GEO-Bench: Toward Foundation Models for Earth Monitoring

Alexandre Lacoste[*1] **Nils Lehmann**[*2] Pau Rodriguez[1] Evan David Sherwin[3]
Hannah Kerner[4] Björn Lütjens[5] Jeremy Irvin[3] David Dao[6]
Hamed Alemohammad[7] Alexandre Drouin[1,8] Mehmet Gunturkun[1] Gabriel Huang[1,9]
David Vazquez[1] Dava Newman[5] Yoshua Bengio[8,9] Stefano Ermon[3]
Xiao Xiang Zhu[2]

[1] ServiceNow Research [2] Technical University of Munich [3] Stanford University
[4] Arizona State University [5] MIT [6] ETH Zurich [7] Clark University
[8] Mila-Quebec [9] University of Montreal

Abstract

Recent progress in self-supervision has shown that pre-training large neural networks on vast amounts of unsupervised data can lead to substantial increases in generalization to downstream tasks. Such models, recently coined *foundation models*, have been transformational to the field of natural language processing. Variants have also been proposed for image data, but their applicability to remote sensing tasks is limited. To stimulate the development of foundation models for Earth monitoring, we propose a benchmark comprised of six classification and six segmentation tasks, which were carefully curated and adapted to be both relevant to the field and well-suited for model evaluation. We accompany this benchmark with a robust methodology for evaluating models and reporting aggregated results to enable a reliable assessment of progress. Finally, we report results for 20 baselines to gain information about the performance of existing models. We believe that this benchmark will be a driver of progress across a variety of Earth monitoring tasks.

1 Introduction

Earth monitoring with machine learning-based methods plays an increasing role in climate change mitigation and adaptation as well as climate science [57]. Related applications include methane source detection [61, 16], forest carbon quantification [44], extreme weather prediction [49], and crop monitoring [34, 14]. Across many of these applications, pre-trained models (e.g., a ResNet trained on ImageNet) have been used to increase generalisation performance. Improvement of the pre-trained models has been shown to reduce the need for large labelled datasets in some contexts [11] and can improve model generalisation outside of the training distribution [28]. Recent studies exploring the scaling of such pre-trained models found that increasing the size of an unsupervised (or weakly supervised) dataset as well as properly scaling the model led to an even greater increase in performance under various metrics [33, 55].

While the training of such large-scale models is usually reserved for industrial research groups with very large computer clusters, the publication of pre-trained models creates vast opportunities for the entire research and technology community (including communities of domain experts outside of machine learning). These large pre-trained models were recently coined as *foundation models* [6] as they might serve as foundations for sub-fields of machine learning. Specifically, the publication of large pre-trained models like BERT [15] and GPT-3 [7] led to a paradigm shift in the field of natural language processing (NLP). This inspired a similar shift in the field of computer vision with the release of models like CLIP [55] and DINO [9]. While CLIP performs well on various types of vision tasks, it still under-performs

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

on Earth monitoring tasks [55]. This is not surprising as it is trained mainly on RGB images taken from a ground perspective at a single point in time.

While there are many similarities between Earth observation datasets and typical ML image datasets, there are also many important differences to consider when designing effective ML models. Earth observation images are taken from an overhead rather than ground perspective, usually from a fixed distance from the Earth's surface (defined by a satellite's orbit). The satellite revisits provide a temporal axis that is sometimes irregular (e.g., a few times per year) or regular (e.g., every five days) with cloud coverage causing spurious occlusions. Images are acquired with sensors containing multiple spectral bands (e.g., thirteen for Sentinel-2), or even with different kinds of sensors, e.g., synthetic aperture radar (SAR), which can penetrate cloud coverage. Moreover, the GPS coordinates and timestamp of each acquisition offer the opportunity to combine data from multiple sources, e.g., weather data, semantic maps, and elevation. This leads to a rich multi-modal signal with potentially missing information that can be inferred from other elements of the signal. There are currently petabytes of accessible satellite datasets containing images of the Earth under various modalities from the present day to as far back as the 1960s. Distilling this large amount of information into pre-trained models of various sizes offers the opportunity to redistribute this information and make it accessible to various labs for increasing the performances on a large range of downstream tasks.

The fundamental goal of these large pre-trained models is to improve generalization performance on downstream tasks. Hence, to support the machine learning community in producing better pre-trained models, it is crucial to provide a benchmark with a wide variety of downstream tasks, covering a range of modalities and dataset shapes that are likely to be encountered in practice. At the moment, existing works on pre-training models from earth observations e.g., [13, 46, 69], evaluate on different sets of downstream tasks, making it impossible to directly compare performance. Moreover, the set of tasks is often narrow in terms of diversity and the statistical methodologies do not adequately report the uncertainties in the evaluation.

The present work aims to fill this void by providing a wide range of tasks across various countries with various modalities of sensors. Also, the transformed versions of the datasets are smaller than their original form, and all results can be replicated on single GPUs. This increases accessibility to research labs with limited resources and reduces overall energy consumption. Our proposed benchmark, GEO-Bench[1], is composed of six image classification and six semantic segmentation tasks, which were curated by domain experts to ensure their diversity and relevance toward sustainable development. We expect this contribution to:

- Stimulate and facilitate the development of foundation models for Earth monitoring
- Provide a systematic way of measuring the quality of models for better scientific progress
- Provide insights into which pre-trained models work best
- Potentially reduces negative impacts of foundation models through an open evaluation procedure.

In what follows, we start by discussing sources of data that can serve to train foundation models for earth monitoring (Sec. 2). We then present the details of GEO-Bench (Sec. 3) and how it can be used for the evaluation of foundation models (Sec. 4). Further, we review existing benchmark datasets for earth monitoring and discuss why GEO-Bench is complementary (Sec. 5). Finally, we present an extensive set of experiments, showing the performance of 20 state-of-the-art models on the benchmark to lay down reference points and to gain valuable information on existing pre-trained models (Sec. 6).

2 Remote sensing data for self-supervision

The development of foundation models does not typically rely on a specific dataset for the pre-training phase. The choice of data source is part of the design of the model, e.g., a very large corpus of text from the internet [50] or pairs of text associated with images from the web [55]. As such, we do not provide data for training foundation models with this benchmark. However, for completeness, we outline potential sources of Earth observation data that could be used for pre-training foundation models.

Multispectral images with revisits Satellite data sources such as Sentinel-2 [20, 23] and Landsat 8 [66] provide images in multiple spectral bands with periodic revisits. This yields a four-dimensional array of structured data (longitude, latitude, wavelength, time) which can be used to perform various forms of self-supervision, e.g., predicting adjacent tiles [30] or contrasting the different seasons for the same region [46].

[1] https://zenodo.org/communities/geo-bench

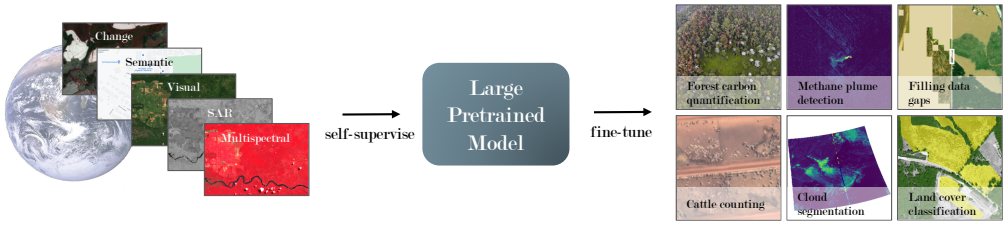

Figure 1: Foundation models encapsulate multimodal data streams through self-supervised training. The trained models can then be fine-tuned for a variety of climate-related remote sensing tasks. Image sources: quantification [44], detection [32], generation [43], counting [36], segmentation [75], and multi-class classification [51].

Other sensors Synthetic Aperture Radar (SAR) and terrain elevation are also frequently available and can be matched to other sources of data through geolocalisation [54]. Such data are complementary to optical spectral bands and may encourage the model to learn higher-level semantic representations.

