Supporting information for

Capturing the diversity of mesoscale trade wind cumuli using complementary approaches from self-supervision

Dwaipayan Chatterjee[1], Sabrina Schnitt[1], Paula Bigalke[1], Claudia Acquistapace[1], Susanne Crewell[1]

[1]**Institute for Geophysics and Meteorology, University of Cologne, Cologne, Germany**

November 3, 2023, 12:55pm

Contents of this file

Corresponding author: D. Chatterjee, Institute for Geophysics and Meteorology, University of Cologne, Cologne, Germany. (dchatter@uni-koeln.de)

- Figure S5
- Text S5

S1 Domain description

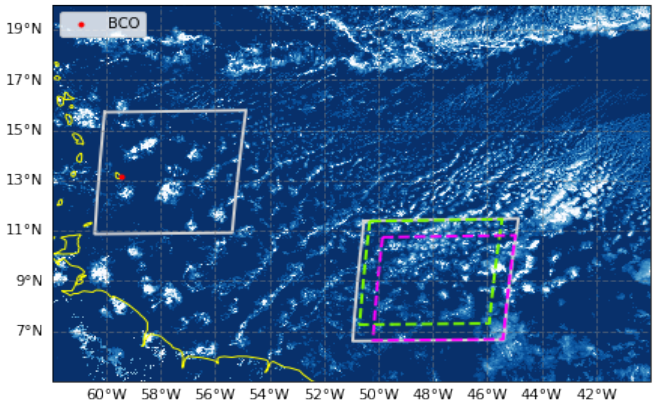

Figure S1 (Domain). GOES's COD image on February 2, 2020, at 13:00 UTC with coastal boundaries (thick yellow) and Barbados Cloud Observatory (red dot). One (out of five) random and a fixed (Barbados domain) 256 x 256-pixel crop over EUREC^{4}A domain are shown. During the learning process, each crop is twice randomly sub-cropped (pink and green dashed lines) by the network, leading to a spatial dimension of 75% (192 x 192 pixels) of the original crop. The Barbados domain enables comparison with ground-based measurements in future studies.

S2 Network architectures Here, N1 and N2 (Chatterjee et al., 2023) architectures are described in detail.

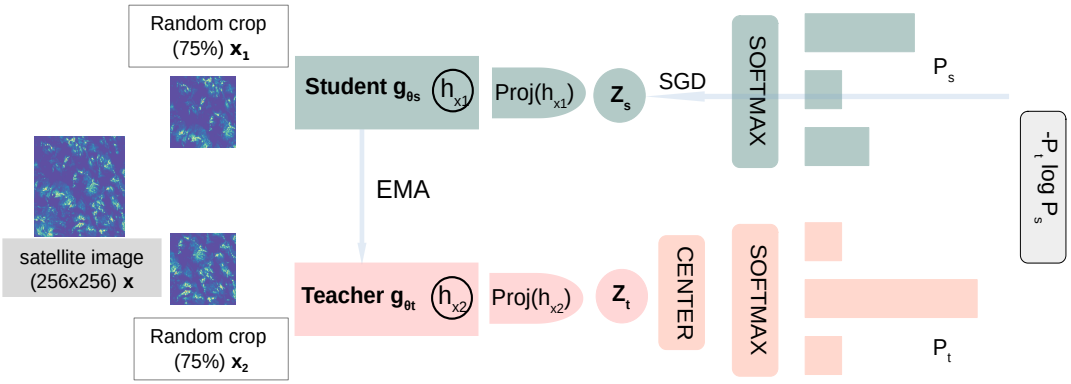

Figure S2.1 (Schematic diagram of N1). This work adopts a deep learning architecture from Caron et al. (2021), where x_1 and x_2 are 75% random crops of the parent satellite image x. The student and teacher vision transformers ($g_{\theta s/t}$) have the same number of trainable parameters (weights and biases) θ. The feature output h_{xi} from g_{xi} subsequently connects to $Proj(h_{xi})$, a 3-layer multilayer perceptron activated by Gaussian error linear units (GELU, with the last layer, l_2 normalized). Softmax (Bridle, 1989) normalizes MLP's raw activation ($z_{s/t}$), and centering maintains teacher activations (z_t) near batch mean properties. P_s and P_t represent normalized student distribution of z_s and centered and normalized distribution of teacher activation z_t. The student network optimizes its parameters through stochastic gradient descent (SGD), minimizing cross-entropy between P_t and P_s. Teacher parameters ($g_{\theta t}$) are exponential moving averages of students ($g_{\theta s}$), aligning the networks. This interaction forms the architecture's backbone, enhancing performance and knowledge transfer in the deep learning framework.

