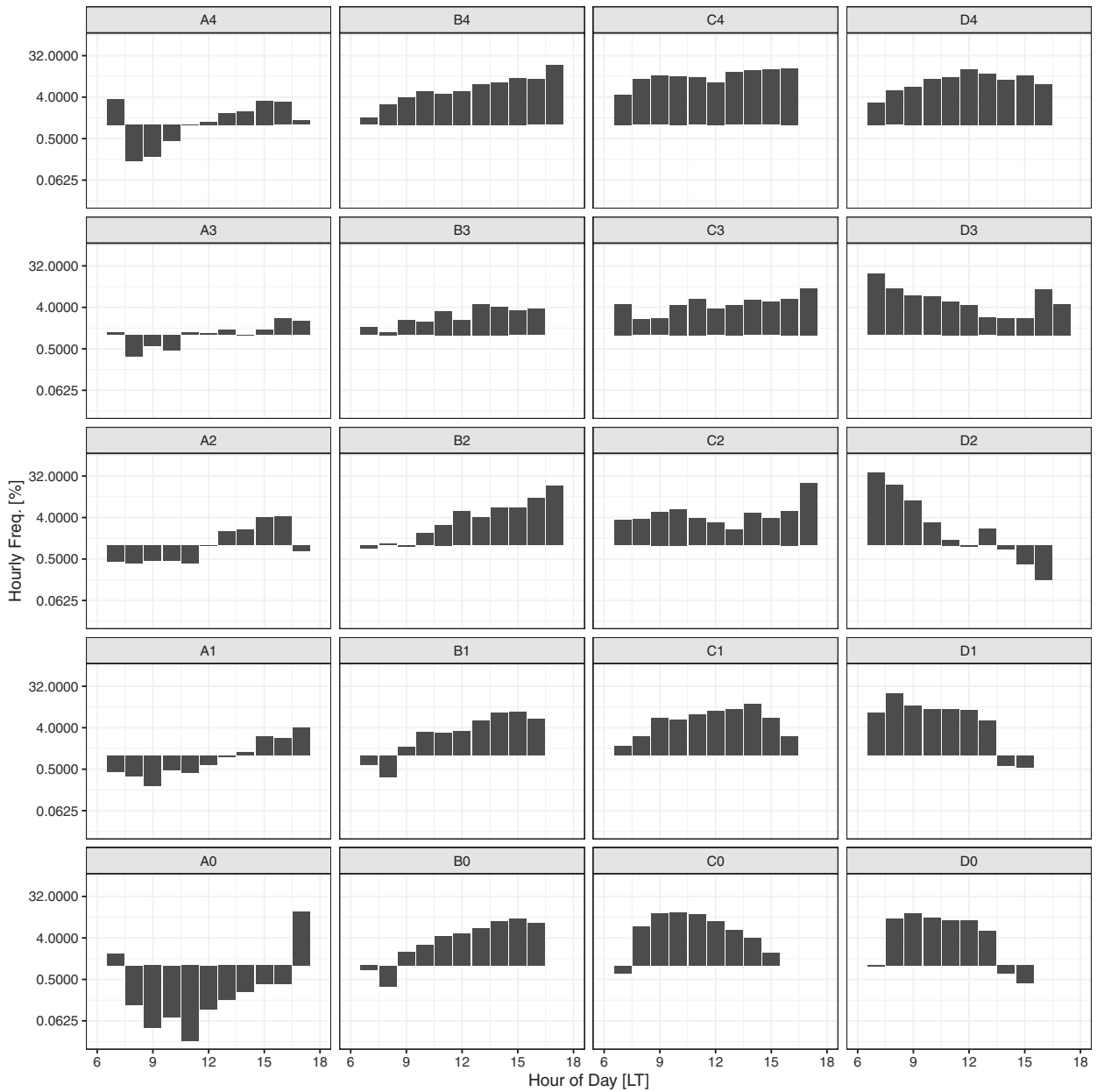


FIG. 7. Diurnal variation (0700–1700 LT) of the hourly cluster frequency shown in Fig. 5. The logarithmic y axis shows downward bars when the values are less than 1%.

in each column. Some exceptions exist, such as nodes A0, A3, and B3. Even when node A0 is located in row 0, which is mostly clear sky, it has a higher cloud concentration, populated mostly with early-morning and late-evening images. Most of the clouds found in this node are below 2.5 km. A3 and B3 present a lower percentage of cloud concentrations. Those nodes concentrate most of the cumulus-like clouds in B3 (2000 m) and altocumulus in A3 (>2500 m).

On the other side of the spectrum, taking into account the nodes with more cloudy events, we can see that for rows 3 and 4, there is a tendency of altostratus clouds to dominate the scene as we move from A to D. This can be seen in Fig. 8,

where altitudes above 5000 m dominate when we move from A to D. When there are enough cloud events to draw more accurate statistics, such as in row A, we can see a clear tendency along the horizontal dimension that shows a greater preponderance of higher-altitude (altostratus) clouds as we move from A to D. This is exhibiting a qualitatively different clustering property of the algorithm, which is highlighting cloud type, beyond cloud coverage or diurnal variation.