Semantic data Through georeferencing, text-based data such as Wikipedia articles can be linked to satellite images [67]. It is also possible to join content from non-image data layers like OpenStreetMap [39]. By predicting or contrasting information from these sources, the model may learn useful and transferable semantic representations.

3 GEO-Bench

GEO-Bench is composed of 6 classification tasks and 6 segmentation tasks. Detailed characteristics are presented in Table 1, examples are depicted in Figure 2 and 3, and the spatial coverage on the world map is presented in Figure 8 (supplementary material). In what follows, we describe the procedure for collecting and transforming the datasets.

3.1 Design Principles

GEO-Bench was established by modifying and gathering geospatial datasets, adhering to principles that secure accessibility, usability, and effective model performance assessment across tasks.

Ease of Use A fundamental goal was to create an accessible, simple-to-use benchmark, and a compact dataset assortment with code for loading the data in a consistent schema. A key aim was to harmonize data to reduce the engineering work needed to tailor pre-trained architectures, while maintaining sensor type and resolution diversity.

Sector Experts and Steering Committee To align GEO-Bench with practical use-cases, we assembled a team of six sector experts from fields such as forestry and climate science. A steering committee of respected scientists guides high-level benchmark decisions, assuring relevance and impact.

Diversity of Modalities The objective is to evaluate model adaptability to varied geospatial sensors. Thus, the benchmark encompasses multispectral, SAR, hyperspectral, elevation, and cloud probability modalities, with spatial resolutions from 0.1 to 30 m/pixel.

Diversity of Tasks We ventured beyond image classification, incorporating object detection and semantic segmentation. To maintain *ease of use*, detection and counting tasks were transformed into semantic segmentation. This led to two task sets: six image classification tasks, and six semantic segmentation tasks [25, 38].

Original Train, Validation, and Test Splits Original dataset splits were preserved when available; otherwise, we generated validation and test sets from the train set while ensuring no spatial overlap.

Permissive License Most datasets needed to be adapted from their original form to satisfy the above criteria and be included in the benchmark. Hence, we include only datasets with permissive licenses.

Classification

Name	Image Size	# Classes	Train	Val	Test	# Bands	RGB res	Sensors	Cite	License
m-bigearthnet	120 x 120	43	20000	1000	1000	12	10.0	Sentinel-2	[64]	CDLA-P-1.0
m-so2sat	32 x 32	17	19992	986	986	18	10.0	Sentinel-2 + Sentinel-1	[76]	CC-BY-4.0
m-brick-kiln	64 x 64	2	15063	999	999	13	10.0	Sentinel-2	[37]	CC-BY-SA 4.0
m-forestnet	332 x 332	12	6464	989	993	6	15.0	Landsat-8	[29]	CC-BY-4.0
m-eurosat	64 x 64	10	2000	1000	1000	13	10.0	Sentinel-2	[27]	MIT
m-pv4ger	320 x 320	2	11814	999	999	3	0.1	RGB	[48]	MIT

Segmentation

Name	Image Size	# Classes	Train	Val	Test	# Bands	RGB res	Sensors	Cite	License
m-pv4ger-seg	320 x 320	2	3000	403	403	3	0.1	RGB	[48]	MIT
m-chesapeake-landcover	256 x 256	7	3000	1000	1000	4	1.0	RGBN	[56]	CDLA-P-1.0
m-cashew-plantation	256 x 256	7	1350	400	50	13	10.0	Sentinel-2	[74]	CC-BY-4.0
m-SA-crop-type	256 x 256	10	3000	1000	1000	13	10.0	Sentinel-2	link	CC-BY-4.0
m-nz-cattle	500 x 500	2	524	66	65	3	0.1	RGB	[1]	CC-BY-4.0
m-NeonTree	400 x 400	2	270	94	93	5	0.1	RGB + Hyperspectral + Elevation	[71]	CC0 1.0

Table 1: **GEO-Bench:** Characteristics of datasets in the benchmark. Since datasets are *modified*, we prepend their name with "m-" to distinguish them from the original dataset.

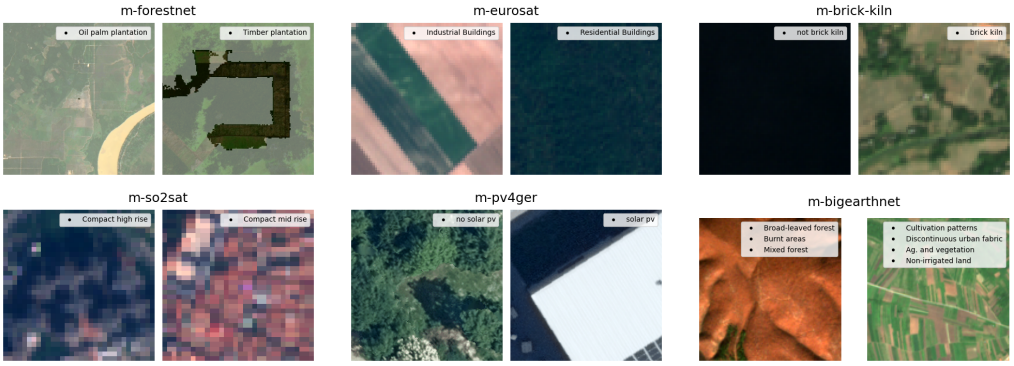

Figure 2: Representative samples of the **classification benchmark**.

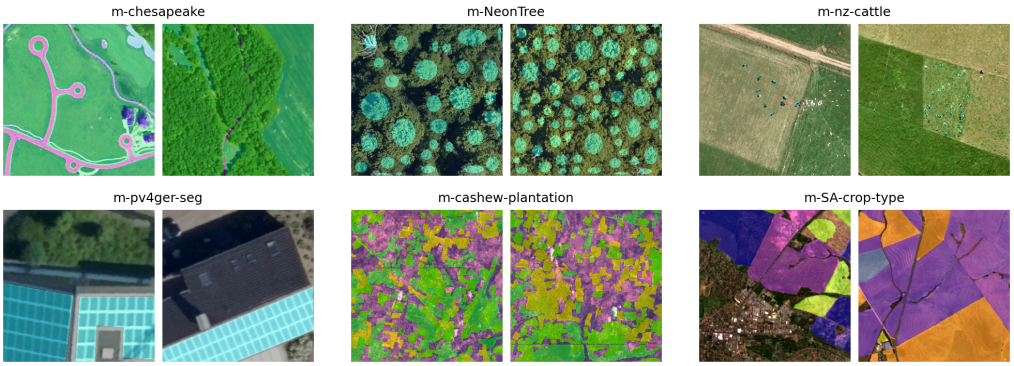

Figure 3: Representative samples of the **segmentation benchmark**.

3.2 Dataset Transformations

To produce a benchmark that complies with the design choices of Section 3.1, we applied the following transformations to each dataset. The procedure that was used to download and transform each dataset is fully documented and open-sourced in the GEO-Bench GitHub repository[2].

[2] https://github.com/ServiceNow/geo-bench

Subsampling Large Datasets To be more representative of typical downstream tasks, where data is usually scarce, datasets larger than 20000 samples were randomly subsampled. Avoiding large downstream tasks also comes with other benefits:

- In Appendix A, we show that larger downstream datasets can decrease the ability to discriminate between two models that are similar in performance.
- Downstream tasks with very large training sets will not usually benefit from pre-training[3]. Hence they are less useful for our evaluation purpose.
- A smaller benchmark is faster to download, yields results quicker and requires less energy for computation.
- We can increase the variety of experiments and the number of seeds to improve the knowledge gained from experiments.

Removing Class Imbalance We randomly subsampled large classes to have near-uniform class sizes across datasets. This was done to prevent users of the benchmark from increasing their score by using clever class imbalance techniques instead of making progress on better pre-trained models. While good performance on highly imbalanced (long tail of classes) datasets would be a desired property of a pre-trained model, we have not found a good dataset containing a large number of classes.