1. Continuous network (N1)

1.1. Definition of the network input

N satellite images of COD built the input training data set $X = \{x_1, x_2, x_3....., x_N\}$ of the deep learning architecture illustrated in Fig. S2.1 (Schematic diagram of N1). The only intuitive augmentation we opt for here is global random cropping for learning continuous representations. For random cropping, we opt for two global crops (x_1, x_2) with a random 0.75 fraction (192 x 192 pixels) of the parent satellite image to focus on the global distribution of the cloud system. Figure S2.1 (Schematic diagram of N1) shows each random crop fed into different branches of the network, and from the learning aspect of the neural network, it becomes challenging for one side of the network to know what part of the parent satellite image the other is being fed with; therefore, it focuses on learning the critical semantics of global cloud distribution.

1.2. General network architecture

The neural network's task is to learn visual features from each satellite image. A function g represents the transformations performed by the network's vision Transformer (ViT) as $g(x_i) = h_j$ with $i = 1...N, j = 1....M$ that maps the image x_i into the array of features h $=\{h_1, h_2, h_3....., h_M\}$, where M is the output dimension of ViT feature arrays. The selected dimension of M is equal to 384, which means the information contained in the

192 x 192 satellite observation space is being non-linearly dimensionally reduced to 384 vector space. ViT is a sequence of self-attention (Vaswani et al., 2023), and feed-forward layers paralleled with skip connections. The mechanism of ViT (Dosovitskiy et al., 2021) takes non-overlapping contiguous image patches of resolution NxN pixels, where N=16 for this work, along with their positional encoding as an input. Without the positional encoding, the output feature vector from ViT is invariant to the arrangement of these NxN patches. But with positional encoding, it learns the relative position of the objects in the image. This becomes helpful if we want the model to learn the relationship between the patches. Thus it learns how a particular arrangement of cloud distribution usually occurs, and it learns the constrained settings behind the appearance of a given cloud distribution. Therefore, the ViT architecture can identify long-range spatial dependencies (Khan et al., 2022) by learning relevant information in the image. The activation function used in the ViT is Gaussian error linear units (Hendrycks & Gimpel, 2023) (GELU), as the GELU function behaves smoother when values are closer to zero and thus is more effective at learning complex patterns in the data.

Further, h_j is non-linearly projected to z_k with $k = 1....K$ using a three-layer multilayer perceptron activated by GELU followed by l_2 normalization and a linear layer. Here z $=\{z_1, z_2, z_3....., z_K\}$ is the final output dimension of the pipeline. The feature space dimensions are decided based on input dimensions, the complexity of information context, and neural network complexity. Caron et al. (2021) suggest that if the training dataset size is much less than 1.3 million ImageNet datasets (Russakovsky et al., 2015), then the final dimensions of z_k be reduced compared to the default dimensions of 65536. We iterated on

a smaller K dimension of 128 and a higher K dimension of 8192; in this case, we visually found the latter working better to understand the similarity between cloud fields. Our intuition is giving the final output feature more dimensions gives the model more freedom to observe small semantic details of cloud distributions. Also, since the loss function used here (explained in section 1.3) is non-contrastive, higher dimensional features are still computationally inexpensive. Our aim here is not to find the optimal feature vector size but a functional size that can optimize the network and smoothly converge the training. Therefore, the optimal dimension size of the dimensionally reduced atmospheric fields in self-supervised learning is not the focus of this work. Figure S2.1 (Schematic diagram of N1) shows two different branches in the network: student and teacher. The point to note here is that they have the same general architecture and pipeline, but the parameters (weights and biases) learned during training are different.

