Benchmarking Self-supervised Learning Methods in Remote Sensing

Fabien Merceron, Vincent Partimbene, Gohar Dashyan, Sébastien Saubert,
Guillaume Peltier, Kévin Sanchis and Pierre-Antoine Ganaye
Safran.AI
Paris, France
Email: <name>.<surname>@safran-ai.safrangroup.com

***Abstract*—Self-supervised pretraining has proved to be a competitive tool to improve downstream task performance in the field of remote sensing. Attempts to create geospatial foundation models based on such pretraining techniques are increasing in numbers, and are a promising solution to exploit the vast amount of unannotated remote sensing imagery. Due to the widespread availability of various self-supervised techniques, either generic or specific to remote sensing, it becomes of importance for practitioners to find a way to identify the best performing pretraining method based on the downstream task being tackled. In this paper, we present a systematic benchmark of commonly used self-supervised pretraining methods and provide insights into the most appropriate approach depending on the chosen downstream tasks. Our results indicate that Masked Auto Encoders (MAE), a reconstruction-based method, seems to be the overall winner on most use-cases. We also show that ImageNet remains a powerful pretraining dataset and can produce competitive baselines, while building a tailored pretraining dataset using high-resolution satellite images can effectively improve the downstream performance compared to such baselines. Finally, we study the computational efficiency of pretraining methods and provide recommendations based on the available budget.**

***Index Terms*—deep learning, computer vision, remote sensing, optical imagery, foundation model, self-supervised learning, land use classification, object detection, semantic segmentation, benchmark**

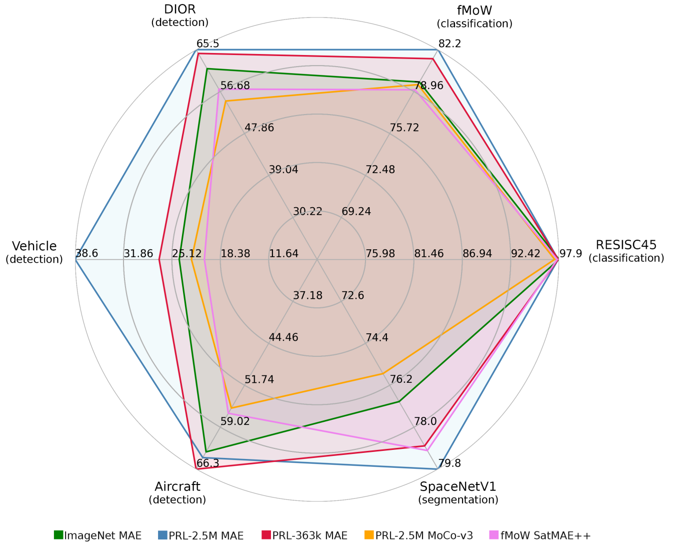

Fig. 1. Pretraining a ViT-Large with MAE on our internal datasets (PRL-2.5M and PRL-363k) improves performance on most downstream tasks compared to publicly available weights.

I. Introduction

The emergence of foundation models, a family of general-purpose models for solving a wide range of computer vision tasks, has overturned the traditional methodology of using a dedicated model for each task. This paradigm shift allows a common backbone to be used for any downstream task, especially classification, segmentation and detection. The training of a foundation model usually consists of two steps: pretraining and finetuning. The pretraining is done with the self-supervised learning (SSL) methodology, it makes it possible to train an encoder on a large quantity of unlabeled data using a pretext task, in order to learn how to extract meaningful visual features. Once complete, the finetuning step is applied to learn the downstream task, reusing the model's encoder while creating a new decoder.

If the pretraining is carried out correctly, the weights of the encoder can be reused in a variety of downstream tasks. Thus, selecting the most adapted SSL method is crucial as it highly impacts the generalization capabilities of the resulting model. Furthermore, the efficiency of the method is another important factor to be considered as pretraining is the main source of computational costs, especially when using large datasets.

Faced with all these technological opportunities, the question logically arises as to which is the optimal pretraining solution to use within the context of remote sensing.

In this article, we provide details on the efficiency of some SSL methods for vision tasks. Specifically, we study the effectiveness of a chosen joint embedding and reconstruction-based method on several downstream tasks commonly performed in remote sensing: land use classification, object detection and semantic segmentation. To do this, we study performance variation by exploring two major criteria: the size of the backbone and the pretraining dataset's size and composition.

Our main contributions are as follows:

- We explore the effectiveness of SSL methods in the context of remote sensing in several downstream tasks, including dense tasks such as object detection and semantic segmentation.
- We use several pretraining datasets of different size and composition, including very high-resolution commercial images and applications in both civilian and military

contexts. Furthermore, we study the benefits of such datasets compared to publicly available ones.

