# A. Optimization on Pretraining Data

## A.1. Performance with other choices of pretraining data.

To optimize GeoPile-2, we initially focused on optimizing GeoPile-2-RGB. As previous research has indicated, a successful pretraining dataset requires rigorous testing of each component [43]. Thus, we conducted a series of experiments on each individual dataset. These experiments involved the use of ImageNet [17] with 3 million images, GeoLifeCLEF with 3.3 million images, and the Functional Map of the World (FMoW) [14]. For FMoW, we segmented the dataset into tiles of size 384, leading to a total of 6 million images. This diverse selection of datasets allowed us to comprehensively test and optimize our pretraining approach for GeoPile-2.

To ensure other variables, such as the backbone architecture and pretraining methodologies, do not skew our results, we chose to employ the Swin-base [35] and committed to pretraining from scratch. In line with our aim for equitable comparison, we also adhered to the same seven downstream tasks as delineated in the previous report [39]. The results are shown in Table 7 and Table 8. This approach creates a consistent testing environment across all datasets, reducing the potential for bias or error.

Interestingly, upon integrating the GeoLifeCLEF into our testing framework, we observed a downturn in performance on downstream tasks. This result signifies that not all datasets necessarily contribute to improved model performance, and their selection demands careful consideration.

Even though the addition of both the Functional Map of the World dataset and ImageNet gave rise to performance metrics that were commensurate with those achieved by GeoPile-2-RGB, these new dataset additions were not as efficient. The key reason for this inefficiency was the significantly larger size of the pretraining dataset, which introduced higher computational costs and longer processing times. This finding highlights the importance of carefully balancing dataset size and complexity with computational efficiency in the model training process.

## A.2. Performance without RGB modality

RGB modalities are singled out because of the abundance of RGB datasets that come from various sources beyond just Sentinel-2. For instance, MillionAID [37], a dataset comprised of a wide range of RGB images, is sourced from multiple satellites, including GeoEye, WorldView, QuickBird, IKONOS, and SPOT satellites, among others. Additionally, a previous study [15] found that using only Sentinel-2 data for pretraining does not yield optimal performance in the downstream evaluation. Therefore, we sought to diversify our sources and include a wider range of RGB images in our pretraining data. This breadth of data sources significantly enriches the diversity of the RGB modality in our study.

Despite overlapping GSD in some RGB modality, more geospatial features will be included. Although these datasets may not provide an imaging spectrum as wide as Sentinel-2, they enhance the entropy of pre-training data, which has been proven to be effective in [39], which is demonstrated by Table 9.

# B. Pretraining Details

## B.1. Pretraining Settings

**Masking.** All hyper-parameters are listed in Table 10. We implement a masking strategy that maintains consistency around different channels within the same sensor, applying the mask at the same locations. However, when it comes to different sensors, we employ a varying masking approach, ensuring that the mask is applied at different locations. This methodology allows us to preserve sensor-specific information while investigating inter-sensor discrepancies effectively.

**Heterogeneous batch size.** Given the disparity in the number of images obtained from different sensors, we employ a heterogeneous batch size strategy for our training process. This methodology adjusts the batch size in proportion to the amount of data sourced from each individual sensor. In essence, during each epoch of our training process, every type of sensor is iterated through once, irrespective of the data volume associated with that particular sensor. This ensures that all sensor types have an equal chance to contribute to the model's learning process, fostering a more balanced and comprehensive training regimen. Alongside this, we also adjust the learning rate proportionally in accordance with the batch size allocated per sensor.

# C. Downstream Experiments

## C.1. Model size

Regarding the number of parameters, we followed a standard backbone for pretraining, the details of which have been reported in [63]. Comparisons between training from scratch and using ImageNet pretrained weights have been provided in Table 11 and corroborated by previous studies [8, 15, 38, 39, 62].

