

Fine-tune Smarter, Not Harder: Parameter-Efficient Fine-Tuning for Geospatial Foundation Models

Francesc Marti Escofet^{1,*}, Benedikt Blumenstiel^{1,*}(✉), Linus Scheibenreif²,
Paolo Fraccaro¹, and Konrad Schindler²

¹ IBM Research Europe

² ETH Zurich

benedikt.blumenstiel@ibm.com

Abstract. Earth observation (EO) is crucial for monitoring environmental changes, responding to disasters, and managing natural resources. In this context, foundation models facilitate remote sensing image analysis to retrieve relevant geoinformation accurately and efficiently. However, as these models grow in size, fine-tuning becomes increasingly challenging due to the associated computational resources and costs, limiting their accessibility and scalability. Furthermore, full fine-tuning can lead to forgetting pre-trained features and even degrade model generalization. To address this, Parameter-Efficient Fine-Tuning (PEFT) techniques offer a promising solution. In this paper, we conduct extensive experiments with various foundation model architectures and PEFT techniques to evaluate their effectiveness on five different EO datasets. Our results provide a comprehensive comparison, offering insights into when and how PEFT methods support the adaptation of pre-trained geospatial models. We demonstrate that PEFT techniques match or even exceed full fine-tuning performance and enhance model generalisation to unseen geographic regions, while reducing training time and memory requirements. Additional experiments investigate the effect of architecture choices such as the decoder type or the use of metadata, suggesting UNet decoders and fine-tuning without metadata as the recommended configuration. We have integrated all evaluated foundation models and techniques into the open-source package TerraTorch to support quick, scalable, and cost-effective model adaptation.

Keywords: Foundation Models · Earth Observation · PEFT

1 Introduction

Earth observation (EO) is concerned with collecting information about the Earth using remote sensing, and it has become indispensable for monitoring environmental changes, managing natural resources, and enabling rapid responses to natural disasters [40, 4]. The ever-growing volume and complexity of EO data

* Equal contribution.

pose substantial challenges for effective analysis and interpretation. Advances in deep learning have helped to mitigate these challenges by automating and improving the accuracy of EO data analysis across diverse applications, including flood detection, land use classification, and climate monitoring [40].

Foundation models (FMs), which leverage self-supervised learning, have recently emerged as powerful tools capable of capturing general-purpose representations from data. Domain-specific geospatial foundation models (GeoFMs), like Clay [9] or Prithvi [21,31], are pre-trained on large-scale satellite datasets. These models promise superior performance, improved generalization capabilities, and ease of use across geospatial tasks [24,31,13]. For instance, recent studies have demonstrated that GeoFMs outperform traditional deep learning baselines on tasks such as land cover mapping and crop classification [31].

However, as GeoFMs follow the general trend towards ever larger models [9,31], fine-tuning them becomes more expensive and requires substantial computational resources. This limitation restricts their broader application and adoption. Parameter-Efficient Fine-Tuning (PEFT) techniques provide a viable solution to this challenge. By updating only a small subset of the model parameters during training, PEFT significantly reduces computational demands and memory usage without sacrificing performance [15]. PEFT techniques have established themselves in natural language processing and computer vision [15], but to the best of our knowledge, they have not yet been systematically investigated in the context of GeoFMs.

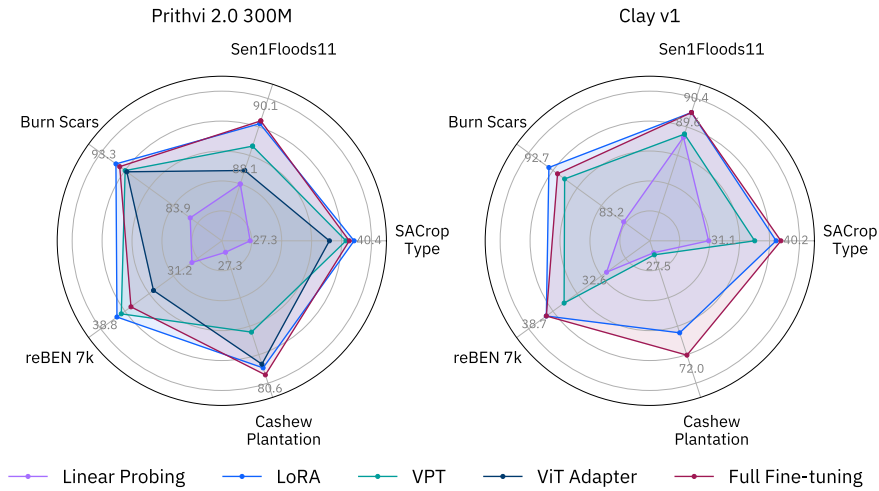


Fig. 1: Comparison between PEFT techniques for Prithvi 2.0 300M (left) and Clay v1 (right) with linear decoders. The test mIoU (%) \uparrow is reported for all five datasets. Axes are scaled by min-max with dataset-specific buffers. The best and worst performance for each dataset is annotated.

Our work presents the first comprehensive evaluation of PEFT techniques applied to GeoFMs. Specifically, we investigate multiple GeoFM architectures and PEFT methods across five distinct EO datasets, providing insights into their effectiveness and limitations. Our experimental findings, illustrated in Figure 1, indicate that the Low-Rank Adaptation (LoRA) method matches or even exceeds the performance of full fine-tuning, while enhancing generalization to unseen geographic regions.

