Few shot Classification for Meso-cloud formations

Jim Vos

Supervisors:
Marcel J.T Reinders
Jing Sun
Geet George

A thesis presented for the degree of Master in Computer Science

Delft University of Technology
The Netherlands

May 15, 2026

Abstract

Automated cloud classification is severely bottlenecked by the need for massive, region-specific annotated datasets. This thesis investigates few-shot learning (FSL) approaches for the classification of mesoscale cloud formations across different geographic regions under limited data conditions.

To address this problem, the influence of varying amounts of additional data on generalisation to novel cloud regimes is evaluated. The study compares representation learning strategies, including self-supervised and supervised approaches, to assess their effectiveness in structuring the latent space for distinguishing cloud types. In parallel, two learning paradigms, transfer learning and episodic meta-learning, are analysed to determine how effectively they incorporate additional data when adapting to novel classes.

The results show that self-supervised learning is most effective in the strict few-shot regime, while supervised transfer learning makes the most effective use of additional data. In particular, Barlow Twins achieves the strongest performance under minimal data by avoiding reliance on noisy labels. When additional out-of-domain data, such as ImageNet and auxiliary cloud datasets, is introduced, supervised pre-training combined with transfer learning attains performance comparable to standard supervised learning, while requiring only a small number of labelled examples.

Contents

1 Introduction

The precision of global climate models is inextricably linked to our understanding of clouds, whose spatial organisation govern the Earth's radiative energy balance. Clouds play a dual role in the climate system—reflecting incoming solar radiation while trapping outgoing thermal heat—but their net effect depends heavily on their formation, altitude, and type [1]. A major source of uncertainty in predicting future climate scenarios is how these cloud regimes will shift as the Earth warms, potentially creating feedback loops that either accelerate or dampen global temperature rises [2]. While modern satellite imagery provides a near-global, view of the atmosphere, extracting actionable, large-scale climate insights from this raw visual data remains a significant computational challenge.

Historically, automated cloud classification from satellite data has operated at the pixel level, utilising single measurements or joint histograms to determine properties like optical depth or phase [3] [4] [5]. However, the radiative impact of marine low clouds is largely dictated by their mesoscale morphology—spatial patterns organised on the scale of tens to hundreds of kilometres. At this scale, clouds form distinct phenomenological classifications that represent complex, underlying physical processes [6]. Recognising this, researchers established four reproducible patterns of mesoscale shallow cloud organisation in the North Atlantic trades: Sugar, Gravel, Fish, and Flowers visible in Figure 1a. Because identifying these patterns requires a holistic, spatial understanding of the image rather than pixel-by-pixel analysis, deep learning has emerged as the most promising tool for the task.

While deep learning algorithms can successfully mimic human pattern recognition in satellite imagery, their application is severely bottlenecked by the need for massive amounts of annotated training data. To train robust classification models, an effort has been made to combine crowdsourcing with deep learning, relying on 67 scientists to manually roughly 49,000 mesoscale cloud clusters. Although this demonstrated the feasibility of automated mesoscale classification, it also highlighted a critical limitation: the resulting models were geographically restricted to a single ocean basin and season. Because cloud morphologies and visual distributions vary significantly across different global regimes, scaling this supervised approach requires crowdsourcing new, massive datasets for every new region. From a practical and financial standpoint, relying on exhaustive human labelling for global climate analysis is unsustainable.

This scarcity of labelled data is the central problem that Few-Shot Learning (FSL) is designed to address. Rather than requiring thousands of annotations, FSL frameworks enable a machine learning model to generalise to new, previously unseen classes or domains using only a very small "support set" of k labelled examples. To achieve this, FSL exist that rely on robust representation learning—pre-training a model on a large background dataset so it learns to extract highly adaptable, generalized features. When applied to new data, FSL typically utilises one of two paradigms: transfer learning, where the pre-trained model is simply fine-tuned on the small support set, or meta-learning, where the model is explicitly trained over many learning episodes to optimise its ability to learn from limited data. Although FSL has become a well known technology it has not been applied to mesoscale cloud classification.

This thesis addresses that gap by investigating how few-shot learning can be leveraged to classify mesoscale cloud formations without the need for exhaustive crowdsourced labelling. Specifically, this research is guided by the following research questions:

RQ1. How do different sources of pre-training data affect the performance and algorithmic choices in few-shot cloud classification?

- **RQ1.a** Which representation learning objective yields the most adaptable feature space for few-shot adaptation?
- **RQ1.b** How does the target support set size (k) dictate the optimal pre-training and fine-tuning strategy?
- **RQ1.c** How do transfer learning and meta-learning differ in their ability to utilise prior background data for few-shot adaptation?

1.1 Problem Formulation

The primary challenge in automated cloud classification is geographic generalization. Because cloud morphologies are governed by local meteorological conditions, different geographical locations exhibit distinct cloud regimes [7]. Consequently, expanding atmospheric analysis to a new geographical region requires classifying a novel set of region-specific cloud types.

To address this without relying on exhaustive manual labelling, region-adaptive cloud classification is formulated here as a few-shot learning problem. In practical scenarios, abundant existing data from previously annotated domains can be leveraged as a set of *base classes* ($\mathcal{C}_{base}$) for initial model training. The objective is to transfer these learned representations to a new target region, treating its localised cloud types as a set of *novel classes* ($\mathcal{C}_{novel}$) where labelled data is inherently scarce.

Formally, let a satellite observation be represented as an image tensor $x \in \mathbb{R}^{H\times W\times 3}$. The task is framed as an N-way K-shot classification problem. During inference, the model is provided with a small support set containing representative examples from the new target region, defined as

$$\mathcal{S}_{novel} = \{(x_i, y_i)\}_{i=1}^{N\times K}, \tag{1}$$

where x_i represents an image patch and $y_i \in \mathcal{C}_{novel}$ is its corresponding novel class label. This support set contains exactly K labelled examples for each of the N classes present in the new region.

Given this setup, the goal is to learn a mapping function f_θ, parameterized by neural network weights θ. For an unlabelled query image x_q originating from the same new region, the model predicts its class label $\hat{y}_q$ conditioned strictly on the novel support set:

$$\hat{y}_q = f_\theta(x_q, \mathcal{S}_{novel}). \tag{2}$$

1.2 Datasets

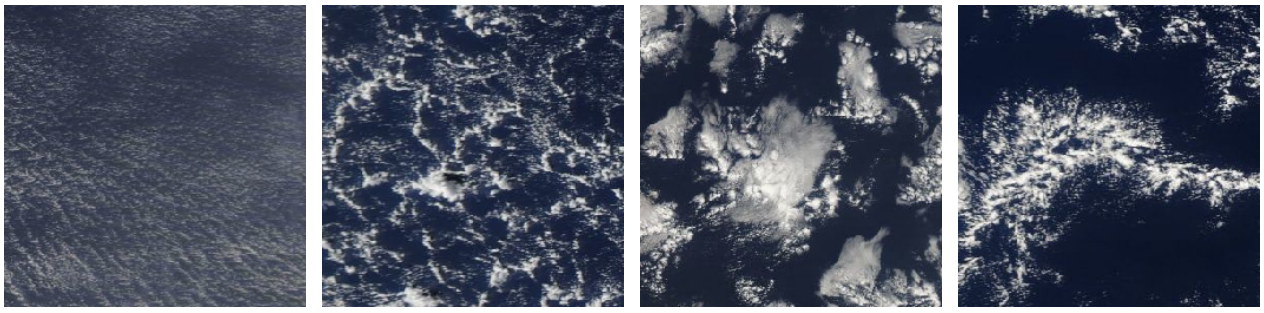
Because the objective of this thesis is to overcome the limitations of sparsely labelled cloud data, this study relies on two existing datasets generated through manual annotation. These datasets define distinct sets of cloud types across different geographic regions. All satellite imagery was acquired from the Moderate Resolution Imaging Spectroradiometer (MODIS) instruments, accessed via NASA's Earth Observing System[8]. While MODIS captures data across a wide range of spectral bands, this research focuses specifically on the visible spectrum (red, green, and blue wavelengths), named "true colour" band combination[9]. Utilising visible imagery ensures that the data processed by computational models directly corresponds to the visual information annotated by human labellers, establishing a robust and future-proof baseline for evaluation.

1.2.1 Sugar, Flower, Fish, and Gravel (SGFF)

To investigate the influence of deep convection on radiative heat loss, prior research has categorised distinct cloud regimes within the North Atlantic trade wind region. Previous studies have found the climatological importance of mesoscale cloud formations, which can span from 20 to 2,000 kilometres [6]. These large-scale formations modulate deep convection, thereby influencing atmospheric circulation and Earth's radiative

balance. To systematically categorise these formations, a panel of 12 atmospheric scientists identified four visually distinct and common mesoscale cloud patterns that account for approximately 90% of the observed cloud fields in the region [10].

- **Sugar**: Fine clouds evenly spread without large gaps or cloud-free regions, visible in 1a.
- **Gravel**: Granular cloud patterns with small clumps of cloud evenly spread over a meso-β scale of 20 to 100 kilometres, visible in 1b.
- **Flower**: Large clusters/bouquets of clouds separated from each other in a regular pattern over mese-β scale of 20 to 200 kilometres, visible in 1c.
- **Fish**: Large elongated series of clouds in (fish) skeletal structures on a meso α scale 200 to 2,000 kilometres, visible in 1d.



(a) Sugar (b) Gravel (c) Flower (d) Fish

Figure 1: The 4 mesoscale cloud patterns in SGFF classification and detection datasets

Initially, experts manually annotated 900 MODIS satellite images of the Northern Hemisphere, combined with radar cross-section analysis they found out that these clouds regimes were not only visually different but also had different impact deep convection. Despite expert involvement, the probability of four out of six labellers agreeing on a classification for a given image was only $p = 0.4$.

To expand this dataset for training large-scale deep learning models, Rasp et al. (2020) [11] initiated a crowdsourcing campaign via Zooniverse[12]. Rather than assigning a single global label to an image, participants were tasked with drawing bounding boxes around recognised cloud patterns within larger scenes. This approach facilitated localised object detection, enabling the implementation of architectures such as RetinaNet. Utilizing approximately 250 person-hours across 67 participants, the campaign generated 49,000 bounding box labels across 10,000 images.

As image classification models require standardised input dimensions we cannot use the raw bounding boxes. These bounding boxes were used to generate a high quality classfication set[13]. In this paper, the dataset was transformed into fixed-size 256 × 256 pixel image chips. To ensure high data quality, chips were exclusively

extracted from regions where a minimum of three independent labellers agreed on the cloud category. To correct for class imbalance the dataset was uniformly subsampled to 2,200 images per class giving us the following datasets of the mesoscale cloud patterns: Sugar Flower Fish Gravel.

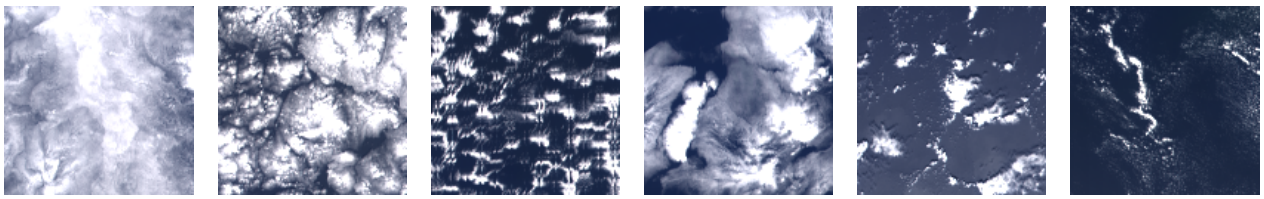
Table 1: Distribution of labels on SGFF dataset

Cloud type	Labels
Sugar	2,200
Gravel	2,200
Fish	2,200
Flower	2,200

1.2.2 Cumulus

As different regions of the earth have unique cloud types, there were other studies that tried to identify cloud patterns over the South-Eastern Pacific. The foundational classifications originate from Yuan et al. [14], who established an expert-labelled dataset using MODIS satellite imagery (128×128 pixel scenes). This initial dataset was generated entirely through manual annotation, again through the Zooniverse[12] platform where researchers were tasked to categorise each image under one of the cloud types below.

- **Solid Stratus:** Uniform cloud deck without any gaps or features, visible in 2a.
- **Closed MCC (Mesoscale Cellular Convection):** Continuous clusters of clouds forming a honeycomb structure with little to no gaps, visible in 2b.
- **Open MCC:** Inverse honeycomb cellular pattern with small clusters of clouds divided by larger gaps, visible in 2c.
- **Disorganised MCC:** Clusters of clouds that group together but lack any pattern, visible in 2d.
- **Clustered Cu (Cumulus):** Isolated clusters of clouds that are separated from any other cloud formation, visible in 2e.
- **Suppressed Cu:** Sparse, isolated longer stretches of clouds, visible in 2f.



(a) Solid Stratus (b) Closed MCC (c) Open MCC (d) Disorg MCC (e) Clustered Cu (f) Suppressed Cu

Figure 2: Examples of the 6 cloud types in the Cumulus datasets

Building upon these human-generated labels, subsequent research trained a residual neural network to automate and expand the classification[15]. To guarantee the correctness of this expanded dataset, the network's

classifications were verified using secondary Level-2 satellite cloud properties (e.g., cloud-top height, optical thickness) in combination with further validation checks by human labellers. This resulted in a dataset of with 38,756 labels during day and night time, which helps the field of atmospheric science but still required quite some human labelling. This dataset was used to train a CNN achieving a 91% accuracy in a supervised learning setting.

While the global dataset was shared in the thermal infrared channels to enable nighttime classification, this study leverages the provided metadata making it possible to retrieve the original granule with the corresponding coordinates of the data points. With this information the dataset was converted into RGB channels to ensure visual uniformity with the previously discussed SGFF dataset. To standardise network inputs, the 128×128 pixel chips were upscaled to 256×256 pixels. This dataset will be referred to as Cumulus for the rest of this paper.

Table 2: Distribution of labels on Cumulus dataset

Cloud type	**Labels**
Solid stratus	3,548
Closed MCC	6,277
Open MCC	3,345
Disorganized MCC	6,739
Clustered Cu	8,947
Suppressed Cu	9,900

2 Background

This chapter provides the theoretical foundation required for the rest of the thesis. First, it introduces the core concepts of machine learning. Second, it details the specific context of few-shot learning methods used in the subsequent experiments.

2.1 Multilayer Perceptron

A Multilayer Perceptron (MLP), or deep feedforward network, is the foundational deep learning architecture, inspired by the human brain connecting multiple neurons to form a neural network. This standard MLP form is constructed with an input layer, one or more hidden layers, and an output layer, as illustrated in Figure 3 [16].

