Deep Learning Approaches for Cloud Property Retrieval: Leveraging Geospatial Foundation Models and Multitask Frameworks

REU Site: Online Interdisciplinary Big Data Analytics in Science and Engineering

Danielle Murphy[1], Kevin Zhang[2], Caleb E. Parten[3], Autumn Sterling[4], Haoxiang Zhang[5], Xingyan Li[6], Jordan Caraballo-Vega[7], Jie Gong[7], Mark Carroll[7], Jianwu Wang[6]

[1]Department of Mathematics, University of California, Berkeley
[2]Department of Computer Science, University of Maryland, College Park
[3]Department of Mathematical Sciences, Eastern New Mexico University
[4]Department of Computer Science, George Mason University
[5]Fairfax Christian School, Herndon, VA
[6]Department of Information Systems, UMBC
[7]NASA Goddard Spaceflight Center

Technical Report HPCF–2025–4, hpcf.umbc.edu > Publications

Abstract

With the rapid growth of Earth-observation datasets, geospatial foundation models (FMs) provide a scalable approach to learn transferable features across diverse satellite sensor data. However, their cross-sensor adaptation ability needs more exploration. To study this issue, we present a benchmarking study of **SatVision-TOA**, an FM pre-trained on over 20 years of MODIS data, when adapted to the GOES NOAA ABI sensor for four downstream cloud properties: **cloud mask, cloud phase** (segmentation), and cloud optical depth (**COD**) and cloud particle size (**CPS**) (regression). We propose a multi-task learning fine-tuning pipeline with a U-Net-based decoder and a lightweight preprocessor to address band-mismatch handling (14 MODIS bands for pre-training vs. 16 ABI bands for fine-tuning). To evaluate our pipeline, we benchmark **fine-tuned models** against **from-scratch baselines,** evaluate **full fine-tuning (FFT)** versus **parameter-efficient fine-tuning (PEFT)** methods (LoRA, VPT), and compare **14-band** versus **16-band** inputs. Our experiments show that multi-task learning improves efficiency and predictive quality in both fine-tuned and from-scratch settings. For the other four comparisons (FT vs. from-scratch, FFT vs. PEFT, 14-bands vs. 16-bands and loss functions), the results are mixed and there is no setup that always performs the best for all segmentation and regression tasks.

Key words: Deep learning, Fine-tuning, Cloud property retrieval, Multi-task learning

1 Introduction

Cloud property retrieval is essential for understanding Earth's climate, energy balance, and hydrological cycle [1]. Satellites provide the primary source of global cloud observations, and retrieval algorithms convert remote sensing measurements into key cloud properties such as cloud mask (cloudy or not cloudy), cloud phase, top height, and optical thickness.

In recent years, the rapid growth of Earth observation data has motivated the exploration of Foundation Models (FMs) as a new technique to effectively leverage these large-scale datasets. Often trained with self-supervised learning, FMs are powerful tools for Earth science remote sensing for their ability to learn generalizable representations from large-scale satellite imagery [2]. During pre-training on vast datasets, vision foundation models detect spatial patterns and learn to encode meaningful feature representations. Vision transformers use attention mechanisms to capture long-range dependencies and global context.

The development of FMs involves two key stages: pre-training and fine-tuning. In the pre-training stage, models are trained on large volumes of satellite data to learn spatial and spectral features; existing studies on geospatial FMs have primarily focused on this stage [3]. During fine-tuning, the knowledge acquired during pre-training is transferred to downstream tasks such as object detection or semantic segmentation. The typical fine-tuning paradigm places the pre-trained FM as the encoder, followed by an optional decoder and a task-specific head. This approach has been shown to improve performance on downstream tasks compared to training from scratch [2]. While pre-training is computationally expensive, fine-tuning enables the reuse of learned representations across multiple tasks, thereby improving efficiency and generalization.

Despite recent advances, current geospatial FMs lack systematic development toward generalization across diverse downstream tasks. Unlike regular computer-vision problems with natural images, which typically have consistent RGB channels, Earth observation data vary substantially across sensors in terms of spectral bands, modalities, and resolution. Such diversity leads to discrepancies between the data used for FM pre-training and the datasets required for fine-tuning specific applications. Adapting a geospatial FM to a dataset from a different satellite sensor thus needs more exploration.

In this work, we conduct a benchmarking study to evaluate how a geospatial FM performs in cloud property retrieval when fine-tuned on data from an unseen satellite sensor. Specifically, this work focuses on fine-tuning strategies for SatVision-TOA [4], a recently developed geospatial FM pre-trained on 20 years of MODIS observations, and adapts it for cloud property retrieval from a different satellite sensor ABI. A particular challenge arises from the mismatch in spectral bands when adapting SatVision-TOA to data from a different satellite sensor than its pre-training source: SatVision-TOA was pre-trained on 14 selected bands of the Moderate-resolution Imaging Spectroradiometer (MODIS), while Advanced Baseline Imager (ABI) provides 16 spectral bands. Previous work [4] addressed this mismatch by selecting the 14 ABI bands most similar in wavelength to the MODIS bands, matching the model's pre-trained input size, which yielded promising results in image reconstruction and 3D cloud retrieval. In contrast, we explore whether incorporating all 16 ABI bands through a lightweight preprocessing module can better exploit the additional spectral information, despite the FM's original 14-band input.

We assess fine-tuning approaches across four downstream tasks derived from ABI satellite data: cloud mask and cloud phase segmentation, and cloud optical depth (COD) and cloud particle size (CPS) regression. Our study examines parameter-efficient techniques to address the size of the FM, compares multi-task and single-task setups, and benchmarks fine-tuned models against models trained from scratch to evaluate performance and efficiency. This work makes the following key contributions:

- **Benchmarking Fine-tuning Strategies:** We conduct a comprehensive benchmarking study of fine-tuning strategies for SatVision-TOA, using full fine-tuning, alongside parameter-efficient fine-tuning methods including Low-Rank Adaptation [5], and Visual Prompt Tuning [6], identifying the most memory-efficient and performance-effective strategies across two classification and two regression tasks. In addition, we compare performance when using 14 bands and 16 spectral bands, highlighting the trade-offs in leveraging additional spectral bands.

- **Developing a Multi-task Learning Fine-Tuning Pipeline:** We design and implement a multi-task fine-tuning pipeline for cloud property retrieval for optimization across multiple related tasks. Our pipeline leverages hierarchical classification to exploit interdependencies among cloud properties, improving predictive accuracy compared to independent single-task training. This framework demonstrates enhanced efficiency, scalability, and adaptability of geospatial foundation models to complex downstream retrieval problems.

2 Related Work

Prior work has benchmarked foundation models on downstream tasks and surveyed their design and applications, including over fifty remote sensing FMs discussed in [2].

Six models pre-trained on satellite imagery were benchmarked in [7], trained with full fine-tuning. They reported that FMs exceed or meet baseline performance for tasks like segmentation and classification, but struggle more with change detection and oriented object detection. This work is also relevant as they worked with heterogeneous datasets, where the data used for fine-tuning is different from the data used during pre-training. Finally, this work showed the potential for multi-task learning to be used simultaneously with fine-tuning – they found success with multi-task pre-training (MTP), where the encoder outputs different feature representations for each task. Our work performs similar benchmarking with SatVision-TOA. By evaluating performance on two classification and two regression tasks, it is valuable to know whether similar task-related discrepancies between performances can be found.

Rotich et al. studied multi-task pre-training where the encoder is designed specifically for a downstream multi-task model [7]. In comparison, SatVision-TOA outputs a single feature representation, which we use to achieve promising multi-task downstream performance. Li et al. found success using an end-to-end deep learning model with multi-task learning to predict cloud mask, cloud phase (classification tasks), and COT (a regression task) [8]. Compared with baseline methods, their model MT-HCCAR, performed optimally across a variety of datasets and metrics. We also predict cloud mask and phase with two regression tasks using a multi-task model, building off of the pre-trained knowledge of the SatVision-TOA encoder.

In contrast to full fine-tuning, parameter-efficient fine-tuning (PEFT) methods leave much of the FM frozen during training, focusing on a specific subset of parameters. These strategies are often competitive with full fine-tuning, while requiring much less computational resources. Numerous different strategies have been used for fine-tuning computer vision models, surveyed in [9], and LoRA proved to be particularly effective among strategies tested with geospatial foundation models [10]. We evaluate how PEFT can be leveraged for this particular encoder.