4 Using The Benchmark

Fine Tuning In the self-supervised learning literature, it is common to use the pre-trained model to encode a fixed representation of each image in the dataset and learn to classify images based on this representation [30]. While this works relatively well, this method highly depends on the pre-training task as it may not learn to encode information that is important for the downstream task [65, 53]. In practice, fine-tuning the pre-trained model often mitigates this issue and is known to frequently yield a much higher generalization performance than a model trained from random weights [46, 11]. Since this is more representative of practical usage, we encourage users of the benchmark to report the results of fine-tuned models. On the other hand, we do not discourage users from also reporting results with fixed backbones (pre-trained weights) as this can provide valuable information about the pre-trained model. In all cases, we ask users to report their fine-tuning methodology with enough details for reproducibility.

Hyperparameter Tuning Deep learning algorithms often require the adjustment of hyperparameters, especially when an architecture is fine-tuned on a small dataset. For this reason, we recommend adjusting hyperparameters, but within a maximum budget of 16 trials per task[4]. Early stopping based on validation metrics is also recommended.

Data Augmentation Data augmentation plays a crucial role in the training of deep learning models, especially with small training datasets. Hence, we consider it to be part of the fine-tuning process. As a guideline, we propose limiting the augmentations to 90° rotations and vertical and horizontal flips[5]. On the other hand, we also encourage users to study what are the best data augmentations for remote sensing as this could lead to useful findings for practitioners and the benchmark is well-suited for evaluating such findings.

Toolbox To facilitate the usage of the benchmark, we provide a collection of tools for various parts of the experimental pipeline as part of the open-sourced codebase[6]. This includes tools for loading datasets and visualising results. We also provide tools based on PyTorch-Lightning [24] to facilitate model training.

4.1 Reporting Results

For reliable and comparable results across different publications, we recommend that users follow this procedure to report results. The aim is to report results on individual tasks as well as aggregated across all tasks, with reliable confidence intervals (inspired by [2]). Code is provided to generate figures based on raw results.

[3] From Bayes rule, we know that the influence of the prior (pre-trained model) decreases as the size of the training data increases.

[4] While 16 is fairly small, we believe it's enough to adjust sensitive hyperparameters such as learning rate. Also, this favours models that are less sensitive to hyperparameter tuning.

[5] Random crop and resize are also common in vision, but in remote sensing, this reduces the spatial resolution, which is often crucial for high performances.

[6] https://github.com/ServiceNow/geo-bench

Random Seeds As demonstrated in [2], 3-5 seeds are not enough to obtain reliable confidence intervals. Since pre-training and hyperparameter search are usually the computational bottlenecks, we recommend retraining the selected hyperparameter configuration for at least 10 different seeds.

Interquartile Mean (IQM) We recommend using IQM. This metric removes the outliers by trimming the 25% highest values as well as the 25% lowest value and computing the average of the remaining values. The resulting finite sample estimator is less biased than the median and has less variance than the mean, often resulting in smaller confidence intervals [2].

Normalising Results To aggregate performance metrics across multiple tasks, one must first normalise their values. A common approach consists of applying a linear transformation based on reference points [4]. As such, we propose to use the lowest and highest metric values achieved by a set of strong baselines (see Sec. 6) as *official reference points*. For each individual task, we scale the results such that the maximum score is 1 and the lowest one is 0. Hence, if a future model were to achieve a score superior to 1, it would imply that progress is being made on the benchmark. All reference points will be published alongside the benchmark.

Bootstrapping To quantify uncertainty over observed IQMs, we use bootstrapping [21]. That is, we sample n times, with replacement, the results from training with n different seeds, and we compute IQM. Repeating this procedure $n = 1000$ times provides a distribution over IQM results, from which confidence intervals can be extracted.

Aggregated Results After normalizing the results we simply compute IQM across all datasets and all results of a given model. For confidence intervals, we use *stratified bootstrap*, where seeds are sampled with replacement *individually* for each dataset, but IQM is computed across all datasets.

Displaying the results In Figure 4, we show how to compactly display results from a wide range of baselines across the benchmark as well as aggregated results and statistical uncertainties. In Figure 5, we display the results for a growing training set size (with fixed validation and test set). This compactly reports the results of thousands of experiments.

Publishing the results We ask experimenters to publish the results of all seeds on all datasets for all models as a CSV file along with the open-sourced code of their experiments. This will allow future authors to incorporate existing results in their comparison figures.

5 Related Works

SustainBench consists of 15 public datasets covering 7 sustainable development goals [73]. Seven of these datasets are two-dimensional remote sensing. It includes a public leaderboard for tracking model performance. A featured task is the Brick Kiln classification, selected for its georeferenced, high-quality ground truth labels. SustainBench's purpose is monitoring progress in specified tasks, thus comprising a diverse set of datasets. It doesn't aim for solution under a single framework or aggregate result tracking.

TorchGeo is a Python library designed to streamline the integration of remote sensing datasets into the PyTorch deep learning ecosystem [62]. TorchGeo currently features data loaders for 52 publicly available datasets of satellite, aerial, and drone imagery for classification, regression, change detection, semantic segmentation, instance segmentation, and object detection tasks. Our benchmark directly interfaces with TorchGeo and uses its data loaders for several datasets included in the benchmark.

EarthNets is a concurrently developed platform to evaluate deep learning methods on remote sensing datasets [72]. In their methodology, they analyse the metadata of 400 publicly available remote sensing datasets. Using meta-information such as the number of samples, the size of each sample, and the type of annotations, they analyse the correlation between each dataset and identify a variety of clusters. Based on this analysis, they recommend two classification, two segmentation, and two detection datasets for benchmarking. In contrast, we provide a collection of 12 datasets and we propose a robust methodology for aggregating results and reporting statistical uncertainties of the evaluation process.

AiTLAS recently proposed a benchmark of 22 classification datasets[17], 3 of which intersect with our classification benchmark. They proposed a standardised version of train, valid, test splits for existing datasets as well as a fine-tuning procedure. By leveraging the overlap of labels across datasets, they also

provide a more accurate test metric for real-world applications. Experiments are conducted using 10 different model families across the 22 datasets, using RGB images as input.

6 Experiments

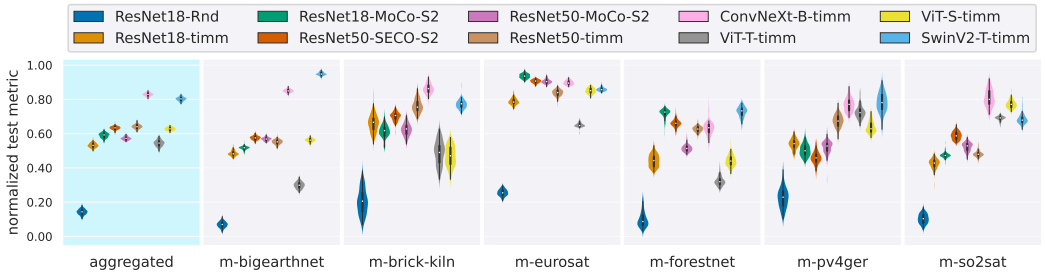

Figure 4: **Classification Benchmark RGB Only:** Normalised accuracies of various baselines (higher is better). Violin plots are obtained from bootstrap samples of normalized IQM (Section 4.1). The left plot reports the average across all tasks.

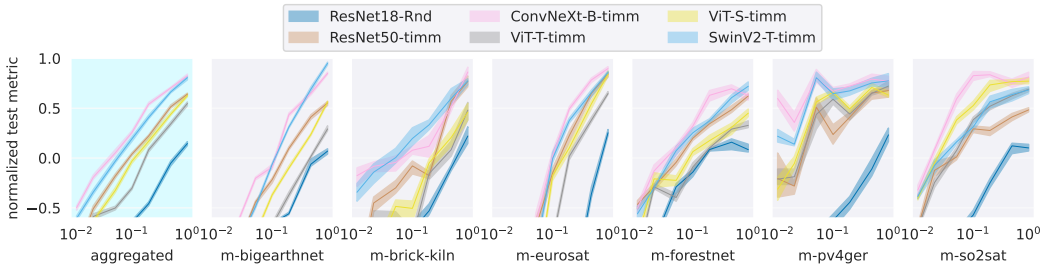

Figure 5: **Classification vs Train Size:** Normalised accuracies of a subset of the baselines on Classification benchmark with a growing size of the training set. The shaded region represents an 80% confidence interval, obtained from bootstrap samples of normalized IQM (Section 4.1). The left plot reports the average across all tasks.