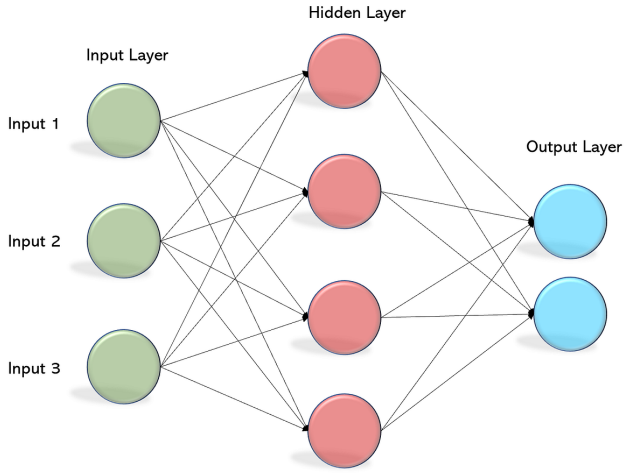


Figure 3: A Multilayer Perceptron architecture. Information flows sequentially from the input layer (left), through the hidden layer (middle), to the output layer (right). Each connection represents a learnable weight, and each node in the hidden and output layers applies a non-linear activation function.

Information flows through the network via forward propagation, where the output of one layer serves as the input to the next. Formally, the activations $\mathbf{a}^{(l)}$ for a given layer l are computed by taking a weighted sum of the previous layer's outputs, adding a bias, and applying a non-linear activation function [16]:

$$\mathbf{a}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}) \tag{3}$$

where $\mathbf{W}^{(l)}$ represents the weight matrix, $\mathbf{b}^{(l)}$ the bias vector, and σ the activation function. Without this non-linearity, stacking multiple layers would just generate a simple single linear transformation, but functions such as the Rectified Linear Unit (ReLU), defined as

$$\text{ReLU}(z) = \max(0, z) \tag{4}$$

enable the network to approximate complex, non-linear mappings [16]. Through this forward passing all the data accumulates at the output layer, generating a final prediction used to calculate the error against a

predefined loss function. Training the MLP requires finding the optimal weights and biases to minimise this loss, which can be approximated through backpropagation combined with a gradient-based optimisation algorithm [16]. Backpropagation utilises the intermediate signals from the forward pass to flow the error gradient in the opposite direction. By applying the chain rule of calculus layer by layer, it efficiently computes the gradient of the loss with respect to every parameter, indicating the direction of steepest increase in loss, which the optimizer then uses in reverse to minimize it [16]. A gradient descent optimiser uses these computed gradients to iteratively update the network's parameters, taking small steps to minimize the overall loss [16]. Because this process relies on incremental adjustments rather than a single calculation, the network requires multiple updates across many data points to converge on an optimal solution. Furthermore, as networks grow larger they represent increasingly complex functions. Optimizing this greater number of weights inherently requires significantly more training data to learn the underlying mappings accurately [17].

2.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is an architectural variant of the MLP explicitly designed for grid-structured data, such as images [18]. Standard MLPs require flattening the input, which discards spatial structure and introduces a prohibitively large number of parameters. CNNs overcome this using the convolution operator, which employs small, learnable weight matrices called kernels or filters. Instead of processing the entire input at once, a filter strides step-by-step across the image. At each spatial position, it computes a weighted combination of the local pixel values within its window, producing a single activation in a new feature map. In deeper layers, subsequent filters stride across these newly generated feature maps, combining simpler local patterns like edges into increasingly complex features [18].

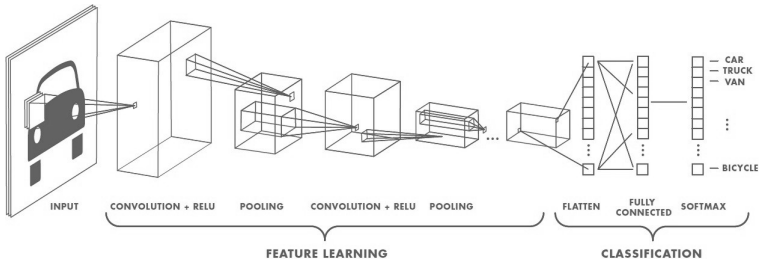


Figure 4: Overview of a typical CNN architecture used for image classification. Convolutional and pooling layers form an encoder where fully connected layers map the flattened latent space into classification output

A typical CNN architecture, shown in Figure 4, alternates these convolutional layers with pooling layers. Pooling layers progressively reduce the spatial resolution of the feature maps [18]. Allowing the model to extract spatial features whilst keeping the parameter count controllable. Together, the convolutional and pooling layers act as a feature extractor, collectively referred to as an encoder. This encoder distils the image into a dense, flattened feature array known as the latent space. This latent representation is then passed to fully connected layers to execute the model's final task, such as classification [19, 18].

2.3 Residual Networks

Increasing the depth of convolutional neural networks generally improves their representational capacity [20]. However, stacking more layers does not unconditionally improve accuracy. Beyond a certain depth, training error often increases due to degradation and the vanishing gradient problem [21]. In practice, these very deep architectures become exceedingly difficult to optimise.

Residual Networks (ResNets) address this limitation by introducing identity shortcut connections that skip one or more layers, as illustrated in Figure 5 [22]. Instead of forcing the network to learn a complete transformation of the data from scratch, a residual block is designed to learn only a small correction or residual denoted as $\mathcal{F}(\mathbf{x})$. As shown in the figure, this learned residual mapping is then added directly back to the original input $\mathbf{x}$, making the final output of the block $\mathbf{x} + \mathcal{F}(\mathbf{x})$.

This formulation significantly simplifies optimisation. These shortcut connections act as gradient highways during backpropagation, safely routing error signals to earlier layers without degradation. This enables the stable and effective training of significantly deeper architectures without introducing additional parameters or computational overhead [22].

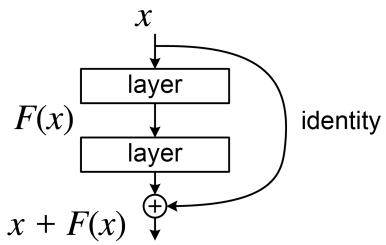


Figure 5: A residual building block. The shortcut connection adds the input $\mathbf{x}$ directly to the output of the stacked layers $\mathcal{F}(\mathbf{x})$. This formulation eases optimisation by allowing the network to learn only the residual correction rather than the full transformation.

2.3.1 ResNet-50

ResNet-50 is a 50-layer residual architecture structured into four sequential stages of bottleneck blocks with varying depths of $\{3, 4, 6, 3\}$ [22]. As illustrated in Figure 6, this bottleneck design mitigates the computational cost associated with deep networks by performing dimensionality reduction and restoration around the spatial convolutions. This limits the total parameter count to approximately 23 million whilst keeping a deep network.

Consequently, the combination of identity shortcut connections and bottleneck parameter efficiency enables stable optimisation without degradation. These characteristics establish ResNet-50 as a standard, highly capable feature-extraction backbone for complex visual tasks.

2.3.2 ImageNet Pretraining

In practice, ResNet-50 is typically pretrained on the ImageNet dataset, a large-scale corpus containing over 1.2×10^6 labelled images distributed across 1000 object categories [23]. This pretraining yields generic,

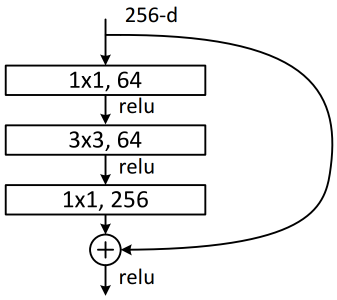


Figure 6: A bottleneck residual block used in ResNet-50. The block acts as a dimensionality reduction and restoration mechanism, making the spatial feature extraction computationally efficient, allowing for a deeper architecture.

hierarchical visual representations, progressing from low-level texture and edge detection in the initial layers to high-level semantic abstractions in the deeper layers.

These learned feature mappings are highly transferable to downstream computer vision applications [24]. Particularly in few-shot learning paradigms, ImageNet pretraining establishes a strong visual prior, substantially improving generalisation performance when optimising over a strictly limited subset of labelled target data.

2.4 Classification

In deep learning architectures, the feature extraction backbone (such as a CNN or ResNet) distils the input data into a dense latent representation. To perform the final task of categorisation, this latent vector must be mapped to discrete class probabilities.

2.4.1 Linear Classification

The most common approach to classification is applying a fully connected linear layer. Given a feature vector $f(\mathbf{x}) \in \mathbb{R}^d$ extracted from the backbone, a linear classifier computes the unnormalized logits z_i for each class i via a linear transformation:

$$z_i = \mathbf{w}_i^\top f(\mathbf{x}) + b_i, \tag{5}$$

where $\mathbf{w}_i \in \mathbb{R}^d$ is the class-specific weight vector and b_i is the bias term. This operation is essentially a dot product, meaning the final score depends on both the angle between the feature vector and the weight vector, as well as their respective magnitudes [16].

2.4.2 Cross-Entropy Loss

For multi-class classification tasks, the most commonly used objective function to optimise this linear mapping is the cross-entropy loss. Given a predicted probability distribution $\hat{\mathbf{y}}$ and a one-hot encoded target vector $\mathbf{y}$, the cross-entropy loss for a single sample is defined as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{K} y_i \log(\hat{y}_i), \tag{6}$$

where K denotes the number of classes.
The predicted probabilities are typically obtained using the softmax function:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \tag{7}$$

where z_i represents the logit corresponding to class i. The softmax function converts the logits into a valid probability distribution that sums to one. Cross-entropy loss penalises incorrect predictions while encouraging high confidence for the correct class. Due to its probabilistic interpretation and stable optimisation behaviour, it is widely used in deep learning classification tasks.

2.5 Prototypical Analysis

In few-shot learning scenarios, where data is exceedingly sparse, a common and effective approach is Prototypical Analysis. This method circumvents the need for gradient-based training or the complex optimisation of classifier weights often required by other paradigms. Instead, Prototypical Analysis shifts the classification problem from defining rigid decision boundaries to simple distance-based metric learning. This concept is illustrated in Figure 7.

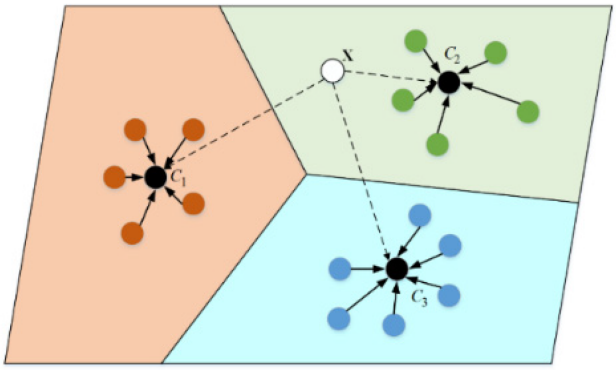


Figure 7: A visualisation of a 2D latent space divided through Prototypical analysis. The datapoints gets assigned the label of the closest centroid based on Euclidean distance.

As shown in the figure, the latent space is divided into distinct class regions. Each class is represented by a single prototype, calculated as the mean of its support examples (labeled C_1, C_2, and C_3 for the three classes). To classify a new, unlabeled data point, such as point X in the diagram, its distance is calculated relative to each of these prototypes. The point is then assigned to the nearest class. This fundamental shift to a dynamic distance-based approach makes it particularly suited for the sparse-data regime.

2.5.1 Distance Metrics

The standard straight-line distance between two points in Euclidean space is the most common metric used in baseline implementations. As depicted in Figure 7, the point X is evaluated based on its spatial proximity to the class centroids. If $\mathbf{x}$ and $\mathbf{c}$ are vectors representing the query point and a class prototype, respectively in an n-dimensional space, the Euclidean distance is given by:

$$d(\mathbf{x}, \mathbf{c}) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \tag{8}$$

While Euclidean distance measures spatial magnitude, an alternative approach is to evaluate the angular orientation between two vectors using cosine similarity. For a feature vector $f(x)$ and class-specific weight vector $\mathbf{w}_i$, the cosine similarity is defined as:

$$S(\mathbf{w}_i, f(x)) = \frac{\mathbf{w}_i^\top f(x)}{\|\mathbf{w}_i\| \|f(x)\|}. \tag{9}$$

By normalising both vectors, cosine similarity classification removes magnitude differences between classes and produces more balanced decision boundaries between base and novel categories.
The similarity scores are subsequently passed through a softmax function to obtain class probabilities:

$$p(y = i|x) = \frac{\exp(\tau \cdot S(\mathbf{w}_i, f(x)))}{\sum_j \exp(\tau \cdot S(\mathbf{w}_j, f(x)))}, \tag{10}$$

where τ denotes a scaling factor controlling the sharpness of the probability distribution [25], since the cosine similarity output is bounded between -1 and 1, the application of the softmax function often yields flat probability distributions

2.6 Cosine Similarity Classification

While non-parametric prototypical methods are demonstrated to be highly data-efficient and perform strongly by deterministically computing static class centroids, they limit their boundary expressiveness[26]. Parameterised approaches offer significant potential by overcoming this limitation. In this alternative architecture, prototypes are parameterised directly as the weight vectors ($\mathbf{w}_i$) of a cosine similarity classification layer, allowing them to be learned dynamically[27].
During training, a neural network maps input data into an optimised embedding space. Instead of explicitly averaging support set features, the dense feature representation is classified by measuring its cosine similarity against the learned weight vectors. By continuously updating these prototypes via standard backpropagation, this approach bridges the geometric clustering of prototypical analysis with the powerful optimisation of modern gradient methods [27].

2.7 Contrastive Learning

Contrastive learning is a representation learning paradigm where an encoder learns to map input samples into a latent space in which semantically similar datapoints are positioned close together, while dissimilar datapoints are separated as shown in Figure 8 [28, 29]. Rather than directly learning a decision boundary through classification, the objective is to structure the latent space itself. The formation of these compact clusters optimizes the feature space for separability, simplifying the objective for any classifier operating on the learned latent space.

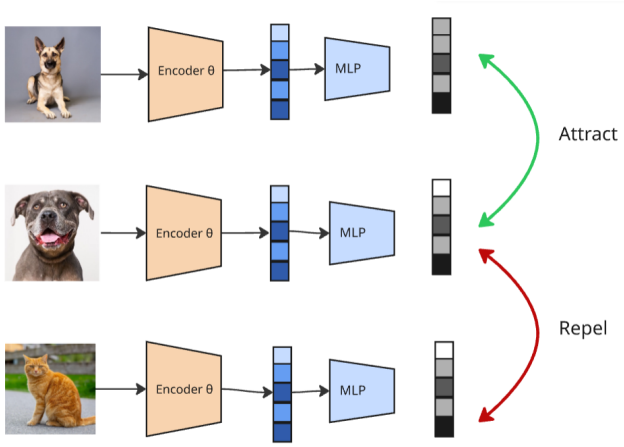


Figure 8: Visualization of the contrastive learning pipeline. The model is trained to map similar images (e.g., two different views of a dog) to the same area in the latent space, while pushing different images (e.g., a dog and a cat) far apart.