The dataset of full-sky images analyzed in this section was collected by using our equipment shown in Fig. 1. Such data do not have labels characterizing the different visual appearances highlighted by the network. Therefore, the representative



FIG. 8. Frequency distribution of lowest cloud-base height for the clusters shown in Fig. 5. The CBH measurements are sampled only for the periods when the cluster was present for 10 min or longer. On average, around 75% of the CBH are reported missing because of the clear-sky conditions and low signal-to-noise ratio. The valid CBH observations in each cluster (n) and the percentage of missing CBH measurements (NA) are given at the top of each panel.

features emerging in the analysis have been determined by visual inspection of the images.

b. Labeled dataset

1) CLUSTERING

We have tested and validated the model using approximately 85 000 images collected from a sky-facing camera. The model has been shown to be able to cluster different atmospheric conditions in different zones on the output feature space in an invariant manner (Fig. 3). The invariance of the model has been shown by conducting different modifications to the input images and observing that their output feature vectors map to similar spaces in the output feature map (Fig. 4). Nevertheless, the question of the robustness of the model properties for different images collected with different camera specifications remains open. Therefore, we conducted additional experiments on a different dataset. To that end, we used the WSISEG-Database,

which contains 400 uncropped whole-sky images and corresponding labels.³

We conducted inference on the 400 images, obtained a new output feature space, and clustered it using the k -means algorithm. By visual inspection, we picked 7 clusters as a good number given the characteristics of the output feature space. We distributed the images to different clusters and computed the mean cloud coverage for each cluster using the labels of the images. We noticed that with this number of clusters, k -means was able to distribute images in different clusters in coherence with the physical appearance of the sky images, better than when using other clusterization methods such as SOM and DBSCAN. This conclusion was reached by using visual inspection of all the images in the dataset. Basically, the cloud coverage is the percentage of pixels in an image labeled

³ In these annotation images, cloud, clear sky, and undefined areas are marked with gray values 255, 100, and 0, respectively.

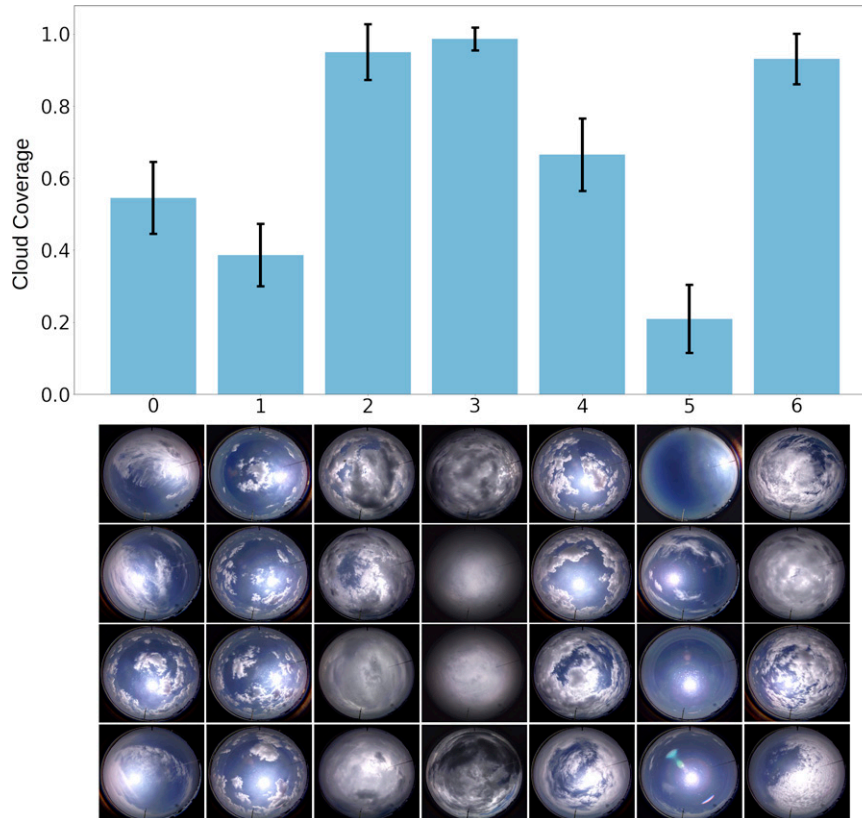


FIG. 9. Computation of mean cloud coverage and STD for 7 clusters obtained using dimensionality reduction by PCA with MLE (Minka 2000) to 383 dimensions and k -means clustering. Each column of images is composed of four random samples from the corresponding clusters (from 0 to 6). Cloud coverage is computed by using image labels. The dataset used, WSISEG-Database, includes 400 whole-sky images and manual-marked labels. Source at CV-Application (2021).

as cloud, relative to the sum of cloud plus clear-sky pixels. Consequently, the mean cloud coverage is the mean of the cloud coverage of all the images inside the same cluster.