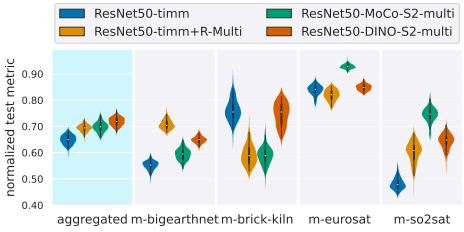

Figure 6: **Effect of Multispectral with ResNet50**. Only Sentinel-2 tasks are reported. Normalised accuracies (Sec 4.1).

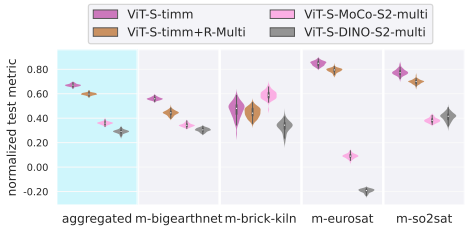

Figure 7: **Effect of Multispectral with ViT-S**. Only Sentinel-2 tasks are reported. Normalised accuracies (Sec 4.1).

In this section, we provide a range of baselines for the classification and segmentation benchmarks. These will serve as reference points for future evaluation[7]. We also seek to answer the following questions:

- Which new architecture performs best on remote sensing data (Section 6.2.2)?
- What is the effect of training set size on the performance of each model (Section 6.2.3)?
- Can we leverage multispectral channels to improve performance (Section 6.2.4)?
- Are smaller datasets better at discriminating the performance of different models (Section A.5)?

[7]We recall that all datasets have been modified from their original version. Hence, our results are not directly comparable to other published results.

6.1 Protocol

For each model, we replaced the last layer with a randomly initialised layer of the appropriate shape for the task at hand. We use different learning rates for the last layer (which starts from random weights) and for the backbone (which starts from pre-trained weights). The best learning rates were selected using the highest accuracy or Intersection over Union (IoU) on the validation set over 16 trials [8]. After choosing the hyperparameters, we repeated the training for 10 seeds. To minimize overfitting, we selected the best time step using accuracy (or IoU) on the validation set and we reported the test metrics at the chosen time step. We use AdamW [42] to train convolution architectures and SGD to train transformer architectures.

6.2 Classification

6.2.1 Baselines Naming Schema

Each baseline name starts with the corresponding architecture: **ResNet18 and ResNet50:** standard ResNet architectures [26]; **ConvNeXt-B:** the base architecture of ConvNeXt [41]; **ViT-T and ViT-S:** ViT architectures [19] of size tiny and small respectively; **SwinV2-T:** a SwinV2-tiny architecture [40];

Then, keywords provide details about the training procedure: **SeCo:** a ResNet50 model trained on Sentinel 2 data with temporal contrastive loss across seasons [46]; **MoCo-S2 and DINO-S2:** model trained with self-supervision on Sentinel data [70] (RGB and Multispectral pre-trained weights); **Rnd:** weights are randomly initialised; **timm:** pre-trained weights are obtained from the timm library, usually from training on ImageNet; **+R-Multi:** we manually augment an RGB architecture by randomly initialising the weights of the missing channels in the 1st layer; **multi:** the pre-trained model has multispectral channels.

6.2.2 Comparing Baselines on RGB only

In Figure 4, we report bootstrapped IQM of the normalized accuracy (Sec 4.1) for the six datasets of the classification benchmark, as well as aggregated results[9]. In this first experiment, all models can only see the RGB channels.

These results offer valuable information across 10 common baselines in the literature. We denote the outstanding performance of ConvNext and SwinV2 compared to other models. It is by a large margin the best models in aggregated results and almost systematically outperforms all models on all datasets. We can also observe the large difference between Scratch ResNet18 and ResNet18 on all datasets. This highlights the importance of using a pre-trained model. Also, perhaps disappointingly, the existing model pre-trained on remote sensing data does not exhibit any improvement compared to their timm pre-trained weights, i.e., ResNet18-MoCo-S2, ResNet50-MoCo-S2, and ResNet50-SeCo-S2 are all comparable to ResNet18 on the aggregated performance. On the other hand, in Section 6.2.4, we see that ResNet50-MoCo-S2-multi can leverage multispectral data to slightly surpass ResNet50-timm.

Another insight that can be gained from these results is how useful a dataset is at discriminating baselines, i.e., a dataset where most baselines perform equally would have limited utility in our benchmark. To this end, we had to discard GeoLifeClef 2022 [12] as all models were performing equally badly[10]. m-eurosat also offers limited discriminativity as most models obtain very high accuracy (see Figure 9). To make this dataset harder, we subsample down to 2000 training samples. We can now see that smaller models tend to perform better on this dataset, but the discriminativity remains fairly low.

6.2.3 Accuracy vs training set size

As part of the benchmark, we also provide official subsets of the training sets with train ratios of (0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1) [11].

[8] The range of selected learning rates is different for each model and is selected based on early experiments, see appendix for details.

[9] We note that the variance of the results represents the uncertainty of the mean (IQM) which is significantly smaller than the variance of the raw seeds presented in Figure 9 in Appendix.

[10] We suspect this dataset to have high aleatoric uncertainties.

[11] Reporting results on all 7 subsets increases the number of experiments by 7x. However, in Figure 13 (see Appendix), we show that the convergence time is proportional to the training set size. This means that training on all seven subsets takes on average about 1.88 times longer than just training on the full training set.

Figure 5 depicts a different perspective on the models. First, we can observe the noise due to the hyperparameter selection process that is not accounted for by repeating 10 seeds with fixed hyperparameters. Also, we see that ConvNeXt often becomes better than SwinV2 as the training set decreases. This coincides with the common observations that transformer architectures tend to be more data-hungry, but also tend to outperform convolution architectures in the high data regime [18]. We note also, that ConvNeXt-B-timm only requires 2% of the training set to obtain aggregated performances comparable to that of ResNet18-Rnd. This impressive factor of 50x on data efficiency highlights the importance of developing new architectures and new pre-training methods. Finally, we can observe an increase in the discriminativity of the datasets as the training set decreases, specifically for m-eurosat, when the task becomes more difficult, the strong baselines stand out even more. The discriminativity of datasets is further studied in Section A.5.

6.2.4 Leveraging Multispectral Information

We now study the effect of leveraging multispectral information during the pre-training phase and during the fine-tuning phase. We do so by fixing the backbone to either ResNet50 (Fig. 6) or ViT-S (Fig. 7) and exploring various weight initialisation schema. Since we could only find pre-trained models for Sentinel-2, we limit this experiment to the four datasets satisfying this criterion.

We found that using a model pre-trained on RGB-only (timm pre-trained) and augmenting the architecture by randomly initialising the weights of the missing channels in the first layer (+RMulti) does not lead to systematic improvement. Moreover, the fine-tuning time is largely extended since we have to wait until the newly initialised weights on the first layer fully converge. On the other hand, the ResNet50 pre-trained on Sentinel-2 using DINO or MoCo [70] leads to a modest performance increase on average. When looking at ViT-S (Fig. 7), incorporating multi-spectral only leads to a systematic performance decrease.

6.3 Segmentation

We defer experiments on the Segmentation benchmark to Appendix A.3, where we provide experiments on six baselines (ResNet18, ResNet50, ResNet101) $\times$ (U-Net, DeepLabV3) with pre-trained weights provided by the timm library. While ResNet101-DeepLabV3 performs best in aggregate, it still underperforms on some datasets.

6.4 Resource Usage

See Appendix A.6 for detailed resource usage of each algorithm evaluated in this section. We report the number of parameters, memory usage, the time required for a forward pass, and the convergence time for fine-tuning on downstream tasks. While memory footprint can increase by a factor of 4x for a model like SwinV2 and ConvNeXt-B compared to ResNet50, their forward pass is only twice as slow.