1.3. Upper branch of the network

The upper branch of the network, represented in Figure S2.1 (Schematic diagram of N1), by the student transformer g_s and further projected by multi-layer perceptron (MLP) (Rumelhart et al., 1986), ingests one random augmented global crop of the parent satellite image and outputs feature vector z_s. z_s is normalized and converted to a probability distribution. This means that the original feature vector z_s (which has 8192 dimensions); some values could be negative or greater than one, but after applying soft-max (Bridle, 1989), it normalizes to (0,1), and all the 8192 dimensions will add to one. Additionally, the larger input components will correspond to larger probabilities. This probability distribution of the feature vector z_s is an input to the cross entropy loss function described

later. The soft-max probability for an input x_i of the student network can be described as

$$p_s^{(i)} = \frac{\exp(\frac{1}{\zeta_s} Z_s^{(i)})}{\sum_{m=0}^{k} \exp(\frac{1}{\zeta_s} Z_s^{(m)})}. \quad (1)$$

where ζ_s is the temperature parameter for the student network and is set to 0.1. The ζ parameter controls the sharpening of the probability distribution. A higher value of ζ implies smoothed probability.

1.4. Lower branch of the network

The lower branch of the network represented in Figure S2.1 (Schematic diagram of N1) by the teacher transformer applies function g_t to the other remaining global crop of the parent satellite image, and the MLP projects outputs feature vector z_t. Unlike p_s, before normalizing p_t individually with soft-max, vector z_t is centered around the mean properties of all images in a batch. A batch refers to the number of samples propagating through the neural network before updating the model parameters. Centering is done to prevent any feature from dominating, as the mean will be somewhere in the middle of the batch sample properties. While applying the temperature ζ_t parameter for the teacher, it is kept lower to 0.05 to sharpen the probability of z_t artificially. Therefore, the feature vector z_t of the teacher branch went through centering and sharpened soft-max before being input to the loss function.

1.5. Cross entropy loss of the network

When the feature vectors of the two branches capture similar information from the global crops of the satellite parent image, the loss becomes lower and vice-versa. That's

how the network branches are encouraged to focus on the common image characteristics, progressively making the feature vectors similar.

$$\min_{\theta_s} \sum_{x \in (x_1, x_2)} p_t(x) log(p_s(x)) \tag{2}$$

This is achieved through the cross-entropy loss function applied on the centered and sharpened probability distribution of the teacher branch p_t and smoothened distribution of the student branch p_s. As shown in equation 2, the loss function minimizes θ_s, i.e., the student network's parameters (weights and biases). Teacher network parameters or p_t guide the student network during the training phase, as discussed in subsection 1.6.

1.6. Optimization for convergence

The loss function minimization happens progressively layer by layer, derivating the loss function with respect to θ_s parameters and adjusting parameter values in each layer by backpropagation. At the end of the minimization, we obtain a configuration of parameters for the student network that will be ready for the next iteration with a new batch of images. Stochastic gradient descent (Bottou, 2012) (SGD) is only applied to the student network parameters θ_s, and the teacher parameters θ_t are built through past iterations of the student network (Caron et al., 2021). As shown in equation 3, θ_t is the exponential moving average (EMA) of θ_s with λ following a cosine scheduled from 0.996 to 1 during training.

$$\theta_t = \lambda \theta_t + (1 - \lambda)\theta_s \tag{3}$$

During optimization, a collapse can occur regardless of the input provided to the model; the output becomes constant or is predominantly influenced by a single dimension. In other words, the model's predictions across different dimensions or features become uni-

form, leading to zero ideal loss value. Therefore, centering and sharpening introduced in subsection 1.3 and 1.4 and EMA (subsection 1.6) are the easiest acceptable ways to prevent collapsing in the described teacher-student framework.

1.7. Training and libraries

To set up this architecture, we use the software package DINO from Facebook Artificial Intelligence Research (FAIR) (Caron et al., 2021) based on PyTorch. The open-source VISSL computer vision library (Goyal et al., 2021) adapted the DINO neural network to our requirements. Based on sensitivity tests on training loss, visualization of dimensionally reduced feature space, and ablation study of the original network on longer training showing improving performance, we train the model up to 800 epochs. Training the neural network for 800 epochs on 4 V100 GPUs took 16.5 hours or 66 core hours.