The Swin transformer [11], which is the architecture of the FM SatVision-TOA, has been used with the U-Net in previous work, such as for classification with medium-resolution satellite remote sensing images [12]. Shortly thereafter, much research discussed the use of the Swin transformer with U-Net for medical image segmentation [13–17]. This architecture has even been used for single-task pavement crack detection [18]. More recently, the study in [19] used Swin-U-Net for multi-task learning for segmentation, image reconstruction, and classification tasks for medical images. This approach used U-Net decoders connected to a Swin encoder for the segmentation and reconstruction tasks, while the classification task had a simpler classification head composed of a global average pooling layer and a linear layer.

Our work builds on the current research by leveraging the use of a pre-trained foundation model for the Swin transformer in the Swin-U-Net framework. By positioning SatVision-TOA as the encoder, we use a single decoder connected to lightweight task heads to predict four cloud variables at once. Where [19] achieved promising results for segmentation and image reconstruction tasks, we study the compatibility of this architecture with a more complex multi-task environment, extending the work to a setting with two segmentation and two regression tasks.

3 Dataset

The source of the dataset used for training and evaluation is GOES-18 Advanced Baseline Imager (ABI) observations, accessed via the National Oceanic and Atmospheric Administration and Amazon Web Services' Open Data Registry (NOAA's AWS) (noaa-goes18 bucket) [20]. The ABI sensor provides full-disk coverage every 15 minutes from a geostationary orbit, capturing 16 spectral bands at native resolutions of 0.5 km (bands 1–2), 1 km (bands 3–6), and 2 km (bands 7–16) [21]. Our model inputs are from Level-1b (L1b) top-of-atmosphere radiances for all 16 spectral bands, with corresponding Level-2 (L2) cloud properties from the NOAA ABI Cloud Algorithm as model ground truth. L1b and L2 files were temporally matched to ensure concurrent L1b and L2 observations.

For model training, the dataset consists of 15000 image chips (128 × 128 pixels) collected from the source data spanning March to June in 2023. Full-disk ABI data were divided into 128 × 128 pixel blocks at 2 km resolution (native for L2 products; L1b bands were resampled by cubic interpolation to 2 km). Our preprocessing pipeline follows established practices from SatVison-TOA [4], including: 1) geolocation-based solar zenith angle filtering excluded blocks with angles > 72 (limiting atmospheric path distortion). 2) converting visible/NIR bands to top-of-atmosphere (TOA) reflectance and thermal bands to brightness temperature (BT). 3) using a min-max scaler for data normalization. The generation process is illustrated as Figure 3.1.

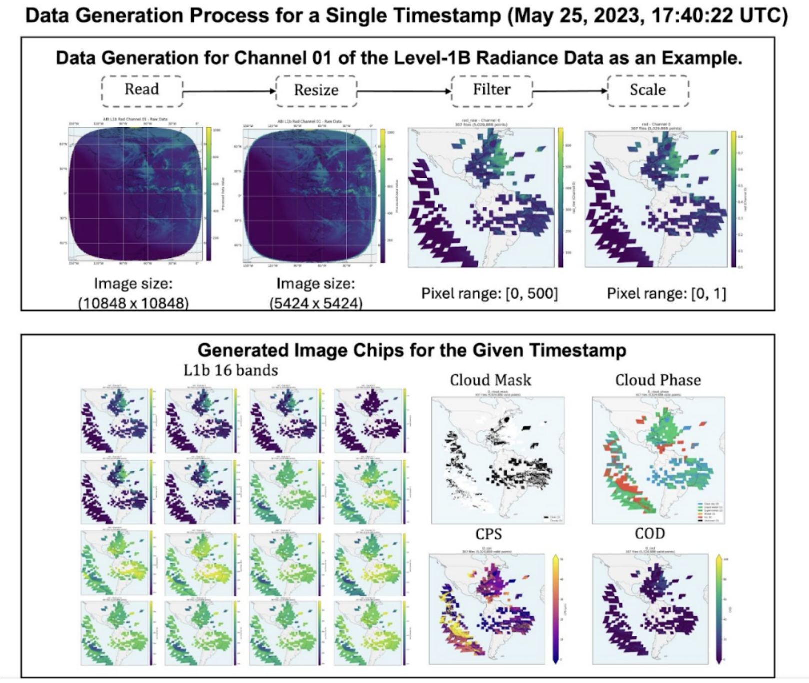
Data Generation Process for a Single Timestamp (May 25, 2023, 17:40:22 UTC)
Data Generation for Channel 01 of the Level-1B Radiance Data as an Example.
Read
Resize
Filter
Scale
Image size: (10848 x 10848)
Image size: (5424 x 5424)
Pixel range: [0, 500]
Pixel range: [0, 1]
Generated Image Chips for the Given Timestamp
L1b 16 bands
Cloud Mask
Cloud Phase
CPS
COD

Figure 3.1: Data generation process for a single timestamp (May 25, 2023, 17:40:22 UTC), illustrated using Channel 01 of the Level-1B radiance data. The workflow includes: (1) Read: reading the raw data and splitting it into 128 × 128 image chips; (2) Resize: resizing by interpolation if the spatial resolution is not 2 km; (3) Filter: filtering by discarding chips with NaN values or SZA > 72; and (4) Scale: scaling by converting to TOA reflectance or brightness temperature (BT), followed by normalization.

To adapt the pre-trained FM SatVision-TOA to our target task, the first step is to characterize the spectral differences between the pre-training sensor and the fine-tuning sensor. As noted in the introduction, SatVison-TOA FM was pre-trained on a subset of **14** spectral bands selected out of all **36** bands from MODIS [4], whereas our ABI dataset for fine-tuning contains full 16 bands. Table 3.1 details the alignment between these instruments. While 14 ABI bands have direct spectral counterparts in the MODIS configuration, the ABI sensor includes two distinct bands—Band 8 (6.19μm) and Band 13 (10.35μm)—that are absent in the MODIS subset for pre-training. These unmatched bands represent a shift in feature space that our fine-tuning strategy must accommodate.

Table 3.1: Spectral band comparison between the target sensor (ABI) and the source foundation model sensor (MODIS). The SatVision-TOA model was pre-trained on 14 MODIS bands. Two ABI bands (08 and 13) have no direct equivalent in the pre-training data. **WL (μm)** stands for **central wavelength** in micrometer.

	ABI (Fine-Tuning)			**MODIS (Pre-Training)**		
Index	**Band**	**WL (μm)**	**Description**	**Band**	**WL (μm)**	**Description**
1	2	0.64	Red	1	0.659	Land/Cloud/Aerosols Boundaries
2	3	0.864	Vegetation	2	0.865	Land/Cloud/Aerosols Boundaries
3	1	0.47	Blue	3	0.47	Land/Cloud/Aerosols Properties
4	5	1.61	Snow/Ice	6	1.64	Land/Cloud/Aerosols Properties
5	6	2.24	Cloud Particle Size	7	2.13	Land/Cloud/Aerosols Boundaries
6	7	3.90	Shortwave Window	21	3.96	Surface/Cloud Temperature
7	4	1.373	Cirrus	26	1.375	Cirrus Clouds, Water Vapor
8	**8**	**6.19**	**Upper-level Water Vapor**	– ***No Equivalent*** –		
9	9	6.93	Mid-level Water Vapor	27	6.72	Cirrus Clouds, Water Vapor
10	10	7.34	Lower/Mid-level Water Vapor	28	7.33	Cirrus Clouds, Water Vapor
11	11	8.44	Cloud-top Phase	29	8.55	Cloud Properties
12	12	9.61	Ozone	30	9.73	Ozone
13	**13**	**10.33**	**Clean IR Longwave**	– ***No Equivalent*** –		
14	14	11.21	IR Longwave	31	11.03	Surface/Cloud Temperature
15	15	12.29	Dirty IR Longwave	32	12.02	Surface/Cloud Temperature
16	16	13.28	CO_2	33	13.34	Cloud Top Altitude

4 Methodology

First, we will introduce some of the modules that are used in either the models from-scratch and/or the fine-tuning models. We will introduce our multi-task model architectures, strategies used for fine-tuning, and finally the loss functions and metrics used to evaluate the models.

4.1 Machine Learning Frameworks

Machine learning algorithms are able to capture nonlinear relationships in high-dimensional image data. Two approaches with image data may be taken, per-pixel analysis and whole image segmentation/regression. In pixel-by-pixel learning, the 16 spectral bands of a pixel are used as input to predict the four cloud properties. Several machine learning methods that take a one-dimensional vector as input are suited for this task, such as multi-layer perceptrons (MLPs), random forests, and histogram gradient boosting. These methods are lightweight and relatively simple, but may still provide sufficiently accurate results. However, this research primarily works with spatial models that consider neighboring pixels when determining a pixel's properties. These models use the entire image patches, represented as $(128, 128, 16)$ tensors, as input.