Figure 9 shows how the algorithm distributed the images in the different clusters. The top bars show the mean cloud coverage for each cluster. In the bottom, we show a column with four random images collected for each cluster. The idea in the bottom array of images is to visually show atmospheric conditions for each cluster beyond the cloud coverage shown at the top in the bar plot.

We highlight two important aspects from Fig. 9. First, from the 7 clusters, clusters 0, 1, 4, and 5 clearly show different cloud coverage conditions. This situation can be seen not only in the bar plot but also in the example images at the bottom. For instance, cluster 0 concentrates a higher percentage of altostratus clouds with more extensive clouds than the ones in cluster 1. Clusters 1 and 4 hold mostly cumulus-like clouds; cluster 1 has more scattered cumulus, while cluster 4 is more concentrated and has larger cumulus and more cloud coverage. Cluster 5 is populated mostly with clear-sky images.

Second, clusters 2, 3, and 6 cannot be separated by just considering cloud coverage. Nevertheless, by visual inspection, one can see a clear differentiation between such clusters

based on different aspects of the clouds. For instance, in cluster 3, the presence of the sun is minimal. This cluster is characterized by very dark skies. On the other hand, in cluster 6, sunlight has the strongest presence in the scene. In the middle between those two extremes is cluster 2, where there is some sunlight.

One of the important aspects here is that the model was able not only to catch what human observers evaluated when labeling this dataset regarding the cloud coverage conditions, but also to make sense of other important aspects of atmospheric conditions such as the presence of sunlight in cloudy images. These aspects of the atmospheric conditions of clouds were never injected into the model as input information during training. Furthermore, these images were never used to train the model, and they have camera specifications different from those used for training the algorithm.

2) SEGMENTATION

We used the attentional maps from the network architecture to produce different kinds of segmentation from the 6 different heads of the network. In Fig. 10, we can see two examples from two heads in the two rows of the figure. On the left (first column), we have the attentional maps; in the second

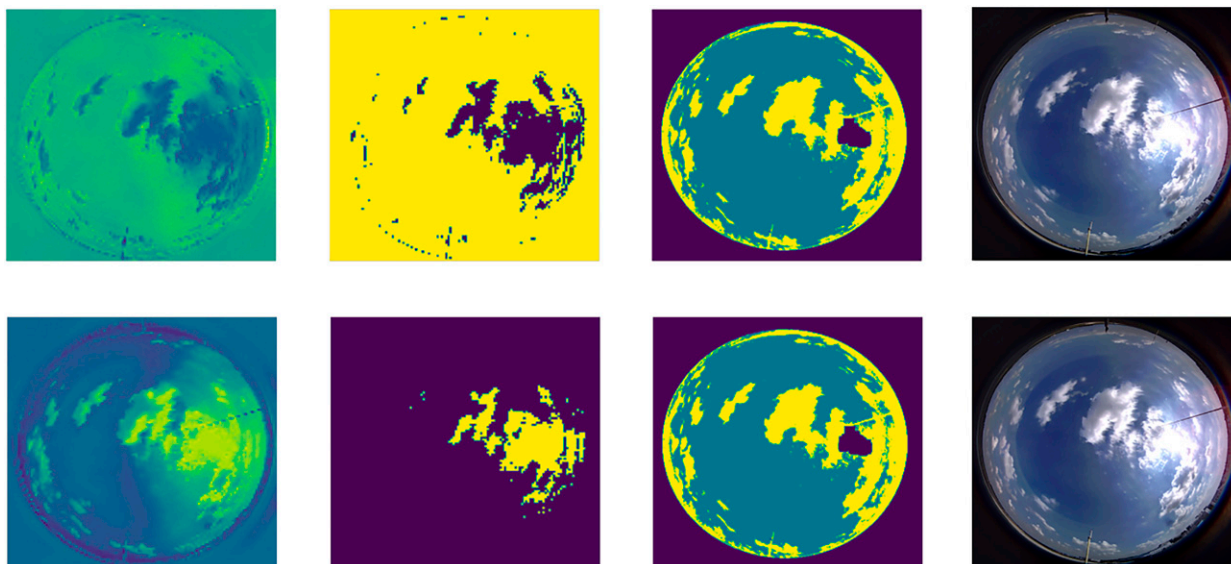


FIG. 10. Two examples of cloud segmentation produced by DINO in two rows. (first column) Attentional map, (second column) segmentation produced from the attentional map in the first column, and (third column) WSISEG-Database label. (fourth column) Original image.

column, we show the masks produced for segmentation from the attentional maps information; and in the third column, we show the labels produced by humans. On the right, we show the original image. To compute the masks, we convert the attentional map into a probability by normalizing it. Then we establish a threshold fixing a real number between 0 and 1. This number represents the percentage in the probability mass to be considered as an active mask. The rest is considered as void of mask.