2.7.1 Supervised Contrastive Learning

Supervised Contrastive Learning extends this paradigm by explicitly using class labels to determine which feature embeddings should be attracted or repelled [29]. Samples sharing the same class label are defined as positive pairs, while samples from different classes form negative pairs. The encoder is therefore optimised to minimise intra-class variance while maximising inter-class separation within the latent space.

For a batch of feature embeddings, let $i \in I$ denote an anchor sample. The supervised contrastive loss is formulated as:

$$\mathcal{L}_{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\tau \cdot \mathbf{z}_i \cdot \mathbf{z}_p)}{\sum_{a \in A(i)} \exp(\tau \cdot \mathbf{z}_i \cdot \mathbf{z}_a)}, \tag{11}$$

where $P(i)$ denotes the set of positive samples sharing the same class label as sample i, $A(i)$ represents all samples excluding the anchor itself, $\mathbf{z}$ is the L_2-normalised embedding vector produced by the encoder, and τ is a temperature scaling parameter [29].

The numerator measures the similarity between embeddings belonging to the same class, while the denominator contrasts these similarities against all remaining samples in the batch. Consequently, the optimisation objective directly structures the latent space by pulling semantically related feature vectors together and pushing unrelated feature vectors apart.

2.7.2 Barlow Twins

Barlow Twins is a self-supervised contrastive learning framework that learns latent representations without requiring explicit negative pairs [30]. Instead of separating embeddings through pairwise repulsion, the method learns invariant feature representations by reducing redundancy between embedding dimensions.

Two distorted views are generated from the same input image and fed to two identical encoder-projector networks as seen in Figure 9. This ensures that the labels of these two images are the same whilst not

needing any labels. The resulting embeddings are then used to compute a cross-correlation matrix between both views as follows. Given two batches of projected embeddings z^A and z^B, the empirical cross-correlation matrix $\mathcal{C}$ is defined as:

$$\mathcal{C}_{ij} = \frac{\sum_{b=1}^{N} z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_{b=1}^{N} (z^A_{b,i})^2} \sqrt{\sum_{b=1}^{N} (z^B_{b,j})^2}}, \tag{12}$$

where N denotes the batch size and $\mathcal{C}_{ij}$ measures the correlation between embedding dimensions i and j. The Barlow Twins loss is subsequently defined as:

$$\mathcal{L}_{BT} = \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2, \tag{13}$$

where λ is a positive constant scaling the redundancy reduction term. The invariance term, $\sum_i (1 - \mathcal{C}_{ii})^2$, drives the diagonal elements to 1, encoding the correlation between 2 images. The redundancy reduction term, $\lambda \sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2$, drives the off-diagonal elements to 0. This is to counterfit the fact that there is no dissimilarty training against other classes. Forcing the information in a smaller space forces the model to find general commonalities instead of learning the transformations[30].

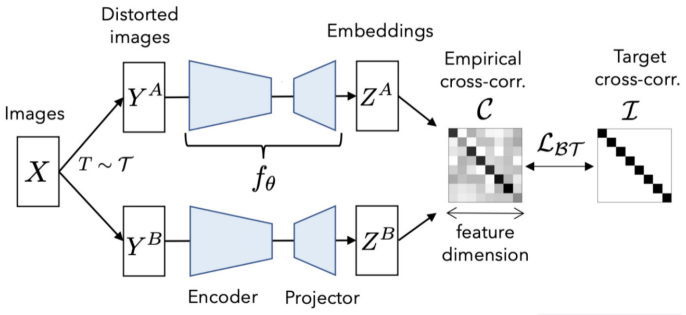


Figure 9: Visualisation of the Barlow Twins pre-training pipeline. Two distorted versions of a single image are passed through the siamese network where the model forces full correlation on the diagonal of the cross correlation matrix.

Following the self-supervised pre-training phase, the projector network is discarded. Only the encoder is retained to extract representations for downstream tasks. This framework can be applied to various domains. For example, in remote sensing, it is applied to self-supervised cloud classification, mapping satellite imagery into feature embeddings that encode mesoscale cloud morphologies without requiring large manually annotated datasets[31].

2.8 Few-Shot Learning

Few-shot learning (FSL) is a field in machine learning that aims to train models that are able to generalise with only a limited number of labelled examples [32]. Because only a limited number of labelled samples are available, directly training deep neural networks on $\mathcal{S}_{novel}$ typically results in severe overfitting. Few-shot learning approaches thus often rely on prior knowledge acquired from other sources of data before adapting to the novel classes [32]. Two dominant paradigms have emerged to address this problem: transfer learning

and meta-learning. As this thesis focuses on few-shot classification, these approaches are explained in a classification context.

2.9 Transfer Learning

Transfer learning is a methodology that relies on transferring knowledge acquired from a source domain $\mathcal{D}_S$ with abundant data to improve learning in a target domain $\mathcal{D}_T$ where data is scarce [33]. In the context of few-shot image classification, this is typically executed in two sequential phases.
First, during the pretraining phase, an encoder is trained on a large dataset containing base classes $\mathcal{C}_{base}$, which are strictly disjoint from the novel classes ($\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$). The objective here is to learn a global, highly discriminative latent representation space.
Second, during the fine-tuning or inference phase, this acquired knowledge is transferred to the N-way K-shot target task. To mitigate overfitting on the limited $\mathcal{S}_{novel}$, the parameters of the pretrained encoder are typically frozen to preserve the generalized feature extraction. Only a simple task-specific classification head is trained using the K available support examples. If the encoder has successfully captured the underlying semantic structure of the visual data during pretraining, the classifier can rapidly adapt to predict $\mathcal{C}_{novel}$ with minimal gradient updates.

2.10 Meta-Learning

Meta-learning approaches few-shot learning from a different perspective by explicitly optimising a model to rapidly adapt to unseen tasks [34]. Rather than only learning transferable feature representations, the objective is to learn a general adaptation strategy capable of solving new tasks using minimal supervision.
To achieve this, training is performed episodically. Each episode simulates an independent few-shot task sampled from the base dataset. Similar to the final evaluation setting, every episode is divided into a support set $\mathcal{S}_{novel}$ and a query set Q. The support set is used for temporary adaptation, while the query set evaluates how effectively the model generalises after observing only a few labelled examples.
By repeatedly optimising over many episodic tasks, the model learns parameters that generalise efficiently to novel class distributions. Consequently, meta-learning aligns the training procedure directly with the downstream few-shot evaluation setting.

2.10.1 Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (MAML) is an optimisation-based meta-learning algorithm designed to learn model parameters that can rapidly adapt to new tasks through only a few gradient updates [35]. The objective is therefore not to optimise directly for a single task, but to learn an initial parameter configuration that is highly adaptable across many tasks.
This meta-optimisation process is structured in two stages for each sampled episodic task $\mathcal{T}_i$. First, in the inner loop, a gradient-based classification model performs a small set of supervised learning steps. Using the K labelled examples per class in the support set $\mathcal{S}_{novel}$, the model evaluates the task-specific classification loss $\mathcal{L}_{\mathcal{T}_i}$ and updates its initial weights θ accordingly. For a single gradient update with an inner-loop learning

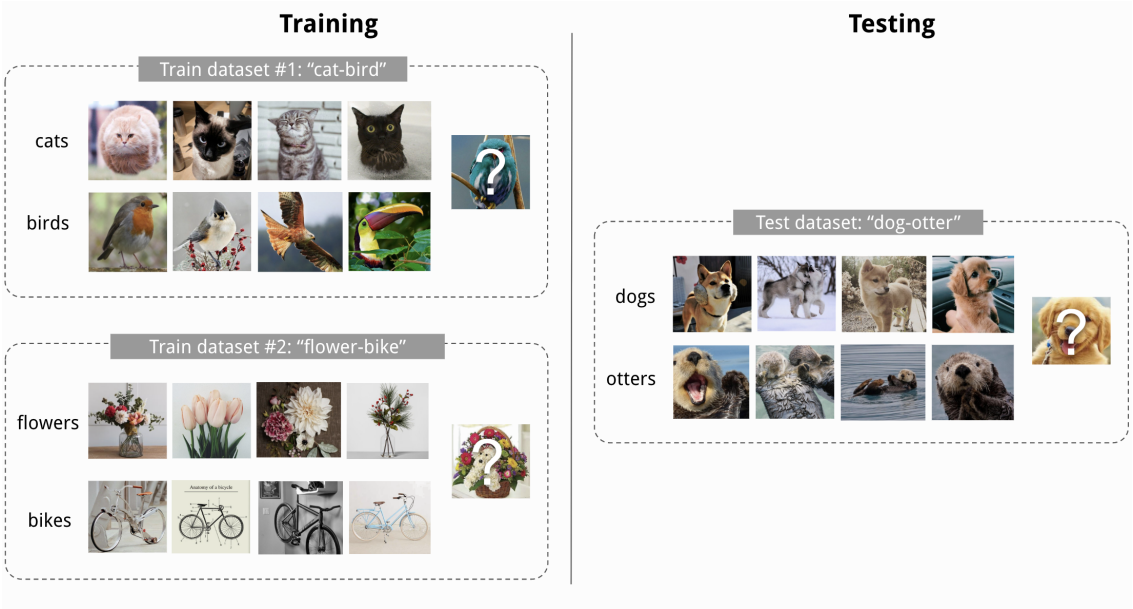


Figure 10: Episodic meta-learning setup. Each training episode simulates a small few-shot classification task containing a support set $\mathcal{S}_{novel}$ and a query set Q. The model repeatedly learns to adapt to new tasks using only the limited support samples.

rate α, this adaptation produces task-specific parameters θ_i':

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta), \tag{14}$$

where f_θ represents the model parameterised by θ.

Second, in the outer loop, the algorithm evaluates the effectiveness of these inner-loop updates. The model uses the newly adapted parameters θ_i' to make predictions on the task's distinct query set Q. The resulting loss dictates how the starting weights θ should be adjusted to improve future adaptation. Using a meta-learning rate β, the outer-loop meta-update is computed across a distribution of tasks $p(\mathcal{T})$:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}). \tag{15}$$

By continuously adjusting the global initialisation based on the generalisation performance of the temporarily adapted weights, MAML explicitly trains the network to converge quickly and accurately on novel classes.

2.10.2 Metric based learning

While optimisation-based methods like MAML focus on rapid parameter adaptation, metric-based meta-learning aims to construct a highly discriminative embedding space, similarly to prototypical analysis. The core principle is that if a neural network can learn a sufficiently robust feature representation, novel classes can be distinguished using non-parametric distance metrics, removing the need for any task-specific gradient updates.

Prototypical Networks [26] formalise this approach within the episodic training framework. The inner model

uses the support set $\mathcal{S}_{novel}$ strictly to establish geometric class representations. The prototypes $\mathbf{c}_k$ are computed with a static function during the inner loop and are thus very cheap to evaluate.
The meta-learning objective is then formulated by evaluating the generalisation on the query set Q. For a given query instance $\mathbf{x}_q$, the model computes its distance which is converted into the probability density as described in Equation 10 During the meta-training phase, the network is optimised by minimising the negative log-probability of the true class k for all query samples across the episode:

$$\mathcal{L}_{\text{proto}} = -\log p(y = k \mid \mathbf{x}_q). \tag{16}$$

Minimising this episodic loss explicitly trains the latent space again to tightly cluster features around their class prototypes. This is the same the cosine similarity classifier only now optimised for many different class combinations allowing for more generalised solutions across many tasks.

3 Methodology

The following chapter outlines the technical framework designed to solve the problem of few-shot cloud classification. First, the foundational network architectures and data augmentation methods are defined. Second, the representation learning objectives used to optimise the feature space are described. Third, the two distinct classification heads are introduced. Finally, these individual components are combined into four specific training paradigms: pure few-shot optimisation, in-domain transfer learning, out-of-domain transfer learning, and episodic meta-learning.

3.1 Encoder Architectures

All learning strategies rely on the ResNet architecture family [22]. ResNet is selected over domain-specific models pre-trained on satellite imagery because the latter often require multi-channel inputs, whereas the chosen datasets rely on standard RGB images. Allowing the model to make use of large image datasets that are made of RGB images. While Vision Transformers and self-supervised ResNets tailored to remote sensing were considered, Vision Transformers proved prohibitively large for extensive retraining. Furthermore, supervised ImageNet pre-training has been shown to outperform self-supervised remote sensing models on downstream tasks once fine-tuning is applied [36], making it the optimal foundation for both the in-domain and out-of-domain training pipelines

ResNet is selected over domain-specific satellite models because it natively processes standard RGB images, enabling the model to leverage massive pre-training datasets like ImageNet[23]. While Vision Transformers and self-supervised remote sensing models were also considered, the former proved prohibitively large for extensive retraining, which is not possible in few-shot learning setting. Furthermore, supervised ImageNet pre-training has been shown to outperform self-supervised remote sensing models on downstream tasks once fine-tuning is applied [36]. Consequently, this architecture provides the optimal foundation for both the in-domain and out-of-domain training pipelines.

The ResNet topology is a standard structure that provides high classification accuracy across many domains [13]. Every used network replaces the standard classification layer with a projection head. The projection head maps the extracted features into a 2048-dimensional latent space. Fixing the latent dimensionality across all tests ensures that variations in classification accuracy stem strictly from the chosen learning objective.

The pure few-shot classification methods use a scaled-down ResNet architecture. As training on limited data causes instability when updating a large number of parameters. The few-shot encoder mitigates instability by keeping the same depth but reducing the kernel sizes. This generates a model with a learnable parameter count of 4,298,624.

All the other experiments that use additional data other than the support set $\mathcal{S}_{cls}$ use the standard ResNet-50 topology. The standard ResNet-50 topology contains 23,557,888 learnable parameters. Loading ImageNet prior weights establishes a comparative baseline to evaluate out-of-domain transfer learning accuracy.