- We conduct our experiments on several backbone sizes to study the scaling capabilities of each method, including a large backbone that isn't commonly used in other works.
- We compare in-domain pretrainings with publicly available weights that serve as practical baselines for limited budgets and question the necessity to perform dedicated pretrainings to achieve high downstream performance.

As shown in Fig. 1, our internal datasets, combined with MAE, are able to consistently outperform other approaches. However, we note that publicly available weights are competitive alternatives when custom pretraining is not an option. We hope that our work proves useful to practitioners in facilitating the selection of an SSL method for remote sensing applications.

II. Related Work

A. Self-supervised Learning

SSL methods can be classified into different families based on how they interact with data. joint embedding methods learn to map similar data points close to each other in a latent space, while performing the opposite for dissimilar data points. MoCo [1] is arguably one of the reference methods of this family, building positive pairs using random data augmentation on a given image, and negative pairs using other images. Additionally, a contrastive loss that incentivizes similar embeddings for positive pairs and dissimilar embeddings for negative pairs is computed. To store negative pairs, MoCo uses a dictionary queue of fixed size (also called memory bank), which is decorrelated from the batch size. SimCLR [2] follows the same strategy, but uses the images contained in the current batch to form negative pairs, which alleviates the need of maintaining a queue, but requires a large batch size to achieve good performance. Later on, MoCo-v2 [3] and MoCo-v3 [4] improve the performance of the original method by using components from SimCLR and by replacing the original ResNet50 encoder with a ViT.

Another family of interest is reconstruction-based methods. Specifically, MAE [5] is arguably one of the most commonly used method of this type, which uses a ViT-based encoder-decoder architecture. The input image is first divided into non-overlapping patches of equal size. Then, most of the patches (typically 75%) are masked and the remaining patches are sent through the encoder as a sequence. Finally, the decoder aims at reconstructing the masked parts of the input as a reconstruction loss is used for learning. MAE significantly improves the efficiency of pretraining by only processing the visible patches.

B. Self-supervised Learning in Remote Sensing

The application of SSL methods in the field of remote sensing has received much attention in recent years [6], which can be explained by the large amount of available data and the high cost of the annotation process. These methods proved to be a competitive alternative to the traditional supervised learning methods [7]. Additionally, remote sensing imagery comes with its own set of characteristics that can be exploited to further improve performance, e.g., varying Ground Sample Distances (GSD), temporal dependencies between images, or the availability of additional bands in the context of multispectral imagery. Following this direction, adaptations of existing SSL methods have been proposed to take into account such specificities. SatMAE [8] leverages temporal and multispectral information during the reconstruction task by concatenating image timestamps to the positional encoding, and by using a curated selection of additional multispectral bands on top of the usual RGB ones for pretraining. Scale-MAE [9] incorporates a form of multi-scale decoding using Laplacian pyramids and proposes an update of the positional encoding to take into account the GSD of satellite images. SatMAE++ [10] extends Scale-MAE to work with multiple image scales by downsampling the input image twice and only sending the most downsized image through the encoder-decoder before upsampling it back to its original resolution. In this paper, we compare generic SSL methods with ones that take advantage of these specificities, and measure the expected gains associated with this additional complexity.

C. Self-supervised Learning Benchmarks in Remote Sensing

In the context of remote sensing, few works are dedicated to benchmarking SSL methods. Wang et al. [11] proposes an exhaustive review of SSL methods, including MoCo-v2, and provides a benchmark of several methods on three datasets. However, evaluation is only performed on a classification task using linear probing. Corley et al. [12] studies the impact of image sizing and normalization during pretraining on downstream task performance, and argues that ImageNet pretraining is a solid competitor. Nonetheless, this work only studies classification as a downstream task and MoCo-v2 as a pretraining method, and doesn't perform any finetuning.

On another note, some works that aim at building geospatial foundation models also provide extensive benchmarks of several SSL methods and downstream tasks. Similar to our work, Cha et al. [13] investigates the impact of increasing the number of model parameters in the context of SSL pretraining applied on object detection and semantic segmentation, primarily using ViTs. However, this work only explores a single pretraining method and a single pretraining dataset. More recently, Guo et al. [14] compares their proposed SkySense foundation model with a handful of SSL methods built for remote sensing, on various downstream tasks including classification, segmentation and detection. However, no attention is given to generic SSL methods nor ImageNet pretraining, as well as pretraining efficiency.

In contrast, our work focuses on the impact of different SSL methods on downstream task performance (classification, detection and segmentation) using two backbone sizes. It also investigates the impact of the pretraining dataset size and composition, while providing a pragmatic look on the necessity to perform a dedicated pretraining by comparing the achievable performance with more frugal approaches based on publicly available weights, e.g., ImageNet.