4.1.1 2D Convolution

Convolutional layers form the basis of spatially-aware models, extracting features by sliding small kernels (e.g., 3×3) across an image with N channels and computing weighted sums. This produces feature maps that capture rich spatial patterns. Applying multiple kernels yields stacked feature maps, which can be further processed with activations, normalization, pooling, and dropout to enhance nonlinearity and reduce overfitting. Compared to fully connected layers, convolutions handle arbitrary input sizes more efficiently, with faster inference and improved accuracy [22]. Networks built from stacked convolutions, Convolutional Neural Networks (CNNs), are widely used for image analysis across domains, including atmospheric science.

4.1.2 Encoder Architecture: SwinV2 Transformer

The foundation model encoder we use for fine-tuning, SatVision-TOA, utilizes the hierarchical SwinV2 Transformer architecture [11]. Swin stands for shifted windows – the input image is divided into non-overlapping windows, and self-attention is performed within each window. Between consecutive attention layers, these windows are shifted. The hierarchical aspect of this architecture refers to the stages of the transformer, which output multi-resolution feature maps that can be leveraged alongside the final output of the transformer. We leverage the different feature maps in conjunction with decoder structures like the U-Net.

4.1.3 U-Net

Consisting of an encoder, bottleneck, and decoder, the U-Net is a convolutional neural network architecture designed for pixel-wise prediction tasks. The encoder downsamples an input through repeated convolutions. Before the decoder, the bottleneck performs two final convolutions. During this process, the spatial information in the data decreases while feature information grows. At the bottleneck, the model should theoretically have detected many high-level features.

Finally, features are upsampled through transposed convolutions in the decoder. The key detail of the U-Net is its use of skip-connections. In between each upsampling step, the corresponding feature map from the encoder is stacked onto the current feature map. The motivation for this design choice is that the feature maps are at their smallest spatial size at the bottleneck due to the repeated convolutions of the encoder. Skip connections from earlier stages of the encoder provide the missing spatial information. With this architecture, the model learns from the features at every level. Fine-tuned models used the U-Net with SatVision-TOA as the encoder, whereas the models trained from scratch used ResNet for the encoder.

4.1.4 Fine-tuning Strategy

Full fine-tuning, where all of the FM's parameters are trainable, provides performance advantages but is computationally expensive. In PEFT, much of the foundation model encoder is left frozen and only a smaller subset of encoder parameters is trained. Often, a much smaller set of new parameters might be introduced to guide the model's learning.

VPT is a parameter-efficient fine-tuning method inspired by prompt tuning for LLMs, where the input to the pre-trained model is wrapped with learnable visual prompts. For vision transformer models, the image patches are wrapped with these prompts. In VPT-Shallow, prompts are added only to the initial patch representations, whereas VPT-Deep adds prompts at multiple layers throughout the encoder.

LoRA is an alternative method in which the pre-trained transformer weights are frozen, and trainable low-rank matrices are inserted to approximate the updates. Conceptually, LoRA relies upon the hypothesis that updates to weight matrices during training have low intrinsic dimension during fine-tuning adaptation. Instead of updating the full-rank weight matrices, the low-rank matrices A and B are trained:

$$W_{updated} = W_{frozen} + \left(\frac{\alpha}{r}\right) AB.$$

Weight matrices in the FM's layers typically have full rank. A lower rank means there are fewer trainable parameters, resulting in a more computationally efficient method compared to full fine-tuning. For instance, if the original weight matrix is $n \times n$, A is $n \times r$ and B is $r \times n$, where $r << n$. This reduces the trainable parameters from n^2 to $2nr$.

4.2 Model Descriptions

The from-scratch multi-task model uses four U-Net modules; one per each task. Each U-Net uses the Resnet34 encoder, a pre-trained model on general image classification with around 21m parameters. Compared to working with large foundation models, this lightweight encoder allows for greater flexibility when designing and training models.

On the other hand, the fine-tuned multi-task model uses a larger U-Net, connected to SatVision-TOA as the encoder. SatVision-TOA is a 3 billion parameter foundation model pre-trained on MODIS data with masked-image-modeling. This pre-trained knowledge offers a unique advantage compared to ResNet's general classification training.

Both multi-task models set the loss to be a weighted sum of the four task losses: $loss = \lambda_1 CE + \lambda_2$ phase loss $+ \lambda_3 MSE_{COD} + \lambda_4 MSE_{CPS}$.

4.2.1 Multi-task Fine-tuned Model

Figure 4.1 shows the architecture for the multi-task model built off of the SatVision-TOA encoder. The U-Net decoder reconstructs higher-resolution representations of the data while preserving feature information from the encoder for accurate cloud attribute prediction. The U-Net's complexity makes it a good choice for a shared decoder in a multi-task model. Using information from each stage of the FM encoder, the decoder learns robust representations of the feature data that can be used for all four lightweight task heads, which are convolutional layers.

The logits from the cloud mask prediction are appended to the decoder outputs used for the prediction of the other three cloud properties.

4.2.2 From Scratch Multi-task Model Architecture

Among the models trained from scratch, the Multi-task model as in Figure 4.2 demonstrated the best overall performance. It begins with a single convolutional layer that functions as an encoder, extracting shared spatial features while progressively increasing channel depth in subsequent convolutions. The initial encoder feature map is then passed through four separate U-Net branches, each dedicated to one of the target properties. To enhance predictive consistency, the cloud mask logit matrix is appended to the output feature maps of the other three properties, under the assumption that when the cloud mask equals zero (indicating no cloud), the remaining properties should also be zero. Finally, a single convolutional layer serves as the decoder, enabling the cloud mask to exert a more direct influence on the predictions of cloud phase, COD, and CPS.

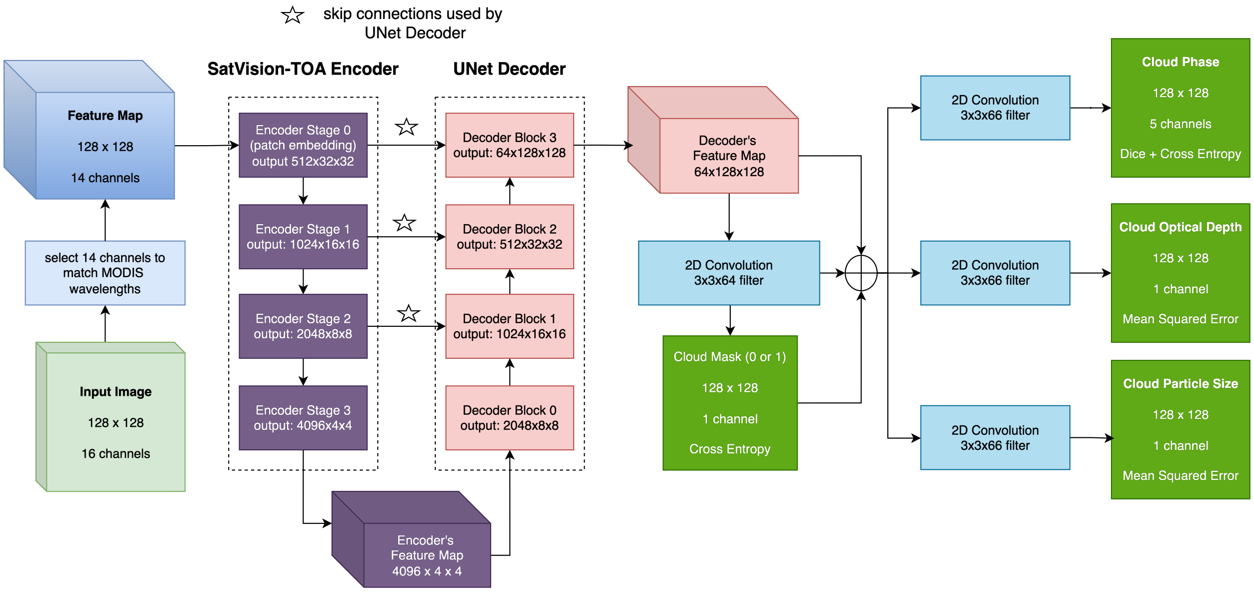
☆ skip connections used by UNet Decoder
SatVision-TOA Encoder
UNet Decoder
Feature Map
128 x 128
14 channels
select 14 channels to match MODIS wavelengths
Input Image
128 x 128
16 channels
Encoder Stage 0 (patch embedding) output 512x32x32
Encoder Stage 1 output: 1024x16x16
Encoder Stage 2 output: 2048x8x8
Encoder Stage 3 output: 4096x4x4
Decoder Block 3 output: 64x128x128
Decoder Block 2 output: 512x32x32
Decoder Block 1 output: 1024x16x16
Decoder Block 0 output: 2048x8x8
Encoder's Feature Map 4096 x 4 x 4
Decoder's Feature Map 64x128x128
2D Convolution 3x3x64 filter
Cloud Mask (0 or 1)
128 x 128
1 channel
Cross Entropy
2D Convolution 3x3x66 filter
2D Convolution 3x3x66 filter
2D Convolution 3x3x66 filter
Cloud Phase
128 x 128
5 channels
Dice + Cross Entropy
Cloud Optical Depth
128 x 128
1 channel
Mean Squared Error
Cloud Particle Size
128 x 128
1 channel
Mean Squared Error

Figure 4.1: Fine-tuned Multi-task Model.