3.2 Image augmentation

To mitigate data scarcity before the images reach the encoder, each datapoint undergoes transformation per epoch. Artificial data generation forces the network to learn invariant properties of the target classes rather than memorising the exact pixel layouts of the limited support set. The transformation pipeline is applied sequentially to every input image across all evaluated methods. The specific transformations are defined as follows:

- **Spatial Jitter:** The image undergoes a random circular spatial shift (roll) along both the vertical and horizontal axes by a maximum of 5 pixels.
- **Rotation:** The image is randomly rotated by an angle of 90, 180, or 270 degrees with a probability of 0.75.
- **Vertical Flip:** The image is inverted along the horizontal axis with a probability of 0.5.
- **Horizontal Mirror:** The image is inverted along the vertical axis with a probability of 0.5.
- **Contrast Adjustment:** The image contrast is scaled by a random factor uniformly sampled from the range of 0.5 to 1.5.
- **Brightness Adjustment:** The image brightness is scaled by a random factor uniformly sampled from the range of 0.5 to 1.5.

Applying geometric transformations like rotation and mirroring produces physically valid representations that remain strictly within the target label space of the cloud domain. Because the orientation of the Earth relative to the satellite sensor is arbitrary, the rotational augmentations simulate genuine observational variance [37]. Structuring the augmentation pipeline strictly around these physical realities prevents the generation of out-of-domain edge cases.

3.3 Representation Learning Objectives

To optimise the latent space extracted by the ResNet encoder, three representation learning objectives are evaluated. During this optimisation phase, each method passes the latent features through a temporary projection or classification head to compute the loss. Once the encoder is optimised, these task-specific heads are discarded, exposing the final encoder layer representing the latent space for downstream few-shot classification.

3.3.1 Standard Supervised Baseline

The first objective establishes a standard supervised baseline. The ResNet encoder is appended with a standard Multi-Layer Perceptron (MLP) as a classification head. The network is optimised using the Cosine Similarity loss, mapping the extracted features directly to the available class labels. Once training is complete, the MLP head is discarded. This approach provides a control metric for traditional supervised classification, allowing for a direct measurement of the performance of the other proposed methods.

3.3.2 Supervised Contrastive Learning (Sup-Con)

The second objective is Supervised Contrastive Learning (Sup-Con). The Sup-Con loss, $\mathcal{L}_{sup}$, utilises label information to group samples of the same class together while pushing apart samples from different classes within a batch[29]. Cloud classes exhibit high intra-class variance, which complicates the formation of distinct decision boundaries when the training data is constrained to k samples. Sup-Con addresses this by enforcing intra-class compactness whilst maximizing inter-class seperation. During optimization, the 2048-dimensional features are passed through a non-linear projection network to a lower-dimensional space where $\mathcal{L}_{sup}$ is calculated. Evaluating this objective tests whether enforcing explicit spatial clustering and strict class separation prevents feature overlap in few-shot regimes.

3.3.3 Barlow Twins

The third objective evaluates the Barlow Twins method. Rather than relying on positive and negative sample pairs, the Barlow Twins loss, $\mathcal{L}_{\mathcal{BT}}$, focuses on attraction learning bringing the same classes together[30]. Sup-Con depends on batch composition to provide sufficient negative samples, which becomes a constraint in few-shot settings. Barlow Twins bypasses this requirement by driving the empirical cross-correlation matrix of the feature representations toward the identity matrix. Maximizing the diagonal elements enforces similarity between augmented views of the same image, while minimizing the off-diagonal elements decorrelates the individual feature dimensions. This prevents representational collapse without explicitly forcing inter-class separation [30]. Similar to Sup-Con, $\mathcal{L}_{\mathcal{BT}}$ is computed on the output of a temporary projection head, which is removed after the pre-training phase.

3.4 Classification Strategies

After the optimisation of the representation learning objectives, the temporary projection layers are discarded, and the encoder network is frozen. The remaining task is to map the optimised latent space to discrete cloud classifications using only the available k support samples. To evaluate this mapping, two distinct classification heads are tested: a non-parametric Prototypical classifier and a parameterised Cosine classifier.

3.4.1 Prototypical Classification Head

The first classification strategy evaluates a non-parametric Prototypical head. The class prototype is computed by calculating the Euclidean mean vector of the k transformed support images. Query images are then classified by measuring their Euclidean distance to each prototype, assigning the label of the nearest class mean [26]. With limited labels available, a new neural layer is prone to overfitting. By eliminating gradient-based updates, the Prototypical head reduces the risk of overfitting common in few-shot scenarios and instead provides a direct analysis of the latent space.

3.4.2 Cosine Similarity Head

The second classification strategy evaluates a parameterised Cosine Similarity head. This head acts as a trainable extension of the Prototypical approach by using learnable weight vectors as dynamic class prototypes. During the few-shot adaptation phase, these dynamic prototypes are optimised via gradient descent using the k support samples. To compute the classification, the network normalises both the 2048-dimensional feature vectors and the learned prototypes to calculate the angular distance. We use a standard architecture with two fully connected hidden layers across all tests, resulting in an average of 2.6 million learnable parameters across both datasets, depending on the final classification layer. Using parameterised cosine similarity allows for effective feature mapping in complex few-shot regimes [38]. However, this flexibility introduces a higher risk of overfitting and increased computational cost compared to the non-parametric Prototypical head. Contrasting these two approaches reveals whether the initial latent space can adequately separate cloud classes, or if explicit parameter adaptation is capable of generating more stable and complex decision boundaries.

3.5 Training Pipelines

The final component of the methodology defines the training pipelines. These pipelines specify the exact sequence of data presentation and parameter optimisation, dictating how the previously defined representation learning objectives and classification heads are applied. To evaluate the impact of prior knowledge on classification accuracy, three distinct data paradigms are established: pure few-shot optimisation, in-domain transfer learning, and out-of-domain transfer learning.

3.5.1 Pure Few-Shot Optimization

The first paradigm evaluates representation learning strictly from scratch, utilising only the target novel classes C_{novel}. The ResNet encoder is initialised with random weights, denoted as θ_{random}. During the first phase of this pipeline, the encoder and a temporary projection head are optimized jointly using exclusively the $k \times n$ labelled support samples from the target dataset. Following this optimisation, the temporary projection head is discarded, and the encoder parameters are frozen. In the second phase, the final classification head (Prototypical or Cosine) is attached to the latent layer and optimised using the exact same $k \times n$ support set.

3.5.2 In-Domain Transfer Learning

The second paradigm evaluates the integration of a domain-specific prior through transfer learning. The ResNet-50 encoder is initialised with random weights, but parameter optimisation is split sequentially across two disjoint datasets. First, the encoder and temporary projection head are pre-trained on either the full SGFF or Cumulus dataset. This phase yields network weights highly specialized to cloud morphologies, denoted as θ_{base}. The temporary head is then discarded. In the adaptation phase, the final classification head is attached, and the network is optimised on the target dataset using only the $k \times n$ support set. The objective is to minimize the loss $\mathcal{L}$ on the novel support set $\mathcal{S}_{novel}$, formally defined as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\mathcal{S}_{novel}; \theta_{base}). \tag{17}$$

Pre-training on a highly correlated source domain provides a structurally sound warm start. This paradigm tests the hypothesis that pre-structuring the latent space with domain-specific features reduces the required labels of the novel target classes.

3.5.3 Out-of-Domain Transfer learning

The third paradigm sequentially aggregates broad out-of-domain knowledge and specific in-domain knowledge before adapting to the target classes. Instead of starting from scratch, the ResNet-50 architecture is initialised with weights pre-trained on the external ImageNet dataset (θ_{ext})[23]. The network then proceeds through the exact same two-step pipeline as the in-domain transfer paradigm. It is first pre-trained on the large-scale atmospheric source dataset to yield hybrid weights, denoted as θ_{hybrid}, after which the temporary projection head is discarded. Finally, the network is adapted to the $k \times n$ target support set. The final optimisation step is formulated as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\mathcal{S}_{novel}; \theta_{hybrid}). \tag{18}$$

This setup tests the advantage of using a foundational model to provide general features that the limited adaptation data would otherwise be insufficient to teach.

3.5.4 Episodic Meta-Learning Frameworks

The final training paradigm evaluates meta-learning. While this paradigm utilises the same large-scale atmospheric source dataset as the in-domain transfer pipeline, it alters the data routing. Instead of processing data in global batches to extract generalised features, the source data is restructured into simulated n-way, k-shot episodes [39]. By forcing the model to solve simulated few-shot tasks during pre-training, the network explicitly optimises for rapid adaptation. Within this paradigm, two strategies are evaluated: metric-based and optimisation-based meta-learning.

The metric-based category evaluates Prototypical Networks [26]. Conceptually, this framework encapsulates the standard supervised representation objective and the Cosine Similarity head within the episodic pre-training loop. For the Prototypical network the objective is to find an optimal global weight initialisation, θ_{meta}, that is structured for distance-based classification. During the final few-shot adaptation phase, the encoder's parameters remain completely frozen. Mirroring the pure few-shot strategy, the target support set is processed in a purely feed-forward manner to calculate class centroids. This evaluation examines whether meta-learning via episodic simulation can produce a different separable latent space compared to task-specific gradient optimisation.

In contrast, Model-Agnostic Meta-Learning (MAML) employs an optimisation-based adaptation strategy [35]. Like the metric-based approach, MAML seeks an optimal meta-initialisation, θ_{meta}. However, MAML explicitly executes an inner optimisation loop via stochastic gradient descent on the support set. The inner model is a standard Multi-Layer Perceptron (MLP) with a Cosine Similarity classification head to allow for

a fully gradient flow. Furthermore, to mitigate the immense computational cost of calculating second-order derivatives during the meta-update, a first-order gradient approximation is utilised. For the final adaptation on the target domain, the network updates its parameters starting from θ_{meta}, using the novel support set, $\mathcal{S}_{novel}$, and an inner-loop learning rate, α. The final adapted weights, θ^*, are computed as:

$$\theta^* = \theta_{meta} - \alpha \nabla_{\theta_{meta}} \mathcal{L}(\mathcal{S}_{novel}; \theta_{meta}). \tag{19}$$

This comparison evaluates whether the ability to adapt weights through task-specific gradients outperforms a static approach that relies solely on the model's existing latent modelling.

4 Results

In this chapter, we present the experimental results for the classification problem, structured to evaluate both data constraints and specific learning methodologies. The chapter is organised into three primary testing phases based on data volume: pure few-shot experiments, intermediate in-domain pre-training setups, and complete transfer learning configurations. Within each of these three phases, additional experimental value is provided by evaluating specific algorithmic approaches, such as classifier choices and meta-learning strategies. Finally the best performing configurations are tested for their scalability.

4.1 General Experimental Setup

A fixed set of experimental settings is shared across all phases to enable fair quantitative comparisons between methods.

All models are evaluated across support set sizes $k \in \{1,2,4,8,16,25\}$, spanning one-shot ($k = 1$) to more data-rich few-shot regimes ($k = 25$), capturing how each method scales as labeled data availability increases. A configuration is defined as a unique combination of representation learning objective, classifier type, and training pipeline, corresponding to a single row in the results Tables (4, 5, 9, 10, 12, 13). Within a configuration, the pre-trained model is identical across all values of k, reducing computational cost and ensuring that comparisons on the accuracy differences across k are fair. Each k-shot experiment is repeated five times to guard against outliers. All tables report the mean accuracy μ and variance σ^2:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{20}$$

where $N = 5$ and x_i is the classification accuracy of the i-th run.

Model evaluation is conducted on disjoint test sets not used for support or validation. For SGFF, the test set comprises $4 \times (2200 - k)$ images across 4 classes. For Cumulus, the test set comprises $6 \times (3345 - k)$ images across 6 classes.

4.2 Pure Few-Shot Experiments

For the first experimental setup, the aim is to classify cloud types using only k labelled samples per class, thereby evaluating the model's ability to learn in a pure few-shot environment. In this setting, both the encoder and classification head are optimised solely on the available $k \times n$ labelled samples, where n denotes the number of classes. The ResNet encoder is initialised with random weights, ensuring that all learned representations originate from the provided k data.

In order to improve generalisation and prevent overfitting, data augmentation as mentioned in chapter 3 is applied during both encoder optimisation and classifier training. Three representation learning objectives are evaluated: (i) a standard supervised Cosine Similarity loss (Standard), (ii) Supervised Contrastive Learning (Sup-Con), and (iii) Barlow Twins. As the same encoder architecture gets optimised with varied loss functions, the encoder will generate unique feature spaces.

After the training of the encoder, the network parameters are frozen. The same k-shot support set is then reused to construct and train the classification head. Two classification strategies are considered: (i) a non-parametric Prototypical classifier (Proto), where class prototypes are computed directly from the support set, and (ii) a parameterised Cosine Similarity classifier (Cos) implemented as an MLP, architecture details can be found in the appendix Table 16.

This results in six experimental configurations, defined by the combination of representation learning objective and classifier type (e.g., Sup-Con Proto, Barlow Cosine).

All experiments are conducted using a fixed set of hyperparameters, summarised in Table 3.

Table 3: Standardised hyperparameters for pure few-shot experiments

Parameter	**Value**
Architecture	Custom ResNet (Table 14)
Encoder Loss Objectives	Standard (Cosine), Sup-Con, Barlow Twins
Encoder Train Epochs	100
Optimizer	Adam
Learning Rate	10^{-4}
Classifier Heads	Prototypical, Cosine Linear Layer
Epoch Cosine head	50
Learning rate Cosine head	10^{-3}
Cosine classifier τ	20
Batch Size	$\min(n*k, 32)$
Encoder State during Head Training	Frozen

Table 4: Mean classification accuracy on the SGFF dataset under Pure Few-Shot settings.

Configuration	**1**	**2**	**4**	**8**	**16**	**25**
Standard Cosine	41.47 ± 1.04	40.32 ± 0.95	48.06 ± 0.12	48.13 ± 0.98	48.45 ± 0.61	53.28 ± 1.34
Standard Proto	$\mathbf{45.24} \pm 0.67$	46.91 ± 0.43	52.98 ± 0.15	51.62 ± 0.19	51.84 ± 0.47	53.21 ± 0.28
Sup-Con Cosine	36.83 ± 0.91	38.56 ± 0.84	40.82 ± 0.55	45.79 ± 0.63	44.31 ± 0.58	48.64 ± 0.52
Sup-Con Proto	38.97 ± 0.54	$\mathbf{47.53} \pm 0.41$	51.85 ± 0.49	50.18 ± 0.16	51.82 ± 0.13	54.15 ± 0.17
Barlow Cosine	31.84 ± 0.65	45.71 ± 0.88	42.19 ± 0.82	47.56 ± 1.03	49.42 ± 1.35	$\mathbf{65.97} \pm 0.04$
Barlow Proto	41.78 ± 0.52	46.93 ± 1.14	$\mathbf{55.26} \pm 0.57$	$\mathbf{54.34} \pm 0.18$	$\mathbf{58.51} \pm 0.26$	57.65 ± 0.49

Table 5: Mean classification accuracy on the Cumulus dataset under pure Few-shot settings.