Our key contributions can be summarized as follows: (1) we perform the first extensive comparative analysis of PEFT methods for GeoFMs across diverse EO tasks, (2) we explore the impact of architectural decisions, such as decoder choice and the inclusion of metadata, and (3) we evaluate model generalization to unseen geographical regions and inputs, demonstrating the robustness of PEFT. All evaluated models and techniques are integrated into the open-source package **TerraTorch** [13] to promote the application of efficient model adaptation techniques in Earth observation. The fine-tuning configs and code for our experiments are available at <https://github.com/IBM/peft-geofm>.

2 Related Work

Due to the growing volume of EO data available, machine learning techniques have become crucial to extract insights from these observations. A large body of work addresses the automated processing of EO data with deep learning techniques for a broad set of tasks, including, among others, environmental monitoring, land-cover mapping, and agricultural applications (see [40] for a review).

Foundation Models in EO. Foundation models are large neural network models pre-trained on vast datasets through self-supervised learning [3]. They have led to significant progress in domains such as natural language processing [5] or computer vision [6], and multiple foundation models for EO have recently been released [21, 36, 33]. They all leverage extensive collections of unlabelled remote sensing data to pre-train a generic data representation, using techniques such as contrastive learning [14] or masked autoencoding [16]. E.g., Prithvi 1.0 [21] uses time-series of remote sensing data in conjunction with masked autoencoding to create an EO foundation model. Similarly, approaches like Clay [9] and DOFA [36] leverage multi-modal remote sensing data with wavelength encodings to pre-train models for EO tasks. Contrastive GeoFM methods such as DeCUR [33] achieve strong performance on some downstream tasks with relatively small model architectures. Overall, foundation models owe their performance to large model capacity (resp., size) and massive amounts of unlabeled data for pre-training. In the EO domain, recent models like Prithvi 2.0 [31] demonstrate improved downstream performance by further scaling up both model size and data volume.

PEFT Methods. EO foundation models are typically adapted for a concrete downstream task by fine-tuning all model parameters in a supervised fashion

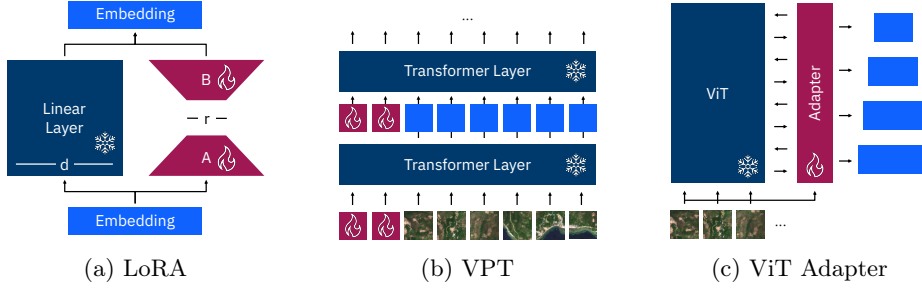


Fig. 2: Architectures of PEFT techniques, visualized with embeddings (blue), frozen (dark blue), and learnable layers (red). LoRA introduces small weight matrices that modulate linear layers. VPT adds learnable prompt tokens to each layer of a ViT. The ViT Adapter is a smaller parallel network that interacts with multiple transformer layers and maps their activations to a feature hierarchy.

for the target task [31,33]. With growing model size, such full fine-tuning incurs increasingly high computational costs and memory requirements. That bottleneck can be alleviated with PEFT methods, which drastically limit the number of fine-tuned parameters while still maintaining good downstream performance. The core idea of PEFT is to remain efficient by making an informed choice about which parameters of the pre-trained model to update during fine-tuning [38], or by introducing a small set of new parameters into the model [7] or its input space [22] and training only those. Several recent works introduce PEFT techniques specifically for EO data. For instance, methods like AiRs [19] and TEA [18] introduce adapters with additional residual connections for efficient fine-tuning of pre-trained transformers for EO. Similarly, UPetu [10] combines quantization with a learnable prompt to adapt pre-trained convolutional models to dense EO tasks. DEFLECT [32] further contributes to this direction by introducing a patch embedding layer (UPE) and an attention mechanism (uAtt) to adapt RGB-pretrained GeoFMs to multispectral images. Recent work also evaluates different PEFT methods on specific EO tasks, like the segmentation of winter-wheat fields [37], or employs PEFT for unsupervised domain adaptation in EO [30].

Unlike existing works, this work compares multiple PEFT methods across different multispectral EO foundation models and geospatial tasks.

3 Methods

We first introduce the different PEFT methods as well as the GeoFMs models and decoders used in our experiments.

PEFT Techniques. We combine three different PEFT methods with GeoFMs, see Figure 2. Low-Rank Adaptation [17] (LoRA) introduces low-rank matrices into pre-trained transformer models (A and B in Figure 2). Fine-tuning the

low-rank matrices reduces the amount of trainable parameters for a pre-trained weight matrix $W \in \mathbb{R}^{d \times d}$ from d^2 to $2 \cdot d \cdot r$, where r is the bottleneck dimension.

Instead of adapting the pre-trained model, visual prompt tuning [22] (VPT) introduces additional trainable prompt parameters into the input sequence of the embedded image patches before the GeoFM processes them. A variant of that idea introduces learnable prompts in all transformer layers (VPT-Deep), the configuration we adopted here.

ViT Adapters [7] address shortcomings of vision transformers (ViT) [11] in dense prediction tasks by introducing spatial inductive biases. A small convolutional model extracts spatial features from the input image, which are then injected into the ViT through cross-attention modules. Eventually, the ViT adapters extract multi-scale features from the transformer layers which are combined for a final dense prediction. As ViT Adapters involve integrating convolutional branches, cross-attention modules, and multi-scale feature fusion, their implementation is non-trivial. Due to this significant engineering overhead, we only study them for the Prithvi family of models.