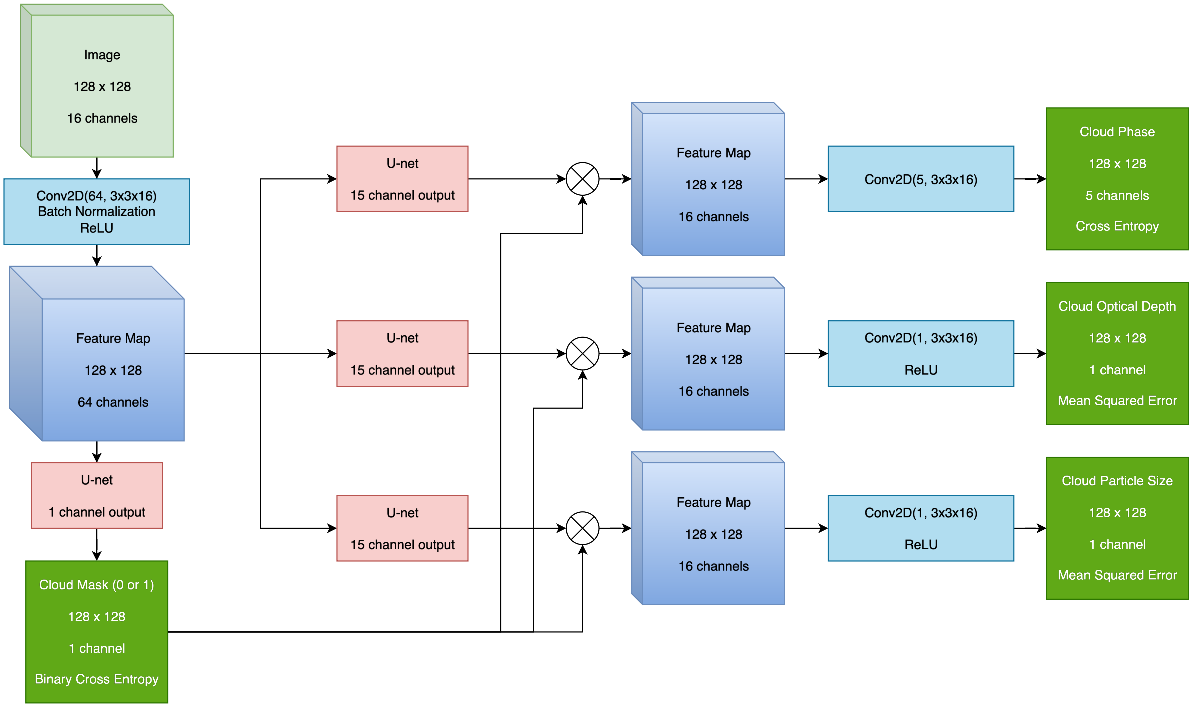

Figure 4.2: Multi-task Model from Scratch.

4.3 Loss Functions

4.3.1 Cross Entropy Loss and Focal Loss

Cross Entropy (CE) Loss, also known as Logarithmic Loss, is the standard loss function for classification tasks that output a discrete probability distribution for a given input [23].

The main drawback of CE Loss is its limited ability to handle imbalanced class data. Focal Cross

Entropy (FCE) loss is a variant of CE that is theoretically more suited for the imbalanced Cloud Mask and Cloud Phase data. Focal loss addresses class imbalance in the dataset by down-weighting easy predictions, CE loss is modified via multiplication by $(1-p)^{\gamma}$.

4.3.2 Dice Loss

The Dice-Sørensen coefficient (DSC) was introduced in two independent papers as a statistic used to gauge the similarity of two samples [24, 25].

The equation for dice loss is 1 - DSC. Considering P to be the model's segmentation prediction and G the ground truth,

$$DSC = 2\frac{|P \cap G|}{|P| + |G|}$$

Dice loss is commonly used for segmentation tasks as it directly considers the overlap between the model's prediction and the ground truth, rather than treating each pixel independently as in CE loss. This makes it a natural choice for segmentation: the dice coefficient is a standard method of evaluation for medical image segmentation [26].

Dice loss can be considered unstable – for instance, the loss gradient is only zero when there is zero overlap between the prediction and ground truth, a counterintuitive behavior [26]. Combining Dice loss with FCE or CE loss allows for a balanced alternative that still provides some of the advantages that dice loss provides [27, 28].

4.3.3 MSE Loss

Mean squared error is a common loss function for regression tasks. Compared to Mean Absolute Error, MSE trades bias for low variance by penalizing drastically wrong projections more heavily. Ultimately, MSE rewards predictions that are in the general vicinity of the target while being very stable.

4.4 Performance Evaluation Metrics

4.4.1 Mean Intersection Over Union (mIOU)

Since pixel accuracy can be misleading for tasks with imbalanced data, we evaluate our models' performance on classification tasks with (unweighted) mIOU. For a single class a, we consider the overlap between what is predicted to be in class a and what is not predicted to be class a.

mIOU evaluates the model's ability to segment the image into meaningful regions, which allows us to evaluate whether the model is capable of identifying different cloud structures.

4.4.2 R^2 Score

Also known as the coefficient of determination, the R^2 score is a simple way to evaluate a regression model. R^2 score lies within $(-\infty, 1]$. If a regression model fits perfectly, MSE is 0 and $R^2 = 1$. On the other hand if $R^2 = 0$, the model is as effective as predicting the mean each time.

5 Experiments

In this section, we first compare the best overall multi-task and individual models across the fine-tuned models and models designed from scratch, displaying the metrics in Section 4.4. Following this, we display results to discuss the following topics: 1) Fine-tuned vs. from-scratch for cloud property retrieval. 2) Comparison of full-finetuning and parameter-efficient fine-tuning strategies. 3) Using full 14-bands vs. selected 16-bands for fine-tuning. 4) Hyperparameter tuning of from-scratch model. 5) Loss weights tuning for both fine-tuned and from-scratch models.

As discussed in Section 4.2, both multi-task models leverage the U-Net decoder. The fine-tuned model uses one large U-Net that positions SatVision-TOA (3B parameters) as the encoder, whereas the model trained from scratch uses four smaller U-Nets, one per task. The hyperparameters of fine-tuning and training from scratch models are in Table 5.1. All fine-tuned results besides the single-task cloud mask model, which was trained with LoRA, are trained with full fine-tuning. Single-task fine-tuned models used the following architecture, with only the regression tasks using the preprocessor: preprocessor → encoder → fully convolutional decoder → single convolutional layer → prediction.

Table 5.1: Multi-task common hyper-parameters.

Images	14973
Train/Validation/Test Split	80/10/10
Optimizer	Adam
Batch size	128
Learning rate	.00002
Learning rate scheduler	Patience=3, Factor=.5
Epochs	100
Loss	Weighted sum of individual losses

5.1 Overall Comparison between Fine-Tuning and From-Scratch Models

Table 5.2 compares fine-tuned models with models trained from scratch. First, considering the single-task models, the fine-tuning models and the models from scratch are very competitive — fine-tuned single-task models obtain the best results for cloud phase and cloud optical depth, while the baseline models from scratch excel with cloud mask and cloud particle size. The models trained from scratch take significantly less time to train, however, which highlights the drawbacks of fine-tuning with a particularly large foundation model.