III. Experimental Setup

In the following, we thoroughly compare the performance of joint embedding methods and reconstruction-based methods on various downstream tasks, in the context of remote sensing. Specifically, we choose to compare the performance of MoCo-v3 and MAE as two reference approaches, respectively, for these two families, and also as commonly used methods in remote sensing. The objective of these experiments is to get a better understanding of how each family of methods behaves with respect to the chosen downstream task, which in turn is useful to decide which pretraining method to favor when working on a downstream task. One might also wonder if remote sensing specific data and designs are necessary and significantly beneficial for downstream performance. For this reason, we also compare our results with SatMAE and SatMAE++ as the best performing remote sensing specific SSL methods, and with several in-domain pretraining datasets of various sizes. Finally, to measure the impact of backbone size, we systematically use ViT-Base and ViT-Large as backbones for all our experiments.

A. Pretraining

We study the impact of the pretraining data on downstream performance by first performing a self-supervised pretraining step on several remote sensing datasets. One of these is a public reference in the field, while others are internal datasets that we use to measure the expected gain when scaling the pretraining dataset size; our largest dataset being around 2.5 million images, which is approximately seven times larger than the public dataset we are using. In the following, we describe each of these datasets and discuss implementation details for the self-supervised methods we choose to focus on.

1) Datasets:

a) fMoW RGB: Functional Map of the World (fMoW) [15] is a large-scale dataset for functional land use classification. The dataset offers a wide range of ground resolutions from 0.5m to 35m per pixel. Since the original image size of fMoW varies, we pre-process the images identically to [15] and resize the input images to 224 × 224 pixels.

b) PRL-363k and PRL-2.5M: We build two high-resolution datasets by using mostly commercial data from the *Maxar/DigitalGlobe* satellites WorldView 2, 3, 4 and Pléiades Neo 3, 4. PRL-363k consists of the same number of images as fMoW RGB, i.e. 363,571 images. PRL-2.5M is a larger dataset consisting of 2,552,188 images. Both datasets consist of a curated collection of optical images with native GSDs ranging from 0.3m to 0.7m per pixel, all resampled to 0.3m. The geographic location of the images is not specified. The original image size is 512×512 but a 224×224 random crop of the input image is taken for pretraining.

2) Implementation Details:

a) MAE: The model configuration, optimizer and learning rate scheduler are the same as in [5]. We use 32 NVIDIA A100 to pretrain our model for 800 epochs with a base learning rate of $2.4e{-}4$ and an effective batch size of 16,384 for ViT-Base and 8192 for ViT-Large. We adopt the linear learning rate scaling rule [16]: $lr = base\_lr \times \frac{batchsize}{256}$. We apply data augmentation by performing a random flip with probability 0.5, and images are normalized using the standard ImageNet normalization. We use ImageNet MAE pretrained weights as initialization before pretraining following [17]. For all subsequent finetunings, we use the last epoch to initialize the backbone weights.

b) MoCo-v3: The model configuration, optimizer and learning rate scheduler are the same as in [4]. We use 32 NVIDIA A100 to pretrain our model for 300 epochs with a base learning rate of $2.4{-}4$ and an effective batch size of 4096 for both ViT-Base and ViT-Large. As for MAE, we adopt the linear learning rate scaling rule [16]. We apply the same data augmentation as described in [4]. We use ImageNet MoCo-v3 pretrained weights as initialization before pretraining following [17]. For all subsequent finetunings, we use the last epoch to initialize the backbone weights.

B. Downstream Tasks

Our pretrained models are finetuned on various downstream tasks, including classification (fMoW, RESISC45), segmentation (SpaceNetV1) and object detection (DIOR, PRL-Vehicle and PRL-Aircraft). In the following, we describe the content of each dataset as well as the implementation details for finetunings. Finally, we present the evaluation metrics used to measure downstream performance and the selected baselines.

1) Datasets:

a) fMoW RGB: We also use fMoW as a downstream classification task. We follow the official train and validation splits, which consist of 363,571 images and 50,041 images, respectively, distributed across 62 fine-grained and diverse categories.

b) RESISC45: Northwestern Polytechnical University (NWPU) developed RESISC45 [18], including 31,500 images distributed across 45 different scene categories from over 100 countries extracted from Google Earth. Each category contains 700 labeled images of size 256 × 256 pixels, resulting in a well-balanced distribution of scenes. The spatial resolutions of the images range from 0.2 to 30 meters per pixel. We use the dataset splits defined in [19] and keep the original input image size of 256 × 256.

c) SpaceNetV1: A dataset of 6,940 WorldView 2 satellite images at 0.5m per pixel [20]. We convert the original building footprint annotations (i.e. polygons) into segmentation masks and use the same dataset split as [8]. We keep the original image size of 400 × 400 pixels.

d) DIOR: A large-scale benchmark dataset for object detection in optical remote sensing images, which consists of 23,463 images of resolution varying from 0.5 to 30m per pixel and 192,518 object instances annotated with non-oriented bounding boxes. We follow the official train, validation and