In the first row of Fig. 10, we can see that the attentional map is highlighting everything except the clouds and sun. In this way and from such an attentional map, we can configure a mask that is the negative of the clouds as shown when comparing the mask with the label. In the second row, the situation is exactly the opposite. The attentional map is highlighting only the clouds and sun. The mask obtained shows a positive similarity with the label.

To analyze the characteristics of the masks produced by the model, we compared the masks with the labels produced by humans. The labels in the WSISEG-Database contain three classes: cloud, sky, and sun plus periphery, as can be seen in Fig. 10. We conducted the comparison by computing the total number of correctly predicted pixels (cloud plus sky) and divided that number by the total number of pixels (cloud plus sky).

We computed this average similarity for each cluster shown in Fig. 9, producing a sweep of thresholds to generate the masks from 10% to 90% of the probability mass in the attentional maps.

In Fig. 11 we show four different examples in four rows. For each row, on the left, we present a plot of bars showing the mean similarity of the segmentation produced by the masks for different values of the threshold (from 10% to 90%) computed on each cluster. An arrow on the horizontal

axis indicates the threshold for which the highest similarity is obtained.

On the right, we show an example from the same cluster with four images from right to left. The first image is the attentional map, the second is the mask generated from the attentional map using the most accurate threshold, the third is the label, and the fourth is the real image.

For example, in the first row of the figure we present on the left the performance of head 2 of the network for the images in cluster 0 in Fig. 9 (around 50% of cloud coverage, with a considerable percentage of altostratus clouds). We chose 40% as the best-performing threshold of the masks with a similarity above 60%. On the right, we have an example from cluster 0, whose similarity matching the label is about 75% using the optimal threshold.

In the rest of the rows of the same figure, we present other examples for other clusters in Fig. 9. In the second row, we have an example for cluster 1, which is mostly partially cloudy (<40%) with small cumulus-like clouds. Even though the similarity representing the label is 75%, the attentional map in head 3 is paying attention to more subtle aspects of the sky image beyond a restrictive cloud/no-cloud dichotomy. An important aspect to take into account is that head 3 is masking out not only the sun, but also all the clouds that are more affected by the sunlight. In the third row of the figure, a similar situation is shown. Again, sunlight and everything affected by sunlight are marginally ignored by the network.

The third and fourth rows show more extreme situations such as clear-sky images (cluster 5, <20% cloud coverage) and overcast images (cluster 6, ~90% cloud coverage). Similarity when matching the generated masks with the labels is above 80% in both cases.

In Fig. 11, for some segmentations the border part of the predicted mask (second column) is not always the same. In