Configuration	**1**	**2**	**4**	**8**	**16**	**25**
Standard Cosine	32.35 ± 0.31	38.64 ± 0.37	34.48 ± 0.92	42.91 ± 1.76	58.83 ± 0.14	64.12 ± 1.58
Standard Proto	$\mathbf{41.67} \pm 0.38$	40.72 ± 0.54	42.29 ± 0.93	47.53 ± 0.75	47.16 ± 0.42	43.14 ± 0.29
Sup-Con Cosine	33.51 ± 0.35	34.28 ± 1.72	37.24 ± 0.78	37.59 ± 0.74	38.93 ± 0.71	46.06 ± 0.69
Sup-Con Proto	40.38 ± 0.22	$\mathbf{43.51} \pm 0.27$	37.65 ± 0.46	44.32 ± 0.49	43.74 ± 1.13	43.37 ± 0.81
Barlow Cosine	35.32 ± 0.26	39.76 ± 0.45	40.91 ± 0.74	46.38 ± 0.33	$\mathbf{63.15} \pm 0.18$	$\mathbf{68.52} \pm 0.07$
Barlow Proto	38.45 ± 0.43	43.29 ± 0.24	$\mathbf{46.63} \pm 0.28$	$\mathbf{49.97} \pm 0.35$	52.92 ± 0.31	57.08 ± 0.39

4.3 Pure Few-Shot results

The mean classification accuracies of the pure few-shot tests on the SGFF and Cumulus datasets are presented in Table 4 and Table 5. Across all configurations, accuracy scales positively with the number of shots (k), with an average of 36.95 at $k = 1$ to 53.71 at $k = 25$. This trend confirms that additional support data consistently yields higher classification accuracy.

Standard and Supervised Contrastive (Sup-Con) representation learning methods generally yield lower accuracy relative to Barlow Twins, which achieves the highest accuracy in most scenarios for $k \geq 4$. A distinct accuracy shift also occurs between classification heads based on data volume. In low-shot regimes ($k \leq 8$), the Prototypical head consistently yields higher accuracy than the Cosine similarity head, with Barlow Proto achieving the highest scores. With increased data at $k = 16$ and $k = 25$, the Cosine similarity head achieves the highest overall accuracy, particularly when paired with the Barlow Twins objective.

A marginal difference in classification accuracy exists between the two datasets, with aggregate mean accuracies of 48.0 for SGFF and 44.3 for Cumulus. The SGFF dataset evaluates a 4-way classification task, whereas the Cumulus dataset requires 6-way classification. The lower accuracy on Cumulus directly reflects the inherently harder baseline probability of a 6-class problem, rather than a degradation in the models' ability to learn.

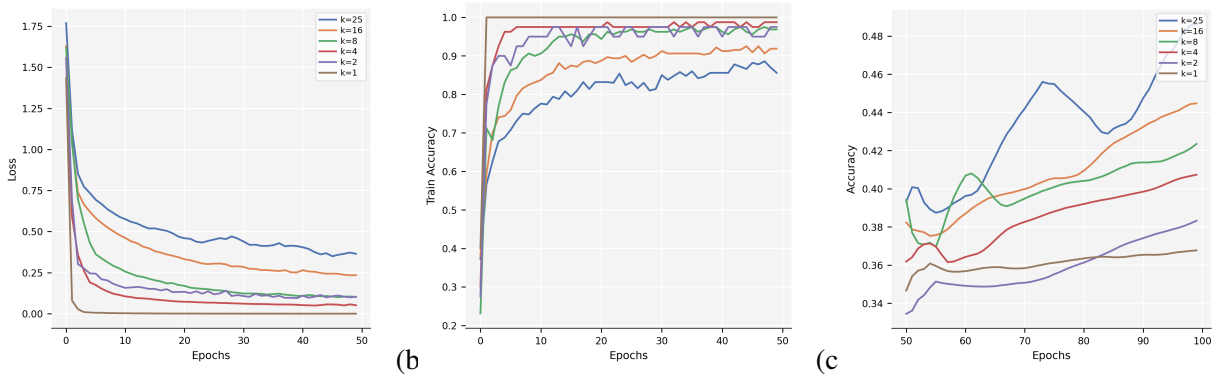


Figure 11: Training dynamics of the cosine similarity head for the Sup-Con configuration on the SGFF dataset. The panels display (a) training loss, (b) training accuracy, and (c) validation accuracy across varying support set sizes (k). While extreme low-shot regimes ($k = 1$) quickly achieve perfect training accuracy, they suffer from severe overfitting. Increasing the support set size (k) mitigates this memorisation effect, yielding lower training accuracy but significantly higher validation accuracy.

The training curves in Figure 11 plot the average performance of the classifiers' head adaptation in the configuration Sup-Con Cosine on the SGFF dataset, but represent trends observed across all Cosine Similarity experiments. As in all tests, a substantial accuracy gap is visible between the training and test metrics, with runs at $k = 1$ reaching a perfect training accuracy of 100% (Figure 11 a) within the initial epochs. As the number of shots k increases, this overfitting effect diminishes; training accuracy decreases while test accuracy (Figure 11 c) increases as the classifier is forced to generalise during training. Furthermore, while the loss plateaus early during training, test accuracy continues to improve across all shot counts, with the rate of increase being notably faster for larger values of k. Ultimately, larger support sets compel the model to learn generalizable features rather than memorise the limited training data.

4.4 In-Domain Pre-Training Experiments

Models are pre-trained on the complete SGFF or Cumulus dataset and subsequently adapted to the alternate dataset using a few-shot support subset of $k \times n$ samples. When pre-training on Cumulus (and adapting to SGFF), the pre-training phase utilises $n \times (2100 - 100) + k$ samples. Conversely, pre-training on SGFF (and adapting to Cumulus) utilizes $n \times (3345 - 100) + k$ samples. The other 100 samples are used as a validation set for prototypical analysis during training to generate plot Figure 12 c.

During the pre-training phase, the ResNet-50 architecture is optimised again with the Standard, SupCon, or Barlow Twins learning objectives. As the pre-training datasets provide significantly more data than the pure few-shot regime the hyperparameters utilise an increased learning rate as bigger batches allow for more stable gradients, but for a similar amount of epochs at 100 as visible in Table 6.

To maintain alignment with the pre-trained feature space, the encoder is fine-tuned using the exact same pre-training loss function. The learning rate and number of epochs are explicitly reduced during adaptation to prevent overfitting on the small k-shot support set. Following encoder adaptation, a non-parametric Prototypical classifier head performs the final classification. Selecting the Prototypical head isolates the evaluation strictly to the quality of the in-domain representations, removing variance caused by classifier optimisation. Further hyperparameters for these configurations can be seen in Table 6.

Table 6: Hyperparameters for standard in-domain transfer learning

Parameter	**Value**
Architecture	ResNet-50
Pre-training Datasets	SGFF, Cumulus
Pre-training Epochs	100
Pre-train Learning Rate	10^{-3}
Pre-training / Adaptation Optimizer	Adam
Batch Size	32
Encoder Loss Objectives	Standard (Cosine), SupCon, Barlow Twins
Adaptation Epochs	50
Adaptation Learning Rate	10^{-4}
Classifier Head	Prototypical
Encoder State during Adaptation	Unfrozen

Alongside standard transfer learning, two episodic meta-learning algorithms are evaluated: Prototypical Networks and Model-Agnostic Meta-Learning (MAML) utilizing a standard MLP with Cosine Similarity head in the inner loop with a first-order gradient approximation. The meta-learning methods leverage the exact same pre-training datasets and data volumes as the standard baselines, but restructure the training data into episodes to optimise for rapid adaptation. To maintain episodic variance while ensuring comparability with the baseline tests, the number of classes per episode (n-way) depends on the pre-training dataset. Pre-training on Cumulus utilises a 4-way setup to perfectly match the SGFF evaluation classes. Conversely, pre-training on the 4-class SGFF dataset utilises a 3-way setup. Dropping the n-way size below the total class count ensures the base classes change between episodes, forcing the meta-learning algorithms to generalise across varying class distributions. All episodes utilise 15 query samples ($n_{query} = 15$) per class and employ identical

data augmentations as the standard transfer learning configurations. Further parameter configuration can be seen in Table 7.

Table 7: Hyperparameters for episodic meta-learning configurations

Parameter	**Value**
Architecture	ResNet-50
Algorithms	Prototypical Networks, MAML
n-way (Cumulus Pre-training)	4-way
n-way (SGFF Pre-training)	3-way
Query Samples (n_{query})	15
Training Episodes	2000
Optimizer	Adam
MAML Inner Loop	Cosine Similarity, 1st-order gradient
Meta-Learning Rate	10^{-4}
Inner Loop Learning Rate (MAML)	10^{-2}
Inner Loop Steps (MAML)	1

Table 8: Accuracy of pre-trained encoders on in-domain data, analysed with non-parametric Prototypical head per

Latent Representation	**SGFF**	**Cumulus**
Standard	82.33	84.34
Sup-Con	84.92	86.72
Barlow	80.21	82.89

Table 9: Accuracy of in-domain approaches on SGFF dataset

Configuration	**1**	**2**	**4**	**8**	**16**	**25**
Standard	45.33 ± 0.18	$\mathbf{58.92} \pm 0.24$	58.54 ± 0.13	63.75 ± 0.03	70.66 ± 0.17	71.98 ± 0.11
Sup-Con	47.02 ± 0.02	52.66 ± 0.35	$\mathbf{61.75} \pm 0.11$	$\mathbf{67.41} \pm 0.06$	$\mathbf{70.74} \pm 0.04$	$\mathbf{73.69} \pm 0.09$
Barlow	40.01 ± 0.24	45.84 ± 0.05	52.51 ± 0.18	58.58 ± 0.11	63.47 ± 0.16	64.50 ± 0.14
Proto Network	42.19 ± 0.32	49.34 ± 0.29	52.71 ± 0.42	46.33 ± 0.51	51.12 ± 0.49	53.45 ± 0.67
MAML	$\mathbf{48.10} \pm 0.24$	48.90 ± 0.30	53.70 ± 0.29	52.80 ± 0.27	57.00 ± 0.63	57.40 ± 0.70

4.5 In-Domain Pre-Training Results

First, as shown in Table 8, the pre-training results demonstrate clear performance trends. The Cumulus dataset is consistently easier to classify, achieving higher accuracy across all latent space representations. Furthermore, when comparing the three representation learning methods, the Supervised Contrastive approach performs best, followed by the standard method, while the Barlow Twins objective yields the lowest performance.

Consistent with the pure few-shot experiments, classification accuracy positively correlates with the volume of available support data (k) across both the SGFF (Table 9) and Cumulus (Table 10) datasets. Furthermore, the in-domain pre-training configurations have a significantly increased accuracy over the pure few-shot

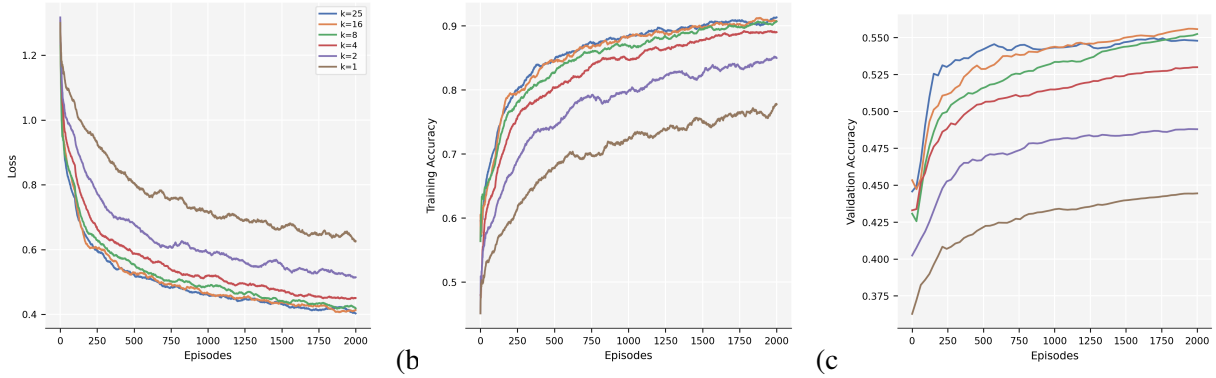
Table 10: Accuracy of in domain approaches on Cumulus dataset

Configuration	1	2	4	8	16	25
Standard	40.10 ± 0.15	$\mathbf{48.90} \pm 0.38$	$\mathbf{57.60} \pm 0.07$	$\mathbf{63.40} \pm 0.31$	$\mathbf{67.10} \pm 0.15$	$\mathbf{69.50} \pm 0.18$
Sup-Con	36.60 ± 0.24	43.70 ± 0.03	49.60 ± 0.92	57.90 ± 0.23	63.20 ± 0.02	64.10 ± 0.09
Barlow	31.40 ± 0.45	44.70 ± 0.08	52.70 ± 0.30	56.70 ± 0.14	64.10 ± 0.06	64.20 ± 0.01
Proto Network	40.47 ± 0.15	41.93 ± 0.12	45.91 ± 0.25	50.18 ± 0.47	52.26 ± 0.33	52.49 ± 0.24
MAML	$\mathbf{40.52} \pm 0.35$	44.68 ± 0.23	46.15 ± 0.27	48.24 ± 0.21	48.97 ± 0.36	50.32 ± 0.54

baselines, showing an average accuracy increase of 9.89 for the SGFF dataset and 11.30 for the Cumulus dataset for all k. Additionally, the low standard deviations across all configurations demonstrate that these approaches are highly stable and robust to the specific support samples selected.

When analysing the transfer learning approaches (Standard, Sup-Con, and Barlow Twins), performance is dataset-dependent. On the SGFF dataset, Sup-Con demonstrates the best scaling, consistently achieving the highest accuracy from $k = 4$ onwards and peaking at 73.69%. This is rather close to pre-training on this dataset, showing its few-shot capability. On the Cumulus dataset, the Standard configuration is the dominant approach, outperforming all other methods from $k = 2$ through $k = 25$. While Barlow Twins shows steady improvement as k increases, it lags behind both Standard and Sup-Con, and fails even to significantly outperform the pure few-shot Barlow-Cosine configuration.