TABLE I
STATISTICS FOR PRL-VEHICLE AND PRL-AIRCRAFT OBJECT DETECTION DATASETS

Observable	Split	Images	Pos. tiles	Neg. tiles	Num. Objects
Vehicle	train	204	49,932	9,963	369,851
	val	63	20,297	43,409	170,864
	test	88	4,712	46,269	32,550
Aircraft	train	3,239	19,110	3962	51,136
	val	202	2,067	5,348	5,386
	test	83	496	17,393	1,235

test splits, which are composed of 5,862 images, 5,863 images, and 11,738 images, respectively.

e) PRL-Vehicle and PRL-Aircraft: Our internal downstream datasets consist of Maxar WorldView-3 satellite images at 0.3m resolution, divided into tiles of 224 × 224 pixels and 512 × 512 pixels for vehicle and aircraft, respectively. The statistics of our datasets are reported in Table I. The vehicle dataset covers 8 military and civilian classes, while the aircraft dataset covers 6 military and civilian classes. Note that for PRL-Vehicle, we expect results to be on the lower end compared to other datasets as objects are very small.

2) Implementation Details:

a) Classification: We use a linear classification head on top of the ViT backbone. Augmentations, optimizer and learning rate scheduler are the same as in [5]. We use an effective batch size of 2048 for ViT-Base and 1024 for ViT-Large. Regarding the training we use a base learning rate of $2e{-}3$ and $4e{-}3$ for ViT-Base and ViT-Large, respectively. For the other configurations, we use a base learning rate of $1.5e{-}4$. We apply a layer-wise learning rate decay [21] of 0.75 for ViT-Large and 0.65 for ViT-Base following [22]. We use 4 NVIDIA V100 and finetune for 50 epochs on fMoW-RGB and 100 epochs on RESISC45, as in [9].

b) Segmentation: We use the UPerNet [23] head to perform semantic segmentation, as well as the feature pyramid implementation of ViTDet [24] to exploit multi-scale features. Augmentations, optimizer and learning rate scheduler are the same as in [8]. We use a base learning rate of $1.5e{-}4$ and an effective batch size of 64 for both ViT-Base and ViT-Large. We use 4 NVIDIA V100 and finetune for 100 epochs on SpaceNetV1, as in [8].

c) Detection: We use a RetinaNet [25] head to perform the detection task as well as the feature pyramid implementation of ViTDet [24]. Augmentations, optimizer and learning rate scheduler are the same as in [8]. The effective batch size is 64 for each dataset. For PRL-Aircraft and PRL-Vehicle, we use 4 NVIDIA V100 and finetune for 50 epochs. For DIOR, we use 4 NVIDIA A100 and finetune for 100 epochs. During finetuning we use LoRA [26] to get a better and faster convergence. Note that we do not use LoRA for other downstream tasks as it results in a performance drop.

C. Evaluation Metrics

We use the following metrics for our evaluations:

- **Classification** Top-1 accuracy. The evaluation epoch is selected based on the highest top-1 accuracy achieved on the validation set.
- **Segmentation** Mean Intersection over Union (IoU). The evaluation epoch is selected based on the highest mean IoU achieved on the validation set.
- **Detection** Mean average precision (mAP@0.5) of the PASCAL VOC object challenge [27]. The evaluation epoch is selected based on the highest mAP@0.5 achieved on the validation set.

D. Baselines

We compare our own pretrained backbones to several reference baselines that are identical for all downstream tasks. The first group of selected baselines consists of ImageNet pretrained weights. For that, we select supervised, MAE and MoCo-v3 weights for both ViT-Base and ViT-Large backbones. The second group consists of in-domain baselines, composed of SatMAE and SatMAE++ as they achieve state-of-the-art performance in the field of remote sensing. For compatibility reasons with the input data of our downstream tasks, we only use the RGB weights (not the multi-temporal or multi-spectral versions) of these methods. As weights are only available for ViT-Large, SatMAE and SatMAE++ will only be used as reference for this backbone.

IV. EXPERIMENTAL RESULTS

In this section, we discuss the results of our experiments with the aim of drawing insights about the behavior or MAE and MoCo-v3 when presented with various pretraining and downstream datasets. In some cases, the performance achieved by most methods are very close to each other. We argue that a variance study would have been beneficial to consolidate our conclusions, but we were not able to do so due to the high amount of additional experiments to be run.

A. Classification

Table II shows the Top-1 accuracies for RESISC45 and fMoW RGB. First, comparing the ImageNet baselines with our own built pretrained weights, we can see that the ImageNet baselines are strong. Specifically for RESISC45, the best weights for ViT-Base are ImageNet supervised and MoCo-v3 pretraining on PRL-2.5M. For ViT-Large, the best performance are obtained with ImageNet pretraining with MAE, closely followed by PRL-2.5M pretrained with MAE. Regarding fMoW RGB, our pretrainings outperform baseline approaches but the gap is mainly noticeable on PRL-2.5M pretrained with MAE, which ranks first for ViT-Base and ViT-Large. We argue that the competitive performance of ImageNet baselines can be explained by the fact that ImageNet is a dataset built for classification with centered, well-sized observables. Thus, features generated by supervised or SSL pretraining with ImageNet should be adequate by design for any classification task.