S2 Network architectures

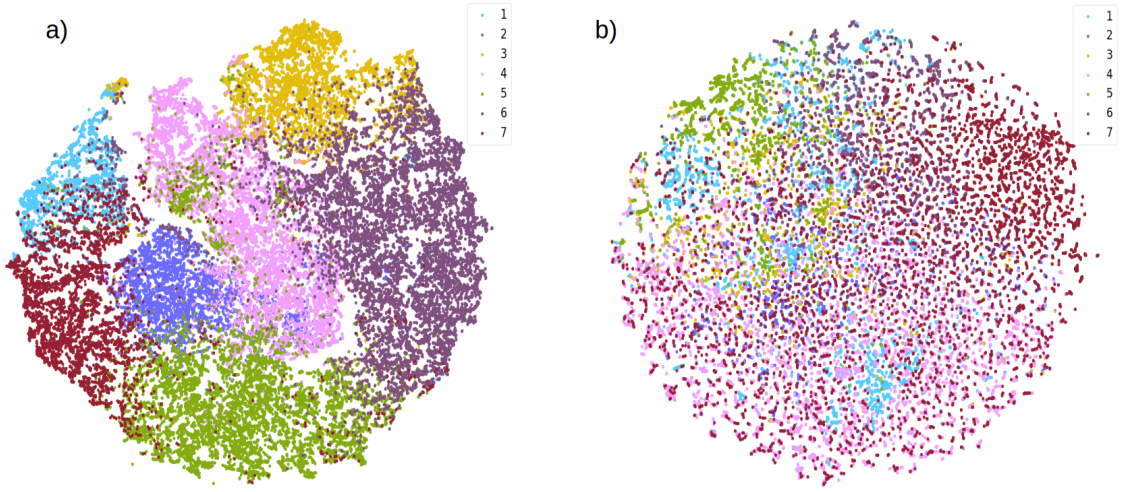

Figure S2.2 (N2 outputs). a) Sparse 2D feature space obtained from N2 by applying the tSNE algorithm on z_x features of 51,000 satellite images. The perplexity and epsilon derived from auto-configuration for t-SNE runs is 30 and 1150. b) Same as b but using direct clustering on the satellite images using N2. Here, the labels are overlaid on the continuous feature space from N1 for comparison with Figure 2.b in the main article.

2. Discrete network (N2)

We briefly describe the functional mechanism of the discrete neural network (N2) and its learning scheme. Refer to section 3 from (Chatterjee et al., 2023) for a detailed network description. The data loading nature of N2 remains the same as of N1 (subsection 1.1 of text S2). The general architecture has a pipeline similar to the continuous approach setup, with the image processing backbone here being a convolutional residual network with 50 layers of depth (ResNet-50, (He et al., 2015)), followed by a projection head of MLP with ReLU activations (Fukushima, 1975) and a linear layer. Therefore, similar to Figure S2.1 (Schematic diagram of N1), there are two branches. For the upper branch, the features obtained at the end of the pipeline (like Z_s in

the continuous approach) are clustered using spherical k-means (where k=7), and features are allocated a pseudo-label (L) according to their closest centroid. Further, the features obtained from the lower branch are compared with the calculated upper branch centroids using cosine distance (D_L). Finally, L from the upper branch and D_L from the lower branch are inputs of the cross-entropy loss function as discussed in subsection 1.5 of text S2 and is progressively minimized during training. We call the labels as pseudo-labels during the training stage as they can change to minimize the loss function better. Finally, at the end of the training, we collect the labels for each satellite image and further evaluate their separation using auxiliary datasets.