In the extreme low-shot regime ($k = 1$), episodic meta-learning proves superior. The MAML configuration outperforms all standard transfer learning approaches at single-shot learning across both datasets. However, MAML and Prototypical Network suffer from a significant plateau effect as the support set grows. For example, on the SGFF dataset, Sup-Con accuracy increases by over 26% from $k = 1$ to $k = 25$, whereas MAML accuracy increases by less than 10%. A similar trend is observed on the Cumulus dataset, where transfer learning methods scale efficiently with additional data, ultimately leaving the meta-learning approaches far behind at higher values of k.



(a (b (c

Figure 12: Training dynamics of the Model-Agnostic Meta-Learning (MAML) with Cumulus as S_{base} and SGFF as S_{novel}. The panels display (a) training loss, (b) training accuracy, and (c) validation accuracy across varying support set sizes (k). The model is capable of using the increased support set for improved classification. As loss and accuracy stabilise after 2000 episodes, the model fails to generalise between the base and novel classes.

Figure 12 visualises the training dynamics of the Model-Agnostic Meta-Learning (MAML) configuration.

A substantial accuracy gap exists between the training accuracy (Figure 12b) and the validation accuracy (Figure 12c). The MAML algorithm achieves a high training accuracy of 90 at $k = 25$, yet the validation accuracy remains stagnant at a significantly lower value. The converging loss (Figure 12a) combined with the plateauing validation accuracy demonstrates that the MAML algorithm fails to generalise from the base classes to the novel classes. The in-domain meta-learning approach cannot effectively leverage additional support data, resulting in final accuracies that even fail to outperform the pure few-shot baselines.

4.6 Out-of-Domain Initialization

The final experimental phase evaluates the impact of large-scale, out-of-domain prior knowledge on few-shot cloud classification. The ResNet-50 encoder is initialised with weights pre-trained on the ImageNet dataset prior to the standard in-domain pre-training and adaptation phases. By starting with ImageNet weights, the model learns from 1.2 million generic images and the full cloud dataset before fine-tuning on just $(k \times n)$ support samples. Integrating massive out-of-domain datasets tests whether generic visual features provide a stronger foundational representation for few-shot adaptation than domain-specific learned features.
The ImageNet weights are applied identically to both the standard transfer learning and episodic meta-learning configurations. Beyond the initial weight loading, the training procedure mirrors the previous in-domain methodology. The encoder is optimised on the full cloud dataset using the Standard Cosine Similarity, Supervised Contrastive, and Barlow Twins objectives, followed by few-shot adaptation and Prototypical classification on the target dataset. Maintaining an identical downstream training pipeline isolates the effect of the ImageNet prior, ensuring any downstream accuracy improvements stem strictly from the out-of-domain initialisation.

Table 11: Accuracy of pre-trained encoders with loaded ResNet weights and in-domain training, analysed with non-parametric Prototypical head per

Latent Representation	**SGFF**	**Cumulus**
Standard	84.74	85.28
Sup-Con	86.02	88.11
Barlow	82.67	84.53

Table 12: Accuracy of out-of-domain SGFF

Configuration	**1**	**2**	**4**	**8**	**16**	**25**
Standard	**54.71** $\pm$ 0.45	65.34 $\pm$ 0.10	**70.93** $\pm$ 0.13	**74.12** $\pm$ 0.13	**77.08** $\pm$ 0.02	77.65 $\pm$ 0.02
Sup-Con	51.73 $\pm$ 1.09	**65.42** $\pm$ 0.27	69.07 $\pm$ 0.26	73.28 $\pm$ 0.09	75.21 $\pm$ 0.01	**78.89** $\pm$ 0.02
Barlow	43.48 $\pm$ 0.59	51.65 $\pm$ 0.24	59.72 $\pm$ 0.05	68.04 $\pm$ 0.05	71.27 $\pm$ 0.05	72.03 $\pm$ 0.02
Proto Network	41.72 $\pm$ 0.29	52.91 $\pm$ 0.13	53.95 $\pm$ 0.18	60.56 $\pm$ 0.02	64.34 $\pm$ 0.03	66.31 $\pm$ 0.08
MAML	45.06 $\pm$ 0.02	50.18 $\pm$ 0.06	52.54 $\pm$ 0.10	55.39 $\pm$ 0.13	59.13 $\pm$ 0.05	61.74 $\pm$ 0.12

Table 13: Accuracy of out-of-domain approaches on Cumulus dataset

Configuration	1	2	4	8	16	25
Standard	53.27 ± 1.07	$\mathbf{65.24} \pm 0.66$	$\mathbf{72.41} \pm 0.25$	76.23 ± 0.05	$\mathbf{85.78} \pm 0.13$	85.04 ± 0.01
Sup-Con	$\mathbf{56.45} \pm 0.45$	62.73 ± 0.03	70.37 ± 0.09	$\mathbf{78.52} \pm 0.08$	84.59 ± 0.04	$\mathbf{86.95} \pm 0.02$
Barlow	40.06 ± 0.35	52.62 ± 0.09	62.69 ± 0.06	74.54 ± 0.05	79.61 ± 0.02	81.28 ± 0.01
Proto Network	45.73 ± 0.06	51.08 ± 0.14	54.85 ± 0.19	56.01 ± 0.12	58.27 ± 0.23	55.44 ± 0.07
MAML	46.37 ± 0.04	49.62 ± 0.08	53.51 ± 0.13	54.19 ± 0.05	55.34 ± 0.02	54.76 ± 0.17

4.7 Out-of-Domain Results

Initialising the ResNet-50 encoder with out-of-domain ImageNet weights increases classification accuracy compared to in-domain pre-training across both datasets. While this improvement is visible even in the extreme low-shot regime ($k = 1$)—for example, the Sup-Con approach on the Cumulus dataset increases from 36.60% to 56.45%—the absolute performance at single-shot remains low. To achieve competitive and practically useful performance, a larger support set (e.g., $k \geq 8$) is still required, as demonstrated in Tables 12 and 13. At higher shot counts ($k = 25$), Sup-Con ultimately achieves the highest classification accuracies on both datasets, reaching 78.89% on SGFF and 86.95% on Cumulus. Comparing these results to the pretrainig in Table 11 the few-shot training methods achieve results comparable to in-domain pre-training. Notably, the minimal improvement over pre-training suggests that the full dataset inherently contains enough samples for the models to learn generalized feature extraction.

The introduction of out-of-domain weights also alters the relative performance of the tested methods at $k = 1$. Under in-domain pre-training, the MAML configuration yielded the highest single-shot accuracy; however, with ImageNet initialisation, standard transfer learning approaches (Standard, Sup-Con) achieve superior initial accuracy. As the support set grows, these standard transfer learning methods, including Barlow Twins, start outperforming the meta configurations by larger margins.

The episodic meta-learning models (Prototypical Network, MAML) demonstrate an inability to leverage the out-of-domain weights as k increases. Furthermore, these meta-learning approaches do not show a significant overall performance boost when transitioning from in-domain to out-of-domain initialisation. Across both datasets, the accuracy of the meta-learning configurations plateaus early. On the Cumulus dataset, performance even unexpectedly decreases at higher shot counts. This indicates that while standard transfer learning benefits substantially from additional support data, the episodic training structure impedes the effective fine-tuning of out-of-domain representations.
To further analyse the learned representations, we visualise the latent space using t-SNE projections [40] (Figure 13). As shown in Figure 13(a), the supervised pre-training successfully groups the data into separable clusters, confirming that the encoder learns discriminative features during this initial phase. However, these pre-trained representations cannot be directly applied to the target domain without adaptation. A notable exception is the "Sugar" class, which is immediately grouped in the corner, also reflected in its high accuracy within the SGFF confusion matrix (Figure 14a). Following adaptation (Figure 13c), the model

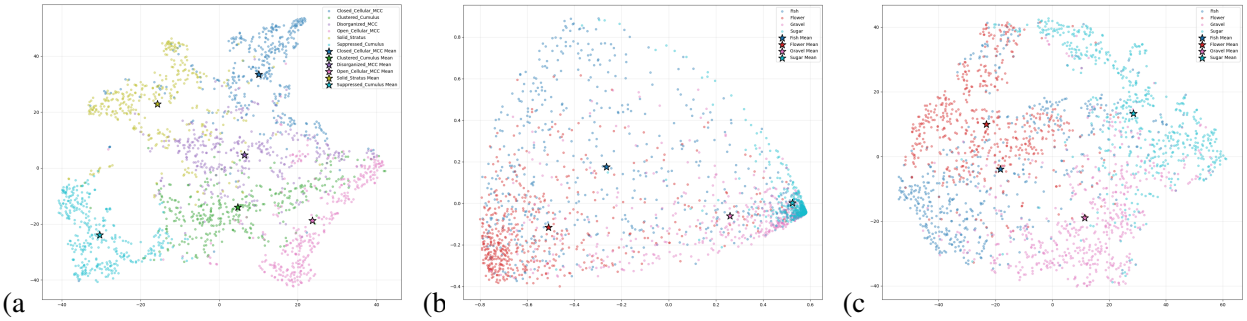


Figure 13: t-SNE plots of the latent space of a Sup-Con configuration on 25 k (2000 samples per plot) showing the model's adaptation process. The plots show (a) the pretraining data, (b) unseen SGFF data before adaptation, and (c) the SGFF data after adaptation. The supervised encoder clearly separates the Cumulus data (a), but does not directly transfer to SGFF (b). Adaptation generates a separable latent space for SGFF dataset (c).

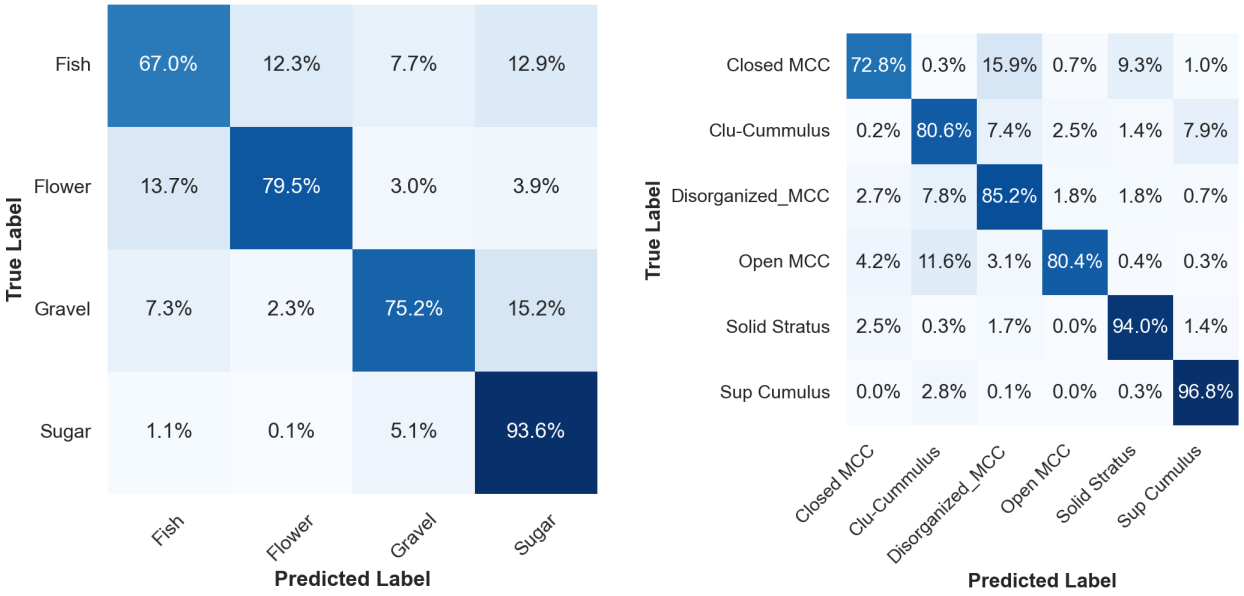


Figure 14: Confusion matrices for the Sup-Con configuration with a prototypical head ($k = 25$). Left: Evaluated on the SGFF dataset. Right: Evaluated on the Cumulus dataset.

successfully shifts the feature space, generating cleaner and clearly separable clusters for the entire SGFF dataset.

While the overall adaptation is successful, the confusion matrices (Figure 14) reveal that classification difficulty is not uniform across all morphological classes. In the SGFF dataset, the "Sugar" class is highly distinct and easily identified by the model, achieving an accuracy of 93.6%. Conversely, the "Fish" class proves much more challenging, achieving only 67.0% accuracy. A similar variance is observed in the Cumulus dataset, where the "Suppressed Cumulus" class reaches 96.8% accuracy, while the "Closed MCC" class drops to 72.8%. This suggests that certain cloud patterns are inherently more distinct, whereas others share overlapping visual features that complicate classification even at higher shot counts.

4.8 Scalability Analysis

To evaluate the scalability and upper performance bounds of the evaluated methods, the best-performing configuration from each of the three training paradigms—pure few-shot, in-domain pre-training, and out-of-domain initialisation was selected. For this evaluation, both the in-domain and out-of-domain configurations were adapted to use a parameterised Cosine Similarity classification head. All hyperparameters remain identical to those established in their respective baseline training pipelines. The primary objective of these tests is to see the full potential and to determine the volume of support data (k) required for these few-shot learning techniques to converge towards fully supervised accuracy levels.

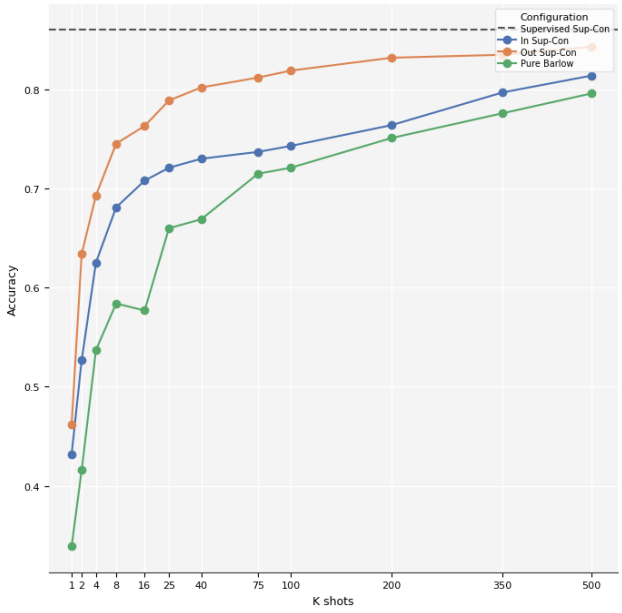


Figure 15: Performance comparison of the best configurations from the three training paradigms evaluated on the SGFF dataset across extended support set sizes (k). While all configurations exhibit substantial accuracy improvements in the low-data regime, performance gains become marginal beyond $k = 40$. Overall accuracy strongly correlates with the total volume of data: the pure few-shot paradigm yields the lowest performance, followed by in-domain pre-training, whereas out-of-domain initialisation achieves the highest accuracy, asymptotically approaching the fully supervised baseline.