Regarding SatMAE and SatMAE++ with the ViT-Large backbone, we can see that none of them outperform the ImageNet MAE baseline, but, it should be noted that they support

TABLE II
Top-1 accuracy on the RESISC45 and fMoW classification datasets

Backbone	Dataset	Method	RESISC45	fMoW
ViT-Base	-	Random init.	76.7	66.0
ViT-Base	IN	MoCo-v3	97.4	78.2
ViT-Base	IN	MAE	97.5	78.6
ViT-Base	IN	Sup.	**97.6**	79.0
ViT-Base	fMoW	MoCo-v3	97.5	79.2
ViT-Base	fMoW	MAE	97.5	79.5
ViT-Base	PRL-363k	MoCo-v3	97.5	78.9
ViT-Base	PRL-363k	MAE	97.5	**80.1**
ViT-Base	PRL-2.5M	MoCo-v3	**97.6**	79.8
ViT-Base	PRL-2.5M	MAE	97.4	**80.1**
ViT-Large	-	Random init.	70.5	68.6
ViT-Large	IN	MoCo-v3	97.2	78.0
ViT-Large	IN	MAE	**97.9**	79.7
ViT-Large	IN	Sup.	97.4	79.0
ViT-Large	fMoW	SatMAE	97.0	76.1
ViT-Large	fMoW	SatMAE++	97.7	79.1
ViT-Large	fMoW	MoCo-v3	97.7	79.4
ViT-Large	fMoW	MAE	97.5	80.3
ViT-Large	PRL-363k	MoCo-v3	97.1	78.9
ViT-Large	PRL-363k	MAE	97.8	81.5
ViT-Large	PRL-2.5M	MoCo-v3	97.4	79.5
ViT-Large	PRL-2.5M	MAE	97.8	**82.2**

TABLE III
mAP@0.5 on the DIOR, PRL-Vehicle and PRL-Aircraft detection datasets

Backbone	Dataset	Method	DIOR	Vehicle	Aircraft
ViT-Base	-	Random init.	29.2	12.6	29.9
ViT-Base	IN	MoCo-v3	51.8	19.9	59.5
ViT-Base	IN	MAE	55.7	22.2	60.6
ViT-Base	IN	Sup.	56.2	20.6	**65.4**
ViT-Base	fMoW	MoCo-v3	52.2	20.2	57.0
ViT-Base	fMoW	MAE	55.2	23.4	55.6
ViT-Base	PRL-363k	MoCo-v3	51.7	21.9	59.6
ViT-Base	PRL-363k	MAE	55.6	22.9	58.3
ViT-Base	PRL-2.5M	MoCo-v3	53.1	24.3	57.2
ViT-Base	PRL-2.5M	MAE	**60.5**	**25.4**	64.6
ViT-Large	-	Random init.	21.4	4.9	30.0
ViT-Large	IN	MoCo-v3	55.2	19.8	65.3
ViT-Large	IN	MAE	61.5	24.1	63.3
ViT-Large	IN	Sup.	57.4	22.5	64.7
ViT-Large	fMoW	SatMAE	53.1	21.0	54.2
ViT-Large	fMoW	SatMAE++	57.2	20.6	56.6
ViT-Large	fMoW	MoCo-v3	56.6	22.3	58.1
ViT-Large	fMoW	MAE	59.7	26.5	62.5
ViT-Large	PRL-363k	MoCo-v3	54.7	21.0	60.1
ViT-Large	PRL-363k	MAE	64.7	26.9	**66.3**
ViT-Large	PRL-2.5M	MoCo-v3	54.7	22.4	55.7
ViT-Large	PRL-2.5M	MAE	**65.5**	**38.6**	64.3

multi-spectral / multi-temporal inputs that our experimental setup does not.

Looking at the benefit of our internal datasets against the fMoW RGB dataset, we can see a positive performance impact. Indeed, PRL-2.5M always ranks higher than fMoW RGB, especially with the MAE paradigm, which highlights the usefulness of scaling up the amount of pretraining data. PRL-363k shows weaker results than PRL-2.5M, but still manages to improve performance over fMoW RGB, especially with the MAE paradigm, which could be explained by the higher resolution of PRL-363k images compared to fMoW.

Finally, by focusing our attention on pretraining methods, we can observe that the MAE paradigm shows better performance than MoCo-v3. Indeed, except for the RESISC45 dataset with the ViT-Base backbone, MAE consistently outperforms MoCo-v3. On top of that, we can see that the use of PRL-363k or PRL-2.5M over fMoW has a negative performance impact on MoCo-v3, which is not the case with MAE.