The multi-task model from scratch outperformed its fine-tuned counterpart, and multi-task learning improves the predictions of from-scratch models across all tasks. The fine-tuned multi-task model also sees significant gains, especially for cloud mask and CPS, where the single-task fine-tuned model underperformed relative to the from-scratch baseline. Given the high computational cost of training large foundation model encoders, multi-task learning is especially advantageous. Instead of training single-task models for each task, some of which take nearly two hours to train, a single multi-task model is trained for all four tasks in under 2.5 hours. This approach offers a much more efficient use of time and computation. Moreover, multi-task learning capitalizes on the expressive capacity of the large encoder, effectively leveraging shared representations across diverse cloud prediction tasks.

While fine-tuned models are still generally competitive, the models from scratch perform better in many cases, and are much more pragmatic to train. Future work could study how the different aspects of the encoder affect performance results. For instance, it is possible that the sheer size of the FM used (3B parameters) may have affected the ability of the model to make generalizations. Fine-tuning smaller variants of SatVision, such as SatVision-base (84.5M parameters) and SatVision-huge (695.3M parameters), could highlight the trade-offs between model size, performance, and training efficiency.

Table 5.2: Performance of multi-task and individual models on cloud attribute prediction.

Model	Task	mIOU	Task	R^2	Train Time
Multi-task Models					
Fine Tuned MT	Mask	**0.895**	COD	**0.762**	2:30:07
	Phase	0.701	CPS	**0.793**	
From Scratch MT	Mask	**0.909**	COD	**0.775**	45:59
	Phase	**0.700**	CPS	**0.786**	
Individual Models:	**Classification**				
Fine Tuned	Mask	0.838			1:46:21
Fine Tuned	Phase	**0.713**			1:57:28
Scratch U-Net	Mask	**0.896**			19:47
Scratch U-Net	Phase	0.664			20:18
Individual Models:	**Regression**				
Fine Tuned			COD	**0.754**	1:51:52
Fine Tuned			CPS	0.680	1:41:11
Scratch U-Net			COD	0.717	17:07
Scratch U-Net			CPS	**0.738**	17:00

In summary, by comparing FT and from-scratch, in single-task settings, fine-tuning excels on phase and COD, whereas from-scratch U-Nets lead on mask and CPS. In multi-task settings, from-scratch baselines are better at most tasks, which indicates the multi-task FT pipeline might need more hyper-parameter tuning. Figure 5.1 presents example predictions from the models for a randomly selected image chip in the test set.

5.2 Benchmarking PEFT Fine-Tuning Strategies

This section benchmarks parameter-efficient fine-tuning (PEFT) methods against full fine-tuning (FFT), and the selection of loss functions. PEFT methods aim to reduce computational and memory costs by updating only a subset of parameters while maintaining competitive performance compared to FFT [29]. Table 5.3 summarizes the best results from each strategy across tasks. In further sections, we discuss the hyperparameters that were adjusted for the fine-tuning strategies.

5.2.1 Overall Comparison of FFT, LoRA, VPT

In Table 5.3, we consider the best performance of each training strategy for each individual task. Overall, while FFT achieves the strongest performance, LoRA narrows the gap substantially while reducing training time, and VPT underperformed compared to full fine-tuning and LoRA on metrics with training time significantly reduced. LoRA struggles to catch up with full fine-tuning when it comes to Optical Depth and Cloud Phase, but produces competitive results for Particle Size and Cloud Mask.

In the following subsections, we describe how LoRA and VPT were implemented with the transformer-based FM encoder and examine the impact of hyperparameter tuning. Low-Rank Adaptation updates model weights deeper in the encoder compared to VPT, where prompts are only incorporated at the first transformer layer. With this in addition to hyperparameters such as both rank and α to adjust, LoRA was more flexible to be experimented with for each task.

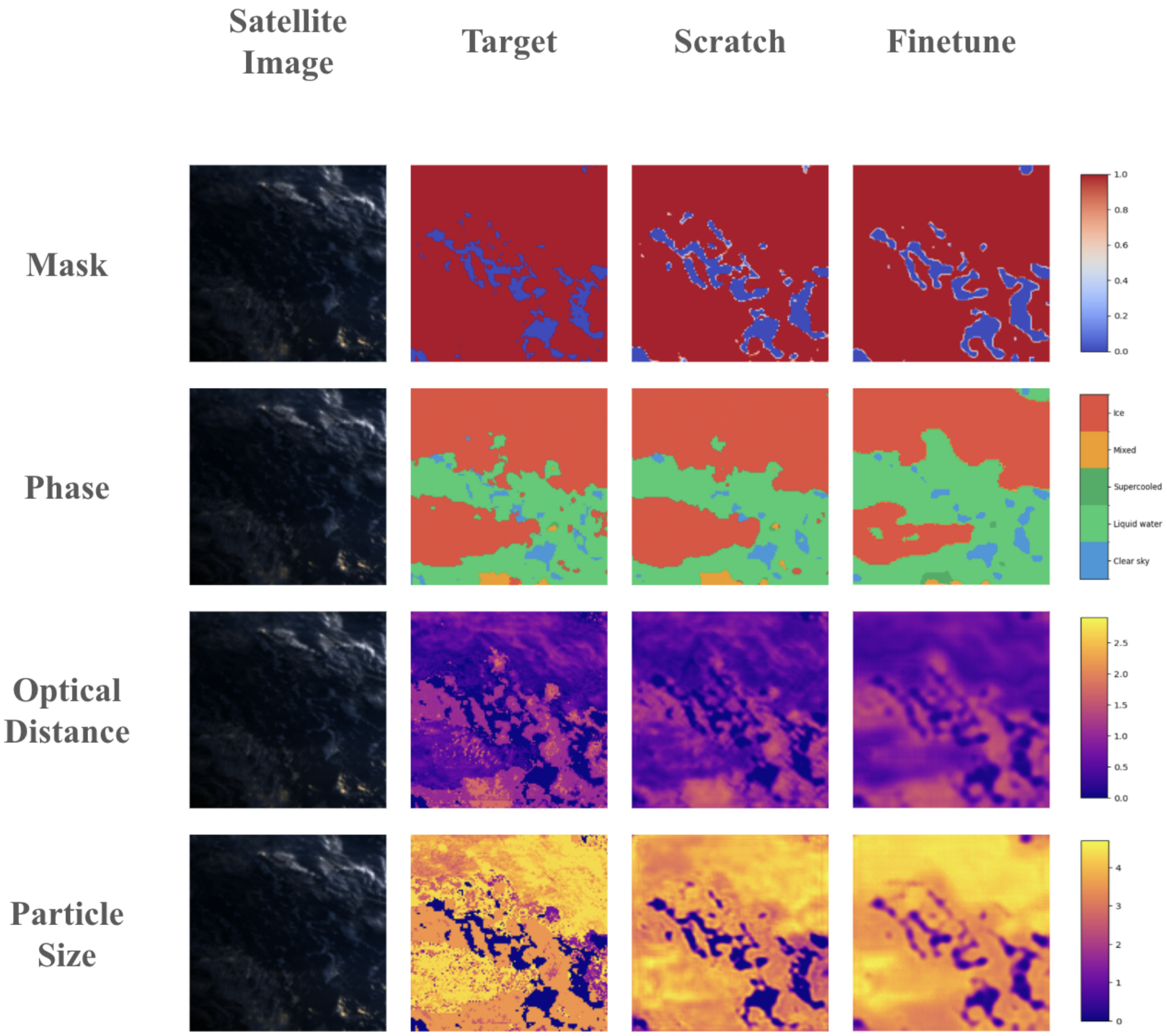

Figure 5.1: Predicted Cloud Properties for Image in Test Set.

5.2.2 Parameter Efficient Fine-Tuning via Visual Prompts

The first stage of SatVision-TOA is the patch embedding layer, where the input image is split into patches for input to the next transformer layer. Each patch is projected to a fixed-size embedding vector of size 512 for SatVision-TOA.

We implemented a variant of VPT-Shallow [6] by adding prompts element-wise to the first n patches of the first patch embedding layer of the encoder, where n is the number of prompts. Each 128×128 input image corresponds to 1024 patches. Since each prompt corresponds to a patch for a light-weight fine-tuning strategy, we ideally want $n << 1024$. Making n too small will reduce the amount of learning the model can do. As n grows larger, the prompts may cause ”catastrophic forgetting”, where the patch embedding layer suddenly forgets its pre-trained knowledge. We experimented with values of n to investigate what number might balance these goals - results are given in Table 5.4.

Table 5.3: Best Individual Task Performance for each Fine-Tuning Strategy.