7 Conclusion

We developed a new benchmark for evaluating pre-trained models on remote sensing downstream tasks. This involves adapting a variety of remote sensing datasets to a more conventional machine learning pipeline and providing code for fine-tuning and evaluating individual tasks. We expect that this benchmark will stimulate the development of new foundation models that could lead to better generalization on a variety of earth monitoring downstream tasks and could open up opportunities for new applications.

Limitations Our benchmark does not extensively evaluate all desired features of a pre-trained model for earth monitoring. For example, it does not evaluate its ability to fine-tune temporal data nor perform fusion with other types of data such as text or weather. The spatial coverage of the benchmark covers most continents and improves coverage over individual datasets. However, the spatial coverage could still be largely improved to include a much wider range of countries and biomes. Finally, as pre-trained models become stronger, they will get closer to the theoretical limit of generalization performance, i.e. approaching the aleatoric uncertainty of the dataset. Under such a regime, we expect a bigger overlap between error bars when comparing 2 different models.

References

[1] Diab Abuaiadah and Alexander Switzer. Remote sensing dataset for detecting cows from high resolution aerial images. 2022.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

[3] Hamed Alemohammad. The case for open-access ML-ready geospatial training data. In *International Geoscience and Remote Sensing Symposium*. IEEE, 2021.

[4] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[5] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

[6] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[8] Marshall Burke, Anne Driscoll, David B Lobell, and Stefano Ermon. Using satellite imagery to understand and promote sustainable development. *Science*, 371(6535), 2021.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[12] Elijah Cole, Benjamin Deneu, Titouan Lorieul, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. The geolifeclef 2020 dataset. *arXiv preprint arXiv:2004.04192*, 2020.

[13] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *arXiv preprint arXiv:2207.08051*, 2022.

[14] Walter T Dado, Jillian M Deines, Rinkal Patel, Sang-Zi Liang, and David B Lobell. High-resolution soybean yield mapping across the us midwest using subfield harvester data. *Remote Sensing*, 12(21):3471, 2020.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Sonu Dileep, Daniel Zimmerle, J Ross Beveridge, and Timothy Vaughn. Automated identification of oil field features using cnns. 2020.

[17] Ivica Dimitrovski, Ivan Kitanovski, Dragi Kocev, and Nikola Simidjievski. Current trends in deep learning for earth observation: An open-source benchmark arena for image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:18–35, 2023.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020. *arXiv preprint arXiv:2010.11929*, 2010.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[20] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012.

[21] B Eforn. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979.

[22] EPA. Greenhouse Gas Emissions: Understanding Global Warming Potentials. Technical report, US Environmental Protection Agency, February 2017.

[23] ESA. Sentinel-2. Technical report, European Space Agency, Paris, France, 2021.

[24] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.

[25] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[28] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

[29] Jeremy Irvin, Hao Sheng, Neel Ramachandran, Sonja Johnson-Yu, Sharon Zhou, Kyle Story, Rose Rustowicz, Cooper Elsworth, Kemen Austin, and Andrew Y Ng. Forestnet: Classifying drivers of deforestation in indonesia using deep learning on satellite imagery. *arXiv preprint arXiv:2011.05479*, 2020.

[30] Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3967–3974, 2019.

[31] Forrest Johnson, Andrew Wlazlo, Ryan Keys, Viren Desai, Erin Wetherley, Ryan Calvert, and Elena Berman. Airborne methane surveys pay for themselves: An economic case study of increased revenue from emissions control. preprint, Environmental Monitoring, July 2021.

[32] Siraput Jongaramrungruang, Christian Frankenberg, Andrew K. Thorpe, and Georgios Matheou. Methanet - an ai-driven approach to quantifying methane point-source emission from high-resolution 2-d plume imagery. *ICML Workshop on Tackling Climate Change with AI*, 2021.

[33] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[34] Hannah Kerner, Gabriel Tseng, Inbal Becker-Reshef, Catherine Nakalembe, Brian Barker, Blake Munshell, Madhava Paliyam, and Mehdi Hosseini. Rapid response crop maps in data sparse regions. *arXiv preprint arXiv:2006.16866*, 2020.

[35] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[36] Issam Laradji, Pau Rodriguez, Freddie Kalaitzis, David Vazquez, Ross Young, Ed Davey, and Alexandre Lacoste. Counting cows: Tracking illegal cattle ranching from high-resolution satellite imagery. *arXiv preprint arXiv:2011.07369*, 2020.

[37] Jihyeon Lee, Nina R. Brooks, Fahim Tajwar, Marshall Burke, Stefano Ermon, David B. Lobell, Debashish Biswas, and Stephen P. Luby. Scalable deep learning to identify brick kilns and aid regulatory capacity. *Proceedings of the National Academy of Sciences*, 118(17), 2021.

[38] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 23, 2010.

[39] Haifeng Li, Xin Dou, Chao Tao, Zhixiang Wu, Jie Chen, Jian Peng, Min Deng, and Ling Zhao. Rsi-cb: A large-scale remote sensing image classification benchmark using crowdsourced data. *Sensors*, 20(6):1594, 2020.

[40] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[43] Björn Lütjens, Brandon Leshchinskiy, Christian Requena-Mesa, Farrukh Chishtie, Natalia Díaz-Rodríguez, Océane Boulais, Aruna Sankaranarayanan, Aaron Pina, Yarin Gal, Chedy Raissi, Alexander Lavin, and Dava Newman. Physically-consistent generative adversarial networks for coastal flood visualization. *ICML Workshop on AI for Modeling Oceans and Climate Change (AIMOCC)*, 2021.

[44] Björn Lütjens, Lucas Liebenwein, and Katharina Kramer. Machine learning-based estimation of forest carbon stocks to increase transparency of forest preservation efforts. *2019 NeurIPS Workshop on Tackling Climate Change with AI (CCAI)*, 2019.

[45] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing*, 152:166–177, 2019.

[46] Oscar Manas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9414–9423, 2021.

[47] M Maskey, H Alemohammad, KJ Murphy, and R Ramachandran. Advancing ai for earth science: A data systems perspective. *Eos*, 101, 2020.

[48] Kevin Mayer, Benjamin Rausch, Marie-Louise Arlt, Gunther Gust, Zhecheng Wang, Dirk Neumann, and Ram Rajagopal. 3d-pv-locator: Large-scale detection of rooftop-mounted photovoltaic systems in 3d. *Applied Energy*, 310:118469, 2022.

[49] Amy McGovern, Kimberly L. Elmore, David John Gagne, Sue Ellen Haupt, Christopher D. Karstens, Ryan Lagerquist, Travis Smith, and John K. Williams. Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, 98(10), 2017.

[50] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. *Communications of the ACM*, 61(5):103–115, April 2018.

[51] Cassandra Pallai and Kathryn Wesson. Chesapeake bay program partnership high-resolution land cover classification accuracy assessment methodology, 2017.

[52] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

[53] Otávio AB Penatti, Keiller Nogueira, and Jefersson A Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 44–51, 2015.

[54] Karissa Pepin, Howard A. Zebker, and William Ellsworth. High-Pass Filters to Reduce the Effects of Broad Atmospheric Contributions in Sbas Inversions: A Case Study in the Delaware Basin. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 1030–1033, Waikoloa, HI, USA, September 2020. IEEE.

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[56] Caleb Robinson, Le Hou, Kolya Malkin, Rachel Soobitsky, Jacob Czawlytko, Bistra Dilkina, and Nebojsa Jojic. Large scale high-resolution land cover mapping with multi-resolution data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12726–12735, 2019.

[57] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[59] Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. 2021.

[60] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

[61] Hao Sheng, Jeremy Irvin, Sasankh Munukutla, Shawn Zhang, Christopher Cross, Kyle Story, Rose Rustowicz, Cooper Elsworth, Zutao Yang, Mark Omara, et al. Ognet: Towards a global oil and gas infrastructure database using deep learning on remotely sensed imagery. *arXiv preprint arXiv:2011.07227*, 2020.