B. Detection

Table III shows the mAP@0.5 for DIOR and both PRL-Aircraft and PRL-Vehicle. The ImageNet baselines on all datasets are competitive especially on the PRL-Aircraft dataset where it ranks first for ViT-Base and among the top for ViT-Large. For DIOR and PRL-Vehicle, the best performance is achieved by pretraining with MAE on the biggest dataset (PRL-2.5M). We argue that the particularly high performance of ImageNet supervised pretraining on PRL-Aircraft might be due to the fact that, compared to PRL-Vehicle and DIOR, objects are closer to what can be found in ImageNet dataset, i.e., covering a large portion of the image.

As for in-domain baselines, both SatMAE and SatMAE++ are among the worst performing methods except for DIOR where SatMAE++ ranks in the middle near MoCo-v3 methods.

All other things being equal, using internal datasets such as PRL-363k and PRL-2.5M instead of fMoW seems to be more beneficial. However, the MAE pretraining on fMoW always ranks higher that MoCo-v3 on PRL-363k, and sometimes PRL-2.5M. In light of these results, we argue that the choice of the pretraining method is essential in order to fully exploit the benefits of large-scale pretraining datasets.

Finally, when comparing pretraining methods, we can observe that MAE always outperforms MoCo-v3 at comparable settings. It has also better scaling properties, as going from PRL-363k to PRL-2.5M, results in the highest gain in metrics for MAE, whereas MoCo-v3 only achieves small or nonexistent gains. This is also the case when going from ViT-Base to ViT-Large. From that, we conclude that scaling the backbone seems to have a greater impact than scaling the dataset.

C. Segmentation

Table IV shows the mean IoU for SpaceNetV1. First, comparing the ImageNet baselines with our own pretrained weights, we can see that the ImageNet baselines are competitive, especially the ImageNet supervised one. Indeed, this baseline is only outperformed by the MAE pretraining on PRL-2.5M for the ViT-Base backbone and the MAE pretraining for both PRL-363k and PRL-2.5M for the ViT-Large backbone.

Regarding in-domain baselines, SatMAE is the worst performing method, on the opposite of SatMAE++, which manages to rank second among all models for the ViT-Large backbone, making it a solid baseline.

TABLE IV
MEAN IOU ON THE SPACENETV1 SEGMENTATION DATASET

Backbone	Dataset	Method	SpaceNetV1
ViT-Base	-	Random init.	70.8
ViT-Base	IN	MoCo-v3	74.5
ViT-Base	IN	MAE	76.9
ViT-Base	IN	Sup.	78.4
ViT-Base	fMoW	MoCo-v3	77.0
ViT-Base	fMoW	MAE	76.0
ViT-Base	PRL-363k	MoCo-v3	77.4
ViT-Base	PRL-363k	MAE	75.9
ViT-Base	PRL-2.5M	MoCo-v3	77.7
ViT-Base	PRL-2.5M	MAE	**79.8**
ViT-Large	-	Random init.	72.2
ViT-Large	IN	MoCo-v3	74.4
ViT-Large	IN	MAE	76.9
ViT-Large	IN	Sup.	77.9
ViT-Large	fMoW	SatMAE	75.3
ViT-Large	fMoW	SatMAE++	79.0
ViT-Large	fMoW	MoCo-v3	77.0
ViT-Large	fMoW	MAE	77.1
ViT-Large	PRL-363k	MoCo-v3	76.2
ViT-Large	PRL-363k	MAE	78.8
ViT-Large	PRL-2.5M	MoCo-v3	75.7
ViT-Large	PRL-2.5M	MAE	**79.8**

When comparing our internal datasets with fMoW, we see that pretraining with PRL-363k or PRL-2.5M shows benefits over fMoW but mainly with MAE as pretraining with MoCo-v3 often results in a performance loss.

At last, when looking at pretraining methods, we observe that MAE performs better overall than MoCo-v3. Indeed, except for the ViT-Base backbone pretrained on PRL-363k, the performance of MAE is higher than MoCo-v3 in any other case.

V. DISCUSSION

In this section, we provide general insights with the aim of facilitating the choice of a pretraining methods. First, we question the benefits of a custom in-domain SSL pretraining over existing ImageNet pretrained weights. Table V shows the difference between several aggregations from our different downstream results. If not mentioned otherwise and applicable, all aggregations consider both backbones (ViT-Base and ViT-Large), both paradigms (MAE and MoCo-v3) and exclude results from SatMAE and SatMAE++, as they tend to achieve sub-par performance in our RGB setting.