Table 1: Overview of the EO backbones and their pre-training characteristics.

Model	Archi.	# Params.	Sensors	# S-2 bands	# Samples	Metadata
DeCUR [33]	ResNet50	25M	S1, S2	13	1M	✗
Clay v1 [9]	ViT-B/8	92M	various	10	70M	✓
Prithvi 1.0 [21]	ViT-B/16	86M	HLS	6	$250k \times 3$	✗
Prithvi 2.0 [31]	ViT-L/16	304M	HLS	6	$4.2M \times 4$	✗ / ✓

Foundation Models. We evaluate the different PEFT schemes with four GeoFMs and provide an overview in Table 1. DeCUR [33] is pre-trained with contrastive learning. Specifically, the Decoupling Common and Unique Representations (DeCUR) approach separates shared and modality-specific features in the embedding. The model utilises per-modality ResNet-50 backbones and was pre-trained on the SSL4EO-S12 [34] dataset, using all 13 Sentinel-2 L1C bands and two Sentinel-1 GRD bands.

Prithvi 1.0 [21] employs masked autoencoder (MAE) [16] pre-training and is based on a ViT-B backbone with a patch size of 16. It is trained on Harmonized Landsat and Sentinel-2 data exclusively from the USA, limited to six spectral bands. The data consists of short time series with three timestamps to support multi-temporal inputs naturally. Like its predecessor, Prithvi 2.0 [31] employs MAE pre-training but uses a global dataset and includes larger backbones based on ViT-L/16 and ViT-H/14. While still training on HLS and covering six bands, optional temporal and location embeddings are introduced to enable better spatiotemporal reasoning. We evaluate the Prithvi 2.0 300M version, and do not use metadata unless specified otherwise.

Clay v1 [9] also pre-trains a ViT-B backbone, with a patch size of 8, by MAE augmented with DINO-style self-distillation [6]. It is trained on a diverse dataset that includes Landsat-8 and -9, Sentinel-1 and -2, NAIP, and LINZ imagery. It features dynamic patch embedding weights similar to DOFA [36], adjusted according to the input spectral bands. The model integrates additional metadata like latitude/longitude, ground sampling distance (GSD), and acquisition time into the positional embeddings.

Decoder. The downstream performance of a GeoFM depends on the quality of the encoder features and the decoder’s capability to leverage them. We evaluate four decoder architectures: a linear decoder, Fully Convolutional Network (FCN) [26], UperNet [35], and UNet [29]. With its minimal complexity of a single transposed convolution without nonlinearities, the linear decoder serves as a way to evaluate the feature embeddings as directly as possible.

While the linear decoder is well-suited for controlled evaluations of the encoder, real-world applications require decoders capable of exploiting features in context. The FCN decoder [26] reconstructs spatial details through a sequence of convolutional blocks, each composed of a transposed convolution for upsampling, followed by a 3×3 convolution, Layer Normalization, and GeLU activation. The UperNet decoder [35] further improves upon this by combining multi-scale features through a Feature Pyramid Network (FPN) [25] and capturing global context with a Pyramid Pooling Module (PPM) [39]. Lastly, the UNet decoder [29] integrates multi-scale encoder features via skip connections, preserving detailed spatial information through decoder blocks consisting of interpolation, concatenation, and successive convolutional layers paired with Batch Normalization and ReLU activations.

The UperNet and UNet decoders require hierarchical, multi-scale features from multiple encoder layers. In our experiments, we use features from three intermediate encoder layers and the final one. While ResNet backbones naturally contain these hierarchical features, ViT embeddings require spatial upscaling. We therefore add learned upsampling layers that generate appropriate multi-scale features compatible with the UperNet and UNet decoders [13].

4 Experimental Setup

We have conducted extensive experiments to compare different PEFT techniques and GeoFMs. In the following, we describe the datasets used for evaluation and the fine-tuning protocol in detail.

Datasets. An overview of the data used for downstream evaluations is provided in Table 2. We restrict all experiments to multispectral optical inputs only.

Sen1Floods11 [4] includes Sentinel-1 (S1) GRD and Sentinel-2 (S2) L1C satellite images at 10m resolution covering eleven different flooding events on six different continents. The task is to segment surface water. In our evaluation, we use the 431 binary masks annotated by expert analysts. A hold-out set of 15 samples from Bolivia serves to test geographic generalization.

Table 2: Overview of the evaluation datasets and their sample counts, including geographic hold-out sets (GHOS).

Dataset	Sensors	Patch size	Classes	# Train	# Val.	# Test	# GHOS
Sen1Floods11 [4]	S1, S2	512×512	2	252	89	90	15
Burn Scars [28]	HLS	512×512	2	524	160	120	–
reBEN 7k [8]	S1, S2	120×120	19	4690	946	945	807
m-Cashew plant. [23]	S2	256×256	7	1350	400	50	–
m-SA Crop type [12]	S2	256×256	10	3000	1000	1000	–

The Burn Scars dataset [28] contains images of wildfire scars, gathered with the help of the Monitoring Trends in Burn Severity (MTBS) historical fire database. Annotations from the continental US, from 2018 to 2021, were co-registered with Harmonized Landsat Sentinel-2 (HLS) observations at 30m resolution. The 804 images include six spectral channels: Blue, Green, Red, NIR Narrow, SWIR 1, and SWIR 2. The original Burn Scars dataset provides a training and validation split. We propose new splits that include a test set. To further exclude the possibility of data leakage, we create the splits with a 5km buffer between training and evaluation samples and ensure that close-by patches are assigned to the same split. We make the splits public at <https://github.com/IBM/peft-geofm> to ensure reproducibility.