As illustrated in Figure 15, classification accuracy depends strongly on data availability. While all config-

urations improve with larger sample sizes, their absolute performance is strictly ordered by the volume of prior training data. The out-of-domain configuration, pre-trained on ImageNet, consistently outperforms the other paradigms and comes close to the supervised baseline at $k = 500$. Furthermore, the accuracy curves plateau after $k = 40$, indicating that near-optimal classification is achievable with a relatively small subset of labelled samples. Ultimately, these results show that few-shot learning can significantly reduce the need for large annotated datasets.

5 Analysis

This section evaluates the results of the few-shot classification experiments. First, it compares parameter-free and parameterised classifier heads across different data sizes. Second, it analyses how self-supervised and supervised representation learning methods perform under varying pre-training conditions. Finally, it explains why standard transfer learning is better suited for novel cloud classes than episodic meta-learning.

5.1 Classifier heads

In low-data regimes ($k \leq 8$), the Prototypical classification head consistently achieves higher accuracy than the parameterised Cosine Similarity head. The Cosine head relies on iterative weight updates and multiple hyperparameters, which causes severe overfitting on small support sets. This is demonstrated by the perfect training accuracy paired with low validation accuracy at $k = 1$, which was demonstrated in Figure 11. In contrast, the Prototypical classifier requires no tuning and categorises novel images based solely on their distance to the geometric mean of the support samples. Because this parameter-free approach prevents training instability and provides a robust decision boundary across varying configurations, it proved to be the superior method for low-shot classification. Therefore, the Prototypical classification head was used in all the subsequent tests for the other in- and out-of-domain configurations.
However, the fixed geometric centroids of the Prototypical classifier limit its accuracy as the support set grows. As k increases, the Cosine Similarity head becomes competitive and ultimately achieves higher classification accuracy. The increased data prevents the overfitting seen in sparse regimes, allowing the parameterised layer to properly optimise. Consequently, the Cosine head uses these learned parameters to shape its decision boundaries around the actual data variance, rather than being locked to the simple geometric mean of the rigid Prototypical approach.

5.2 Representation learning

The superiority of the Barlow Twins objective at higher shot counts ($k \geq 16$) in the pure few-shot setting highlights a vulnerability in supervised representation learning when labels are scarce. In the Standard and Sup-Con objectives, relying heavily on a small and potentially unrepresentative support set can introduce noise into the learned representations. If the selected support images are not typical of their class, the supervised loss functions force the model to anchor its latent space around these outliers. This vulnerability is evident in the confusion matrices (Figure 14), which show high confusion between classes with overlapping visual features, such as 'Fish' and 'Flower', which are hard to distinguish for the best model and even expert scientist who made the dataset[11]. This shows that sampling an unrepresentative support set is possible in the cloud domain. This high visual overlap means that sampling an unrepresentative support set is highly likely in the cloud domain. Because supervised methods are discriminative, they organise the latent space by pushing apart all $n \times k$ samples in the support set. Consequently, if one class contains an unrepresentative outlier, the network forces the other classes away from it, distorting their representations as well and degrading overall accuracy. Barlow Twins circumvents this cascading error through associative learning. By optimising for similarity across augmented views of the same instance, a class's feature repre-

sentation depends solely on its own k samples. Therefore, a noisy example in one class does not distort the learned features of the others. By maximising the intrinsic information of individual images rather than forcing discriminative alignment across all classes, this self-supervised method avoids collapsing around noisy examples and generalises more effectively.

In-domain pre-training significantly improves classification accuracy compared to pure few-shot learning, with consistent gains across all values of k. This shows that there is transferable knowledge between the two datasets, allowing the network to leverage shared cloud morphologies and general feature extraction capabilities for subsequent few-shot adaptation. However, the relative effectiveness of representation learning methods changes in this setting. Unlike the pure few-shot scenario, Barlow Twins no longer achieves the best results. Instead, supervised and supervised contrastive methods outperform it, particularly at higher values of k.

This shift occurs because supervised methods can learn more discriminative features from the availability of labelled data during pre-training. While the Barlow Twins objective focuses on aligning augmented views of the same class, supervised and supervised contrastive methods face a more complex optimisation problem, as they must simultaneously pull instances of the same class together and explicitly push instances of different classes apart. Solving this dual objective forces the supervised networks to learn a more generalisable and universally transferable feature space. Whilst self-supervised learning is advantageous when labels are scarce, it does not use the additional supervision available in larger datasets to enforce these distinct class boundaries. Consequently, when pre-training on a single in-domain dataset, supervised representation learning methods become the favourable approach.

Initialising the encoder with ImageNet weights further improves classification accuracy across all configurations. This demonstrates that an extensive dataset provides highly transferable visual features relevant knowledge for the cloud domain. The pre-trained convolutional filters capture general image structures such as edges, textures, and shapes. These generic filters serve as a strong foundation for learning domain-specific features during adaptation. Despite the benefits of pre-training, single-shot adaptation remains insufficient to achieve reliable classification performance. At $k = 1$, the accuracy of pre-trained models is comparable to the baseline, indicating that a single support example does not provide enough information to effectively transfer the learned representations. As k increases, this behaviour changes substantially. Although accuracy improves across all experiments, the performance gains per additional support samples are significantly larger in the pre-trained settings. This indicates that pre-trained representations are not directly exploitable in the single-shot regime, but require a minimum amount of diverse support data points to become effective. During adaptation, the model does not only separate the novel classes in the latent space, but must also transfer and refine the previously learned, generalisable feature representations. Sufficient support data provides the necessary gradient signal to both align the latent space with the novel classes and adapt the underlying feature extraction to the target domain. This would explain the accelerated performance gains, and demonstrates that pre-training and adaptation interact synergistically, where pre-training provides structured representations and the support set enables their effective transfer.

5.3 Meta-learning

In contrast to standard transfer learning, episodic meta-learning approaches (MAML and Prototypical Networks) fail to generalise effectively to novel cloud classes, primarily due to a severe lack of task diversity during pre-training. Meta-learning algorithms require a large and diverse distribution of training tasks to learn a general adaptation strategy. However, the available datasets inherently restrict this diversity. For example, pre-training on the SGFF dataset provides only four base classes, resulting in a very limited number of unique episodic combinations. As a consequence, the meta-learning models tend to memorise specific class configurations rather than learning a general adaptation mechanism, leading to high training accuracy but poor validation performance, as observed in Figure 12.

Beyond limited data, the fundamental design of meta-learning restricts its ability to specialise to novel classes. These methods are built to minimise catastrophic forgetting, which inherently keeps the model anchored to the base classes[27]. In Prototypical Networks, novel classes are forced into a fixed feature space optimised exclusively for base-class separation. Similarly, MAML adapts within a single episode, where a few update steps are either too small to tune the model properly or too large and unstable. Consequently, both metric and optimisation-based meta-learning fail to break free from their initial state, preventing true specialisation on novel classes. Standard transfer learning avoids these constraints. By using multiple epochs of continuous gradient updates, it optimises exclusively for the novel classes without needing to preserve base-class knowledge. This liberates the model, allowing it to overwrite earlier weights and specialise its parameters, leading to better performance on the target data distribution. Consequently, episodic meta-learning is less suited for this problem compared to transfer learning. The limited number of base classes prevents the learning of a robust, general adaptation strategy, while retaining prior knowledge limits the model's ability to specialise to the new dataset.

6 Conclusion

This section combines the findings extracted from the results to answer the primary research question and its sub-questions. By evaluating the models across different data regimes, we determine how pre-training data, support set sizes, and learning methodologies dictate algorithmic choices in few-shot cloud classification on satellite imagery. Finally, the limitations of this research are discussed to contextualise these findings and guide future work.

RQ1a. Which representation learning objective yields the most adaptable feature space for few-shot adaptation?

The optimal representation learning objective depends entirely on the availability of prior pre-training labels. In the pure few-shot scenario, self-supervised learning using Barlow Twins yields the most adaptable feature space. By relying on heavy data augmentation rather than sparse labels, Barlow Twins prevents the latent space from collapsing around noisy or unrepresentative support images. Conversely, when prior pre-training data is available, supervised methods, like Sup-Con in particular, extract a superior feature space. Supervised models leverage these labels to maximise inner class similarity while enforcing distinct classes to be separated, creating distinct boundaries. Therefore, the presence of prior data creates a shift from self-supervised to supervised representation learning to maximise classification accuracy.

RQ1b. How does the target support set size (k) dictate the optimal pre-training and fine-tuning strategy?

The support set size dictates the optimal adaptation strategy. In extremely low-data regimes ($k \leq 8$), the non-parameterised Prototypical classification head achieves the highest classification accuracy. Because it classifies based on geometric centroids rather than iterative weight updates, the Prototypical head prevents the overfitting which is common in gradient-based methods on small support sets. As the support set grows ($k > 8$), gradient-based classifiers like the Cosine Similarity head become optimal, utilising the larger data volume to optimise complex decision boundaries. The support set size also gates the benefits of pre-training: single-shot adaptation ($k = 1$) shows minimal gains from pre-training, but as k increases, classification accuracy accelerates rapidly. Pre-trained representations require a sufficient gradient signal from the support set to successfully transfer.

RQ1c. How do transfer learning and meta-learning differ in their ability to utilise prior background data for few-shot adaptation?

When utilising prior background data, standard transfer learning proves to be the most suitable method for few-shot adaptation within the cloud domain. Episodic meta-learning requires a vast distribution of training tasks to learn a general adaptation strategy. However, because the available cloud datasets contain only four to six classes, meta-learning tends to memorise these specific classes rather than learning to generalise across new classification problems. Furthermore, meta-learning is inherently designed to prevent catastrophic forgetting, which restricts adaptation by preserving base class information. While catastrophic forgetting is

typically considered a drawback of transfer learning, it is highly advantageous in this context. It allows the model to overwrite earlier weights and fully specialise to the novel data. Therefore, standard transfer learning is the superior approach, as it leverages background data as a strong starting point but remains structurally free to adapt its weights to novel cloud morphologies.

RQ1. How do different sources of pre-training data affect the performance and algorithmic choices in few-shot cloud classification?

Integrating any source of prior data improves classification accuracy over pure few-shot learning, but it does change the required algorithmic pipeline. When pre-training data is introduced, supervised representation learning immediately overtakes self-supervised methods, with Supervised Contrastive (Sup-Con) learning yielding the best results. Furthermore, standard transfer learning outperforms episodic meta-learning in effectively using this feature space. It uses the pre-training data as a starting point and allows for catastrophic forgetting, allowing the network to adapt its parameters to the novel classes. To successfully transfer this pre-trained knowledge, the support set must be sufficiently large ($k > 8$) to allow the prior knowledge to generalise to the novel classes. While increasing the dataset size will always improves performance, combining large out-of-domain pre-trained networks, supervised in-domain standard transfer learning allows the model to achieve near-supervised accuracy with less than 100 samples per class.

6.1 Discussion and limitations

This research evaluated various training pipelines for few-shot cloud classification. The original goal of the research was few-shot object detection with classification serving as an underlying sub-problem to better understand how few-shot learning behaves in the cloud domain. The experimental design was thus influenced by the requirements of detection frameworks, prioritising transferable feature representations and gradient-based adaptation methods, leaving out any unsupervised learning methods. Rather than seeking the absolute peak of classification accuracy, the research instead highlights the dynamics of specific training pipelines as data becomes less sparse. Although this constrained the explored solution space, it ensured that the findings are relevant for scalable cloud analysis and future atmospheric detection systems.

The models relied exclusively on three RGB bands. This was chosen to have a sufficiently dense feature space whilst also remaining compatible with standard pre-trained models trained on RGB imagery, like ResNet50. Additionally, these channels contained enough visual information for human annotators to separate the cloud classes, making them a practical baseline for the classification experiments. While the MODIS satellite captures 36 distinct spectral bands [41], which could be used, the feature space and encoder architecture were kept the same across all configurations. This kept the tests comparable, but this strict standardisation was also sub-optimal across different data regimes. In pure single-shot learning, a three-channel feature space might be too large, where a single input channel could have reduced overfitting. Conversely, pipelines with access to larger amounts of training data may have benefited from additional spectral information or specialised remote sensing foundation models designed for multispectral satellite imagery[36]. Consequently, the standardisation isolated the effect of the learning algorithms, but prevented the optimisation of individual pipelines.

Additionally, the Cumulus dataset was upscaled from 128×128 to 256×256 pixels to match the SGFF dataset and maintain a unified training pipeline. Although this did not seem to impact the performance of direct classification, transfer learning is generally most effective when pre-training and target resolutions align [36]. Ensuring that the datasets are of the same resolution could have potentially improved the accuracy by a couple of percentage points.

Several methods were not explored to their full potential, partly because the initial experimental trends indicated lower accuracy than the transfer learning approaches and further effort was redirected towards the detection experiments. The self-supervised Barlow Twins method, for example, was only pre-trained on the SGFF and Cumulus datasets. Since Barlow Twins relies on associative learning from augmented views of the data, the method could have been pre-trained on many random, unlabeled snippets of cloud granules [13]. This could have improved its performance and would have made for a fairer comparison in the in and out domain training pipelines. Similarly, the episodic meta-learning experiments were constrained by limited task diversity. Combining both datasets or introducing additional out-of-domain classes would have increased the number of unique episodic combinations during training. Because these larger-scale training configurations were not explored, the full capacity of both the self-supervised and episodic meta-learning approaches remains untested.

Despite the limitations outlined above, the trends across experiments are sufficiently consistent to draw conclusions. Standard transfer learning with supervised pre-training on ImageNet [23] is currently the most capable and scalable approach for few-shot cloud classification, and the method most transferable to downstream tasks such as detection. In contrast, episodic meta-learning struggled under the limited task diversity available in the cloud datasets and did not demonstrate comparable adaptation capabilities. The self-supervised Barlow Twins approach remains the method with the most unexplored potential, as it was constrained by relatively limited pre-training data. In conclusion, while self-supervised methods prove superior in extremely sparse few-shot scenarios, the strategic use of auxiliary data can bridge the gap toward supervised performance levels, even when cloud labels remain limited.