S3 Identify the optimal class

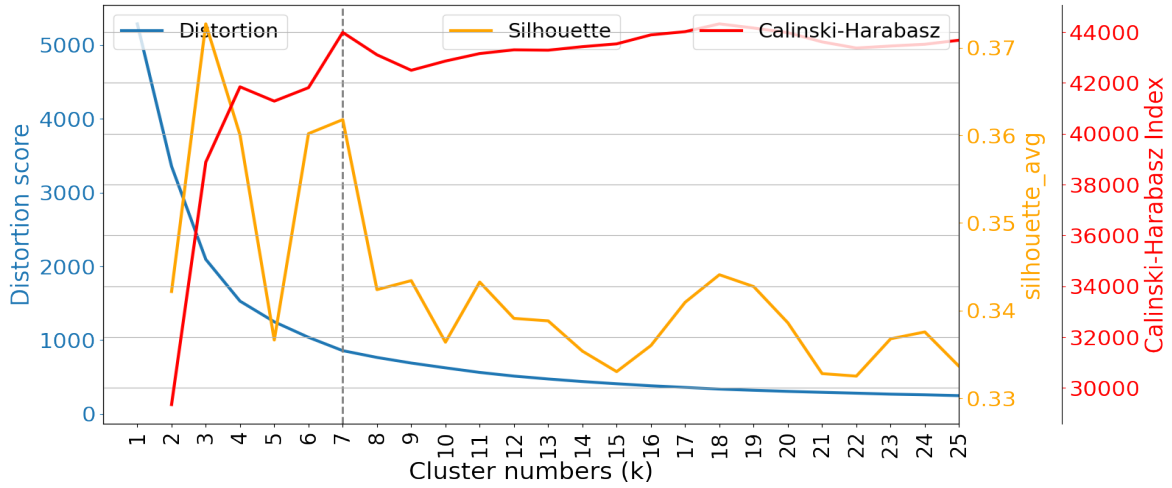

Figure S3 (Metric scores). Results of three different metric scores of distortion, silhouette, and Calinski-Harabasz, shown along with varying cluster numbers along the abscissa. The vertical-dashed line is drawn at cluster 7, which shows the chosen inflection point for the optimal cluster.

Text S3: Determination of optimal cluster number

We apply the following metrics to two-dimensionally reduced representations (using tSNE) on h_j from N1 to identify the best optimal cluster:

1. **Distortion metric:** The distortion metric considers the cluster's tightness by computing the sum of squared distances (SSD) from each point to its assigned center, which tends to decrease toward 0 as we increase the number of clusters (K). This shows an exponential shape leveling off such that the shape of the curve results in an elbow, but

the optimal cluster or the point of inflection represents the point where adding additional clusters stops adding useful information. Also, adding clusters beyond the inflection point also makes the clusters harder to separate; thus, we start to observe diminishing returns by increasing k. The elbow blue line curve in figure S3 (Metric scores) shows k = 7 as the sweet spot of optimal clustering.

2. **Silhouette metric:** Apart from taking cluster closeness into account, this metric also considers distances between points of one cluster and the nearest other cluster center. This means that in order to have a good silhouette score, clusters generally need to be tighter and farther apart from each other. If the Silhouette coefficient for each point is close to 0, it means that the point is between two clusters; if it is close to -1, then that point is in the wrong cluster, and if it is close to +1, it is in the correct cluster. The average silhouette coefficient calculated for all 51,000 samples shows two local maxima at values of 0.37 (k=3) and 0.36 (k=7), as shown in Figure S3 (Metric scores). Note that the values are not close to one, meaning the cluster doesn't lie very far from each other, further suggesting the continuous nature of cloud organizations.

3. **Calinski-Harabasz metric:** In comparison, the Calinski-Harabasz metric assesses the separation and compactness of the clusters. It denotes the ratio of the sum of inter-cluster dispersion and the sum of intra-cluster dispersion for all clusters. A good clustering result has a high Calinski-Harabasz Index value. The maximum lies at cluster 7, having a score of 43000.

In summary, the two metrics directs towards k=7, and the difference between the two maxima (k=3 and 7) in silhouette is insignificant. Therefore, we take the common agree-

ment of k=7 as the optimal cluster number and train N2 (section 2) from scratch using 7 clusters.

S4 Visualizing the internal layers of the trained network

Text S4 Here, we investigate whether N1 learns reasonable visual features of satellite images. This will help us to understand our network's decision-making and may boost our confidence in the neural network's final representations Z_k. From a human perspective, cloud system distributions may appear to be relatively chaotic and noisy, and while trying to decide their visual characteristics, we may pay attention to some or all of the following: the organizational semantics of convective organization, the semantics of the clear sky regions, deep convective cell distributions, open and closed cells, and shallow convection distributions. Similarly, to build trust in the network's performance, it is crucial to see what the trained N1 architecture has learned to pay attention to when deciding the features h_j of cloud system distributions.