reBEN 7k is a subset of the larger Refined BigEarthNet (reBEN) [8], sampled to preserve class balance. reBEN has over 400k samples, each consisting of S1 GRD and S2 L2A patches as well as land-use/land-cover annotations with 19 classes. While reBEN supports both semantic segmentation and multi-label classification, we only use the semantic segmentation task. Repeated fine-tuning on the full reBEN dataset would be computationally infeasible for our experiments. Therefore, we create a smaller subset following the example of BEN-ge 8k [27]. Compared to BEN-ge 8k, we derive our version from the latest edition of BigEarthNet [8] and introduce a geographic hold-out set (GHOS) for generalization experiments. Our subset is based on the land-cover multi-labels and designed to retain class diversity. To ensure fair representation of all classes, we draw 250 samples per class from the official training set and 50 samples per class for validation and testing, excluding Austria and Ireland. Then, up to 50 samples per class were drawn only from those countries’ test sets to obtain an additional GHOS for testing model generalization to unseen geographic regions.

As further test cases, we include the m-Cashew Plantation [23] and m-SA Crop Type [12] datasets provided by GEO-bench [24]. The former provides labeled images of cashew plantations across a 120 km² region in central Benin. Each pixel is assigned to one of seven classes, including *well-managed plantation*, *poorly-managed plantation*, and *no plantation*, enabling detailed land-use analysis. SA Crop Type [12] contains images and crop type labels for South Africa, with each pixel classified into one of ten categories, such as *weeds*, *wheat*, or *rooibos*. In both cases, we use the versions preprocessed by GEO-bench [24].

Table 3: Parameter counts for EO backbones and PEFT components. Percentages refer to the number of PEFT parameters relative to the encoder parameters.

Model	# Encoder Params.	# VPT Params.	# LoRA Params.	# ViT Adapt. Params.
Clay v1 [9]	92M	0.9M (1.0%)	2M (2.2%)	-
Prithvi 1.0 100M [21]	86M	0.9M (1.1%)	2M (2.4%)	12M (13.8%)
Prithvi 2.0 300M [31]	304M	2.5M (0.8%)	5.5M (1.8%)	20M (6.6%)

Fine-tuning Procedure. To ensure robust and reproducible results, we follow established experimental protocols [24,31] involving hyperparameter optimization (HPO) and repeated runs. For each combination of model architecture, dataset, and PEFT method, we carry out Bayesian hyperparameter optimization with 16 trials to select the optimal learning rate, implemented using TerraTorch Iterate [20]. With the best learning rate, we run the fine-tuning five times for each configuration, starting from different random seeds, and report the average.

Based on prior experiments, PEFT methods are configured as follows: For LoRA, adapters with bottleneck dimension $r=16$ are added to the linear projections in the feed-forward layers, as well as to the query and value matrices of the attention layers of the GeoFMs. For VPT, we include 100 learnable prompts in all transformer layers (VPT-Deep). The specific parameter counts for each model and PEFT configuration are described in Table 3. VPT and LoRA increase the model parameters by only up to 2.4%. ViT Adapter introduces significantly more parameters, increasing the model parameter count by up to 13.8%.

For all experiments, we use the AdamW optimizer with $\beta_1=0.9$, $\beta_2=0.999$ and ReduceLROnPlateau scheduler (*patience*=4 epochs, *factor*=0.5). The batch size was set to 8, except for reBEN-7k, where we increased it to 32 to account for the smaller input size and higher number of images. We find that a small batch size with a correspondingly higher number of gradient updates leads to faster convergence on relatively small EO datasets. All models were trained on a single NVIDIA A100 GPU for up to 100 epochs, applying early stopping after 15 epochs without improvement. As far as possible, each GeoFM receives the Sentinel-2 bands it was pre-trained for: Ten bands for Clay, six for Prithvi, and all 13 L1C bands for DeCUR. If any bands are missing in a dataset (e.g., Burn Scars only has six bands), the patch embedding automatically adapts to the available subset via TerraTorch [13]. For reBEN-7k, the images are padded to 128×128 pixels with the *reflect* method to ensure compatibility across all architectures.

5 Experiments

We conduct a series of experiments to evaluate the effectiveness of PEFT methods for GeoFMs. Our analyses cover three aspects: (1) a comparison of PEFT techniques against full fine-tuning, (2) an assessment of generalization across unseen geographic regions, and (3) an evaluation of different decoder architectures.

5.1 Parameter-Efficient Fine-Tuning

Our experiments use a linear decoder to isolate the impact of PEFT techniques on geospatial foundation models. It ensures a controlled evaluation, as more complex decoders may obscure differences by significantly increasing model capacity. Table 4 presents the results of PEFT experiments, including three baseline models: a randomly initialized UNet [29] and a ViT-B/16 pre-trained on ImageNet-21k [11]. DeCUR [33], a ResNet-based GeoFM, is also considered a baseline since the selected PEFT methods are designed for ViTs. We perform five training runs with varying random seeds and report average performances for readability. Standard deviations are provided in the supplementary material. In most cases, the standard deviation is below 0.6pp, with some exceptions for the Cashew Plantation dataset.

Table 4: Test mIoU (%) \uparrow for the evaluated models and PEFT techniques, averaged over five runs. We highlight the best-performing combination in bold and underline the second-best one.