[62] Adam J Stewart, Caleb Robinson, Isaac A Corley, Anthony Ortiz, Juan M Lavista Ferres, and Arindam Banerjee. Torchgeo: deep learning with geospatial data. *arXiv preprint arXiv:2111.08872*, 2021.

[63] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.

[64] Gencer Sumbul, Arne De Wall, Tristan Kreuziger, Filipe Marcelino, Hugo Costa, Pedro Benevides, Mario Caetano, Begüm Demir, and Volker Markl. Bigearthnet-mm: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 9(3):174–180, 2021.

[65] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[66] USGS. Landsat 8. Technical report, United States Geological Survey, Reston, Virginia, USA, 2021.

[67] Burak Uzkent, Evan Sheehan, Chenlin Meng, Zhongyi Tang, Marshall Burke, David Lobell, and Stefano Ermon. Learning to interpret satellite images using wikipedia. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An empirical study of remote sensing pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–1, 2022.

[70] Yi Wang, Nassim Ait Ali Braham, Zhitong Xiong, Chenying Liu, Conrad M Albrecht, and Xiao Xiang Zhu. Ssl4eo-s12: A large-scale multi-modal, multi-temporal dataset for self-supervised learning in earth observation.

[71] Ben G Weinstein, Sarah J Graves, Sergio Marconi, Aditya Singh, Alina Zare, Dylan Stewart, Stephanie A Bohlman, and Ethan P White. A benchmark dataset for canopy crown detection and delineation in co-registered airborne rgb, lidar and hyperspectral imagery from the national ecological observation network. *PLoS computational biology*, 17(7):e1009180, 2021.

[72] Zhitong Xiong, Fahong Zhang, Yi Wang, Yilei Shi, and Xiao Xiang Zhu. Earthnets: Empowering ai in earth observation. *arXiv preprint arXiv:2210.04936*, 2022.

[73] Christopher Yeh, Chenlin Meng, Sherrie Wang, Anne Driscoll, Erik Rozi, Patrick Liu, Jihyeon Lee, Marshall Burke, David B Lobell, and Stefano Ermon. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. *arXiv preprint arXiv:2111.04724*, 2021.

[74] Jin Z., Lin C., Weigl C., Obarowski J., and Hale D. Smallholder cashew plantations in benin, 2021.

[75] Valentina Zantedeschi, Fabrizio Falasca, Alyson Douglas, Richard Strange, Matt J Kusner, and Duncan Watson-Parris. Cumulo: A dataset for learning cloud classes. *arXiv preprint arXiv:1911.04227*, 2019.

[76] Xiao Xiang Zhu, Jingliang Hu, Chunping Qiu, Yilei Shi, Jian Kang, Lichao Mou, Hossein Bagheri, Matthias Häberle, Yuansheng Hua, Rong Huang, et al. So2sat lcz42: A benchmark dataset for global local climate zones classification. *arXiv preprint arXiv:1912.12171*, 2019.

[77] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix C
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.6.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] They are creative common datasets
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Let's discuss it here. There is no text data and remote sensing data is unlikely to contain PII. The only high-resolution dataset is NeonTree and it is collected over protected forests in the United States.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Extended Results

In this section, we continue the experiment sections to include other results, that were deferred due to space limitations.

A.1 Benchmark Coverage

In Figure 8, we depict the coverage of the benchmark on the world map. While there are still large uncovered areas such as China, Russia, North Africa, and South America, the benchmark covers all continents except Antarctica. Most importantly, the coverage of the benchmark is greater than that of individual datasets.

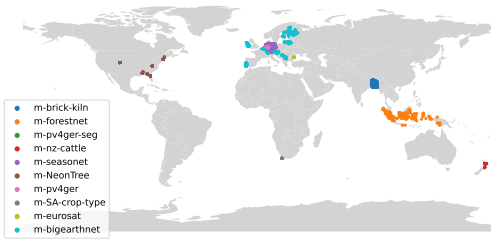

Figure 8: World coverage of the different datasets.

A.2 Classification

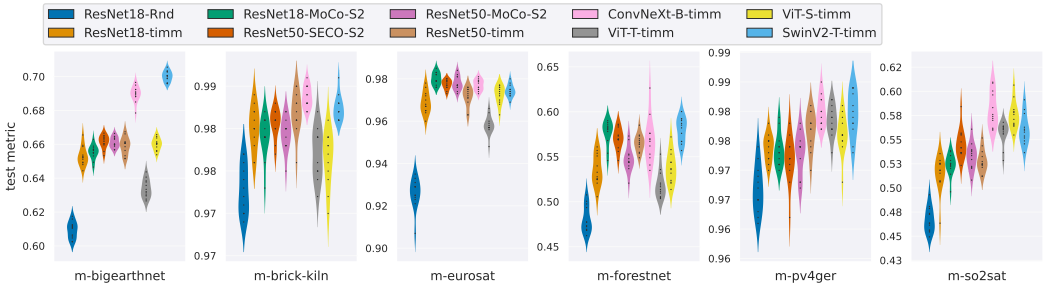

Figure 9: **Classification Benchmark:** Raw accuracies of all seeds of various baselines (higher is better). Violin plots represents the distribution of seeds.

In Figure 9, we report the raw accuracies, before normalisation and IQM. This different perspective, gives a sense of the variance of the results as well as how close it is to the maximum. We recall that uncertainty of the mean expressed in Figure 4 is lower than the variance of the results in Figure 9. This follows from the central limit theorem.

A.3 Segmentation

In this section, we report results for six baselines on the Segmentation benchmark. First, we introduce the baselines that are evaluated.

A.3.1 Baselines

ResNet18 U-Net - ResNet101 U-Net ResNet augmented with the U-Net architecture [58] with pre-trained weights from the timm library.

ResNet18 DeepLabV3 - ResNet101 DeepLabV3 ResNet augmented with the DeepLabV3 architecture [10] with pre-trained weights from the timm library.

A.3.2 Comparing Baselines on RGB only

In Figure 10, we report the bootstrapped IQM of the normalized Intersection over Union (IoU) (Sec. 4.1) for the 6 segmentation datasets. In Figure 11, we report the seeds for the 10 experiments.

From the results, we can observe a stronger performance with U-Net architecture and the larger backbone ResNet101 performs a bit less than ResNet50 in general but has less stable performance, leading to slightly lower performance than ResNet50 backbone.

In Figure 12, we observe the behaviour of the baselines as the size of the training set grows from 1% to 100%.

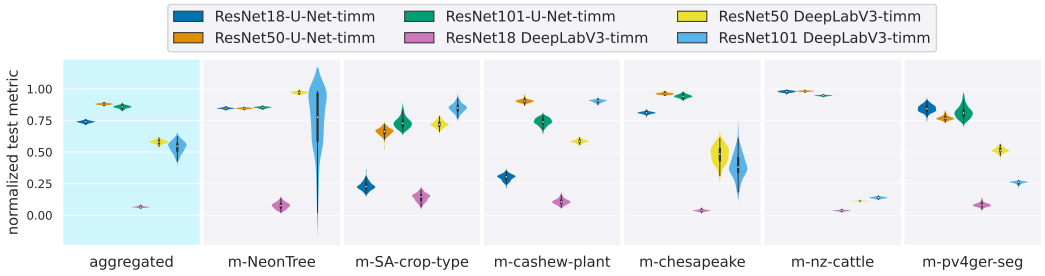

Figure 10: **Segmentation Benchmark:** Normalised intersection over union (IoU) of various baselines (higher is better). Violin plots are obtained from bootstrap samples of normalized IQM (Section 4.1). Left plot reports average across all tasks.