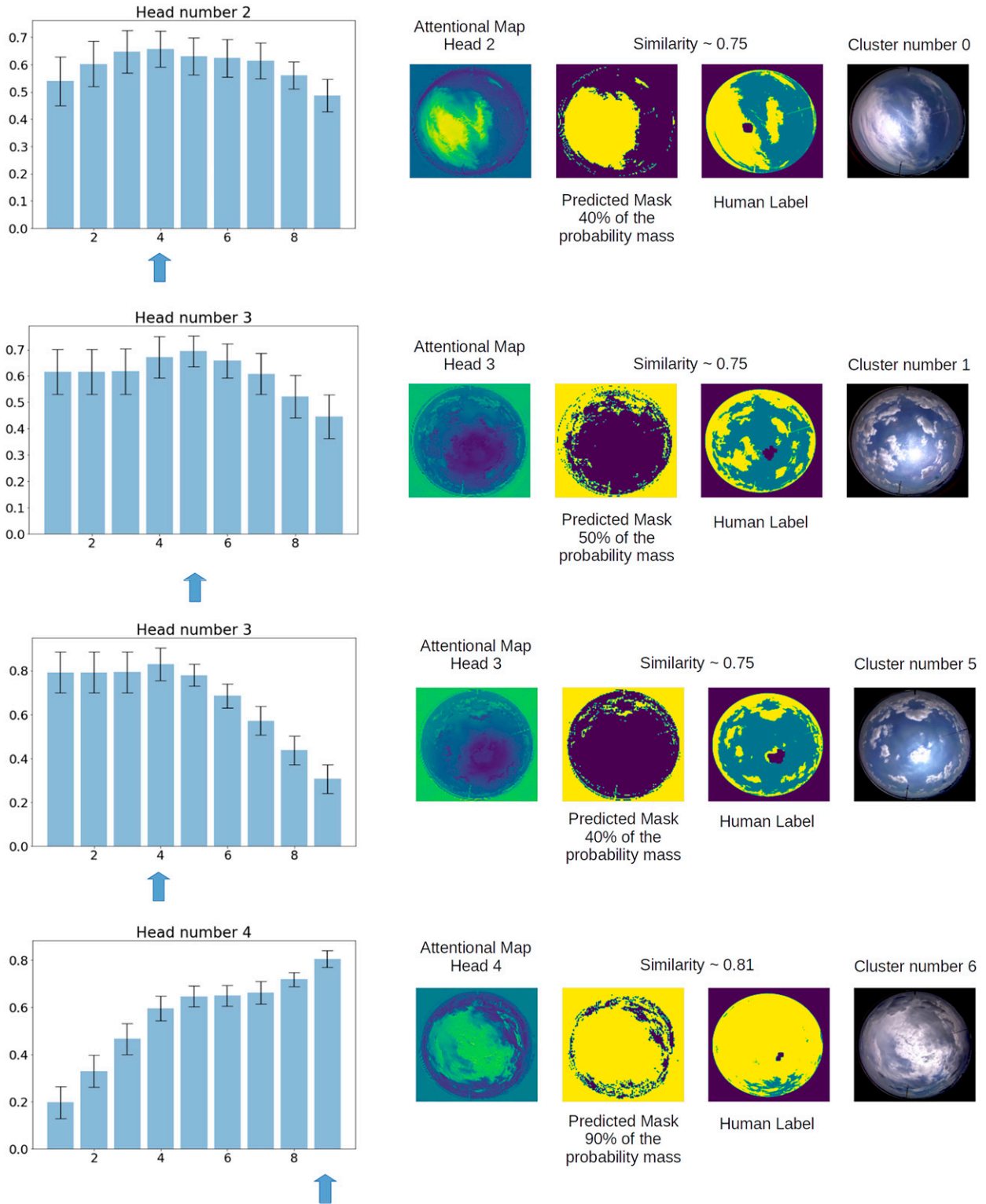


FIG. 11. Performance stats of the segmentation produced by the model accompanied by individual images as examples. (left) Segmentation similarity taking as reference the labels produced manually by humans. One head was chosen from the six heads of the model in each example. Vertical axes show the segmentation similarity; horizontal axes show the threshold level to compose the predicted mask (1 is 10%, 2 is 20%, ... 9 is 90%). (right) The first image is the corresponding attentional map of the head. The second image is the predicted mask computed using the threshold and the attentional map. The third image is the label produced by humans. The fourth image is the original sky image. Taking the first row of the figure as an example, a threshold of 40% to compute the predicted mask gets the best performance for all the images in cluster number 0. Such images present altostratus clouds, as can be seen in the example on the right. The segmentation computed using 40% of the probability mass in the attentional map has a 75% coincidence with the label produced manually by humans. The mean average for all the images in the cluster is around 65%, as shown by the bars on the left.

some cases, the border part is active, while in others it is inactive, reflecting the fact that the attentional values are above or below the imposed threshold, respectively. Nevertheless, the values do not affect the similarity computation since the pixels in such areas are not taken into account in the computations.

For reference, we note that Xie et al. (2020) reported a segmentation accuracy of 96.24% (on average on the same dataset). This value was obtained through a supervised approach that was validated by using only 60 images collected from the same camera. We highlight that even though the network proposed in our work never received those images during training, it was still able to semantically segment clouds from the sun and clear sky. This fact is even more remarkable when we take into account the fact that the images in this dataset were captured from different cameras with likely very different specifications. Furthermore, since the method used in this paper never used such images during training, we were able to test it on the complete dataset of 400 images.

4. Conclusions and future scope

In this paper, we proposed the use of a recently developed SSL method for sky condition monitoring. We demonstrated that the performance of a novel joint embedding architecture for the characterization and segmentation of ground-based all-sky-view red–green–blue (RGB) images is a viable strategy without the restrictive need for harvesting labels produced by humans. We showed that this architecture can acquire acute features of atmospheric conditions, such as cloud coverage, cloud altitude, and their diurnal distribution. The CBH information is also useful for the probabilistic height assignment (low, mid-, and high) to the cloud motion vectors from the camera images (Raut et al. 2023).

We also showed that the attentional maps of a pretrained network with our full-sky imagery equipment keep performance resemblance in completely new datasets captured with cameras with different specifications. In fact, with completely different images from a different camera, the algorithm shows a zero-shot learning performance and can cluster the new dataset for different atmospheric conditions, such as cloud coverage and sun cloud illumination. Furthermore, this zero-shot property is also demonstrated by the network showing competitive performance segmenting clouds.

These kinds of SSL approaches are in their nascent stage, but they hold the potential to reduce the requirement for human intervention, not only in meteorological observations, but also in the labeling process to train new automatic methods.