Model	Method	Sen1F11	Burn Scars	reBEN 7k	Cashew Plant.	SACrop	Mean
UNet (Rand.)	Full FT	90.75	89.34	34.62	81.81	34.99	66.30
ViT (IN-21k)	Full FT	89.19	92.31	36.15	80.05	37.98	67.14
DeCUR	Linear Prob.	80.83	78.17	28.29	16.05	20.93	44.85
	Full FT	86.87	89.48	36.05	79.59	34.21	65.24
Clay v1	Linear Prob.	89.57	83.17	32.65	27.50	31.10	52.80
	VPT	89.67	90.67	36.87	28.38	36.89	56.50
	LoRA	<u>90.41</u>	92.74	<u>38.67</u>	62.29	39.64	64.75
	Full FT	<u>90.41</u>	91.58	<u>38.67</u>	72.03	<u>40.22</u>	66.58
Prithvi 1.0 100M	Linear Prob.	88.78	83.23	27.07	25.33	26.06	50.10
	VPT	89.03	85.17	29.98	29.24	29.62	52.61
	LoRA	89.33	89.34	30.60	53.00	31.58	58.77
	ViT Adapter	87.72	89.47	32.90	73.31	32.34	63.15
	Full FT	89.02	89.31	31.82	77.41	32.59	64.03
Prithvi 2.0 300M	Linear Prob.	88.08	83.90	31.25	27.29	27.26	51.55
	VPT	89.31	92.16	38.40	62.00	39.33	64.24
	LoRA	90.04	93.33	38.84	77.53	40.35	<u>68.02</u>
	ViT Adapter	88.52	91.95	35.14	75.92	37.24	65.75
	Full FT	90.13	<u>92.85</u>	37.42	<u>80.58</u>	39.74	68.14

While performance varies across datasets, Prithvi 2.0 300M is the best model overall, outperforming Clay v1 and the ImageNet-pre-trained ViT by +1pp. Notably, the randomly initialized UNet surpasses DeCUR and Prithvi 1.0. However, the superior UNet decoder may contribute to this advantage. Prithvi 1.0 exhibits

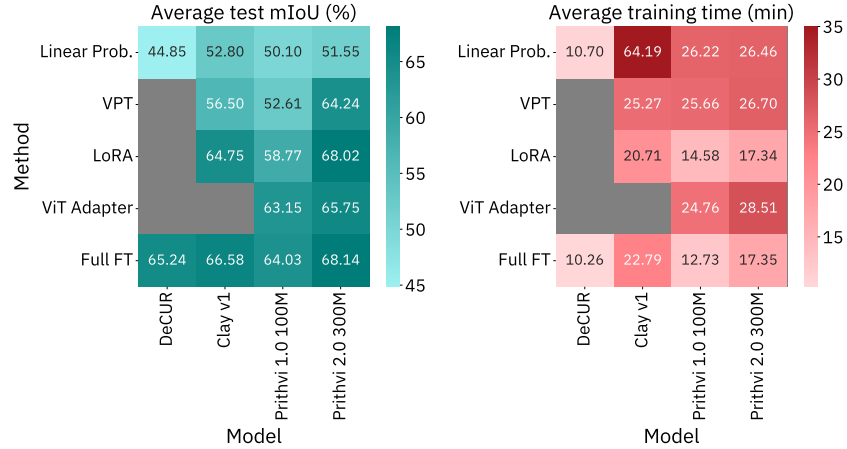


Fig. 3: Test mIoU \uparrow and training time \downarrow averaged over five datasets with five runs each. We trained all models up to 100 epochs and used early stopping after 15 epochs, resulting in best efficiencies for full fine-tuning and LoRA.

the lowest fine-tuning performance, likely due to its limited pre-training data. In contrast, Prithvi 2.0, trained on a larger dataset, achieves +4pp improvement, reinforcing the importance of scaling both data and model size [31]. Among PEFT techniques, LoRA performs on par with, or better than, full fine-tuning for Clay and Prithvi 2.0. In contrast, VPT and ViT Adapter generally underperform full fine-tuning. On reBEN 7k, VPT and LoRA outperform full fine-tuning using Prithvi 2.0, while with the smaller Prithvi 1.0, ViT Adapter scores best. In general, PEFT methods are particularly performant when combined with large GeoFMs, like Prithvi 2.0.

Figure 3 shows test performance compared with training time. LoRA and full fine-tuning require a similar training time for Prithvi 2.0, despite LoRA freezing most layers. This suggests that the fixed batch size for all experiments may limit LoRA’s expected speedup. However, the lower memory footprint of LoRA makes it well-suited for GPUs with limited capacity and enables larger batch sizes for efficiency gains on larger datasets (see memory footprints in the supplementary material). Training time is also strongly influenced by model architecture. Clay, which uses a patch size of 8, must process four times more tokens than Prithvi 1.0 and takes more than twice as long, despite their similar parameter counts. Prithvi 2.0, being three times larger than Clay, still trains faster.

An interesting finding is the unexpectedly long training time of linear probing shown in Figure 3, as the fitting of a less flexible linear function converges slower. Notably, Clay outperforms Prithvi 2.0 by 1pp in linear probing, but ranks lower in all other settings. However, this trend does not hold for larger decoders, which can model non-linear patterns. Additional experiments with a UNet decoder showed training times comparable to full fine-tuning. When combined with larger decoders, our results suggest that freezing the encoder is viable for simple tasks

like water mapping but leads to performance drops in more complex tasks like crop type classification.

Overall, our results indicate that PEFT techniques offer an effective trade-off between full fine-tuning and frozen encoders, particularly when memory efficiency is critical. Our findings further suggest that LoRA can match or surpass full fine-tuning, depending on the GeoFM and downstream task.