Task	Hyperparams	Time to Train	mIOU/R^2
Mask			
FFT		1:46:21	**0.838**
LoRA	rank 32	1:11:56	0.816
VPT	300 prompts	1:01:32	0.675
Phase			
FFT		1:57:28	**0.713**
LoRA	rank 64	1:12:57	0.614
VPT	300 prompts	1:02:33	0.512
Optical Depth			
FFT		1:51:52	**0.754**
LoRA	rank 16	1:09:37	0.645
VPT	200 prompts	0:58:40	0.586
Particle Size			
FFT		1:41:11	**0.680**
LoRA	rank 32	1:08:10	0.664
VPT	100 prompts	0:58:55	0.574

Table 5.4: Performance of Fine-tuned Single-task Models with VPT.

Task	100 Prompts	200 Prompts	300 Prompts
Classification mIOU			
Mask	0.610	0.670	**0.675**
Phase	0.496	0.488	**0.512**
Regression R^2 Score			
COD	0.512	**0.586**	0.551
CPS	**0.574**	0.520	0.508

Performance for classification tasks increases with the larger number of prompts. Results are more mixed for regression: the R^2 score for CPS decreases with more prompts, and the best performance for COD is achieved with 200 prompts.

5.2.3 Parameter Efficient Fine-Tuning via Low-Rank Adaptation

Moving forward, we examine a PEFT strategy that works more directly with the encoder, SatVision-TOA. LoRA updates the model weights through a low-rank approximation of the weight matrices. We implemented LoRA by first applying it to attention query, key, and value layers within the attention block, as well as to the attention projection layer, an addition that we saw improved performance. Initial experiments with LoRA held alpha fixed at 16, then we used a consistent rank:alpha ratio of 1:2, which obtained stronger results.

Following this finding, we spent time adjusting rank. Figure 5.2 shows the results of adjusting the rank used for LoRA for the single-task models. Due to the large size of the encoder's layers, it was expected that the models would prefer higher ranks, but there is overall disagreement regarding one "best" rank from the tasks. Performance was not very sensitive to rank in general – for each task, there was not a drastic difference in performance with respect to the three ranks tested.

Finally, we observe how LoRA performs when used to train the fine-tuned multi-task model.

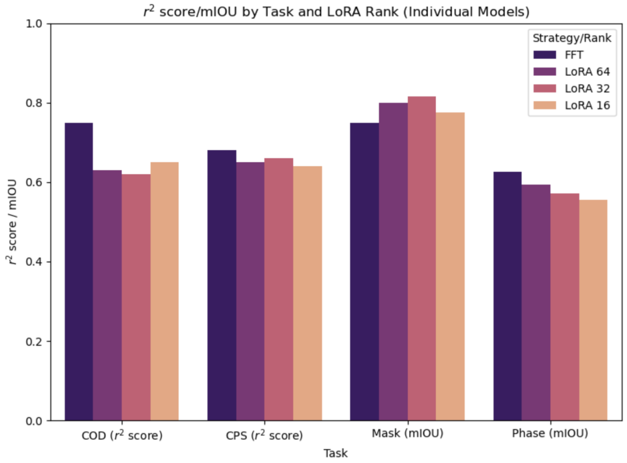

Figure 5.2: Adjusting LoRA Rank for Individual Fine-tuned Models.

Model	Task	mIOU	Task	R^2
FFT	Mask	0.757	COD	**0.754**
	Phase	**0.630**	CPS	**0.680**
LoRA 16	Mask	0.796	COD	0.645
	Phase	0.571	CPS	0.653
LoRA 32	Mask	**0.816**	COD	0.621
	Phase	0.600	CPS	0.664
LoRA 64	Mask	0.775	COD	0.629
	Phase	0.614	CPS	0.644

Table 5.5: Single-task Models: Full Fine Tuning vs. LoRA.

Table 5.6 shows the best 14-band MT model trained with full fine-tuning compared to the best 14-band MT model trained with LoRA. Task performance decreased by 11.375% on average, with the most drastic change seen in the R^2 score for CPS, which dropped 16.3%. Training time is reduced significantly, from almost two hours to just under an hour.

Table 5.6: Multi-task Model: Full Fine Tuning vs. LoRA (Rank 16).

Model	Task	mIOU	Task	R^2	Train Time
FFT	Mask	**0.895**	COD	**0.762**	1:53:54
	Phase	**0.701**	CPS	**0.793**	
LoRA	Mask	0.817	COD	0.672	**58:29**
	Phase	0.591	CPS	0.664	

In summary, when comparing FFT and PEFT, FFT delivers the strongest accuracy overall; LoRA narrows the gap substantially with much lower training time, while VPT is fastest but least accurate.

5.3 Using 14 or 16 Bands for Model Input

As noted in the introduction, SatVision-TOA was pre-trained on 14 spectral bands from MODIS, while the target dataset contains 16 spectral bands from ABI. To investigate methods of handling this difference, we introduced a lightweight preprocessor that incorporates all 16 ABI bands. We experimented with convolutional neural networks and multi-layer perceptrons to project the 16-channel ABI data to 14 channels. The most consistent architecture was a CNN with two hidden layers, each of dimension 16.

This experiment was designed to test whether leveraging all 16 bands provides an advantage over simply selecting the 14 ABI bands most similar in wavelength to the MODIS inputs. The two ABI without a corresponding MODIS band (and were dropped) were band 8 (6.19 µm) and band 13 (10.3 µm). For both strategies, we trained multi-task models under different hyperparameter settings, and the best results for each are reported in Table 5.7. Performance improved across all tasks, most notably for cloud phase and optical depth, when using the 14-band input. We hypothesize that any information gained from using all of the ABI bands does not outweigh the increased gap between fine-tuning and pre-training data. Learning how to adapt the 16-band input into a 14-channel input for the encoder was not as efficient nor effective as passing the input directly to the pre-trained encoder.

Table 5.7: Hyperparameters and Performance of 14-band and 16-band Multi-task Models.

Attribute	**14 Bands**	**16 Bands**
Loss Weights	2, 1, 0.75, 0.75	2, 1, 0.75, 0.75
Dice Weight	0.40	0.30
Learning Rate	3e-4	3e-4
Mask mIOU	**0.895** (+7.1%)	0.836
Phase mIOU	**0.701** (+31.3%)	0.534
COD R^2	**0.762** (+21.3%)	0.628
CPS R^2	**0.793** (+18.7%)	0.668

For single-task learning, the 14-band and 16-band models excel at different tasks. We used full fine-tuning and LoRA for each individual task, adjusting: learning rates, dice weight, and rank (if training with LoRA). The best results for each task are shown in Figure 5.3 – the 14-band models obtain stronger performance for segmentation tasks, but lag behind in regression. Considering the regression tasks are more complex than classification tasks, the preprocessor step may allow the model to access meaningful information from the additional two bands, an added complexity that might be beneficial for regression tasks.

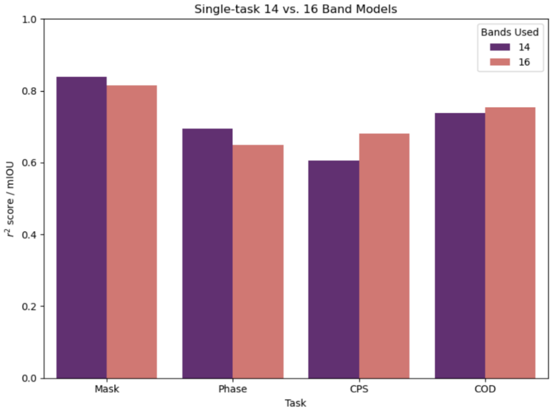

Figure 5.3: Single-task FT Models: 14 versus 16 Bands.

Attribute	**14 Bands**	**16 Bands**
Mask mIOU	**0.838** (+2.6%)	0.817
Phase mIOU	**0.713** (+8.4%)	0.658
COD R^2	0.742	**0.754** (+1.6%)
CPS R^2	0.611	**0.680** (+11.3%)

Table 5.8: 14-band vs 16-band Single-task Model Results.

In summary, for band mismatch, 14-band inputs outperform 16-band preprocessing in multi-task training; in single-task training, 16-bands tend to help regression, while 14-bands favor segmentation.

5.4 Ablation Studies of From-scratch Experiments

5.4.1 Per-pixel Analysis

As a baseline for spatially aware deep learning models, several preliminary pixel-by-pixel models were trained on just 200 images, with results displayed in Table 5.9. The first of these was a Multilayer Perceptron (MLP) with 3 hidden layers, trained with a learning rate of 0.001, 25 epochs, and a batch size of 2048 pixels. The same architecture was applied to all four prediction tasks with only minor modifications to the output layer to match the task.