Acknowledgments. The Sage project is funded through the U.S. National Science Foundation’s Mid-Scale Research Infrastructure program, NSF-OAC-1935984. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. The U.S. Department of Energy Atmospheric Radiation Measurement (ARM) user facility supported the work under the field campaign AFC 07056 “ARMing the Edge: Demonstration of Edge Computing.” The

ceilometer data (<https://doi.org/10.5439/1181954>) were obtained from the ARM user facility, a U.S. DOE Office of Science user facility managed by the Biological and Environmental Research Program.

Data availability statement. The final release of the code to reproduce the experiments presented in this paper can be found at <https://zenodo.org/record/7391603> (Dematties 2022). All the data to reproduce the experiments are openly available at <https://zenodo.org/record/7032194> (Dematties and Sankaran 2022).

REFERENCES

- Bardes, A., J. Ponce, and Y. LeCun, 2022: VICReg: Variance-invariance-covariance regularization for self-supervised learning. arXiv, 2105.04906v3, <https://doi.org/10.48550/arXiv.2105.04906>.
- Beckman, P., R. Sankaran, C. Catlett, N. Ferrier, R. Jacob, and M. Papka, 2016: Waggle: An open sensor platform for edge computing. *2016 IEEE SENSORS*, Orlando, FL, Institute of Electrical and Electronics Engineers, 1–3, <https://doi.org/10.1109/ICSENS.2016.7808975>.
- Boucher, O., and Coauthors, 2013: Clouds and aerosols. *Climatic Change 2013: The Physical Science Basis*, T. F. Stocker et al., Eds., Cambridge University Press, 571–657.
- Brown, T. B., and Coauthors, 2020: Language models are few-shot learners. arXiv, 2005.14165v4, <https://doi.org/10.48550/arXiv.2005.14165>.
- Byrd, D., 2014: What is the blue hour? EarthSky, accessed, <https://earthsky.org/earth/what-is-the-blue-hour/>.
- Cabrera, G. F., C. J. Miller, and J. Schneider, 2014: Systematic labeling bias: De-biasing where everyone is wrong. *14th 22nd Int. Conf. on Pattern Recognition*, Stockholm, Sweden, Institute of Electrical and Electronics Engineers, 4417–4422, <https://doi.org/10.1109/ICPR.2014.756>.
- Calbó, J., J.-A. González, and D. Pagès, 2001: A method for sky-condition classification from ground-based solar radiation measurements. *J. Appl. Meteor.*, **40**, 2193–2199, [https://doi.org/10.1175/1520-0450\(2001\)040<2193:AMFSCC>2.0.CO;2](https://doi.org/10.1175/1520-0450(2001)040<2193:AMFSCC>2.0.CO;2).
- Cao, K., Y. Liu, G. Meng, and Q. Sun, 2020: An overview on edge computing research. *IEEE Access*, **8**, 85 714–85 728, <https://doi.org/10.1109/ACCESS.2020.2991734>.
- Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, 2020: End-to-end object detection with transformers. arXiv, 2005.12872v3, <https://doi.org/10.48550/arXiv.2005.12872>.
- Caron, M., I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, 2021a: Unsupervised learning of visual features by contrasting cluster assignments. arXiv, 2006.09882v5, <https://doi.org/10.48550/arXiv.2006.09882>.
- , H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, 2021b: Emerging properties in self-supervised vision transformers. arXiv, 2104.14294v2, <https://doi.org/10.48550/arXiv.2104.14294>.
- Chen, J., and X. Ran, 2019: Deep learning with edge computing: A review. *Proc. IEEE*, **107**, 1655–1674, <https://doi.org/10.1109/JPROC.2019.2921977>.
- Chen, M., A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, 2020a: Generative pretraining from pixels. *Proc. 37th Int. Conf. on Machine Learning*, Vol. 119, PMLR, 1691–1703, <https://proceedings.mlr.press/v119/chen20s.html>.