5.2 Model Generalization

We evaluate geographic generalization using geographic hold-out sets (GHOS). First, we compare Prithvi 2.0 300M embeddings after pre-training, full fine-tuning, and LoRA fine-tuning. Figure 4 visualizes the Sen1Floods11 embeddings, while reBEN 7k is provided in the supplementary material. The model trained without geographic metadata naturally clusters images by region. The geographic hold-out set from Bolivia forms two clusters after pre-training. Full fine-tuning disrupts this structure, suggesting that it forgets some pre-trained features. At the same time, LoRA better preserves geographic clustering, highlighting its potential to mitigate catastrophic forgetting, a key challenge in fine-tuning foundation models.

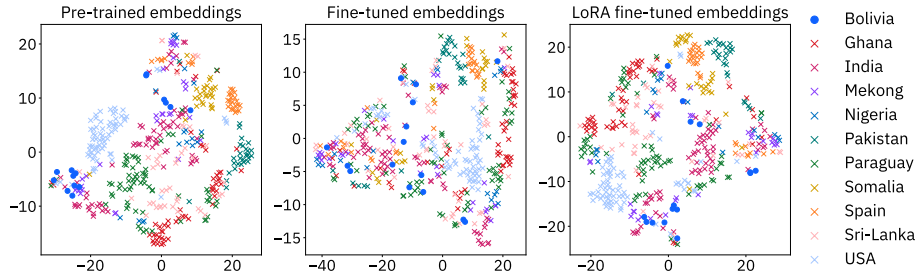


Fig. 4: Prithvi 2.0 300M embeddings of the Sen1Floods11 dataset colored by region. We averaged the patch embeddings per image and applied t-SNE.

To quantify these patterns, we compute the minimum Euclidean distance of each sample to the training set and average over validation, test, and hold-out splits. The results (see supplementary material) confirm that GHOS samples are significantly farther from training data than validation and test samples, which follow the training distribution. Full fine-tuning reduces distances for all Sen1Floods11 splits by a factor of two but unexpectedly increases them for reBEN 7k. LoRA fine-tuning increases distances for both datasets but maintains the pre-training embedding structure in the t-SNE plot. In all cases, GHOS distances remain higher than those of validation and test sets, highlighting the challenge of geographic generalization.

Table 5 quantifies the geographic generalization gaps: on average, mIoU drops by 7.79pp for reBEN 7k and 7.31pp for Sen1Floods11 across all settings and

Table 5: Comparison between the in-distribution test set and the geographic hold-out set (GHOS) using mIoU (%) \uparrow , averaged over five runs. We highlight the two best-performing combinations in bold and underlined.

Model	Method	Sen1Floods11		reBEN 7k	
		Test mIoU	GHOS mIoU	Test mIoU	GHOS mIoU
UNet (Rand.)	Full FT	90.75	87.81	34.62	25.43
ViT (IN-21K)	Full FT	89.19	82.67	36.15	27.14
DeCUR	Linear Prob.	80.83	74.76	28.29	23.29
	Full FT	86.87	85.84	36.05	27.60
Clay v1	Linear Prob.	89.57	84.88	32.65	24.38
	VPT	89.67	87.29	36.87	28.77
	LoRA	<u>90.41</u>	88.88	<u>38.67</u>	28.44
	Full FT	<u>90.41</u>	<u>88.33</u>	<u>38.67</u>	29.07
Prithvi 1.0 100M	Linear Prob.	88.78	79.57	27.07	24.04
	VPT	89.03	64.35	29.98	22.65
	LoRA	89.33	61.25	30.60	24.09
	ViT Adapter	87.72	82.31	32.90	25.66
	Full FT	89.02	74.16	31.82	24.53
Prithvi 2.0 300M	Linear Prob.	88.08	83.19	31.25	24.86
	VPT	89.31	86.19	38.40	<u>29.83</u>
	LoRA	90.04	87.57	38.84	30.21
	ViT Adapter	88.52	84.94	35.14	26.99
	Full FT	90.13	82.07	37.42	28.12

models. DeCUR exhibits the smallest drop with full fine-tuning (-4.74pp), while Prithvi 1.0 experiences the largest with LoRA (-17.30pp). The best-performing models across both in-distribution and GHOS sets are Prithvi 2.0 300M with LoRA, Clay (full fine-tuning), and Clay with LoRA. Notably, LoRA significantly improves geographic generalization for Prithvi 2.0 300M but has minimal impact for Clay. These results highlight that geographic generalization remains an open challenge for GeoFMs. However, they outperform a randomly initialized UNet (+2.2pp on GHOS) and an ImageNet-pretrained ViT (+3.7pp), with LoRA proving particularly beneficial for generalization which is in line with results from literature [2]. That being said, PEFT methods may struggle in scenarios with extreme distribution shifts, such as across sensor modalities (e.g., from optical to SAR) or when dealing with very high-resolution imagery. In such cases, pre-trained features are expected to transfer poorly, and PEFT’s restricted capacity to adapt the encoder weights could become a bottleneck [1].

To assess model robustness to varying input bands, we conducted experiments with DeCUR and Clay using only six spectral bands (Prithvi bands) under both linear probing and full fine-tuning (results in the supplementary material). Reducing input channels resulted in a minor performance drop, typically within

1-2pp. No significant difference was observed between frozen encoder models and fully fine-tuned ones, suggesting that GeoFMs can adapt to missing bands without updating early layers. The performance loss is therefore attributed to the missing spectral information rather than to poor model adaptation.

5.3 Decoder Architecture

Figure 5 compares different decoder architectures across all evaluated foundation models using full fine-tuning. We also include Prithvi 2.0 300M TL, which incorporates additional metadata for reference. The decoder performance varies across GeoFMs, but the UNet decoder consistently achieves strong results, ranking within the top two across all architectures. FCN performs best for Prithvi 2.0 but lags behind UNet for other models. UperNet, despite having a similar parameter count, underperforms across most settings.