To further probe the regression tasks (COD and CPS) where the MLP showed weaker performance, several algorithm based models were employed such as linear regression, random forest, and histogram-based gradient boosting. Histogram-based gradient boosting in particular performed surprisingly well compared to the MLPs. However, given the limited sample size, these results should be interpreted with caution as they may not generalize well to larger datasets. Overall, the pixel based models provided decent performance all around, but more substantial improvements are expected from architectures such as U-Nets which incorporate spatial context.

Table 5.9: Per-pixel benchmark evaluation trained from scratch.

Model	**Task**	**mIOU**	**Task**	$\mathbf{R^2}$
MLP	Mask	0.823	COD	0.724
	Phase	0.610	CPS	0.640
Decision Tree	Mask	**0.903**		
	Phase	**0.729**		
Linear Regression			COD	0.212
			CPS	0.299
Regression Forest			COD	0.663
			CPS	0.609
Hist Grad Boosting			COD	**0.786**
			CPS	**0.739**

5.4.2 Single U-Net Models

The U-Net was the spatial model of choice leveraged for all four cloud property variables. Each U-Net utilizes the Resnet-34 encoder with 4 skip connections and consists of approximately 36 million weights. With inputs now as full images, we trained on a full set of 15,000 images with a batch size of 128 images. With a learning rate of .00002, the models converged quickly to peak performance without major fluctuation of the validation accuracy. Results for each task are shown in Table 5.2.

Overall, the single U-Net models showed modest improvement over the MLPs as performance for all properties except COD improved. However, the single U-Net models and the subsequently tested U-Net based multi-task models did not perform as well compared to the Gradient Boosting model. A possible explanation is that with 75 times more images trained on, there was more variability in the training data, and spatial patterns became harder to detect.

5.4.3 Multi-task Model Evolution

The first version of the multi-task model from scratch used four U-Nets, one for each task. Cloud mask logits are appended to the feature map inputs for the other three variables (prior to the U-Net) to incorporate sequential dependency.

A new multi-task model was designed to make this dependency more explicit. The input first passes through a single 2D convolution that expands the channel dimension from 16 to 64. As in the previous version, four U-Nets are then used. The main distinction is that rather than adding the cloud mask logits at the input stage of the other three tasks, they are appended to the U-Net outputs before a final convolution layer.

Table 5.10: Evaluation of different multi-task configurations for from-scratch models.

Model	**Task**	**mIOU**	**Task**	$\mathbf{R^2}$
Logits appended before U-Net	Mask	0.819	COD	0.740
	Phase	0.642	CPS	0.742
Logits appended after U-Net	Mask	0.911	COD	0.767
(without batch normalization)	Phase	0.692	CPS	0.776
Logits appended after U-Net	Mask	**0.915**	COD	**0.769**
(with batch normalization)	Phase	**0.696**	CPS	**0.781**

As seen in Table 5.10, results improved across all metrics when the cloud mask logits are appended to the outputs of the U-Nets. This suggests the key design choice is determining which stage dependencies were introduced. By appending the cloud mask logits after the heavier computation in U-Nets, this additional channel will operate as an activation switch. This heuristic enforced a direct relationship between cloud mask prediction and the downstream tasks by reducing the risk that the dependency is diluted within intermediate feature maps.

Additionally, batch normalization after the first convolutional layer unsurprisingly improved all metrics, though not as substantially as altering the main model design. Normalization led to more stable training and faster generalization, resulting in slightly better results through the same number of epochs.

5.5 Tuning Loss Weights

This subsection discusses hyperparameter tuning of loss weights for different cloud retrieval tasks for both modeling approaches. Both multi-task approaches – training from scratch and fine-tuning – were evaluated under different task-weighting schemes. For the fine-tuned model, performance improved when regression losses were downweighted relative to classification losses. In contrast, the from-scratch model achieved its best results when regression losses were weighted twice as heavily as classification losses.

Incorporating the weighted sum of dice and CE loss caused the performance of the fine-tuned phase models to improve. Improvements were more marginal for the 14-band and multi-task models, but alternative fine-tuning strategies – training with 16 bands, LoRA, and VPT – saw more dramatic improvements. The following sections discuss details on loss-tuning experiments.

5.5.1 Loss Weight Tuning for Fine-Tuned Models

The mIOU values and visualizations of predictions when using CE loss for cloud phase revealed the models had difficulty handling edges and were adversely affected by the class imbalance in the dataset. To address this, we changed the phase loss to be a weighted sum of Dice loss and CE loss: $loss = dice\_weight \cdot dice\_loss + (1 - dice\_weight) \cdot CE\_loss$. Results of adjusting the dice weight are shown in Figure 5.4.

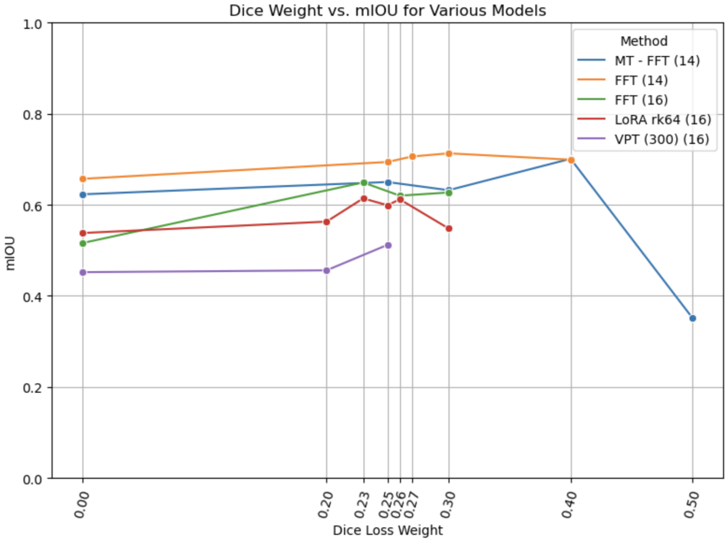

Figure 5.4: Adjusting Dice Weight for Phase Loss.

The model with the biggest improvement is the 16-band full fine-tuning model, whose mIOU improved from 0.514, lagging behind most other strategies, to 0.649.

5.5.2 Loss Weight Tuning for From-Scratch Models

Table 5.12 shows the results for a selection of different loss weights for the best-performing from-scratch multi-task architecture.

Prioritizing the cloud mask over regression degraded performance across all tasks, including cloud mask itself. This was unexpected, since emphasizing cloud mask was expected to improve accuracy in predicting the other properties. Instead, weighting regression boosted R^2 for both regression tasks, which was expected, as well as cloud phase's mIOU. While prioritizing regression led to improvements in 3 out of 4 evaluation metrics, the mIOU and R^2 scores are closely clustered and discrepancies may be explained by model variance rather than the loss weights.

In summary, the experiment finds that combining Dice loss and CE loss improves phase; task-weighting that emphasizes regression benefits from-scratch multi-task models, while fine-tuned models prefer relatively higher weights on classification.

Model	Dice Weight	Cloud Phase mIOU
MT-FFT	0.00	0.620
MT-FFT	0.25	0.651
MT-FFT	0.30	0.633
MT-FFT	0.40	0.700
MT-FFT	0.50	0.351
FFT 14-bands	0.00	0.657
FFT 14-bands	0.25	0.694
FFT 14-bands	0.27	0.711
FFT 14-bands	0.30	0.719
FFT 14-bands	0.40	0.700
FFT 16-bands	0.00	0.514
FFT 16-bands	0.23	0.649
FFT 16-bands	0.26	0.619
FFT 16-bands	0.30	0.630
LoRA rank-64 (16)	0.00	0.538
LoRA rank-64 (16)	0.20	0.561
LoRA rank-64 (16)	0.23	0.615
LoRA rank-64 (16)	0.25	0.598
LoRA rank-64 (16)	0.26	0.614
LoRA rank-64 (16)	0.30	0.546
VPT 300 prompts (16)	0.00	0.453
VPT 300 prompts (16)	0.20	0.455
VPT 300 prompts (16)	0.25	0.512

Table 5.11: Dice Loss across Finetuned Models Results.

Table 5.12: Adjusting Loss Weights in MT From-Scratch model with logits appended after U-Net.