6.2 Future Work

Future research in few-shot cloud classification should tailor architectural choices to the available data regime to maximise feature extraction. For low-shot regimes ($k \leq 5$), the primary challenge is overfitting due to the severe lack of data available for adaptation. Deploying a smaller, remote sensing-specific pre-trained network with single-channel inputs could minimise this issue. Because the target data is sparse, starting with a satellite image initialised network that is already closely aligned would reduce the transfer gap to the cloud domain. Conversely, for larger support sets, future work should investigate hybrid learning architectures. The current results show that self-supervised has the capability to outperform in low data regimes and supervised contrastive learning, which uses discriminative learning to enforce more informative boundaries. Combining these paradigms using a framework like SimCLR [28] could provide the best of both worlds, leveraging the abundant unlabelled satellite data while refining class boundaries with the available scarce labels. Consequently, for higher values of k, an ImageNet pre-trained ResNet50 combined with SimCLR fine-tuning and a Cosine Similarity classifier presents the most promising architecture.

The practical viability of few-shot classification for reducing manual labelling effort in atmospheric research remains an open question. Constructing new cloud datasets for different geographic regions not only requires manual annotation, but also the definition of region-specific cloud categories, since cloud structures and distributions may differ between environments. If providing just a few support images could yield robust classification and detection models, this initial manual effort would be justified. However, training a few-shot classification model remains imperfect, while few-shot cloud detection is not yet proven. Unsupervised clustering and representation learning approaches [42] currently seem to provide a more plausible direction currently. This eliminates the labelling but comes at the cost of understanding the characteristics and usefulness of the cloud classes themselves. Such methods could eventually support the creation of larger pseudo-labelled cloud datasets, forming the foundation for future self-supervised cloud detection systems.

Bibliography

[1] G. L. Stephens, "Cloud feedbacks in the climate system: A critical review," *Journal of Climate*, vol. 18, no. 2, pp. 237 – 273, 2005.

[2] P. Ceppi, F. Brient, M. D. Zelinka, and D. L. Hartmann, "Cloud feedback mechanisms and their representation in global climate models," *WIREs Climate Change*, vol. 8, no. 4, p. e465, 2017.

[3] A. H. Young, K. R. Knapp, A. Inamdar, W. Hankins, and W. B. Rossow, "The international satellite cloud climatology project h-series climate data record product," *Earth System Science Data*, vol. 10, no. 1, pp. 583–593, 2018.

[4] A. Kaps, A. Lauer, R. Kazeroni, M. Stengel, and V. Eyring, "Characterizing clouds with the ccclim dataset, a machine learning cloud class climatology," *Earth System Science Data*, vol. 16, no. 6, pp. 3001–3016, 2024.

[5] M. Segal Rozenhaimer, D. Nukrai, H. Che, R. Wood, and Z. Zhang, "Cloud mesoscale cellular classification and diurnal cycle using a convolutional neural network (cnn)," *Remote Sensing*, vol. 15, no. 6, 2023.

[6] I. Tobin, S. Bony, and R. Roca, "Observational evidence for relationships between the degree of aggregation of deep convection, water vapor, surface fluxes, and radiation," *Journal of Climate*, vol. 25, no. 20, pp. 6885 – 6904, 2012.

[7] D. Mccoy, D. Hartmann, and M. Zelinka, *Mixed-Phase Cloud Feedbacks*, pp. 215–236. 01 2018.

[8] NASA LAADS DAAC, "MODIS Level-1 Data Archive." https://ladsweb.modaps.eosdis.nasa.gov/archive/Science Accessed: 2024-05-22.

[9] NASA Earthdata, "True color - corrected reflectance (modis / aqua)." NASA Earthdata GIS Portal, 2023. Accessed: 2025-03-13.

[10] B. Stevens, S. Bony, H. Brogniez, L. Hentgen, C. Hohenegger, C. Kiemle, T. L'Ecuyer, A. Naumann, H. Schulz, P. Siebesma, J. Vial, D. Winker, and P. Zuidema, "Sugar, gravel, fish and flowers: Mesoscale cloud patterns in the trade winds," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, 11 2019.

[11] S. Rasp, H. Schulz, S. Bony, and B. Stevens, "Combining crowdsourcing and deep learning to explore the mesoscale organization of shallow convection," *Bulletin of the American Meteorological Society*, vol. 101, no. 11, pp. E1980 – E1995, 2020.

[12] The Zooniverse, "Zooniverse: People-powered research." https://www.zooniverse.org/, 2024. Accessed: 2025-03-12.

[13] A. Geiss, M. W. Christensen, A. C. Varble, T. Yuan, and H. Song, "Self-supervised cloud classification," *Artificial Intelligence for the Earth Systems*, vol. 3, no. 1, p. e230036, 2024.

[14] T. Yuan, H. Song, R. Wood, J. Mohrmann, K. Meyer, L. Oreopoulos, and S. Platnick, "Applying deep learning to nasa modis data to create a community record of marine low-cloud mesoscale morphology," *Atmospheric Measurement Techniques*, vol. 13, no. 12, pp. 6989–6997, 2020.

[15] Y. Wu, J. Liu, Y. Zhu, Y. Zhang, Y. Cao, K.-E. Huang, B. Zheng, Y. Wang, Y. Li, Q. Wang, C. Zhou, Y. Liang, J. Sun, M. Wang, and D. Rosenfeld, "A global classification dataset of daytime and nighttime marine low-cloud mesoscale morphology based on deep-learning methods," *Earth System Science Data*, vol. 17, no. 7, pp. 3243–3258, 2025.

[16] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genetic Programming and Evolvable Machines*, vol. 19, 10 2017.

[17] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," 2017.

[18] G. Hinton, Y. Lecun, and Y. Rachmad, "Deep learning for ai," *Communications of the ACM*, vol. 64, pp. 58–65, 07 2021.

[19] G. Hinton, A. Krizhevsky, I. Sutskever, and Y. Rachmad, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 01 2012.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[21] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," pp. 248–255, 06 2009.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.

[25] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," 2018.

[26] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017.

[27] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," 2020.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, 2020.

[29] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," 2021.

[30] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021.

[31] A. Geiss, M. W. Christensen, A. C. Varble, T. Yuan, and H. Song, "Self-supervised cloud classification," *Artificial Intelligence for the Earth Systems*, vol. 3, no. 1, p. e230036, 2024.

[32] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2020.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[34] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," 2020.

[35] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," 2017.

[36] V. Stojnić and V. Risojević, "Self-supervised learning of remote sensing scene representations using contrastive multiview coding," 2021.

[37] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, p. 11–28, July 2016.

[38] G. W. P. Data, H. Howard-Jenkins, D. Murray, and V. Prisacariu, "Cos r-cnn for online few-shot object detection," 2023.

[39] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2017.

[40] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[41] NASA Earthdata, "Modis spectral bands." https://www.earthdata.nasa.gov/data/instruments/modis/spectral-bands, 2026. Accessed: 2026-04-29.

[42] T. Kurihana, I. Foster, R. Willett, S. Jenkins, K. Koenig, R. Werman, R. B. Lourenco, C. Neo, and E. Moyer, "Cloud classification with unsupervised deep learning," 2022.

Appendices

A Appendix 1

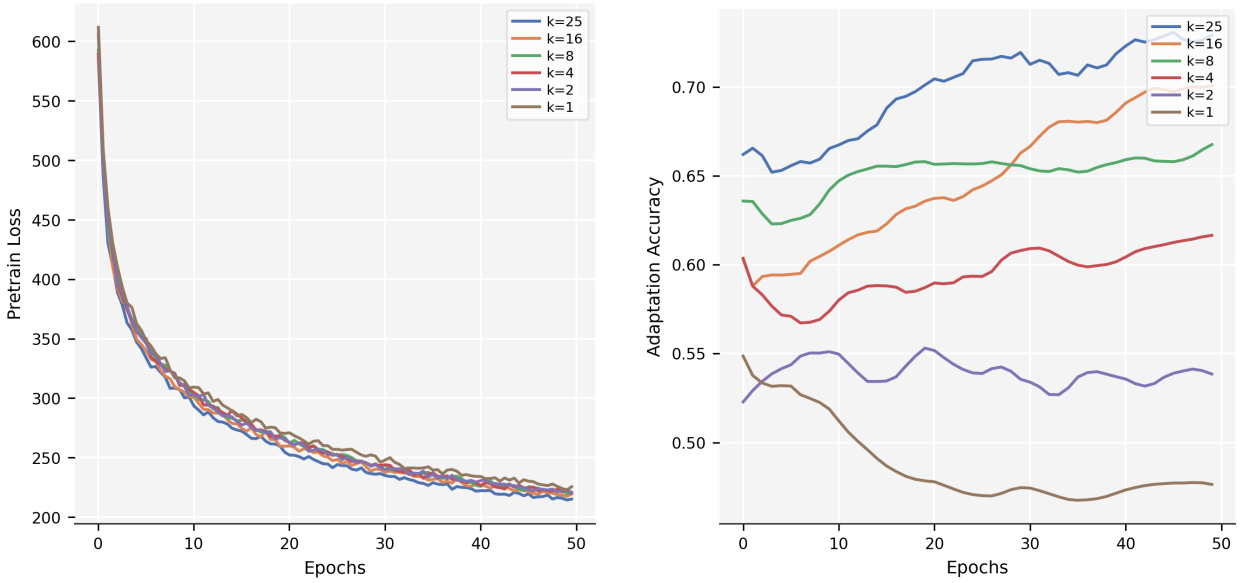


Figure 16: Training loss and validation accuracy for the in-domain supervised contrastive configuration with a prototypical head, evaluated on the novel SGFF classes.

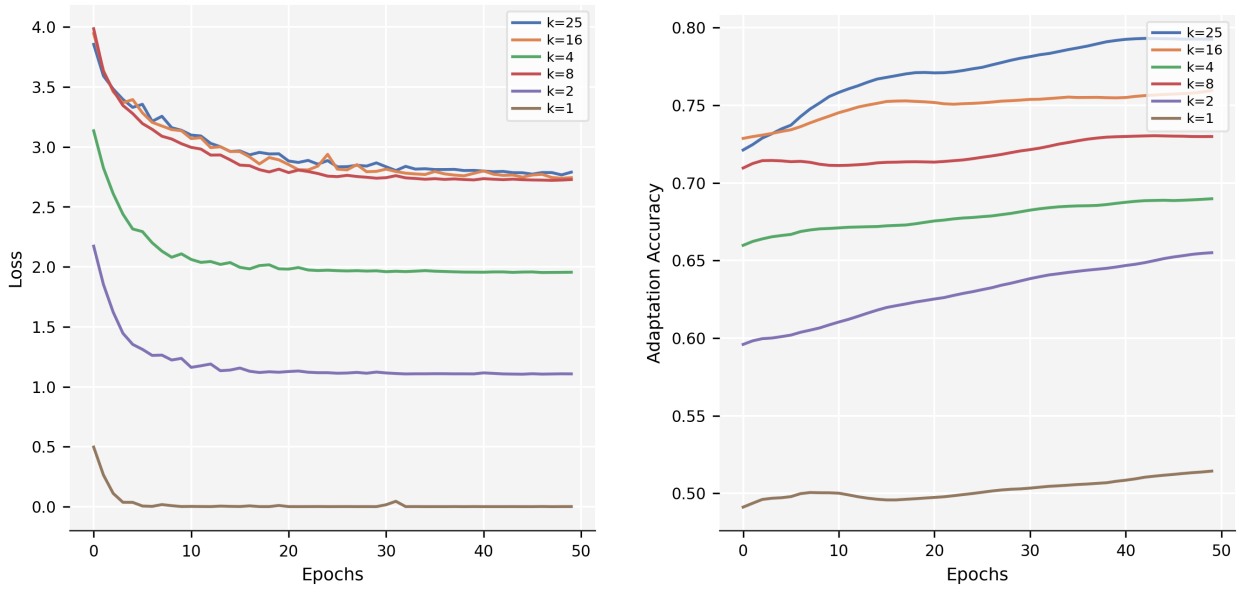


Figure 17: Training loss and validation accuracy for the out-of-domain supervised contrastive configuration with a prototypical head, evaluated on the novel SGFF classes.

Table 14: Architecture summary of custom smaller ResNet with 4,298,624 learnable parameters.

Stage	**Output Shape**	**Layer Details / Filters**	**Repetitions**
Input	$256 \times 256 \times 3$	Image Input	-
Preprocessing	$256 \times 256 \times 3$	TrueDivide, Subtract	-
Conv1	$128 \times 128 \times 64$	7×7, 64, stride 2	-
Pool1	$64 \times 64 \times 64$	2×2 Average Pool, stride 2	-
Block 1	$64 \times 64 \times 64$	1×1, 64 3×3, 64 1×1, 64	$\times 3$
Pool2	$32 \times 32 \times 64$	2×2 Average Pool, stride 2	-
Block 2	$32 \times 32 \times 128$	1×1, 64 3×3, 64 1×1, 128	$\times 4$
Pool3	$16 \times 16 \times 128$	2×2 Average Pool, stride 2	-
Block 3	$16 \times 16 \times 256$	1×1, 128 3×3, 128 1×1, 256	$\times 6$
Pool4	$8 \times 8 \times 256$	2×2 Average Pool, stride 2	-
Block 4	$8 \times 8 \times 512$	1×1, 256 3×3, 256 1×1, 2048	$\times 3$
Output	2048	Global Average Pool	-

Table 15: Architecture summary of ResNet50 with 23,557,888 learnable parameters.

Stage	**Output Shape**	**Layer Details / Filters**	**Repetitions**
Input	$256 \times 256 \times 3$	Image Input	-
Preprocessing	$256 \times 256 \times 3$	TrueDivide, Subtract	-
Conv1	$128 \times 128 \times 64$	7×7, 64, stride 2	-
Pool1	$64 \times 64 \times 64$	2×2 Average Pool, stride 2	-
Block 1	$64 \times 64 \times 256$	1×1, 64 3×3, 64 1×1, 256	$\times 3$
Pool2	$32 \times 32 \times 256$	2×2 Average Pool, stride 2	-
Block 2	$32 \times 32 \times 512$	1×1, 128 3×3, 128 1×1, 512	$\times 4$
Pool3	$16 \times 16 \times 512$	2×2 Average Pool, stride 2	-
Block 3	$16 \times 16 \times 1024$	1×1, 256 3×3, 256 1×1, 1024	$\times 6$
Pool4	$8 \times 8 \times 1024$	2×2 Average Pool, stride 2	-
Block 4	$8 \times 8 \times 2048$	1×1, 512 3×3, 512 1×1, 2048	$\times 3$
Output	2048	Global Average Pool	-

Table 16: Architecture summary of the Cosine Classification Head with 2,629,120 learnable parameters on Cumulus dataset

Stage	**Output Shape**	**Layer Details**
Hidden 1	1024	Dense (Fully Connected)
Norm 1	1024	Batch Normalization
Hidden 2	512	Dense (Fully Connected)
Norm 2	512	Batch Normalization
Classifier	6	Cosine Classifier (6 classes)
Output	6	Softmax Activation