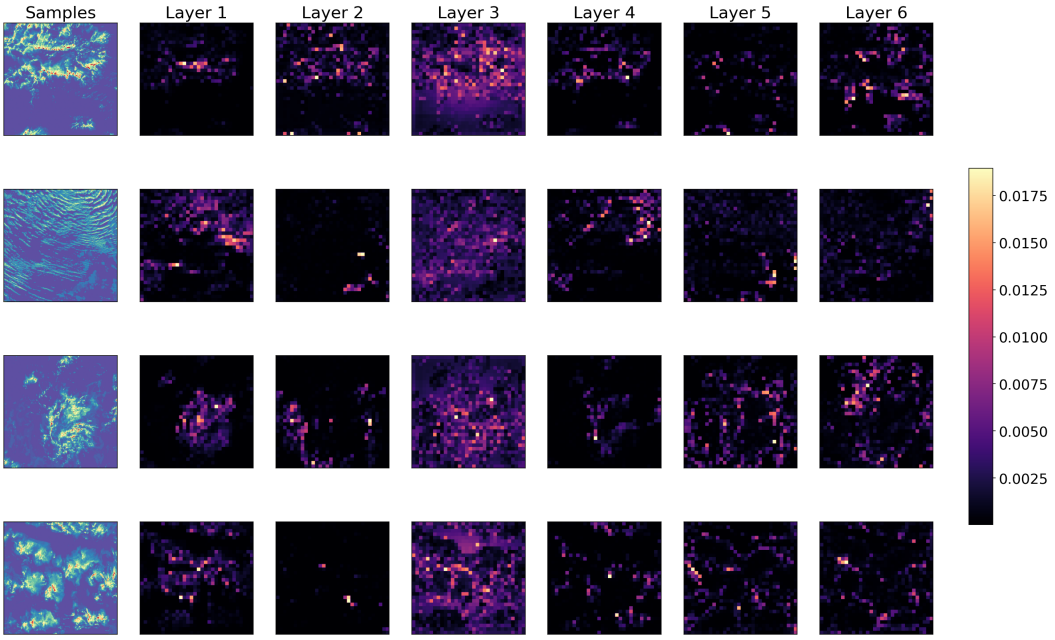

Figure S4 (Visualization of different layers). Four cloud systems with different organizations are selected as examples. Their respective self-attention maps from the final head of the teacher ViT trained with 8 x 8 patches are shown in layers 1 - 6. The color bar indicates the range of the Gaussian error linear units (GELU) activation function for the activation maps. Higher values indicate more important features. All experiments are run with a default of six self-attention heads.

Given a satellite image, the activation space in a neural network allows us to visualize whether a neuron should be activated, indicating what part of the image is important for the network. The self-attention layers in ViT try to decompose the input samples and learn relatively independent features. Thus, this experiment aims to see whether the activation space reveals the abstract patterns that we, as humans, can make sense of while deciding the feature's importance. In this setup, we use a single satellite image sample and pass it through the trained model, freezing the weights. The granularity (N x N), or the number of pixels in a single patch, is controlled by the patch size, which is 8 x 8 pixels in this experiment.

Figure S4 (Visualization of different layers) shows that layer one activates at the dominant convective cells and deactivates at thin spread-out convection while layer 2 activates the thin spread convection. Layer three seems to try to learn and activate the clear sky features. In contrast to layer one, layer four activates the rest of the prominent convections. Like layer two, layer five tries to look at the rest of the thin-spread convection. Layer six is uncertain and is not obvious to our eyes, and it may somehow try to deactivate for all the clear sky regions in the majority of cases and look for boundary semantics in the satellite image. Examining other example cases shows the same consistency, and therefore, it can be concluded that although the cloud system distributions are different, each attention map has learned to pay attention to relatively different, consistent, sensible semantics of the cloud systems distribution and further indicates that we can trust the embedding space of the network.