Weights	Task	mIOU	Task	R^2	Train Time
(1, 1, 1, 1)	Mask	**0.915**	COD	0.769	44:30
	Phase	0.696	CPS	0.781	
(2, 2, 1, 1)	Mask	0.909	COD	0.767	44:45
	Phase	0.689	CPS	0.777	
(1, 1, 2, 2)	Mask	0.909	COD	**0.775**	45:59
	Phase	**0.700**	CPS	**0.786**	
(2, 1, 1, 1)	Mask	0.887	COD	0.734	38:40
	Phase	0.654	CPS	0.743	

6 Conclusions and Future Work

Our study offers comprehensive benchmarking of SatVision-TOA for individual and multi-task cloud property retrieval, exploring its adaptability for both classification and regression tasks across both single-task and multi-task architectures. From the present work, the following conclusions are drawn:

1. Fine-tuned models are competitive with models trained from scratch, while multi-task learning proves itself to be especially effective

2. From-scratch multi-task models achieve the best results across all four downstream tasks, and fine-tuned multi-task models outperform their single-task counterparts

3. Multi-task frameworks are able to efficiently share and leverage information across tasks

4. Low-Rank Adaptation was found to be the most competitive parameter-efficient fine-tuning alternative to full fine-tuning

5. Training directly with the 14 ABI bands most aligned with SatVision-TOA's bands achieved substantially better results than projecting all 16 ABI bands into 14 channels.

Future work could study how the different aspects of the encoder affect performance results – fine-tuning smaller variants of SatVision-TOA would address long training times and reveal the impact the number of parameters has on performance. The source code for this research can be accessed through the UMBC Big Data REU Github Repository [30].

Acknowledgments

This work is supported by the grant "REU Site: Online Interdisciplinary Big Data Analytics in Science and Engineering" from the National Science Foundation (grant no. OAC–2348755). We acknowledge the UMBC High Performance Computing Facility `hpcf.umbc.edu` and the financial contributions from NIH, NSF, CIRC, and UMBC for this work.

References

[1] N. A. of Sciences, Medicine, D. on Engineering, P. Sciences, S. S. Board, C. on the Decadal Survey for Earth Science, and A. from Space, *Thriving on our changing planet: A decadal strategy for Earth observation from space*. National Academies Press, 2019.

[2] S. Lu, J. Guo, J. R. Zimmer-Dauphinee, J. M. Nieusma, X. Wang, P. VanValkenburgh, S. A. Wernke, and Y. Huo, "Vision Foundation Models in Remote Sensing: A Survey," 2025. [Online]. Available: https://arxiv.org/abs/2408.03464

[3] S. Lu, J. Guo, J. R. Zimmer-Dauphinee, J. M. Nieusma, X. Wang, S. A. Wernke, Y. Huo *et al.*, "Vision foundation models in remote sensing: A survey," *IEEE Geoscience and Remote Sensing Magazine*, 2025.

[4] C. S. Spradlin, J. A. Caraballo-Vega, J. Li, M. L. Carroll, J. Gong, and P. M. Montesano, "SatVision-TOA: A Geospatial Foundation Model for Coarse-Resolution All-Sky Remote Sensing Imagery," 2024. [Online]. Available: https://arxiv.org/abs/2411.17000

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[6] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual Prompt Tuning," 2022. [Online]. Available: https://arxiv.org/abs/2203.12119

[7] G. Rotich and S. Sarkar, "Evaluating the Robustness of Foundation Models for Satellite Imagery," *IEEE Access*, vol. 13, pp. 107 720–107 735, 2025.

[8] X. Li, A. M. Sayer, I. T. Carroll, X. Huang, and J. Wang, "MT-HCCAR: Multi-task Deep Learning with Hierarchical Classification and Attention-Based Regression for Cloud Property Retrieval," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 3–18.

[9] Y. Xin, J. Yang, S. Luo, Y. Du, Q. Qin, K. Cen, Y. He, B. Fu, X. Yang, G. Zhai, M.-H. Yang, and X. Liu, "Parameter-Efficient Fine-Tuning for Pre-Trained Vision Models: A Survey and Benchmark," 2025. [Online]. Available: https://arxiv.org/abs/2402.02242

[10] F. Marti-Escofet, B. Blumenstiel, L. Scheibenreif, P. Fraccaro, and K. Schindler, "Fine-tune Smarter, Not Harder: Parameter-Efficient Fine-Tuning for Geospatial Foundation Models," 2025. [Online]. Available: https://arxiv.org/abs/2504.17397

[11] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin Transformer V2: Scaling Up Capacity and Resolution," 2022. [Online]. Available: https://arxiv.org/abs/2111.09883

[12] J. Yao and S. Jin, "Multi-Category Segmentation of Sentinel-2 Images Based on the Swin UNet Method," *Remote Sensing*, vol. 14, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/14/3382

[13] Z.-J. Gao, Y. He, and Y. Li, "A Novel Lightweight Swin-Unet Network for Semantic Segmentation of COVID-19 Lesion in CT Images," *IEEE Access*, vol. 11, pp. 950–962, 2023.

[14] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023, pp. 205–218.

[15] J. Pan, "Swin UNet: a memory-efficient and accurate deep learning model for medical image segmentation," in *Third International Conference on Machine Vision, Automatic Identification, and Detection (MVAID 2024)*, R. Jin, Ed., vol. 13230, International Society for Optics and Photonics. SPIE, 2024, p. 132300J. [Online]. Available: https://doi.org/10.1117/12.3035725

[16] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D. Zhang, "DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[17] J. Chen, X. Zhang, R. Li, and P. Zhou, "Swin-HAUnet: A Swin-Hierarchical Attention Unet For Enhanced Medical Image Segmentation," in *Pattern Recognition and Computer Vision*, Z. Lin, M.-M. Cheng, R. He, K. Ubul, W. Silamu, H. Zha, J. Zhou, and C.-L. Liu, Eds. Singapore: Springer Nature Singapore, 2025, pp. 371–385.

[18] S. Chen, Z. Feng, G. Xiao, X. Chen, C. Gao, M. Zhao, and H. Yu, "Pavement Crack Detection Based on the Improved Swin-Unet Model," *Buildings*, vol. 14, no. 5, 2024. [Online]. Available: https://www.mdpi.com/2075-5309/14/5/1442

[19] K. Hirata and T. Okita, "Brain Hematoma Marker Recognition Using Multitask Learning: SwinTransformer and Swin-Unet," 2025. [Online]. Available: https://arxiv.org/abs/2505.06185

[20] "NOAA Geostationary Operational Environmental Satellites (GOES) 16, 17, 18 & 19," last accessed 2025-09-01. [Online]. Available: https://registry.opendata.aws/noaa-goes/

[21] G.-R. C. W. Group and G.-R. S. Program, "NOAA GOES-R Series Advanced Baseline Imager (ABI) Level 1b Radiances."

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[23] A. Mao, M. Mohri, and Y. Zhong, "Cross-Entropy Loss Functions: Theoretical Analysis and Applications," 2023. [Online]. Available: https://arxiv.org/abs/2304.07288

[24] Dice, Lee R., "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409

[25] Thorvald Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biologiske Skrifter / Kongelige Danske Videnskabernes Selskab*, vol. 5, pp. 1–34, 1948.

[26] H. Kervadec and M. de Bruijne, "On the dice loss gradient and the ways to mimic it," 2023. [Online]. Available: https://arxiv.org/abs/2304.04319

[27] Y. Mu, J. Sun, and J. He, “The Combined Focal Cross Entropy and Dice Loss Function for Segmentation of Protein Secondary Structures from Cryo-EM 3D Density maps,” in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 3454–3461.

[28] Y. Mu, T. Nguyen, B. Hawickhorst, W. Wriggers, J. Sun, and J. He, “The combined focal loss and dice loss function improves the segmentation of beta-sheets in medium-resolution cryo-electron-microscopy density maps,” *Bioinformatics Advances*, vol. 4, no. 1, p. vbae169, 11 2024. [Online]. Available: https://doi.org/10.1093/bioadv/vbae169

[29] F. Marti-Escofet, B. Blumenstiel, L. Scheibenreif, P. Fraccaro, and K. Schindler, “Fine-tune Smarter, Not Harder: Parameter-Efficient Fine-Tuning for Geospatial Foundation Models,” 2025. [Online]. Available: https://arxiv.org/abs/2504.17397

[30] D. Murphy, K. Zhang, C. Parten, A. Sterling, H. Zhang, X. Li, J. A. Caraballo-Vega, J. Gong, M. Carroll, and J. Wang, “UMBC Big Data REU 2025 Team 1 Github,” 2025. [Online]. Available: https://github.com/big-data-lab-umbc/big-data-reu/tree/main/2025-projects/team-1