The first column shows the average performance gap of using in-domain SSL pretraining over ImageNet baselines. It shows that in-domain SSL pretrained weights provide benefits in term of downstream performance for segmentation, but with mitigated results for classification and detection. We argue that the supervised pretraining on ImageNet, a classification task, can yield better representations for a classification downstream task. Considering these results, we recommend using already available ImageNet weights when working on classification, and potentially go for a dedicated pretraining with the MAE paradigm to try to push the performance further.

The second column shows the potential benefit of using PRL-363k over fMoW, knowing that they have the exact same number of images but that PRL-363k is of higher native resolution. We compare the difference of the average metrics for PRL-363k against fMoW, and show that there are no strong positive benefits for the classification task and close to no benefits for the segmentation and detection tasks, except for the aircraft outlier. We argue that the domain gap between PRL-363k, which contains high resolution images, and downstream task datasets, which contain a mix of resolutions closer to the ones in fMoW, can be responsible for this absence of significant benefits. Finally, we hypothesize that pretraining on fMoW should yield better downstream performance on fMoW as the model has already seen the data during the pretraining step.

The third column shows the average performance gap of pretraining on PRL-2.5M over fMoW, e.g. with a bigger and high resolution dataset. Results show that there are clear benefits using the PRL-2.5M, confirming that pretraining on larger datasets can be beneficial. Additional experiments could be dedicated to studying the impact of pretraining resolution on downstream performance by lowering the resolution of PRL-2.5M . Based on the results of this column and the previous one, we would recommend building an internal pretraining dataset only if its expected size is higher that publicly available datasets. Building an aggregate of various public datasets can also be an interesting alternative, as in [28].

The last column shows the performance gap from using MAE over MoCo-v3. Overall, MAE provides better performance than MoCo-v3, and the gap between MAE and MoCo-v3 grows larger using a ViT-Large backbone when using the same pretraining dataset. Thus, we argue that MAE should be favored as it scales positively on all downstream tasks and backbone sizes.

On another note, pretraining requires a significant amount of computing power to converge within few hours or days. During our experiments we have observed a large difference in terms of speed and memory usage, thus we choose to report the efficiency as a major criteria of evaluation, to compare the MoCo-v3 and MAE. Figure 2 shows that the time required to pretrain using one image sample is much higher with MoCo-v3. In addition, we note that the memory consumption significantly increases with MoCo-v3, forcing us to reduce the batch size (4 times smaller than MAE for ViT-Base, as discussed in Experimental Setup) and thus increasing the number of iterations for each epoch. We believe that MAE is the best compromise in terms of efficiency, reaching the best performance with a given compute budget.

In light of these results, we can draw the following conclusions:

- **Dedicated pretraining is beneficial for most downstream tasks** As shown in the results, using weights from a dedicated in-domain pretraining outperforms ImageNet baselines except for two downstream tasks (RESISC45 and PRL-Aircraft) when pretraining is performed with ViT-Base.

TABLE V
SUMMARY OF DOWNSTREAM TASK PERFORMANCE FOR ALL METHODS

Downstream task	Dataset	(fMoW, PRL-363k, PRL2.5M) vs baselines	PRL-363k vs fMoW*	PRL-2.5M vs fMoW*	MAE vs MoCo-v3
Segmentation	SpaceNetV1	+0.9	+0.3	+1.5	+1.4
Classification	RESISC45	0	-0.1	0	+0.19
	fMoW	+1.2	+0.25	+0.8	+1.3
Detection	DIOR	+0.2	+0.8	+1.1	+6.8
	Vehicle	+2.9	+0.1	+3.9	+5.1
	Aircraft	-4.1	+3.9	+1.4	+2.7

*Comparing average metric considering ViT-Base and ViT-Large backbones. SatMAE and SatMAE++ always excluded.

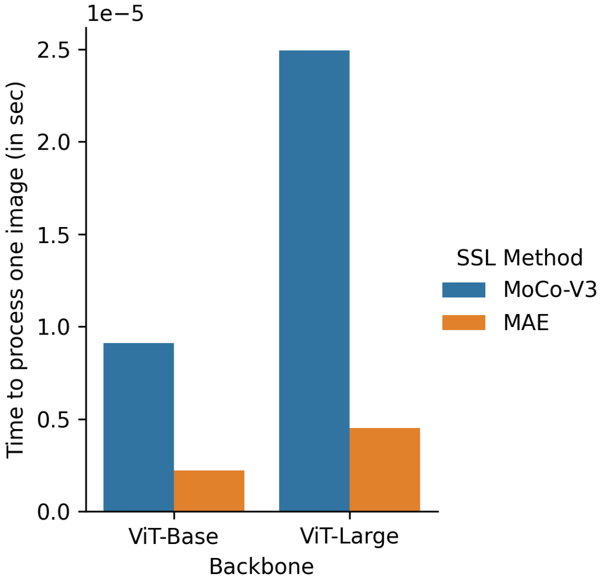

Fig. 2. Time required to process one image during the pretraining, depending on the SSL Method and backbone type. The measured time includes forward and backward passes.