## C.2. Experimental settings

There are primarily two ways to leverage pretrained weights, as depicted in Figure 6. The first approach involves feeding each sensor through encoders that share weights. The resulting embeddings are then concatenated and fed into the classifier. In the second approach, all sensor data are stacked together in the color channel prior to patchification. This approach resembles the multiMAE method [5], where the projected patches from all modalities are concatenated

| Dataset | # Image | OSCD (F1) | DSFIN (F1) | BEN 10% | BEN 1% |
|---|---|---|---|---|---|
| GeoPile [39] | 600K | **57.5** | 66.2 | 86.4 | 79.3 |
| GeoPile-2-RGB | 1.7M | 57.1 | **70.4** | **86.8** | **79.6** |
| GeoPile-2-RGB + ImageNet [17] | 3M | **57.5** | 69.2 | 86.4 | 79.5 |
| GeoPile-2-RGB + GeoLifeCLEF | 3.3M | 56.1 | 61.6 | 86.1 | 78.9 |
| GeoPile-2-RGB + FMoW [14] | 6M | 58.2 | 69.3 | 86.2 | 79.1 |

Table 7. Results of downstream tasks with different pretraining datasets: change detection and classification

| Dataset | # Image | WHU | Vai. | SN2 (PSNR) | SN2 (SSIM) |
|---|---|---|---|---|---|
| GeoPile [39] | 600K | 90.1 | 75.1 | **22.626** | 0.645 |
| GeoPile-2-RGB | 1.7M | **90.6** | 75.9 | 22.599 | **0.658** |
| GeoPile-2-RGB + ImageNet [17] | 3M | 90.5 | **76.1** | 22.107 | 0.631 |
| GeoPile-2-RGB + GeoLifeCLEF | 3.3M | 89.1 | 74 | 16.663 | 0.512 |
| GeoPile-2-RGB + FMoW [14] | 6M | 90.2 | 75.7 | 22.448 | 0.638 |

Table 8. Results of downstream tasks with different pretraining datasets: segmentation and super-resolution

| Pretraining sensor modality | 10% BEN | cloud removal |
|---|---|---|
| Metric | mAP (↑) | SAM (↓)) |
| SAR (in Figure 3) | 84.2 | $8.37 \pm 0.057$ |
| Sentinel-2 (in Figure 3) | 86.6 | $8.33 \pm 0.034$ |
| RGB | 86.4 | $10.45 \pm 0.12$ |
| w/o RGB | 86.6 | $9.67 \pm 0.12$ |
| GeoPile-2 | **87.7** | **7.51** $\pm 0.057$ |

Table 9. Results pretrained with single modality or without RGB modality.

into a single sequence. Our experiments on 1% of the BEN dataset [56], listed in Table 12, demonstrate that both methods yield comparable results. However, the latter approach is more computationally efficient, meaning that the former approach takes longer time to reach the optimal performance and consumes more memory as well. Therefore, all results mentioned in the main text utilize this second approach. Importantly, in both cases, no masking is performed during the transfer phase.

### C.3. Visualization

We present some quantitative results of segmentation in Figure 7 respectively.

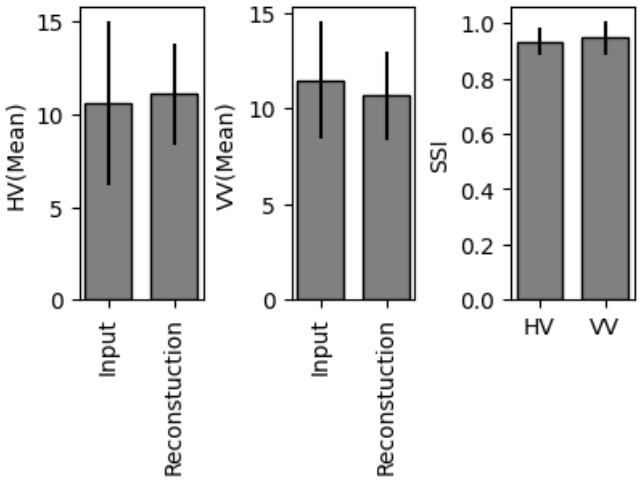


Figure 5. Figure R1: SAR backscatter statistics comparing input and reconstruction using the MIM. The two bands of SAR are HV and VV. The mean and standard deviation for the HV band are shown on the left, while those for the VV band are displayed on the right. The Speckle Suppression Index (SSI) values are presented in the right panel. An SSI value closer to one indicates that the mean and standard deviation remain consistent before and after reconstruction.