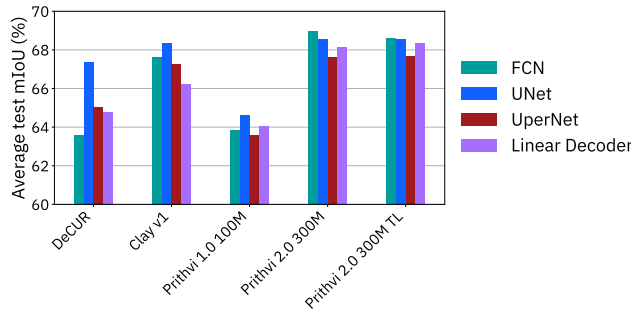


Fig. 5: Average test mIoU (%) \uparrow over five datasets for each model and decoder combination with five runs each.

Surprisingly, with only a single layer, the linear decoder performs competitively and even outperforms UperNet for Prithvi models. However, this result is primarily influenced by a 5pp drop in UperNet’s performance on the Cashew Plantation dataset. Combined with our findings from frozen encoder experiments, it suggests that GeoFMs can effectively learn non-linear functions within the encoder, reducing reliance on complex decoders for performance comparisons. However, the lower capacity of the linear decoder cannot capture small features, as shown in Figure 6, and is unsuitable for applications.

While FCN achieves a high performance, it has a major drawback: Visual analysis of predictions reveals that FCN (and the linear decoder) produces patchy outputs, whereas UNet and UperNet generate more spatially consistent results (see Figure 6). This is likely because UNet and UperNet access multi-scale feature maps, allowing them to preserve low-level spatial information, which helps with fine-grained segmentation. The visualized images are selected to show failures and do not fully represent the overall prediction quality. The UNet decoder leads to the best predictions, considering performance and visual quality.

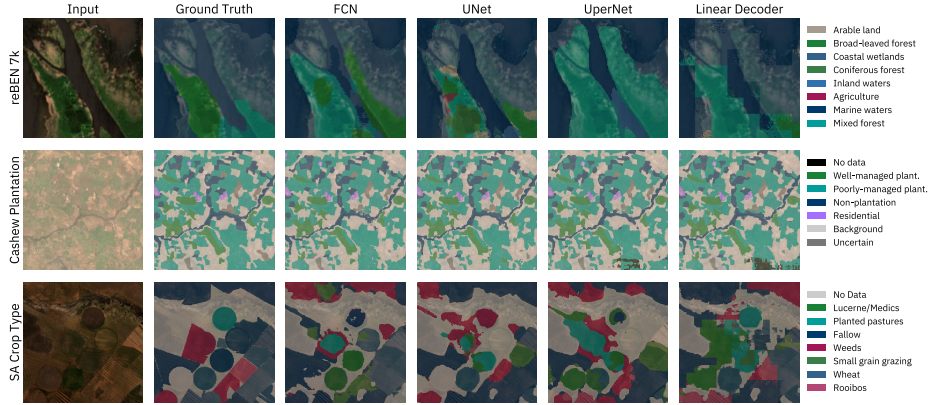


Fig. 6: Predictions of Prithvi 2.0 300M with different decoders and unfrozen backbone. The examples are selected to showcase patchy predictions from FCN and the linear decoder, while UperNet and UNet results are smoother.

6 Conclusion

We have extensively evaluated PEFT techniques for geospatial foundation models, analyzing different fine-tuning strategies, geographic generalization, and decoder architectures. Our results show that LoRA matches or exceeds full fine-tuning performance while significantly reducing memory and potentially training time, making it the most effective PEFT method. The decoder comparisons highlight UNet as the best-performing architecture, while metadata has minimal impact on fine-tuning. Geographic generalization remains an open challenge, though GeoFMs outperform standard baselines, especially when fine-tuned with LoRA. These findings suggest LoRA as a suitable fine-tuning strategy for EO applications, particularly in resource-constrained settings with large datasets. However, our study is limited to three models, and PEFT effectiveness may vary with architecture and data. Future work should explore more models and tasks, and assess additional PEFT variants like TEA [18] or DEFLECT [32] to identify robust strategies. All models and techniques are available in TerraTorch, enabling scalable adaptation for real-world use.

References

1. Bafghi, R.A., Bagwell, C., Ravichandran, A., Shrivastava, A., Raissi, M.: Fine tuning without catastrophic forgetting via selective low rank adaptation. arXiv preprint arXiv:2501.15377 (2025)
2. Bafghi, R.A., Harilal, N., Monteleoni, C., Raissi, M.: Parameter efficient fine-tuning of self-supervised vits without catastrophic forgetting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 3679–3684 (2024)
3. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the Opportunities and Risks of Foundation Models. arXiv preprint arXiv:2108.07258 (2021)
4. Bonafilia, D., Tellman, B., Anderson, T., Issenberg, E.: Sen1Floods11: A Georeferenced Dataset to Train and Test Deep Learning Flood Algorithms for Sentinel-1. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020)
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language Models are Few-shot learners. *Advances in Neural Information Processing Systems* **33** (2020)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-supervised Vision Transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
7. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision Transformer Adapter for Dense Predictions. In: The Eleventh International Conference on Learning Representations (2023)
8. Clasen, K.N., Hackel, L., Burgert, T., Sumbul, G., Demir, B., Markl, V.: reBEN: Refined BigEarthNet Dataset for Remote Sensing Image Analysis. arXiv preprint arXiv:2407.03653 (2024)
9. Clay Foundation: Clay Foundation Model (2024), <https://huggingface.co/made-with-clay/Clay>
10. Dong, Z., Gu, Y., Liu, T.: UPetu: A Unified Parameter-efficient Fine-tuning Framework for Remote Sensing Foundation Model. *IEEE Transactions on Geoscience and Remote Sensing* (2024)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2020)
12. ESA: Fusion Competition (2023), <https://source.coop/repositories/esa/fusion-competition/description>
13. Gomes, C., Blumenstiel, B., Almeida, J.L.d.S., de Oliveira, P.H., Fraccaro, P., Marti-Escobet, F., Szwarcman, D., Simumba, N., Kienzler, R., Zadrozny, B.: TerraTorch: The Geospatial Foundation Models Toolkit. arXiv preprint arXiv:2503.20563 (2025)
14. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2. IEEE (2006)
15. Han, Z., Gao, C., Liu, J., Zhang, J., Zhang, S.Q.: Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *Transactions on Machine Learning Research* (2024)