- Chen, T., S. Kornblith, M. Norouzi, and G. Hinton, 2020b: A simple framework for contrastive learning of visual representations. arXiv, 2002.05709v3, <https://doi.org/10.48550/arXiv.2002.05709>.
- Chen, X., and K. He, 2020: Exploring simple Siamese representation learning. arXiv, 2011.10566v1, <https://doi.org/10.48550/arXiv.2011.10566>.
- CV-Application, 2021: WSISEG-Database. Github, accessed 24 January 2019, <https://github.com/CV-Application/WSISEG-Database>.
- Dematties, D., 2022: Cloud image analysis using DINO: Reproducibility and experiments replication. Zenodo, accessed 2 December 2022, <https://doi.org/10.5281/zenodo.7391603>.
- , and R. Sankaran, 2022: SAGE sky images and model checkpoints for “Let’s Unleash the Network Judgement: A Self-supervised Approach for Cloud Image Analysis” paper. Zenodo, accessed 29 August 2022, <https://doi.org/10.5281/zenodo.7032194>.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova, 2019: BERT: Pre-training of deep bidirectional transformers for understanding. arXiv, 1810.04805v2, <https://doi.org/10.48550/arXiv.1810.04805>.
- Dosovitskiy, A., and Coauthors, 2020: An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv, 2010.11929v2, <https://doi.org/10.48550/ARXIV.2010.11929>.
- Fabel, Y., and Coauthors, 2022: Applying self-supervised learning for semantic cloud segmentation of all-sky images. *Atmos. Meas. Tech.*, **15**, 797–809, <https://doi.org/10.5194/amt-15-797-2022>.
- Grill, J.-B., and Coauthors, 2020: Bootstrap your own latent: A new approach to self-supervised learning. arXiv, 2006.07733v3, <https://doi.org/10.48550/arXiv.2006.07733>.
- Guo, J., J. Yang, H. Yue, and K. Li, 2021: Unsupervised domain adaptation for cloud detection based on grouped features alignment and entropy minimization. *IEEE Trans. Geosci. Remote Sens.*, **60**, 1–13, <https://doi.org/10.1109/TGRS.2021.3067513>.
- Guo, P., X. Su, H. Zhang, M. Wang, and F. Bao, 2020: A multi-scaled receptive field learning approach for medical image segmentation. *ICASSP 2020–2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, Institute of Electrical and Electronics Engineers, 1414–1418, <https://doi.org/10.1109/ICASSP40776.2020.9054030>.
- Heinle, A., A. Macke, and A. Srivastav, 2010: Automatic cloud classification of whole sky images. *Atmos. Meas. Tech.*, **3**, 557–567, <https://doi.org/10.5194/amt-3-557-2010>.
- Huo, J., and D. Lu, 2009: Cloud determination of all-sky images under low-visibility conditions. *J. Atmos. Oceanic Technol.*, **26**, 2172–2181, <https://doi.org/10.1175/2009JTECHA1324.1>.
- Karimi, D., H. Dou, S. K. Warfield, and A. Gholipour, 2020: Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Med. Image Anal.*, **65**, 101759, <https://doi.org/10.1016/j.media.2020.101759>.
- Kazantzidis, A., P. Tzoumanikas, A. F. Bais, S. Fotopoulos, and G. Economou, 2012: Cloud detection and classification with the use of whole-sky ground-based images. *Atmos. Res.*, **113**, 80–88, <https://doi.org/10.1016/j.atmosres.2012.05.005>.
- Kreuter, A., M. Zangerl, M. Schwarzmam, and M. Blumthaler, 2009: All-sky imaging: A simple, versatile system for atmospheric research. *Appl. Opt.*, **48**, 1091–1097, <https://doi.org/10.1364/AO.48.001091>.
- Kurihana, T., and Coauthors, 2019a: Cloud classification with unsupervised deep learning. arXiv, 2209.15585v1, <https://doi.org/10.48550/arXiv.2209.15585>.
- , and Coauthors, 2019b: Cloud classification with unsupervised deep learning. *Proc. Ninth Int. Workshop on Climate Informatics: (CI 2019)*, Paris, France, École Normale Supérieure, 37–42.
- , E. Moyer, R. Willett, D. Gilton, and I. Foster, 2021: Data-driven cloud clustering via a rotationally invariant autoencoder. *IEEE Trans. Geosci. Remote Sens.*, **60**, 1–25, <https://doi.org/10.1109/TGRS.2021.3098008>.
- LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *Nature*, **521**, 436–444, <https://doi.org/10.1038/nature14539>.
- Li, X., Z. Lu, Q. Zhou, and Z. Xu, 2019: A cloud detection algorithm with reduction of sunlight interference in ground-based sky images. *Atmosphere*, **10**, 640, <https://doi.org/10.3390/atmos10110640>.
- Liou, K.-N., 1986: Influence of cirrus clouds on weather and climate processes: A global perspective. *Mon. Wea. Rev.*, **114**, 1167–1199, [https://doi.org/10.1175/1520-0493\(1986\)114<1167:IOCCOW>2.0.CO;2](https://doi.org/10.1175/1520-0493(1986)114<1167:IOCCOW>2.0.CO;2).
- Liu, L., X. Sun, F. Chen, S. Zhao, and T. Gao, 2011: Cloud classification based on structure features of infrared images. *J. Atmos. Oceanic Technol.*, **28**, 410–417, <https://doi.org/10.1175/2010JTECHA1385.1>.
- Long, C. N., J. M. Samburg, J. Calbó, and D. Pagès, 2006: Retrieving cloud characteristics from ground-based daytime color all-sky images. *J. Atmos. Oceanic Technol.*, **23**, 633–652, <https://doi.org/10.1175/JTECH1875.1>.
- Minka, T. P., 2000: Automatic choice of dimensionality for PCA. M.I.T. Media Laboratory Perceptual Computing Section Tech. Rep. 514, 16 pp., <https://vismod.media.mit.edu/tech-reports/TR-514.pdf>.
- Morris, V. R., 2016: Ceilometer Instrument Handbook. Tech. Rep. DOE/SC-ARM-T, 26 pp., DOE/SC-ARM-TR-020, <https://doi.org/10.2172/1036530>.
- NOAA, 1998: Automated surface observing system: ASOS user’s guide. Federal Aviation Administration, U.S. Navy, U.S. Department of the Air Force, 74 pp., <https://www.weather.gov/media/asos/aum-toc.pdf>.
- Oktay, O., and Coauthors, 2018: Attention U-Net: Learning where to look for the pancreas. arXiv, 1804.03999v3, <https://doi.org/10.48550/arXiv.1804.03999>.
- Onishi, R., and D. Sugiyama, 2017: Deep convolutional neural network for cloud coverage estimation from snapshot camera images. *SOLA*, **13**, 235–239, <https://doi.org/10.2151/sola.2017-043>.
- Park, S., Y. Kim, N. J. Ferrier, S. M. Collis, R. Sankaran, and P. H. Beckman, 2021: Prediction of solar irradiance and photovoltaic solar energy product based on cloud coverage estimation using machine learning methods. *Atmosphere*, **12**, 395, <https://doi.org/10.3390/atmos12030395>.
- Peura, M., A. Visa, and P. Kostamo, 1996: A new approach to land-based cloud classification. *Proc. 13th Int. Conf. on Pattern Recognition*, Vienna, Austria, Institute of Electrical and Electronics Engineers, 143–147, <https://doi.org/10.1109/ICPR.1996.547250>.
- Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, 2019: Language models are unsupervised multitask learners. OpenAI blog, 24 pp., https://d4mucfpxyww.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- Raut, B. A., and Coauthors, 2023: Optimizing cloud motion estimation on the edge with phase correlation and optical flow. *Atmos. Meas. Tech.*, **16**, 1195–1209, <https://doi.org/10.5194/amt-16-1195-2023>.