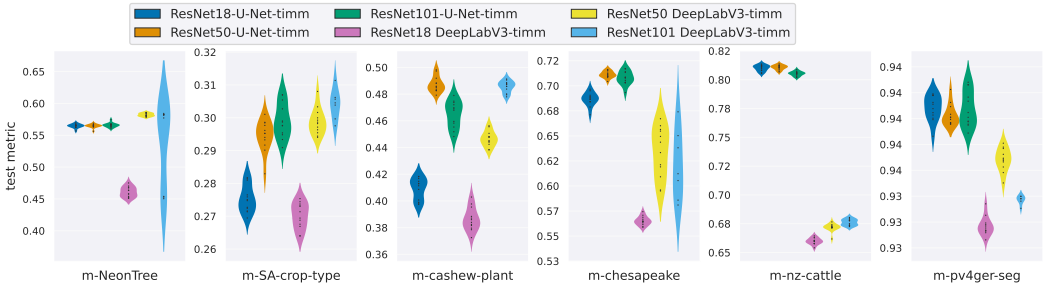

Figure 11: **Segmentation Benchmark:** Raw IoU of all seeds of various baselines (higher is better). Violin plots represents the distribution of seeds.

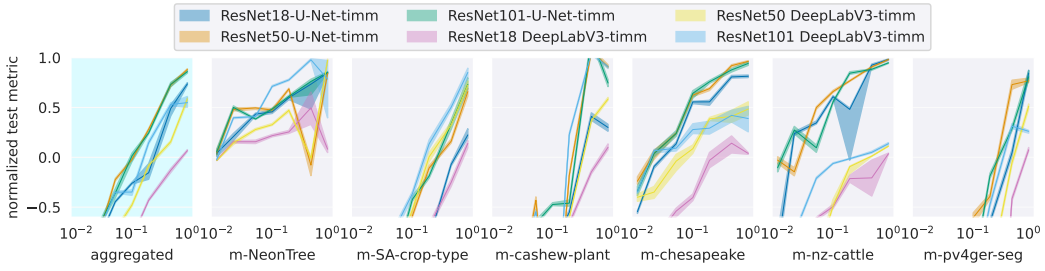

Figure 12: **Segmentation vs Train Size:** Normalised IoU (higher is better) on the Segmentation benchmark with a growing size of the training set. Shaded region represents 80% confidence interval, obtained from bootstrap samples of normalized IQM (Section 4.1). Left plot reports average across all tasks.

A.4 Convergence Time

As seen in Section 6.2.3, conducting experiments with a growing training set size brings a different and important perspective on the performances of the models. In our experiments, this comes with a seven

fold increase in *number* of experiments. On the surface, this may seem like an excessive amount of computation, but most experiments will run much faster with smaller training set. Indeed, if we decrease the training size at an exponential pace, and we assume that the training time is proportional to the size of the training set, we get a more modest increase in computational need. In GEO-Bench, we generate pre-defined subsets using the following ratios of the training set: $(1\times, 0.5\times, 0.2\times, 0.1\times, 0.05\times, 0.02\times, 0.01\times)$. With the proportional training time assumption, this leads to a cumulative $1.88\times$, which is less costly than repeating the experiment twice, and far from the seven fold increase that one could assume.

To confirm the assumption that the training time is proportional to size of the training set, we conduct the following experiment. As a measure of the training time, we use the convergence time i.e., how many training steps[12] are required to achieve the peak performance on the validation set. Let $\tau^r_{i,j,k}$ be the convergence time of model i, on dataset j, with hyperparameter trial k, trained with a ratio r of the training set. Let the reference convergence time be defined as the average convergence time when using 10% of the training set:

$$T_{i,j} = \frac{1}{n_k} \sum_{k=1}^{n_k} \tau^{0.1}_{i,j,k}.$$

Then, the average relative convergence time is:

$$\rho^r_j = \frac{1}{n_i, n_k} \sum_{k=1}^{n_k} \sum_{i=1}^{n_i} \frac{\tau^r_{i,j,k}}{T_{i,j}}.$$

In Figure 13, we plot ρ^r_j for all datasets and all partition size on a log-log plot. We can observe a mild behaviour change near 1% of the training size, but overall we can conclude that the convergence time is proportional the training set size.

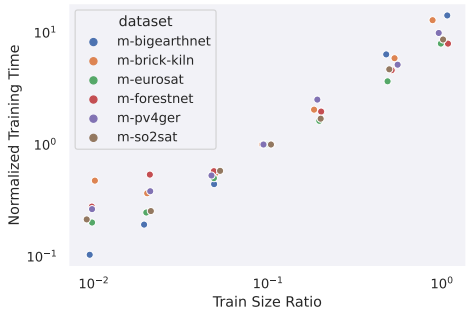

Figure 13: **Convergence Time:** Average time for the training to reach convergence as the training size increase.

A.5 Discriminativity of Datasets

It is common knowledge in machine learning that larger datasets yields better generalization performances. However, when comes the time to compare algorithms against each other, we hypothesises that less is more i.e. smaller datasets are more likely to exhibit statistical difference between a given pair of models.

In this section we provide experimental evidences on this question. Let $p(A^l_i > A^l_j)$ be the probability that algorithm i is better than algorithm j on dataset l. More concretely, A^l_i and A^l_j are random variables corresponding to the accuracies on the validation set of task l. We estimate this probability by repeating the training procedure for 10 random seeds, as recommended in Section 4.1, and comparing all pair of results. Using this quantity, we propose the following discriminativity metric

$$d^l_{ij} := 1 - \mathbb{H}_2[p(A^l_i > A^l_j)],$$

where $\mathbb{H}_2$ corresponds to entropy in bits i.e., using log_2. This measures how good dataset l is good at telling if algorithm i is better than algorithm j. For example, if algorithm i is always better than j or vice

[12] We also multiply by the batch size

versa, then we have: $d^l_{ij} = 1$. On the other end if they perform equally well, then $d^l_{ij} = 0$. Next, to get an estimate at the dataset level, we average discriminativity across all pairs of m models to obtain

$$D_l := \frac{1}{m^2} \sum_{ij} d^l_{ij}.$$

Finally, to obtain en estimate of the uncertainty of this measure, we use stratified bootstrap of all experiments, and repeat this procedure 100 times.

In this experiment, we consider a smaller training set as a different D_l value. Hence, with seven different partitions on the 6 tasks of GEO-Bench classification, this leads to a total of 42 different datasets. In Figure 14, we analyse the influence of the training set size on the discriminativity of a dataset, and we conclude:

- Reducing the dataset size almost systematically improve its ability to discriminate between models, but only by a small value.
- BigEarthNet is the most discriminative dataset
- The full size of pv4ger offers poor discriminativity but could be improved if reduced.
- Our estimation of discriminativity is quite noisy and sensitive to the random seeds.

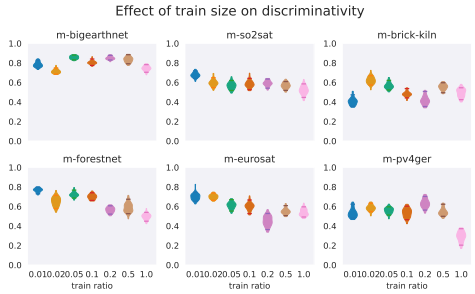

Figure 14: **Discriminativity of datasets:** We report how discriminative are each datasets as we vary the training set size.

A.6 Resources Usage

For convenience, we also provide resource usage of the different models in Figure 15.

Number of Parameters: The number of parameters of a model gives a hint on the overall memory usage, but most importantly on the learning capacity of the model. In Figure 15-left, we see that ConvNeXt is by far the one with the highest capacity. While ViT-T has less parameters than ResNet18. We note that SwinV2 has up to 3 billion parameters with SwinV2-G, but we focus on models that could be run on a single 32 GB GPUs without extra work.

Memory Usage: While the number of parameters gives a hint about the memory usage, to capture the memory usage of all hidden states we need to measure the memory usage in action with `nvidia-smi`. This is reported in the second plot of Figure 15.

Forward time: We measure the time for a forward pass of the network with a batch size of 32. This gives a sense of the efficiency of the network deployed in production. Values are reported in the third plot of Figure 15. Violin-plot report the distribution of several measurements during one epoch of training on BigEarthNet, and the average value is reported as a solid line.