16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders are Scalable Vision Learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: LoRA: Low-rank Adaptation of Large Language Models. Proceedings of the International Conference on Learning Representations 1(2) (2022)
18. Hu, L., Lu, W., Yu, H., Yin, D., Sun, X., Fu, K.: TEA: A Training-efficient Adapting Framework for Tuning Foundation Models in Remote Sensing. IEEE Transactions on Geoscience and Remote Sensing (2024)
19. Hu, L., Yu, H., Lu, W., Yin, D., Sun, X., Fu, K.: AiRs: Adapter in Remote Sensing for Parameter-efficient Transfer Learning. IEEE Transactions on Geoscience and Remote Sensing 62 (2024)
20. IBM: TerraTorch Iterate (2025), <https://github.com/IBM/terratorch-iterate>
21. Jakubik, J., Roy, S., Phillips, C., Fraccaro, P., Godwin, D., Zadrozny, B., Szwarcman, D., Gomes, C., Nyirjesy, G., Edwards, B., et al.: Foundation Models for Generalist Geospatial Artificial Intelligence. arXiv preprint arXiv:2310.18660 (2023)
22. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual Prompt Tuning. In: Proceedings of the European Conference on Computer Vision. Springer (2022)
23. Jin, Z., Lin, C., Weigl, C., Obarowski, J., Hale, D.: Smallholder Cashew Plantations in Benin. Radiant MKHub (2021)
24. Lacoste, A., Lehmann, N., Rodriguez, P., Sherwin, E., Kerner, H., Lütjens, B., Irvin, J., Dao, D., Alemohammad, H., Drouin, A., et al.: GEO-Bench: Toward Foundation Models for Earth Monitoring. Advances in Neural Information Processing Systems 36 (2023)
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
26. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
27. Mommert, M., Kesseli, N., Hanna, J., Scheibenreif, L., Borth, D., Demir, B.: BEN-GE: Extending BigEarthNet with Geographical and Environmental Data. In: IEEE International Geoscience and Remote Sensing Symposium. IEEE (2023)
28. Phillips, C., Roy, S., Ankur, K., Ramachandran, R.: HLS Foundation Burn-scars Dataset. https://huggingface.co/datasets/ibm-nasa-geospatial/hls_burn_scars (2023)
29. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-assisted Intervention–MICCAI. Springer (2015)
30. Scheibenreif, L., Mommert, M., Borth, D.: Parameter Efficient Self-supervised Geospatial Domain Adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
31. Szwarcman, D., Roy, S., Fraccaro, P., Gíslason, P.E., Blumenstiel, B., Ghosal, R., de Oliveira, P.H., Almeida, J.L.d.S., Sedona, R., Kang, Y., et al.: Prithvi-EO-2.0: A Versatile Multi-Temporal Foundation Model for Earth Observation Applications. arXiv preprint arXiv:2412.02732 (2024)
32. Thoreau, R., Marsocci, V., Derksen, D.: Parameter-Efficient Adaptation of Geospatial Foundation Models through Embedding Deflection. arXiv preprint arXiv:2503.09493 (2025)

33. Wang, Y., Albrecht, C.M., Braham, N.A.A., Liu, C., Xiong, Z., Zhu, X.X.: Decoupling Common and Unique Representations for Multimodal Self-Supervised Learning. In: Proceedings of the European Conference on Computer Vision (2024)
34. Wang, Y., Braham, N.A.A., Xiong, Z., Liu, C., Albrecht, C.M., Zhu, X.X.: SSL4EO-S12: A Large-Scale Multi-Modal, Multi-Temporal Dataset for Self-supervised Learning in Earth Observation. *IEEE Geoscience and Remote Sensing Magazine* **11**(3) (2023)
35. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified Perceptual Parsing for Scene Understanding. In: Proceedings of the European Conference on Computer Vision (2018)
36. Xiong, Z., Wang, Y., Zhang, F., Stewart, A.J., Hanna, J., Borth, D., Papoutsis, I., Le Saux, B., Camps-Valls, G., Zhu, X.X.: Neural Plasticity-inspired Foundation Model for Observing the Earth Crossing Modalities. *arXiv preprint arXiv:2403.15356* (2024)
37. Zahweh, M.H., Nasrallah, H., Shukor, M., Faour, G., Ghandour, A.J.: Empirical Study of PEFT Techniques for Winter-Wheat Segmentation. *Environmental Sciences Proceedings* **29**(1) (2023)
38. Zaken, E.B., Goldberg, Y., Ravfogel, S.: BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (2022)
39. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid Scene Parsing Network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
40. Zhu, X.X., Tuia, D., Mou, L., Xia, G.S., Zhang, L., Xu, F., Fraundorfer, F.: Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine* **5**(4) (2017)