- Ronneberger, O., P. Fischer, and T. Brox, 2015: U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, Lecture Notes in Computer Science, Vol. 9351, Springer, 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
- Souza-Echer, M. P., E. B. Pereira, L. Bins, and M. Andrade, 2006: A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *J. Atmos. Oceanic Technol.*, **23**, 437–447, <https://doi.org/10.1175/JTECH1833.1>.
- Sun, W., O. Nasraoui, and P. Shafto, 2020: Evolution and impact of bias in human and machine learning algorithm interaction. *PLOS ONE*, **15**, e0235502, <https://doi.org/10.1371/journal.pone.0235502>.
- Tian, B., M. A. Shaikh, M. R. Azimi-Sadjadi, T. H. V. Haar, and D. L. Reinke, 1999: A study of cloud classification with neural networks using spectral and textural features. *IEEE Trans. Neural Networks*, **10**, 138–151, <https://doi.org/10.1109/72.737500>.
- Wacker, S., and Coauthors, 2015: Cloud observations in Switzerland using hemispherical sky cameras. *J. Geophys. Res. Atmos.*, **120**, 695–707, <https://doi.org/10.1002/2014JD022643>.
- Wagner, T. J., and J. M. Kleiss, 2016: Error characteristics of ceilometer-based observations of cloud amount. *J. Atmos. Oceanic Technol.*, **33**, 1557–1567, <https://doi.org/10.1175/JTECH-D-15-0258.1>.
- Wang, M., S. Zhou, Z. Yang, and Z. Liu, 2020a: CloudA: A ground-based cloud classification method with a convolutional neural network. *J. Atmos. Oceanic Technol.*, **37**, 1661–1668, <https://doi.org/10.1175/JTECH-D-19-0189.1>.
- Wang, X., Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, 2020b: Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tutorials*, **22**, 869–904, <https://doi.org/10.1109/COMST.2020.2970550>.
- Xie, W., and Coauthors, 2020: SegCloud: A novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation. *Atmos. Meas. Tech.*, **13**, 1953–1961, <https://doi.org/10.5194/amt-13-1953-2020>.
- Yang, J., W. Lu, Y. Ma, and W. Yao, 2012: An automated cirrus cloud detection method for a ground-based cloud image. *J. Atmos. Oceanic Technol.*, **29**, 527–537, <https://doi.org/10.1175/JTECH-D-11-00002.1>.
- Zhang, J., P. Liu, F. Zhang, and Q. Song, 2018: CloudNet: Ground-based cloud classification with deep convolutional neural network. *Geophys. Res. Lett.*, **45**, 8665–8672, <https://doi.org/10.1029/2018GL077787>.
- Zhuo, W., Z. Cao, and Y. Xiao, 2014: Cloud classification of ground-based images using texture–structure features. *J. Atmos. Oceanic Technol.*, **31**, 79–92, <https://doi.org/10.1175/JTECH-D-13-00048.1>.