- **MAE outperforms MoCo-v3 overall** MAE is able to consistently outperform MoCo-v3 while being computationally more efficient. Furthermore, it is able to scale better when using a bigger backbone or pretraining dataset.
- **Publicly available weights are solid competitors** Without any additional pretraining, using publicly available weights is enough to achieve competitive performance. When working with a limited budget, it seems fine to use ImageNet, SatMAE, or SatMAE++ pretrained weights. However, performing a dedicated pretraining remains interesting when pushing for the best performance on dense downstream tasks.

VI. Conclusion

In this paper, we study and compare the downstream performance of commonly used pretraining methods in the context of remote sensing imagery. We select a reference method for two families of self-supervised approaches and investigate their benefits on different downstream tasks. We pretrain MoCo-v3 and MAE on several datasets of different scale and composition and show that increasing the amount of pretraining data significantly improves the performance in downstream tasks. Experimental results also show that MAE is a strong competitor that achieves the best overall performance on the chosen downstream tasks while exhibiting better backbone scaling capabilities, and that using publicly available ImageNet weights is usually sufficient to achieve satisfactory performance. Finally, we show that pretraining methods that are specific to remote sensing are competitive alternatives, but do not manage to outperform more generic approaches by a large margin. Future directions of this work include expanding the number of benchmarked SSL methods, as well as studying the impact of GSD in the pretraining data. Furthermore, the study of other backbones may prove useful, as other architectures may yield different results. At last, methods to build datasets that best benefit the pretraining phase also constitute a promising direction of research.

Acknowledgements

This project was provided with computer and storage resources by GENCI at IDRIS thanks to the grant 2023-A0151014554 on the supercomputer Jean Zay's A100 and V100 partitions.

References

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019. [Online]. Available: http://arxiv.org/abs/1911.05722

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learmning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[3] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[4] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233024948

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[6] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, “Self-supervised learning in remote sensing: A review,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022.

[7] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon, “Geography-aware self-supervised learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 181–10 190.

[8] Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. Lobell, and S. Ermon, “SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 197–211, 2022.

[9] C. J. Reed, R. Gupta, S. Li, S. Brockman, C. Funk, B. Clipp, K. Keutzer, S. Candido, M. Uyttendaele, and T. Darrell, “Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4088–4099.

[10] M. Noman, M. Naseer, H. Cholakkal, R. M. Anwer, S. Khan, and F. S. Khan, “Rethinking transformers pre-training for multi-spectral satellite imagery,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 27 811–27 819.

[11] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, “Self-supervised learning in remote sensing: A review,” 2022. [Online]. Available: https://arxiv.org/abs/2206.13188

[12] I. Corley, C. Robinson, R. Dodhia, J. M. L. Ferres, and P. Najafirad, “Revisiting pre-trained remote sensing model benchmarks: Resizing and normalization matters,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024, pp. 3162–3172.

[13] K. Cha, J. Seo, and T. Lee, “A billion-scale foundation model for remote sensing images,” 2024. [Online]. Available: https://arxiv.org/abs/2304.05215

[14] X. Guo, J. Lao, B. Dang, Y. Zhang, L. Yu, L. Ru, L. Zhong, Z. Huang, K. Wu, D. Hu, H. He, J. Wang, J. Chen, M. Yang, Y. Zhang, and Y. Li, “Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery,” 2024. [Online]. Available: https://arxiv.org/abs/2312.10115

[15] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, “Functional map of the world,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6172–6180.

[16] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch sgd: Training imagenet in 1 hour,” *arXiv preprint arXiv:1706.02677*, 2017.

[17] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, K. Keutzer, and T. Darrell, “Self-supervised pretraining improves self-supervised pretraining,” *CoRR*, vol. abs/2103.12718, 2021. [Online]. Available: https://arxiv.org/abs/2103.12718

[18] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[19] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, “In-domain representation learning for remote sensing,” *arXiv preprint arXiv:1911.06721*, 2019.

[20] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, “SpaceNet: A remote sensing dataset and challenge series,” *arXiv preprint arXiv:1807.01232*, 2018.

[21] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” *arXiv preprint arXiv:2003.10555*, 2020.

[22] H. Bao, L. Dong, S. Piao, and F. Wei, “BEiT: BERT pre-training of image transformers,” *arXiv preprint arXiv:2106.08254*, 2021.

[23] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified perceptual parsing for scene understanding,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[24] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX.* Berlin, Heidelberg: Springer-Verlag, 2022, p. 280–296. [Online]. Available: https://doi.org/10.1007/978-3-031-20077-9_17

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.

[27] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[28] M. Mendieta, B. Han, X. Shi, Y. Zhu, and C. Chen, “Towards geospatial foundation models via continual pretraining,” 2023. [Online]. Available: https://arxiv.org/abs/2302.04476