S5 Environmental characteristics of human labels and neighbors

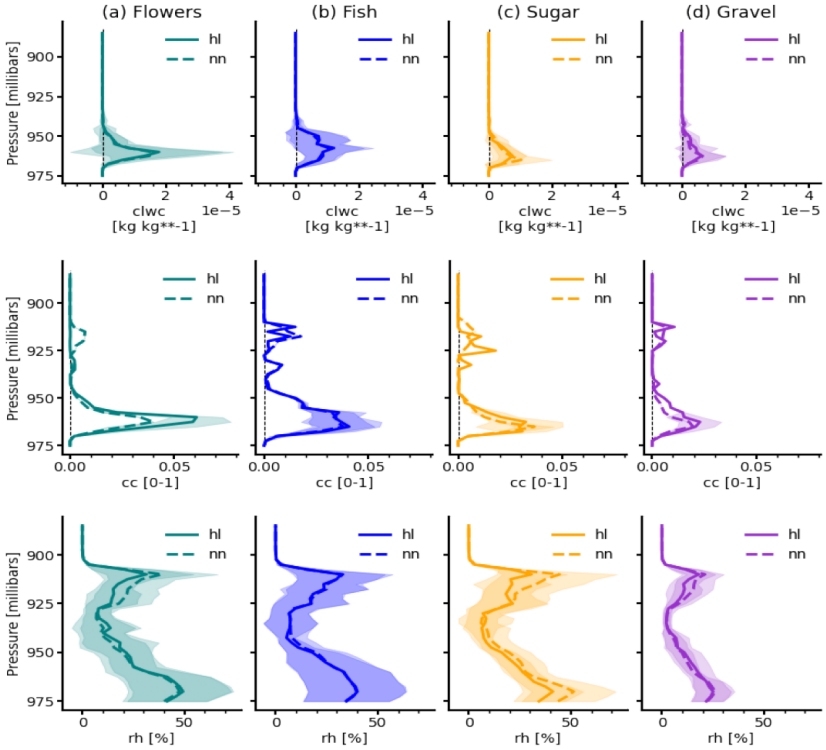

Figure S5. Comparison of 52 human labels (hl) environmental conditions with their nearest 30 neighbors (nn) using ERA-5. The top to bottom rows shows weighted-average and standard deviation profiles of cloud water content (clwc, kg kg $^{-1}$), cloud cover (cc), and relative humidity (rh, %) with the exception of cc variability shown in the interquartile range.

Text S5 Figure 3.c in section 4.2 of the main manuscript showed the occurrences of 30 nearest neighbors of human-labeled satellite images (mentioned as human crops below)

with machine-identified seven classes. Here, we aim to assess their existing environmental conditions. This complementary experiment can further help to trust human crops' relative positions in the feature space. If the human crops and the neighbors have a similar homogenous distribution of their physical properties, this implies that the human crops are in the consistent region of the feature space. Here, we take the ERA-5 vertical profile of cloud water content, cloud cover, and relative humidity (Fig. S5) to compare the weighted averaged vertical profiles between human labels and their 30 nearest neighbors. When calculating these properties for human-labeled scenes, we weigh them with the level of agreement In this way, the contribution of well-agreed organizations will contribute more than less agreed cloud organizations. We observe that there is hardly any difference in the vertical profiles except for the relative humidity of sugar and cloud cover for flowers. This may be due to quantitatively using 30 times more data.

References

Bottou, L. (2012). Stochastic gradient descent tricks. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade: Second edition* (pp. 421–436). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-35289-8_25 doi: 10.1007/978-3-642-35289-8_25

Bridle, J. S. (1989). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Nato neurocomputing.* Retrieved from https://api.semanticscholar.org/CorpusID:59636530

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). *Emerging properties in self-supervised vision transformers.*

Chatterjee, D., Acquistapace, C., Deneke, H., & Crewell, S. (2023). Understanding cloud systems structure and organization using a machine's self-learning approach. *Artificial Intelligence for the Earth Systems*. Retrieved from https://journals.ametsoc.org/view/journals/aies/aop/AIES-D-22-0096.1/AIES-D-22-0096.1.xml doi: https://doi.org/10.1175/AIES-D-22-0096.1

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale.*

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, *20*(3-4), 121–136. Retrieved 2022-08-30, from http://link.springer.com/10.1007/BF00342633 doi: 10.1007/BF00342633

Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., ... Misra, I. (2021). *Vissl.* https://github.com/facebookresearch/vissl.

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Deep Residual Learning for Image Recognition. Retrieved 2022-08-30, from http://arxiv.org/abs/1512.03385 (arXiv:1512.03385 [cs])

Hendrycks, D., & Gimpel, K. (2023). *Gaussian error linear units (gelus).* Retrieved from https://arxiv.org/abs/1606.08415

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022, jan). Transformers in vision: A survey. *ACM Computing Surveys*, *54*(10s), 1–41. doi: 10.1145/3505244

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986, October). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. Retrieved from https://doi.org/

10.1038/323533a0 doi: 10.1038/323533a0

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015, December). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. Retrieved 2022-08-31, from http://link.springer.com/10.1007/s11263-015-0816-y doi: 10.1007/s11263-015-0816-y

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2023). *Attention is all you need.* Retrieved from https://arxiv.org/abs/1706.03762