| Hyper-parameter | Value |
|---|---|
| Image size | $192 \times 192$ |
| Optimizer | AdamW |
| $\beta_1$ | 0.9 |
| $\beta_2$ | 0.999 |
| Eps | $1.0 \times 10^{-8}$ |
| Momentum | 0.9 |
| Weight decay | 0.05 |
| Learning rate | $\{1.0 \times 10^{-4}, 0.25 \times 10^{-4}, 1.0 \times 10^{-5}\}$ for RGB, Sen12MS [52] and MDAS [29] |
| Warm up learning rate | $5.0 \times 10^{-7}$ |
| Weight decay | $10^{-5}$ |
| Batch size | {128, 32, 12} per GPU for RGB, Sen12MS [52] and MDAS [29] |
| Training epochs | 800 or 100 |
| Warm up epochs | 10 |
| Learning rate decay | Multistep |
| Gamma | 0.1 |
| Multisteps | [700,] for 800 or [] for 100 |
| # Experts | 8 |
| MoE blocks | 1, 3, 5, 7, 9, 11, 13, 15, 17 (Every other swin block) |
| Top-value ($k$) | 1 |
| Capacity factor | 1.25 |
| Aux loss weight ($\lambda$) | 0.01 |
| Mask patch size | 32 |
| Mask ratio | 0.6 |

Table 10. Hyperparameters of msGFM pretraining.

| Model | SeCo | SatMAE | MoCoV2 | DINO-MC | GFM | msGFM |
|---|---|---|---|---|---|---|
| # of trainable parameters | 23M | 307M | 23M | 48.6M | 89M | 89M |

Table 11. Model size

| Finetuning Method | BEN 1% |
|---|---|
| 1 | 80.8 |
| 2 | 80.8 |

Table 12. Results of BEN when comparing different downstream transfer methods illustrated in Figure 6

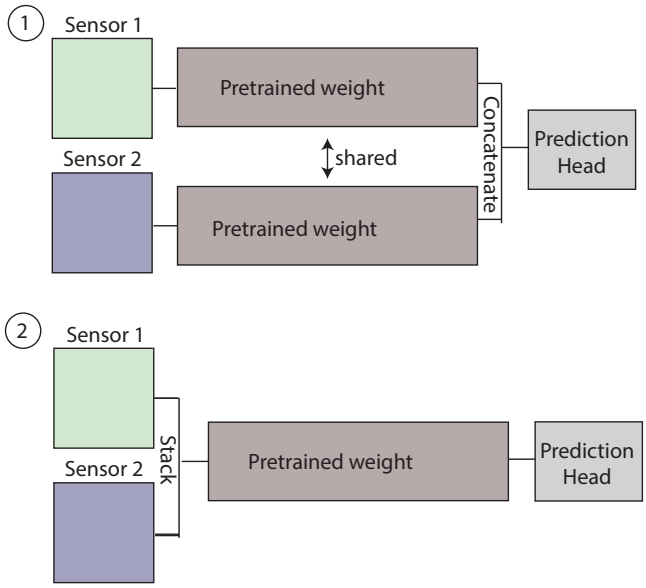


Figure 6. Two methods of downstream transfer. In the top panel, every sensor is fed into a separate encoder initialized with msGFM pretrained weight. The embeddings from the last layer are concatenated, and then fed through the prediction head, such as classifier and segmentation decoder. In lower panel, images are concatenated along the color channel and then fed through one encoder initialized with msGFM pretrained weight.

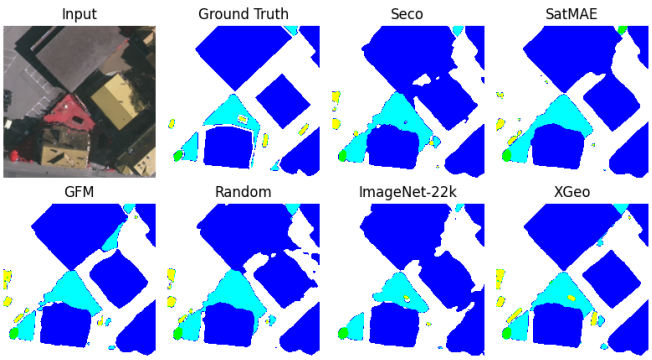


Figure 7. A display of qualitative results showcasing segmentation outcomes from msGFM in comparison to other competitive methods.