Convergence time: In Figure 15 right, we report the number of training step required to reach peak generalization performance on validation. These values are reported on from the 100% training set, and the violin-plot report the distribution of values across the different tasks of GEO-Bench-classification and the different trials of the hyperparameter search.

B Remote Sensing Data Schema

Band Earth monitoring data comes with a challenging amount of heterogeneity. Fortunately, the transformer architecture offers the opportunity to mix various modalities using encodings such as temporal

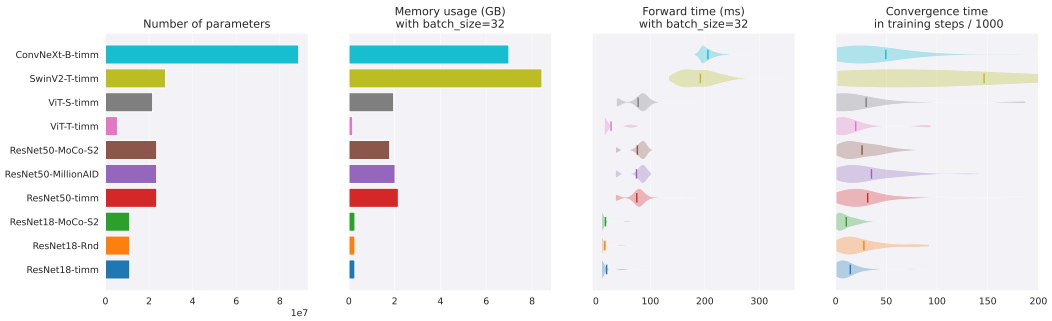

Figure 15: **Resources Usage:** Resources reported for various models. Violin-plots report distribution over several tests and its average as a solid line. See text for more details.

encoding [68] and positional encoding [19]. Similarly, band encoding can be leveraged to communicate the source of the data. To this end, we define `Band` as the core class in our schema. It consists of an array of data with spatial extent accompanied with `BandInfo`, providing information such as the band name, spatial resolution, and spectral range. Through a hierarchy of classes, we also provide the type of sensors (see Figure 16). This provides further information for introspection and flexible information for users to define a *Band Encoding* that could be required for transformer architectures. Finally, a `Sample` is a set of `Band` accompanied by a label which can also be a `Band`.

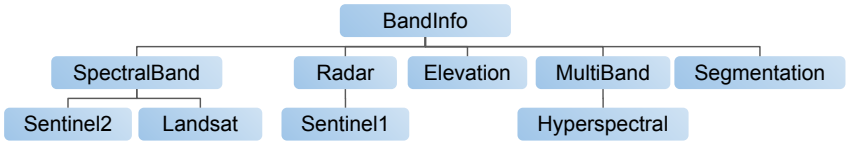

Figure 16: Class hierarchy of `BandInfo` enabling introspection on the type of data.

Task Specifications Each dataset is accompanied by a `TaskSpecifications` object describing the schema of a particular dataset without having to load any samples. It contains the dataset name, the type of labels, the `BandInfo` of each band and their shapes. The aim of this data structure is to let users procedurally generate a machine learning model that is suitable for the given dataset at the beginning of the training.

Band Statistics We also provide band statistics (minimum, maximum, mean, variance, and percentiles). This lets users transform the input data in various possible ways to fit the statistics expected by the pre-trained model.

C Societal Impact of Foundation Models for Earth Monitoring

Remote sensing and Earth monitoring have been transformational in the past decades. Applications include military, insurance, market forecasting, climate science, and more. Much of this impact is not directly attributed to deep learning nor large pre-trained networks and its review extends beyond the scope of this section. In this section, our focus is on the impact of bringing foundation models to Earth monitoring.

C.1 Climate mitigation and adaptation

Machine learning on remote sensing data is widely used to develop solutions for a variety of problems relevant to climate change [8, 57, 77, 45]. The vast majority of these solutions are built by curating datasets for a specific task and require significant resources to develop. Furthermore, the solutions are often tailored to specific regions as extending approaches to new geographies remains a significant challenge, primarily due to the lack of labeled data [77]. Less-economically developed regions of the world are no less susceptible to the impacts of climate change, yet suffer from the lack of effective remote sensing-based solutions [8]. Foundation models for Earth monitoring have the potential to address many of these issues and substantially accelerate and enable the development of new remote sensing solutions for climate change.

C.2 Increased accessibility

Reducing the need for curating a large labeled dataset for each task could democratise access to the development of machine learning models for remote sensing, specifically for groups or organisations with limited budgets [47, 3]. In particular, foundation models may especially benefit non-profit organisations, academic universities, startups, and developing countries. It may also open opportunities for applications that were not previously profitable. Although we believe that increased accessibility to these models will have a largely net positive impact, we acknowledge that this accessibility may lead to unexpected applications with potentially negative impacts [6]. We also note that such models may have dual-use applications, where, for example, they may help oil and gas industries in their operations in ways that increase (or reduce) overall emissions.

C.3 Emissions of large pre-trained models

Recent work has investigated emissions of large neural networks [63, 60, 59, 35, 52]. Specifically, training a large transformer can emit 284 tCO_2e when trained on computers using largely fossil fuel energy (US national average) [63]. When put in perspective with individual actions, such emissions are large—e.g., a roundtrip passenger flight from San Francisco to London is 2.8 tCO_2e , about 100$\times$ smaller. However, the extensive reusability of pre-trained models and their potential for helping efforts to mitigate climate change [57] calls for a different perspective.

When evaluating new tools and systems, it is important to consider the likely net impact on emissions of both the creation and testing of the tool and its eventual deployment. For example, evaluating the performance of airborne methane sensing tools at emission levels commonly found in oil and gas operations can emit about 7 metric tonnes of methane, roughly 600 tCO_2e equivalent using a 20-year global warming potential [22]. However, in a single day of flying, such a single instrument can survey hundreds of sites, often identifying leaks for repair that emit well over 7 metric tonnes of methane per day [31]. Similarly, foundation models may significantly advance our ability to leverage enormous quantities of passively collected satellite data to massively reduce emissions, qualitatively advance our understanding of climate science, or improve our ability to adapt to climate change.

In sum, the potential benefits for climate change mitigation with improved Earth monitoring methods likely outweigh the emissions associated with foundation models. Moreover, various actions can be taken to reduce and mitigate emissions related to the training of your model [35]:

- Select data centers that are certified carbon neutral or largely powered by renewable energy, with good power usage effectiveness (PUE). Such measures can reduce emissions dramatically 50$\times$ reduction in emissions [35].
- Design your code development pipeline to minimize the number of computationally-intensive runs required, e.g. employ modular development and testing when possible.
- Make your code more efficient and sparsify your network when possible [52]. This can reduce emissions up to 10-fold.
- Favour more energy-efficient hardware, e.g., TPUs or GPUs.
- Monitor [59] and report your emissions [35]. Better communication about climate change is fundamental for systemic changes. Better documentation will help other coders pick up where you left off, potentially bypassing some computationally intensive runs.
- Offset the cumulative emissions of your projects.

C.4 Fairness and biases

Large language models are known to amplify and perpetuate biases [5]. While this can lead to serious societal issues, we believe that biases in remote sensing models are likely to have much less impact. We do however anticipate potential biases and fairness issues.

Data coverage and resolution Some satellites cover the whole Earth with standard spatial resolution and revisit rate (e.g., Sentinel-2 covers the whole Earth at 10-60 m/pixel resolution every 5 days). This makes imagery freely available uniformly across the planet. Other satellite data providers such as Maxar acquire images on-demand and have higher spatial resolution (up to 0.3m per pixel), but also have lower revisit rates and high costs. Some countries, such as New Zealand, freely provide aerial imagery with

resolution up to 0.1m per pixel[13]. Finally, it is worth noting that cloudy seasons in some climates may limit data availability for some countries. Overall, while the coverage is fairly uniform, some regions have much higher coverage than others and money can be a limiting factor to access the data. This can lead to some level of biases and fairness issues.

[13] https://data.linz.govt.nz/